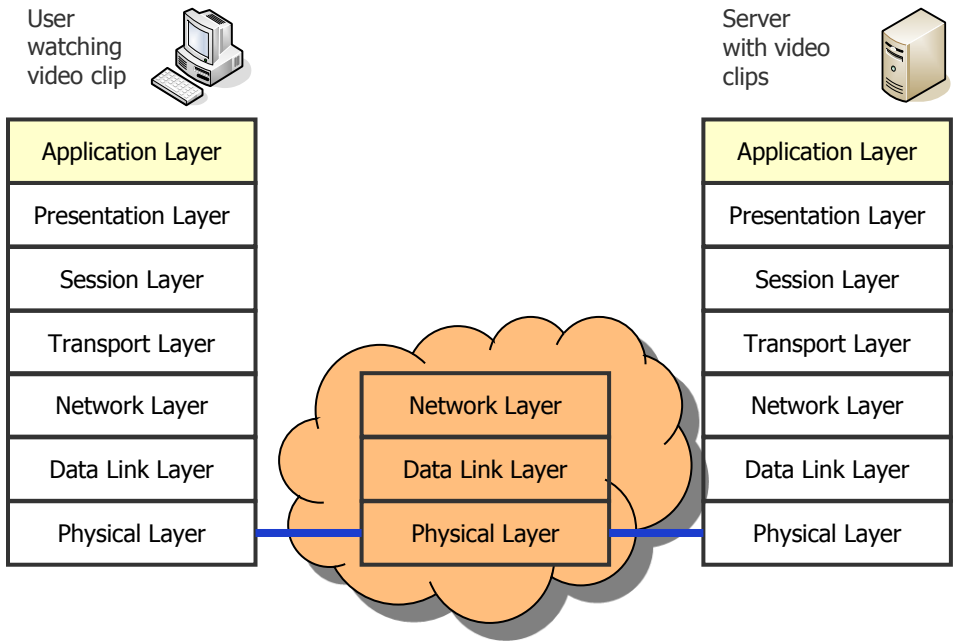
Freie Universität Berlin

Telematics
Chapter 9: Application Layer



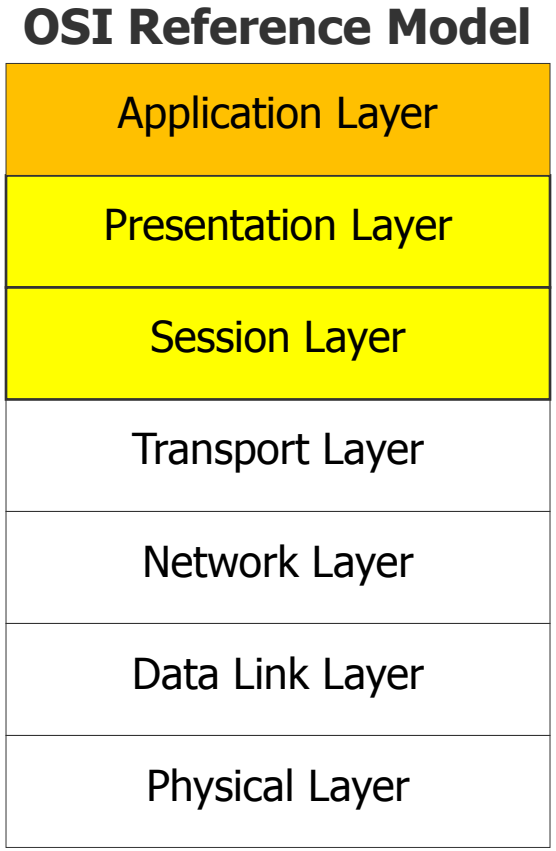
Univ.-Prof. Dr.-Ing. Jochen H. Schiller
Computer Systems and Telematics (CST)
Institute of Computer Science
Freie Universität Berlin
http://cst.mi.fu-berlin.de

Contents

- Design Issues
- Layer 5 and Layer 6
- Domain Name System (DNS)
- Electronic Mail
  - POP3 and IMAP
- File Transfer Protocol (FTP)
- World Wide Web
  - HTTP
  - HTML
- Simple Network Management Protocol (SNMP)
- Firewalls

Design Issues

- Lower layers
  - The meaning of all lower layers is to provide a communication facility for applications
  - Are not really designed for end users
- Application layer
  - Application layer protocols work on top of the transport layer protocols
  - Implement applications for end users
  - A large set of different applications (protocols) with totally different requirements and assumptions
  - According to ISO/OSI three layers, but in the Internet exists only one layer



Layer 5: Session Layer and Layer 6: Presentation Layer

Layer 5: Session Layer

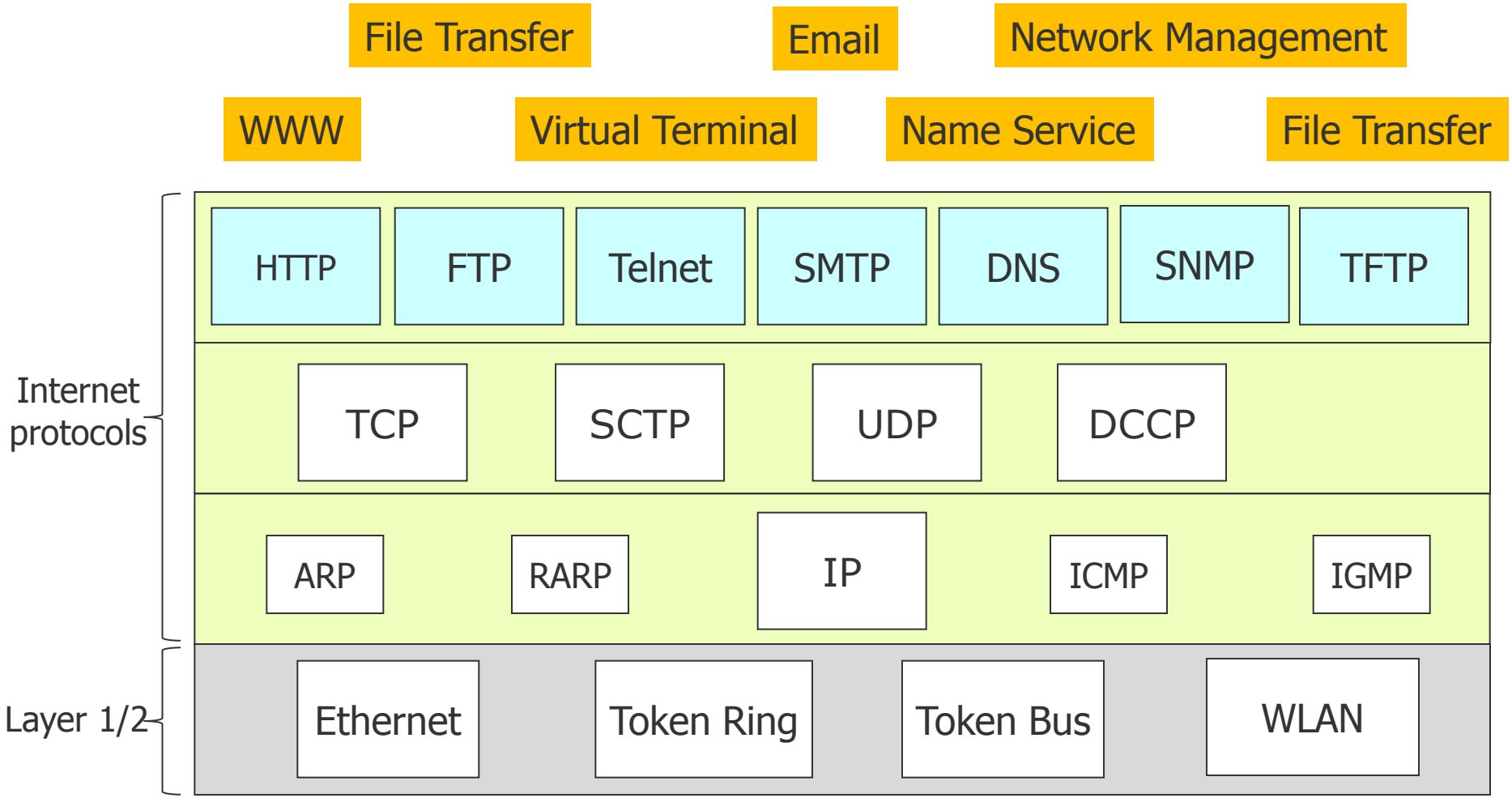
- Layer 5 is the lowest of the application orientated layers
- Layer 5 **controls dialogs:**
  - Synchronization of partner instances at **synchronization points**: data can have been transmitted correctly but have to be nevertheless partially retransmitted. (e.g. Crash of a sender in the mid of the data transmission process)
    - Synchronization points can be set on Layer 5 at arbitrary times
    - If connection breaks down, transmission can restart at the last synchronization point
  - Dialog management during half-duplex transmission
    - Layer 5 controls the order in which the peers send data
  - Connection establishment, data transmission, and connection termination for layer 5 to 7.
  - Use of different tokens for the assignment of transmission authorizations, for connection termination, and for the setting of synchronization points.
- Internet
  - Not explicitly present, however, mechanisms like cookies offer certain synching, TCP offers some session management – but the majority of functions is application specific

Layer 6: Presentation Layer

- Layer 6 hides the use of different data structures or differences in their internal representation
  - The same meaning of the data with the sender and the receiver is guaranteed
- Adapt character codes
  - ASCII 7-bit American Standard Code for Information Interchange
  - EBCDIC 8-bit Extended Binary Coded Digital Interchange Code
  - Unicode
- Adapt number notation
  - 32/40/56/64 bits
  - Little Endian (byte 0 of a word is right) vs. Big Endian (byte 0 is left)
  - ➡Abstract Syntax Notation One (ASN.1) as transfer syntax
- Substantial tasks of layer 6:
  1. Negotiation of the transfer syntax
  2. Mapping of the own data to the transfer syntax
  3. … and further data compression, data encryption (source coding)
- Internet: not explicitly present, however, e.g. HTTP has some of the above aspects, as well as MIME encoding

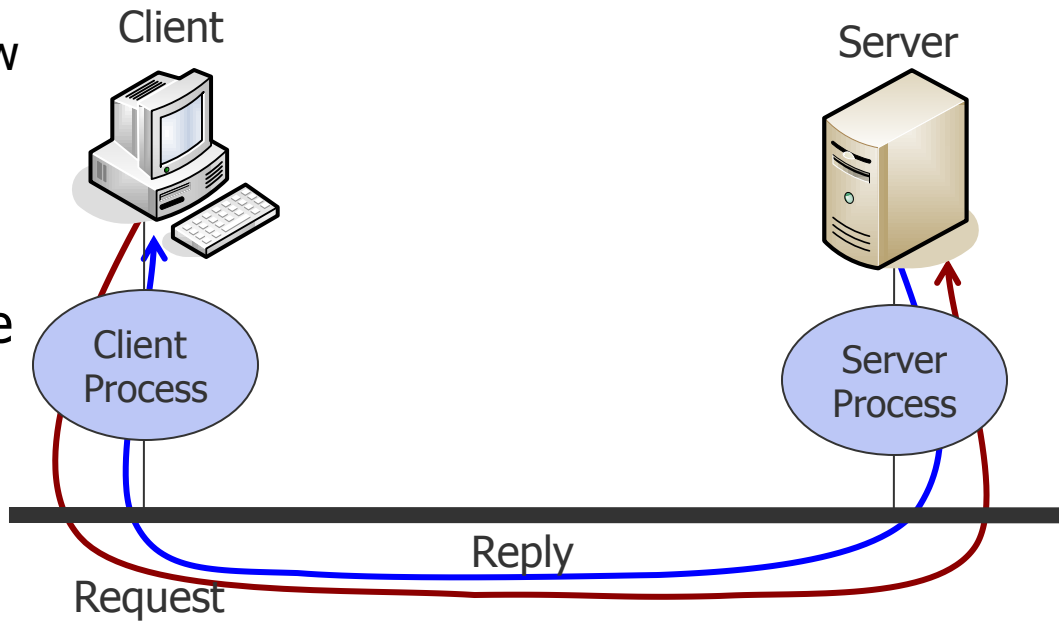
Layer 7: Application Layer

Application Protocols in the TCP/IP Reference Model



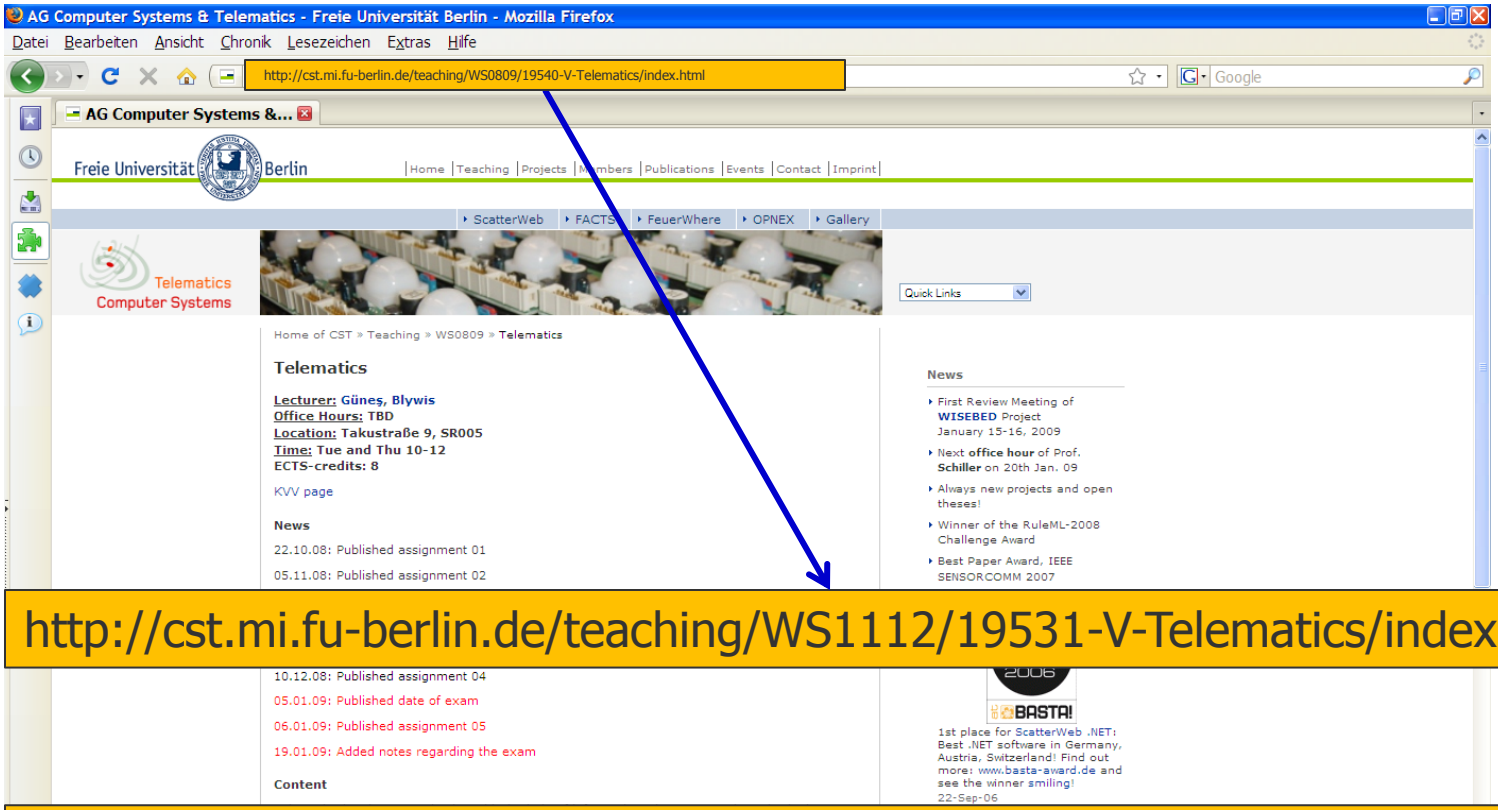
Application Protocols in the TCP/IP Reference Model

- Protocols of the application layer are common communication services
- Protocols of the application layer are defined for special purposes and specify ...
  - the types of the messages
  - the syntax of the message types
  - the semantics of the message types
  - rules for definition, when and how an application process sends a message resp. responses to it
- Usually **client/server** structure
- Processes on the application layer use TCP(UDP)/IP-Sockets



Domain Name System (DNS)

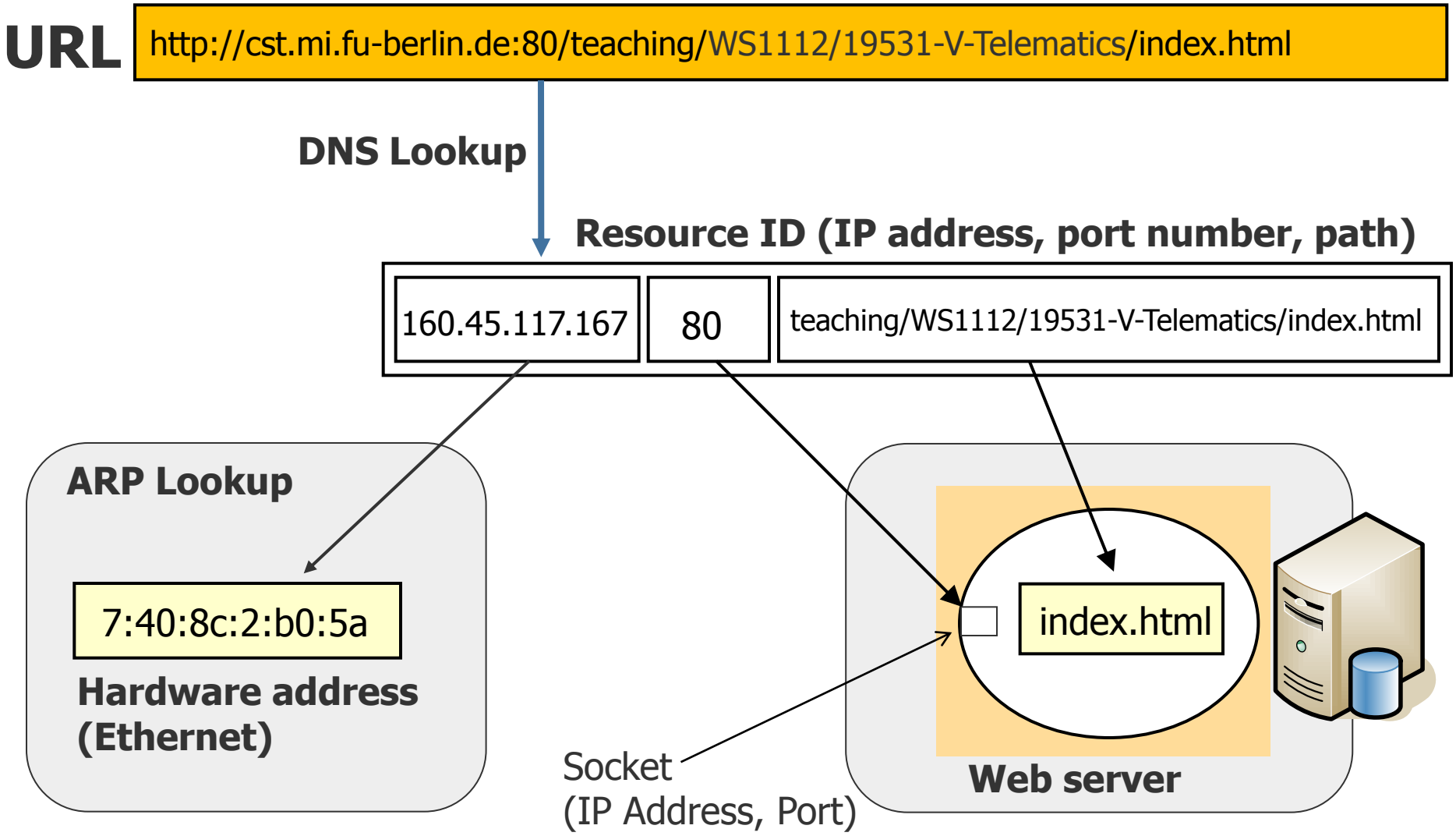
Access to Remote Computers



http://cst.mi.fu-berlin.de/teaching/WS1112/19531-V-Telematics/index.html

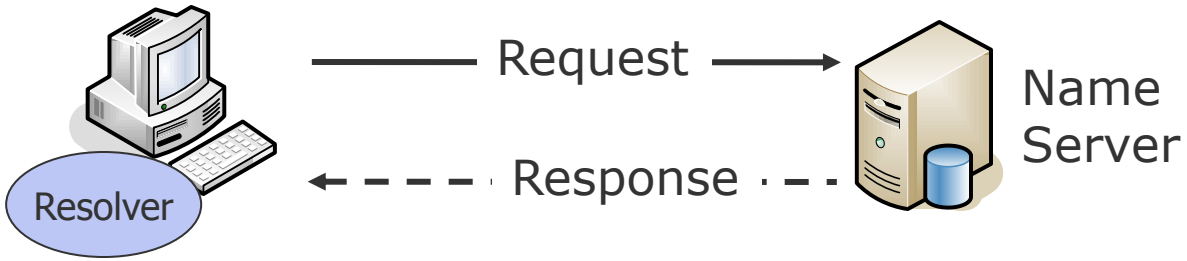
- **IP addresses** are difficult to remember for humans, but computers can deal with them perfectly.
- **Symbolic names** are simpler for humans to handle, but computers can unfortunately not deal with them.
- Question: How to map IP addresses to symbolic names and vice versa?

