

# TI III: Operating & Communication Systems Applications

Domain Name System

Email

World Wide Web

# Content (3)

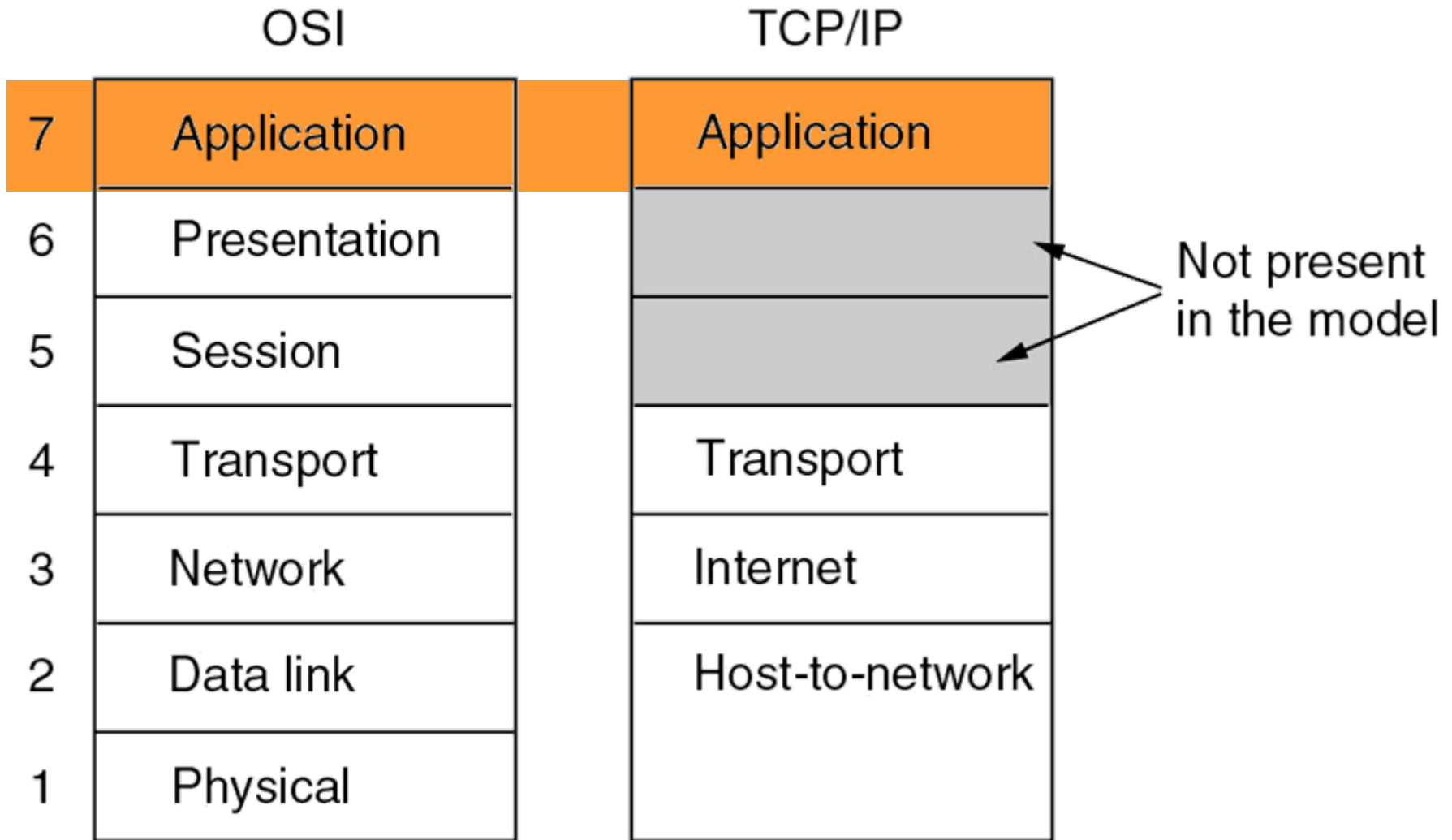
## 14. Security

- Basic concepts & terms
- Cryptology
- Examples
  - Firewalls
  - Virtual Private Networks (VPNs)
  - IP Security
  - Email security with PGP

## 15. Application

- Domain Name System
- Email
- World Wide Web

## 16. Example



## Goals of this chapter

- With the transport layer in place, all essential functionality to build a network is in place
  - Anything else is up to the application programmer
- Nonetheless, there are some services that are almost essential for a practical network but that actually belong to the application layer
  - DNS as prime example
- Moreover, some applications are important enough to warrant a few more words
  - Email, WWW

- Addressing in the Internet uses 4 bytes (IPv4), commonly represented in dotted decimal notation
- Nice for machines, impractical for human beings
  - Do you recognize (or remember) 160.145.117.8 ?
- More convenient: Mnemonic names for communication peers
  - E.g., [www.inf.fu-berlin.de](http://www.inf.fu-berlin.de)
- ***Domain Name System (DNS)*** solves this need
- Plus: IP addresses may change, not the name!

# Example DNS query

- Who is responsible for handling the following email receiver: [j.schiller@ieee.org](mailto:j.schiller@ieee.org)?
- I.e.: How can I reach the mailserver of ieee.org?

```

nslookup -q=mx ieee.org
Server:  einstein.abc.com
Address: 123.45.67.89

MX-Record →  ieee.org      preference = 0, mail exchanger = gemini.ieee.org
NS-Records {  ieee.org      nameserver = auth01.ieee.org
               ieee.org      nameserver = dns.ieee.org
               ieee.org      nameserver = ns.uu.net
               ieee.org      nameserver = krypton.ieee.org
               ieee.org      nameserver = depththought.ieee.org
A-Records {   gemini.ieee.org internet address = 199.172.136.14
               auth01.ieee.org internet address = 199.172.136.2
               dns.ieee.org   internet address = 199.172.136.6
               ns.uu.net      internet address = 137.39.1.3
               krypton.ieee.org internet address = 199.172.136.2
               depththought.ieee.org internet address = 199.172.136.6
    
```

