

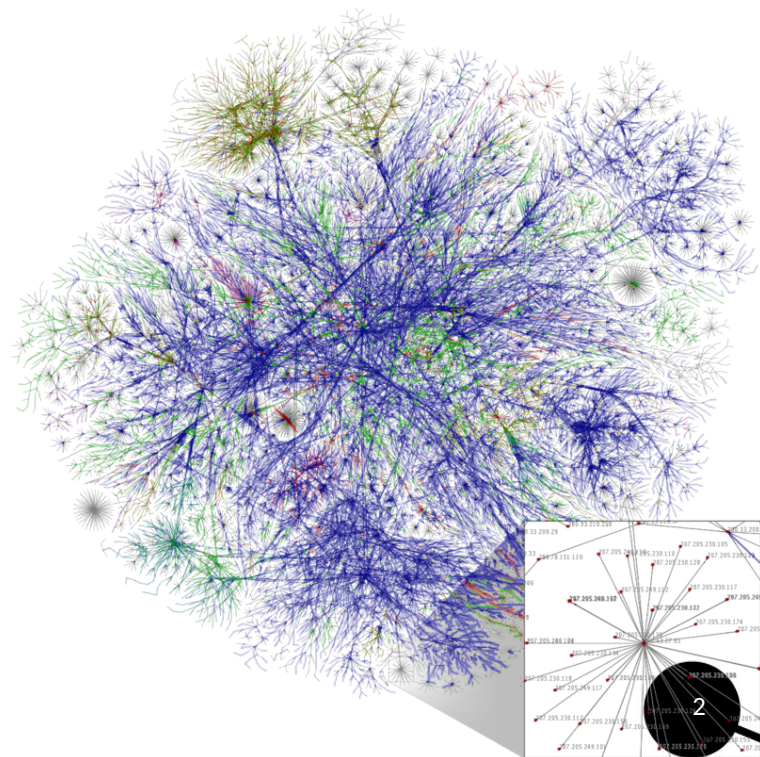


Splet: infrastruktura in arhitektura

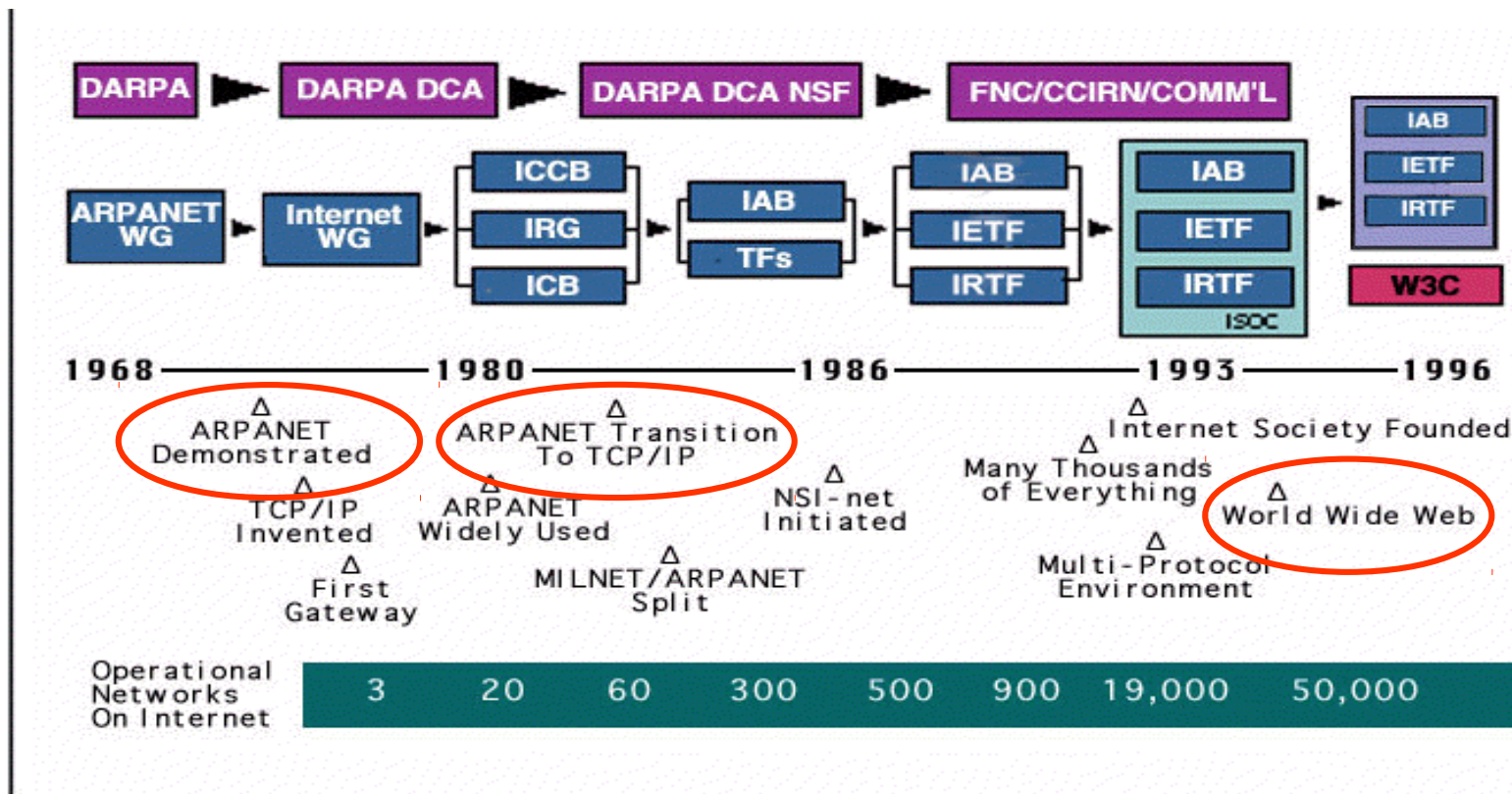


Razvoj interneta

- V 60. letih 20. stoletja ameriški *Department of Defense* začne z razvojem omrežja ARPAnet (Advanced Research Projects Agency Network)
 - prvo vozlišče je bilo postavljeno leta 1969 v UCLA
 - omogoča tekstovno bazirano komunikacije
 - na voljo samo ustanovam s pogodbo z ARPA
- BITNET, CSNET (1970, 1980)
- TCP/IP postane standarden protokola za ARPANET 1982
- NSFnet (National Science Foundation Network, 1986)
 - leta 1990 nadomesti ARPAnet za civilno uporabo
 - leta 1992 omrežje povezuje več kot milijon uporabnikov
 - današnji Internet
- Kaj je Internet, kako deluje?
 - protokol: TCP/IP (UDP/IP)
 - naslovi računalnikov/imena računalnikov



