

# Netzbasierte Informationssysteme (WS 2004/05) Basiswissen World Wide Web

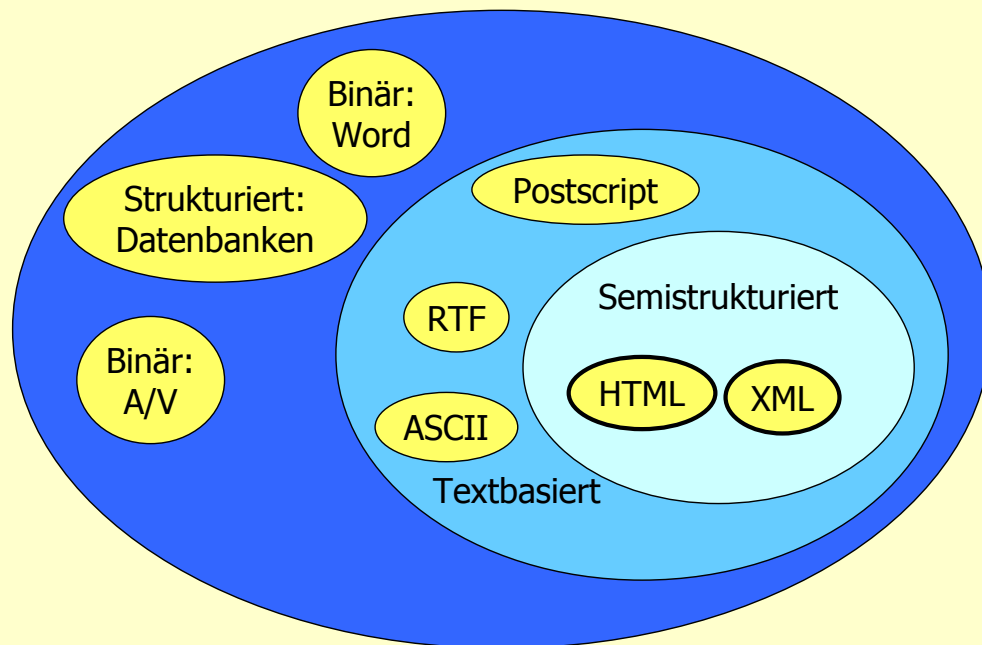
Robert Tolksdorf  
Freie Universität Berlin  
Institut für Informatik  
Netzbasierte Informationssysteme  
mailto:tolk@inf.fu-berlin.de  
http://www.robert-tolksdorf.de  
http://nbi.inf.fu-berlin.de

[1] © Robert Tolksdorf, Berlin

## Informationsquellen im Web

[2] © Robert Tolksdorf, Berlin

## Daten im Netz



[3] © Robert Tolksdorf, Berlin

## HTML

[4] © Robert Tolksdorf, Berlin

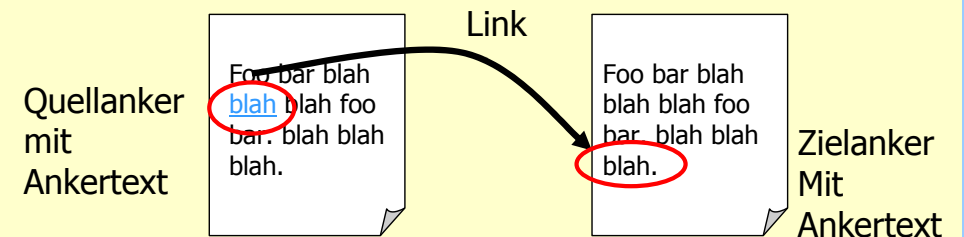
## Hypertext Markup Language

- Dominierende Sprache zur Auszeichnung von Dokumenten im Internet
- Definiert vom World Wide Web Consortium, W3C:
  - MIT
  - ERCIM
  - Keio University
- Jedes Informationssystem im Netz muss:
  - HTML Informationen integrieren können
  - HTML Ausgaben erzeugen
  - Mit HTML-Mitteln mit Nutzern interagieren

[5] © Robert Tolksdorf, Berlin

## Hypertext Markup Language

- Konzepte:
  - Informationen werden als Dokumente aufgefasst
  - Dokumenteninhalte werden als Klartext dargestellt
  - Dokumententeile werden durch Tags ausgezeichnet
    - Inhaltlich (<h1>Einleitung</h1>, <em>wichtig</em>)
    - Gestalterisch (<b>wichtig</b>)
  - Dokumente werden durch Links zu einem Hypertext verbunden (dem Web)



[6] © Robert Tolksdorf, Berlin

## HTML

- Sprache umfaßt
  - Elemente (wie <h1>)  
<h1>Neue Vorlesungen</h1>  
<br>  
<hr>
  - Attribute (wie bei <hr height="3">)
  - Entitäten (wie & ;)
  - Grammatikalische Regeln über Elemente (<html> ist Startsymbol, darin die Elemente <head> und <body>)

[7] © Robert Tolksdorf, Berlin

## HTML Beispiel/1

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html>
  <head>
    <title>FU-Berlin: Institut für Informatik</title>
    <base href="http://www.inf.fu-berlin.de">
  </head>
  <body>
    <p><a href="http://www.fu-berlin.de/">Freie Universität
    Berlin</a><br>
    <a href="http://www.math.fu-berlin.de/">Fachbereich Mathematik
    und Informatik</a></p>
    <h1>Institut für Informatik</h1>
    <p><a href="http://www.inf.fu-berlin.de/index_en.html">Homepage
    in English</a>.</p>
```

[8] © Robert Tolksdorf, Berlin

## HTML Beispiel/2

```
<form method="get" action="http://www.google.com/search">
<a href="http://www.google.de">Google</a>-Sitesearch:
  <nobr><font size=2>
<input type=text name=q size=15 maxlength=255 value="">
</font></nobr>
<input type=hidden name=sitesearch value="inf.fu-berlin.de">
<input type=hidden name=domains value="inf.fu-berlin.de">

</form>
```

<h2>Aktuelle Meldungen</h2>

```
<p>Das Ferienprojekt <a href=
"http://www.inf.fu-berlin.de/~block/schachprojekt.html">
Schachprogrammierung</a>
wird wegen der umfassenden Bauarbeiten im Institut auf die nächsten
Semesterferien verschoben!</p>
</body>
</html>
```

[9] © Robert Tolksdorf, Berlin

## HTML – Elemente für Struktur

- Struktur
  - !DOCTYPE gibt Art des Dokuments an:  
<!DOCTYPE HTML PUBLIC  
-//W3C//DTD HTML 4.01 Transitional//EN>
  - <html>...</html> umfaßt Dokument
  - <head>...</head> enthält Informationen zur Seite
  - <body>...</body> umfaßt Inhalt der Seite

- Festes Seitenschema:

