



TK1100

Internett og WWW

6'te forelesning,

Høsten 0x07E2



Status

- Vi har gått gjennom:
 - Oppbygging
 - Von Neumann modellen
 - Hardware
 - Logiske porter
 - CPU
 - Hovedkort og busser
 - Periferiutstyr
 - Software
 - Datatyper (binære)
 - Instruksjoner (typer)
 - Filformater (noen få)
 - Operativsystem
 - Oppgave
 - Oppbygging
 - Virkemåte
 - Virkemåte:
 - Digital representasjon
 - Tall (hel- og flyt-)
 - Text (kodeskjema)
 - Bilde (bitmap+vektor)
 - Sekvensiell kjøring av instruksjoner
 - Adressering i RAM
 - Aritmetiske og logiske operasjoner
 - Interaksjon (WIMP vs CLI)



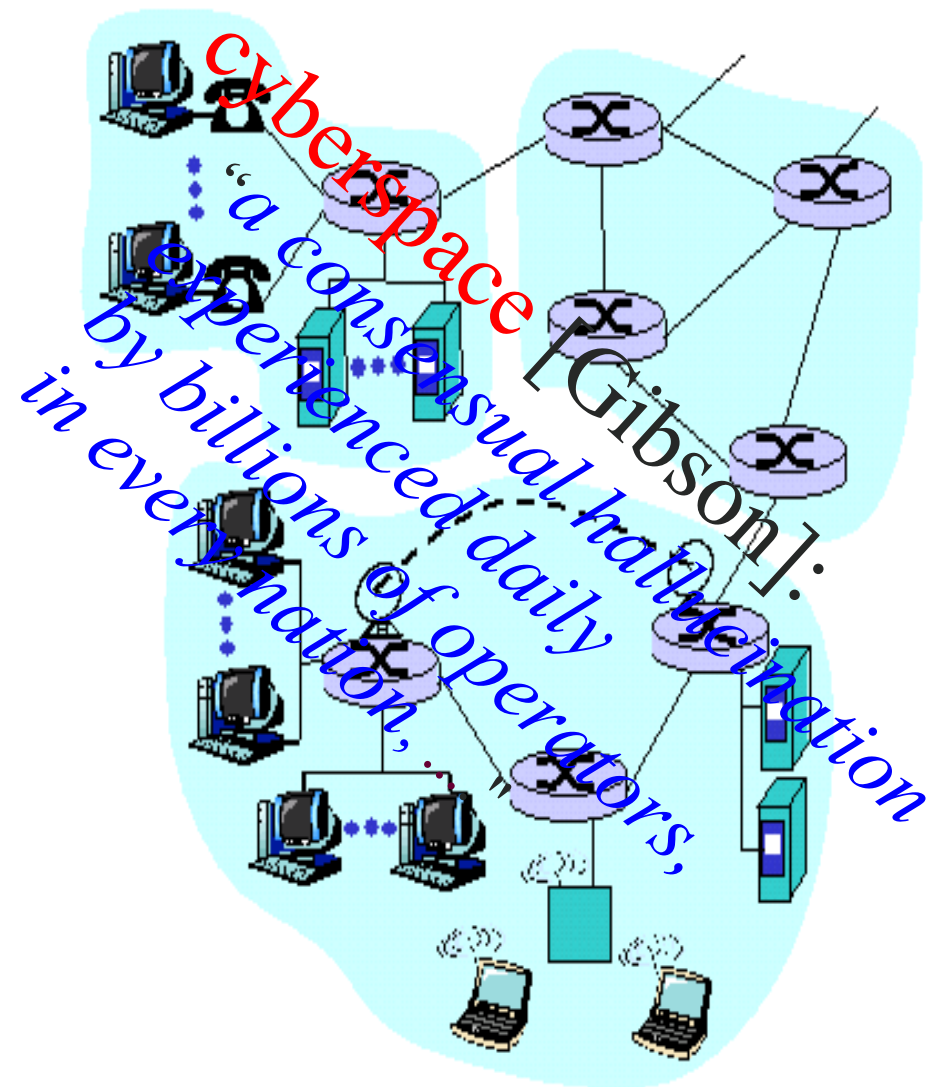
I dag

- Så langt har vi sett på hvordan enkeltmaskiner fungerer på ulike nivåer
 - Fra logisk port
 - Opp til systemprogramvare
 - I PGR100 ser dere på hvordan man lager programmer som bruker OS og Hardware
 - I DB1100 ser dere på hvordan data organiseres og gjøres brukbare
 - I PRO100 har dere sett på hvordan kode og presentere informasjon i web-sider, men hvordan får vi distribuert dem?
- I dag
 - Hvordan kan maskiner samarbeid over datanett?
 - Svaret er protokoller og standarder.



Introduksjon

- Hva er Internett?
 - Milliarder av computere i nett
 - Nettet tilgjengelig for "alle"
 - Enkle brukergrensesnitt
 - Ikke fullt så enkelt bak kulissene
 - Et sett med **standarder** for nettverks-kommunikasjon som sammenkople ulike LAN og WAN ("**TCP/IP-stacken**")





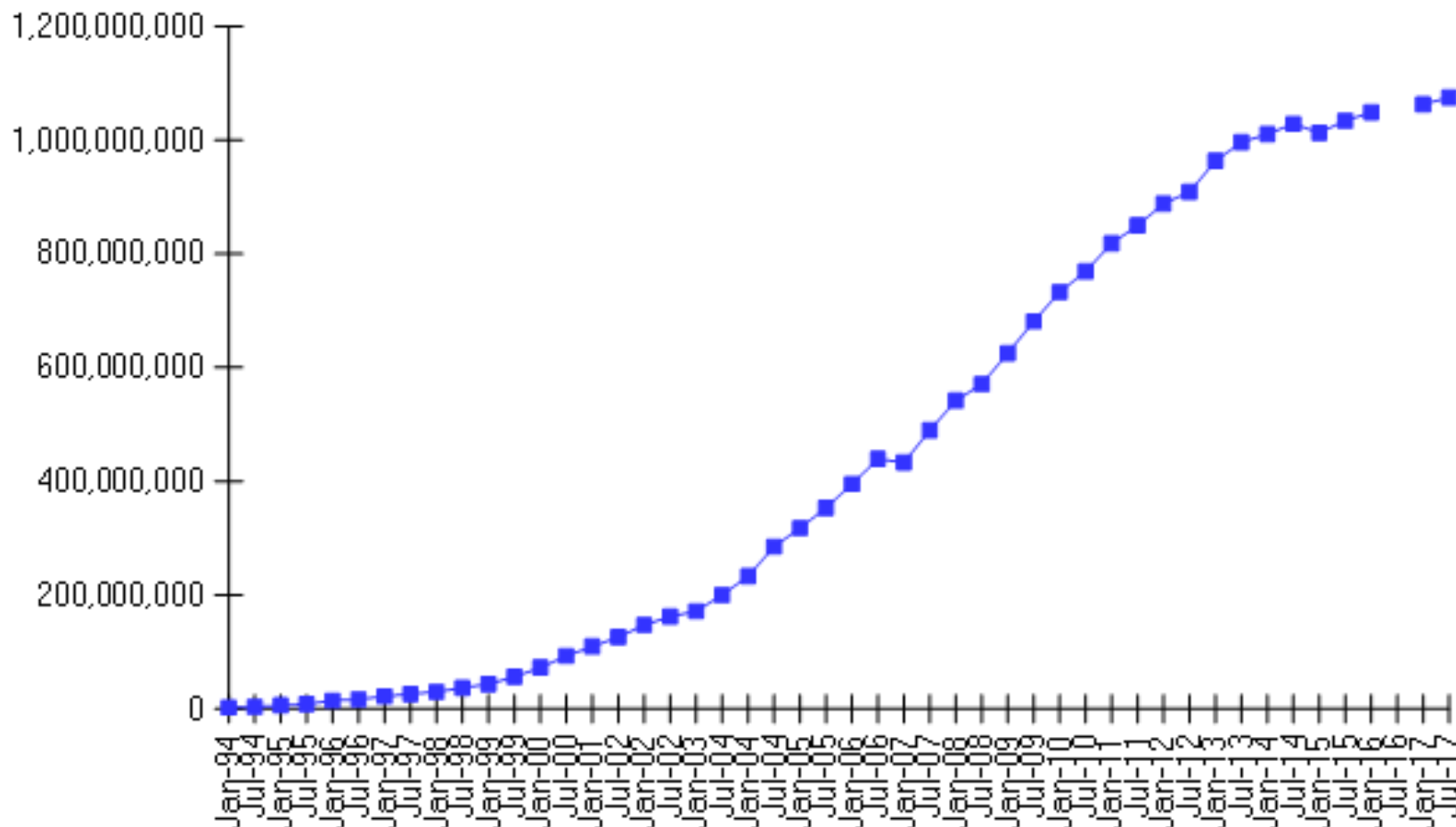
Internett

- Teknisk **Infrastruktur** som kopler sammen ulike nettverk ved hjelp av TCP/IP-suiten av **protokoller**
- **WWW** er IKKE det samme som Internett!!!
 - Uansett om det har blitt vanlig språkbruk i Norge og andre steder.
 - WWW er **en applikasjon** levert med HTTP



Antall domener

Internet Domain Survey Host Count

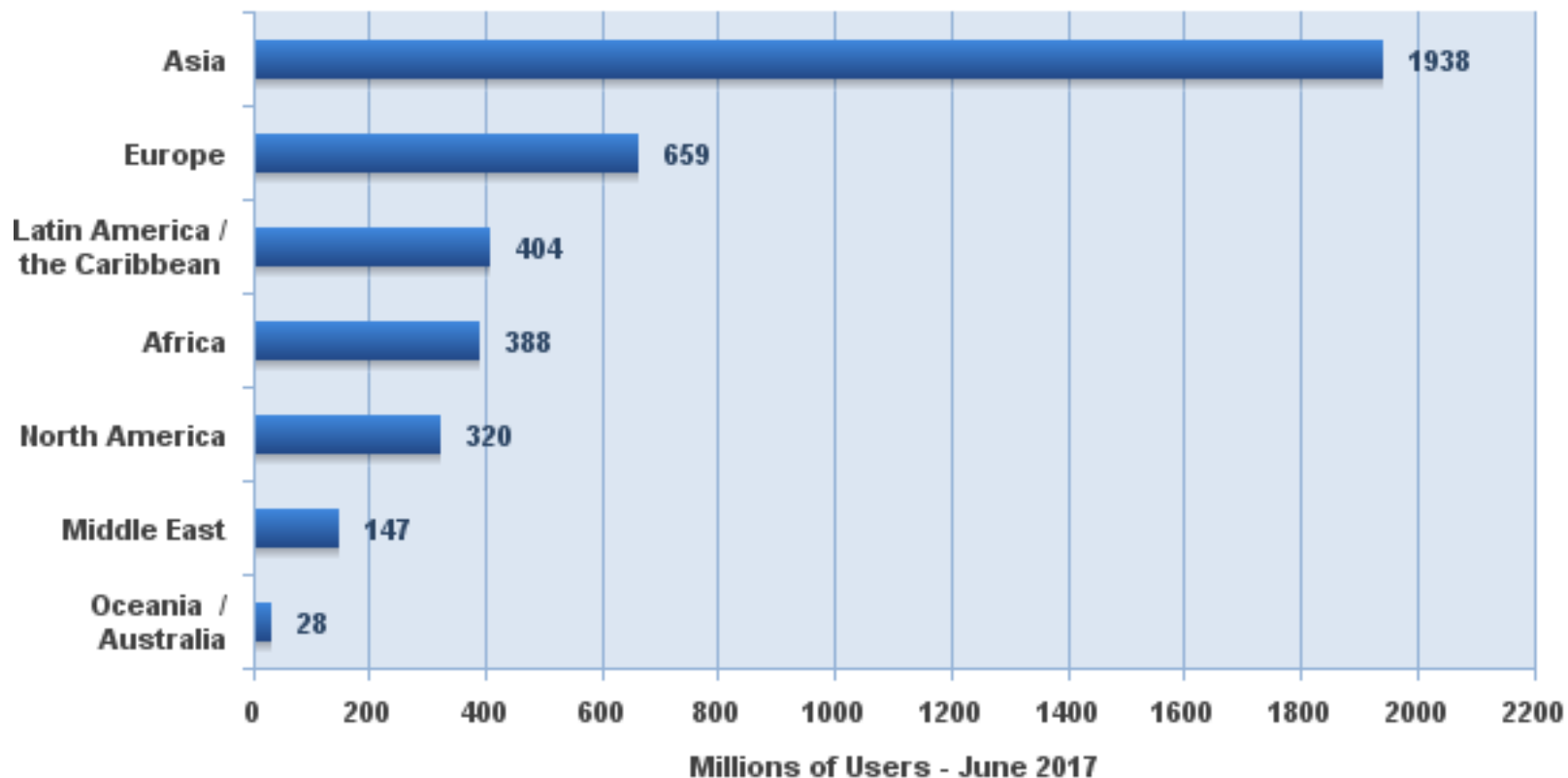


Source: Internet Systems Consortium (www.isc.org)



Antall brukere

Internet Users in the World by Geographic Regions - June 30, 2017



Source: Internet World Stats - www.internetworldstats.com/stats.htm

Basis: 3,885,567,619 Internet users estimated in June 30, 2017

Copyright © 2017, Miniwatts Marketing Group



Tekniske standarder

- ... er normer og krav til tekniske systemer
- Nedskrevet i et dokument som beskriver hvilke krav man må oppfylles for å tilfredsstille standarden.
- *De Jure* (etter lov) vs *De Facto* (faktisk)
 - Standarder som følges ...
 - fordi «alle» finner det fordelaktig kalles **De Facto**.
 - fordi det kreves (typisk i lov) kalles **De Jure**
 - Eksempel:
 - MP3 i stedet for CD Wav på nettet (De facto)
 - HTML frem til 1995 (De facto -> De jure)
 - Microsoft Word DOC (De facto -)



Organisatorisk

RFC'er som
standardiserings-
prosess

– Request for
Comment

IANA, RIPE m.fl.
adm. (DNS-)NAVN
og (IP-)NUMMER

Internet Society (ISOC)

Provides leadership in issues relating to the Internet and serves as the organizational home of groups responsible for Internet standards.