Časovnica razvoja Interneta



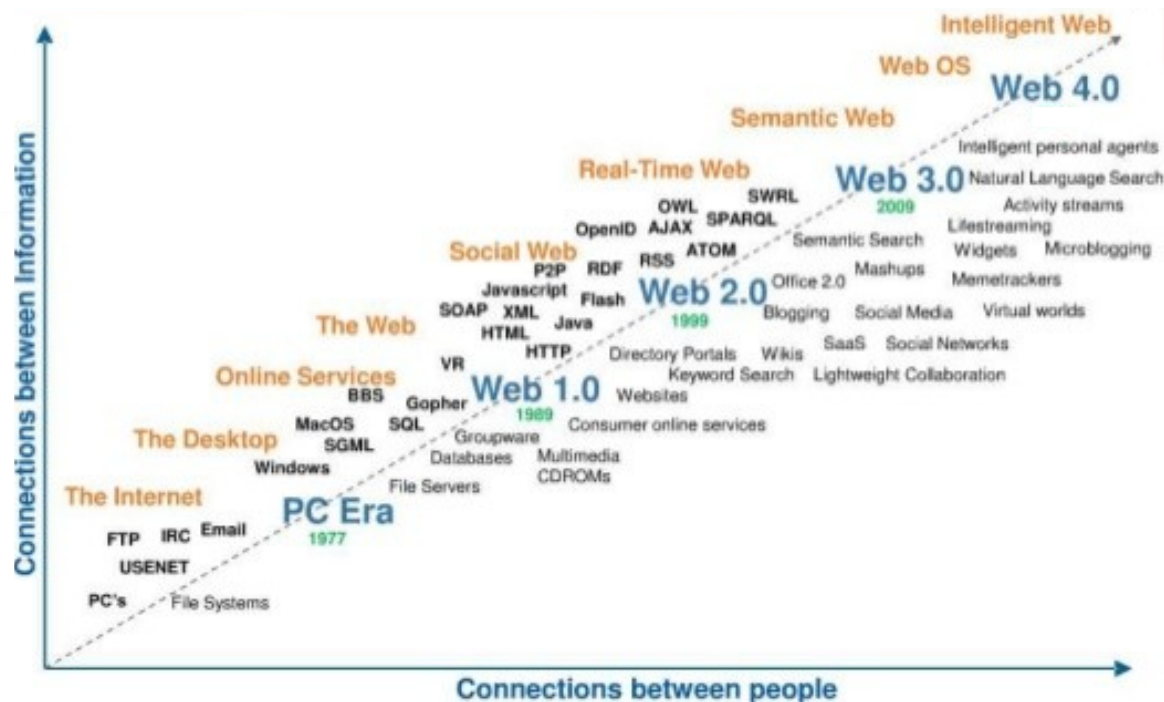


Razvoj spleta

- Splet, svetovni splet, mrežni splet (The World Wide Web)
 - 1989: Tim Berners-Lee in skupina
 - za potrebe znanstvenega komuniciranja
 - protokol HTTP
 - sistem za dostop do dokumentov pri uporabi protokola
- Sistem imenovan hipertekst (nadbessedilo, ...) je uporabljen za dokumente
 - besedilo z vsebovanimi povezavami na druge dokumente
 - omogoča nesekvenčno pregledovanje tekstovnega materiala
- Splet je ogromna zbirka dokumentov
 - dokumenti so povezani preko povezav
 - brskalniki se uporabljajo za dostop do dokumentov
 - dostop do dokumentov se vrši preko URL-jev



Časovnica razvoja spleta





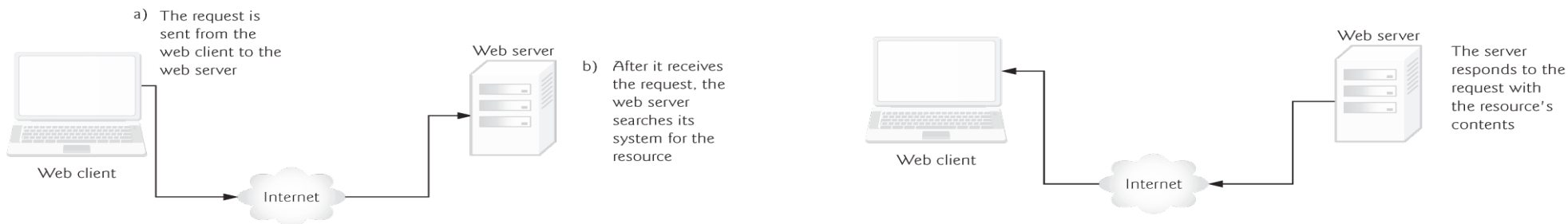
Zakaj programiramo za splet?

- Moorov zakon
 - eksponenten porast kapacitete tehnologij in hkrati tudi eksponenten padec cen komponent
 - porast mobilnega mrežnega prometa
 - porast tehnologij za splet

Web 3.0

radio računanje v oblaku prenosi multimedije porazdeljeni sistemi