```
<!DOCTYPE...>
<html>
  <head>...</head>
  <body>...</body>
</html>
```

[10] © Robert Tolksdorf, Berlin

## HTML – Elemente im Kopfteil

- <base> enthält Basis-Adresse der Seite
- <title> enthält Titel der Seite  
<title>FU-Berlin: Institut für Informatik</title>
- <meta> enthält
  - Inhaltsklassifikation der Seite  
<meta scheme="ISBN" name="identifier"  
content="0-8230-2355-9">
  - oder Protokollinformation  
<meta http-equiv="Expires"  
content="Tue 24 Sep 2002 00:00:00 GMT">
- <link> gibt Beziehung zu anderer Seite an  
<link rel="Glossary" href="glossar.html">

[11] © Robert Tolksdorf, Berlin

## HTML – Elemente für Gestaltung

- Umbruch, Trennungen: wbr br nobr p spacer  
Neue<br>Zeile
- Schriftarten  
b i s strike tt u blink bdo marquee  
Das ist <b>wirklich wichtig</b>!!
- Schriftauszeichnung  
abbr acronym cite code del dfn em ins  
kbd samp strong var ruby rt rb
- Formeln: sub sup
- Schriftgröße: basefont font big small

Das ist **wirklich wichtig**!!

[12] © Robert Tolksdorf, Berlin

## HTML – Elemente für inhaltliche Strukturen

- Überschriften h1 h2 h3 h4 h5 h6  
`<h2>Aktuelle Meldungen</h2>`
- Blöcke  
comment hr div span address pre xmp  
plaintext listing blockquote q banner  
multicol center  
`<center>Ein zentrierter Block<hr>mit  
einer Linie</center>`

Ein zentrierter Block

---

mit einer Linie

[13] © Robert Tolksdorf, Berlin

## HTML – Elemente für inhaltliche Strukturen

- Listen: ol ul dir menu li dl dt dd
- `<font SIZE=+1>Allgemeines</font> <p>  
<ul>  
<li><a HREF="/inst/ag-sek/index.html"  
>Institutsleitung</a>  
<li><a HREF=  
"/inst/lageplan.html">Lageplan</a>  
<li><a HREF=  
"http://www.math.fu-berlin.de/cgi-bin/telefon"  
>Telefonverzeichnis</a> Allgemeines  
<li><a HREF="/inst/announc  
>Votr&auml;ge</a>  
<li><a HREF=  
"http://www.math.fu-berlin.de/o  
>Frauenbeauftragte</a>  
</ul><p>`

- ♦ [Institutsleitung](#)
- ♦ [Lageplan](#)
- ♦ [Telefonverzeichnis](#)
- ♦ [Vorträge](#)
- ♦ [Frauenbeauftragte](#)

[14] © Robert Tolksdorf, Berlin

## HTML – Elemente für inhaltliche Strukturen

- Tabellen  
table th tr td thead tbody tfoot col  
colgroup
- `<table border="1">  
<tr><th align="center">Währung</th>  
<th align="center">1 EUR</th></tr>  
<tr><td>Deutschland (DEM)</td>  
<td align="right">1,95583</td></tr>  
<tr><td>Frankreich (FRF)</td>  
<td align="right">6,55957</td></tr>  
</table>`

- Abbildungen  
img overlay  
caption map area

Währung	1 EUR
Deutschland (DEM)	1,95583
Frankreich (FRF)	6,55957

[15] © Robert Tolksdorf, Berlin

## HTML – Elemente für Interaktion

- Formulare  
form input select option textarea  
htmlarea button label fieldset legend

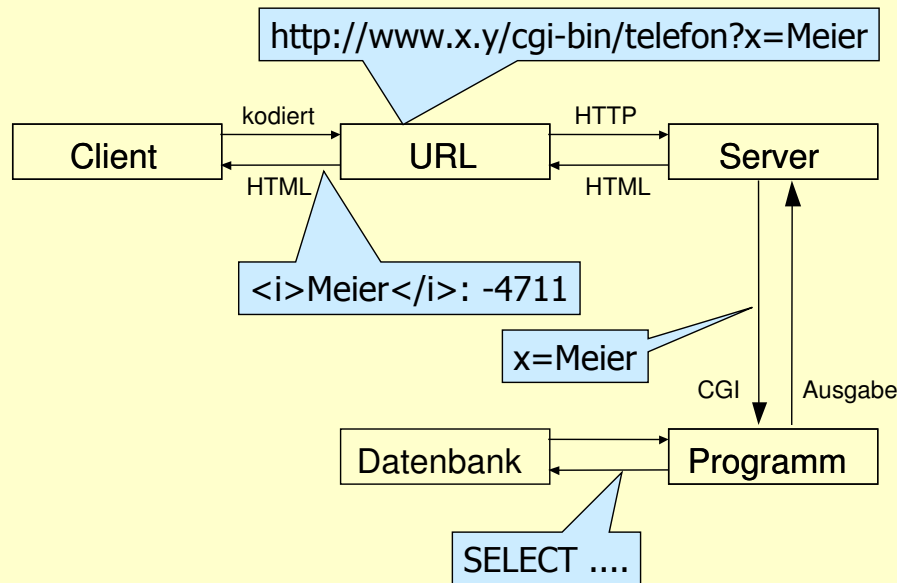
```
<form ACTION="/cgi-bin/telefon" METHOD="GET">  
<i>Die Nummer von</i>  
<input NAME="x" VALUE="" SIZE=30>  
<input TYPE="submit" VALUE="bitte">  
</form>
```

Die Nummer von

bitte

[16] © Robert Tolksdorf, Berlin

## Formularverarbeitung



[17] © Robert Tolksdorf, Berlin

## HTML – Elemente für komplexe Darstellung und Inhalte

- Browserdarstellung  
style frameset frame iframe noframes layer ilayer
- Applets, Skripte und Objekte  
applet param textflow script noscript object embed  
bodytext
- Hyperlinks  
a
  - Zielanker:  
`<a name="lokal">Zielankerbeschriftung</a>`
  - Quellanker:  
`<a href="Zielanker">Quellankertext</a>`  
`<a href="http://x.y.com/seite.html#lokal">Quellankertext</a>`

[18] © Robert Tolksdorf, Berlin

## Dokumentenadressen - URLs

- Uniform Resource Locator definiert eine Syntax für eindeutige Bezeichner im Internet
- Internet Dienste sind (zumeist) definiert durch
  - Aufgabe
  - Portnummer auf dem der Dienst angeboten wird
  - Transportprotokoll (TCP oder/und UDP)
  - Protokoll
- Z.B.: Web Dienst
  - Übertragen von HTML Seiten
  - Port 80
  - TCP
  - HTTP
- Z.B.: Usenet Dienst
  - Übertragen von News
  - Port 119
  - TCP
  - NNTP

[19] © Robert Tolksdorf, Berlin

## URL

- Uniform Resource Locators sind syntaktische Vereinheitlichung von Dienstbezeichnungen:  
`http://grunge.cs.tu-berlin.de:8000/`  
`ftp://ftp.cs.tu-berlin.de/pub/net/www`  
`mailto:tolk@cs.tu-berlin.de`
- Form:  
`http://grunge.cs.tu-berlin.de:8000/resource/data.html#top`  

The diagram shows the components of the URL 'http://grunge.cs.tu-berlin.de:8000/resource/data.html#top' with callout boxes pointing to each part: 'Protokoll' (http), 'Rechnername' (grunge.cs.tu-berlin.de), 'Portnummer' (:8000), 'Pfad' (/resource/data.html), and 'Referenz' (#top). The 'Pfad' and 'Referenz' boxes are grouped under a larger box labeled 'Resource'.
- Bedeutung ist von Protokoll abhängig, URL ist nur als Syntax definiert

[20] © Robert Tolksdorf, Berlin

## Information aus HTML-Seiten erschliessen

- Erschliessen des Hypergraphen selber
  - Crawling
- Erschliessen der Dokumente
  - HTTP Protokoll über Internet
- Erschliessen der Inhalte von Seiten
  - Extraktion aus HTML-Text
  - Nutzung der Informationen in Tags (<address>, <title>)
- Problem: Semantik der Inhalte
  - Wie extrahieren ("Produkt" etc.)
  - Markierung nutzen (<address>, <h\*> etc.)
  - Gestaltung (Bilder, Framesets etc.)

[21] © Robert Tolksdorf, Berlin

## XML: Sprache zur Definition von Auszeichnungssprachen

[22] © Robert Tolksdorf, Berlin

## Auszeichnungssprachen

- Auszeichnungssprachen fügen *Markierungen* zu einem Text hinzu
- Beispiel HTML:

```
<u>Robert Tolksdorf</u>  
<address>  
FU Berlin<br>  
Netzbasierte  
Informationssysteme<br>  
Takustr.9<br>  
D-14195 Berlin<br>  
</address>
```