Internet Engineering Task Force (IETF)

Develops and disseminates the standards and protocols that define the architecture and software of the Internet.

Internet Architecture Board (IAB)

Defines the overall architecture of the Internet, providing guidance and broad direction to the IETF.

Internet Engineering Steering Group (IESG)

Manages the technical activities of the IETF and oversees the process of standards development.

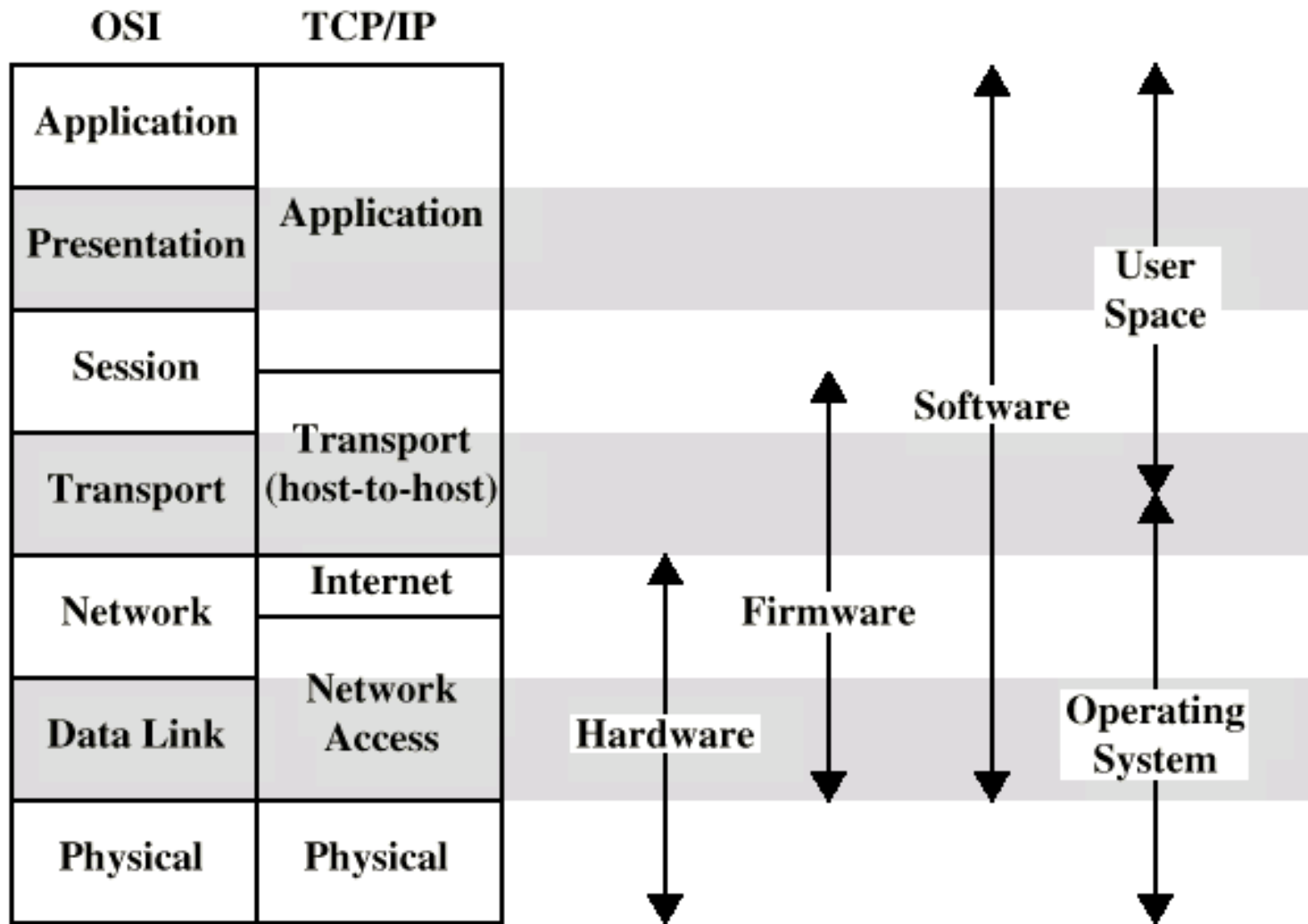
Internet Research Task Force (IRTF)

Organizes research groups on topics related to Internet protocols, applications, architecture, and technology.



OSI vs TCP/IP

- Hvilke modeller (arkitektur) brukes?





KRONOLOGI



Internett historikk: Forspill

1961

- Kleinrock: **Pakkeswitching** som prinsipp

1964

- Baran: Pakkeswitching i militære nett

1967

- ARPAnet (Advanced Research Project Agency) unnfanget

1969

- Første **ARPAnet** node operativ

1972

- ARPAnet demonstrert med 15 noder, NCP, Mail
- **Norge** tilknyttet



Internett historikk: 70-tallet

1970

- ALOHAnet på Hawaii

1973

- Metcalfe: Ethernet

1974

- Cerf & Kahn: Nettverksarkitektur, Internetts "fedre": «Ende-til-ende prinsippet»

1976

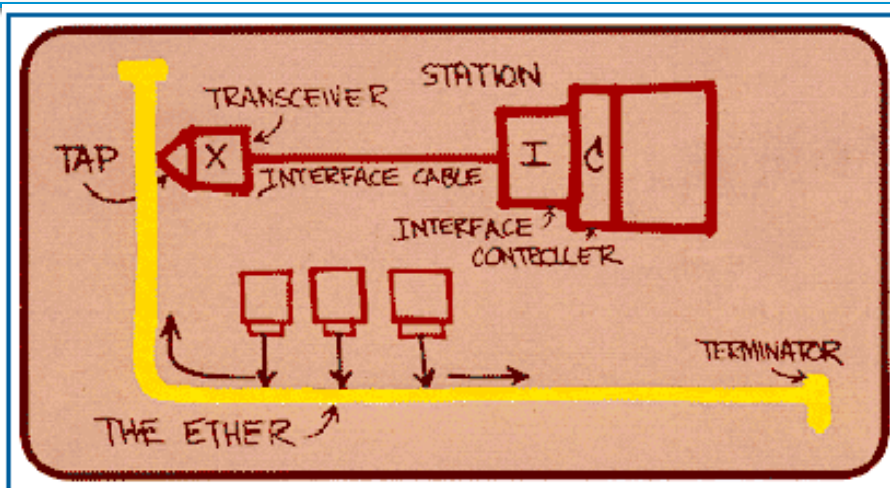
- UNINETT blir etablert for å levere Internett til norske universitet

1977-79

- Proprietære arkitekturer, SNA, XNA
- Pakkeswitching på fast lengde (forløper til ATM)

1979

- ARPAnet har 200 noder



Cerf and Kahn's internetworking prinsipper:

- minimalisme, autonomi – krever ingen indre endringer for å kople sammen netteverk
- Best effort service model
- Tilstandsløse routere
- Desentralisert kontroll

definerer dagens Internet arkitektur



Internett historikk: 80-tallet

1982

- **SMTP** protokoll (**email** sending) definert

1983

- **TCP** blir off. standardprotokoll for ARPAnet

1988

- TCP **trafikk-kork kontroll**, **IPv4**, UDP

1989

- 100 000 verter i totalt nettverk



Internett historikk: 90-tallet

1991

- Berners-Lee: HTML og HTTP

1994

- Mosaic, senere Netscape

1994

- ARPAnet «nedlagt», NSFnet overtar, kommersialiseringen starter

2000

- 600 millioner Internett-brukere
- Mer enn 1 milliard tilgjengelige sider på WWW

2008

- 1,2 milliarder brukere fordelt på 500 millioner hosts



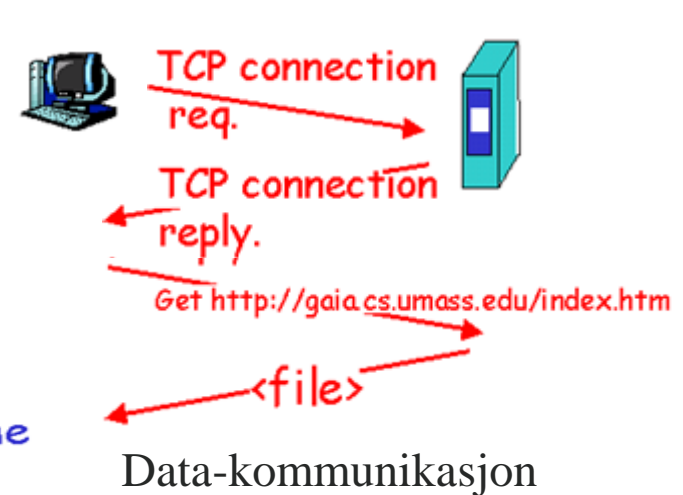
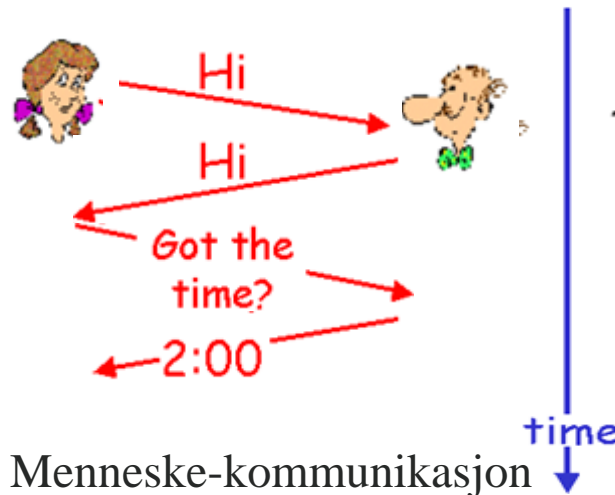
PROTOKOLLER (OG ANDRE BEGREP)



Protokoll

Hva er en protokoll?