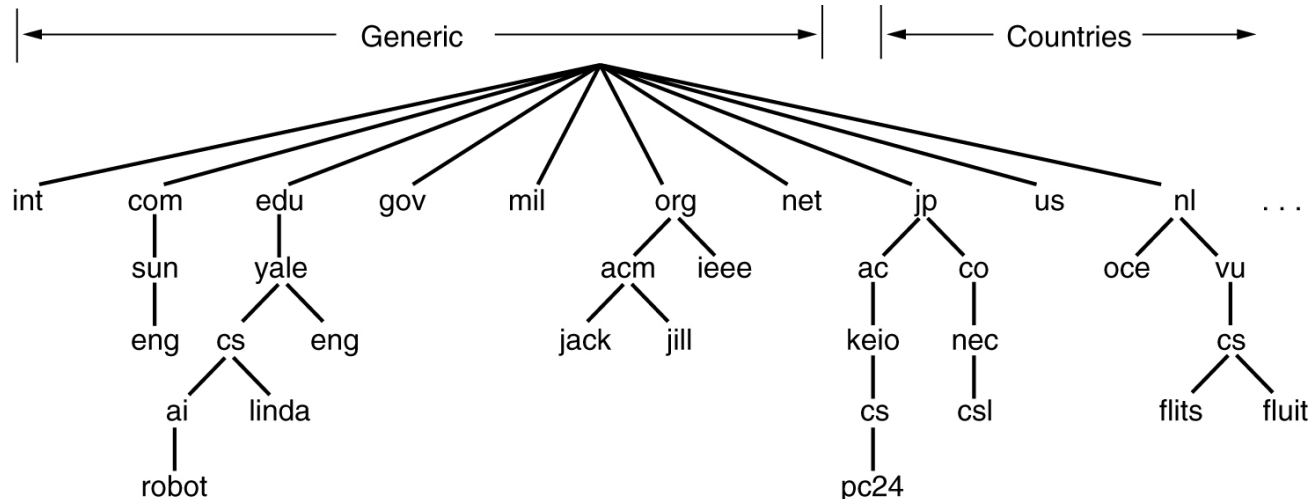
# „DiG“ or „host“ as alternative

```
; <<>> DiG 9.2.4rc5 <<>> mx ieee.org
;; global options:  printcmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 22641
;; flags: qr rd ra; QUERY: 1, ANSWER: 2, AUTHORITY: 5, ADDITIONAL: 4
;; QUESTION SECTION:
;ieee.org.                IN          MX
;; ANSWER SECTION:
;ieee.org.                3600       IN          MX          0 orion.ieee.org.
;ieee.org.                3600       IN          MX          0 hormel.ieee.org.
;; AUTHORITY SECTION:
;ieee.org.                3600       IN          NS          auth01.ieee.org.
;ieee.org.                3600       IN          NS          rushhour.ieee.org.
;ieee.org.                3600       IN          NS          depththought.ieee.org.
;ieee.org.                3600       IN          NS          ns.uu.net.
;ieee.org.                3600       IN          NS          dns.ieee.org.
;; ADDITIONAL SECTION:
;orion.ieee.org.         3600       IN          A           140.98.193.224
;hormel.ieee.org.       3600       IN          A           140.98.193.224
;rushhour.ieee.org.    3600       IN          A           140.98.193.128
;depththought.ieee.org. 3600       IN          A           140.98.194.128

;; Query time: 130 msec
;; SERVER: 160.45.110.15#53(160.45.110.15)
;; WHEN: Fri Aug 27 14:46:00 2004
;; MSG SIZE  rcvd: 246
```

# DNS – Architecture

- DNS maps names to addresses
  - Actually: maps to resource records
- Names are structured hierarchically into a name space
  - Max. 63 characters per component, max. 255 characters total
  - Domains, each domain owner controls name space below it

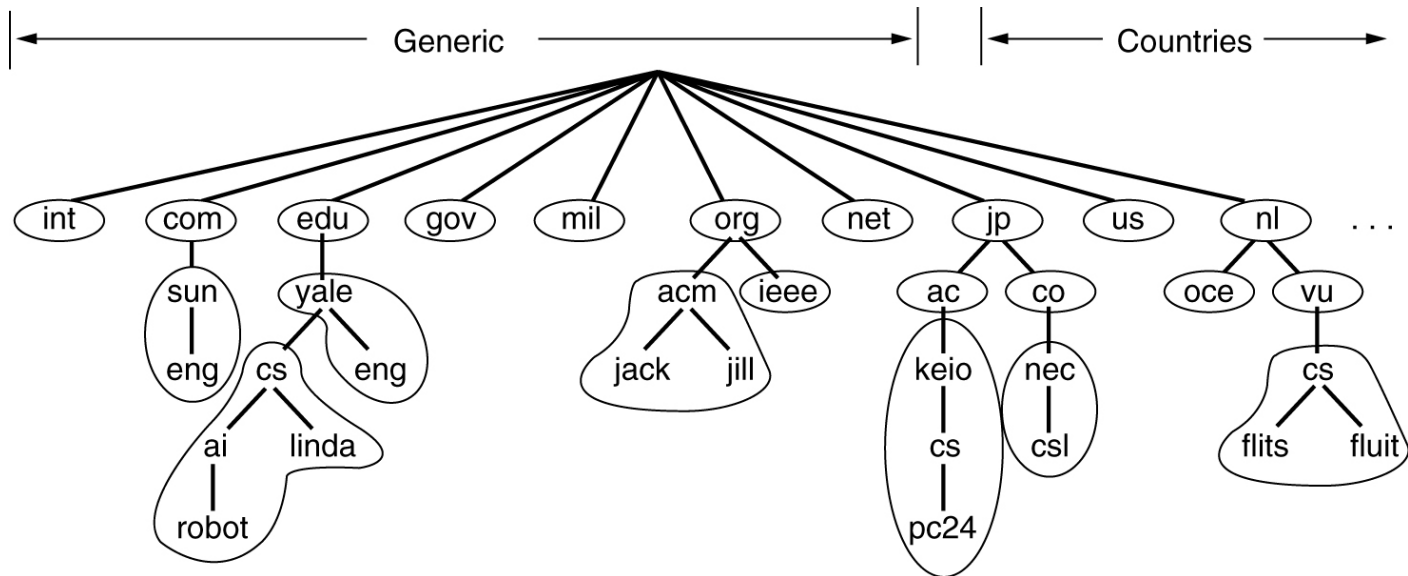


- Mapping done by name servers (well known, hierarchical)