Access to Remote Computers



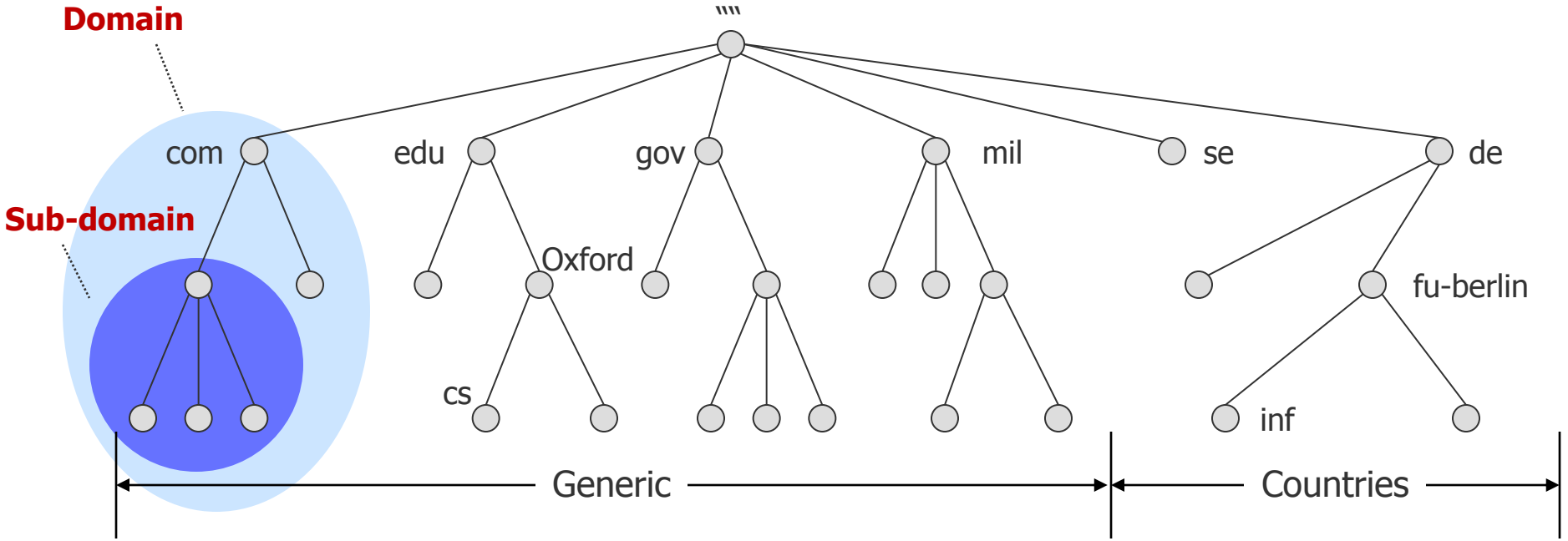
DNS: Concept

- DNS manages the **mapping of names to IP** addresses (and further services)
- DNS is a **distributed database**, i.e., the parts are subject to local control
- The structure of the used **name space** of the database shows the administrative organization of the Internet
- Data of each **local area** are available by means of a client/server architecture in the entire network
- Robustness and speed of the system is achieved by **replication** and **caching**
- Main components:
  - Name Server: Server which manages information about a part of the database
  - Resolver: Client which requests naming information from the server



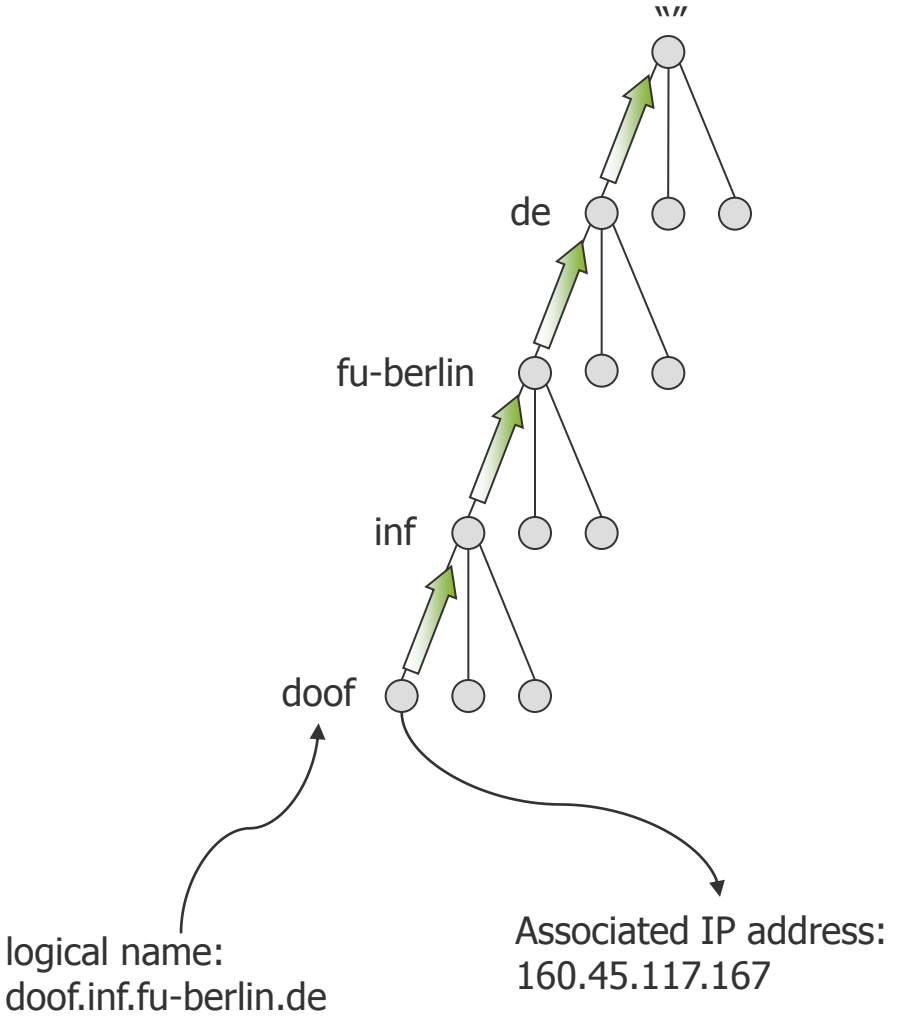
The DNS Database

- The DNS database is structured as a tree
  - Each node of the tree has a **label**, which identifies it relatively to the parent node
  - Each (internal) node is **root of a sub-tree**
  - Each sub-tree represents a **domain**
  - Each domain can be divided into **sub-domains**



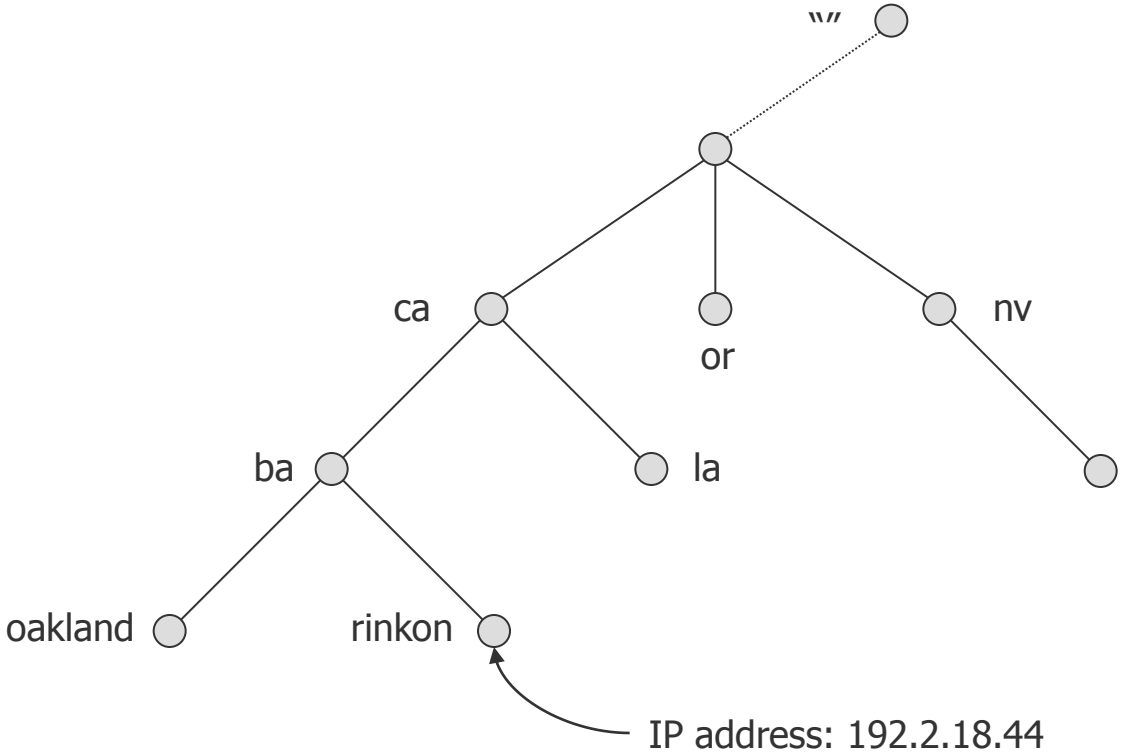
The DNS Database

- The name of a domain consists of a sequence of labels beginning with the root of the domain and going up to the root of the whole tree
- Each label is separated by "."
- In the leaf nodes the IP addresses associated with the names are stored



The DNS Database

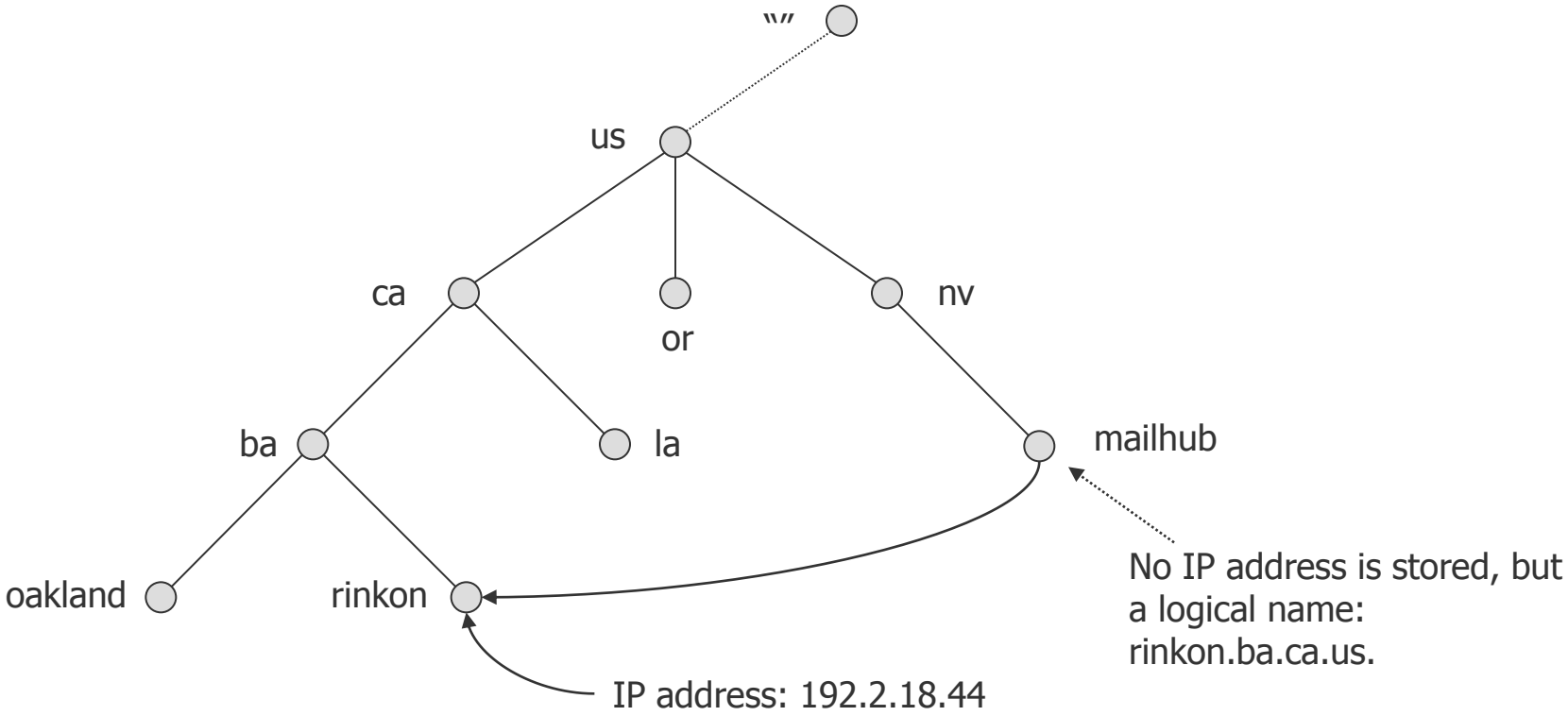
- The names of the domains serve as index for the database
- Each computer in the network has a domain name which refers to further information concerning the computer



The data associated with a domain name are stored in **Resource Records** (RR)

The DNS Database

- Computers can have one or more secondary names
  - Domain Name Aliases
- Aliases are pointers of one domain name to another one
  - Canonical Domain Name



The DNS Database

- The reverse tree represents the Domain Name Space
  - The depth of the tree is limited to 127 levels
  - Each label can have up to 63 characters
  - The whole domain name can have up to 255 characters
  - A label of the length 0 is reserved for the root node ("")
  - The **Fully Qualified Domain Name** (**FQDN**) is the absolute domain name, which is declared with reference to the root of the tree

- Example:
  - inf.fu-berlin.de.
  - www.wikipedia.org.

- Domain names which are declared not with reference to the root of the tree, but with reference to another domain, are called **relative domain names**

The DNS Database

- A domain consists of all computers whose domain name is within the domain
- Leafs of the tree represent individual computers and refer to network addresses, hardware information and mail routing information
- Internal nodes of the tree can describe both a computer and a domain
- Domains are denoted often relatively or regarding their level:
  - Top-Level Domain: child of the root node
  - First Level Domain: child of the root node (top-level domain)
  - Second Level Domain: child of a first level of domain
  - etc.

The DNS Database – Top-Level Domains

- Originally the name space was divided into seven Top-Level Domains:
  - com: commercial organizations
  - edu: educational organizations
  - gov: government organizations
  - mil: military organizations
  - net: network organizations
  - org: non-commercial organizations
  - int: international organizations
- Additionally, each country got its own top-level domain
- The name space was extended in the meantime by further top-level domains
  - aero, biz, coop, info, museum, name, pro (http://www.icann.org/tlds)
- Within the individual top-level domains, different conventions for name structuring are given:
  - Australia: edu.au, com.au, etc.
  - UK: co.uk (for commercial organizations), ac.uk (for academic organizations), etc.
  - Germany: completely unstructured

DNS

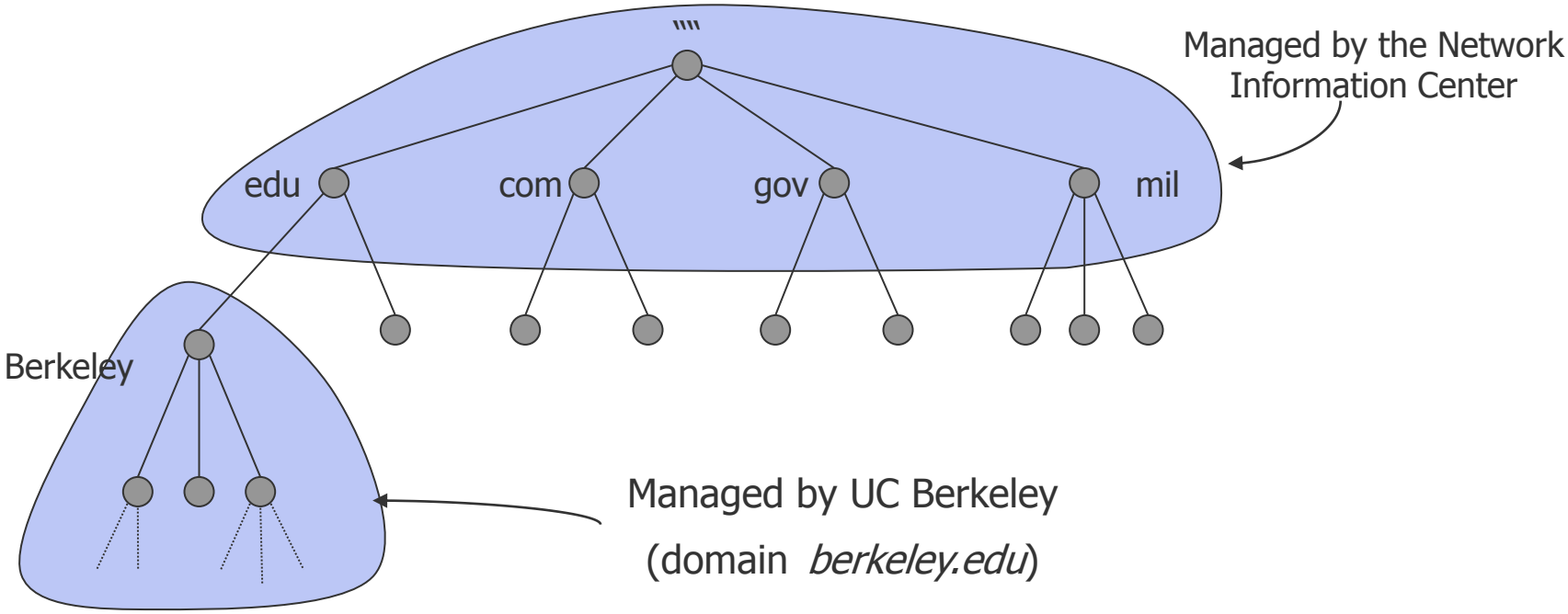
Name Servers and Zones

Name Servers, Zones, and Domains

- Name Server
  - Has database for the name space
  - Part of the name space a name server knows is called zone
  - Name server has authority of the zone
  - May manage multiple zones

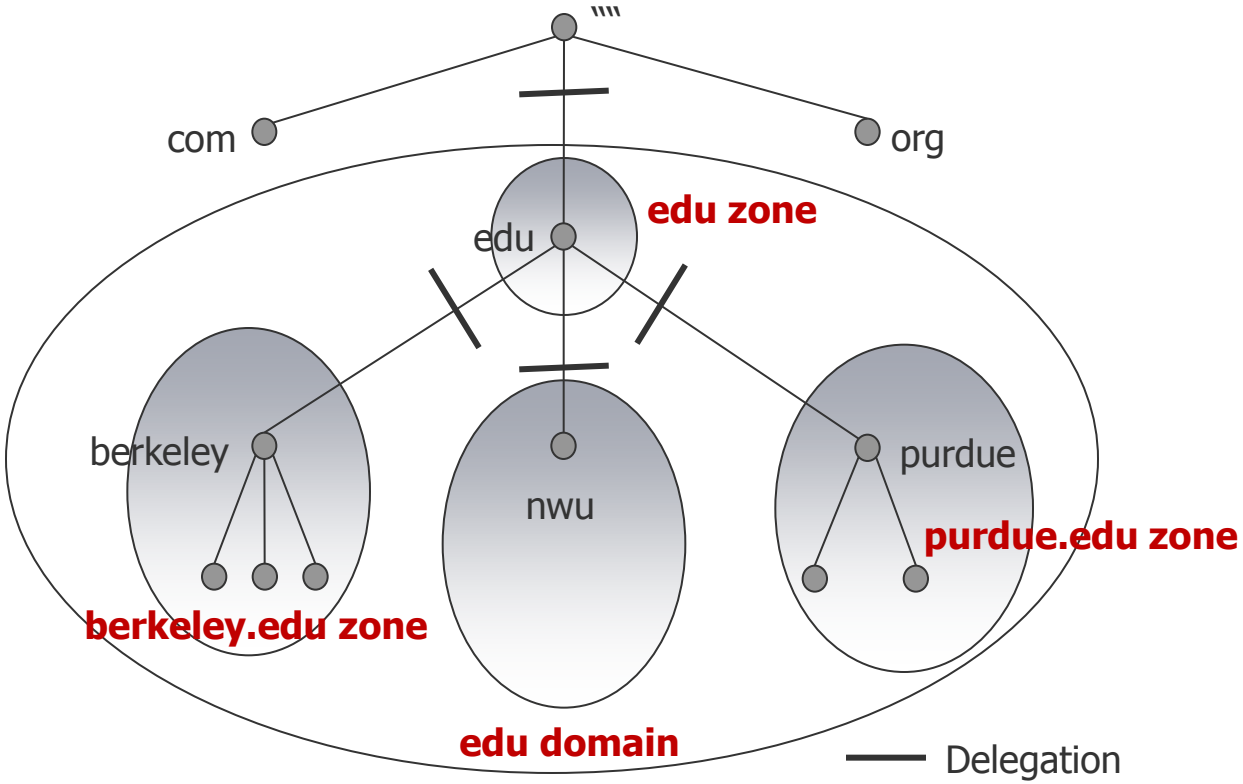
- Domain
  - A domain is managed by an organization
  - The responsible organization can split a domain into sub-domains and delegate the responsibility for them to other organizations
  - The parent domain manages pointers to the roots of the sub-domains to be able to forward requests to them
  - The name of a domain corresponds to the domain name of the root node

Name Servers, Zones, and Domains



Name Servers, Zones, and Domains

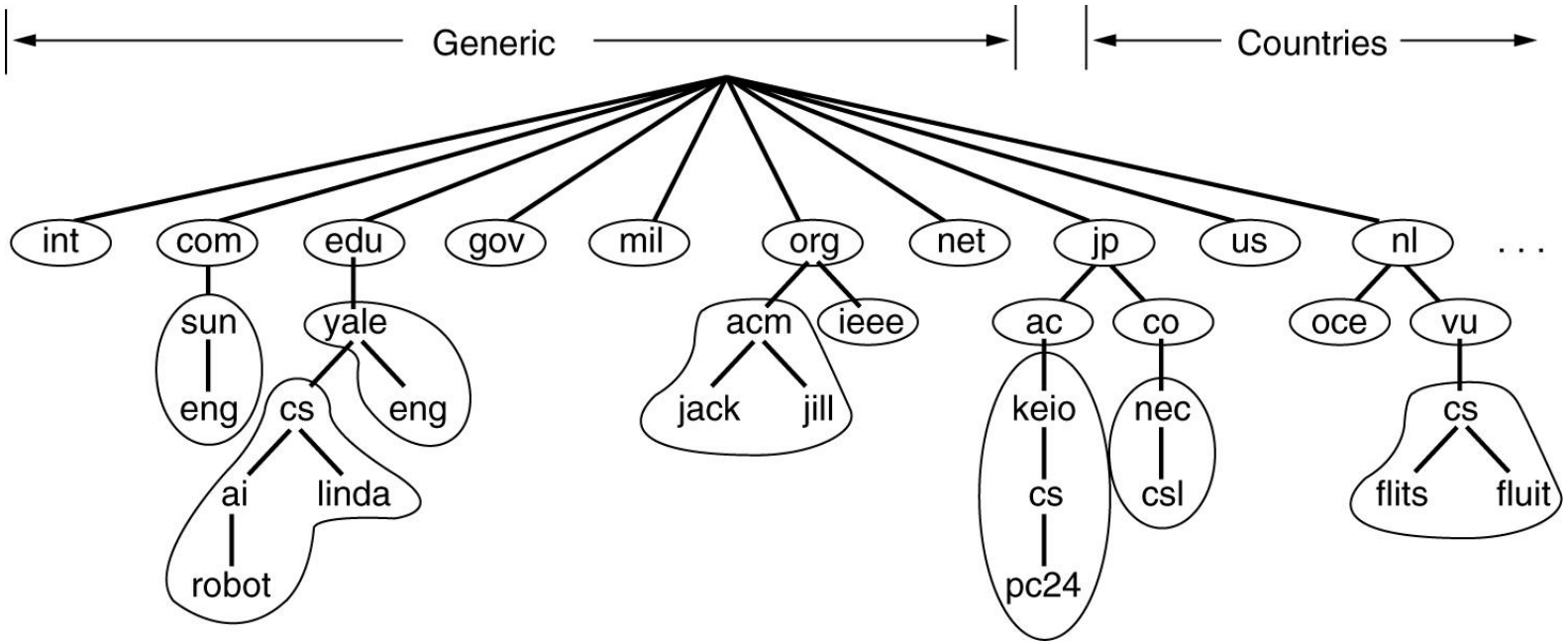
- Domain and zone are different concepts:



- Zones are (except within the lowest levels of the tree) smaller than domains, therefore servers have to manage less name information

Name Servers, Zones, and Domains

There are no guidelines how domains are divided into zones. Each domain can select a dividing for itself.



Some zones (e.g. edu) do not manage IP addresses. They only store references to other zones as information.