lokacijske storitve bančništvo IP telefonija
e-mail IP televizija
socialna omrežja takošnje sporočanje spletne trgovine
prenosi datotek videokonference iskalniki
spletne igre

Arhitektura odjemalec/strežnik



- odjemalec **zahteva** storitev, strežnik odgovori na zahtevo (**streže**)
- **odjemalec** uporablja spletnega odjemalca, ki je ponavadi ... ?
- **strežnik** odgovori na zahteve odjemalca, zagotavlja storitve
 - **storitev** je naloga, ki jo izvede komponenta na strani strežnika
 - opredelitev zahtevka
 - protokol, vrata
- kdo ima težje delo?



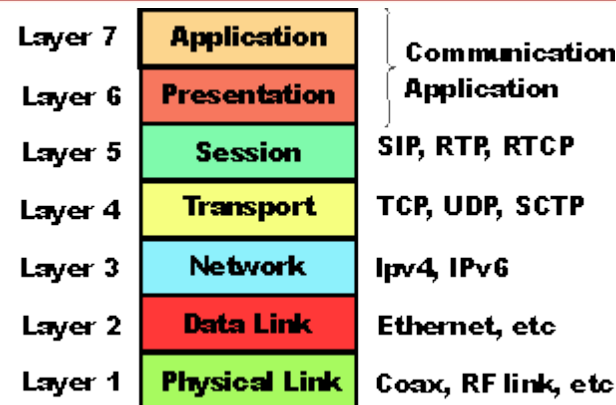
Komunikacija

- komunikacija med procesi
- na najnižjem nivoju gre za pošiljanje sporočil po omrežju
- izmenjava informacij → protokoli
- komunikacija med procesi
 - prvi proces zgenerira sporočilo na svoji strani
 - nato izvede sistemski klic
 - operacijski sistem pa pošlje sporočilo drugemu procesu
 - da je komunikacija mogoča mora obstajati dogovor o formatu → pomen bitov, podatkovni tipi, veljavnost sporočila, ...