- Brukes til kommunikasjon mellom "like" funksjoner
- Må snakke det samme "språk"
- Funksjoner
 - Bruker-applikasjoner
 - E-mail
 - Terminaler
- Systemer
 - Computere
 - Terminaler
 - Sensorer





Protokoll

- Nøkkel-elementer for en **protokoll**
 - **Syntaks**
 - "ord, ordstilling og grammatikk"
 - Data-formater
 - Signal-nivåer
 - **Semantikk**
 - "betydning"
 - Kontroll-informasjon
 - Feil-behandling
 - **Timing**
 - Hastighets-tilpasning
 - Sekvens-tilordning

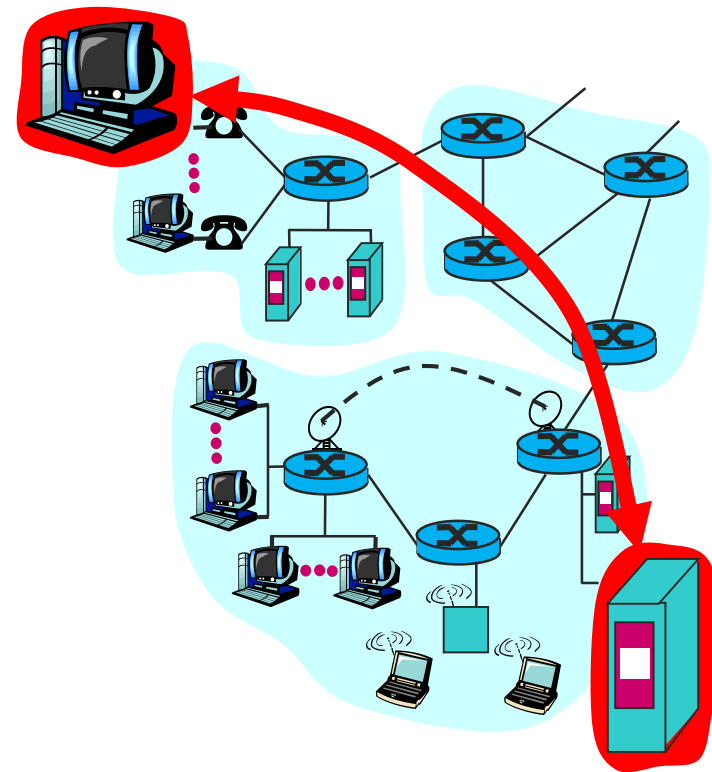
Nettverkets ytterkanter

Ende-systemer (verter = hosts)

- Kjører **applikasjoner**
 - WWW, telnet, email

Tjenestemodeller:

- **Klient-tjener** modell
 - Forespørsler kommer fra klient som mottar service fra tjener
 - F.eks. WWW browser/tjener
- Peer-peer (P2P) modell
 - Alle computere er likeverdige
 - F.eks. BitTorrent, Skype,...





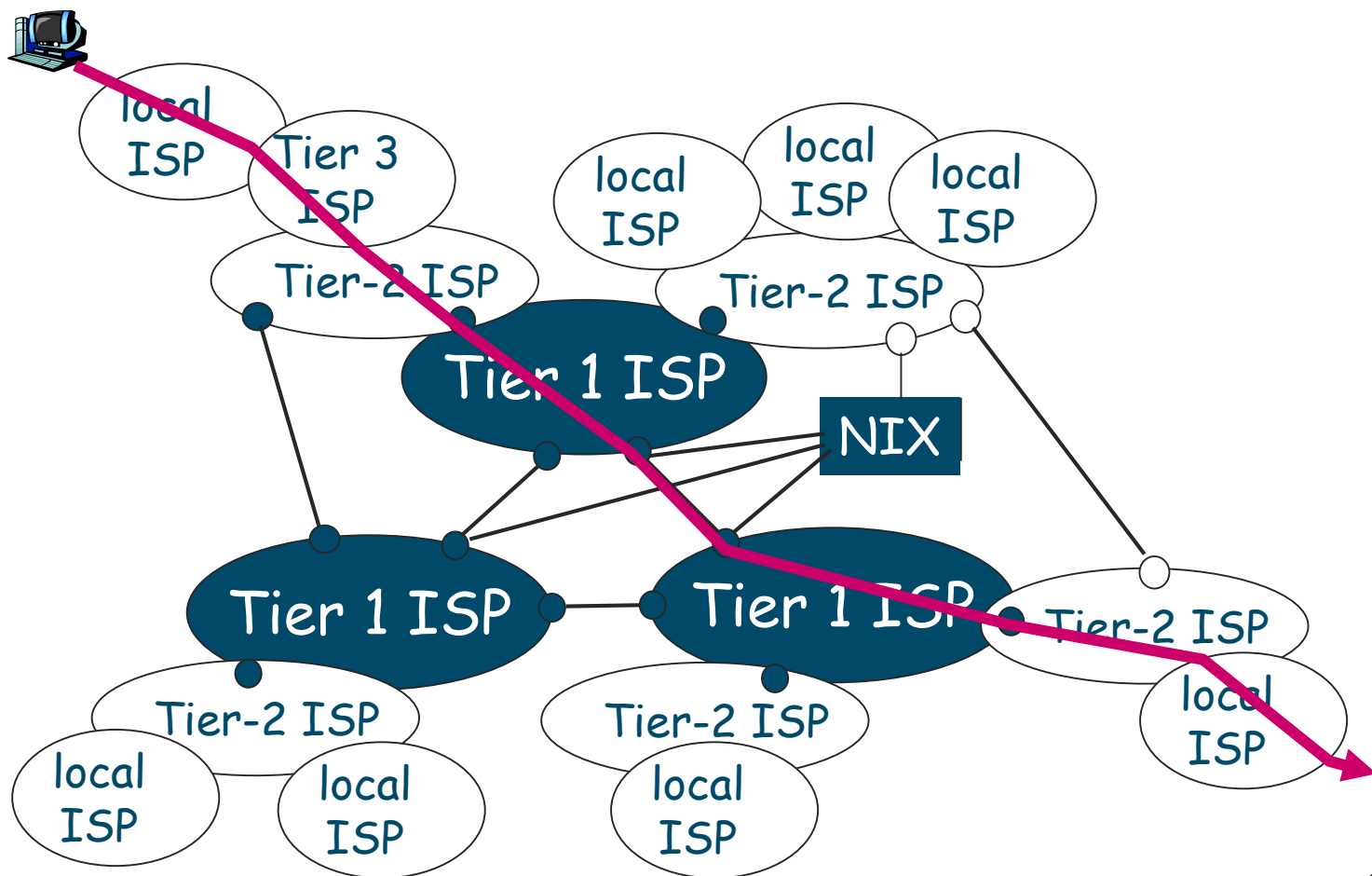
Nettverkets ytterkanter

- **Forbindelses-orientert** service
 - Oppretter kontakt (**handshake**) før data overføres
 - Etablerer overførings-**tilstand** i endesystemene
- **TCP** (Transmission Control Protocol)
 - Internetts mest brukte forbindelses-service
 - Pålitelig og ordnet dataoverføring
 - Service ved tap av data
 - Flytkontroll mot oversvømmelse
 - Reduserer farten på avsender ved for stor trengsel på nettet (metningskontroll)



Nettverk av nettverk!

- Internett var opprinnelig laget for å kople sammen ulike typer lokalnettverk
- En datapakke passerer altså (ofte) gjennom mange ulike typer nettverk!





TRACEROUTE APPLIKASJONEN

- Viser hvilke **routerne** en pakke går gjennom
- Kan måle forsinkelsene mellom nodene med applikasjonen `tracert` (NT: `tracert`)
- Stadig oftere stoppet av brannmur og o.l.

```
C:\WINNT\System32\cmd.exe

C:\>tracert www.cia.gov

Tracing route to www.odci.gov [198.81.129.100]
over a maximum of 30 hops:

  1  <10 ms  <10 ms  10 ms  10.21.11.2
  2  *        *        *        Request timed out.
  3  <10 ms  <10 ms  10 ms  194.19.88.181
  4  10 ms   *        *        oslo-i2.banetele.net [194.19.1.198]
  5  <10 ms  10 ms   <10 ms  POS5-0.GW3.OSL2.ALTER.NET [146.188.57.205]
  6  <10 ms  <10 ms  10 ms  so-3-0-0.XR1.OSL2.ALTER.NET [146.188.15.93]
  7  10 ms   10 ms   10 ms  so-4-2-0.TR1.CPH3.ALTER.NET [146.188.3.41]
  8  111 ms  110 ms  110 ms  so-5-0-0.IR1.DCA4.ALTER.NET [146.188.5.245]
  9  *        110 ms  110 ms  so-0-0-0.IL1.DCA6.ALTER.NET [146.188.13.33]
 10 *        111 ms  110 ms  0.so-1-0-0.TL1.DCA6.ALTER.NET [152.63.9.194]
 11 111 ms  120 ms  120 ms  0.so-6-0-0.CL1.RDU1.ALTER.NET [152.63.34.93]
 12 121 ms  120 ms  120 ms  123.ATM6-0.GW6.RDU1.ALTER.NET [152.63.38.229]
 13 120 ms  131 ms  120 ms  res2rdu1-gw.customer.alter.net [157.130.85.234]
 14 *        *        *        Request timed out.
 15 *        *        *        Request timed out.
 16 *        ^C

C:\>
```

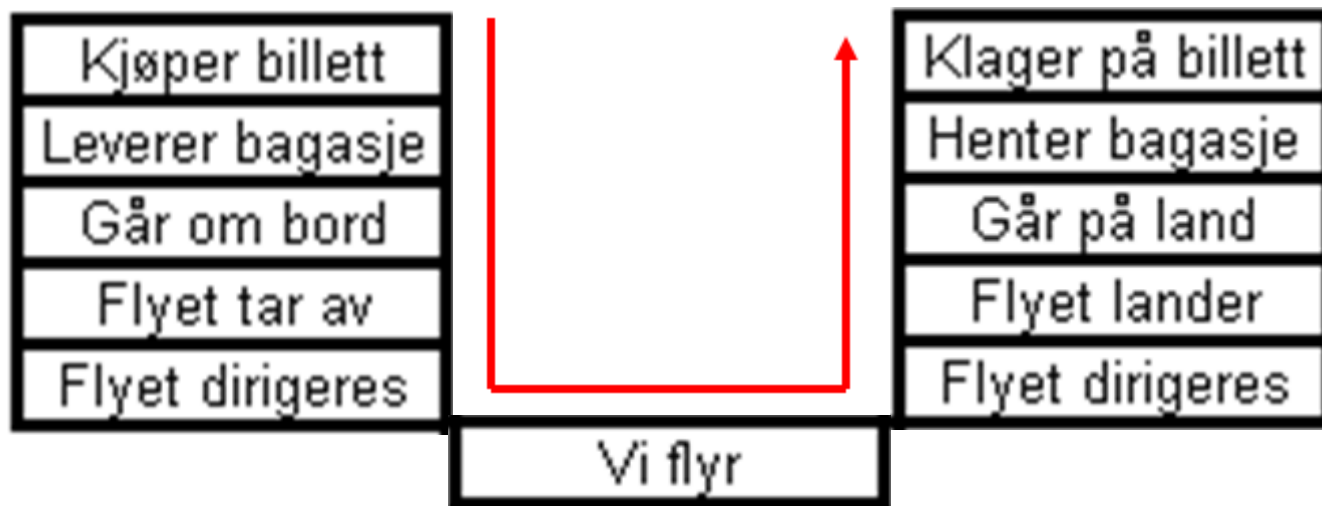


Oppdeling av nettverk

- Nettverket består av mange deler
 - Verter
 - Routere
 - Linker til forskjellige media
 - Applikasjoner
 - Protokoller
 - Programvare
 - Maskinvare
- Hvordan skal vi få organisert alt dette?
 - Splitt og hersk!
 - Del opp i «abstraksjonsnivåer» ut fra funksjonalitet