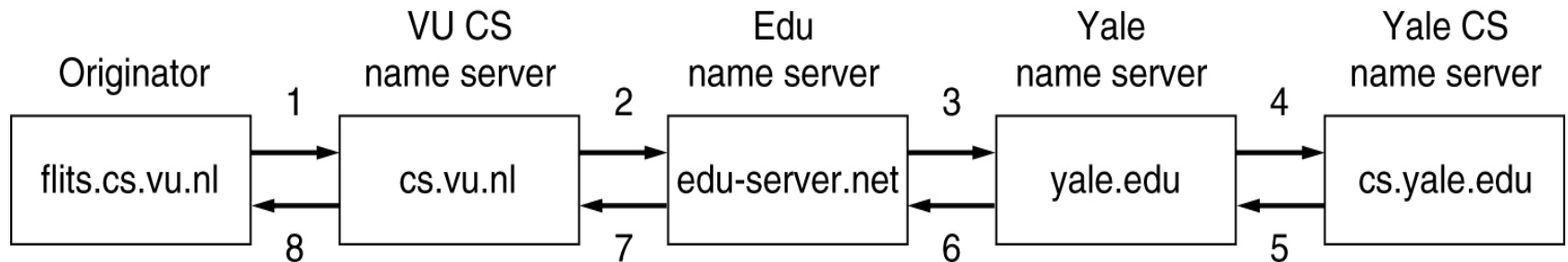
# DNS name servers

- Name space divided into zones
- Each zone has a primary name server with authoritative information
  - Also secondary name server for dependability
- Each name server knows about
  - Its own zone
  - Name servers of all its children zones
  - Their siblings or about some server that knows about the siblings

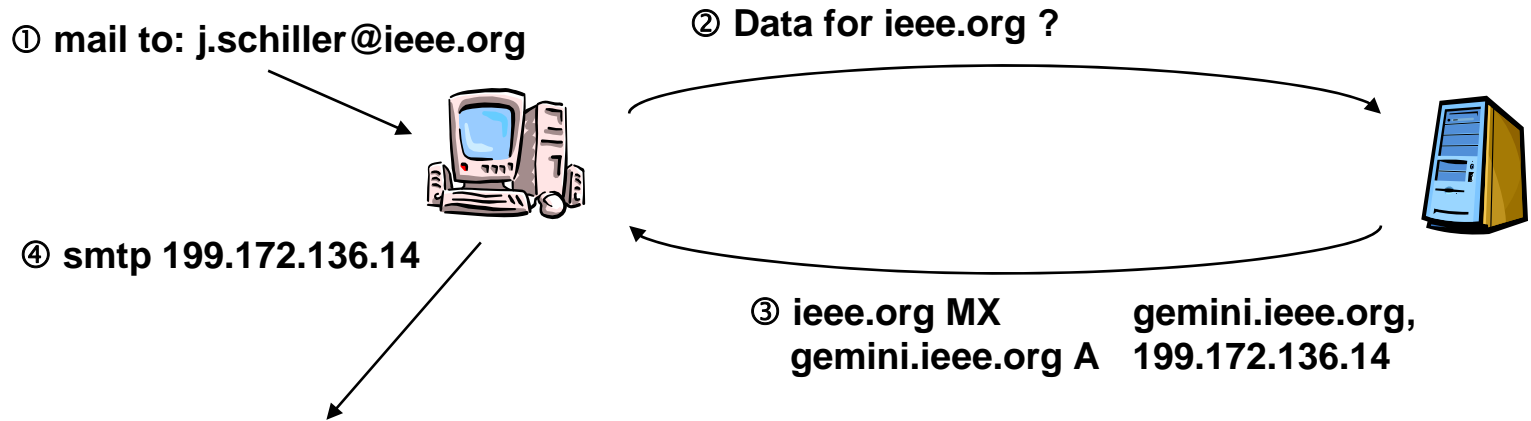
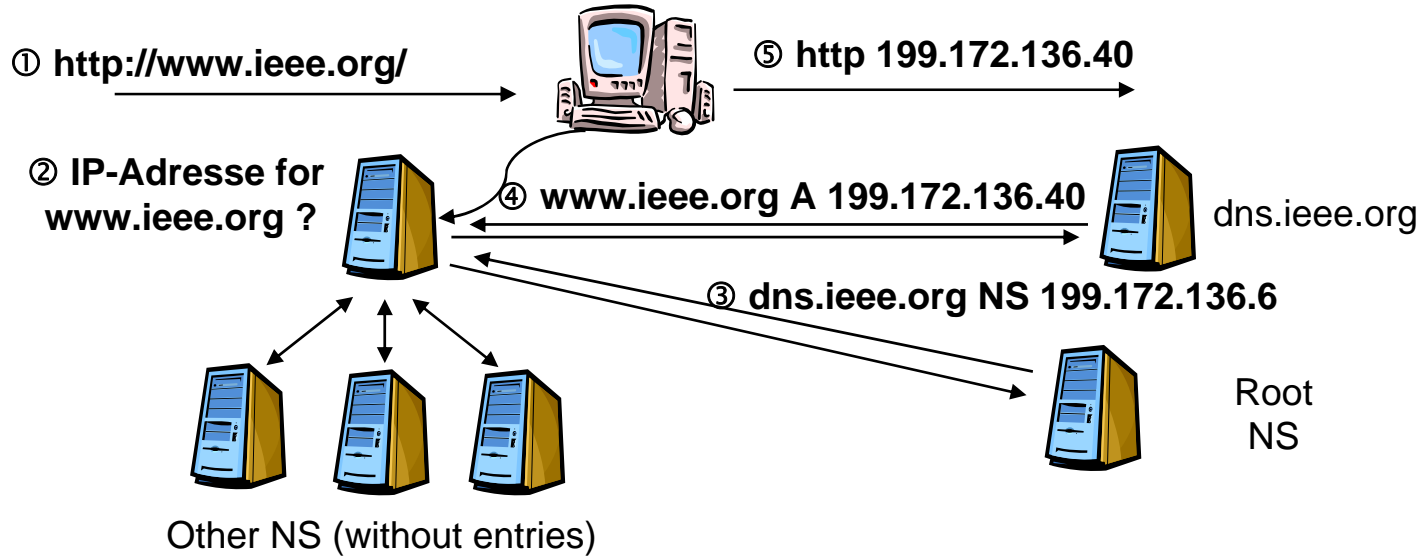


# DNS query resolution

- Queries by an end system are sent to their pre-configured name server (obtained from configuration, DHCP,...)
- If possible, that name server answers query
- If not, it will forward query to the “most suitable” name server in the zone hierarchy it is aware of
  - Continues recursively (or iteratively, esp. root NS)
- Answer sent back through intermediate servers
- Servers may cache replies