Protokol

- protokol je zbirka pravil, ki določajo način prenosa sporočil in določajo interpretacijo vsebine sporočil
- strežnik in odjemalec se morata dogovoriti o protokolu, »o jeziku, ki ga govorita« → rokovanje (*handshake*)
- protokole določimo na različnih nivojih: OSI model

1. fizični nivo → telefonsko omrežje, Wi-Fi, modem, ...
2. podatkovni nivo → detekcija napak, kontrola toka na fiz. povezavi
3. omrežni nivo → Internetni Protokol (IPv4, IPv6, NAT, ...)
4. transportni nivo → TCP, UDP, ...
5. sejni nivo → zagotavlja upravljanje s sejami (SSH, RPC, ...)
6. predstavitveni nivo → SSL
7. aplikacijski nivo → pogovarjanje med aplikacijami (HTTP, FTP, Telnet, ...)



Layer Functions

Application	User interface
Presentation	Data formatting
Session	Establish & maint connect
Transport	TCP - Accurate data
Network	IP - Routers
Data Link	MAC - Switches
Physical	Signals - Cables



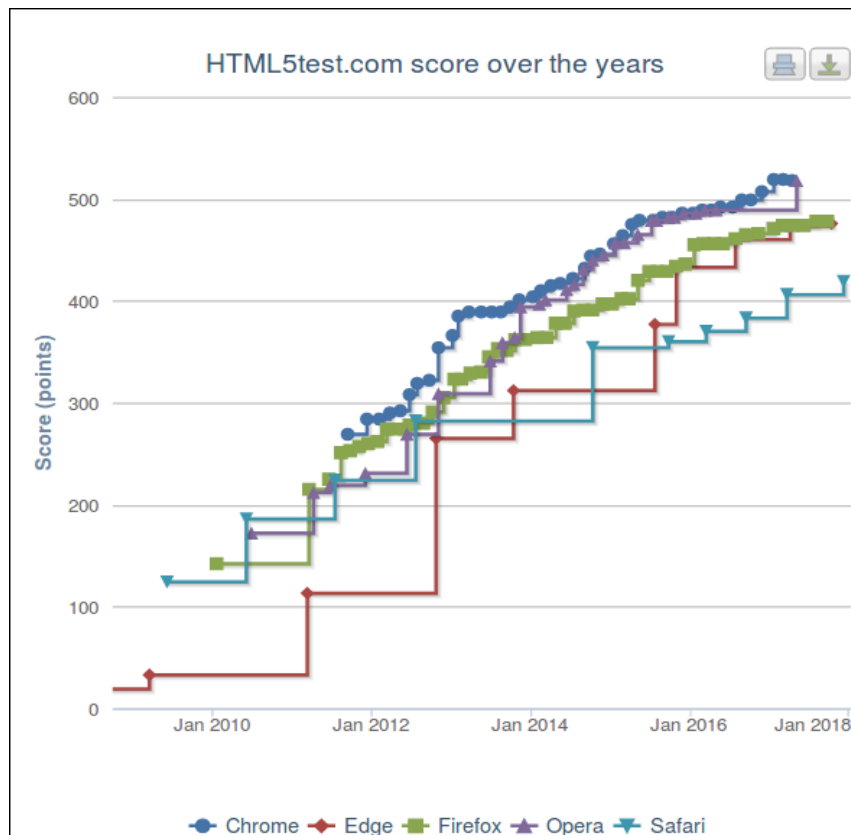
Spletni odjemalci (UA) – brskalniki

- uporabljajo se za komunikacijo s strežnikom
- brskalnik je odjemalec, saj sproži komunikacijo s strežnikom
 - statični dokumenti
 - dokumenti, ki zahtevajo uporabnikov vnos
 - programi, ki se izvedejo na strežniku
- za komunikacijo med odjemalcem in strežnikom se ponavadi uporablja protokol HTTP
- uporabljajo se različni brskalniki različnih proizvajalcev