```
Robert Tolksdorf  
FU Berlin  
Netzbasierte Informati  
Takustr.9  
D-14195 Berlin
```

- *Tags* haben *logische* oder *visuelle* Bedeutung

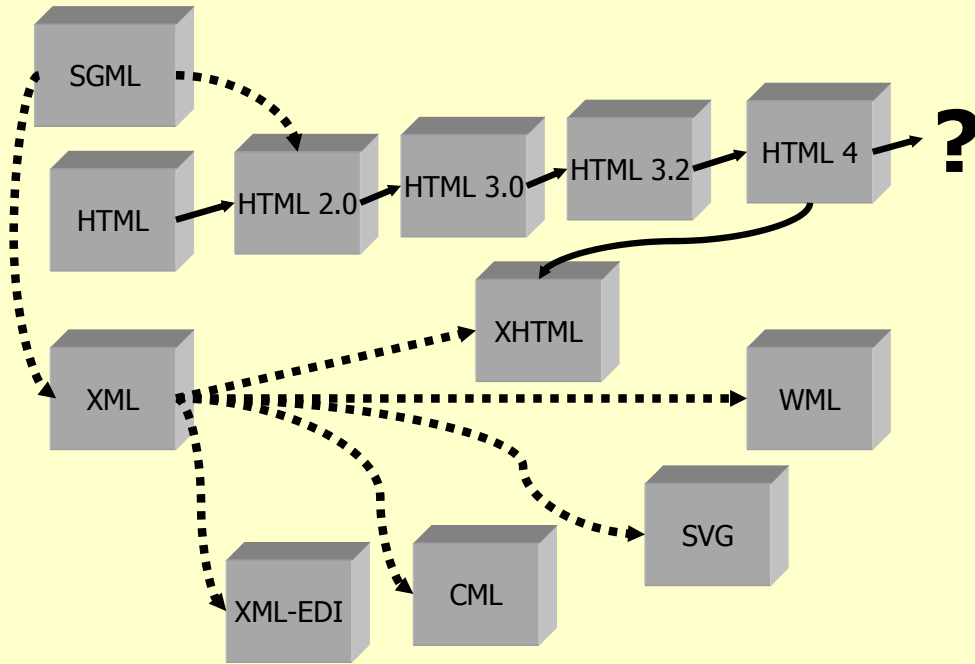
[23] © Robert Tolksdorf, Berlin

## Auszeichnungssprachen

- Kann es eine universelle Auszeichnungssprache geben?
  - Alle visuellen und sonstigen Möglichkeiten aller Ausgabegeräte müßten durch Tags steuerbar sein
  - Alle semantischen Konzepte aller Domänen müßten durch Tags repräsentierbar sein
  - Alle notwendigen Granularitäten der Auszeichnung müßten unterstützt werden:
    - <ADRESSE>...</ADRESSE>
    - <ADRESSE><STRASSE>...</STRASSE><ORT>...</ORT></ADRESSE>
    - <ADRESSE><STRASSE>...</STRASSE><ORT><PLZ>...</PLZ><ORTSNAME>...</ORTSNAME></ORT></ADRESSE>
- Nein: Anwendungsspezifische Auszeichnung nötig

[24] © Robert Tolksdorf, Berlin

## XML als Ergebnis der HTML Entwicklung

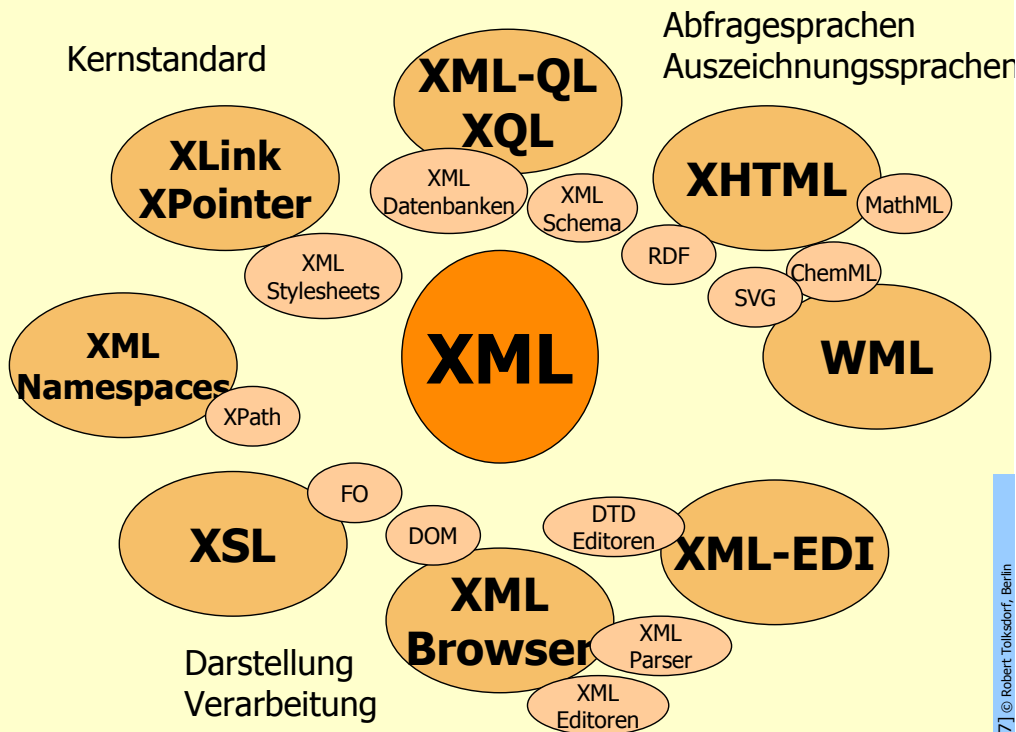


[25] © Robert Tolksdorf, Berlin

## XML Q&A

- *Was ist XML?*  
Die Extensible Markup Language ist die Definition einer Untermenge von SGML, mit der man einfach Auszeichnungssprachen definieren kann
- *Woher kommt XML?*  
XML ist ein Standard des World Wide Web Konsortiums W3C
- *Was macht man mit XML?*  
Anwendungsspezifische Auszeichnungssprachen definieren und standardisieren
- *Was ist der Vorteil von XML-basierten Auszeichnungssprachen?*  
Standardisierung ermöglicht Datenaustausch

[26] © Robert Tolksdorf, Berlin



[27] © Robert Tolksdorf, Berlin

XML basierte Auszeichnungssprachen

[28] © Robert Tolksdorf, Berlin

## Document Type Definition

- Eine XML-basierte Sprache wird durch eine XML-DTD (*Document Type Definition*) definiert
- Eine DTD enthält
  - Definitionen gültiger Elemente (Tags) der Sprache
  - Definitionen von Attributen der Elemente und deren Typen
  - Definitionen von Kürzeln (Entitäten)
  - Grammatikregeln

[29] © Robert Tolksdorf, Berlin

## Beispiel: AdressenML

- Elementdefinitionen und Grammatikregeln:  
`<?xml version="1.0" encoding="UTF-8"?>`  
*XML Direktive      Elementname      Grammatikregel*  
`<!ELEMENT ADRESSBUCH ANSCHRIFT*>`  
`<!ELEMENT ANSCHRIFT (NAME,`  
`(STRASSE | POSTFACH)?, ORT)>`  
`<!ELEMENT NAME ANY>`  
`<!ELEMENT STRASSE ANY>`  
`<!ELEMENT POSTFACH ANY>`  
`<!ELEMENT ORT EMPTY>`
- Beispiel:  
`<ADRESSBUCH><ANSCHRIFT><NAME>Robert  
Tolksdorf</NAME>  
<STRASSE>Franklinstr.28/29</STRASSE>  
<ORT PLZ="10587" NAME="Berlin"/>  
</ANSCHRIFT></ADRESSBUCH>`

[30] © Robert Tolksdorf, Berlin

## Beispiel: AdressenML

- Attributdefinitionen:  
`<!ATTLIST ORT`  
*Attributname      Typ      Optional/Mandatorisch*  
`PLZ    CDATA    #REQUIRED`  
`NAME    CDATA    #REQUIRED`  
`LAND    CDATA    #IMPLIED>`
- Weiter möglich: Defaultwerte
- Mögliche Datentypen:
  - **CDATA**: Zeichenkette
  - **ID**: Eindeutiger Bezeichner
  - **IDREF**: Referenz auf **ID**
  - Selbstdefinierte Token  
`<!ATTLIST PERSON GESCHLECHT (mann|frau) "frau">`

[31] © Robert Tolksdorf, Berlin

## Beispiel: AdressenML

- Kürzeldefinitionen:  
*Entitätsname      Expansionstext*  
`<!ENTITY TubStrasse "Franklinstr. 28/29" >`
- Im Dokument:  
`<STRASSE>&TubStrasse;</STRASSE>`
- Zusätzlich auch DTD-weite Kürzel

[32] © Robert Tolksdorf, Berlin

## Wohlgeformtheit und Validität

- Wohlgeformtheit:
  - Einhaltung der XML-Regeln, z.B.
    - Attributwerte in " eingeschlossen
    - Groß- und Kleinschreibung relevant
  - Alle Elemente sind geschlossen (Als Abkürzung: `<STRASSE></STRASSE> = <STRASSE/>`)
  - Korrekte Schachtelung von Elementen
- Validität:
  - Es gibt eine DTD zu einem XML-Dokument
  - Das XML-Dokument folgt den Regeln seiner DTD

[33] © Robert Tolksdorf, Berlin

## Mehrere XML-Sprachen in einem Dokument

- Dokumentenfragmente in unterschiedlichen Sprachen sind kombinierbar
- Möglichkeit 1: `<element xmlns="URN">` legt fest, daß `<element>...</element>` und alle Tags darin aus der durch URN bezeichneten XML-Sprache stammen sollen