Types of Name Servers

- Types of Name Servers
  - The **Primary Master** of a zone (**Master**) reads the data from a file
    - **Zone Data Files**
  - A **Secondary Master** of a zone (**Slave**) receives the data from another name server, which is authoritative for the zone.
    - Primary Master or another Secondary Master
  - **Primary master** and **secondary masters** are **authoritative** for the zone
  - The distinction between primary master and secondary master serves for a controlled replication
    - Performance and fault tolerance

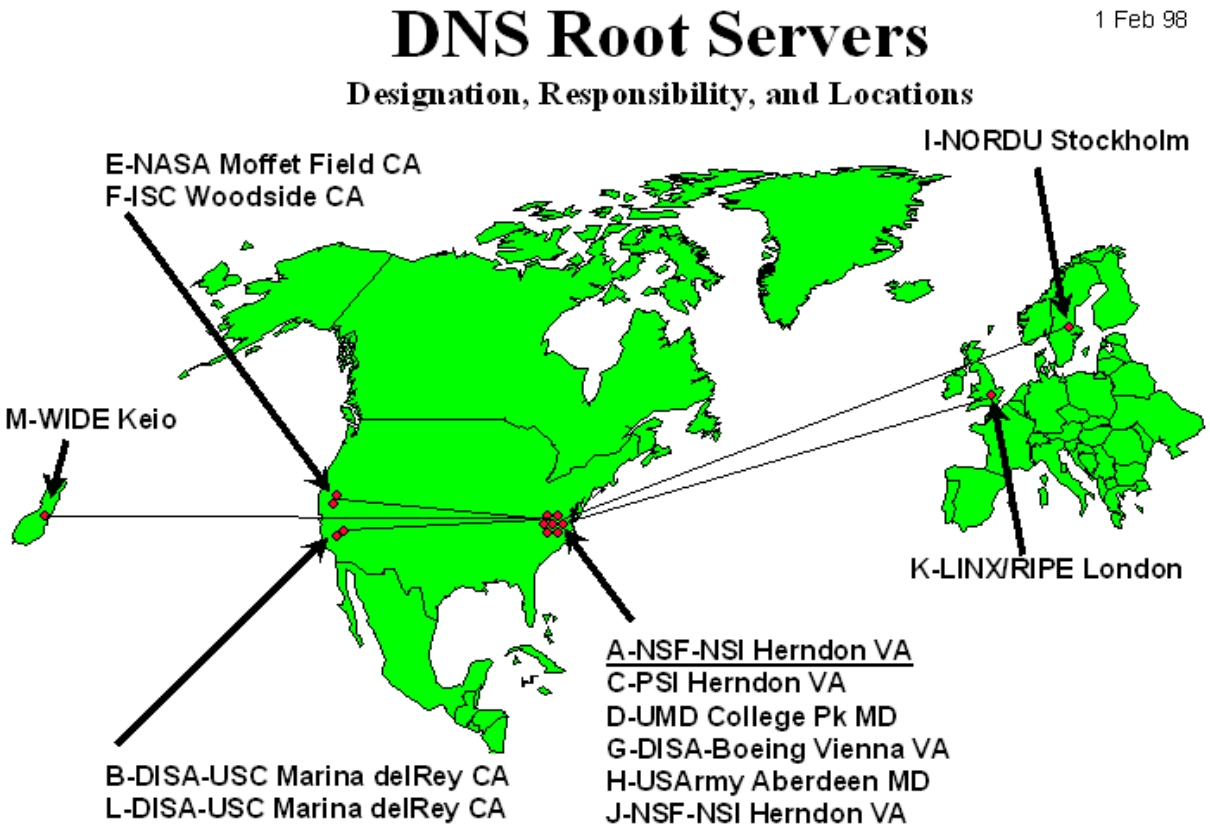
- Resource Records
  - The resource records describe the zone's information
  - The resource records describe all computers in the zone as well as information concerning the delegation of sub-domains

DNS

Root Name Server

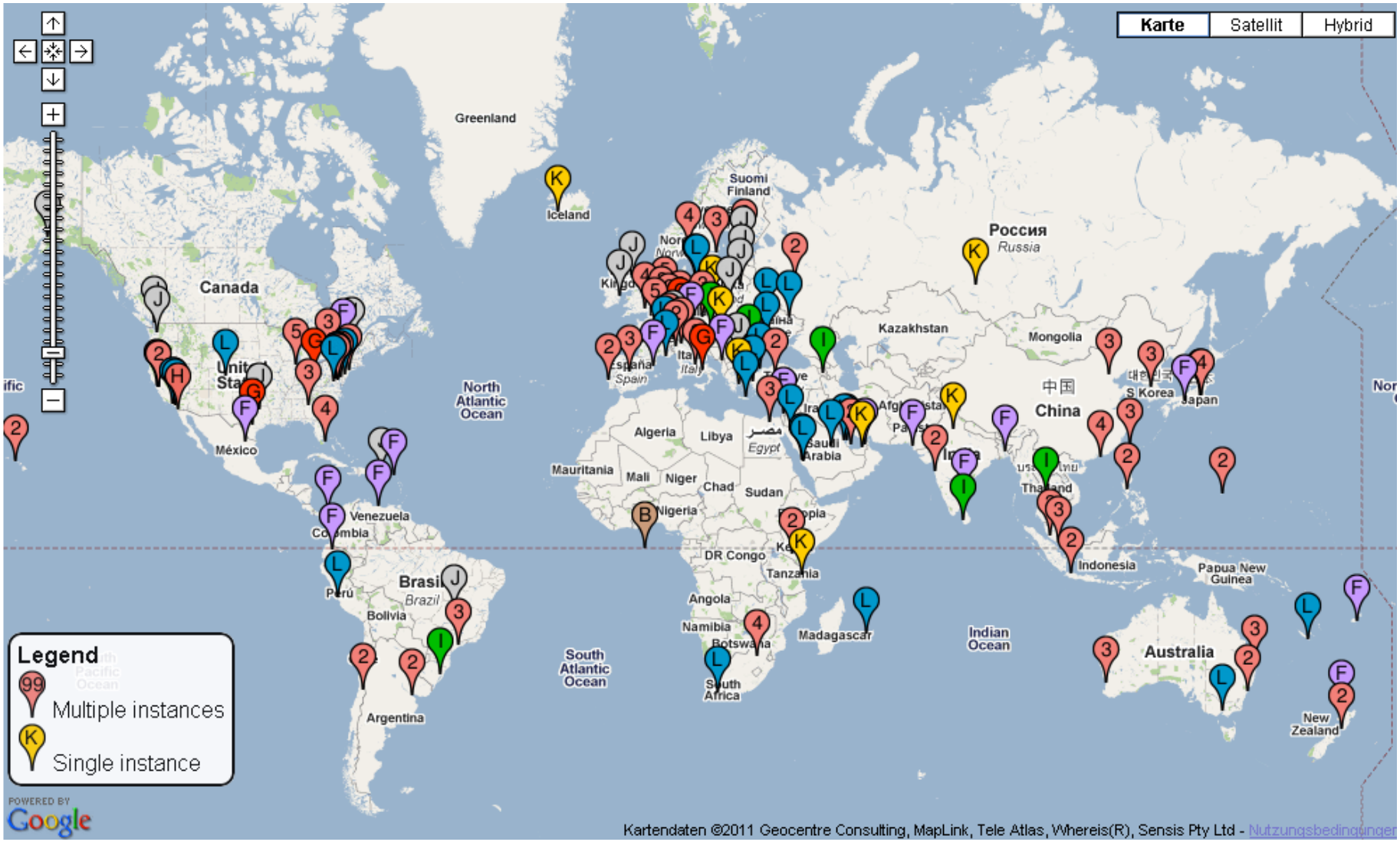
Root Name Server

- Requests to which a name server cannot answer, are handed upward in the tree
- Name server on the upper levels are heavily loaded
- Inquiries, which go into another zone, often run over the root name server
- Thus, the root name server must always be available
- Therefore: **replication** - there are 13 instances of the root name server, more or less distributed over the whole world
- http://www.root-servers.org



Problem: very central placement of the servers!

Root Name Server

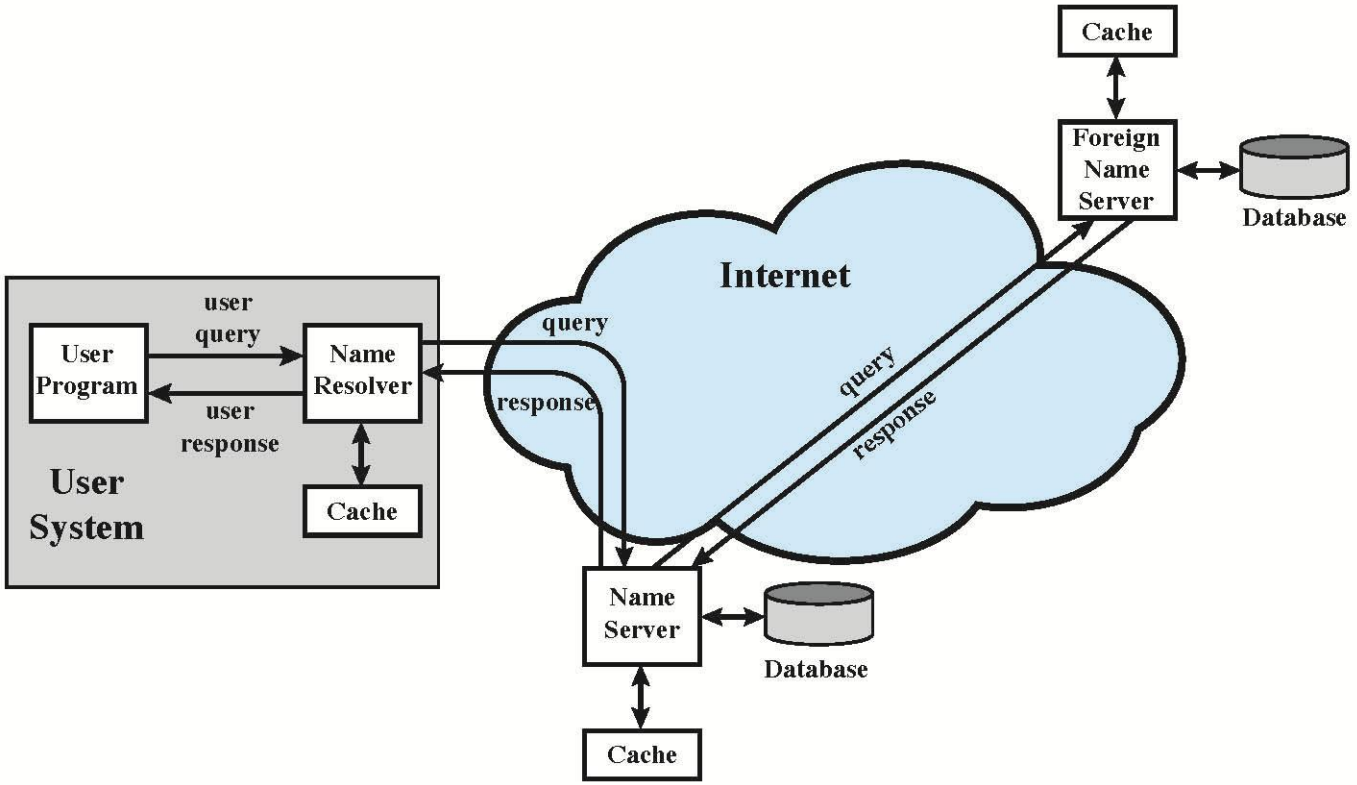


DNS
Name Resolution

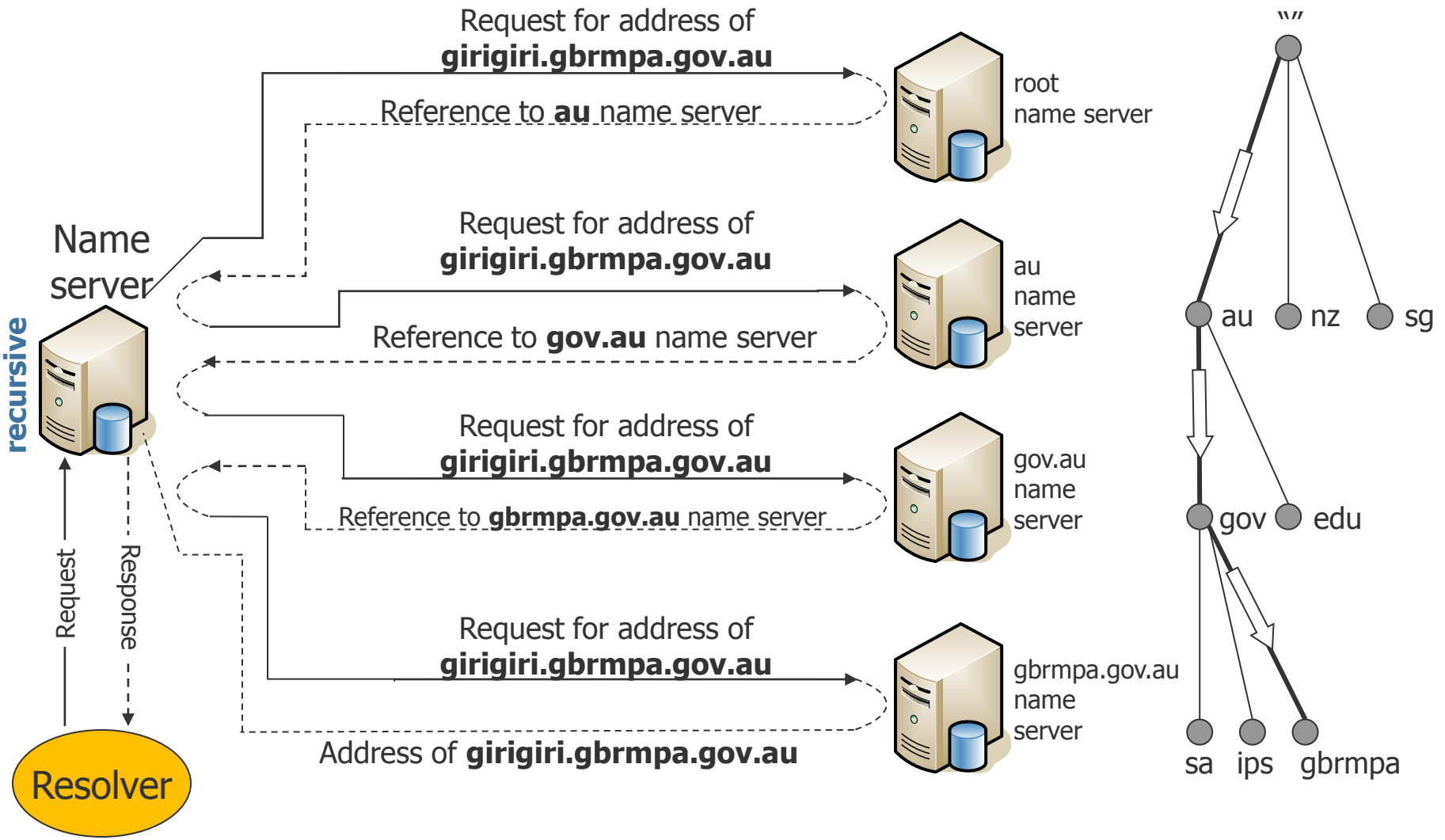
Name Resolution: Recursive and Iterative

- Two types of name resolution: recursive and iterative
  - Recursive resolution
    - The name server replies either with the searched information or an error message
    - The name server is responsible to contact as much other name servers as necessary to find the searched information
  - Iterative resolution
    - A name server replies with the searched information if it has it or with the address of another responsible name server
    - The resolver has to connect other name servers if it does not get the answer

Name Resolution: Recursive and Iterative



Name Resolution: Recursive and Iterative



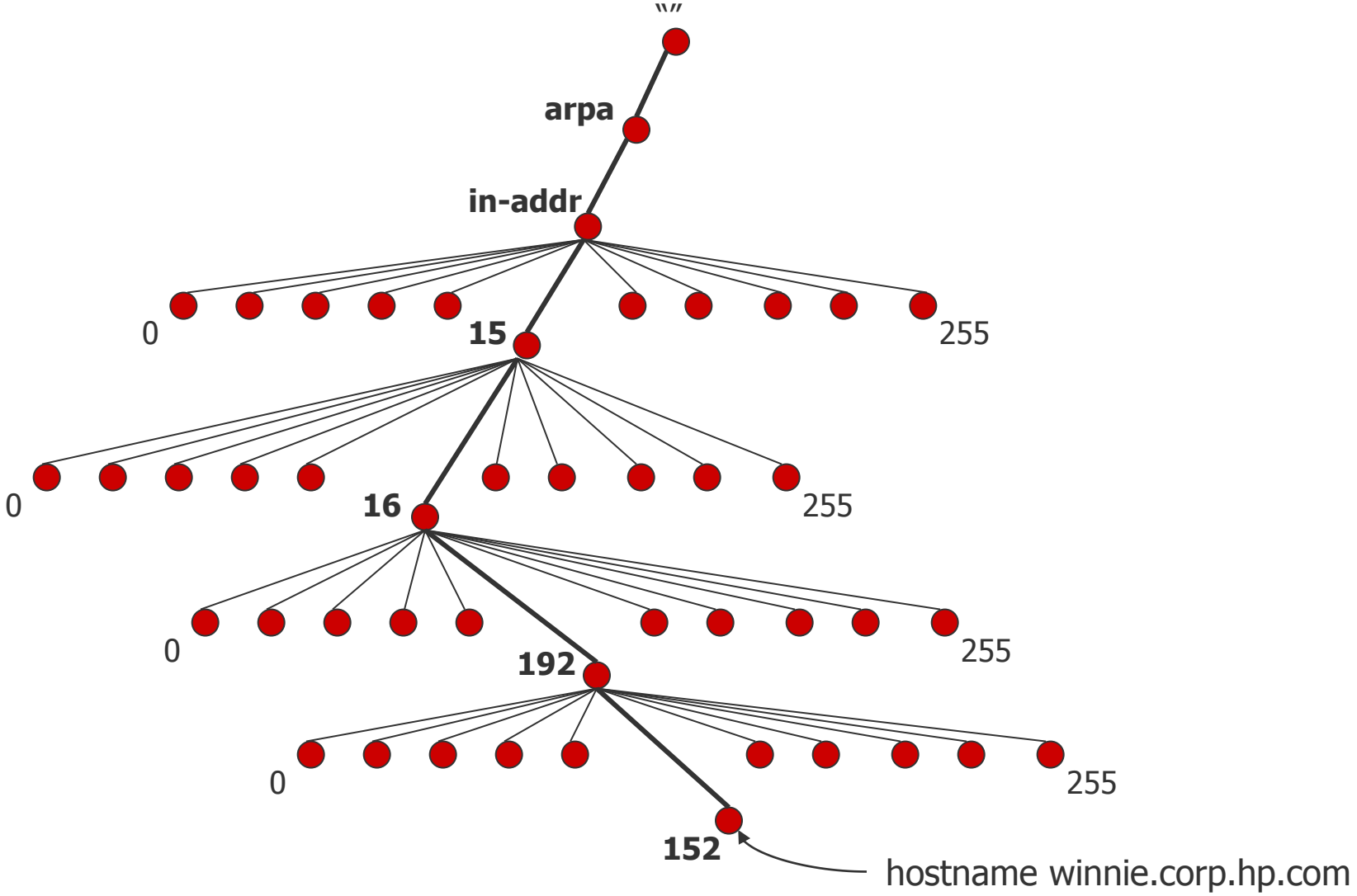
DNS

Mapping of Addresses to Names

Mapping of Addresses to Names

- Information in the database is indicated by names
  - Mapping of a name to an address is simple
- Mapping of an address onto a name is more difficult to realize (complete search of name space)
- Solution:
  - Place a special area in the name space, which uses addresses as label
    - Domain: in-addr.arpa
  - Nodes in this domain are marked in accordance with the usual notation for IP addresses (four octets separated by points)
  - The in-addr.arpa domain has 256 sub-domains, each of which again having 256 sub-domains, …
  - On the fourth level, the appropriate resource records are assigned with the octet, which refers to the domain name of the computer or the network with the indicated address
  - The IP address appears backwards because it is read beginning with the leaf node (IP address: 15.16.192.152
    ➡ sub-domain: 152.192.16.15.in-addr.arpa)

Mapping of Addresses to Names



DNS
Resource Records

Resource Records

- Entries in the zone data files of the name servers are resource records
- General structure: (label, ttl, class, type, value)

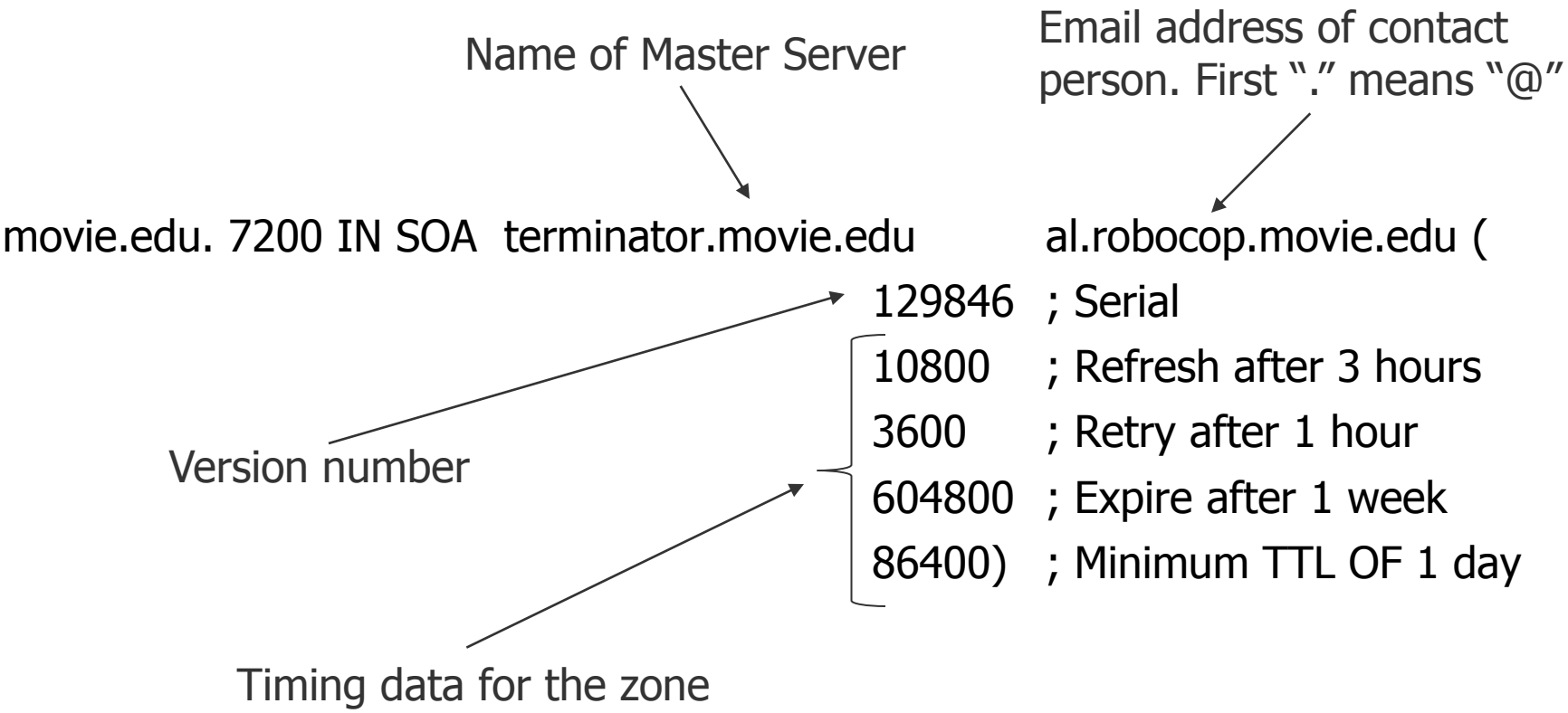
Type	Used in	Description
A	Host	Address of a host; needed for name resolution
CNAME	Node	Canonical name, i.e., reference (alias) to the true name
HINFO	Host	Host information, additional information about the host (CPU, operating system)
MINFO	Domain	Mailbox or mail list information, maps a mailbox or mail list name to a host
MX	Domain	Mail exchange, refers to the mail server of the domain
NS	Zone	Refers to the authoritative name server for the zone
PTR	Host	Domain name pointer, used for the mapping of an address to a name
SOA	Zone	Indicates the authority for the zone data
SRV	Domain	Refers to a server which offers a certain service in the domain
TXT	Arbitrary	Other useful information
WKS	Host	Well-known services, may list the available services at this host.

Example: Resource Records in a Zone File

```
; Authoritative data for cs.vu.nl
cs.vu.nl.          86400   IN  SOA     star boss (952771,7200,7200,2419200,86400)
cs.vu.nl.          86400   IN  TXT     "Divisie Wiskunde en Informatica."
cs.vu.nl.          86400   IN  TXT     "Vrije Universiteit Amsterdam."
cs.vu.nl.          86400   IN  MX      1 zephyr.cs.vu.nl.
cs.vu.nl.          86400   IN  MX      2 top.cs.vu.nl.

flits.cs.vu.nl.    86400   IN  HINFO   Sun Unix
flits.cs.vu.nl.    86400   IN  A       130.37.16.112
flits.cs.vu.nl.    86400   IN  A       192.31.231.165
flits.cs.vu.nl.    86400   IN  MX      1 flits.cs.vu.nl.
flits.cs.vu.nl.    86400   IN  MX      2 zephyr.cs.vu.nl.
flits.cs.vu.nl.    86400   IN  MX      3 top.cs.vu.nl.
www.cs.vu.nl.      86400   IN  CNAME   star.cs.vu.nl
ftp.cs.vu.nl.      86400   IN  CNAME   zephyr.cs.vu.nl

rowboat                    IN  A       130.37.56.201
                           IN  MX      1 rowboat
                           IN  MX      2 zephyr
                           IN  HINFO   Sun Unix

little-sister              IN  A       130.37.62.23
                           IN  HINFO   Mac MacOS

laserjet                   IN  A       192.31.231.216
                           IN  HINFO   "HP Laserjet IIISi" Proprietary
```

Resource Records: SOA Record

- SOA = Start of Authority
  - It indicates that the name server is authoritative for the zone
  - There can be only one SOA record in an appropriate file
- Example:



Resource Records: SOA Record

- Attributes of the SOA record:
  - Serial: Serial number which serves the secondary master for the recognition of new versions of the zone data
  - Refresh: Time interval, at whose expiration the secondary master examines the topicality of its data
  - Retry: time interval; if the secondary master cannot contact the primary master at expiration of the refresh time, then it tries again after expiration of the retry time interval
  - Expire: if the secondary master cannot contact the primary master after the indicated length of time, it stops answering inquiries because it must assume its data is outdated
  - TTL: Refers to all resource records. This value is returned as part of the answer on a request to instruct other servers about the maximal time for caching the data.

Resource Records: NS Record

- NS = Name Server
  - For each name server of a zone a NS record is created
  - Example:
    - movie.edu. IN NS terminator.movie.edu
    - movie.edu. IN NS wormhole.movie.edu
  - There are two name servers in the example; installed on the computers terminator and wormhole

Resource Records: Address and Alias Records

- A = ADDRESS
- CNAME = Canonical Name
  - At least one A record is needed for each host in the zone, CNAME records are optional
  - Example:

```
; Host addresses
localhost.movie.edu.      IN      A       127.0.0.1
robocop.movie.edu.        IN      A       192.249.249.2
terminator.movie.edu.     IN      A       192.249.249.3
diehard.movie.edu.        IN      A       192.249.249.4
misery.movie.edu.         IN      A       192.253.253.2
shining.movie.edu.        IN      A       192.253.253.3
carrie.movie.edu.         IN      A       192.253.253.4
;
; Multihomed host
;
wormhole.movie.edu        IN      A       192.249.249.1
wormhole.movie.edu        IN      A       192.253.253.1
```

Resource Records: Address and Alias Records

```
;
; Aliases
;
bigt.movie.edu.      IN      CNAME     terminator.movie.edu.
dh.movie.edu.        IN      CNAME     diehard.movie.edu.
wh.movie.edu.        IN      CNAME     wormhole.movie.edu.
wh249.movie.edu.     IN      A         192.249.249.1
wh253.movie.edu.     IN      A         192.253.253.1

A = ADDRESS
CNAME = illustrates an alias on its canonical names
```

- For **multi-homed** computers (connected with several networks), an own A record is needed for every secondary name if different aliases are to be stored for the addresses
- For a secondary name, which applies to both addresses, a CNAME record is created

Resource Records: PTR Record

- PTR = Pointer
  - Provides information for the mapping of addresses to names
  - Example:
    - 1.249.249.192.in-addr.arpa. IN PTR wormhole.movie.edu.
    - 2.249.249.192.in-addr.arpa. IN PTR robocop.movie.edu.
    - 3.249.249.192.in-addr.arpa. IN PTR terminator.movie.edu.
    - 4.249.249.192.in-addr.arpa. IN PTR diehard.movie.edu.
  - Addresses should refer only to one name, the original or canonical name

Resource Records: MX Record

- MX = Mail Exchanger
  - MX record serves for the controlling of email routing
  - Specifies an email server responsible for a domain name, which processes or passes on email
  - Additionally, a preference can be indicated if several mail servers are present
  - Example:

    peets.mpk.ca.us. IN MX 10 relay.hp.com.