Spletni odjemalci (UA) – brskalniki

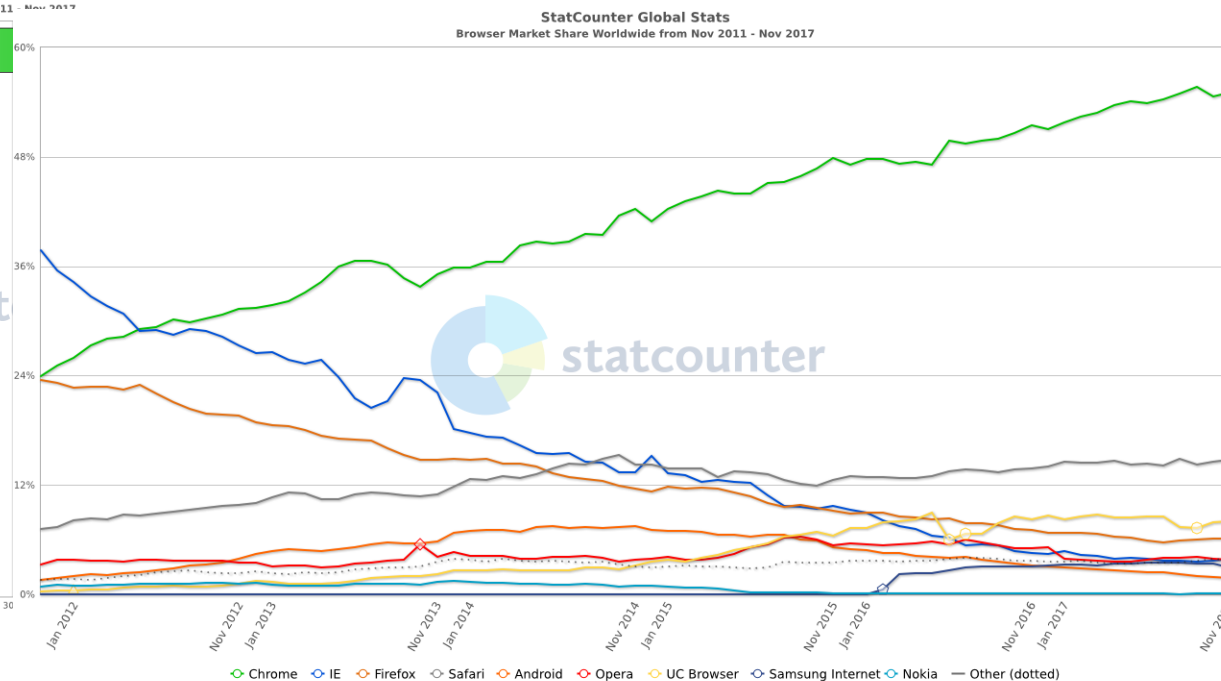
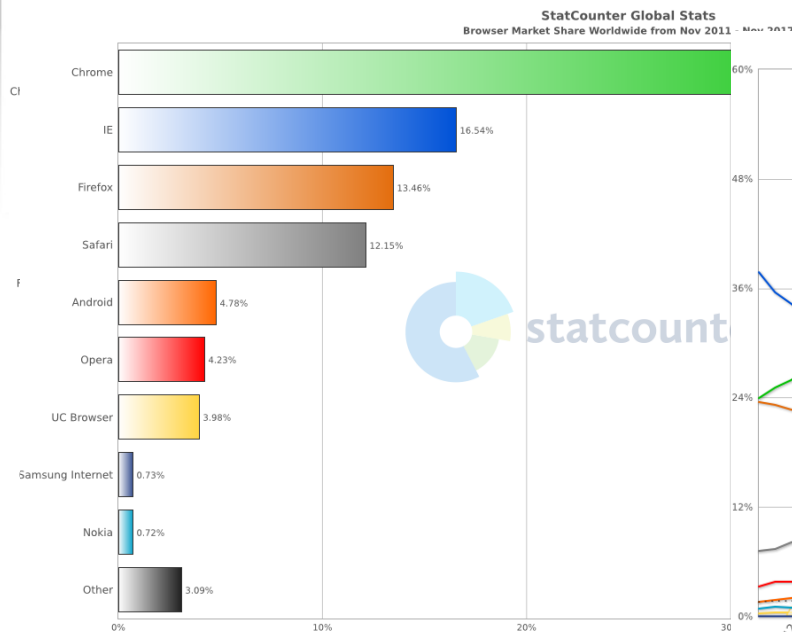
- izziv: zagotavljanje konsistentnega izgleda strani v vseh brskalnikih
IN različicah, ni standardov
- testi:
 - <http://acid1.acidtests.org>
 - <http://acid2.acidtests.org>
 - <http://acid3.acidtests.org>





Spletni odjemalci (UA) – brskalniki

→ tržni delež posameznih brskalnikov (gs.statcounter.com)