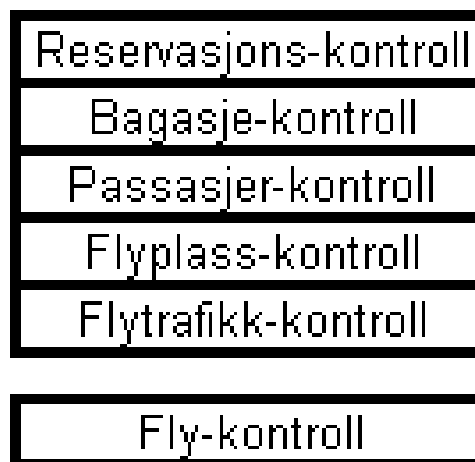


Ex: Organisering av flytur



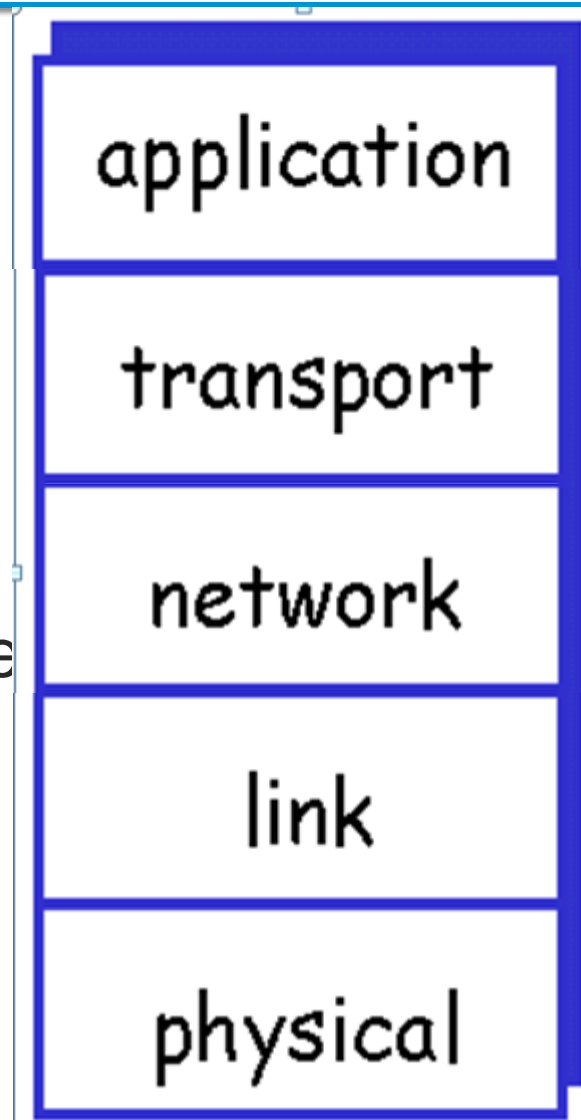
- Vi **abstraherer**:

- Hva er felles funksjonalitet?
- Vi definerer regler (protokoller) for hvordan ting skal foregå på hvert trinn/nivå



Internet protocol stacken

- Lag for nettverks **applikasjoner**
 - FTP, SMTP, HTTP, ...
- Lag for **transport** mellom verter/prosesser
 - TCP, UDP, ..
- Lag for **nettverks-ruting** ende-til-ende
 - IP, ICMP, RIP..
- Lag for overføring nabo-til-nabo
 - Ethernet
- Lag for fysisk overføring
 - Kabling, plugger, signalnivå





IPCONFIG (ifconfig)

```
C:\>ipconfig /all

Windows IP-konfigurasjon

    Vertsnavn . . . . . : NITH-blistog
    Primær DNS-suffiks . . . . . :
    Nodetype . . . . . : Hybrid
    IP-ruting aktivert . . . . . : Nei
    WINS Proxy aktivert. . . . . : Nei
    Søkkeliste for DNS-suffiks. . . . : oslo.nith.no

Ethernet-kort eth0:

    Tilkoblingsspesifikt DNS-suffiks : oslo.nith.no
    Beskrivelse . . . . . : Broadcom NetXtreme 57xx Gigabit
oller
    Fysisk adresse . . . . . : 00-13-72-94-FF-78
    DHCP aktivert. . . . . : Ja
    Automatisk konfigurasjon aktivert: Ja
    IP-adresse . . . . . : 10.21.5.228
    Nettverksmaske . . . . . : 255.255.252.0
    Standard gateway . . . . . : 10.21.4.1
    DHCP-server. . . . . : 10.21.4.131
    DNS-servere. . . . . : 10.21.4.131
                           10.21.21.101
    Primær WINS-server . . . . . : 10.21.4.131
    Leasingavtale mottatt. . . . . : 26. oktober 2008 17:25:30
    Leasingavtale utgår. . . . . : 26. oktober 2008 19:25:30
```

- Viser IP-nettverksparametre som de har blitt satt på (NT) arbeidsstasjon
 - For å endre nettverksparametere må du bruke `netsh`
- Linux/OS X: `ifconfig`
 - "kraftigere" da du også kan endre oppsett med samme



netstat -an

```
C:\>netstat -an
```

```
Aktive tilkoblinger
```

Prot.	Lokal adresse	Ekstern adresse	Tilstand
TCP	0.0.0.0:135	0.0.0.0:0	LISTENING
TCP	0.0.0.0:445	0.0.0.0:0	LISTENING
TCP	0.0.0.0:1028	0.0.0.0:0	LISTENING
TCP	0.0.0.0:3389	0.0.0.0:0	LISTENING
TCP	10.21.5.228:139	0.0.0.0:0	LISTENING
TCP	10.21.5.228:1418	64.233.183.18:443	ESTABLISHED
TCP	10.21.5.228:1445	158.36.191.141:443	TIME_WAIT
TCP	10.21.5.228:1457	64.233.183.18:443	ESTABLISHED
TCP	127.0.0.1:1035	0.0.0.0:0	LISTENING
TCP	127.0.0.1:1416	127.0.0.1:1417	ESTABLISHED
TCP	127.0.0.1:1417	127.0.0.1:1416	ESTABLISHED
TCP	127.0.0.1:1455	127.0.0.1:1456	ESTABLISHED
TCP	127.0.0.1:1456	127.0.0.1:1455	ESTABLISHED
TCP	127.0.0.1:5152	0.0.0.0:0	LISTENING
TCP	127.0.0.1:5354	0.0.0.0:0	LISTENING
UDP	0.0.0.0:161	*:*	
UDP	0.0.0.0:445	*:*	
UDP	0.0.0.0:500	*:*	
UDP	0.0.0.0:1025	*:*	
UDP	0.0.0.0:1029	*:*	
UDP	0.0.0.0:4500	*:*	
UDP	10.21.5.228:123	*:*	
UDP	10.21.5.228:137	*:*	
UDP	10.21.5.228:138	*:*	
UDP	10.21.5.228:5353	*:*	
UDP	127.0.0.1:123	*:*	
UDP	127.0.0.1:1030	*:*	
UDP	127.0.0.1:1040	*:*	

- Viser transportlag-tilstand ("åpne porter")



netstat -r

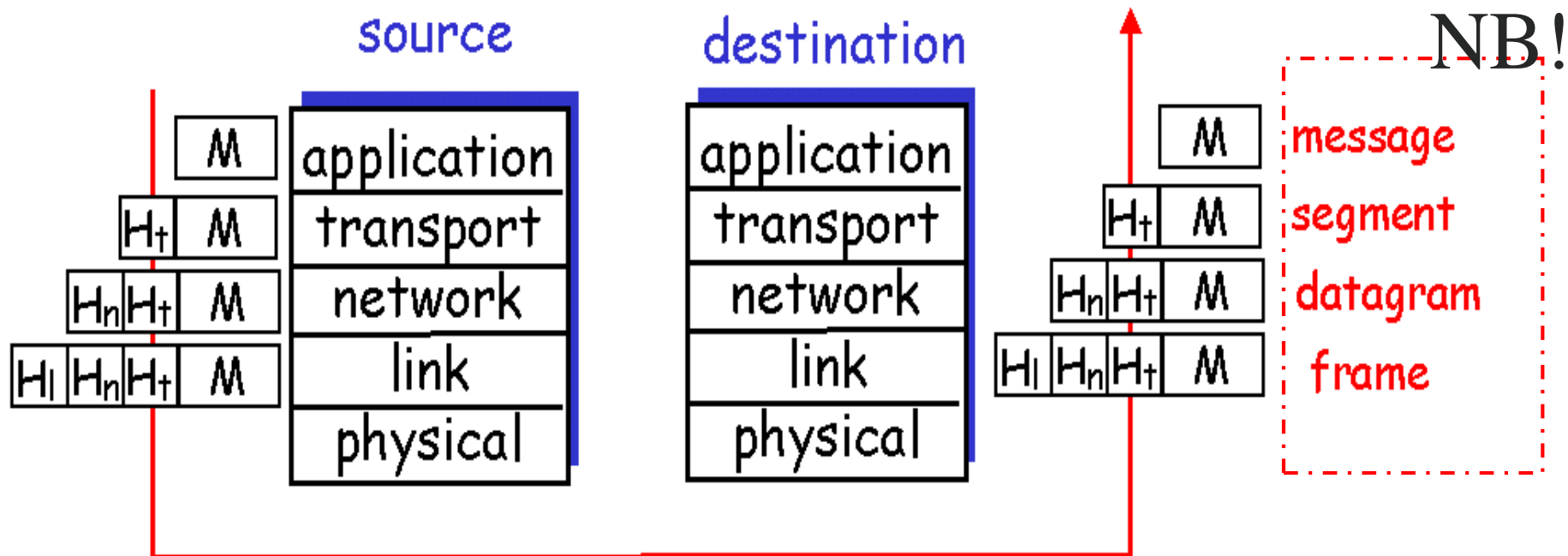
```
C:\>netstat -r

Rutingstabell
=====
Grensesnittliste
0x1 ..... MS TCP Loopback interface
0x2 ...00 13 72 94 ff 78 ..... Broadcom NetXtreme 57xx Gigabit Controller -
iport for pakkeplanlegger
=====
=====
Aktive ruter:
Nettverksmål   Nettverksmaske      Gateway      Grensesnitt  Metrikk
      0.0.0.0           0.0.0.0        10.21.4.1    10.21.5.228    20
      10.21.4.0       255.255.252.0    10.21.5.228    10.21.5.228    20
      10.21.5.228     255.255.255.255  127.0.0.1    127.0.0.1     20
    10.255.255.255     255.255.255.255  10.21.5.228    10.21.5.228    20
      127.0.0.0        255.0.0.0       127.0.0.1    127.0.0.1      1
      169.254.0.0      255.255.0.0     10.21.5.228    10.21.5.228    30
      224.0.0.0        240.0.0.0       10.21.5.228    10.21.5.228    20
    255.255.255.255     255.255.255.255  10.21.5.228    10.21.5.228     1
Std. gateway:      10.21.4.1
=====
Faste ruter:
Ingen
```

- Viser lokal routing-tabell
- `route`-kommandoen lar deg også endre den...

Inn/ut- pakking av data

- Avsender: Hvert lag tar data fra laget ovenfor
 - Legger til informasjon (header), lager ny dataenhet
 - Leverer nye data til laget nedenfor
- Mottaker prosesserer data i motsatt rekkefølge





SIKKERHET?

- Internett var designet for et minimum av **pålitelighet** og med svært lite tanke på **sikkerhet**
 - Har blitt den viktigste spredningsvektoren for **malware**
- Nettverkssikkerhet
 - Hvordan beskytte kommunikasjon mot å bli avlyttet, utnyttet eller "krasjet"
 - Virus, ormer, trojanske hester, spyware, spam,...
 - Denial of Service Angrep (DOS)
 - Mye mer om dette i **TK2100** (etter jul)



Kritiske tjenestenivåer for applikasjoner

- **Tap** av data
 - Noen applikasjoner **tåler litt** tap av data
 - Audio, video
 - Andre må ha 100% pålitelig dataoverføring
 - Filoverføring
- Båndbredde/**bitrate** (bps)
 - Noen applikasjoner må ha en viss båndbredde
 - Multimedia
 - Andre kan bruke båndbredden dynamisk
 - Filoverføring
- **Timing**
 - Noen applikasjoner tåler ikke mye tidsforsinkelse/latens
 - Sanntidsprosesser, spill



BITRATE («båndbredde»)

- Båndbredde heter egentlig **bitrate** på norsk
 - Båndbredde er en relatert størrelse som måles i Hz (signalbehandling)
 - Uttrykk for den **teoretiske** dataoverføringskapasiteten på en **link**
- 1 Mbps = 1 000 000 b/s
 - Merk at dette ikke er **Mi**, men «**m**»
 - Cirka 108 KiB per sekund
- **Latens** = «responstid» er ofte like viktig i forhold til **opplevelse** av ytelse...