# DNS: Examples (A-/MX-Records)



# 13 Original root servers

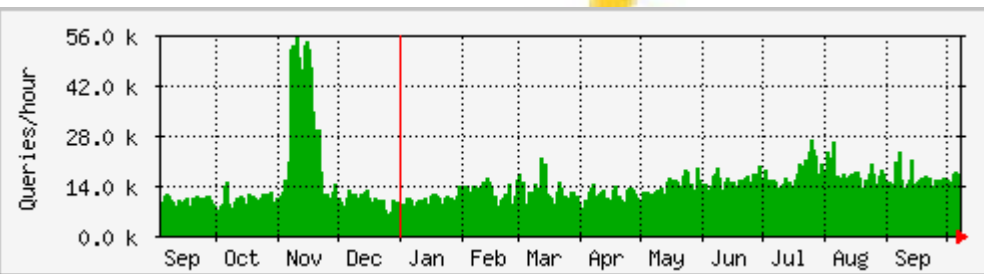


# 13 Root server plus anycasting

<http://zonecheck.denic.de>



Servers without anycasting vulnerable!

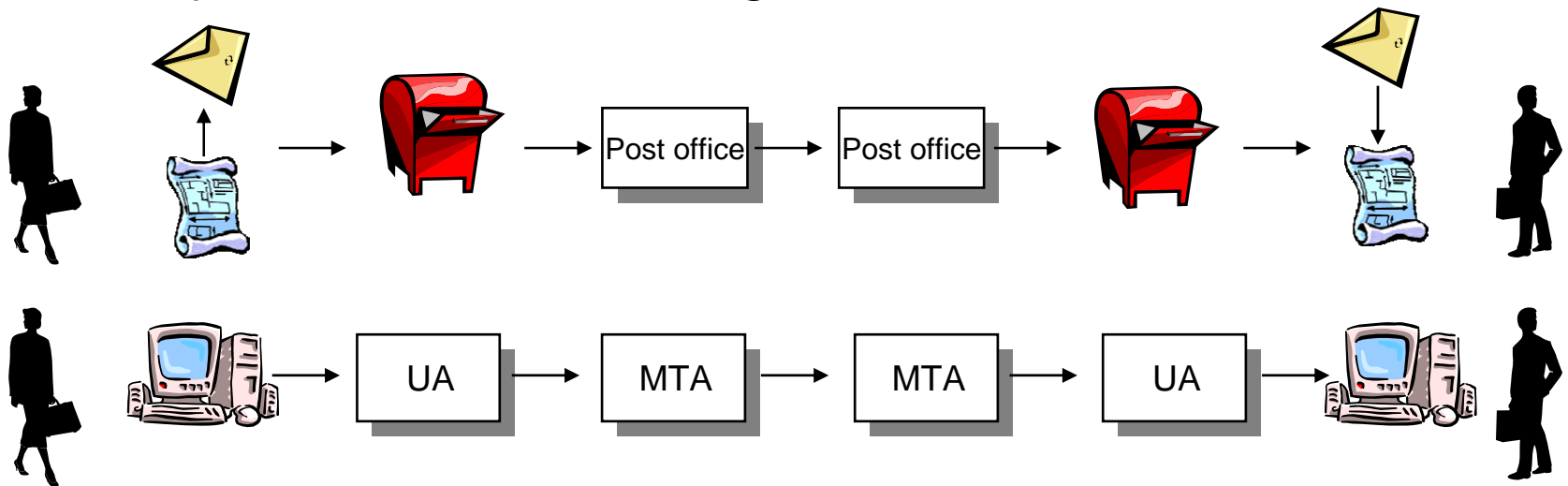


<http://european.de.orsn.net>

- Description here: Email as in RFC 821/822
- Main parts: User agents (UA) & message transfer agents (MTA) plus a protocol (SMTP, simple mail transfer protocol)
- Main services
  - Composition, transfer, reporting, displaying, disposition
- Additional services
  - Forwarding, auto-reply, vacation functions, mailing lists, BCC, ...
- Main structure of an email:
  - Envelope – information required for transport, used by MTA
  - Content
    - Header – control information for the UA
    - Body – actual data of interest

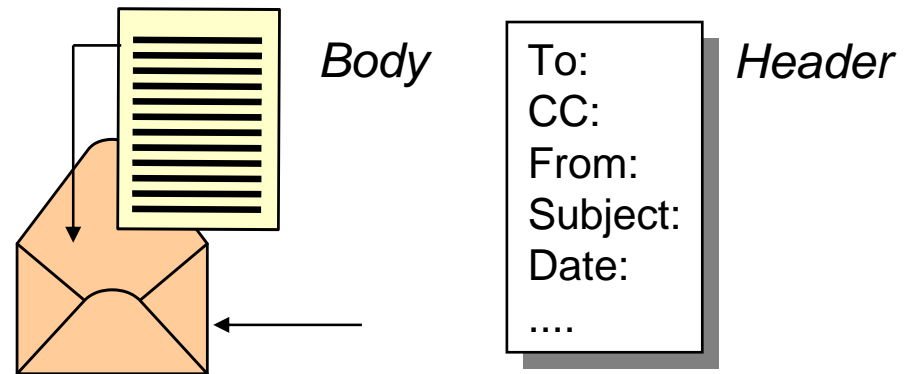
# Email: General Model

- User Agent (UA)
  - Local, graphical/text-oriented program
  - Reading, writing, sending, receiving of Email on a local machine
  - E.g. integrated in browser, Outlook, pine ...
- Message Transfer Agent (MTA)
  - Background process
  - Responsible for forwarding of emails to receiver