- ```
<notes>
  <p xmlns="urn:w3-org-ns:HTML">
    This is a <i>funny</i> book!
  </p>
</notes>
```

Notizensprache  
HTML

[34] © Robert Tolksdorf, Berlin

## Mehrere XML-Sprachen in einem Dokument

- Möglichkeit 2: `<element xmlns:name="URN">` definiert ein Präfix `name`, das allen Tags aus der durch URN bezeichneten XML-Sprache vorangestellt werden
- ```
<notes xmlns:webml="urn:w3-org-ns:HTML"
  xmlns:math="urn:w3-org-ns:MathML">
  <webml:p>
    This is a <webml:i>funny</webml:i> book
    with <math:apply><math:power/>
    <math:ci>10</math:ci><math:ci>2</math:ci>
    </math:apply> pages!
  </webml:p>
</notes>
```

[35] © Robert Tolksdorf, Berlin

## Erschliessung von XML Dokumenten

- Klare formale Definition der Syntax und klare Forderung nach deren Einhaltung
- Tools müssen wohlgeformte Dokumente parsieren können
- Tools können anhand der DTD Validität prüfen
- Eigene Sprachen können formal sicher definiert werden

[36] © Robert Tolksdorf, Berlin

## XML-basierte Sprachen

[37] © Robert Tolksdorf, Berlin

## XML-basierte Sprachen

- Der eigentliche Gewinn durch XML entsteht durch die *Standardisierung* einer Sprache zur Definition von Auszeichnungssprachen
- Der eigentliche Gewinn durch domänenspezifische Auszeichnungssprachen ist ihre *Standardisierung*
- Standardisierung für
  - einheitliche Darstellung
  - Austauschbarkeit von Daten

[38] © Robert Tolksdorf, Berlin

## Formelsatz

- *Mathematical Markup Language MathML* ist Auszeichnungssprache für mathematische Formeln ([www.w3.org/TR/REC-MathML](http://www.w3.org/TR/REC-MathML))

- ```
<math>
  <apply>
    <sum/>
    <bvar><ci>i</ci></bvar>
    <lowlimit><cn>0</cn></lowlimit>
    <uplimit><cn>100</cn></uplimit>
    <apply>
      <power/>
      <ci>x</ci>
      <ci>i</ci>
    </apply>
  </apply>
</math>
```

$$\sum_{i=0}^{100} x^i$$

- Es existieren erste Browsererweiterungen und Autorenwerkzeuge

[39] © Robert Tolksdorf, Berlin

## Multimediapräsentationen

- *SMIL (Synchronized Multimedia Integration Language)* ist Auszeichnungssprache für Multimediapräsentation (<http://www.w3.org/TR/REC-smil>)
- ```
<seq>
  