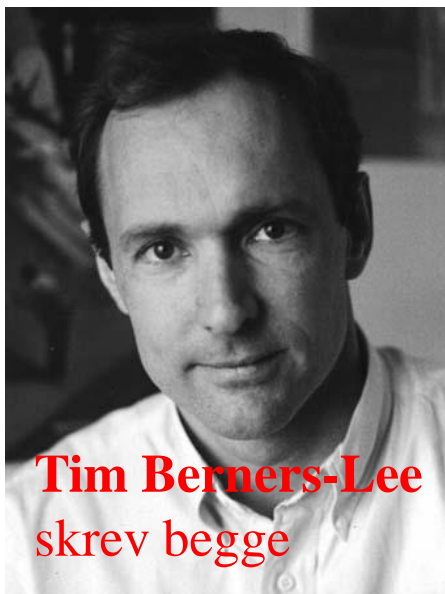


<http://info.cern.ch/hypertext/WWW/TheProject.html>

var den første «hjemmesiden»

publisert 6. August 1991

WORLD WIDE WEB



Tim Berners-Lee
skrev begge



Den første httpd-serveren

Noen sentrale protokoller og
begreper



Klient/tjener

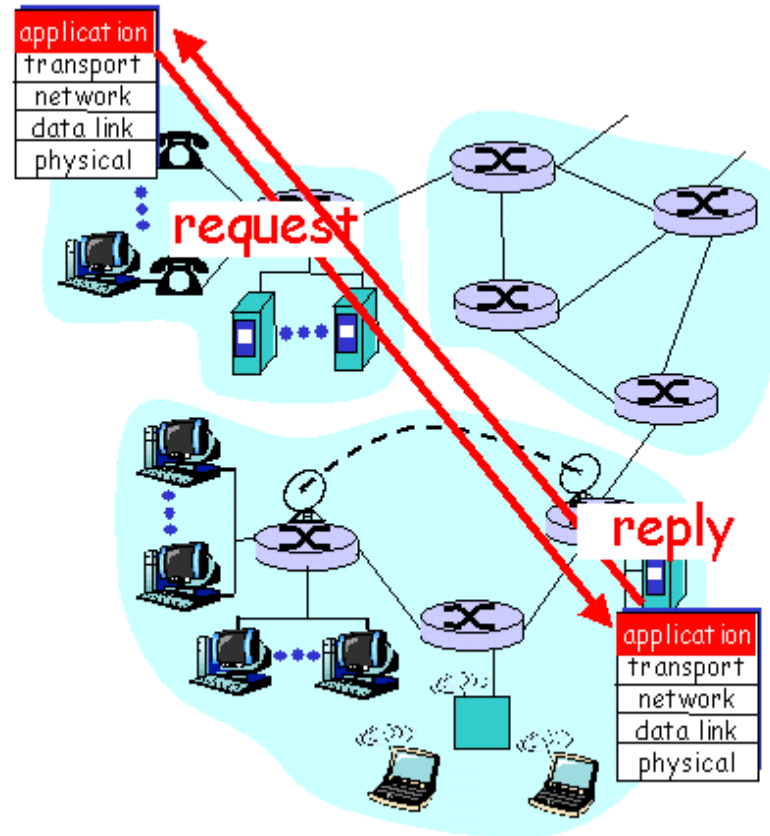
- Typisk oppsett i et nettverk

Klient

- Tar initiativet
- Ber om en service/tjeneste tjeneren
- På web er klienten i **browseren**

Tjener

- Leverer etterspurt service til klienten
- På web er dette **web-serveren**





Klient/tjener: poeng

- Du **ser** svært lite av hva som skjer «under panseret» i browseren.
- Det du oppfatter som «en side» er resultatet av mengdevis av enkeltutvekslinger av forespørsler og svar
 - F.eks. er hvert enkelt bilde i siden resultatet av en separat forespørsel (GET) og svar (200 OK + fil innhold)
- For å kunne feilsøke må vi hele tiden huske hva som egentlig foregår...

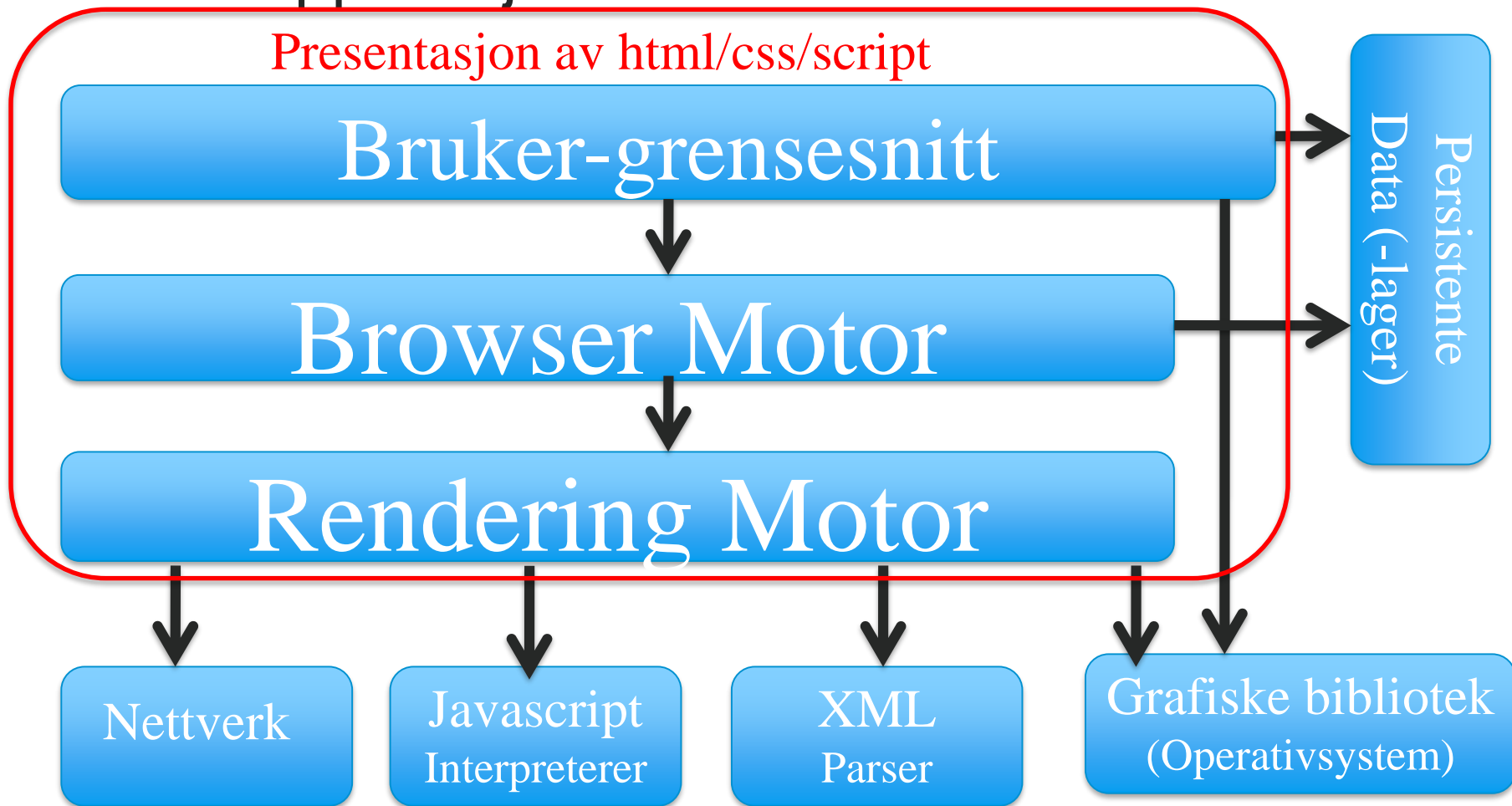


- [illegible]



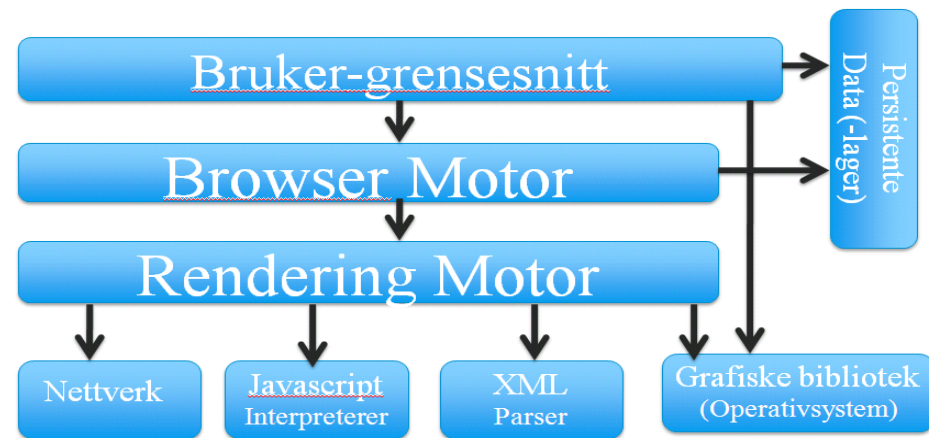
Hva er en browser/nettleser?

- Programvare som kjører klient-delen av en web-applikasjon



Arkitektur

- Bruker-grensesnitt (UI)
 - Toolbar, nedlastingsinfo, knapper, printing,...
- Browser motor
 - Høynivå grensesnitt mot Rendering motor
 - Laster URI, støtter "surfing": frem, tilbake, reload,...
- Rendering motor
 - Beregner side-layout og viser frem.
 - **Parser** html, css, ...
- Nettverk
 - Står for filoverføring (http, ftp, ..)
 - Oversetter mellom char-set, kan MIME
 - **Cache**



- Javascript Interpreterer
 - Kjører Javascript
- XML Parser
 - Parser html/xml til et **DOM**-tre
- Grafikkbibliotek
 - Tegne-, meny- og vindus-rutinene
 - Fonter
- Persistente data
 - Lagre bokmerker, sertifikater, personlig konfigurering



Layout motor = browser + rendering

- Rendering og browser motoren kombineres ofte i en Layout motor.
- Opera
 - Presto
 - men de gikk i 2013 over til samme som...
- Safari + Chrome
 - WebKit
- Firefox (Mozilla)
 - Gecko
- Lynx
 - Libwww + Curses



Arkitektur (2006)