# Email: Format

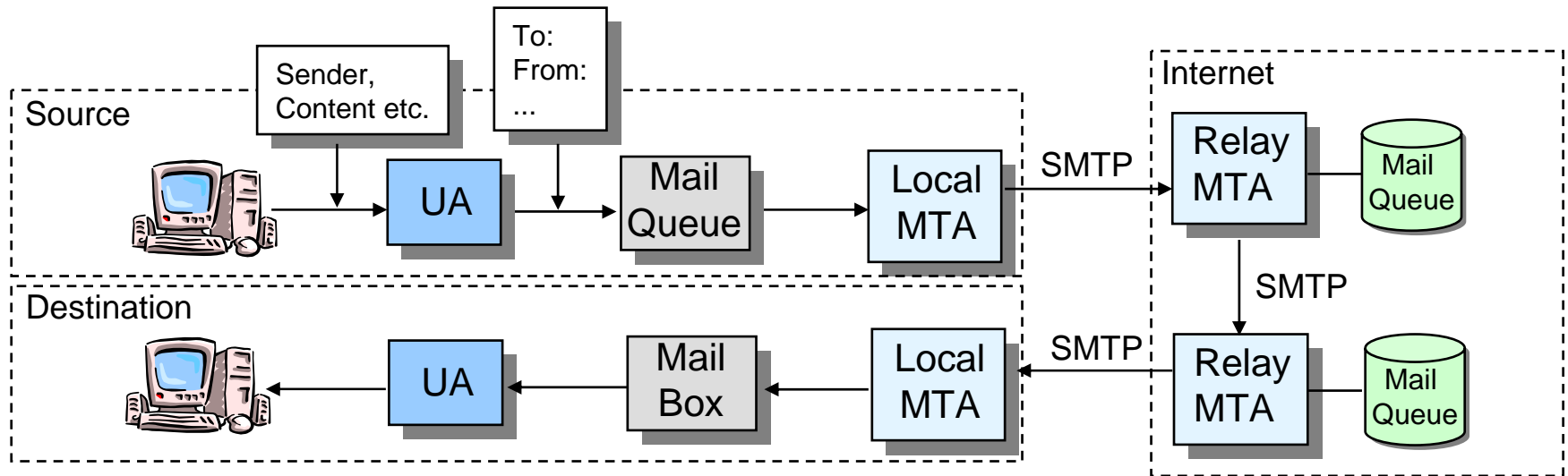
- Envelope
  - Contains all transport-relevant information ("To:")
  - Addressing based on DNS, e.g., schiller@computer.org
  - Interpreted by MTAs
- Content
  - Header
    - Contains additional fields, e.g. "Subject:", "CC:" ...
    - Interpreted by UAs
    - SMTP additionally integrated envelope fields in the header
  - Body
    - Contains the real message (originally ASCII only)





# SMTP (Simple Mail Transfer Protocol)

- SMTP pushes messages over TCP connections (port 25)
  - Text-oriented protocol, originally 7 bit ASCII
  - Few, simple commands, e.g., HELO, MAIL, RCPT, DATA ...
- UA gets all necessary information from user
  - Sends message via local mail queue to local MTA
- MTAs transfer message to receiver
  - Transfer via TCP, relays possible (e.g., campus relay plus MTAs for each institute)
  - Example MTAs: sendmail, qmail, Exchange ...



# SMTP: Example (few mail servers allow this...)

Thus an “old” example:

```
> telnet mailer.inf.fu-berlin.de 25

E: 220 mailer.inf.fu-berlin.de ESMTP
    Sendmail 8.9.3/8.9.3; Wed, 22 Sep 1999 10:41:34 +0200 (MET DST)
S: HELO laptop.inf.fu-berlin.de
E: 250 mailer Hello laptop [123.45.67.89], pleased to meet you

S: MAIL FROM: whoever
E: 250 whoever... Sender ok

S: RCPT TO: webadmin@inf.fu-berlin.de
E: 250 whoever .. Recipient ok

S: DATA
E: 354 Enter mail, end with "." on a line by itself
S: Dear administrator, good to see that this does not work any longer...
S: .
E: 250 KAA12526 Message accepted for delivery

S: QUIT
E: 221 blackfoot closing connection
```

- Original SMTP allows ASCII only as content
  - How to transfer images, sound, arbitrary attachments?
- MIME extends the content by formatting/type information
  - Content-Type: defines the type of the body (part)
    - Text, multipart, message, application (binary), image, audio, video, X-private...
  - Content-Transfer-Encoding: defines transfer syntax for body (part) encoding
    - Base 64, quoted printable, 7 bit, 8 bit, binary...
- Still compatible to classical mail
  - E.g., base 64 allows transfer of binary data though 7 bit ASCII only networks
  - E.g., quoted printable supports national special characters

# Example header with MIME (1)

```

Microsoft Mail Internet Headers Version 2.0
Received: from math.fu-berlin.de ([160.45.40.10]) by spree.pcpool.mi.fu-berlin.de with
    Microsoft SMTPSVC(6.0.3790.1830);
    Thu, 1 Feb 2007 14:12:38 +0100
Received: (qmail 18895 invoked by uid 12368); 1 Feb 2007 14:12:38 +0100
Delivered-To: schiller@mi.fu-berlin.de
Received: (qmail 18887 invoked from network); 1 Feb 2007 14:12:37 +0100
Received: from lusin.mi.fu-berlin.de (HELO mi.fu-berlin.de) (160.45.117.141)
    by leibniz.math.fu-berlin.de with SMTP; 1 Feb 2007 14:12:37 +0100
Received: (qmail 306 invoked by uid 9804); 1 Feb 2007 14:12:37 +0100
Received: from localhost (HELO mi.fu-berlin.de) (127.0.0.1)
    by localhost with SMTP; 1 Feb 2007 14:12:16 +0100
Received: (qmail 32594 invoked by uid 9804); 1 Feb 2007 14:12:16 +0100
Received: from leibniz.math.fu-berlin.de (HELO math.fu-berlin.de) (160.45.40.10)
    by lusin.mi.fu-berlin.de with SMTP; 1 Feb 2007 14:12:16 +0100
Received: (qmail 18610 invoked from network); 1 Feb 2007 14:12:16 +0100
Received: from outpost1.zedat.fu-berlin.de (130.133.4.66)
    by leibniz.math.fu-berlin.de with SMTP; 1 Feb 2007 13:12:16 -0000
Received: from relay2.zedat.fu-berlin.de ([130.133.4.80])
    by outpost1.zedat.fu-berlin.de (Exim 4.66)
    with esmtp
    (envelope-from <xyz@fu-berlin.de>)
    id <1HCbjb-0006sB-RW>; Thu, 01 Feb 2007 14:12:15 +0100
Received: from exchangel.intranet.fu-berlin.de ([130.133.175.130])
    by relay2.zedat.fu-berlin.de (Exim 4.66)
    with esmtp
    (envelope-from <xyz@fu-berlin.de>)
    id <1HCbjb-0007Nm-PL>; Thu, 01 Feb 2007 14:12:15 +0100

```

## The way through MTAs:

1. exchange1
2. relay2
3. outpost1
4. leibniz
5. lusin
6. "localhost"
7. lusin
8. leibniz
9. spree

