



## Verteilte Systeme

Prof. Dr. Stefan Fischer

### Kapitel 6: Standard-Internetanwendungen

---

---

---

---

---

---

---

---

## Überblick

- Standardanwendungen im Internet
  - HTTP: Hypertext Transport Protocol
  - SMTP und POP3: Simple Mail Transfer Protocol und Post Office Protocol
  - FTP: File Transfer Protocol
- Web-Anwendungen

---

---

---

---

---

---

---

---

## Standardanwendungen

- Im Internet gibt es eine Reihe von Standardprotokollen und –anwendungen zur Erledigung relativ generischer und häufig wiederkehrender Aufgaben.
- Die Implementierung dieser Anwendung setzt jeweils auf der Socket-Schnittstelle auf.
- Dieses Kapitel zeigt einige dieser Anwendungen.

---

---

---

---

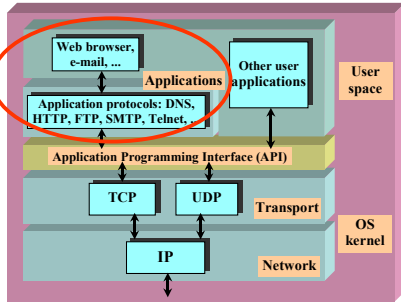
---

---

---

---

# Wo wir sind



---

---

---

---

---

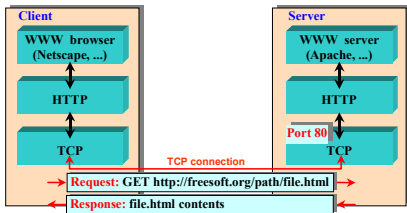
---

---

---

# HTTP

HTTP: HyperText Transfer Protocol  
World-Wide Web protocol seit 1990.  
Aktuelle Version: HTTP/1.1, RFC 2616 (1999).



---

---

---

---

---

---

---

---

# Identifikation von Ressourcen

- **URI: Universal Resource Identifier**
  - Ein String, der eine Ressource im Netz identifiziert, ohne auf die Zugriffsart einzugehen
  - Wird zur Zeit heftig diskutiert, s. auch Kapitel 7 zu Name Service
- **URL: Uniform Resource Locator**
  - URLs sind eine Untermenge der URIs
  - Eine URL identifiziert eindeutig ein Dokument im WWW, auf das z.B. über HTTP zugegriffen wird.
  - URLs haben eine feste Syntax, die das Zugriffsprotokoll und den Ort im Netz identifizieren.
  - Definiert zuerst in RFC 1738, erweitert in RFC1808, RFC2368, [RFC2396](#)
  - Kompromiss zwischen Adresse und Name
  - Problem?

---

---

---

---

---

---

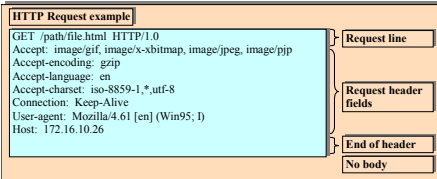
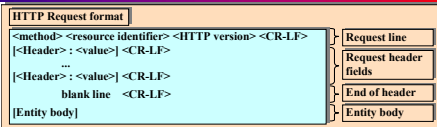
---

---

## Beispiele für URLs

Name	Verwendung	Beispiele
http	Hypertext	<a href="http://www.tu-bs.de">http://www.tu-bs.de</a> <a href="http://localhost:8080/servlet/test">http://localhost:8080/servlet/test</a>
ftp	FTP	<a href="ftp://ftp.cs.vu.nl/pub/minix/README">ftp://ftp.cs.vu.nl/pub/minix/README</a> <a href="ftp://fischer.dummy@localhost/test/file">ftp://fischer.dummy@localhost/test/file</a> (s. RFC 2396, Abschnitt 3.2.2)
File	Lokale Datei	/etc/.passwd
News	Newsgruppe	<a href="News:comp.os.minix">News:comp.os.minix</a>
mailto	Senden von Email	<a href="mailto:stefan.fischer@tu-bs.de">mailto:stefan.fischer@tu-bs.de</a>
telnet	Remote Login	<a href="telnet://www.w3.org:80">telnet://www.w3.org:80</a>