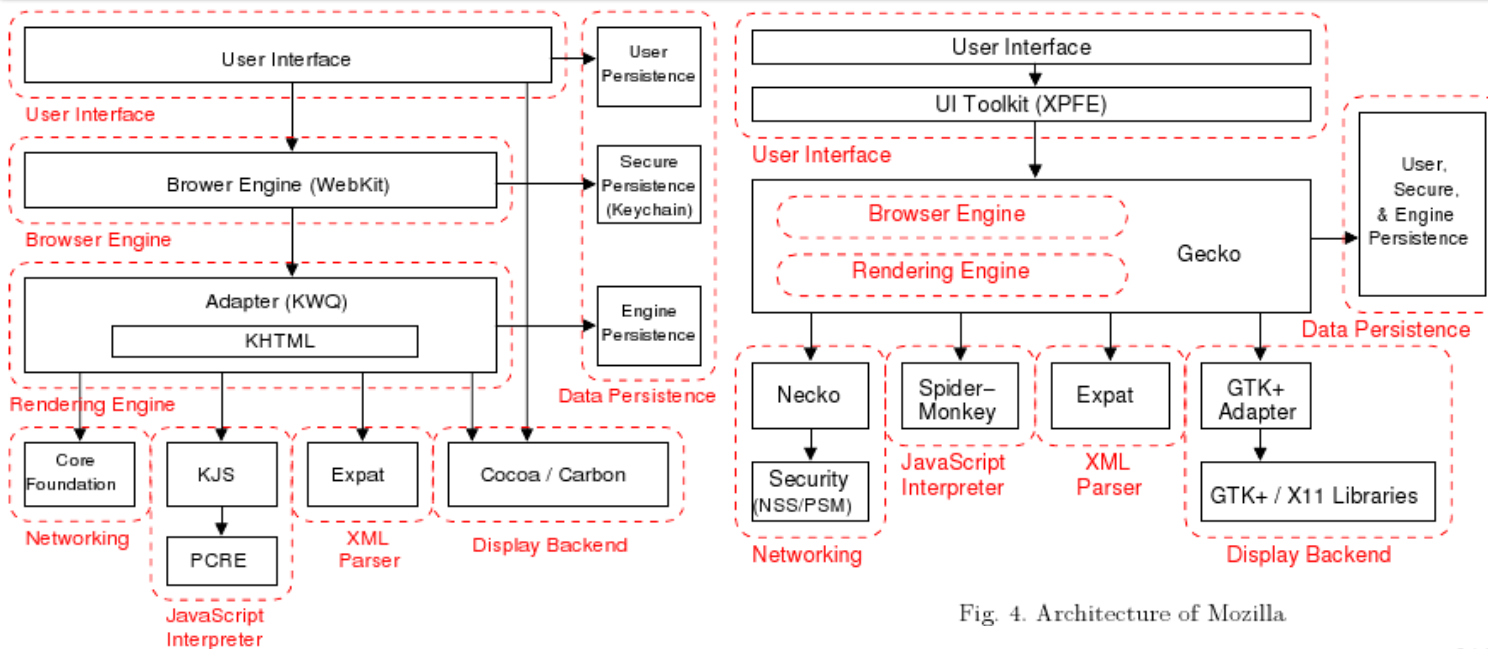


Fig. 4. Architecture of Mozilla

Fig. 7. Architecture of Safari

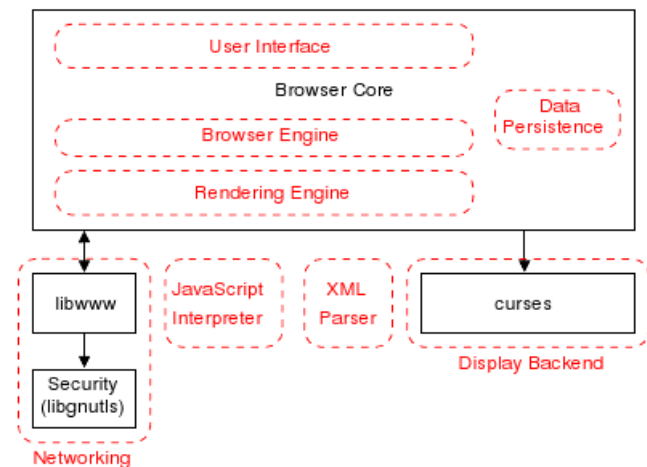
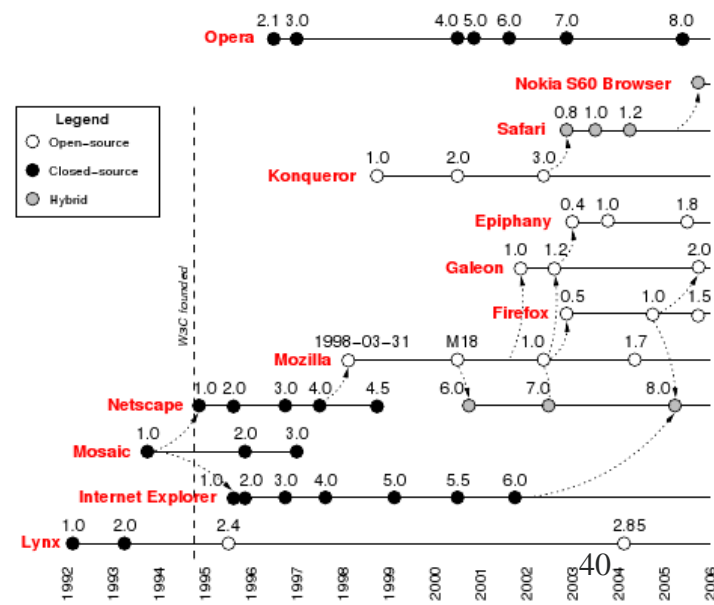


Fig. 8. Architecture of Lynx





Service i **transportlag** protokoller

- **Transportlaget** sørger for kommunikasjon mellom prosesser som kjører på vertsmaskiner («hosts»)
- **TCP**
 - Forbindelsesorientert
 - Pålitelig transport
 - Flytkontroll
 - Metnings-kontroll
 - **Ikke** timing kontroll eller minimumsgaranti for båndbredde
- **UDP**
 - Lettvektsprotokoll uten garantier



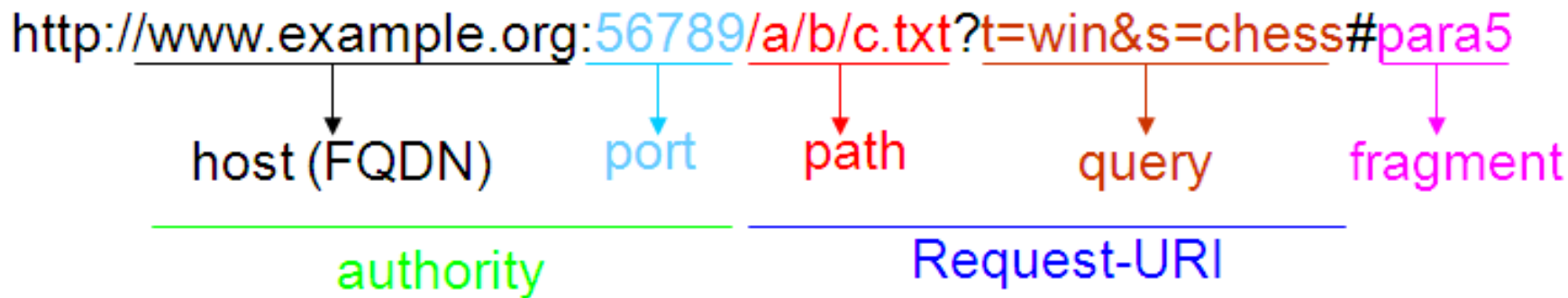
Web-uttrykk

- Web-side
 - Består av "objekter", adresseres av en URI
- Vanligvis har web-siden
 - En base HTML side (index.html), flere objektreferanser
- URI (url) består av
 - protokoll://bruker:passord@vertsnavn:port/filsti/filnavn#anker
?parametre
([protocol://user:pwd@host:port/path?parameters#anchor](#))
 - <http://test.kristiania.no/TK1100/ressursnavn.txt>
- Bruker-agenten på web er en **browser**
 - Netscape, Internet Explorer, Mozilla
- Tjeneren på web kalles web-server
 - Apache, MS IIS





HTTP URL

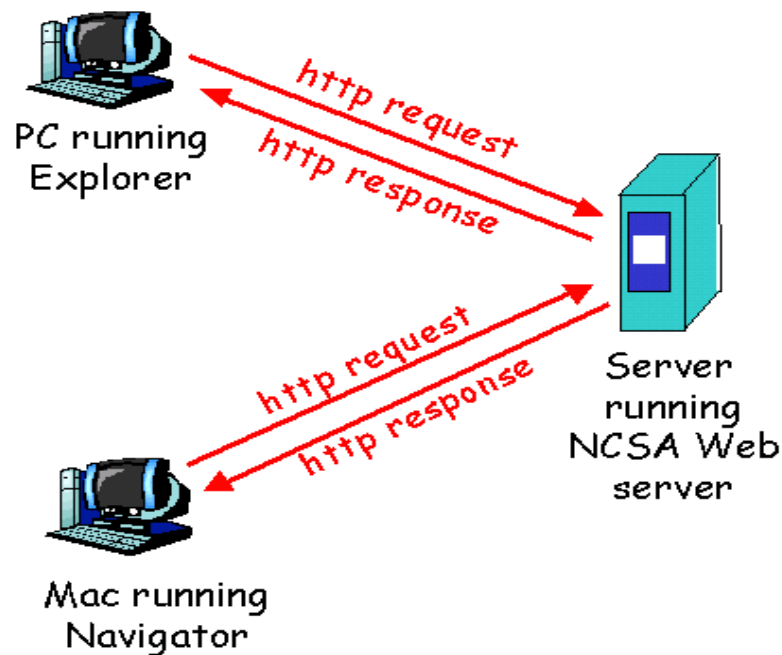


- Browseren foretar et DNS-oppslag og oppretter en TCP-forbindelse til "authority".
- Så følger "filsti" på server (ressurs-ID)
- Etter ? Følger argumenter til script/program
- Etter # typisk et **anker**/posisjon innenfor ressurs ("dokument") (<a href=)



HTTP (HyperText Transfer Protocol)

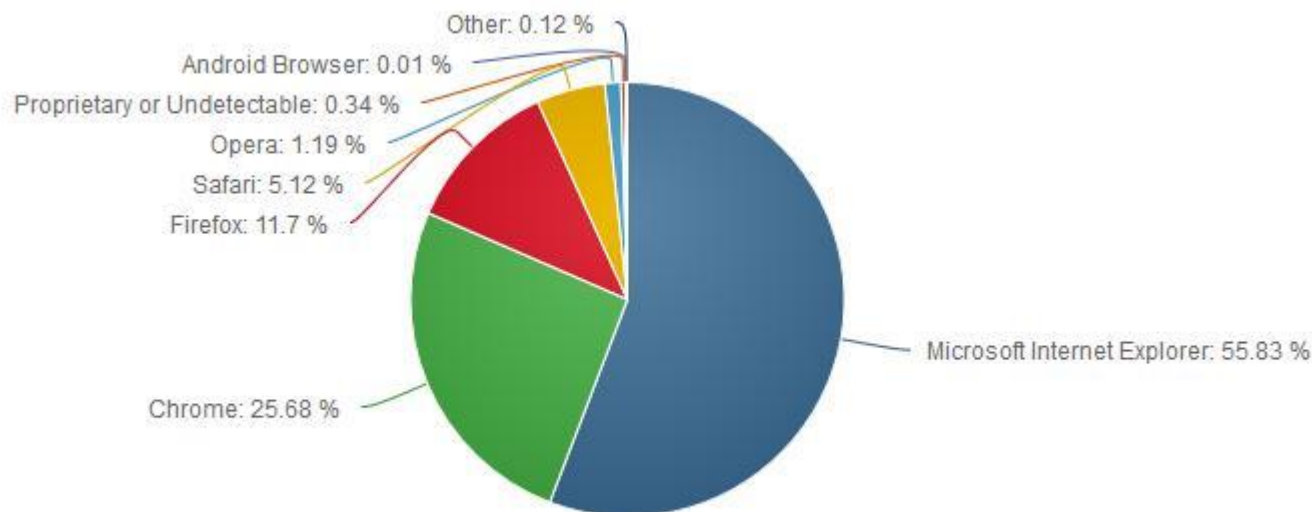
- Webens applikasjons-protokoll
- Klient/tjener modell
 - Klienten spør etter, mottar og viser web "objekter"
 - Tjeneren sender objekter på etterspørsel





Http-klienter

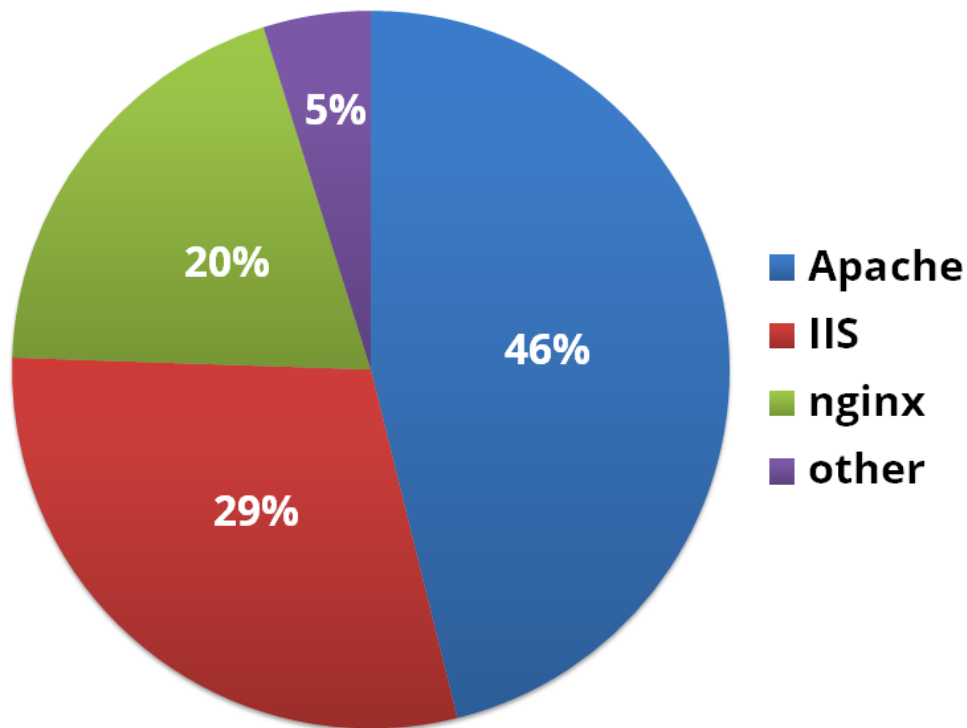
- Utformer og sender http-forespørsler
- Viser frem ("renderer") web-innhold ("layout-engine")
- Kjører script og plug-ins/add-ons...