# Example header with MIME (2)

```
X-MimeOLE: Produced By Microsoft Exchange V6.5
X-Envelope-Sender: xyz@fu-berlin.de
X-Virus-Scanned: by AMaViS 0.3.12pre7-L39+ClamAV[32603](NAI-uvscan@mi.fu-berlin.de)
X-Remote-IP: x.y.z.66
Content-class: urn:content-classes:message
MIME-Version: 1.0
Content-Type: multipart/mixed;
    boundary="----_=_NextPart_001_01C74602.97F495E6"
Subject: WG: Entwurf Papier
Date: Thu, 1 Feb 2007 14:12:15 +0100
Message-ID: <CFA40C3940C1ED438C866941A792C51E2F655E@exchange1.intranet.fu-berlin.de>
X-MS-Has-Attach:
X-MS-TNEF-Correlator:
Thread-Topic: Entwurf Papier
Thread-Index: AccTJlgncRWmtICDQrqvw5pWDHNbdAAhD13AC14hoDABBN0EAAArgEcwAACjpfAABpxL4A==
From: "xyz, abc" <abc.xyz@fu-berlin.de>
To: =?iso-8859-1?Q?Um=FCglaut=2C_Muster?= <Muster.Umülaut@abc.xyz.de>,
    "Jochen Schiller" <schiller@mi.fu-berlin.de>
X-Originating-IP: x.y.z.130
X-ZEDAT-Hint: A
Return-Path: abc.xyz@fu-berlin.de
X-OriginalArrivalTime: 01 Feb 2007 13:12:38.0174 (UTC) FILETIME=[A55FB3E0:01C74602]
```

Header information with many private extensions

# Example header with MIME (3)

```
-----_=_NextPart_001_01C74602.97F495E6
Content-Type: multipart/alternative;
    boundary="-----_=_NextPart_002_01C74602.97F495E6"
```

```
-----_=_NextPart_002_01C74602.97F495E6
Content-Type: text/plain;
    charset="iso-8859-1"
Content-Transfer-Encoding: quoted-printable
```

This is the message as plain text.

```
-----_=_NextPart_002_01C74602.97F495E6
Content-Type: text/html;
    charset="iso-8859-1"
Content-Transfer-Encoding: quoted-printable
```

```
...
<h1>This is the message as <b>HTML</b>.</h1>
...
```

```
-----_=_NextPart_001_01C74602.97F495E6
Content-Type: application/msword;
    name="Relevant report on irrelevant topics.doc"
Content-Transfer-Encoding: base64
Content-Description: Relevant report on irrelevant topics.doc
Content-Disposition: attachment;
    filename="Relevant report on irrelevant topics.doc"
```

```
0M8R4KGxGuEAAAAAAAAAAAAAAAAAAAAAAAAAPgADAP7/CQAGAAAAAAAAAAAAAAAAACAAAAtgAAAAAAAAAAA
pcEAWQAHBAAACBK/AAAAAAAAAEAAAAAAAAABAAAP2sAAA4AYmpiavNX81cAAAAAAAAAAAAAAAAAAAAAAAA
```

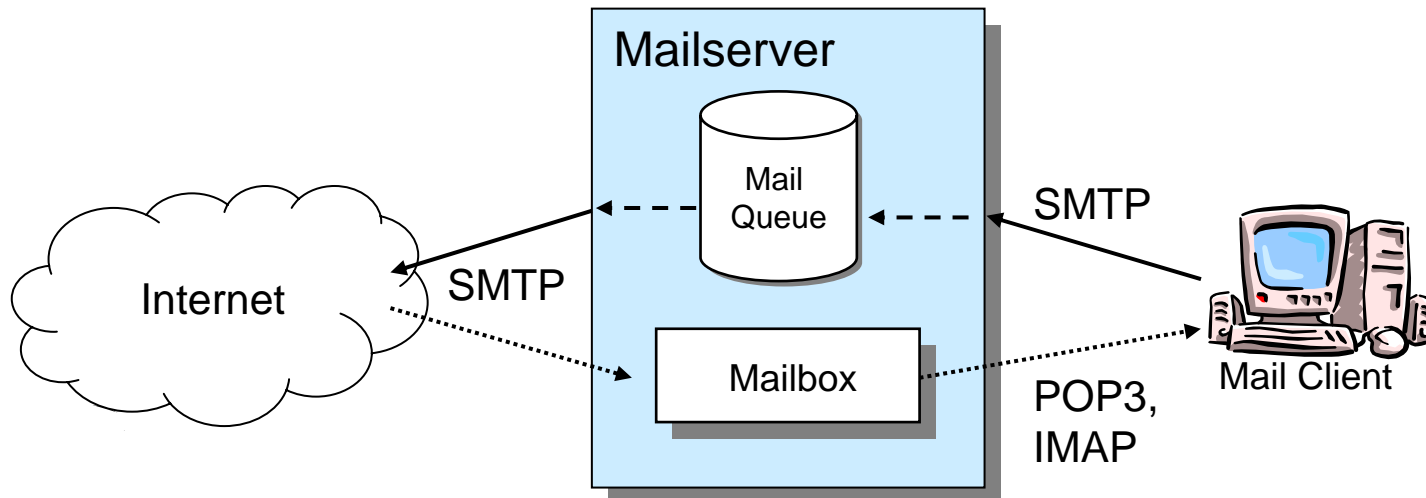
Here: only parts of a header!

Message – alternatively as plain text or HTML

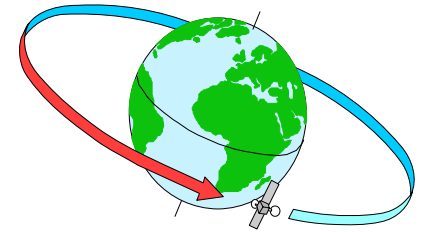
Word document

# Management of emails

- Typically, a central mail server handles all email (always online)
- Clients are not always online, need to pull email (SMTP pushes only)
- POP3 (Post Office Protocol 3)
  - Very simple pull protocol, client pulls email from server
  - Messages can stay on server or server deletes messages
  - With/without authentication, secure transmission
- IMAP (Interactive Mail Access Protocol)
  - Management of emails on a central server, support of several clients
  - Many commands for filtering, forwarding, online/offline operation...