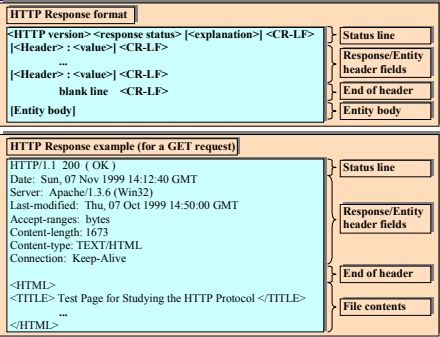
## Format des HTTP Request



## Methoden des HTTP Request

Method	Description
<b>GET</b>	Retrieve the information identified by the URI (Unique Resource identifier).
<b>HEAD</b>	Retrieve HTTP response header for the specified URI, without the contents (same header as for GET method). Used to check hypertext links for validity, accessibility, and recent modification.
<b>POST</b>	Pass enclosed entity to be processed by the resource identified by the URI. Used for annotating resources, posting a message to a newsgroup or mailing list, providing a data block (e.g., interactive form) to a data-handling process.
<b>PUT</b>	Pass enclosed entity to be stored under the specified URI. Used to create/modify documents.
<b>DELETE</b>	Delete the resource identified by the URI.
<b>TRACE</b>	Used for diagnostic tests.

# Format der HTTP Response




---

---

---

---

---

---

---

---

---

---

# Status Codes der HTTP Response

Code	Description
1xx	Informational. 100 Continue; 101 Switching Protocols; ...
2xx	Successful. 200 OK; 201 Created; 202 Accepted; ...
3xx	Redirection. 300 Multiple Choices; 301 Moved Permanently; ...
4xx	Client Error. 400 Bad Request; 401 Unauthorized; ...
5xx	Server Error. 500 Internal Server Error; 501 Not Implemented; ...

---

---

---

---

---

---

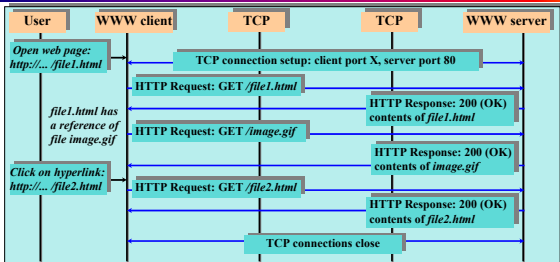
---

---

---

---

# Beispiel einer HTTP (1.1) Session



TCP-Verbindungen überdauern HTTP requests und können für mehrere Anfragen verwendet werden  
Mehrere TCP-Verbindungen können parallel verwendet werden.

---

---

---

---

---

---

---

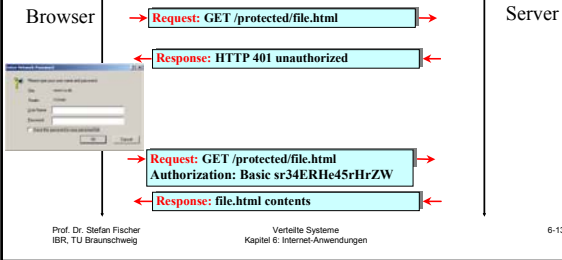
---

---

---

# HTTP Basic Authentication

- Es ist möglich, Webseiten mit einem Usernamen und Passwort zu schützen
  - user:pass werden Base64 kodiert verschickt



---

---

---

---

---

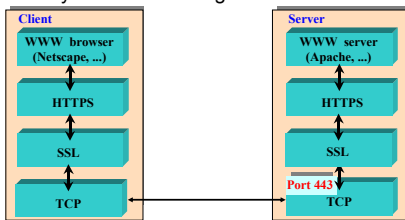
---

---

---

# HTTPS

- Sicheres HTTP funktioniert wie HTTP
  - Verschlüsselung mit Triple DES
  - Vorheriger Austausch des DES Schlüssels über Public Key Verschlüsselung