    - indicates that relay.hp.com is the mail server for peets.mpk.ca.us with the preference 10
  - Only the relative preference value is important; the email server with the smallest value is addressed first

DNS
The DNS Protocol

DNS Protocol

- DNS defines only one protocol format which is used for inquiries and responses:
  - Identification: 16 bits for the definite identification of an inquiry
    - to match requests and responses
  - Flag: 4 Bit, marking of
    1. request/response
    2. authoritative/not authoritative
    3. iterative/recursive
    4. recursion possible
  - „Number of… ": Indication of the contained number of inquiries resp. data records
  - Questions: Names to be resolved

Identification	Flag
Number of Questions	Number of Answers RR
Number of Authority RR	Number of Additional RR
Questions (variable number of RR)	
Answers (variable number of RR)	
Authority (variable number of RR)	
Additional information (variable number of RR)	

  - Answers: Resource records to the previous inquiry
  - Authority: Identification of passed responsible name servers
  - Additional information: further data to the inquiry.
    - If the name searched is only an alias, the belonging resource record for the correct name is placed here.

DNS Extensions

- Dynamic DNS
  - Simple and easy add of DNS data
  - But: Security issues?
- International character sets
  - Original DNS supports only ASCII
- Extended DNS
  - Large data transmission
- Phone number entries
- RFID support
- Geographic location
- Spam defense
  - Accept only emails from hosts which can be successfully resolved
- Security (DNSSEC)
  - Who is who?

DNS
DNS Tools

DNS Tools

- nslookup (deprecated)
  - Lookup DNS information

```
x:₩>nslookup www.google.com
Server:  pyramid.mi.fu-berlin.de
Address:  160.45.110.15

Non-authoritative answer:
Name:    www.l.google.com
Addresses:  209.85.129.104, 209.85.129.147, 209.85.129.99
Aliases:  www.google.com
```

DNS Tools

- Dig: Alternative tool to lookup DNS information

```
x@y:~$ dig www.google.com

; <<>> DiG 9.2.4 <<>> www.google.com
;; global options:  printcmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 27292
;; flags: qr rd ra; QUERY: 1, ANSWER: 4, AUTHORITY: 7, ADDITIONAL: 0

;; QUESTION SECTION:
;www.google.com.                        IN      A

;; ANSWER SECTION:
www.google.com.         9329    IN      CNAME   www.l.google.com.
www.l.google.com.       251     IN      A       209.85.129.147
www.l.google.com.       251     IN      A       209.85.129.99

;; AUTHORITY SECTION:
l.google.com.           5117    IN      NS      a.l.google.com.
l.google.com.           5117    IN      NS      b.l.google.com.
l.google.com.           5117    IN      NS      c.l.google.com.

;; Query time: 1 msec
;; SERVER: 160.45.113.3#53(160.45.113.3)
;; WHEN: Thu Jan 31 09:03:59 2008
;; MSG SIZE  rcvd: 212
```

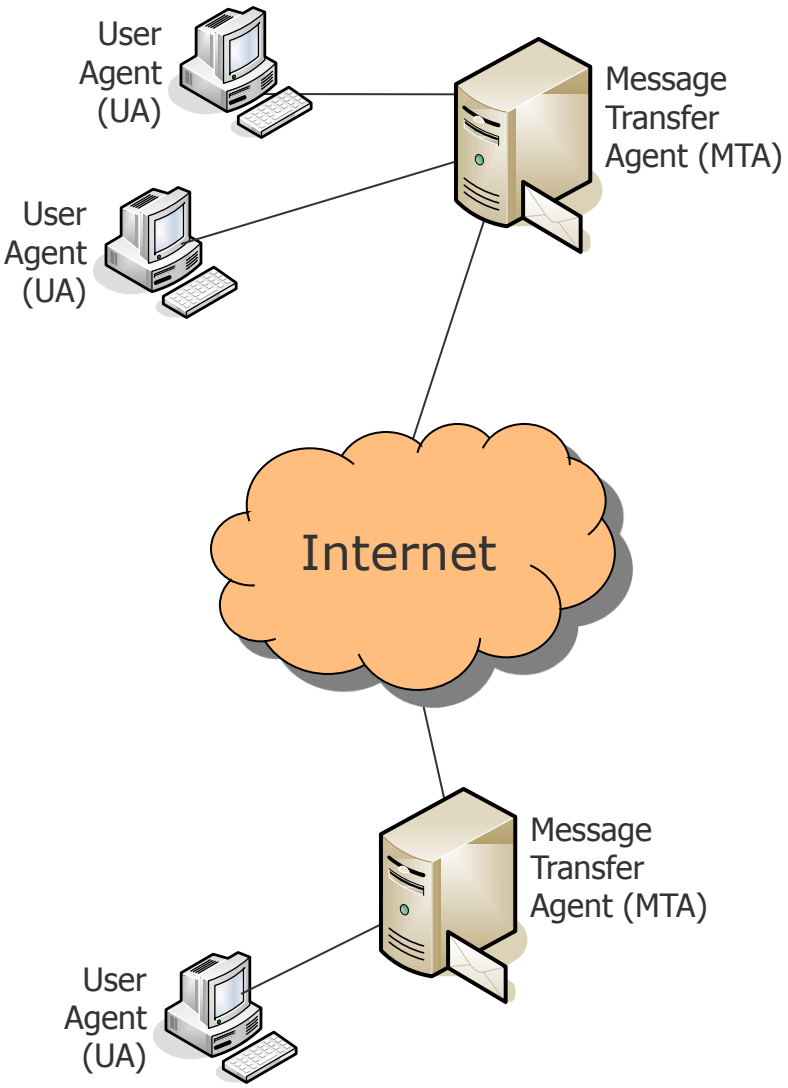
Electronic Mail (Email)

Electronic Mail (Email)

- Early systems
  - A simple file transmission took place, with the convention that the first line contains the address of the receiver of the file.
- Wish list
  - Email to groups
  - Structure of the email
  - Delegation of the administration to a secretary
  - File editor as user interface
  - Mixed media
  - Proof of sending, receipt, security and privacy
- Solution
  - X.400 as standard for email transfer. This specification was however too complex and badly designed. Generally accepted only became a simpler system, cobbled together "by a handful of computer science students":
    - **Simple Mail Transfer Protocol** (**SMTP**)

Electronic Mail (Email)

- An email system generally consists of two subsystems:
- User Agent (UA)
  - Email client
  - Usually runs on the computer of the user and helps during the processing of emails
  - Composition of new and answering of old email
  - Receipt and presentation of email
  - Administration of received email
- Message Transfer Agent (MTA)
  - Email server
  - Runs in the background (around the clock)
  - Delivery of email which is sent by User Agents
  - Intermediate storage of messages for users or other Message Transfer Agents



Structure of an Email

- For sending an email, the following information is needed from the user:
  - Message: usually normal text + attachments, e.g., word file, GIF image, …
  - Destination address: generally in the form mailbox@location, e.g., jochen.schiller@fu-berlin.de
  - Possibly additional parameters concerning priority, security, etc.
- Email formats: two used standards
  - Internet Message Format [RFC 2822]
  - Multipurpose Internet Mail Extensions (MIME) [RFC 1521]
- With RFC 2822 an email consists of
  - a simple "envelope"
    - created by the Message Transfer Agent based on the data in the email header
  - a set of header fields
    - each one line ASCII text
  - a blank line
  - the message (Message Body)

Email Header

Header	Meaning
To:	Address of the main receiver (possibly several receivers or also a mailing list)
Cc:	Carbon copy, email addresses of less important receivers
Bcc:	Blind carbon copy, a receiver which is not indicated to the other receivers
From:	Person who wrote the message
Sender:	Address of the actual sender of the message (possibly different to "From" person)
Received:	One entry per Message Transfer Agent on the path to the receiver
Return Path:	Path back to the sender (usually only email address of the sender)
Date:	Transmission date and time
Reply to:	Email address to which answers are to be addressed
Message-Id:	Clear identification number of the email (for later references)
In-Reply-to:	Message-Id of the message to which the answer is directed
References:	Other relevant Message-Ids
Subject:	One line to indicate the contents of the message (is presented the receiver)

Email Header

- RFC 2822: only suitable for messages of pure ASCII text without special characters. Today, there is demand for:
  - Email in languages with special characters, e.g. French, German, Turkish
  - Email in languages not using the Latin alphabet, e.g. Russian, Arabic
  - Email in languages not at all using an alphabet, e.g. Japanese, Chinese
  - Email not completely consisting of pure text, e.g. audio, video, image
- MIME keeps the RFC 2822 format, but additionally defines a structure in the message body (by using additional headers), and coding rules for non-ASCII characters.

Header	Meaning
MIME-Version:	Used version of MIME is marked
Content-Description:	String which describes the contents of the message
Content-Id:	Clear identifier for the contents
Content-Transfer-Encoding:	Coding which was selected for the contents of the email (some networks understand e.g. only ASCII characters). Examples: base64, quoted-printable
Content-Type:	Type/Subtype regarding RFC 1521, e.g., text/plain, image/jpeg, multi-part/mixed

Email Header

- Content-Type
  - Specifies the type of the body in the format: **type/subtype**

Type	Subtype	Description
Text	Plain	Unformatted text
	Enriched	Text including simple formatting commands
Image	Gif	Still picture in GIF format
	Jpeg	Still picture in JPEG format
Audio	Basic	Audible sound
Video	Mpeg	Movie in MPEG format
Application	Octet-stream	An uninterpreted byte sequence
	Postscript	A printable document in PostScript
Message	Rfc822	A MIME RFC 822 message
	Partial	Message has been split for transmission
	External-body	Message itself must be fetched over the net
Multipart	Mixed	Independent parts in the specified order
	Alternative	Same message in different formats
	Parallel	Parts must be viewed simultaneously
	Digest	Each part is a complete RFC 822 message

Email Header: Example

Microsoft Mail Internet Headers Version 2.0

Received: from mail.math.fu-berlin.de ([160.45.40.10]) by spree.pcpool.mi.fu-berlin.de with Microsoft SMTPSVC(6.0.3790.3959);

Thu, 24 Jan 2008 17:48:26 +0100

Received: (qmail 9044 invoked by alias); 24 Jan 2008 17:48:26 +0100

Delivered-To: schiller@inf.fu-berlin.de

Received: (qmail 9038 invoked from network); 24 Jan 2008 17:48:26 +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; 24 Jan 2008 17:48:26 +0100

Received: (qmail 8626 invoked by uid 9804); 24 Jan 2008 17:48:26 +0100

Received: from localhost (HELO mi.fu-berlin.de) (127.0.0.1)

by localhost with SMTP; 24 Jan 2008 17:48:06 +0100

Received: (qmail 23135 invoked by uid 9804); 24 Jan 2008 17:15:01 +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; 24 Jan 2008 17:15:01 +0100

Received: (qmail 152 invoked from network); 24 Jan 2008 17:15:01 +0100

Received: from sigma.informatik.hu-berlin.de (HELO mailslv1.informatik.hu-berlin.de) (141.20.20.51)

by leibniz.math.fu-berlin.de with (DHE-RSA-AES256-SHA encrypted) SMTP; 24 Jan 2008 16:15:01 -0000

Email Header: Example

Received: from ex.sar.informatik.hu-berlin.de (sar.informatik.hu-berlin.de [141.20.23.63])
by mailslv1.informatik.hu-berlin.de (8.13.8+Sun/8.13.8/INF-2.0-MA-SOLARIS-2.10-25) with ESMTP id m0OGEabt015579 for <schiller@inf.fu-berlin.de>; Thu, 24 Jan 2008 17:14:36 +0100 (CET)

X-Envelope-Sender: mm@informatik.hu-berlin.de

X-Virus-Scanned: by AMaViS 0.3.12pre7-L41+ClamAV[8175](NAI-uvscan@mi.fu-berlin.de)

X-Remote-IP: 141.20.20.51

Content-class: urn:content-classes:message

MIME-Version: 1.0

Content-Type: multipart/alternative;
boundary="----_=_NextPart_001_01C85EA4.35AB5B2E"

Subject: RE: Frohes neues Jahr

X-MimeOLE: Produced By Microsoft Exchange V6.5

Date: Thu, 24 Jan 2008 17:14:33 +0100

Message-ID: <BD8398D4C88E2C458083D1D2B04C4DA3207F4A@ex.sar.informatik.hu-berlin.de>

In-Reply-To: <6FE71171187F564EA019A177D00043B230418A@spree.pcpool.mi.fu-berlin.de>

X-MS-Has-Attach:

X-MS-TNEF-Correlator:

Thread-Topic: Frohes neues Jahr

Email Header: Example

Thread-Index: AchNIy4Op6zY/HruSXS/HroQsbGWmgBgaQBwApbvmuABYRONYAAGTKOgAAF20KA=

References: <6FE71171187F564EA019A177D00043B2304027@spree.pcpool.mi.fu-berlin.de>
<BD8398D4C88E2C458083D1D2B04C4DA3207E49@ex.sar.informatik.hu-berlin.de>
<6FE71171187F564EA019A177D00043B2304108@spree.pcpool.mi.fu-berlin.de>
<BD8398D4C88E2C458083D1D2B04C4DA3207F47@ex.sar.informatik.hu-berlin.de>
<6FE71171187F564EA019A177D00043B230418A@spree.pcpool.mi.fu-berlin.de>

From: "Max Mustermann" <mm@informatik.hu-berlin.de>

To: "Jochen Schiller" <schiller@inf.fu-berlin.de>

X-Greylist: Sender IP whitelisted, not delayed by milter-greylist-3.0 (mailslv1.informatik.hu-berlin.de [141.20.20.51]); Thu, 24 Jan 2008 17:14:36 +0100 (CET)

X-Virus-Status: No (sigma)

Return-Path: mm@informatik.hu-berlin.de

X-OriginalArrivalTime: 24 Jan 2008 16:48:26.0547 (UTC) FILETIME=[F0AD6030:01C85EA8]

------_=_NextPart_001_01C85EA4.35AB5B2E

Content-Type: text/plain; charset="iso-8859-1"

Content-Transfer-Encoding: quoted-printable

------_=_NextPart_001_01C85EA4.35AB5B2E

Content-Type: text/html; charset="iso-8859-1"

Content-Transfer-Encoding: quoted-printable

------_=_NextPart_001_01C85EA4.35AB5B2E--

MIME

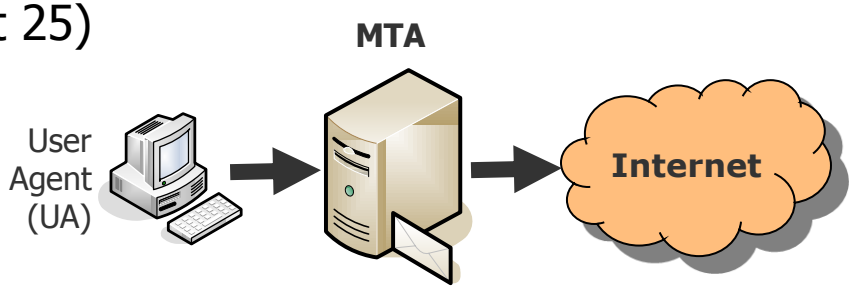
MIME-Version: 1.0
Content-Type: MULTIPART/MIXED;
BOUNDARY= "8323328-2120168431-824156555=:325"
--8323328-2120168431-824156555=:325
Content-Type: TEXT/PLAIN; charset=US-ASCII
A picture is in the appendix
--8323328-2120168431-824156555=:325
Content-Type: IMAGE/JPEG; name="picture.jpg"
Content-Transfer-Encoding: BASE64
Content-ID: <PINE.LNX.3.91.960212212235.325B@localhost>
Content-Description:
/9j/4AAQSkZJRgABAQEAlgCWAAD/2wBDAAEBAQEBAQEBAQEBAQEBAQIBAQEBAQIBAQECAgI
CAgICAgIDAwQDAwMDAwICAwQDAwQEBAQEAgMFBQQEBQQEBAT/2wBDAQEBAQEBAQI
BAQIEAwIDBAQEBA [...]
KKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKK
KKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAD//Z
---8323328-2120168431-824156555=:325 —

Electronic Mail (Email)

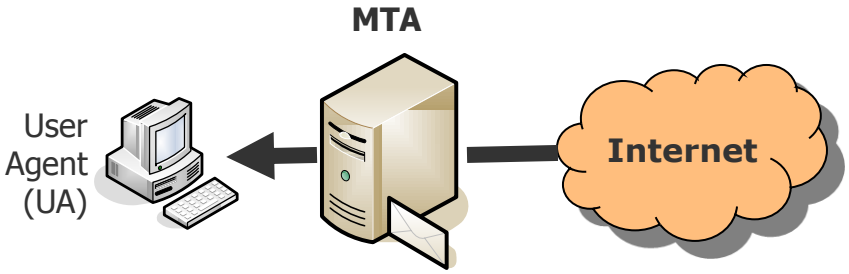
SMTP, POP3, and IMAP

Email over POP3 and SMTP

- Simple Mail Transfer Protocol (SMTP, RFC 5321)
  - Sending emails over a TCP connection (Port 25)
  - SMTP is a simple ASCII protocol
  - No checksums, no encryption

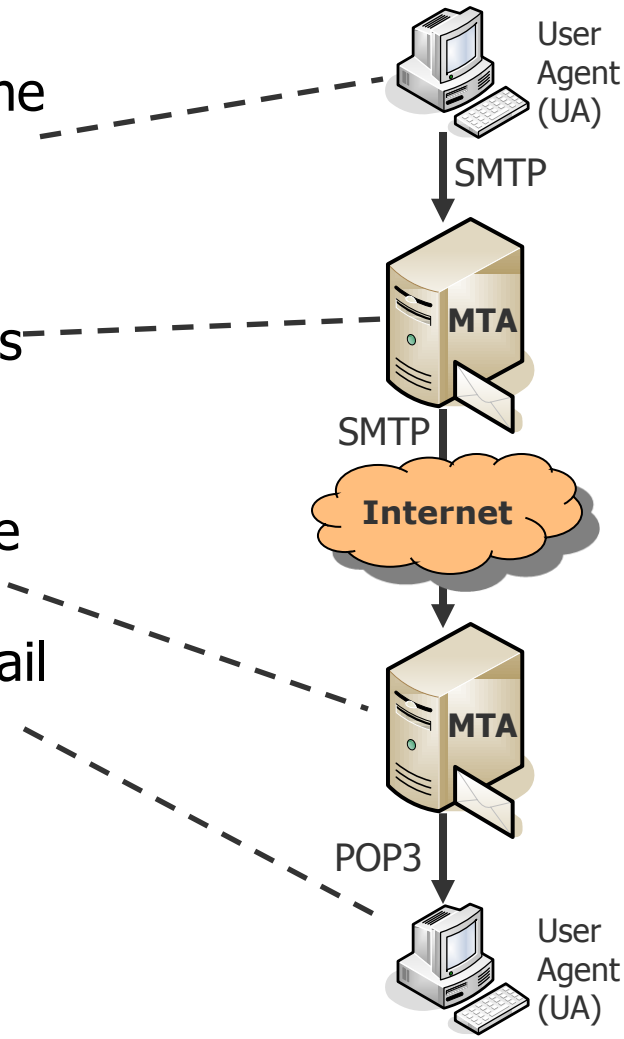


- Post Office Protocol version 3 (POP3, RFC 1939)
  - Get emails from the server over a TCP connection (Port 110)
  - Commands for
    - Logging in and out
    - Message download
    - Deleting messages on the server



Email over POP3 and SMTP

- User 1: writes an email
- Client 1 (UA 1): formats the email, produces the receiver list, and sends the email to its mail server (MTA 1)
- Server 1 (MTA 1): Sets up a connection to the SMTP server (MTA 2) of the receiver and sends a copy of the email
- Server (MTA 2): Produces the header of the email and places the email into the appropriate mailbox
- Client 2 (UA 2): sets up a connection to the mail server and authenticates itself with username and password (unencrypted!)
- Server (MTA 2): sends the email to the client
- Client 2 (UA 2): formats the email
- User 2: reads the email



SMTP: Command Sequence

Communication between partners (from abc.com to beta.edu) in text form of the following kind:

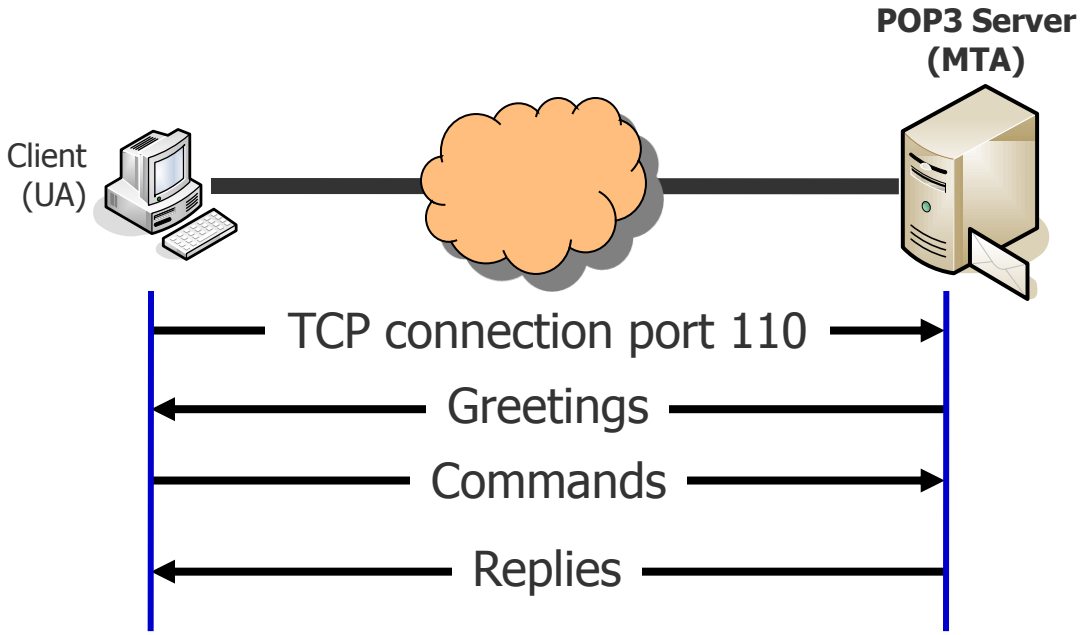
S: 220 <beta.edu> Service Ready	/* Receiver is ready/*
C: HELO <abc.com>	/* Identification of the sender/*
S: 250 <beta.edu> OK	/* Server announces itself */
C: MAIL FROM:<bob@abc.com>	/* Sender of the email */
S: 250 OK	/* Sending is permitted */
C: RCPT TO:<alice@beta.edu>	/* Receiver of the email */
S: 250 OK	/* Receiver known */
C: DATA	/* The data are following */
S: 354 Start mail inputs; end with "<crlf>.<crlf>" on a line by itself	
C: From: bob@…. <crlf>.<crlf>	/* Transfer of the whole email, including all headers */
S: 250 OK	
C: QUIT	/* Terminating the connection */
S: 221 <beta.edu> Server Closing	

S = server, receiving MTA / C = Client, sending MTA

POP3

- Authorization phase
  - USER name
  - PASS string
- Transaction phase
  - STAT
  - LIST [msg]
  - RETR msg
  - DELE msg
  - NOOP
  - RSET
  - QUIT
- Minimal protocol with only two command types:
  - Copy emails to the local computer
  - Delete emails from the server

Get emails from the server with POP3



POP3

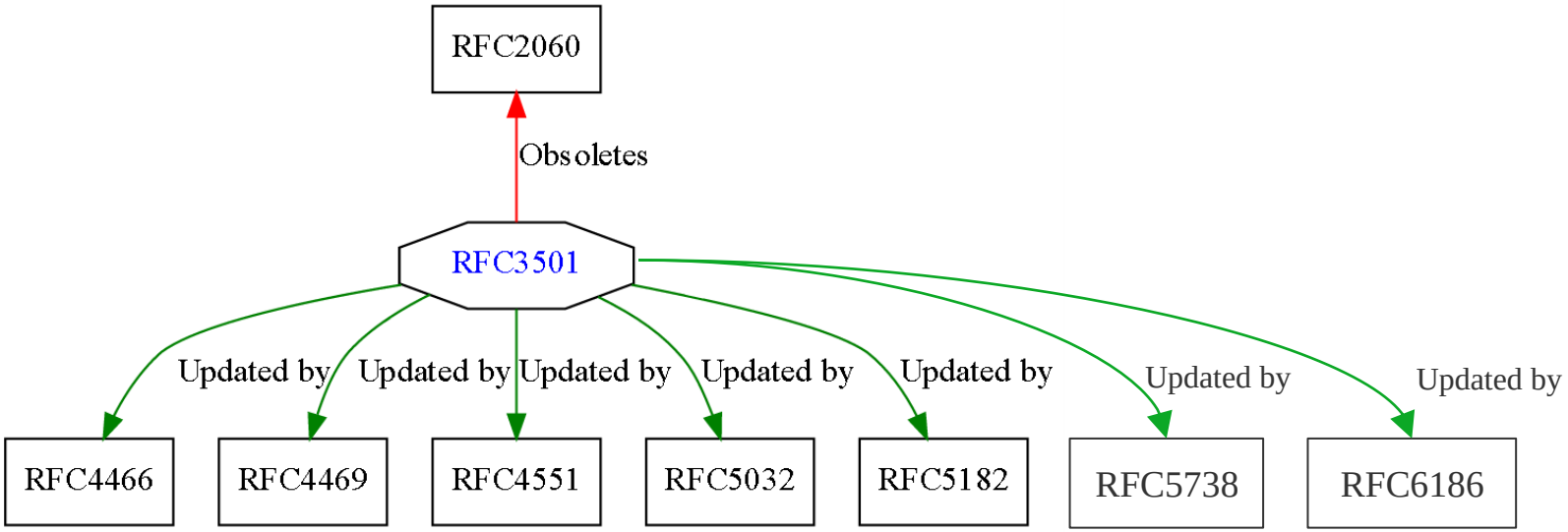
- Authorization phase
  - user identifies the user
  - pass is its password
  - +OK or -ERR are possible server answers
- Transaction phase
  - list for the listing of the message numbers and the message sizes
  - retr to requesting a message by its number
  - dele deletes the appropriate message

```
S: +OK POP3 server ready
C: user alice
S: +OK
C: pass hungry
S: +OK user successfully logged in
```

```
C: list
S: 1  498
S: 2  912
S: .
C: retr 1
S: <message 1 contents>
S: .
C: dele 1
C: retr 2
S: <message 2 contents>
S: .
C: dele 2
C: quit
S: +OK
```

IMAP as POP3 "Variant "

- Enhancement of POP3: Internet Message Access Protocol (IMAP4rev1, RFC 3501)



IMAP as POP3 "Variant "

- IMAP characteristics
  - TCP connection over port 143
  - Emails are not downloaded, but remain on the server
    - The client performs all actions remotely
    - The vast work is shifted to the server (search, change, delete, ...)
  - A user can access his emails from different hosts
    - Nomadic users
    - Provides multiple clients at the same
  - IMAP is more complicated than POP3
    - Set up and manage remote mailboxes
    - Download only header or parts of an email

- Meanwhile many web based email services exist: gmx, web.de, yahoo, google, ...
- In this case, HTTP serves as protocol for the access to the emails. The management is similar as with IMAP, only that the client is integrated into the web server.