- Started as project of the British computer scientist Tim Berners-Lee at the European research center CERN (Geneva)
  - Goal: Simple, world-wide exchange of documents among researchers (first ideas 1989)
- First prototype 1990
  - graphical (based on NEXTStep) and text-oriented
- Break-through of the WWW based on the client Mosaic
  - Developed by Marc Andreessen and Eric Bina (University of Illinois)
  - Originally for X-Windows systems
  - Available as source code via FTP, thus rapid dissemination
  - End of 1993: 500 known web servers generating 1% of the Internet traffic
  - 1995: Marc Andreessen founds Netscape



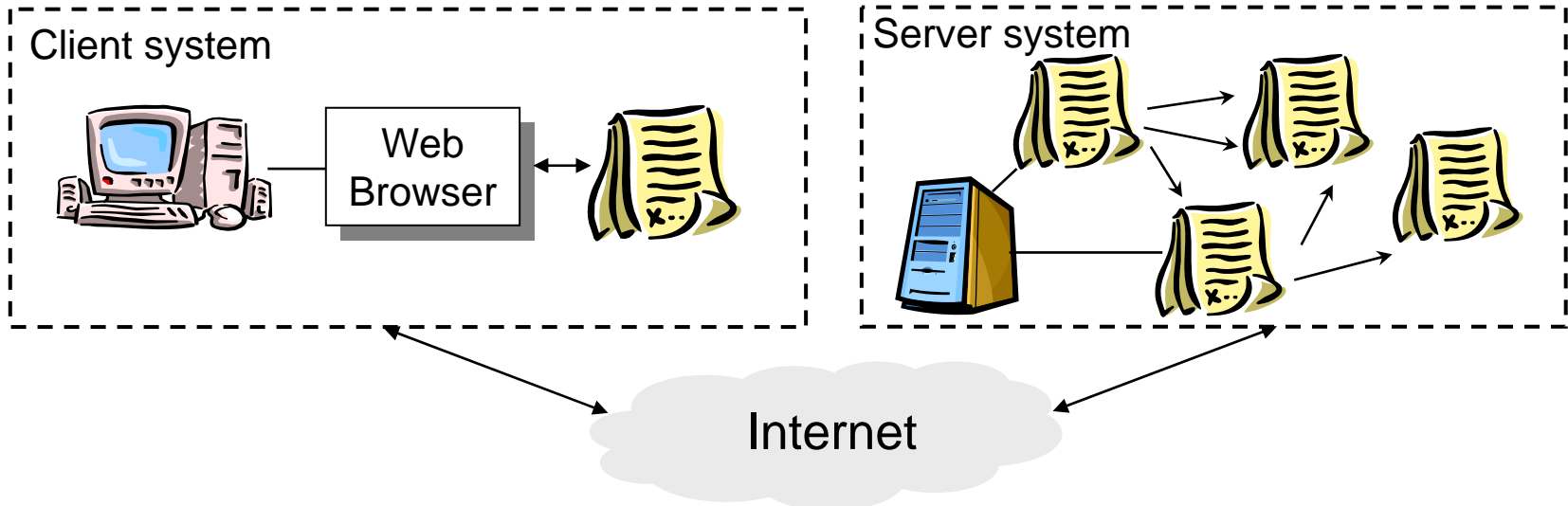


## Development of the WWW (2)

- Foundation of the W3 Consortium in July 1994
  - Goal: further development of the WWW, standardization of HTML
  - More information: <http://www.w3.org>
- End of 1994
  - 10000 servers, 2000 commercial
  - 10 million users
  - Traffic generated equaled roughly all that Shakespeare has written – every second
- And then we all know the history
  - Rise & fall of the .coms, Web 2.0 hype...
  - Today everyone/everything is on/in the Web creating the vast majority of Internet traffic

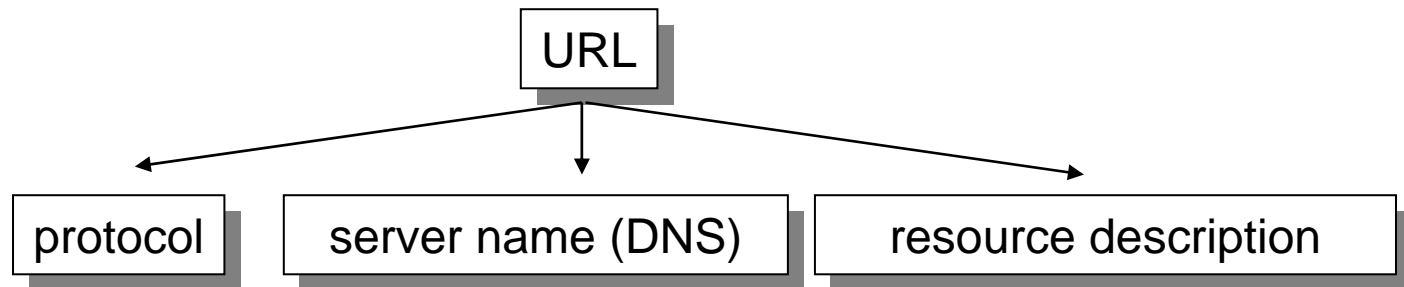
# Classical client/server architecture of the WWW

- Client
  - Contains web browser for displaying hypertext documents, hypermedia object
  - Interprets hyperlinks for navigation, loading of objects
- Server
  - Stores pages as files, runs database which generate pages...
- Open issues
  - Addressing of web pages, resources
  - Transport of web pages, content
  - Content description, syntax of links



# Addressing of web resources

- Uniform Resource Locator (URL)
  - Directs the client software to a certain resource
  - Also applicable for content of other servers (FTP, E-Mail...)
  - E.g., <http://www.mi.fu-berlin.de/index.html>



- Identification of objects on a server via resource description
  - WWW: web page
  - FTP: file
  - Mail: receiver of email
- Web browsers support different protocols/applications
  - e.g. <http://>, <ftp://>, <mailto://>, <telnet://>, <soap://>

- HTTP (HyperText Transfer Protocol)
  - Versions 0.9 and 1.0 described in RFC1945
  - Since January 1997 version 1.1 (RFC2068)
  - Primary usage for web page transfer
    - however, almost everything can be transferred...
- Characteristics
  - ASCII-based application layer protocol
  - Uses a reliable TCP connection (default: port 80)
  - Short-lived connections with version 1.0 (connection per request), changed in 1.1
- Example commands
  - GET: Request a certain resource
  - HEAD: Request the header information of a resource
  - POST: Submits data to a resource
  - PUT: Uploads a new resource

# Example of an HTTP request and HTTP response

## HTTP-Client

```
GET /index.html HTTP/1.1
Host:www.abc.com
Pragma: no-cache
....
```

## HTTP-Server

```
HTTP/1.1 200 OK
Date: Fri, 24 Sep 1999 09:45:51 GMT
Server: Apache/1.3.6 (Unix)
Transfer-Encoding: chunked
Content-Type: text/html

<HTML>
Document according to HTML
</HTML>
```