  <par>
    <video src="film.avi" region="screen" dur="6s"/>
    <audio src="sound.wav" dur="6s"/>
  </par>
</seq>
```
- Erste Unterstützung in Playern (RealPlayer G2) und erste Autorenwerkzeuge

[40] © Robert Tolksdorf, Berlin

## Grafiken

- *Scalable Vector Graphics (SVG)* ist standardisiertes Zeichnungsformat (<http://www.w3.org/Graphics/SVG>)
- ```
<svg width="3in" height="3in">  
  <desc>A blue circle with a red outline</desc>  
  <g><circle cx="200" cy="200" r="100"  
    style="fill: blue; stroke: red"/></g>  
</svg>
```
- Erste Unterstützung in Browsern absehbar

[41] © Robert Tolksdorf, Berlin

HTML → XHTML

[42] © Robert Tolksdorf, Berlin

## Weiterentwicklung von HTML

- *XHTML* ist die Neuformulierung von HTML mit Hilfe einer XML-DTD (<http://www.w3.org/TR/xhtml1>)
- XHTML definiert einen XML Namensraum, in dem die bekannten Tags und Attribute von HTML 4 definiert sind
- Hauptsächlich syntaktische Änderungen wg. Anforderungen an Wohlgeformtheit
- Funktionale Ergänzungen des Sprachumfangs noch nicht
- Zukünftig: Modularisierung der Sprache des Web

[43] © Robert Tolksdorf, Berlin

## XHTML...

- ... ist die Weiterentwicklung von HTML auf XML Basis (Quelle: <http://www.w3.org/MarkUp/>)
- XHTML 1.0
  - Zweck: HTML 4 auf XML Basis
  - Status: W3C Recommendation 26 January 2000
  - Quelle: <http://www.w3.org/TR/xhtml1/>
- XHTML Basic
  - Zweck: Minimale Untermenge von XHTML
  - Status: W3C Recommendation 19 December 2000
  - Quelle: <http://www.w3.org/TR/xhtml-basic/>

[44] © Robert Tolksdorf, Berlin

## XHTML...

- Modularisation of XHTML
  - Zweck: Definition abstrakter Module von XHTML
  - Status: W3C Recommendation 10 April 2001
  - Quelle: <http://www.w3.org/TR/xhtml-modularization/>
- XHTML 1.1 - Module-based XHTML
  - Zweck: Definition konkreter Module von XHTML
  - Status: W3C Recommendation 31 May 2001
  - Quelle: <http://www.w3.org/TR/xhtml11/>

## XHTML 1.0

- Neuformulierung von HTML 4, keine wesentlichen Änderungen der Tag-Menge
- XHTML Dokumente müssen im XML-Sinn wohlgeformt also korrekt geschachtelt sein
- Öffnenden Elemente müssen geschlossen werden (`<p>!`)
- Alle Element- und Attributnamen sind in Kleinschreibung
- Attributwerte müssen in Anführungszeichen stehen (z.B. `start="1"`)
- Attribute müssen immer einen Wert tragen (z.B. `checked="checked"`)
- Leere Tags entsprechen XML markiert: `<br/>`
- `<script>` und `<style>` enthalten #PCDATA, deshalb werden `<`, `&` und Kürzel (`&lt;`;) interpretiert
- Fragmente verweisen immer auf das `id`-Attribut

## XHTML Basic

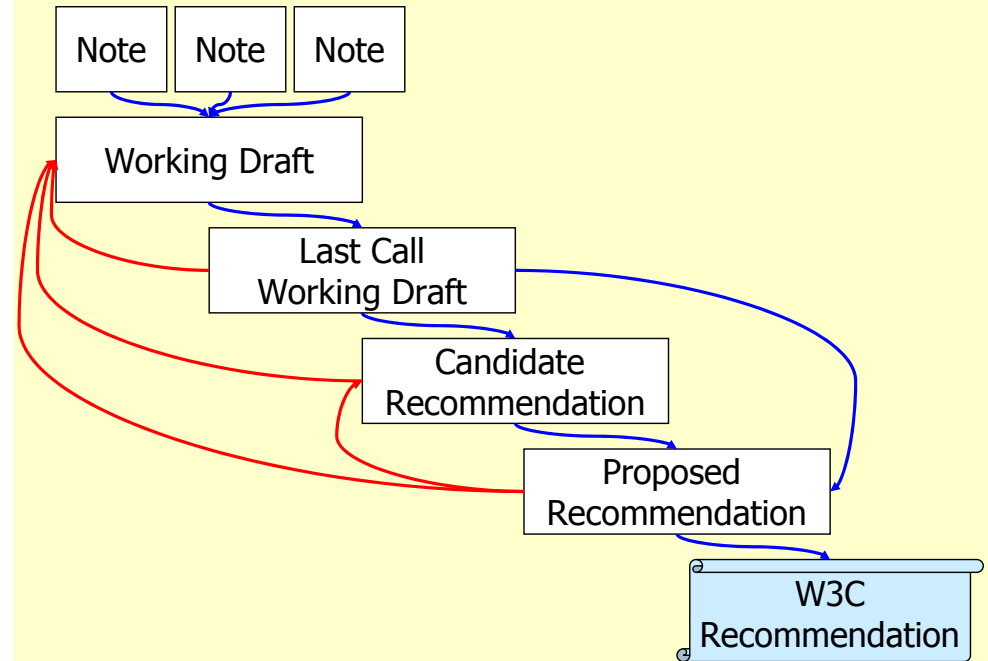
- XHTML für kleine Geräte
- Konkurrenz zu WML/WAP, Konvergenz mit WAP 2.0
- Umfang:
  - Structure Module: `body`, `head`, `html`, `title`
  - Text Module: `abbr`, `acronym`, `address`, `blockquote`, `br`, `cite`, `code`, `dfn`, `div`, `em`, `h1`, `h2`, `h3`, `h4`, `h5`, `h6`, `kbd`, `p`, `pre`, `q`, `samp`, `span`, `strong`, `var`
  - Hypertext Module: `a`
  - List Module: `dl`, `dt`, `dd`, `ol`, `ul`, `li`
  - Basic Forms: `form`, `input`, `label`, `select`, `option`, `textarea`
  - Basic Tables Module: `caption`, `table`, `td`, `th`, `tr`
  - Image Module: `img`
  - Object Module: `object`, `param`
  - Metainformation Module: `meta`
  - Link Module: `link`
  - Base Module: `base`

## XHTML 1.1

- Führt keine neuen Elemente ein
- Umfaßt alle Module aus XHTML Modularization
- Ist zweites Profil neben XHTML Basic
- Praktisch zunächst kaum Auswirkungen

## W3C Dokumente

## Lifecycle The W3C Recommendation Track



## Dokumentenstatus

- Note
  - Arbeitsnotiz
- Working Draft
  - Arbeitspapier einer Arbeitsgruppe
  - Stark veränderlich
  - Kein Bestätigung des aktuellen Arbeitsstands
  - Erklärte Absicht zur Erstellung eines Standards
- Last Call Working Draft
  - Arbeitspapier einer Arbeitsgruppe
  - Arbeitsgruppe hält Ergebnis für Lösung der Arbeitsaufgabe

## Dokumentenstatus

- Candidate Recommendation
  - Entsteht aus Last Call Working Draft durch
    - Veröffentlichung
    - Nachfrage von Implementierungserfahrungen
- Proposed Recommendation
  - Entsteht aus Candidate Recommendation bei hinreichender Implementierungserfahrung (und deren Berücksichtigung)
- W3C Recommendation
  - Ergebnis einer Konsensbildung über Proposed Recommendation
  - Web Standard

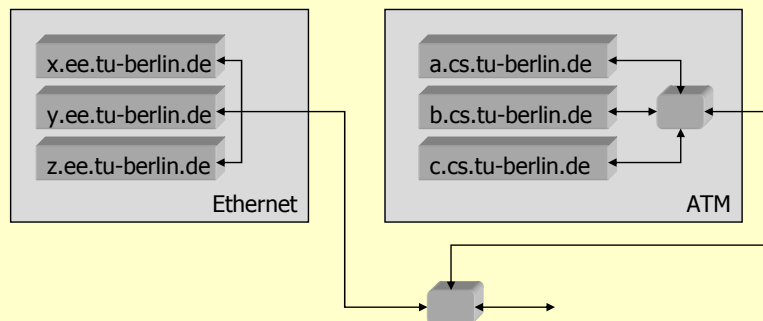
## Internet

## Was ist das Internet

- Eine weltweiter *Verbund von Rechnern*, die über Netze Daten austauschen können.
  - Hardware-bezogene Sicht
  - Zusammenschalten von lokalen Netzen zum Internet
  - Dabei notwendige Verarbeitung von Datenpaketen
- Eine *Protokollfamilie*
  - Netzbezogene Sicht
  - Protokollspezifikationen
- Ein *offenes System*, in dem Dienste genutzt und angeboten werden können.
  - Nutzungs- und anwendungsbezogen
  - Beschreibt die Anwendungsmöglichkeiten des Internet

## Internet als vernetzter Rechnerverbund

- Das Internet Protokoll IP ermöglicht Internetworking durch Etablierung eines Datenformats und Transportprotokollen, die auf unterschiedlichen Datenverbindungen aufgesetzt werden können



## Enveloping / Encapsulating

- Ethernet:

Dest	Source	Type	Data	CRC
------	--------	------	------	-----

  - IP Header | IP Data
  - TCP Header | TCP Data
  - FTP Header | FTP Data
- IEEE802.3:

Dest	Source	Len	LCC	SNAP	Data	FCS
------	--------	-----	-----	------	------	-----

  - IP Header | IP Data
  - TCP Header | TCP Data
  - FTP Header | FTP Data
- Fragmentation / Reassembly von IP Paketen

## IP Adressen

- Aktuell 32 Bit: eg. 130.149.27.12
- Abbildung je nach Medium auf die MAC (Media Access Control), die physikalische Netzadresse
  - ARP, RARP
- Netzwerkmaske definiert, was im lokalen Netz ist, und was nach außen geht
- Netzmaske 255.255.0.0 ->
  - 130.149.0.0 bis 130.149.255.255 sind lokales Netz
  - alles andere muß über einen Router laufen
- Routing: Weiterleiten von Paketen in andere Netzwerke

[57] © Robert Tolksdorf, Berlin

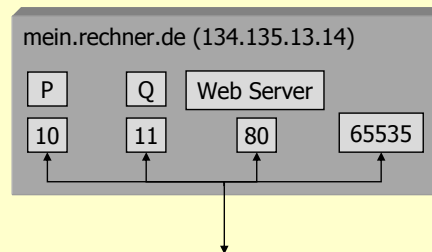
## IP Namen

- Internetadresse (IP Adresse) bezeichnet einen Rechner eindeutig
  - als Nummer  
130.149.27.12
  - als Name  
[grunge.cs.tu-berlin.de](http://grunge.cs.tu-berlin.de)
- Dienste wie DNS (=Domain Name Service) bilden Namen und Nummern aufeinander ab