---

---

---

---

---

---

---

---

# Sicherheit

- Die Kombination von Authentication und HTTPS schützt vor
  - Mithören der Nachrichten
  - Unbefugtem Zugriff auf die Ressourcen
- Einfache und trotzdem mächtige Kombination

---

---

---

---

---

---

---

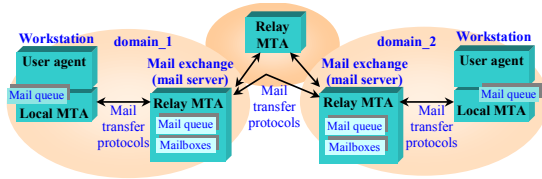
---

# Elektronische Post

## Architektur des Internet-Mail-Systems

Mail User Agent (MUA) zur Interaktion mit dem Benutzer

Mail Transfer Agent (MTA) zur Weiterleitung von Nachrichten.




---

---

---

---

---

---

---

---

---

---

# Standards für Internet Email

## Format von Emails

**Header:** RFC 822 (1982).

**Inhalt:** MIME (Multipurpose Internet Mail Extensions). RFC 1341, 1521, 1522, ...

## Protokolle zur Mail-Übertragung

**Senden und Weiterleiten.** Zwischen MTAs.

SMTP: Simple Mail Transfer Protocol. RFC 821 (1982).

**Mail abrufen.** Zwischen lokalem MTA und Mail Relay (= Mail Server).

POP3: Post Office Protocol. RFC 1225 (1991).

IMAP4: Internet Message Access Protocol. RFC 2060 (1996).

---

---

---

---

---

---

---

---

---

---

# Format von Emails

## E-mail example

```

Return-Path: <jim@elc.fr>
Received: from first.elc.fr (root@first.elc.fr [151.85.254.43])
  by alix.int.fr (8.8.8/tpda-5.3) with ESMTP id RAA23346
  for <tom@int.fr>; Fri, 16 Oct 1998 17:04:01 +0200 (MET DST)
Received: from elc.fr (jim@iris.elc.fr [151.85.43.13])
  by first.elc.fr (8.8.5/8.8.0) with ESMTP id QAA15735;
  Fri, 16 Oct 1998 16:48:48 +0300
Message-Id: <199810161356.QAA08502@elc.fr>
From: Jim Smith <jim@elc.fr>
To: tom@int.fr
Cc: sam@elc.fr
Subject: Important news
Date: Fri, 16 Oct 1998 16:56:40 +0300 (EET DST)
X-Mailer: ELM [version 2.4.PL23]
MIME-Version: 1.0
Content-Type: text/plain; charset=US-ASCII
Content-Transfer-Encoding: 7bit
Content-Length: 123

Next week I'll be on vacation.
Jim.
    
```

Used by local MTA to derive the envelope

MIME headers

Blank line

User's message

Headers added successively by MTA relays

Headers added by the sender user agent (elm)

End of headers

Body

---

---

---

---

---

---

---

---

---

---

# Protokolle zur Übertragung

- SMTP: leite Nachrichten von MTA zu MTA
  - Zuverlässige und effiziente Übertragung
  - Benachrichtigung im Fehlerfall
- POP3: Abrufen von Mail vom Server
  - Authentifizierung
  - Lesen der Mail und Update der Server-Mailbox
- IMAP4: verbesserter Mail-Abruf vom Server
  - Verbesserte Server-Funktionen: Authentifizierung, Verwaltung mehrerer Mailboxen
  - Verbesserte Möglichkeiten zur Mailbox-Verwaltung: Auswahl von Nachrichten, Anwendung von Befehlen auf bestimmte Nachrichten, etc.