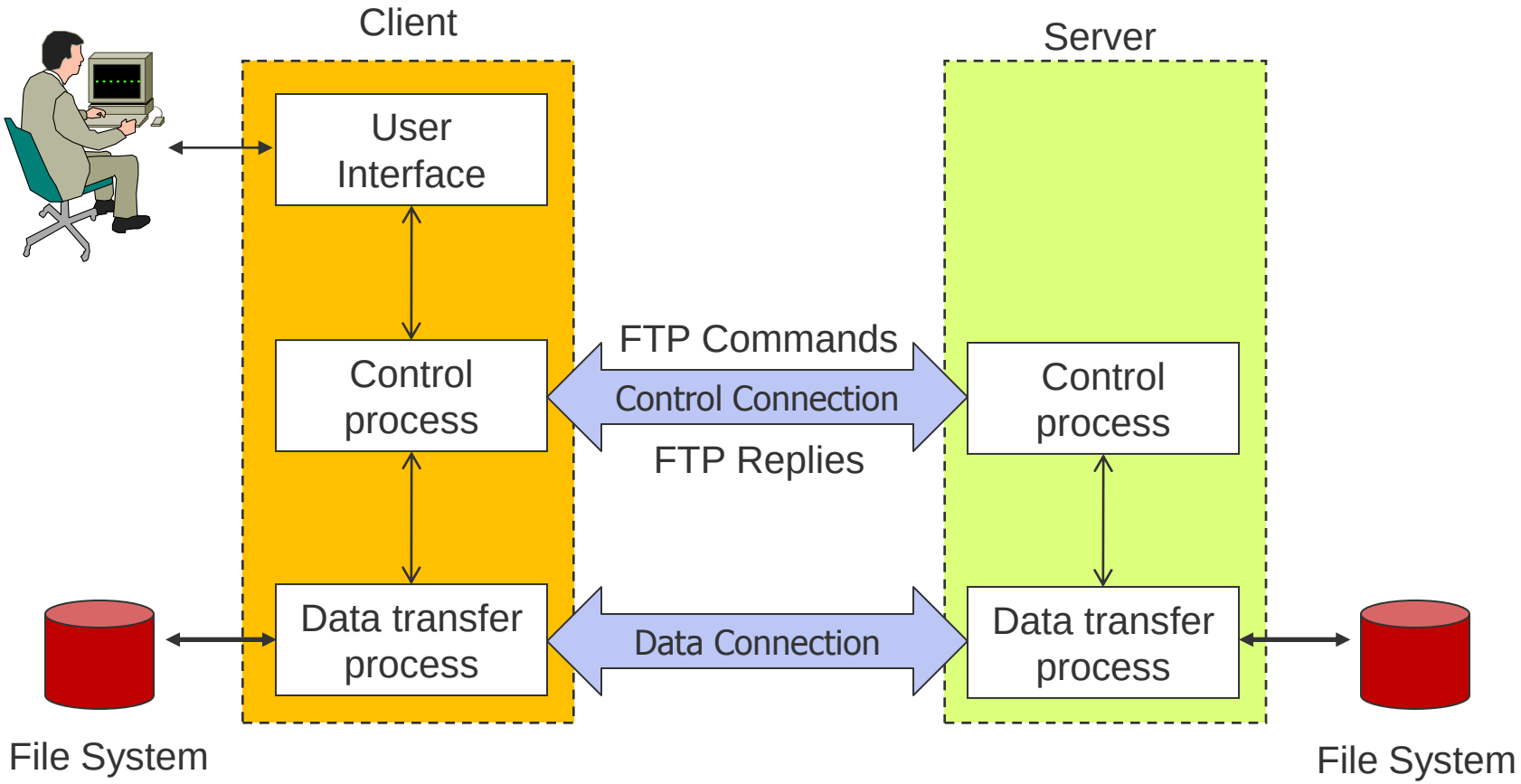
File Transfer Protocol (FTP)

File Transfer Protocol (FTP)

- Used for copying files between computers
  - FTP does not provide online file access like the Network File System (NFS)
  - FTP Instance on TCP port 21
  - ASCII commands for session control, e.g., GET, PUT, …
- FTP options
  - Data type: 7-Bit-ASCII, EBCDIC, Image/Binary (Bitstream), Local
  - File structures: File (byte stream), Record, Page
  - Transmission modes: Stream, Block, Compressed
- FTP services
  - Connection establishment with authentication (password)
  - File transmission, e.g., put, get, …
  - File operations, e.g., cd, dir, …
  - Help functions
  - …

File Transfer Protocol (FTP)

- Client-Server architecture
- Out-of-band control (Control connection, Data connection)



File Transfer Protocol (FTP)

- Problems/Criticisms with FTP
  - Password and file content are sent in clear text
  - Multiple TCP/IP connections are used
    - Problems when FTP is used behind NAT
  - Many FTP server provides anonymous ftp
    - Users do not need an account
  - See SCP for secure copy

World Wide Web (WWW)

Related Standards and RFCs

- Related RFCs
  - RFC 1945: Hypertext Transfer Protocol, HTTP/1.0
  - RFC 2616: Hypertext Transfer Protocol, HTTP/1.1
  - RFC 3986: Uniform Resource Identifier (URI): Generic Syntax
  - RFC 1738: Uniform Resource Locators (URL)
- Web Pages
  - World Wide Web Consortium (W3C): http://www.w3.org/
  - The Internet Archive: www.archive.org

Evolution of the WWW

- World Wide Web (WWW)
  - Access to linked documents, which are distributed over several computers in the Internet
    - Many people think of WWW as being the Internet!
    - Made the Internet popular to non academic people
    - Opened the way for using the Internet for commercial applications
    - Killer application for the networks at that time
- History of the WWW
  - Origin: 1989 in the nuclear research laboratory CERN in Switzerland.
  - Developed to exchange data, figures, etc. between a large number of geographically distributed project partners via Internet.
  - First text-based version in 1990.
  - First graphic interface (Mosaic) in February 1993, developed on to Netscape, Internet Explorer, …
  - Standardization by the WWW consortium (http://www.w3.org).

Communication in the WWW

- The Client/Server model is used

- Client (a Browser)
  - Presents the actually loaded WWW page (web page)
  - Permits navigating in the web, e.g., through clicking on a hyperlink
  - Offers a number of additional functions, e.g., external viewer, helper applications
  - Usually, a browser can also be used for other services, e.g., FTP, email, news, …
  - Popular browsers: Firefox, Internet Explorer, Chrome, Safari

- Server
  - Manages web pages
  - Is addressed by the client, e.g., through indication of an URL
    - Uniform Resource Locator (URL) = logical address of a web page
  - The server sends the requested page (or file) back to the client
  - Popular servers: Apache, Microsoft Internet Information Server

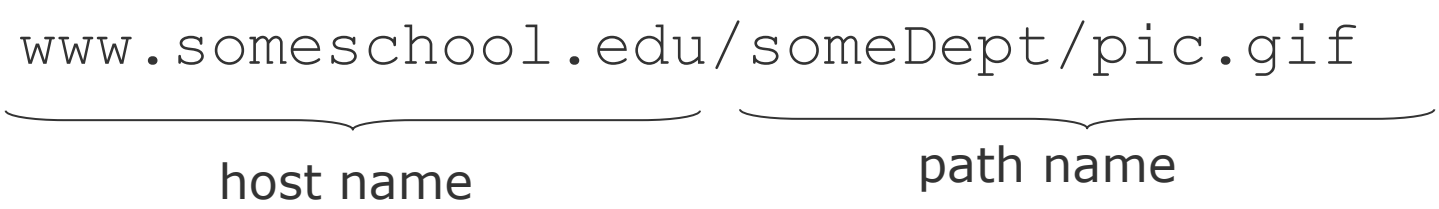
WWW, HTML, URL, and HTTP

- WWW stands for World Wide Web and means the world-wide cross-linking of information and documents
  - Also: The Web
- The standard protocol used between a web server and a web client is the HyperText Transfer Protocol (HTTP)
  - uses the TCP port 80
  - defines the allowed requests and responses
  - is an ASCII protocol
- Each web page is addressed by a unique **URL** (**Uniform Resource Locator**) , e.g. http://cst.mi.fu-berlin.de
- The standard language for web documents is the **HyperText Markup Language** (**HTML**)

WWW, HTML, URL, and HTTP

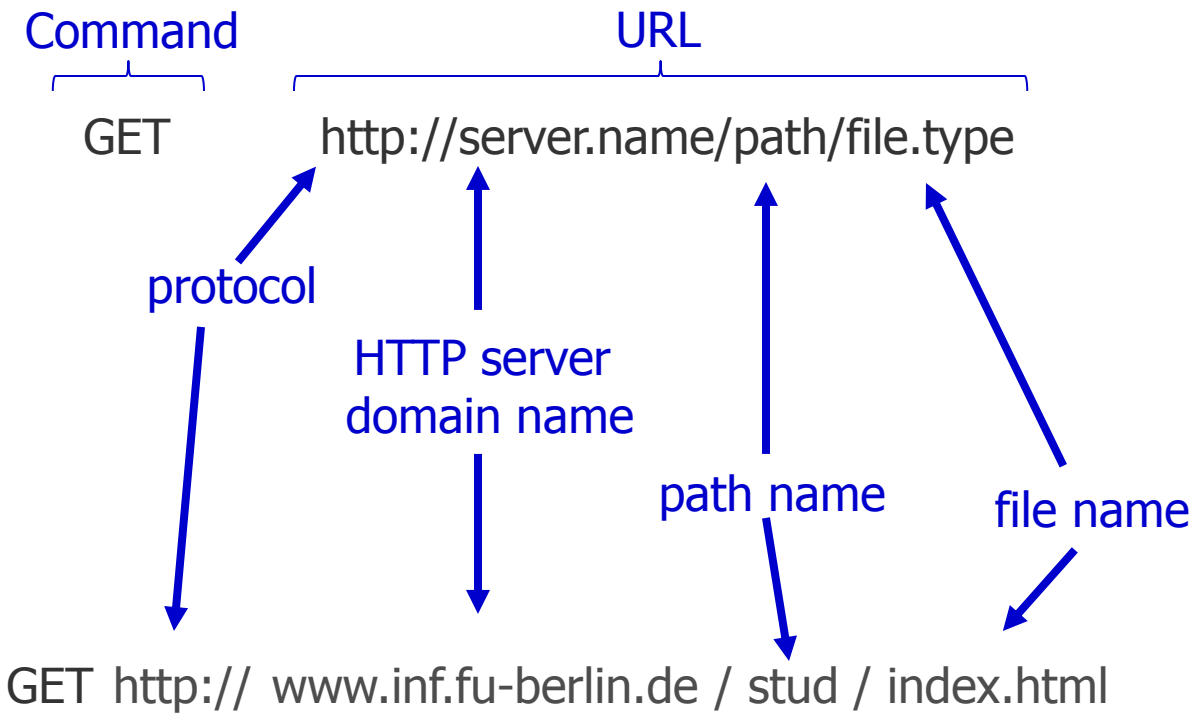
- First some jargon
  - **Web page** consists of **objects**
  - Object can be HTML file, JPEG image, Java applet, audio file, …
  - Web page consists of **base HTML-file** which includes several referenced objects
  - Each object is addressable by a **URL**

- Example URL:



World Wide Web (WWW)
HTTP

HTTP: Message Format



- Instructions on a URL are:
  - GET: Load a web page
  - HEAD: Load only the header of a web page
  - PUT: Store a web page on the server
  - POST: Append something to the request passed to the web server
  - DELETE: Delete a web page

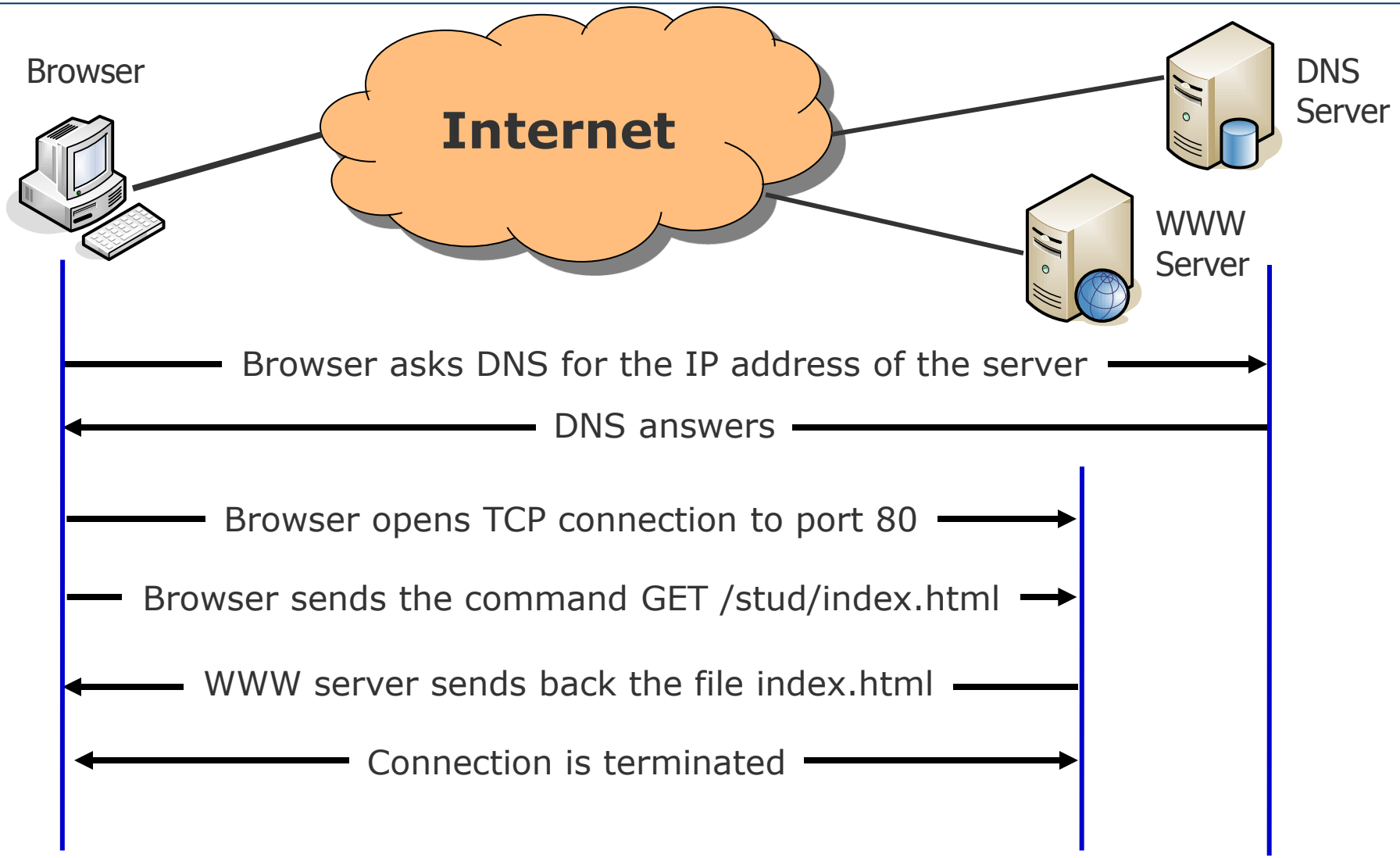
HTTP

- Uses TCP
  - Client initiates TCP connection (creates socket) to server
    - Port 80
  - Server accepts TCP connection from client
  - HTTP messages (application-layer protocol messages) exchanged between browser (HTTP client) and Web server (HTTP server)
  - TCP connection closed
- HTTP is "stateless"
  - Server maintains no information about past client requests
- Protocols that maintain "state" are complex!
  - Past history (state) must be maintained
  - If server/client crashes, their views of "state" may be inconsistent, must be reconciled
- And then there are cookies…
  - Preserve some state, history of browsing

HTTP Connections

- Nonpersistent HTTP
  - At most one object is sent over a TCP connection.
  - HTTP/1.0 uses nonpersistent connections
- Persistent HTTP
  - Multiple objects can be sent over single TCP connection between client and server.
  - HTTP/1.1 uses persistent connections in default mode

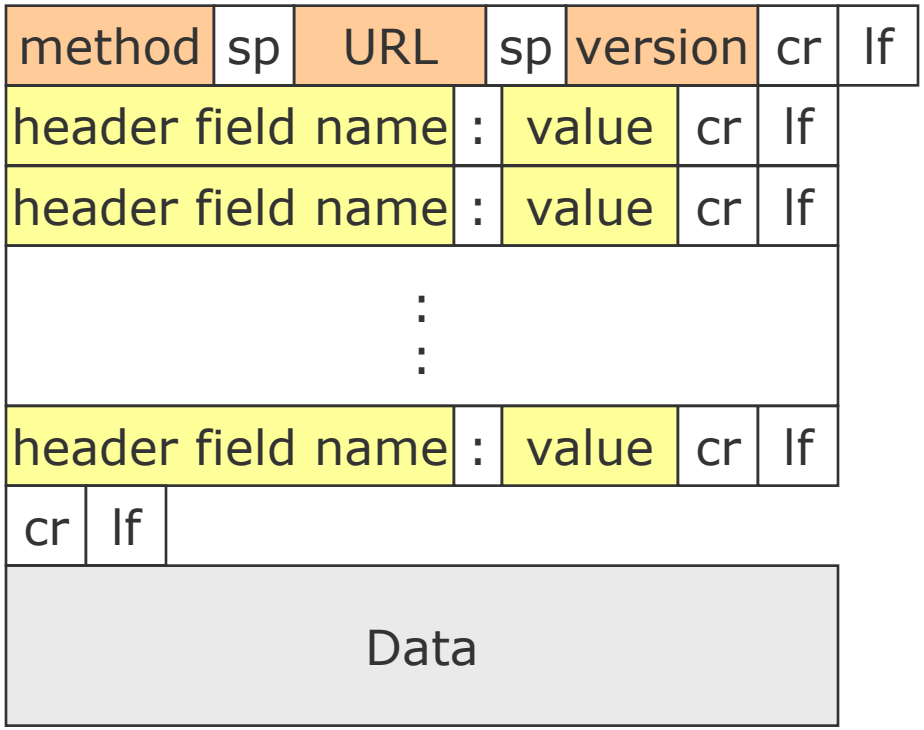
Loading of Web Pages



Loading of Web Pages

- Example: Call of the URL http://cst.mi.fu-berlin.de/teaching/WS0708/19540-V/index.html
  1. The Browser determines the URL (which was clicked or typed)
  2. The Browser asks the DNS for the IP address of the server cst.mi.fu-berlin.de
  3. DNS answers with 160.45.117.167
  4. The browser opens a TCP connection to port 80 of the computer 160.45.117.167
  5. Afterwards, the browser sends the command GET /teaching/WS0708/19540-V/index.html
  6. The WWW server sends back the file index.html
  7. The connection is terminated
  8. The browser analyzes the WWW page index.html and presents it
  9. If necessary, each picture is reloaded over a new connection to the server (The address is included in the page index.html in form of an URL)
- Note!
  - Step 9 applies only to HTTP/1.0! With the newer version HTTP/1.1 all referenced pictures are loaded before the connection termination (more efficiently for pages with many pictures)

HTTP Request Header



Request line: necessary part, e.g.,

GET server.name/path/file.type

Header lines: optionally, further information to the host/document, e.g.

Host: www.fu-berlin.de

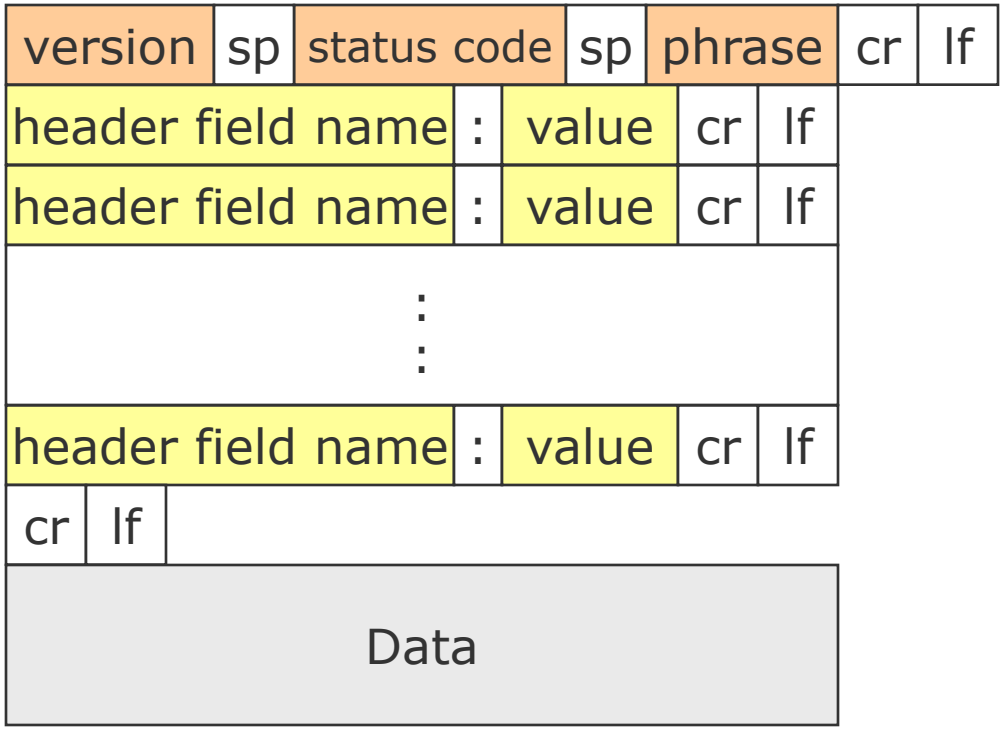
Accept-language: fr

User-agent: Opera /5.0

Entity Body: optionally. Further data, if the Client transmits data (POST method)

sp: space
cr/lf: carriage return/line feed

HTTP Response Header



Entity Body: inquired data

HEAD method: the server answers, but does not transmit the inquired data (debugging)

Status LINE: status code and phrase indicate the result of an inquiry and an associated message, e.g.