Spletni strežniki

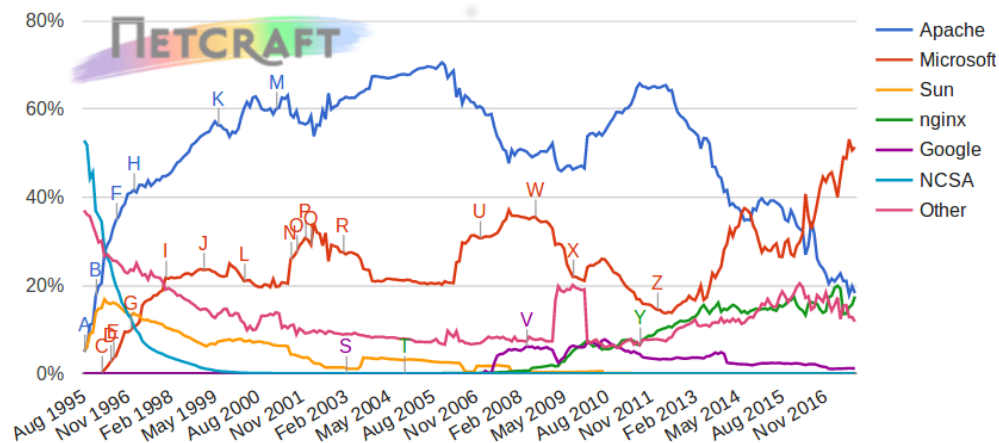
- strežniki so ponavadi programi na oddaljenih računalnikih, ki zagotavljajo odgovore odjemalcem
- streženje dokumenta ni časovno potratno
 - zato lahko vsak strežnik streže več odjemalcem
- odjemalec začne komunikacijo s »pošiljanjem URL-ja«
 - URL lahko določa naslov datoteke na strežniku
 - URL lahko določa program na strežniku



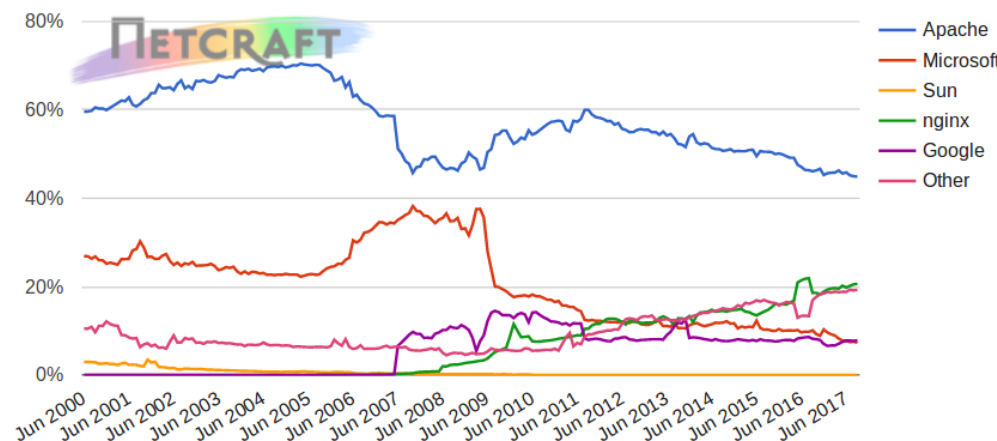


Spletni strežniki

Web server developers: Market share of all sites



Web server developers: Market share of active sites

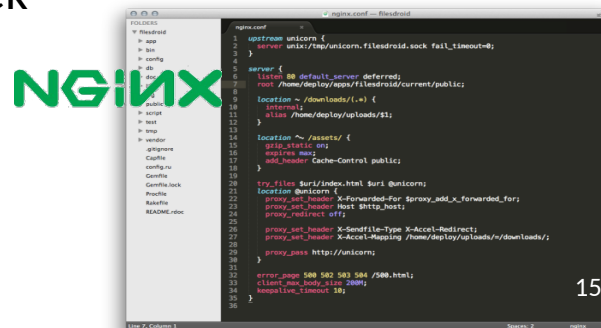
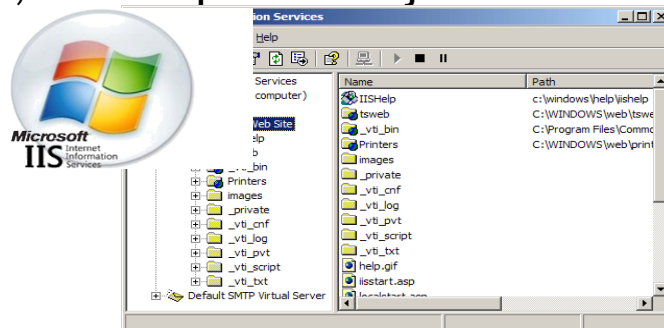