---

---

---

---

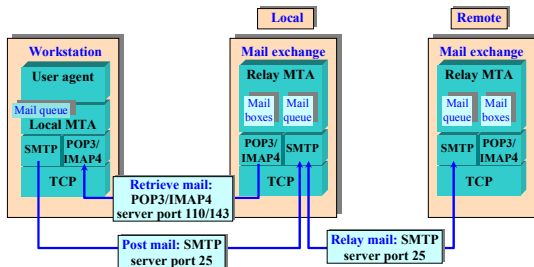
---

---

---

---

# Übertragungsprotokolle (Forts.)




---

---

---

---

---

---

---

---

# SMTP-Befehle

Die sieben wichtigsten Befehle (von 14)

Command	Description
<b>HELO</b> host-name	Client SMTP identification.
<b>MAIL FROM:</b> <reverse-path>	Sender identification.
<b>RCPT TO:</b> <forward-path>	Recipient identification. Repeated if multiple recipients.
<b>DATA</b>	Mail contents follow.
<b>VERFY</b> user-name	Verify recipient user name.
<b>EXPN</b> mail-list	Expand mail list.
<b>QUIT</b>	Mail transfer terminated.

---

---

---

---

---

---

---

---

# Antworten in SMTP

SMTP replies (selection)	
220	<domain> Service ready
221	<domain> Service closing transmission channel
250	Requested mail action okay, completed
251	User not local; will forward to <forward-path>
354	Start mail input; end with <CRLF>.<CRLF>
421	<domain> Service not available, closing transmission channel
450	Requested mail action not taken: mailbox unavailable
451	Requested action aborted: local error in processing
452	Requested action not taken: insufficient system storage
500	Syntax error, command unrecognized
501	Syntax error in parameters or arguments
502	Command not implemented
503	Bad sequence of commands
504	Command parameter not implemented
551	User not local; please try <forward-path>
554	Transaction failed

---

---

---

---

---

---

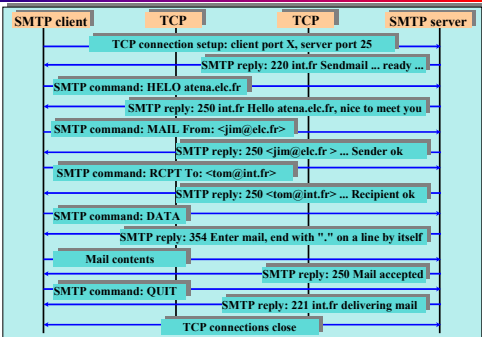
---

---

---

---

# Beispiel für eine SMTP-Sitzung




---

---

---

---

---

---

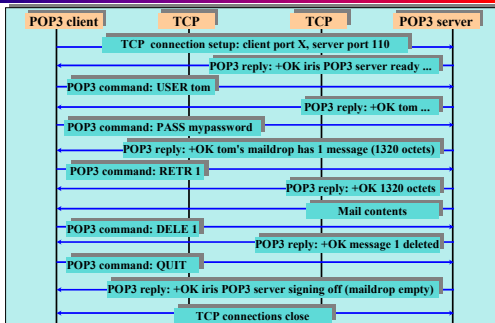
---

---

---

---

# Beispiel für eine POP3-Sitzung




---

---

---

---

---

---

---

---

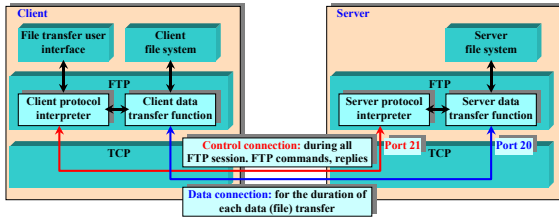
---

---



# FTP: File Transfer Protocol

Zweck: Zugriff auf netzweite Dateiarhive  
 Spezifiziert in RFC 959 (1985).