200 OK

400 Bad Request

404 Not Found

Groups of status messages:

1xx: Only for information
2xx: Successful inquiry
3xx: Further activities are necessary
4xx: Client error (syntax)
5xx: Server error

HTTP Response Header

status line
(protocol status code status phrase)

header lines

```
HTTP/1.1 200 OK
Connection close
Date: Thu, 06 Aug 1998 12:00:15 GMT
Server: Apache/1.3.0 (Unix)
Last-Modified: Mon, 22 Jun 1998 …...
Content-Length: 6821
Content-Type: text/html

data data data data data ...
```

data, e.g., requested HTML file

WWW 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

<DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html>
  <head>
    <title>FU-Berlin: Institut f&uuml;r 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();">
...
```

non persistent

Used in conditional GET, e.g. a cache can update a stored page depending on freshness.

Content of the web page

World Wide Web (WWW)

Cookies

User Identification: Cookies

- Problem with HTTP: stateless protocol
  - An HTTP session corresponds to one TCP connection. After the connection is terminated, the web server "forgets" everything about the request.
  - Simple principle, enough for browsing
  - But, not suitable for web-applications like online shops.
    - Require the storing of information about sessions, e.g., userID, selected items, …
  - Solution: new header field Set-cookie
    - Instructs client to store the received cookie together with the server name and fill it in its own header in following requests to that server.
    - Thus the server is able to identify related requests

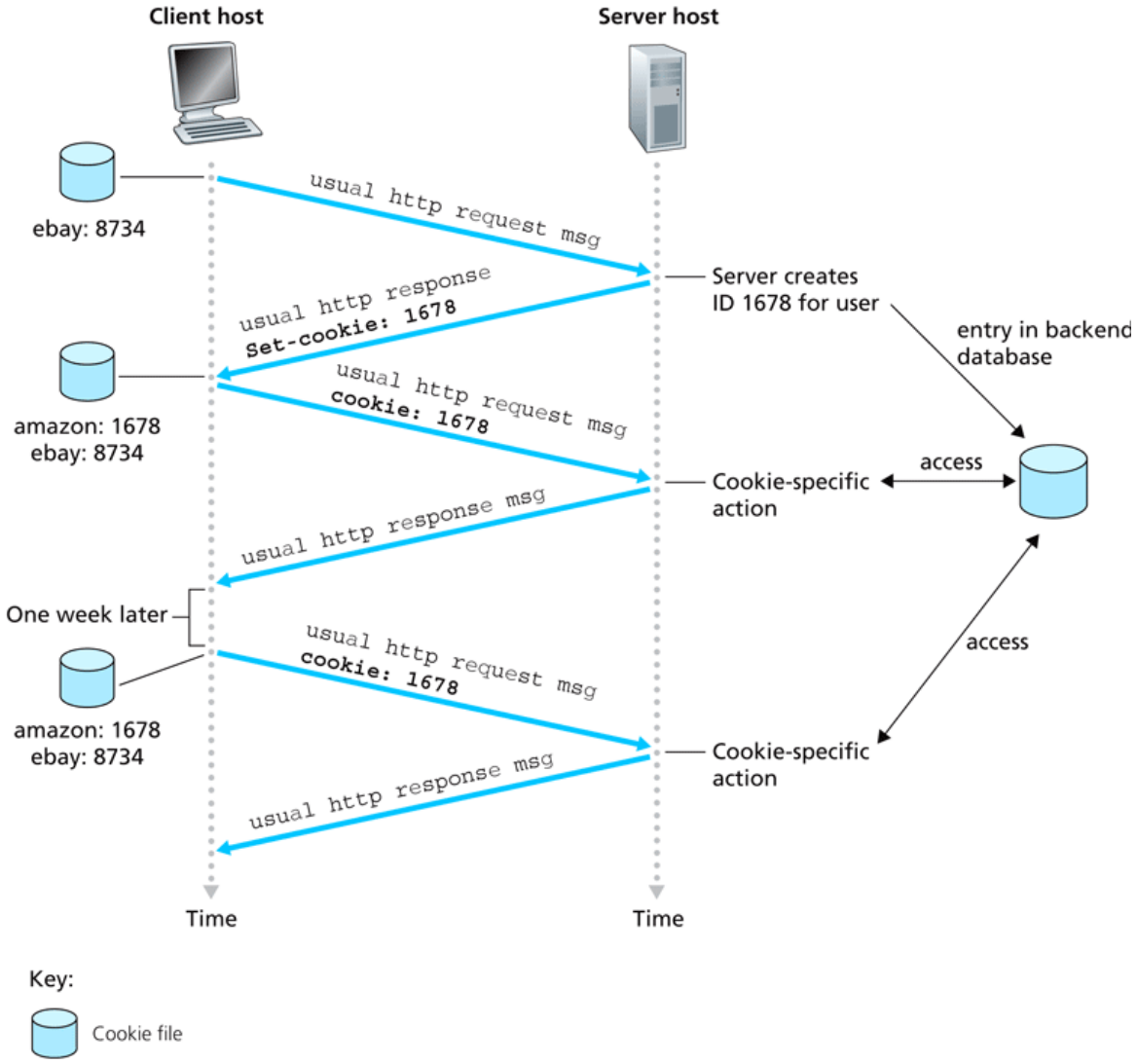
User Identification: Cookies

- Cookie content:
  - Name-value pair defined by the server for identification
  - Optional name-value pairs for, e.g., comments, date, TTL

Domain	Path	Content	Expires	Secure
toms-casino.com	/	CustomerID=497793521	15-10-08 17:00	Yes
joes-store.com	/	Cart=1-00501;1-07031	11-10-08 12:00	No
aportal.com	/	Prefs=Stk:SUNW+ORCL	31-12-08 17:30	No
sneaky.com	/	UserID=2344537333744	31-12-08 18:00	No

- Cookie is valid until expiration time
- A server can delete cookie by resending the cookie with an expiration time in the past
- Client sends cookie together with the request to the server

User Identification: Cookies



User Identification: Cookies

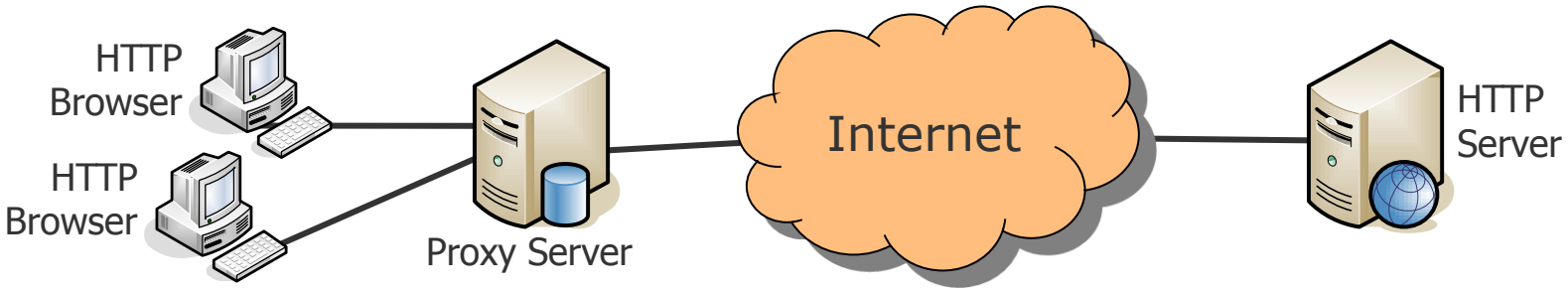
- What cookies can bring
  - Authorization
  - Shopping carts
  - Recommendations
  - User session state (Web email)
- How to keep "state"
  - Protocol endpoints: maintain state at sender/receiver over multiple transactions
  - Cookies: http messages carry state
- Cookies and privacy
  - Cookies permit sites to learn a lot about you
  - You may supply name and email to sites

World Wide Web (WWW)

Proxies

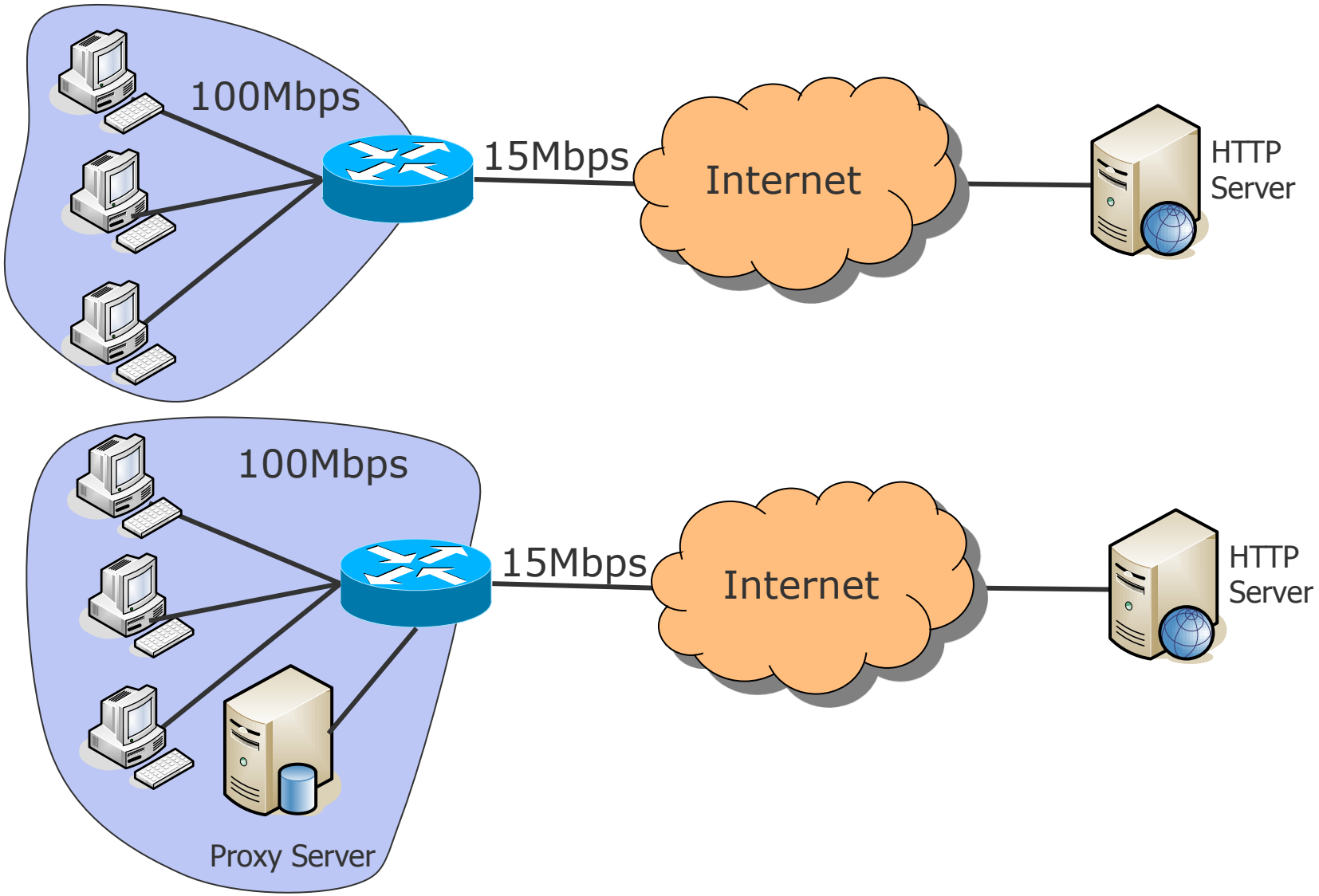
Proxy Server

- A Proxy is an intermediate entity used by several browsers. It takes over tasks of the browsers (complexity) and servers for more efficient page loading!



- Caching of web pages
  - A proxy temporarily stores the pages loaded by browsers. If a page is requested by a browser which already is in the cache, the proxy controls whether the page has changed since storing it. If not, the page can be passed back from the cache. If yes, the page is normally loaded from the server and again stored in the cache, replacing the old version.
- Support when using additional protocols
  - A browser enables also access to FTP, News, Gopher, or Telnet servers, etc.
  - Instead of implementing all protocols in the browser, it can be realized by the proxy. The proxy then "speaks" HTTP with the browser and e.g. FTP with a FTP server.
- Integration into a Firewall
  - The proxy can deny the access to certain web pages (e.g. in schools).

Proxy Server: Performance Gain



Proxy Server: Performance Gain

- Scenario
  - LAN works with 100 Mbps Ethernet
  - Internet connection with 15 Mbps
  - Traffic characteristic
    - 15 requests per sec
    - Every request of 1 Mbit (~ 125000 byte)
    - Internet delay ~2 sec
    - LAN delay ~0.01 sec
  - With proxy server
    - Hit rate of 0.4

- Performance considerations
  - Load on the LAN

$$\frac{15\frac{\text{req}}{s}\times 1\,\text{Mbit}}{100\,\text{Mbps}} = 0.15$$

  - Load to the Internet

$$\frac{15\frac{\text{req}}{s}\times 1\,\text{Mbit}}{15\,\text{Mbps}} = 1.0$$

  - Delay without proxy ~ 2 sec
  - Delay with proxy
    0.4 x 0.01 s + 0.6 x 2 s ~ 1.2 s

World Wide Web (WWW)

HTML

Hyper Text Markup Language (HTML)

- HyperText Markup Language (HTML)
  - HTML documents are **structured** text documents
  - HTML commands (**tags**) define the presentation of the document (see LaTeX)
    - HTML tags are contained in the document, e.g., Bold Font
  - Documents consists of a **header** and a **body**
    - Header defines general properties of the document
    - Body contains the content, e.g., text, images, tables, lists, …
  - Text parts and other documents can be referenced by **hyperlinks**
  - Advantage: Presentation can be localized based on the client
- Standardization currently at HTML 4.0
  - Integration of script languages and Cascading Style Sheets (CSS)
- W3C currently works on HTML 5.0
  - For more information see: http://www.w3.org/html/wg/html5/

HTML Example

- Basic components
  - <...>: Start Tag
  - </...>: End Tag
  - Various structuring components
    - paragraph
    -
new line
    - <h1>Title of level 1</h1>
    - <h2>Title of level 2</h2>
  - Various character forms
    - bold
    - italic
    - fett
  - Standard font encoding ISO 8859-1 (8-Bit, ASCII as subset)
    - HTML 3 introduced other encoding formats, e.g., 16-bit unicode
  - Referencing to other documents
    - Name

```
<html>
  <head>
    <title> Document Title </title>
  </head>
  <body>
   <p>This is an simple example.</p>
  </body>
</html>
```

XML and XSL

- Disadvantages of HTML
  - No structuring of web pages
  - Mixing of content with format instructions
    - Example: In a web page containing products and prices it is difficult to distinguish descriptions and prices for automatic processing.

- Demand for web pages which can be automatically processed
  - XML (eXtensible Markup Language)
    - Structured description of content
  - XSL (eXtensible Style Language)
    - Description of format independently

XML and XSL

A simple Web page in XML.

```
<?xml version="1.0" ?>
<?xml-stylesheet type="text/xsl" href="b5.xsl"?>

<book_list>

<book>
   <title> Computer Networks, 4/e </title>
   <author> Andrew S. Tanenbaum </author>
   <year> 2003 </year>
</book>

<book>
   <title> Modern Operating Systems, 2/e </title>
   <author> Andrew S. Tanenbaum </author>
   <year> 2001 </year>
</book>

<book>
   <title> Structured Computer Organization, 4/e </title>
   <author> Andrew S. Tanenbaum </author>
   <year> 1999 </year>
</book>

</book_list>
```

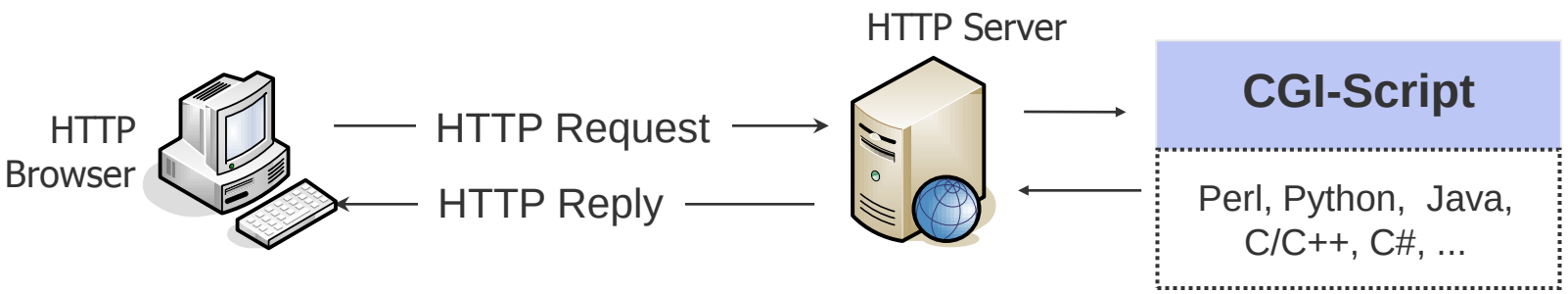
A style sheet in XSL.

```
<?xml version='1.0'?>
<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform" version="1.0">
<xsl:template match="/">

<html>
<body>

<table border="2">
  <tr>
     <th> Title</th>
     <th> Author</th>
     <th> Year </th>
  </tr>

  <xsl:for-each select="book_list/book">
  <tr>
     <td> <xsl:value-of select="title"/> </td>
     <td> <xsl:value-of select="author"/> </td>
     <td> <xsl:value-of select="year"/> </td>
 </tr>
 </xsl:for-each>
</table>