Developer	August 2017	Percent	September 2017	Percent	Change
Microsoft	912,072,085	50.65%	927,540,454	51.39%	0.73
Apache	356,768,928	19.81%	329,105,832	18.23%	-1.58
nginx	280,659,498	15.59%	315,530,746	17.48%	1.89
Google	20,680,107	1.15%	20,906,849	1.16%	0.01

Developer	August 2017	Percent	September 2017	Percent	Change
Apache	77,916,218	44.99%	77,487,531	44.89%	-0.10
nginx	35,566,579	20.54%	35,640,320	20.65%	0.11
Google	13,376,490	7.72%	13,561,655	7.86%	0.13
Microsoft	13,136,286	7.58%	12,629,582	7.32%	-0.27

Spletni strežniki

- sistemska mapa (ServerRoot) – konfiguracija strežnika, moduli
- dokumentna mapa (DocumentRoot) – dokumenti za dano domeno, več domen na strežnik
- popularni strežniki:
 - Apache: nadgrajeni HTTPd, konfiguracijska datoteka, najpopularnejši strežnik od leta 1996
 - IIS: Microsoft, konfiguracija preko uporabniškega vmesnika, del družine Windows NT od verzije 4.0 dalje
 - nginx: razvijan od leta 2002, skromen s porabo resursov, enostavno skaliran tudi na minimalni strojni oprema, odličen pri streženju statičnih datotek



Protokol HTTP

- način za prenos informacij oziroma podatkov na spletu
- protokol **HTTP**, **verzija 1.1**, definirana v RFC 2616, verzija 2.0 izdana 2015, 24 % spletnih strani podpira HTML/2
- deluje na način **zahteva → odgovor**
- gre za pošiljanje sporočil med strežnikom in odjemalcem
 - ko se na odjemalcu zgodi dogodek se zgenerira sporočilo z zahtevo HTTP
 - pridobi se naslov strežnika (IP številka preko Domain Name Server – DNS)
 - vzpostavi se povezava TCP s strežnikom
 - preko TCP povezave se pošlje sporočilo
 - strežnik sprejme sporočilo in sprocesira zahtevo
 - strežnik vrne odgovor odjemalcu
 - odjemalec sprejme odgovor in ustrezno posodobi stran

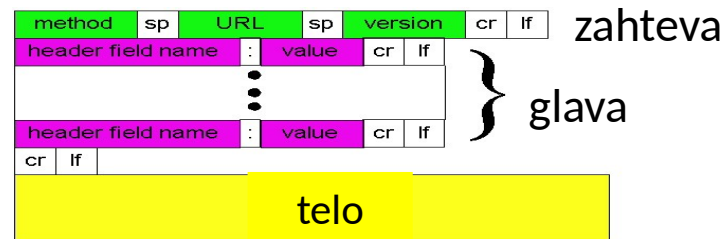




Prenos resursov – komunikacija 1/3

- protokol **HTTP**, verzije 1.1
- odjemalec pošlje **zahtevo HTTP**
- splošna oblika zahteve:
 - HTTP metoda, domenski del URL-ja, HTTP verzija
 - polja glave
 - prazna vrstica
 - telo sporočila
- **HTTP zahteva:**

```
GET /storefront.html HTTP/1.1
Host: unknown.fri.uni-lj.si
Connection: close
...
```





Prenos resursov – komunikacija 2/3

→ HTTP metode

→ štiri kategorije polj glave

- General – splošne informacije kot je na primer datum
- Request – vključen v glavah zahtevkov
- Response – vključen v glavah odgovorov
- Entity – vključen v glavah zahtevkov in odgovorov

→ header fields (Request):

- Accept: text/*
- If-Modified-Since: date

METODA	Opis
GET	Vrne vsebino zahtevanega dokumenta
HEAD	Vrne samo HEADER zahtevanega dokumenta
POST	Posreduje podatke, ki bodo dodani dokumentu ali zbirki
PUT	Shrani vsebino sporočila na strežnik
DELETE	Zbriše zahtevan dokument
OPTIONS	Vrne seznam metod za dostop do vira
TRACE	Vrne kopijo celotnega zahtevka HTTP → testiranje
CONNECT	Za ustvarjanje nekaterih tipov varnih povezav



GET in POST metodi

- ▶ GET: parametri so posredovani za naslovom

primer: <https://www.google.si/search?q=spletne+tehnologije>

```
GET /blog/?name1=value1&name2=value2 HTTP/1.1  
Host: carsonified.com
```

- ▶ POST: parametri so posredovani v zahtevku HTTP

```
POST /blog/ HTTP/1.1  
Host: carsonified.com  
name1=value1&name2=value2
```