---

---

---

---

---

---

---

---

---

---

## Die wichtigsten FTP-Befehle

Command	Description
<b>USER username</b>	User name on server
<b>PASS password</b>	User password on server
<b>PORT n1,n2,n3,n4,n5,n6</b>	Client IP address ( $n1.n2.n3.n4$ ) and port ( $n5 \times 256 + n6$ ) for data connection
<b>LIST filelist</b>	List files and directories (user interface command DIR)
<b>TYPE type</b>	File type: A for ASCII, I for binary (image)
<b>RETR filename</b>	Retrieve specified file from server (user interface command GET)
<b>STOR filename</b>	Store specified file on server (user interface command PUT)
<b>QUIT</b>	Logout from server (user interface command QUIT)

---

---

---

---

---

---

---

---

---

---

## FTP-Antworten

Reply	Description
<b>1yz</b>	Positive preliminary reply. Action started, expect another reply before sending new command.
<b>2yz</b>	Positive completion reply. A new command can be sent.
<b>3yz</b>	Positive intermediate reply. Command accepted, another command must be sent.
<b>4yz</b>	Transient negative completion reply. Action not executed but error condition is temporary
<b>5yz</b>	Permanent negative completion reply. Command not accepted, should not be retried
<b>x0z</b>	Syntax errors
<b>x1z</b>	Information
<b>x2z</b>	Connections (control or data)
<b>x3z</b>	Authentication and accounting.
<b>x4z</b>	Unspecified
<b>x5z</b>	File system status

---

---

---

---

---

---

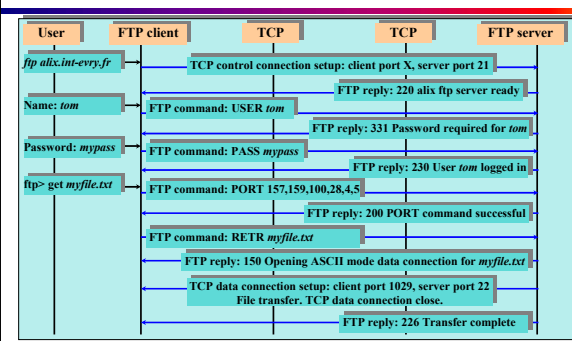
---

---

---

---

## Beispiel einer FTP-Session



---

---

---

---

---

---

---

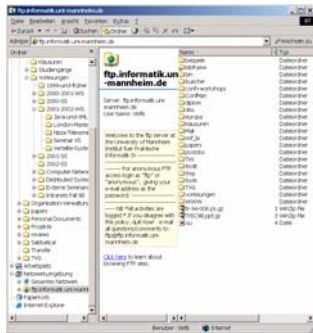
---

---

---

## FTP-Clients

- Einfachste Variante: das Programm `ftp` in UNIX
  - Kommandozeilenschnittstelle, eher kryptische Bedienung
- Heute: fensterbasierte Clients, z.B. WS-FTP, teilweise mit kompletter Einbindung in lokale Dateisysteme
  - Beispiel: Windows-Explorer in Windows 2000



---

---

---

---

---

---

---

---

---

---

## Web-Anwendungen

- Idee: verwende die Infrastruktur eines Anwendungsdienstes wie HTTP, um ohne großen Aufwand mächtige dynamische Anwendungen implementieren zu können
- Wir wollen uns hier einen kurzen Überblick verschaffen, ansonsten: Vorlesung „Web Anwendungen mit Java und XML“

---

---

---

---

---

---

---

---

---

---

# N-Tier Architekturen

- Unternehmensweite Anwendungen werden heute als *N-stufige Anwendungen* (N-Tier Applications) entwickelt (N=2,3,4, ...)
- Jedes „Tier“ (Layer, Stufe, Ebene) hat seine eigene Aufgabe
- Vorteile
  - geringere Komplexität der einzelnen Teile
  - Verteilung der Implementierungsaufgaben
  - Flexibilität bei der Verteilung der einzelnen Aufgaben (thin client)
  - erleichterte Wartbarkeit (keine Client-Software, Austausch von Versionen)
  - Skalierbarkeit, Sicherheit