HTTP-tjenere

- Tar mot og betjener **http-forespørsler**





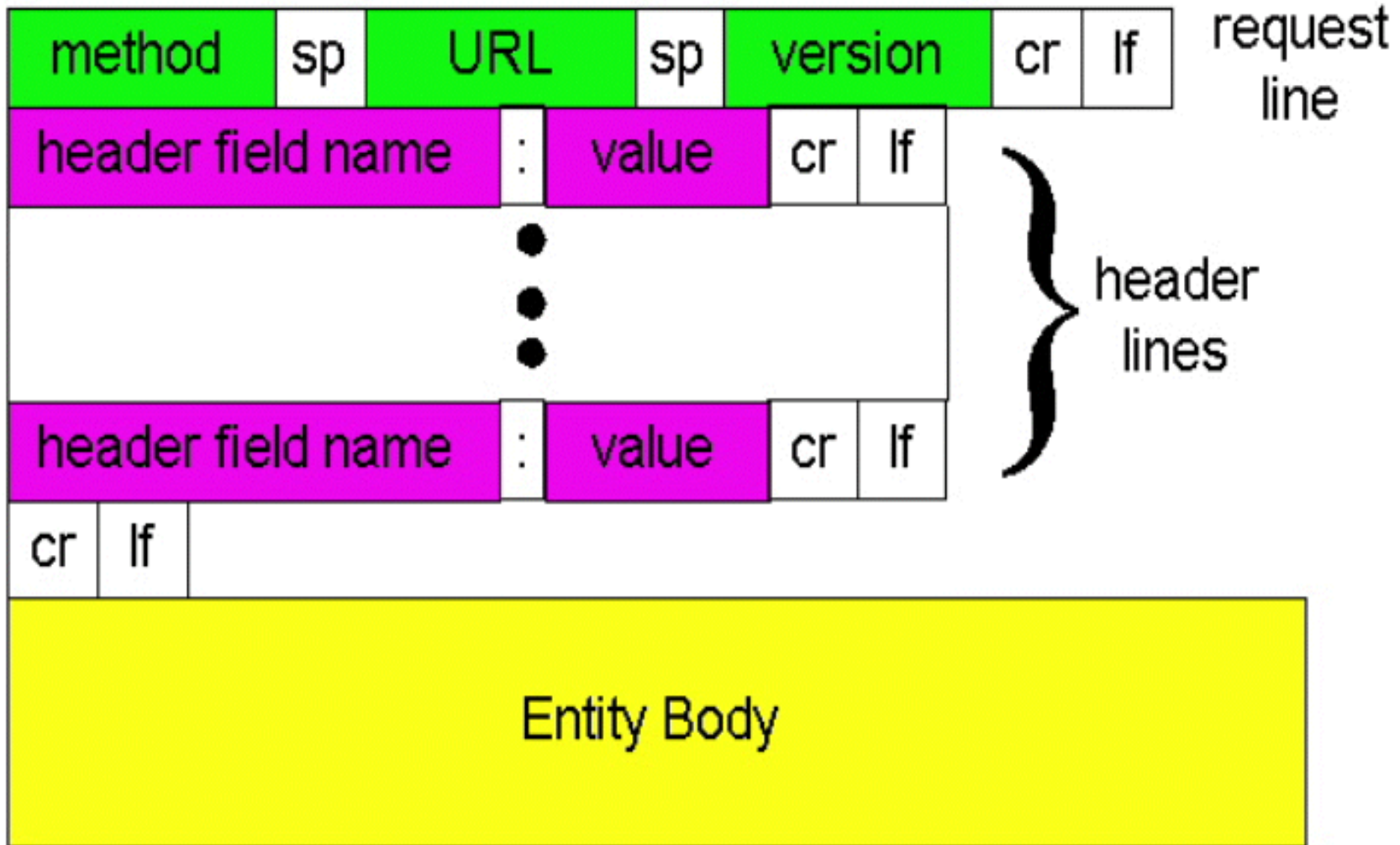
HTTP informasjonsflyt

- Bruker **TCP** transport service
 - Klient setter opp TCP-forbindelsen (lager socket) til tjeneren, port **80**
 - Tjeneren godtar forbindelsen til klienten
 - Klient og tjener utveksler informasjon
 - TCP-forbindelsen lukkes
- HTTP oppbevarer ingen informasjon om tidligere forespørsler (**tilstandsløs** = *stateless*)



HTTP meldingsformat: spørring

- Meldings**headere** er kodet i **7 bit ASCII**-format





HTTP 1.0 meldingsformat

- Spørring

request line
(GET, POST,
HEAD commands)

header
lines

Carriage return
line feed
indicates end
of message

```
GET /somedir/page.html HTTP/1.0
User-agent: Mozilla/4.0
Accept: text/html, image/gif, image/jpeg
Accept-language: fr
```

(extra carriage return, line feed)

- Svar

status line
(protocol
status code
status phrase)

header
lines

data, e.g.,
requested
html file

```
HTTP/1.0 200 OK
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 ...



Typer metoder

HTTP/1.0

- GET
- POST
- HEAD
 - Spør bare server om metainformasjon = headere

HTTP/1.1

- GET, POST, HEAD
- PUT
 - Laster opp en fil til adressen som er spesifisert i URL-feltet
- DELETE
 - Sletter filen som er spesifisert i URL-feltet
- OPTION
- TRACE



HTTP 1.1 Meldingsformat

request line
(GET, POST,
HEAD commands)

GET /somedir/page.html HTTP/1.1

header
lines

Host: www.someschool.edu
User-agent: Mozilla/4.0
Connection: close
Accept-language: fr

Carriage return
line feed
indicates end
of message

(extra carriage return, line feed)

status line
(protocol
status code
status phrase)

HTTP/1.1 200 OK

header
lines

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

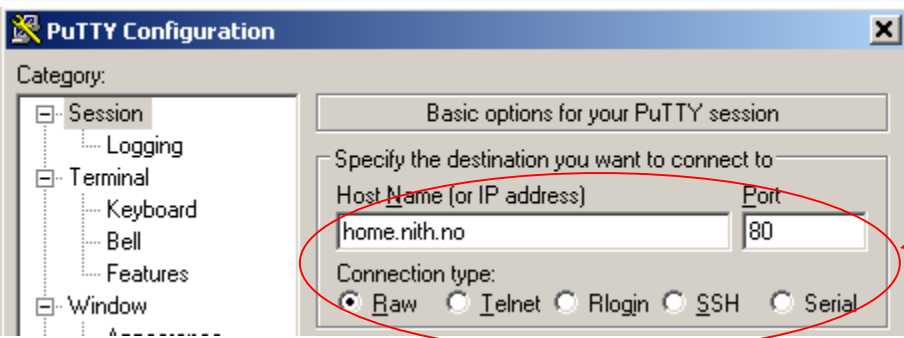
data, e.g.,
requested
HTML file

data data data data data ...

Obligatorisch



Manuell kjøring



Sett opp TCP-forbindelse med WWW-server

GET kommando sendes serveren

```
GET /~blistog/index.html HTTP/1.0
```

```
HTTP/1.1 200 OK
Date: Mon, 13 Oct 2014 11:46:19 GMT
Server: Apache/2.2.22 (Debian)
Last-Modified: Tue, 01 Apr 2014 10:16:38 GMT
ETag: "1ae41f-90c-4f5f876a40d80"
Accept-Ranges: bytes
Content-Length: 2316
Vary: Accept-Encoding
Cache-Control: no-store, no-cache, must-revalidate, pre-check=0, post-check=0, max-age=0
Connection: close
Content-Type: text/html
```

Svar **header**