- Request from client to server
- Command line: <command> <URL> <version>
- Here: client request most current version of the resource, not cached

## TCP connection was already established

- Response line
- Date
- Server
- Coding information
- Type of content
- Main part

# Example

**Request to port 80:** GET / HTTP/1.0  
**or:** GET / HTTP/1.1  
 Host: www.inf.fu-berlin.de

## Response from server

```
HTTP/1.1 200 OK
Date: Wed, 30 Oct 2002 19:44:26 GMT
Server: Apache/1.3.12 (Unix) mod_perl/1.24
Last-Modified: Wed, 30 Oct 2002 13:16:31 GMT
ETag: "2d8190-2322-3dbfdbaf"
Accept-Ranges: bytes
Content-Length: 8994
Connection: close
Content-Type: text/html
```

non persistent



```
<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">
    <link rel="stylesheet" type="text/css" href="http://www.inf.fu-berlin.de/styles/homepage.css">
    <!--script language="JavaScript" src="fuinf.js"-->
    <!--/script-->
  </head>

  <body onResize="self.location.reload();">
  ...
```

- HTML (HyperText Markup Language)
  - HTML documents are structured text documents (but not XML)
  - HTML tags describe the presentation/meaning (cf. T<sub>E</sub>X)
    - HTML tags in plain text
    - E.g. `<b> Bold Font </b>`
  - Documents contain header and body
    - Header defines general properties of the document
    - Body contains the real content
      - Subdivided into headings, paragraphs etc.
  - Hyperlinks refer to labels or other resources
  - Integration of arbitrary non-text elements (e.g. graphics)
  - Each browser can adapt (within certain limits) presentation to local capabilities
- Standardization has reached HTML 4.01
  - Integration of scripting and Cascading Style Sheets (CSS)
  - Transition to XHTML (1.1, 2.0 under discussion)

# Example: A very minimalistic HTML document

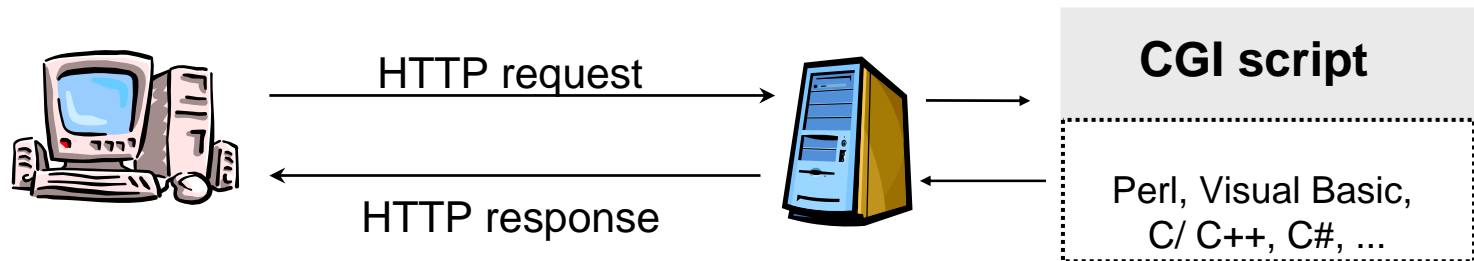
- `<...>`: Start tag
- `</...>`: End tag
- Several possibilities for structuring the document
  - E.g. `<p>`: paragraph;
  - `<br>`: new line;
  - `<hn>`: heading at level *n*

```
<html>
  <head>
    <title> Simple Example </title>
  </head>
  <body>
    <p>This is a simple document.</p>
  </body>
</html>
```

- Several ways for emphasizing characters
  - E.g. `<em>`: emphasize; `<i>`: italic; `<b>`: bold
- Standard character set: ISO 8859-1 (8-Bit, ASCII contained as subset), HTML 3 defines more (e.g. Unicode, 16 bit)
- Referencing of other resources
  - `<a href="www.inf.fu-berlin.de/index.html"> Name </a>`

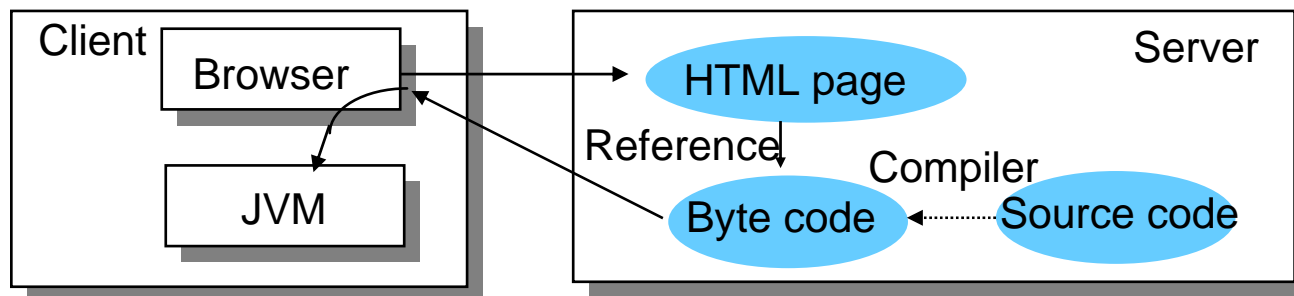


- SSI (Server Side Includes)
  - Simplest way for dynamic documents
  - Idea: Integrate server commands into the HTML document
  - Server replaces the commands by dynamically generated output (e.g. content of a file, output of a program)
  - E.g.: `<!--#include file="lastupdate.txt" -->`
- CGI (Common Gateway Interface)
  - WWW server calls additional program (CGI script) via URL
  - Server transmits output of the CGI script (HTML-Code!) to the client



# Java applets

- Java: developed in 1995 by Sun Microsystems
  - Originally intended for embedded devices
- Java is (almost) platform independent through translation of the sources into Java byte code
- Java Virtual Machine (JVM) interprets Java byte code
- Integration into a web page
  - New HTML tags `<APPLET>`, `<OBJECT>`, `<EMBED>`
  - Example: `<APPLET CODE=game.class WIDTH=100 HEIGHT=200> </APPLET>`
- Client downloads and executes applet



- Similar: CLR (Common Language Runtime) from Microsoft
  - Virtual machine for C++, C#, Haskell, Java, Cobol, ...