Metodi GET in POST - primerjava

	GET	POST
Zgodovina	Ostane v brskalniku	Se ne shrani v brskalniku
Zaznamki	Lahko se shranijo	Se ne da shraniti
Gumb nazaj	Ponovno izvede/medpomnilnik	Ponovno izvede – obvestilo
Parametri	Lahko, omejeni s tem, kar lahko damo v URL (najvarneje do 2K)	Lahko pošilja, vključno z datotekami
Vdiranje	Enostavnejše	Težje
Omejitve tipov	Samo ASCII znaki	Ni omejitev, tudi binarni podatki
Varnost (zasebnost)	Manj varno, podatki so del URL, ki se shrani kot tekst	Bolj varno, ker se podatki ne shranjujejo
Omejitve dolžine obrazca	Omejeni z dolžino vrstice, IE dovoljuje do 2048 znakov	Ni omejitev
Uporaba	Za idempotentne dokumente, ne pri pošiljanju občutljivih informacij	Pri pošiljanju občutljivih informacij, pri spreminjanju podatkovnih baz
Shranjevanje	Se lahko shrani	Ni shranjevanja
Vidljivost	Vidna je vsem, prikazana v naslovni vrstici	Vrednosti spremenljivk niso prikazane



Prenos resursov - komunikacija 3/3

→ strežnik odgovori z odgovorom HTTP (response):

```
HTTP/1.1 200 OK
Date: Tue, 04 Oct 2011 21:54:03 GMT
Server: Apache
Last-modified: 18 May 2004 16:38:38GMT
Content-length: 398
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Content-Type: text/html; charset=utf-8
```

version	sp	status code	sp	phrase	cr	lf	
header name		:	value			cr	lf
.							
.							
header name		:	value			cr	lf
cr	lf						
telo (opcijsko)							

status
glava

→ polja glave:

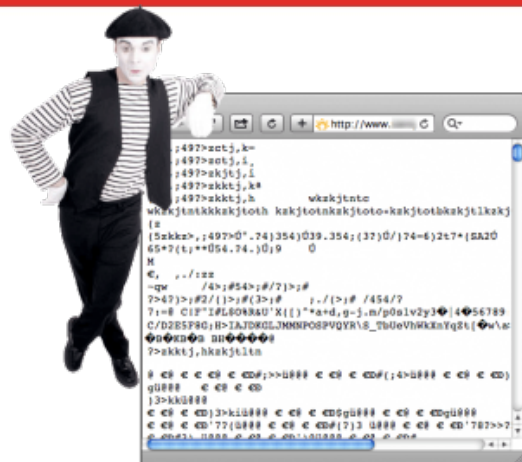
- Content-Length: <value>
- Content-Type: text/html

→ status:

Koda	Kategorija
1xx	Informativno, začasni odgovor (status)
2xx	Uspeh, zahtevek sprejet, razumljen in obdelan
3xx	Preusmeritev, odjemalec mora izvesti dodatne akcije
4xx	Napaka na strani odjemalca, ni datoteke, ni pravic, ...
5xx	Napaka na strani strežnika, strežnik ni izvedel veljavne zahteve

Opredelitev vrste resursa

- odjemalec potrebuje način za ugotavljanje tipa resursa
- tip resursa specificira strežnik v odgovoru
- **specifikacija MIME** (Multipurpose Internet Mail Extensions)
 - oblika specifikacije MIME : type/subtype
primeri: text/plain, text/html, image/gif, image/jpeg
 - če specifikacija ni znana, potem strežnik (včasih pa tudi odjemalec) določi tip z uporabo končnice
- brskalniki tudi ne morejo poznati vseh tipov/podtipov resursov → eksperimentalni tipi MIME
 - začnejo se s predpono x. (npr. video/x.mojvideo) ali x-
 - redka uporaba, standard odvrča pred tako uporabo, uporaba predpone za podtype pa je zastarela, tak tip naj bi se registriral, če je mogoče
 - strežnik mora zagotoviti tudi aplikacijo (pomožno aplikacijo, vtičnik) tako, da lahko odjemalec rokuje z vsebino odgovora



Varnost spletnih aplikacij

- ▶ varnost na strani strežnika
 - katerikoli odjemalec na spletu lahko posreduje zahtevek
- ▶ varnost komunikacije:
 - zasebnost
 - integriteta
 - avtentikacija
 - ne-repudiacija (ne-zavrnitev)
- ▶ varnost na strani odjemalca
 - katerikoli strežnik, na katerega se je odjemalec poveza, lahko naloži programsko za izvajanje