[58] © Robert Tolksdorf, Berlin

## Transport Protokolle

- Drei Protokolle zum Datentransport
  - *UDP*: Ein Paket (Datagramm) von Rechner A nach Rechner B schaffen
  - *TCP*: Pakete werden *geordnet* und *zuverlässig* über eine Verbindung transportiert
- Ports als Kommunikationsadresse
  - Ein *Port* ist ein logischer Netzanschluß, benannt von 0 bis 65535
- *Socket* ist Endpunkt einer Verbindung



[59] © Robert Tolksdorf, Berlin

## Sockets

- Sockets sind die Kommunikationsendpunkte einer Internet-Verbindung
- Die Server Seite:
  - Ein Prozeß „lauscht“ auf einem Rechner an einem Port auf Verbindungswünsche („listen“)
  - Bei einem Verbindungswunsch erzeugt er einen Kommunikationssocket („accept“)
  - Der Kommunikationssocket hat eine andere Nummer als der Verbindungswunschsocket!
- Die Client Seite:
  - Melden des Verbindungswunsches („connect“)
  - „Einstecken“ in den Kommunikationssocket
- Protokollkommunikation:  
Zeichen über Kommunikationssocket schicken

[60] © Robert Tolksdorf, Berlin

## Internet als Protokollfamilie

- *Request For Comments*-Dokumente (RFC) definieren alle technischen Aspekte des Internet
- RFC 1738 :  
T. Berners-Lee, L. Masinter, und M. McCahill. Uniform Resource Locators (URL). RFC 1738, Internet Engineering Task Force, December 1994.
- Internet Engineering Taskforce IETF erstellt RFCs <http://www.ietf.org/rfc.html>
- Standardisierungsprozeß ist als RFC standardisiert: The Tao of IETF: A Novice's Guide to the Internet Engineering Task Force, RFC 3160, August 2001

[61] © Robert Tolksdorf, Berlin

## IETF Arbeitsfelder (7/02)

- Applications Area
- General Area
- Internet Area
- Operations and Management Area
- Routing Area
- Security Area
- Sub-IP Area
- Transport Area

[62] © Robert Tolksdorf, Berlin

## IETF Workinggroups Internet Area (7/02)

atommib	AToM MIB
dhc	Dynamic Host Configuration
dnsext	DNS Extensions
idn	Internationalized Domain Name
ifmib	Interfaces MIB
ipcdn	IP over Cable Data Network
ipoib	IP over InfiniBand
iporpr	IP over Resilient Packet Rings
ipv6	IP Version 6 Working Group
itrace	ICMP Traceback
l2tpext	Layer Two Tunneling Protocol Extensions
magma	Multicast & Anycast Group Membership
mobileip	IP Routing for Wireless/Mobile Hosts
pana	Protocol for carrying Authentication for Network Access
pppext	Point-to-Point Protocol Extensions
zeroconf	Zero Configuration Networking

[63] © Robert Tolksdorf, Berlin

## Ausgewählte Standards und RFCs

Protokoll	Beschreibung	RFCs	STD
	Internet Official Protocol Standards	1880	1
	Assigned Numbers	1700	2
	Host Requirements - Communications	1122	3
	Host Requirements - Applications	1123	3
IP	Internet Protocol	791	5
IP	Subnet Extension	950	5
IP	Broadcast Datagrams	919	5
IP	Broadcast Datagrams with Subnets	922	5
ICMP	Internet Control Message Protocol	792	5
IGMP	Internet Group Multicast Protocol	1112	5
UDP	User Datagram Protocol	768	6
TCP	Transmission Control Protocol	793	7
TELNET	Telnet Protocol	854, 855	8
FTP	File Transfer Protocol	959	9

[64] © Robert Tolksdorf, Berlin

## Ausgewählte Standards und RFCs

Protokoll	Beschreibung	RFCs	STD
SMTP	Simple Mail Transfer Protocol	821	10
SMTP-SIZE	SMTP Service Ext for Message Size	1870	10
SMTP-EXT	SMTP Service Extensions	1869	10
MAIL	Format of Electronic Mail Messages	822	11
CONTENT	Content Type Header Field	1049	11
NTPV2	Network Time Protocol, Version 2	1119	12
DOMAIN	Domain Name System	1034, 1035	13
DNS-MX	Mail Routing and the Domain System	974	14
SNMP	Simple Network Management Protocol	1157	15
SMI	Structure of Management Information	1155	16
Concise-MIB	Concise MIB Definitions	1212	16
MIB-II	Management Information Base-II	1213	17
NETBIOS	NetBIOS Service Protocols	1001, 1002	19
ECHO	Echo Protocol	862	20

[65] © Robert Tolksdorf, Berlin

## Ausgewählte Standards und RFCs

Protokoll	Beschreibung	RFCs	STD
DISCARD	Discard Protocol	863	21
CHARGEN	Character Generator Protocol	864	22
QUOTE	Quote of the Day Protocol	865	23
USERS	Active Users Protocol	866	24
DAYTIME	Daytime Protocol	867	25
TIME	Time Server Protocol	868	26
TFTP	Trivial File Transfer Protocol	1350	33
RIP	Routing Information Protocol	1058	34
TP-TCP	ISO Transport Service on top of the TCP	1006	35
ETHER-MIB	Ethernet MIB	1643	50
PPP	Point-to-Point Protocol (PPP)	1661	51
PPP-HDLC	PPP in HDLC Framing	1662	51
IP-SMDSIP	Datagrams over the SMDS Service	1209	52

[66] © Robert Tolksdorf, Berlin

## Internet-Protokolle und -Dienste

- Einordnung von Internet-Protokollen:

SMTP	NNTP	Finger	HTTP	FTP	SNMP	telnet	RTP	::	::	::	::	Dienstprotokolle
------	------	--------	------	-----	------	--------	-----	----	----	----	----	------------------

UDP	TCP	Multicast	Transportprotokolle
-----	-----	-----------	---------------------

IP	ICMP	Netzverbindungsprotokolle
----	------	---------------------------

Lokale Netze (Ethernet, ISDN, ATM, etc.)	Netzprotokolle
------------------------------------------	----------------

[67] © Robert Tolksdorf, Berlin

## Internet als dienstorientiertes offenes System

- Internet Dienste sind (zumeist) definiert durch
  - Aufgabe
  - Portnummer auf dem der Dienst angeboten wird
  - Transportprotokoll (TCP oder/und UDP)
  - Protokoll
- Z.B.: Web Dienst
  - Übertragen von HTML Seiten
  - Port 80
  - TCP
  - HTTP
- Z.B.: Usenet Dienst
  - Übertragen von News
  - Port 119
  - TCP
  - NNTP

[68] © Robert Tolksdorf, Berlin

## Beispiel: HTTP Protokoll



[69] © Robert Tolksdorf, Berlin

## HTTP (Überblick)

[70] © Robert Tolksdorf, Berlin

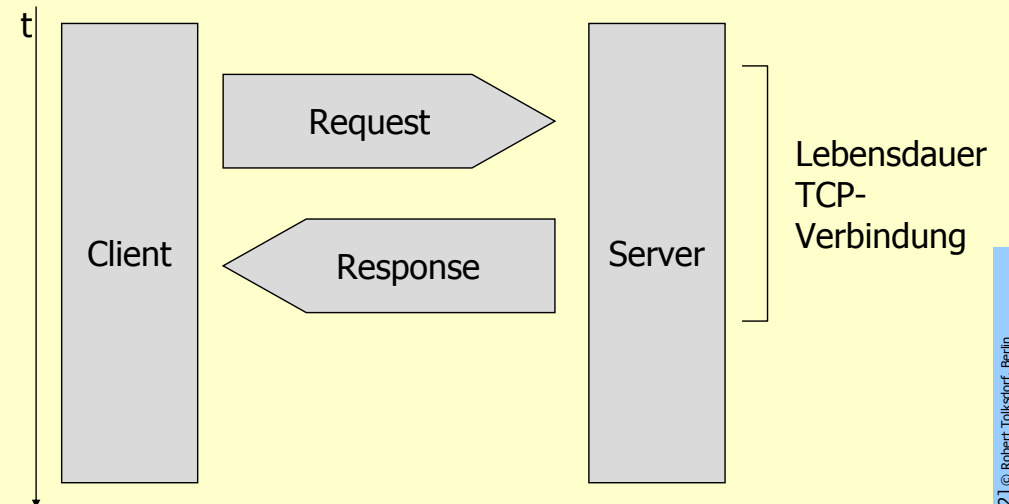
## Hypertext Transfer Protocol

- Aufgabe:  
Transfer von Informationen zwischen Web-Servern und Clients
- Port:  
80 ist für HTTP reserviert
- Transportprotokoll:  
TCP (leider)
- Protokoll:  
R. Fielding, J. Gettys, J. Mogul, H. Frystyk, L. Masinter, P. Leach und T. Berners-Lee. *Hypertext Transfer Protocol - HTTP/1.1*. RFC 2616, <http://www.w3.org/Protocols/rfc2616/rfc2616.txt>

[71] © Robert Tolksdorf, Berlin

## HTTP

- Zustandsloses Protokoll
- Request mit Response beantwortet



[72] © Robert Tolksdorf, Berlin

## Aufbau Request

- Request besteht aus
  - Request
  - Request-Beschreibung durch Header
    - Allgemeine Beschreibungen
    - Request-spezifische Beschreibungen
    - Beschreibung eventuell beiliegenden Inhalts
- Beispiel:

```
GET / HTTP/1.0
Connection: Keep-Alive
User-Agent: Mozilla/3.04Gold (Win95; I)
Host: megababe.isdn:80
Accept: image/gif, image/jpeg, image/pjpeg, */*
```

[73] © Robert Tolksdorf, Berlin

## Requests in HTTP

- Format: *Methode URL HTTP/x.y*
- GET
  - Anforderung einer Informationseinheit vom Server
  - Um `http://www.w3.org/Style/CSS/` zu holen, geht `GET / Style/CSS/ HTTP/1.0` an den Server `www.w3.org`
- HEAD
  - Anforderungen der Informationen über eine Informationseinheit ohne senden der Informationseinheit
- PUT
  - Abspeichern einer Informationseinheit auf einem Server
- POST
  - Hinzufügen von Informationen zu einer Informationseinheit
- DELETE
  - Löschen einer Informationseinheit auf einem Server

[74] © Robert Tolksdorf, Berlin

## Allgemeine Header

- Date: Datum  
Datum und Uhrzeit des Abschickens der
- MIME-Version: *x.y*  
Inhalt nach dem MIME-Standard in der Version *x.y* zusammengesetzt wurde
- X-...  
Zusätzliche, nicht standardisierte Header

[75] © Robert Tolksdorf, Berlin

## Request Header

- Accept: Medienart/Variante; q=Qualität; mxb=Maximale Größe
  - Accept: text/postscript; mxb=200000
- Accept-Charset: Zeichensatz

ISO-8859-1	ISO-8859-2	ISO-8859-3
ISO-8859-4	ISO-8859-5	ISO-8859-6
ISO-8859-7	ISO-8859-8	ISO-8859-9
ISO-2022-JP	ISO-2022-JP-2	ISO-2022-KR
UNICODE-1-1	UNICODE-1-1-UTF-7	UNICODE-1-1-UTF-8
US-ASCII	...	

[76] © Robert Tolksdorf, Berlin

## Request Header

- Accept-Encoding: Kodierung

Binäre Daten	binary	Inhaltskodierung
8-Bit-Daten	8bit	
7-Bit-Daten	7bit	
uuencode-kodiert	quoted-printable	
base64-kodiert	base64	
...		Transfer-codierung
gzip-komprimiert	gzip	
compress-komprimiert	compress	
...		

- Accept-Language: Sprachkürzel
  - Accept-Language: de, en

## Request Header

- If-Modified-Since: *Datum*  
Inhalt einer Informationseinheit nur dann mit Response schicken, wenn nach *Datum* modifiziert
- From: *Absender*  
Nutzer (selten genutzt)
- User-Agent: *Produkt/Version*  
Browser Identifikation (meistens...)
- Referer: *URL*  
Seite auf der ein Link auf die angeforderte Seite stand
- Inhaltsheader: Siehe Response

## Aufbau Response

- Response besteht aus
  - Antwort-Code
  - Response-Beschreibung durch Header
    - Allgemeine Beschreibungen
    - Response-spezifische Beschreibungen
    - Beschreibung eventuell beiliegenden Inhalts
- Beispiel:

```
HTTP/1.0 200 OK
Last-Modified: Sun, 15 Mar 1998 11:26:50 GMT
MIME-Version: 1.0
Date: Fri, 20 Mar 1998 16:43:11 GMT
Server: Roxen-Challenger/1.2beta1
Content-type: text/html
Content-length: 2990
```

```
<HTML><HEAD><TITLE>TU Berlin ---
```

## Antwort Codes

- 200-er Codes: Erfolgreiche Ausführung
  - 200 - OK  
Erfolgreich, Antwort anbei
  - 201 – Created  
Erfolgreiches PUT oder POST
  - 202 – Accepted  
Für spätere Ausführung vermerkt
  - 204 - No Content  
Erfolgreich, kein Antwortinhalt notwendig
  - ...

## Antwort Codes

- 300-er Codes: Weitere Aktion des Client zur erfolgreichen Ausführung notwendig
  - 300 - Multiple Choices  
Verschiedene Versionen erhältlich, Accept-Header nicht eindeutig
  - 301 - Moved Permanently  
Verschoben (Location und URI Header geben Auskunft)
  - 302 - Moved Temporarily  
Verschoben (Location und URI Header geben Auskunft)
  - 304 - Not Modified  
Bei GET mit If-Modified-Since
  - ...

[81] © Robert Tolksdorf, Berlin

## Antwort Codes

- 400-er Codes: Nicht erfolgreich, Fehler bei Client
  - 400 - Bad Request  
Falsche Request Syntax
  - 401 - Unauthorized  
Passwort notwendig
  - 403 – Forbidden  
Ohne Angabe von Gründen verweigert
  - 404 - Not Found  
Nicht auffindbar
  - 405 - Method Not Allowed  
z.B. PUT oder DELETE wird nicht akzeptiert
  - 406 - None Acceptable  
Information vorhanden aber nicht passend zu Accept-Headern
  - 408 - Request Timeout  
Timeout bei Übermittlung der Requests
  - ...

[82] © Robert Tolksdorf, Berlin

## Antwort Codes

- 500-er Codes: Nicht erfolgreich, Fehler bei Server
  - 500 - Internal Server Error
  - 501 - Not Implemented  
Angeforderte Methode nicht unterstützt
  - 502 - Bad Gateway  
Weiterer benutzer Server nicht erreichbar
  - 503 - Service Unavailable  
Server kann Dienst gerade nicht erbringen (Retry-After Header)
  - 504 - Gateway Timeout  
Weiterer benutzer Server antwortet nicht rechtzeitig
  - ...

[83] © Robert Tolksdorf, Berlin

## Response Header

- Server: *Produkt*  
Server: CERNb-HTTPD/3.0 libwww/2.17
- Retry-After: *Datum*  
Bei 503 Antwort Code
- ...

[84] © Robert Tolksdorf, Berlin

## Inhalts Header

- Content-Type: Medienart  
Content-type: text/html
- Content-Length: Länge
- Content-Encoding: Kodierung
- Content-Transfer-Encoding: Kodierung
- Content-Language: Sprachkürzel
- Expires: Datum  
Kann nach Datum aus Caches gelöscht werden
- Last-Modified: Datum  
Letzte Änderung
- Allow: Methoden  
Bei 405 Antwort Code
- X-...  
Zusätzliche, nicht standardisierte Header
- ...

[85] © Robert Tolksdorf, Berlin

## FTP

[86] © Robert Tolksdorf, Berlin

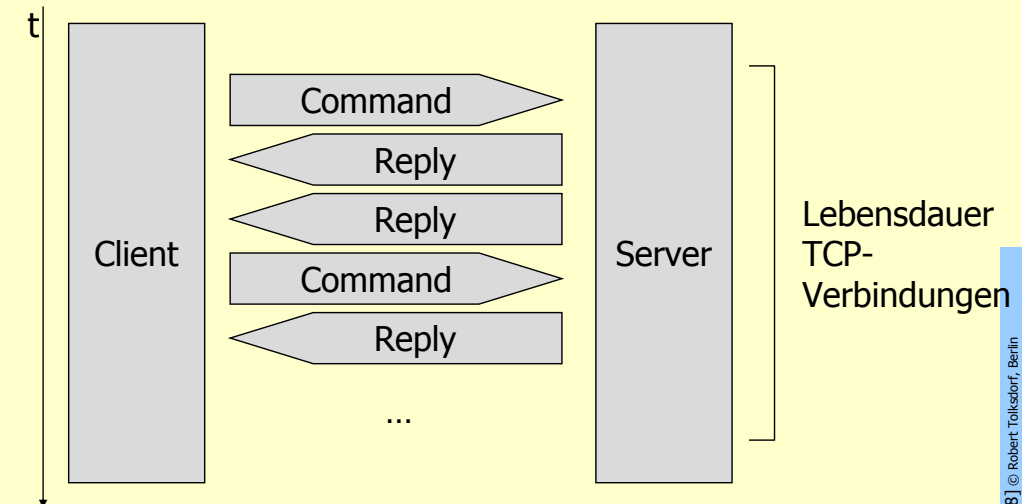
## File Transfer Protocol

- Aufgabe:  
Transfer von Dateien zwischen FTP-Servern und Clients
- Ports:  
20 ist für FTP Kontrollverbindung reserviert  
21 ist für FTP Datenverbindung reserviert
- Transportprotokoll:  
TCP
- Protokoll:  
J. Postel und J. Reynolds. *FILE TRANSFER PROTOCOL (FTP)*, Oktober 1985. RFC 959,  
<http://www.w3.org/Protocols/rfc2616/rfc2616.txt>

[87] © Robert Tolksdorf, Berlin

## FTP

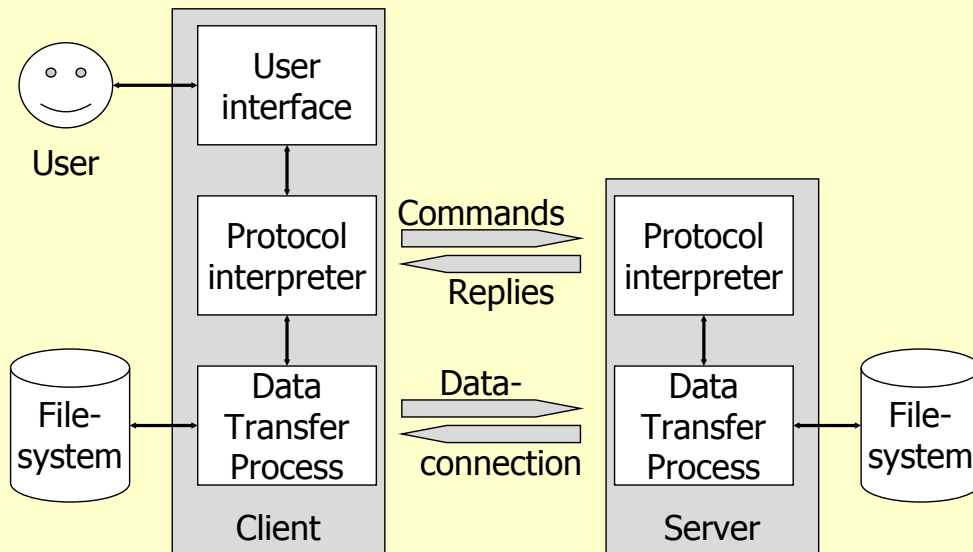
- Zustandshaltiges Protokoll
- Request mit Response beantwortet



[88] © Robert Tolksdorf, Berlin

## FTP

### Modell:



[89] © Robert Tolksdorf, Berlin

## Beispielsitzung

```
Connected to caramba.
220 ftp.cs.tu-berlin.de FTP server ready.
Name (ftp:tolk): ---> USER tolk
331 Password required for tolk.
---> PASS *****
230 User tolk logged in.
ftp> ---> PORT 130,149,17,167,185,53
200 PORT command successful.
---> LIST
150 Opening ASCII mode data connection for /bin/ls.
total 33264
drwxr-xr-x 52 tolk   flp       8704      Jul 22 17:20 .
drwxr-sr-x 44 root   root     2560      Jun 25 14:23 ..
-rw-r--r--  1 tolk   flp     164352    Jul 16 09:22 NBI.ppt
[...]
226 Transfer complete.
12491 bytes received in 0.13 seconds (97.37 Kbytes/s)
```

Login/Passwort-Eingabe

Eingabe "dir"

[90] © Robert Tolksdorf, Berlin

## Beispielsitzung

```
ftp> ---> PORT 130,149,17,167,185,54
200 PORT command successful.
---> RETR test
150 Opening ASCII mode data connection for test (6 bytes).
226 Transfer complete.
local: test remote: test
6 bytes received in 0.052 seconds (0.11 Kbytes/s)
ftp> ---> PORT 130,149,17,167,185,55
200 PORT command successful.
---> RETR nofile
550 nofile: No such file or directory.
ftp> ---> QUIT
221-You have transferred 6 bytes in 1 files.
221-Total traffic for this session was 13058 bytes in 2 transfers.
221-Thank you for using the FTP service on ftp.cs.tu-berlin.de.
221 Goodbye.
```

Eingabe "get test"

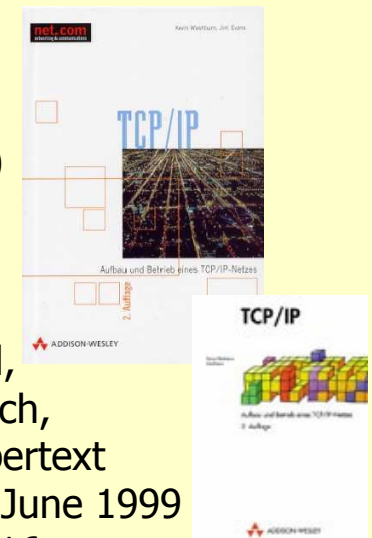
Eingabe "get nofile"

Eingabe "quit"

[91] © Robert Tolksdorf, Berlin

## Literatur

- www.ietf.org
- Washburn, Kevin; Evans, Jim  
TCP/IP, Aufbau und Betrieb  
eines TCP/IP-Netzes  
Preis: 49.95 Euro (Listenpreis)  
2000, Nachdr. 2000. X, 614 S.  
Addison-Wesley, München  
3-8273-1145-4
- R. Fielding, J. Gettys, J. Mogul,  
H. Frystyk, L. Masinter, P. Leach,  
T. Berners-Lee. RFC 2616 Hypertext  
Transfer Protocol - HTTP/1.1. June 1999  
ftp://ftp.isi.edu/in-notes/rfc2616.txt



[92] © Robert Tolksdorf, Berlin