---

---

---

---

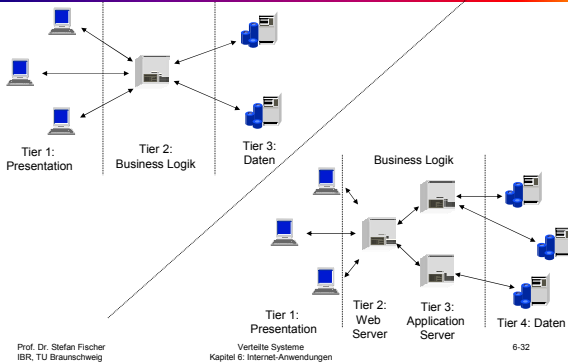
---

---

---

---

## 3- und 4-Tier-Architekturen




---

---

---

---

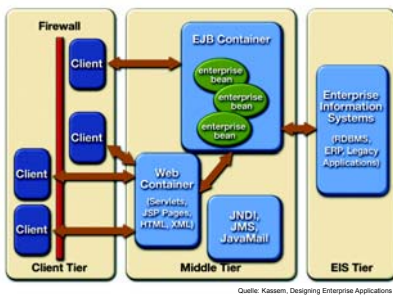
---

---

---

---

## Suns J2EE-Sicht der N-Tier Architecture




---

---

---

---

---

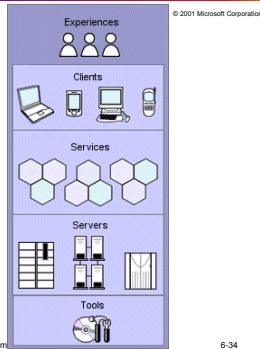
---

---

---

# Microsofts .Net-Sicht

- Microsofts Plattform für XML Web Services
- XML Web Services gestatten es Anwendungen, über das Internet zu kooperieren und Daten auszutauschen.
- Unabhängig von Programmiersprache und Betriebssystem



---

---

---

---

---

---

---

---

# Client Tier

- Benutzerschnittstelle
- heute häufig als „thin client“ ohne jegliche weitere Anwendungslogik, implementiert über den Web-Browser
- wichtigste Funktionen
  - Eingaben des Benutzers „abholen“
  - Ausgaben/Ergebnisse der Server-Seite der Anwendung darstellen
- Wichtigste Technologien
  - HTML und WML, vor allem Formulare (dominierend)
  - Java Applets

---

---

---

---

---

---

---

---

# Middle Tier

- In dieser Ebene findet sich der Großteil der Anwendungslogik wieder, bei E-Commerce z.B.
  - Warenkorbfunktionen
  - Preisberechnungs- und Bezahlungsfunktionen
- Die Ebene kann aus Skalierbarkeits- und Sicherheitsgründen physisch weiter aufgeteilt sein in
  - den Web-Server, der das direkte (und oft einzige) Interface zum Benutzer darstellt
  - den Application Server, der weitere standardisierte und nutzbare Dienste bereit stellt (Transaktionen, Security)
  - Oftmals werden Web Server und Application Server in einem Paket realisiert
- Technologien: Servlets, JSP, ASP, CORBA, EJB

---

---

---

---

---

---

---

---

## EIS Tier

- Diese Ebene hat die Aufgabe, die eigentlichen Daten der Anwendung zu verwalten, bei E-Commerce z.B.
  - Kundendaten
  - Produktdaten
  - Bestellungen
- Typische Implementierungen
  - Datenbanken wie DB2, Oracle, SQL Server
  - Enterprise Resource Planning Systeme wie R/3

---

---

---

---

---

---

---

---

## Literatur

- A. Tanenbaum: Computer Networks, 3<sup>rd</sup> ed., Prentice Hall, 1996 (bzw. die brandneue 4. Auflage).
- A. Eberhart u. S. Fischer: *Java-Bausteine für E-Commerce-Anwendungen*, 2. Auflage, Carl Hanser Verlag, 2001.
- N. Kasseem et al.: *Designing Enterprise Applications with the Java 2 Platform, Enterprise Edition*, Addison-Wesley (The Java Series), 2000.
- Einstiegspunkte im Web:
  - J2EE: <http://java.sun.com/j2ee>
  - .NET: <http://www.microsoft.com/net>

---

---

---

---

---

---

---

---