</body>
</html>
</xsl:template>
</xsl:stylesheet>
```

A loop processing all entries in a book_list

Dynamic Documents with SSI and CGI

- HTML provides only static web pages
- Server Side Includes (SSI)
  - Simple way of generating dynamic documents
  - Principle: Embedding of server-directives into the HTML-Document
  - Server substitutes the directives with dynamic generated content, e.g., content of a file, output of a program. Often PHP is used.
  - Example: <!-- #include file="lastupdate.txt" -->
- Common Gateway Interface (CGI)
  - External program (CGI-Script) is called on the WWW-server, e.g., by input of an URL
  - Output of the CGI-Script (HTML-Code!) is sent to the client

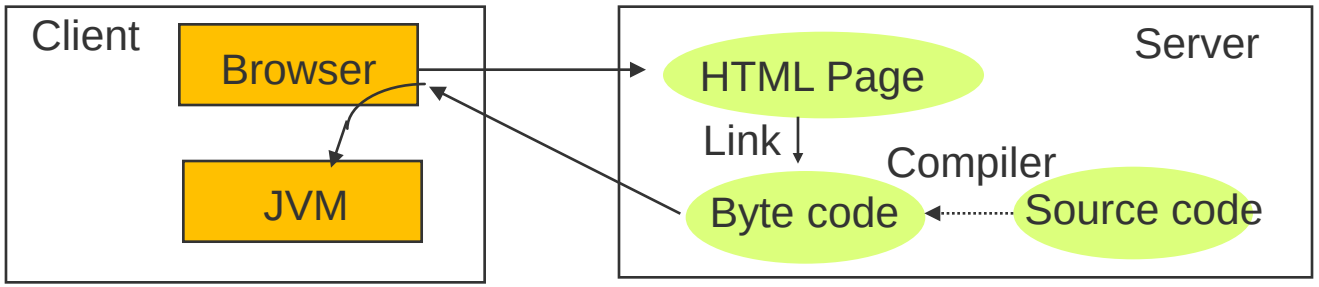


CGI-Scripts

- The WWW server has a special directory for CGI scripts
  - CGI script is started when the HTTP client demands for it
  - Very often PERL (Practical Extraction and Report Language) or Python is used
    - interpreted language, particularly suitable for string processing
- HTTP-Query ➡ CGI-Script
  - Query fields are stored in environment variables
    - e.g. HTTP_USER_AGENT (Version, Type of the browser)
    - e.g. QUERY-STRING (URL of the query)
      - e.g. www.xyz.abc.de/cgi-bin/testscript?parameter1?
  - The CGI script gets the data of a POST query over the standard input device
- CGI-Script ➡ HTTP-Response
  - The CGI script provides over the standard output either
    - a complete response with all header fields, or
    - only the raw data, i.e. the header fields are generated by the server

Java-Applets

- Java was developed by Sun Microsystems
- Originally for consumer/entertainment devices
- Java is platform independent (Java byte code)
- Java Virtual Machine (JVM) interprets Java byte code
- Integration into web pages
  - Introduction of a new HTML tag <APPLET>
  - Example: <APPLET> CODE=game.class WIDTH=100 HEIGHT=200> </APPLET>
- Applet runs on the client computer



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

The Memory of the Web – The Internet Archive

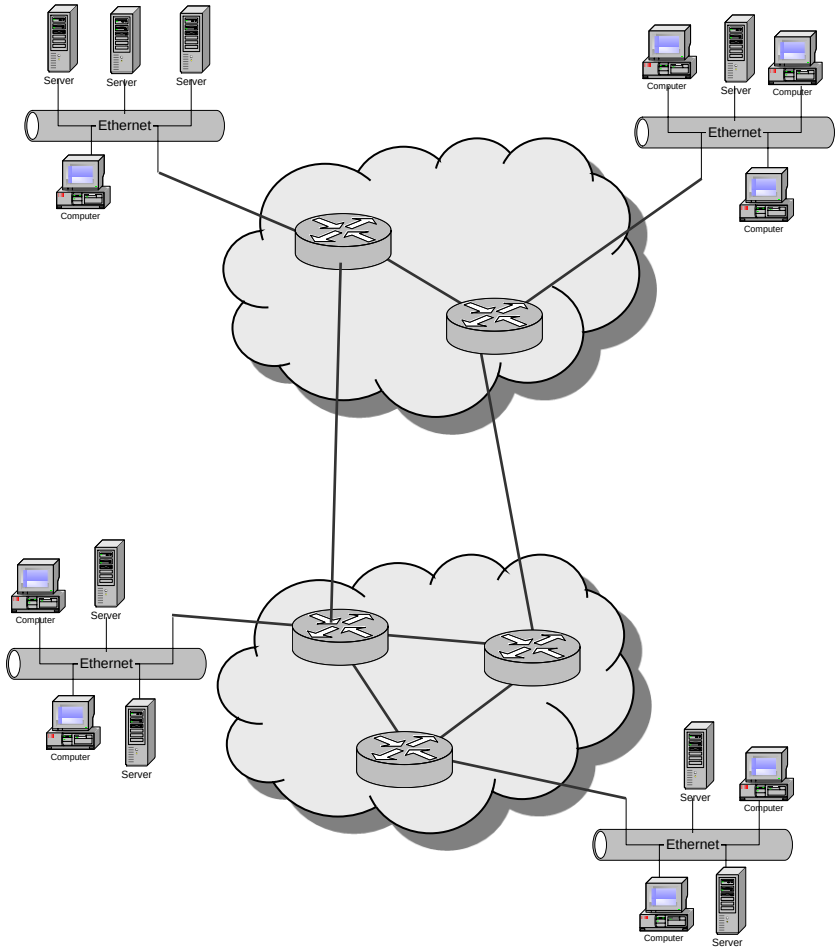
- The Internet Archive: www.archive.org
  - A non-profit organization dedicated to maintaining an on-line library and archive of web and multimedia resources
  - Founded in 1996
  - According to its website:

    "Most societies place importance on preserving artifacts of their culture and heritage. Without such artifacts, civilization has no memory and no mechanism to learn from its successes and failures. Our culture now produces more and more artifacts in digital form. The Archive's mission is to help preserve those artifacts and create an Internet library for researchers, historians, and scholars. The Archive collaborates with institutions including the Library of Congress and the Smithsonian."
  - The Wayback Machine
    - Snapshot of web pages
  - Security, social, and political aspects of the Internet!

Simple Network Management Protocol

What is Network Management?

- Goals of Network Management
  - Monitoring of network equipment
  - Efficient networking
  - Internetworking
  - Accounting of network usage
  - Protected and safe networking
  - Simple modeling of the network (status)
  - Gathering of data for network planning
  - Planning (construction) of manageable networks



Functional Areas of Network Management

- According to the ISO Network Management Model
  - Performance Management
    - Measure and provide various aspects of **network performance** so that internetwork performance can be maintained at an acceptable level.
    - Examples of performance variables include network **throughput**, user **response times**, and **line utilization**.
  - Configuration Management
    - Monitor the **network** and **system configuration** information so that the effects on network operation of various versions of hardware and software elements can be tracked and managed.
  - Accounting Management
    - Measure network **utilization** parameters so that individual or group uses on the network can be regulated appropriately. Such regulation minimizes network problems and maximizes the fairness of network access across all users.
  - Fault Management
    - **Detect**, **log**, **notify** users of, and **automatically fix** network **problems** to keep the network running effectively. Because faults can cause downtime or unacceptable network degradation, fault management is perhaps the most widely implemented of the ISO network management elements.
  - Security Management
    - **Control access** to network resources according to local guidelines so that the network cannot be sabotaged (intentionally or unintentionally) and sensitive information cannot be **accessed** by those without appropriate **authorization**.

Functional Areas of Network Management

- Additional Management issues
  - Asset Management
    - Collect statistics of equipment, facility, and administration personnel
  - Planning Management
    - Provide analysis of trends to help justify a network upgrade or bandwidth increase

SNMP & Network Management History

Date	Event
1983	• TCP/IP replaces ARPANET at U.S. Dept. of Defense, effective birth of Internet • First model for network management • High-Level Entity Management System (HEMS), RFCs 1021,1022,1024,1076

1987	• ISO/OSI proposes
	  • Common Management Information Protocol (CMIP), and
	  • CMOT (CMIP over TCP) for the actual network management protocol for use on the Internet
	• Simple Gateway Monitoring Protocol (SGMP), RFC 1028

1989	• Setup of the SNMP working group to create a common network management framework to be used by both SGMP and CMOT to allow for transition to CMOT • "Internet-standard Network Management Framework" defined (RFCs 1065, 1066, 1067) • SNMP promoted to recommended status as the de facto TCP/IP network management framework (RFC 1098) • IAB committee decides to let SNMP and CMOT develop separately
1990	• IAB promotes SNMP to a standard protocol with a recommended status (RFC 1157)
1991	• Format of MIBs and traps defined (RFCs 1212, 1215) • TCP/IP MIB definition revised to create SNMPv1 (RFC 1213)
1996	• Introduction of community-based SNMPv2 (RFC1901)
1999	• Introduction of SNMPv3 (RFC2570, 3410)

What is SNMP?

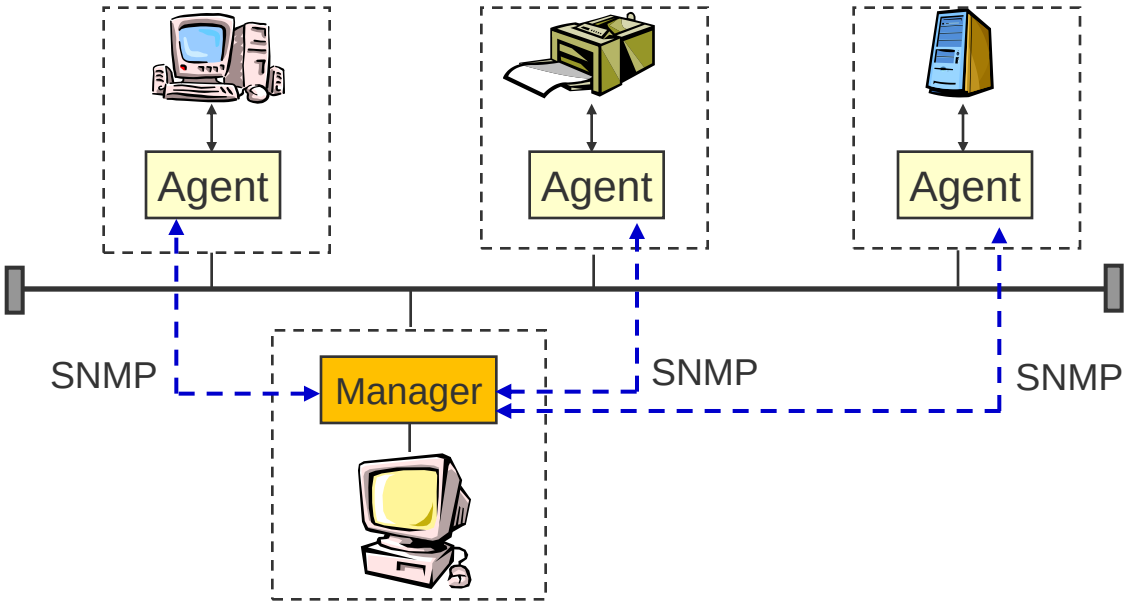
- SNMP allows remote and local management of devices on the network including servers, workstations, routers, switches, …
- SNMP is comprised of **agents** and **manager**s
  - Agent: process running on each **managed node** collecting information about the device it is running on.
  - Manager: process running on a **management workstation** that requests information about devices on the network.
- Two major versions SNMPv1 and SNMPv2
  - SNMP was designed originally as an interim solution, but it is now de-facto standard in LANs
  - SNMPv1 is the recommended standard
  - SNMPv2 has become split into:
    - SNMPv2u - SNMPv2 with user-based security
    - SNMPv2* - SNMPv2 with user-based security and additional features
    - SNMPv2c - SNMPv2 without security
  - Current version is SNMPv3 according to RFC 3411-RFC 3418

The Three Parts of SNMP

- SNMP network management is based on three parts:
  - SNMP Protocol
    - Defines format of messages exchanged by management systems and agents.
    - Specifies the Get, GetNext, Set, and Trap operations
    - Is based on UDP ➡ connectionless
  - Structure of Management Information (SMI)
    - Rules specifying the format used to define objects managed on the network that the SNMP protocol accesses
  - Management Information Base (MIB)
    - A map of the hierarchical order of all managed objects and how they are accessed

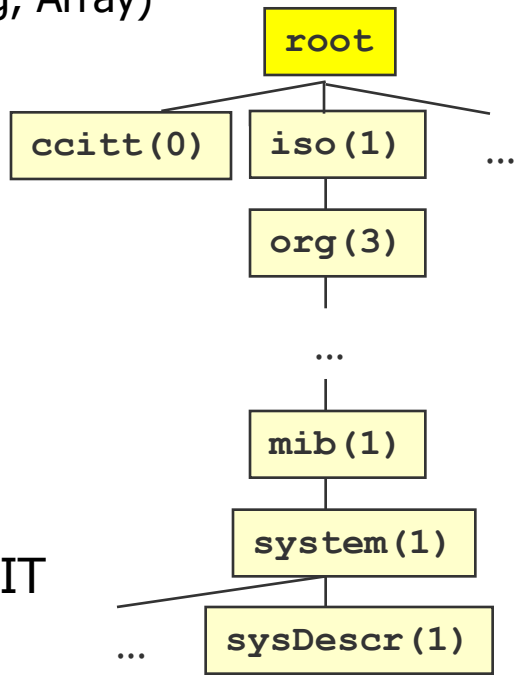
Simple Network Management Protocol (SNMP)

- SNMP service for the management of network resources
  - e.g. printers, bridges, router, hosts, etc.
- Managed resources have an integrated SNMP agent (Software)
  - The agents maintain the management information of the resource
    - e.g. the number of received/dropped/send packets
- The manager (Software) is responsible for the communication with the agents
  - Protocol: SNMP (based on UDP)
- Basis of the communication between manager and agent: Management-Objects



Managed Objects

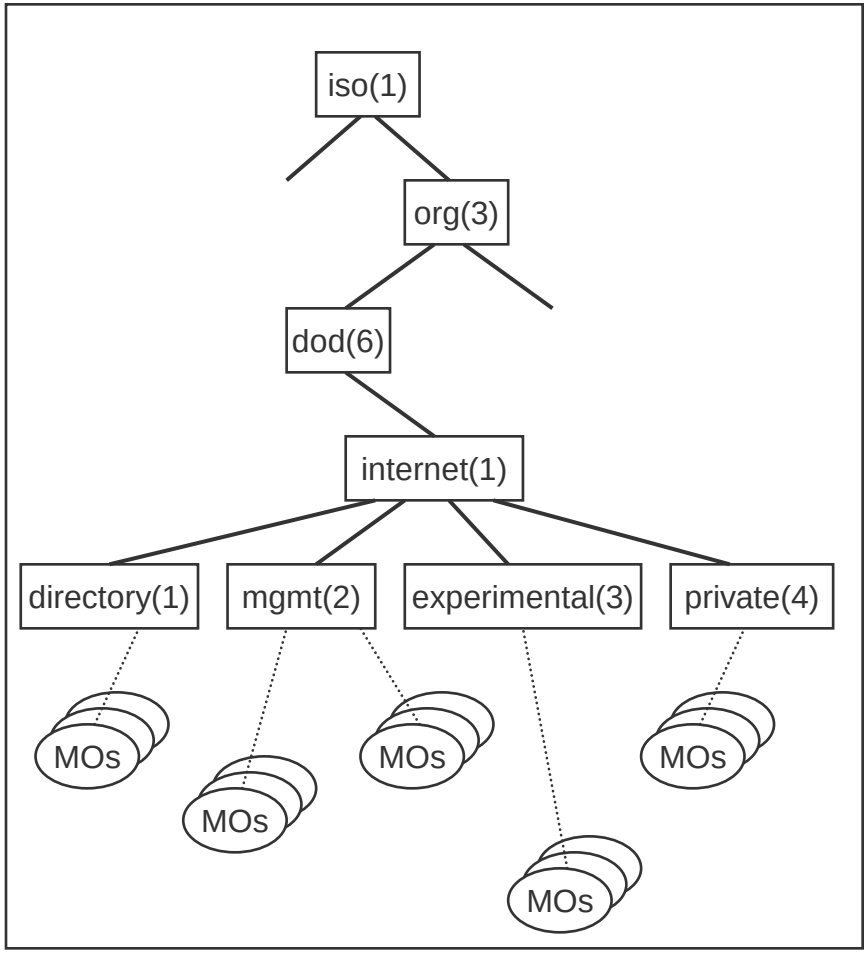
- Managed Object
  - Model of several properties of a network resource
  - An agent maintains the managed objects of "its" resources
  - Components of a managed object in the Internet:
    - Unique name: **iso.org.dod.internet.mgmt.mib.system.sysDescr**
    - Syntax: Simple data types (Integer, String, Array)
    - Access rights: read-only, read-write
    - Status: mandatory, optional



- Management Information Base (MIB)
  - Collection of all managed objects
  - Distributed virtual data base
- Management Information Tree (MIT)
  - Each managed object has a unique position in the MIT
  - Thus provides unique reference

Modeling of Management Information: MIB und SMI

Management Information Base (MIB)



Structure of Management Information (SMI)

sysDescr OBJECT-TYPE

SYNTAX DisplayString (SIZE (0..255))

ACCESS read-only

STATUS mandatory

DESCRIPTION

"A textual description of the entity. This value should include the full name and version identification of the system's hardware type, software operating-system, and networking software. It is mandatory that this only contains printable ASCII characters."

::= { system 1 }

Modeling of Management Information: MIB und SMI

- Freie Universität Berlin
  - iso(1).identified-organization(3).dod(6).internet(1).private(4).enterprise(1).18898
  - 1.3.6.1.4.1.18898
  - Used in the DES-Testbed
- OID Repository
  - Check which OIDs are available
  - http://www.oid-info.com

RMON: Remote Monitoring

- MIB with special Functions (RFC 1757)
  - Gathering of statistics, alarms, events
  - Evaluations (partially), filtering, Packet-Capture, ...

➡ Shifting of "Intelligence" from the management platform to the agent

- Can work on top of TCP/IP since version 2 and log appropriate parameters

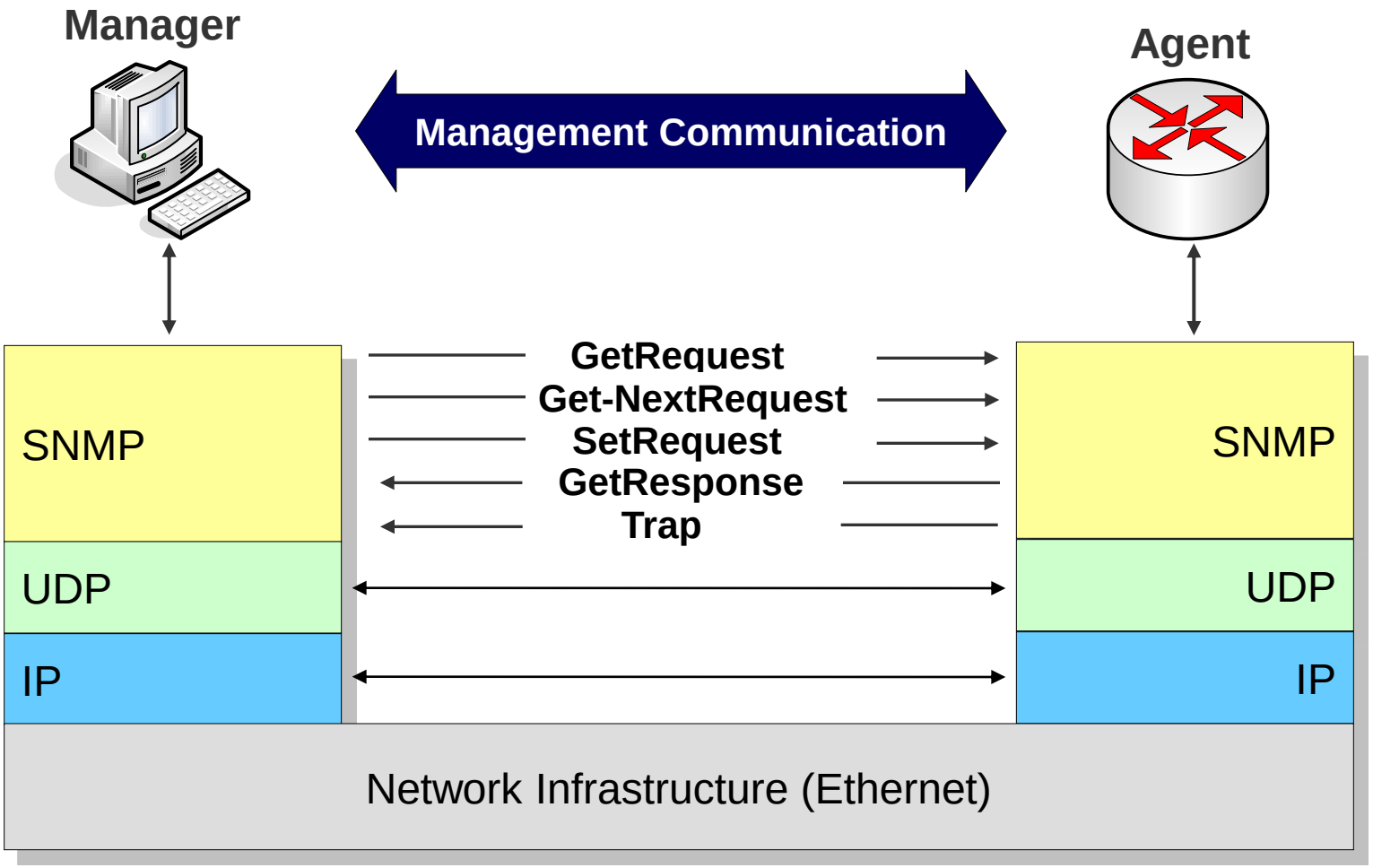
- Example SMI:

```
etherStatsOversizePkts OBJECT-TYPE
  SYNTAX Counter
  ACCESS read-only
  STATUS mandatory
  DESCRIPTION
    "The total number of packets received that were longer than 1518
    octets (excluding framing bits, but including FCS octets) and were
    otherwise well formed."
  ::= {etherStatsEntry 10}
```

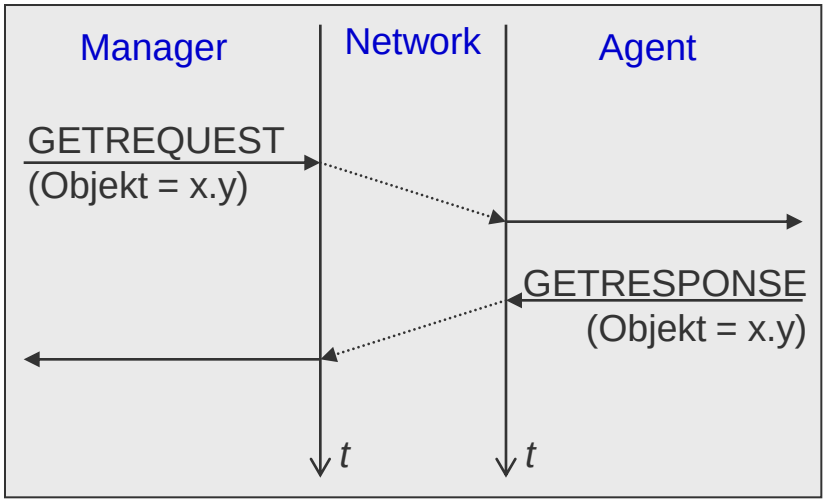
Simple Network Management Protocol

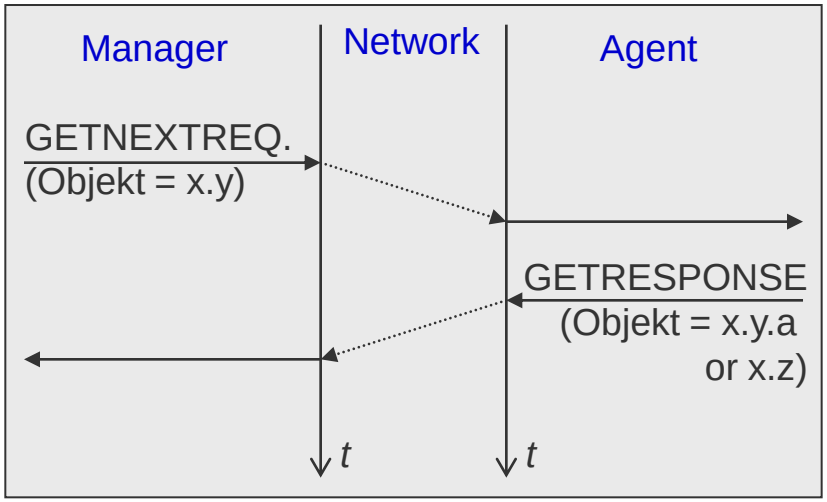
SNMP Operations

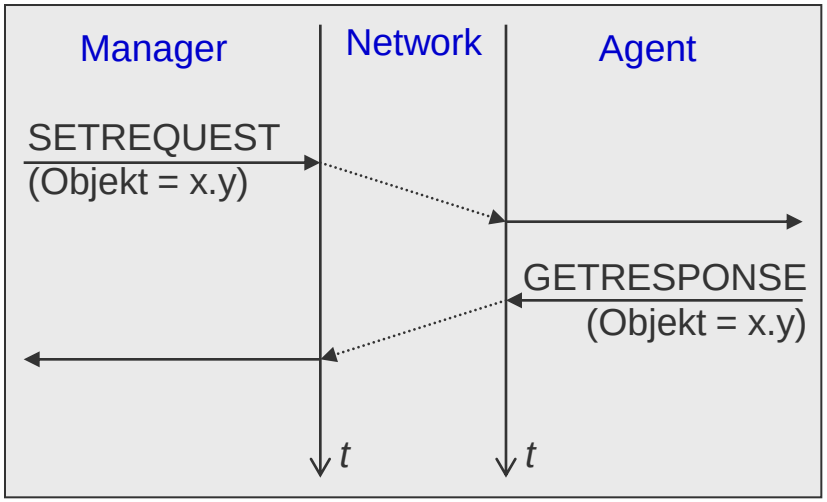
SNMP Operations

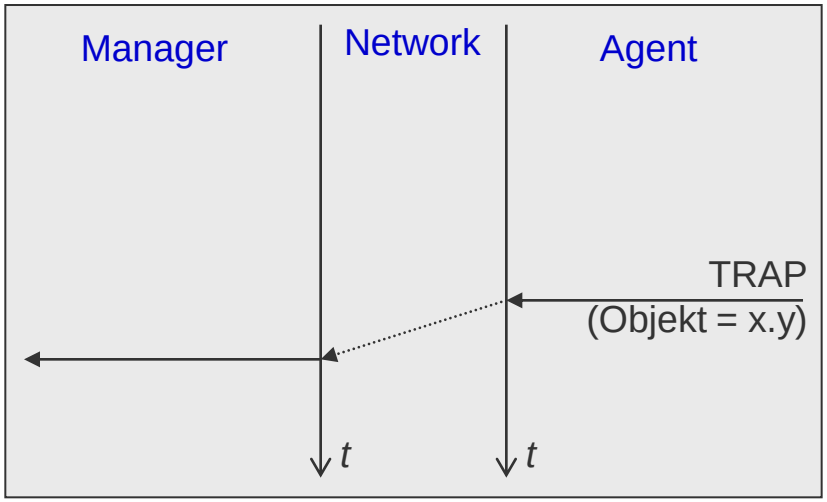


SNMP: Client-Server Principle



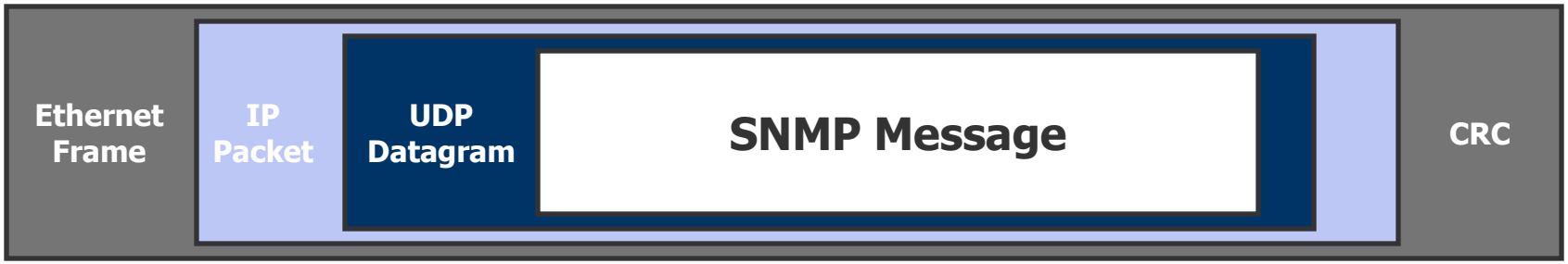






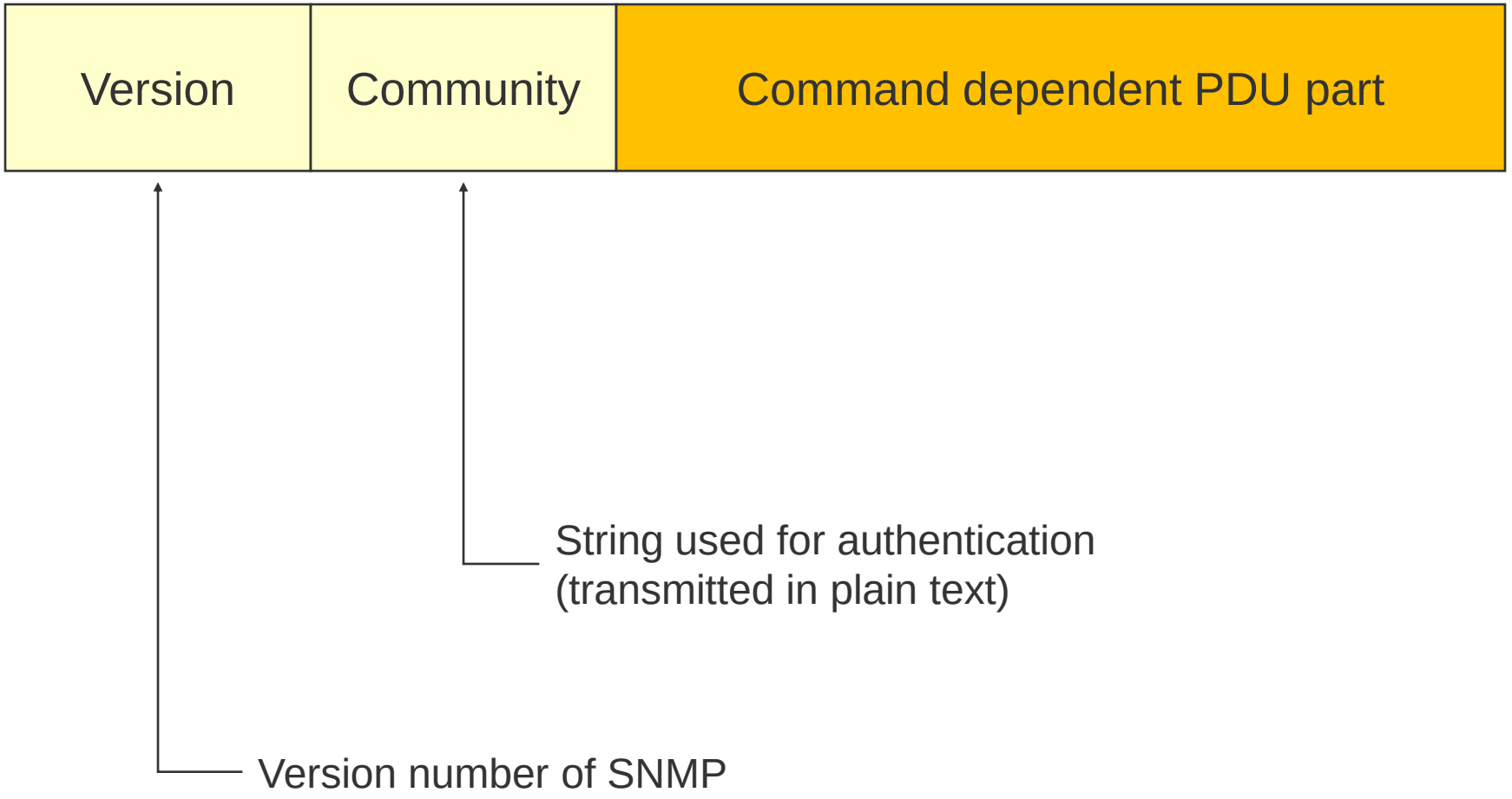
Ports & UDP

- SNMP uses UDP as the transport mechanism for SNMP messages



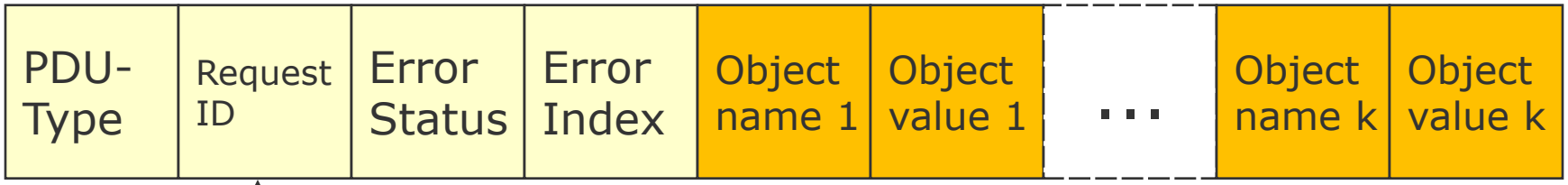
- Like FTP, SNMP uses two well-known ports to operate:
  - UDP Port 161
    - SNMP Messages
  - UDP Port 162
    - SNMP Trap Messages

Structure of an SNMP-PDU



Command Dependent PDU Part

PDU-Fields for query/set of managed objects:



Identification of outstanding requests

Result of the query
0 – noError
1 – tooBig
2 – noSuchName
3 – badValue
4 – readOnly
5 – genErr

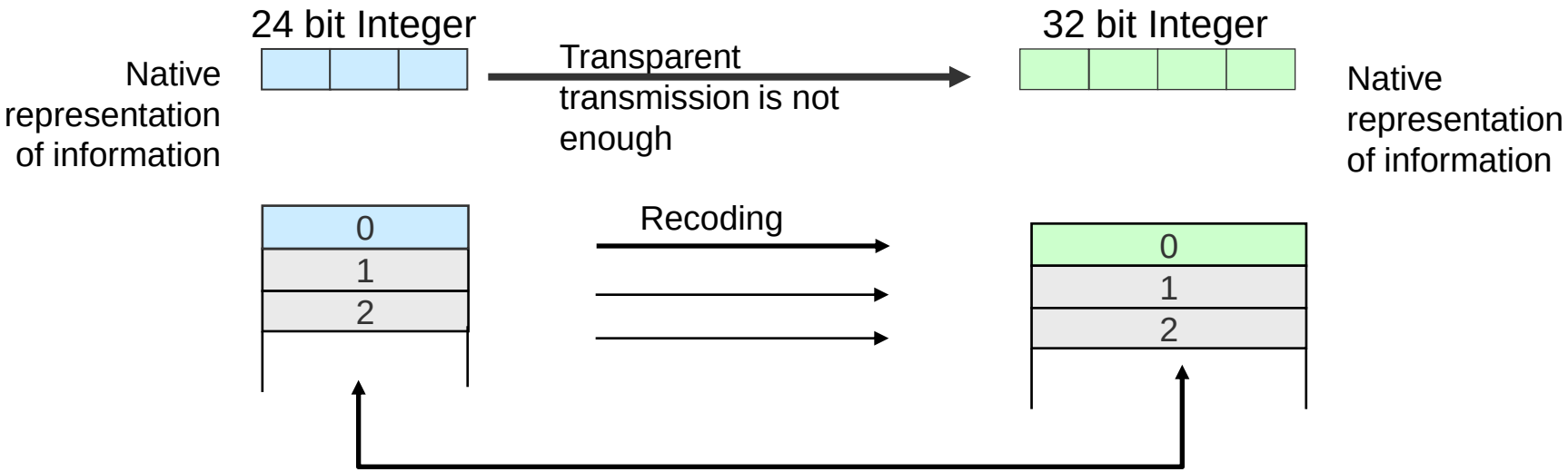
Reference to the variable binding pair which caused the failure.

Simple Network Management Protocol

Heterogeneous Representation of Data with ASN.1 and BER

Heterogeneous Representation

- Problem: Different computer systems use different data representation (Little/Big Endian, 16/24/32 bit etc.)
  - Consequence: Recoding of the data is required
    ➡ Exchange standards are required
- Recoding requires:
  - Coding of the representation (Syntax) of information
  - Retain the meaning (Semantic) of information



ASN.1: Definition

- ASN.1 (Abstract Syntax Notation One)
  - ISO standardized language for representation-independent specification of data types
  - Is used in SNMP to describe management objects
- Elementary data types
  - Boolean, Integer, Bitstring, Octetstring, IA5String, …
- Structured data types
  - Sequence: Ordered list of data types (Like record in PASCAL)
  - Set: Unordered set of data types
  - Sequence Of: Like an array in C
  - Set Of: Unordered set of elements from the same data type
  - Choice: Like union structure in C
- Example:

```
Employee ::= Set {
      Name IA5String,
      Age Integer,
      Personalnr Integer
}
```

SNMP Data Types

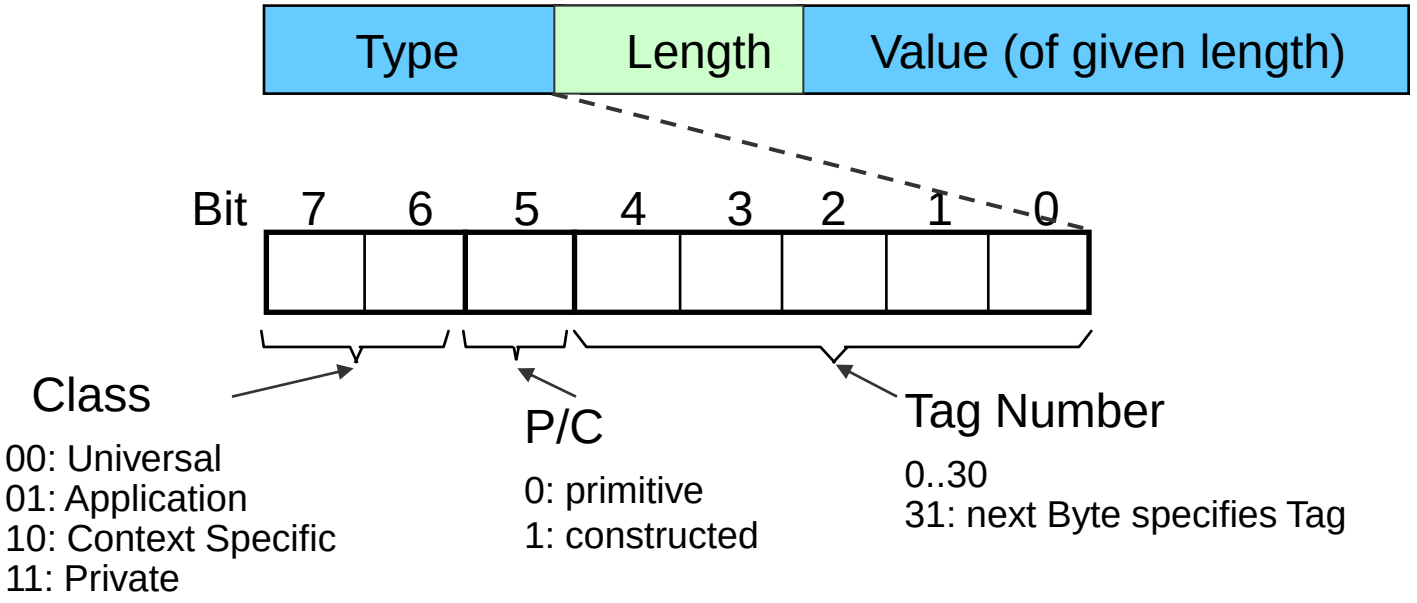
- INTEGER
  - signed 32-bit integer
- OCTET STRING
- OBJECT IDENTIFIER (OID)
- NULL
  - not actually data type, but data value

Defined by ASN.1

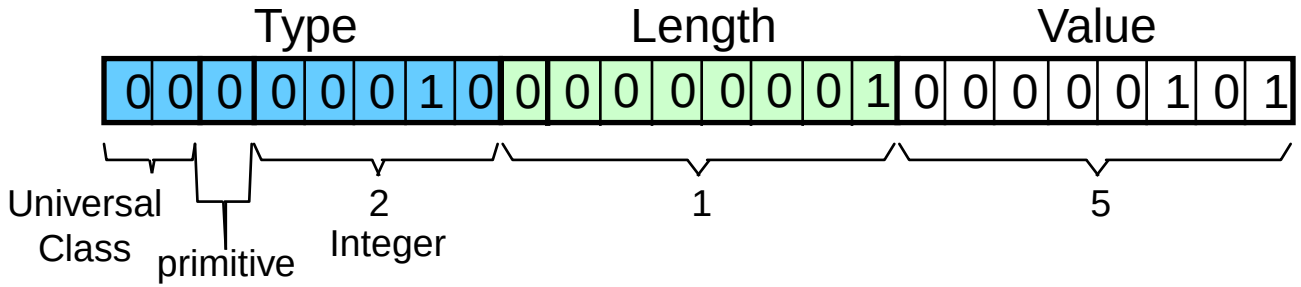
- IpAddress
  - OCTET STRING of size 4, in network byte order
- Counter
  - unsigned 32-bit integer (rolls over)
- Gauge
  - unsigned 32-bit integer (will top out and stay there)
- TimeTicks
  - unsigned 32-bit integer (rolls over after 497 days)
- Opaque
  - used to create new data types not in SNMPv1
- DateAndTime, DisplayString, MacAddress, PhysAddress, TimeInterval, TimeStamp, TruthValue, VariablePointer
  - textual conventions used as types

Defined by RFC 1155

ASN.1: Transfer Syntax - Basic Encoding Rules (BER)

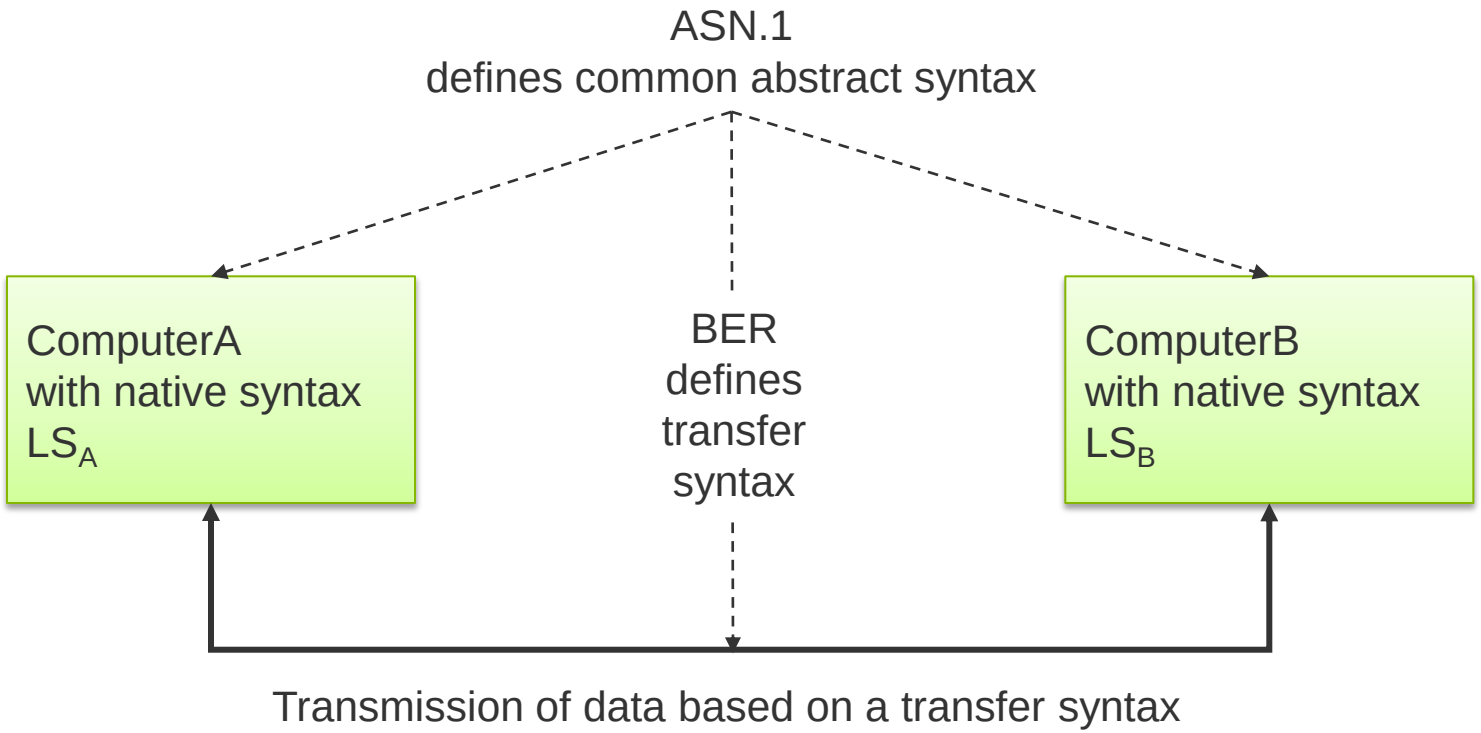


Example: Integer value 5



Relationship between ASN.1 and BER

- Basic Encoding Rules
  - The relationship between ASN.1 and BER parallels that of source code and machine code.
  - CCITT X.209 specifies the Basic Encoding Rules
  - All SNMP messages are converted / serialized from ASN.1 notation into smaller, binary data (BER)



Simple Network Management Protocol

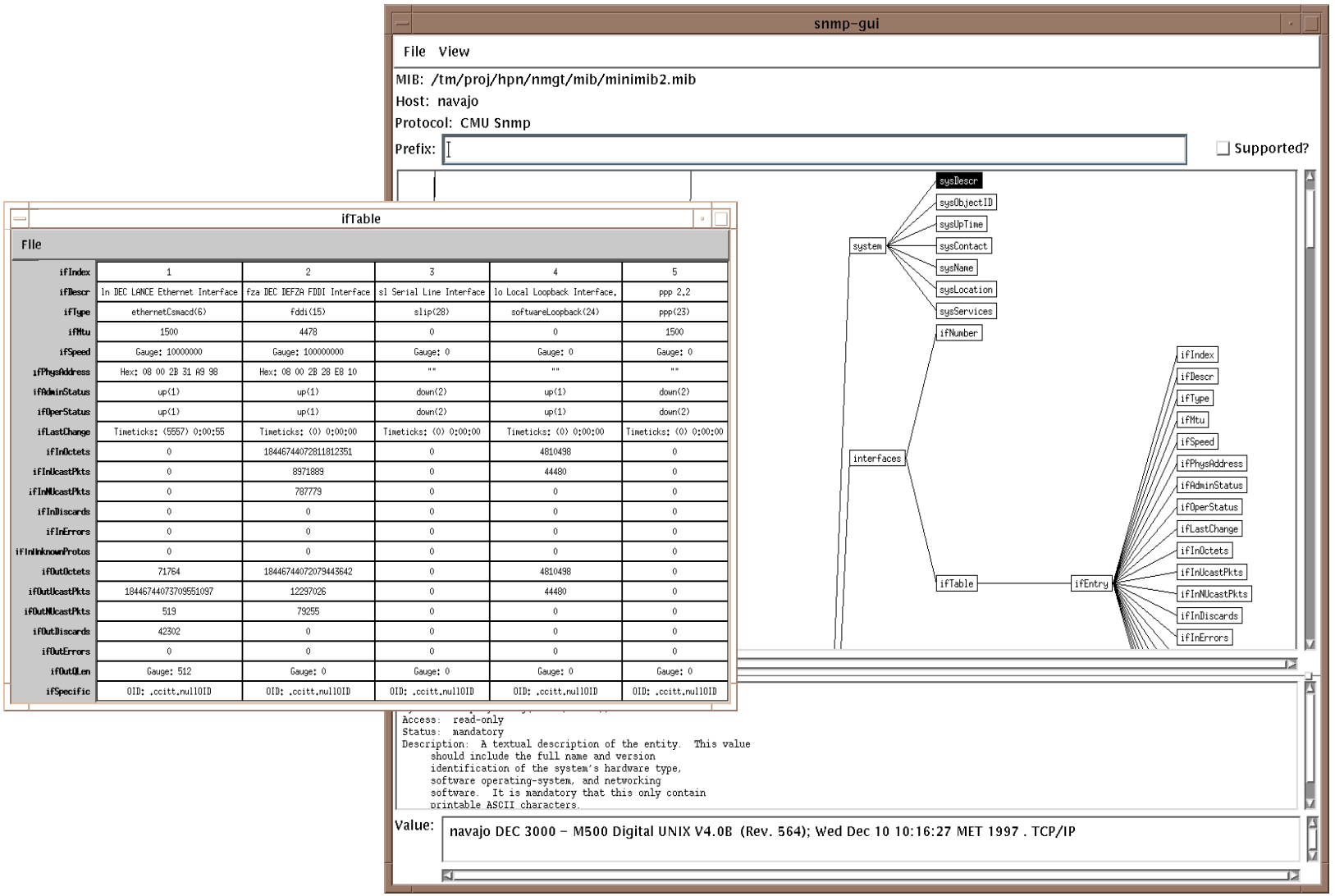
Tool Support

Management Applications: Command Line Tools

- Public Domain Tools (many variants)
  - Commands: snmpget, snmpnext, snmpwalk, snmpset, …
  - Generation and decoding of SNMP data units
  - Sometimes also support for MIB files

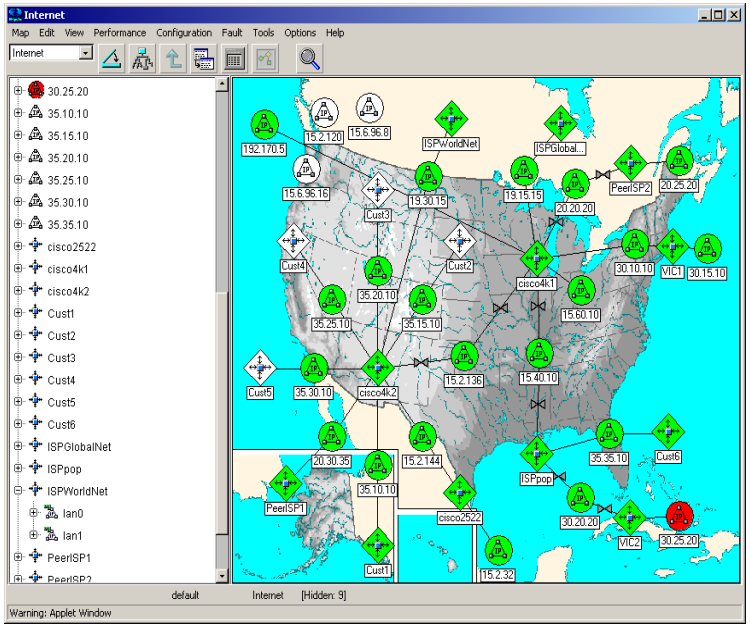
```
snmpget -v 1 172.76.14.213 public .1.3.6.1.2.1.1.1.0
system.sysDescr.0 = „TERArouter Network Platform"

snmpwalk -v 1 172.76.14.213 public .1.3.6.1.2.1.1
system.sysDescr.0 = „TERArouter Network Platform"
system.sysObjectID.0 = OID: enterprises.DEC.2.15.3.3
system.sysUpTime.0 = Timeticks: (456990767) 52 days, 21:25:07
system.sysContact.0 = "mueller@mi.fu-berlin.de"
system.sysName.0 = ""
system.sysLocation.0 = ""
system.sysServices.0 = 10
```

Management Applications: A MIB Browser

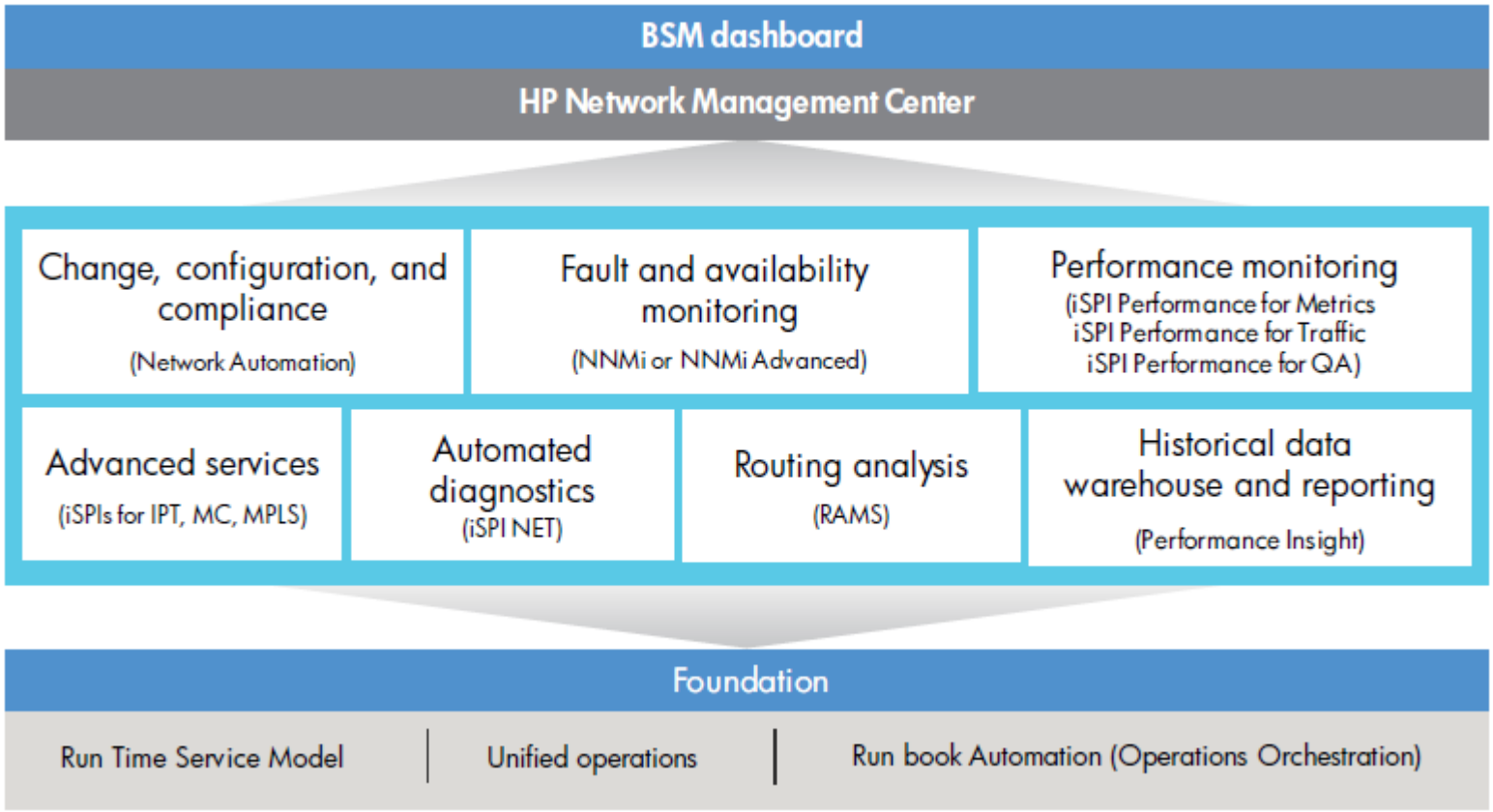


Management Platforms

- There are many products available which use SNMP
  - CiscoWorks2000, IBM Tivoli Netview, HP Network Management Center (was OpenView), Oracle Enterprise Manager...
- Very often the platforms provide more functionality than only component management
  - Administration
  - Business- and management processes
  - Network planning
  - Software distribution



Example: HP Network Management Center



Summary

- Application protocols in the Internet…
  - handle the functionality of layer 5 – 7 of the OSI reference model
  - are unaware of the network, focusing on application-related tasks
  - defining syntax, semantics, and order of exchanged messages
  - Typically relatively simple, text-based protocols

- Other interesting areas are
  - Mobile Communications: network and protocol variants for wireless and mobile networks
  - Simulation: simulation based evaluation of systems, e.g., networks and protocols
  - Distributed Systems: cooperation of application processes as addition to communication
  - Modeling and Evaluation of Communication Systems: methods for analytic evaluation of new systems or protocols
  - Security in Communication Networks: security-related aspects of communication, e.g., encryption, authentication, anonymity, …
  - Multimedia Systems: media formats and coding, quality of service mechanisms, and transfer/storage of multimedia data