```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
```

```
"http://www.w3.org/TR/
```

```
xhtml1/DTD/xhtml1-strict.dtd">
```

```
<!-- Dette er HTML-filen jeg benytter som eksempel i TK110 -->
```

```
<html xmlns="http://www.w
```

```
3.org/1999/xhtml">
```

```
<head>
```

```
<meta http-equiv="Content-type" content="application/xhtml+xml;
```

Svar **data**



HTTP svar statuskoder

- Ligger i første linjen på svarmeldingen

- Eksempler:

200 OK

- spørring vellykket, objektet kommer senere i meldingen

301 Moved Permanently

- etterspurt objekt flyttet, ny adresse senere i meldingen

400 Bad Request

- spørring ikke forstått av tjeneren

404 Not Found

- etterspurt dokument/fil ikke funnet på denne tjeneren

505 HTTP Version Not Supported

System:

- Tre siffrs statuskode

1xx = Informational

2xx = Success

3xx = Redirection
(alternate URL is supplied)

4xx = Client Error

5xx = Server Error



HTTP/2

- Mai 2015
- Helt tilbake kompatibel med 1.1
- Hentet mye fra Googles SPDY prosjekt.
- Binær, i stedet for textbasert
- Bedrer ytelse (brukeropplevelse) ved
 - parallellisering,
 - komprimering
 - lar serverne proaktivt «dytte» responser inn i klient-cache
- Støttes foreløpig av ca 2% av Webserverne.

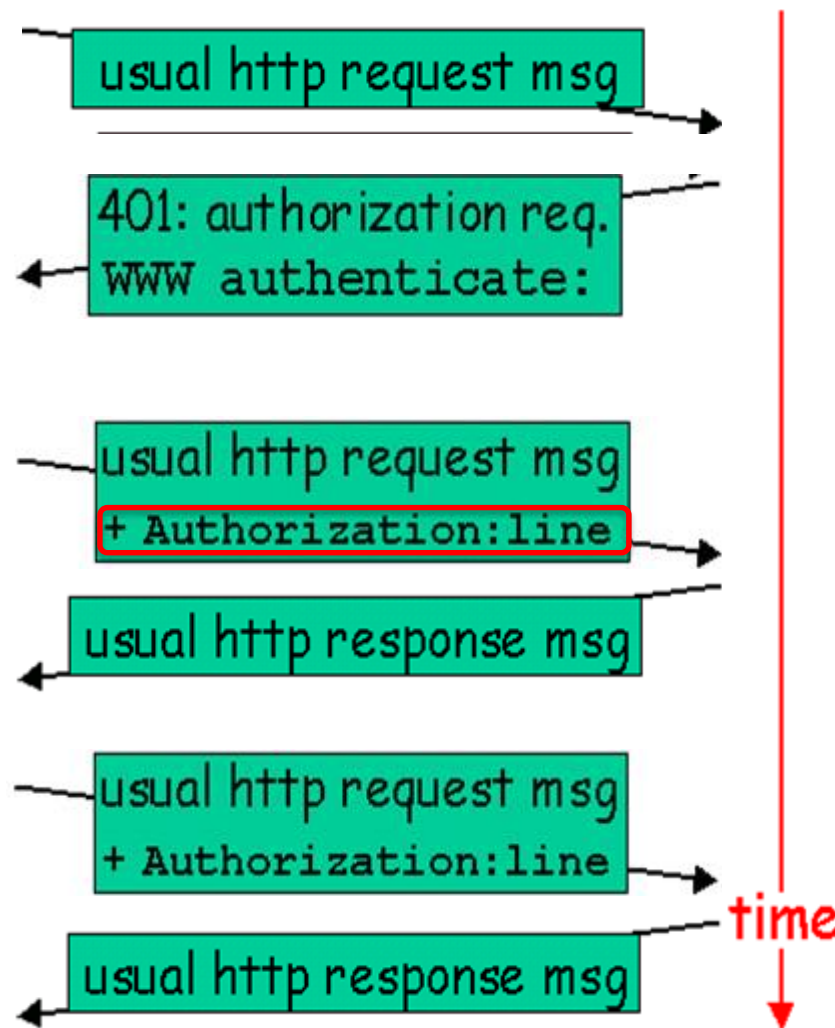


Bruker-tjener kommunikasjon (Ex)

- Ex: Bekreftelse av identitet
 - Navn, passord
 - Ligger i header ved spørring
 - Responsen ligger i svar-header
 - Usikker (**Base 64** koding, ikke kryptering)
- Vanligvis (stateless) må dette gjentas ved hver spørring, men **browseren lagrer navn, passord** for brukeren slik at det bare ses en gang

KLIENT

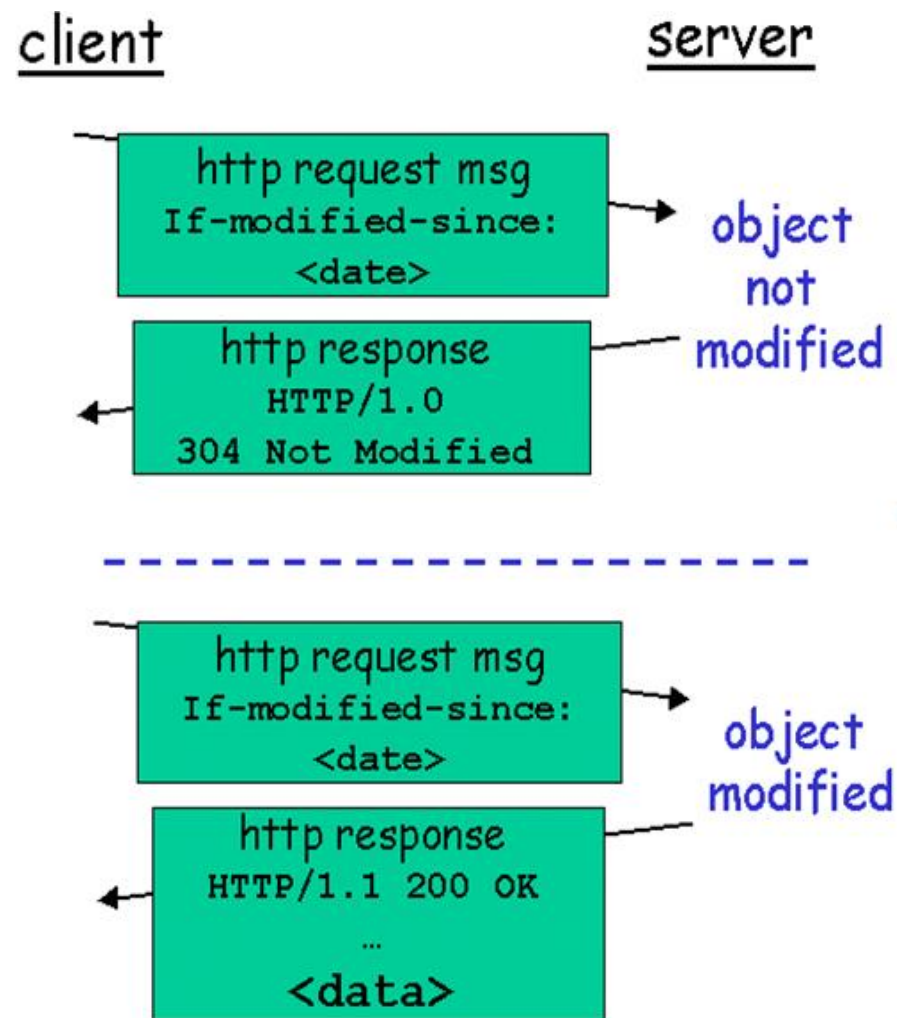
TJENER





Klient-tjener kommunikasjon (Ex)

- Betinget GET
 - If-modified-since:
 - Ikke send svar hvis klient har oppdatert versjon
 - Sjekker tidsstempelet på filen
- Klient
 - Spesifiser dato for **cachet** fil
- Tjener
 - Statuskode **304** dersom ikke oppdatert
- Header-felter som Etag og Vary har Proxy-bruk å gjøre...





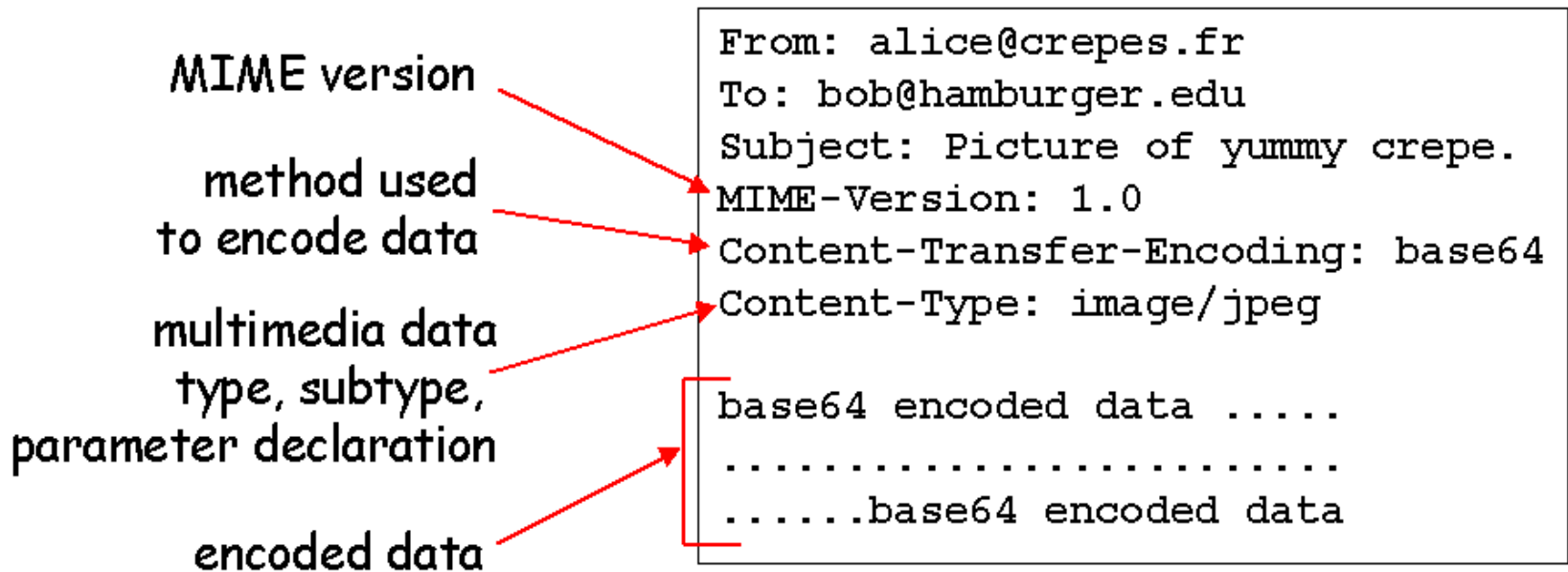
Beholde tilstanden med cookie

- HTTP er en **tilstandsløs** protokoll
 - Fra serveren og klientens perspektiv er alle forespørsler fullstendig uavhengige av hverandre
- Mange Web-steder benytter **cookies**
- En cookie har 3/4 hoved-elementer
 - Cookie **header** linje i http-**responsen**
 - Cookie **header** linje i http-**forespørselen**
 - (Cookie fil som ligger hos klienten)
 - Database over cookies hos tjeneren
- Cookie kan
 - Bevare tilstand
 - "Huske" autorisasjoner og settinger



MIME (Multipurpose Internet Mail Extensions)

- Tilleggslinjer i header for **MIME** innhold
 - Tekst, bilde, audio, video, applikasjon («binære filer»)
 - Opprinnelig for epost, benyttes også av HTTP-klient/tjener
 - Angir koder for innholdst**type** og hvordan det er (om-kodet) under overføringen





DOMAIN NAMES SYSTEM



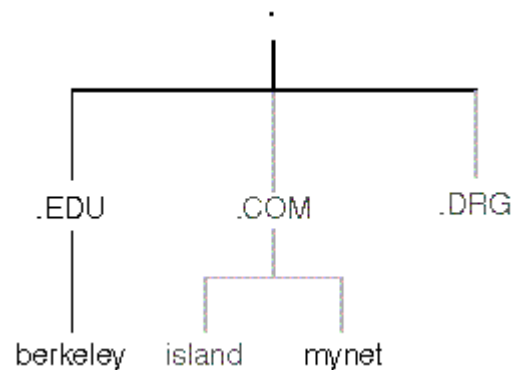
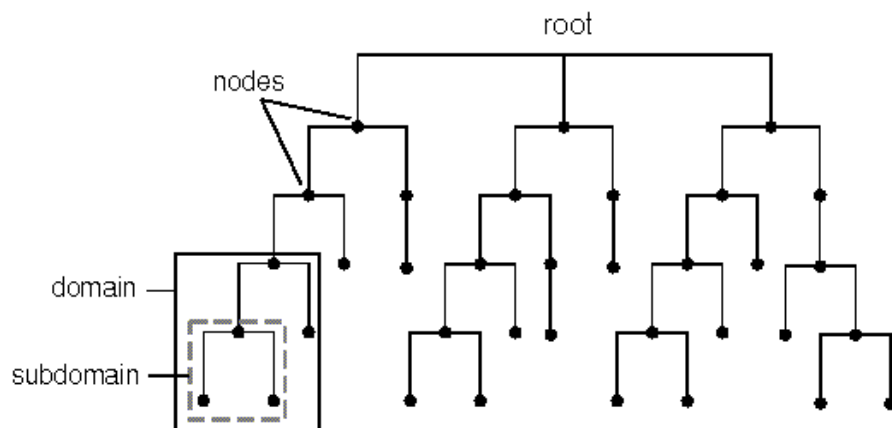
DNS (Domain Name System)

- Mennesker
 - Navn, person-nummer
 - "Ola Nordmann", 161165 42796
- Internett
 - IP-adresse, Navn
 - 10.11.1.21, www.kristiania.no
- Løser dette på Internett med DNS
 - Distribuert database på mange navne-tjenere
 - Protokoll i applikasjons-laget for å knytte navn og IP-adresser



DNS navne-tjenere

- Ikke sentralisert!
 - Unngå at hele nettet går ned med navne-tjeneren
 - Unngå opphopning av trafikk
 - Sentralisert database ligger alltid "langt" vekk
 - Kan skales
- Navne-tjenere fordeles hierarkisk





Hoved (root) navne-tjenere

- Kontaktes av lokale tjenere ved behov
- 13 hoved navne-tjenere





Caching og oppdatering

- Navne-tjenere cacher DNS kartlegging
- Lagring i cache forsvinner etter en tid (timeout)
- Mekanismer for innmelding og oppdatering utvikles hos **IETF** (Internet Engineering Task Force)
 - Dynamisk oppdatering
 - Sikkerhet
 - Mmm
- Navn (og nummere) forvaltes av IANA med underorganisasjoner
 - Europa: RIPE
 - Norge (.no): NorID



DNS records

Distribuert database lagrer RR (resource records)

RR format: navn, verdi, type, ttl

- Type=**A**
 - Navn=vertsnavn, verdi=IP-adresse
- Type=**NS**
 - Navn=domene, verdi=IP-adresse til navne-tjener
- Type=CNAME
 - Navn=alias, verdi=virkelig navn
- Type=**MX**
 - Navn=alias, verdi=post tjener



Win: nslookup

```
C:\Windows\system32\cmd.exe - nslookup
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

C:\Users\blislog>nslookup
Default Server: obelix.ad.nith.no
Address: 158.36.131.10

> ?
Commands: (identifiers are shown in uppercase, [] means optional)
NAME          - print info about the host/domain NAME using default server
NAME1 NAME2    - as above, but use NAME2 as server
help or ?      - print info on common commands
set OPTION     - set an option
    all        - print options, current server and host
    [no]debug  - print debugging information
    [no]d2     - print exhaustive debugging information
    [no]defname - append domain name to each query
    [no]recurse - ask for recursive answer to query
    [no]search - use domain search list
    [no]vc     - always use a virtual circuit
    domain=NAME - set default domain name to NAME
    srchlist=N1[/N2/.../N6] - set domain to N1 and search list to N1,N2, etc.
    root=NAME  - set root server to NAME
    retry=X    - set number of retries to X
    timeout=X  - set initial time-out interval to X seconds
    type=X     - set query type (ex. A,AAAA,A+AAAA,ANY,CNAME,MX,NS,PTR,SOA,SRV)
    querytype=X - same as type
    class=X    - set query class (ex. IN (Internet), ANY)
    [no]mxsfr  - use MS fast zone transfer
    ixfrver=X  - current version to use in IXFR transfer request
server NAME    - set default server to NAME, using current default server
ls server NAME - set default server to NAME, using initial server
root          - set current default server to the root
ls [opt] DOMAIN [> FILE] - list addresses in DOMAIN (optional: output to FILE)
    -a        - list canonical names and aliases
    -d        - list all records
    -t TYPE    - list records of the given RFC record type (ex. A,CNAME,MX,NS,PTR etc.)
view FILE      - sort an 'ls' output file and view it with pg
exit          - exit the program

>
```

nslookup
kan også
brukes i
OSX og Linux,
men der
foretrekkes
dig



AVSLUTTNING



Hva skal vi kunne?

- Definere Internett
- TCP/IP-modellen
 - Hvilke oppgaver løses på de ulike lagene
- Klient/Tjener-modellen
- HTTP
 - Oppbyggingen av URI og URL
 - Request/response
 - GET og ulike typer responskoder
 - Kort om tilstandsløshet og Cookies
- DNS
 - Hvordan organisert: TLD og NS-servere



Dagens øving

- Finn artikler om ARPA Net (helst flere enn den jeg har linket til), og skriv 1 A4 side med dine egne ord om emnet
- Øving T06.00 og T06.01

