

# Web-technológia I.

A Web története

Legfontosabb tulajdonságok és alkalmazott technológiák  
áttekintése

Hatwágner F. Miklós

Széchenyi István Egyetem, Győr

2015. szeptember 10.

# A hiperszöveg (hypertext) és a web születése

- Nagy mennyiségű információ rendszerezése: szótárak, enciklopédiák, lexikonok
- **Vannevar Bush**
  - „**As We May Think**” → agy működése, Atlantic folyóirat, 1945
  - **Memex**
    - Üveg borítás, erre vetítené alulról a képet, szöveget
    - Mechanikus szerkezettel kereshető és vihető be az információ
    - Az információk összekapcsolhatóak
    - Hiperszöveg, hiperhivatkozás (hyperlink)
    - Korlátlan kapacitás



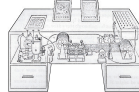
(a) V. Bush



(b) As we  
may...



(c) Memex



(d) Memex

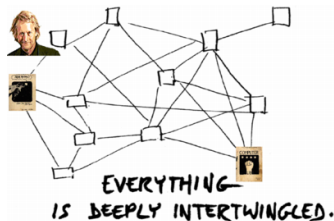
# A hiperszöveg (hypertext) és a web születése

- Theodor Holm Nelson

- hypertext, hypermedia: „Complex Information Processing: A File Structure for the Complex, the Changing, and the Indeterminate”, ACM, 1965
- **Project Xanadu**: 1960 - ?
- „HTML is precisely what we were trying to PREVENT—ever-breaking links, links going outward only, quotes you can't follow to their origins, no version management, no rights management.”



(a) Ted Nelson



(b) Xanadu

# A hiperszöveg (hypertext) és a web születése

- [Douglas Engelbart](#): [NLS](#), 1968 (hiperszöveg, egér, raszteres grafika, ablakkezelés)

Korai alkalmazások:

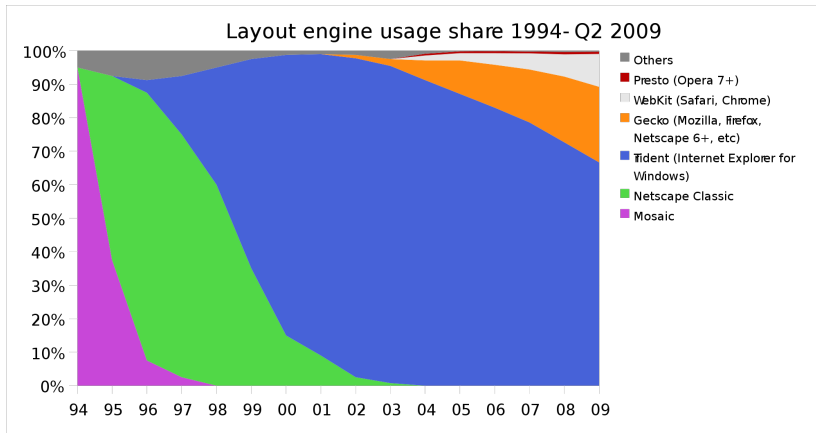
- [Aspen Movie Map](#), MIT, 1977
- Tim Berners-Lee: [ENQUIRE](#), 1980 (Wiki-szerű hiperszöveg adatbázis)
- [Guide](#), 1982 (hiperszöveg személyi számítógépen)
- [Bill Atkinson](#): [HyperCard](#), Apple Inc., 1985

# A hiperszöveg (hypertext) és a web születése

Tim Berners-Lee , R. Cailliau. 12 November 1990, CERN

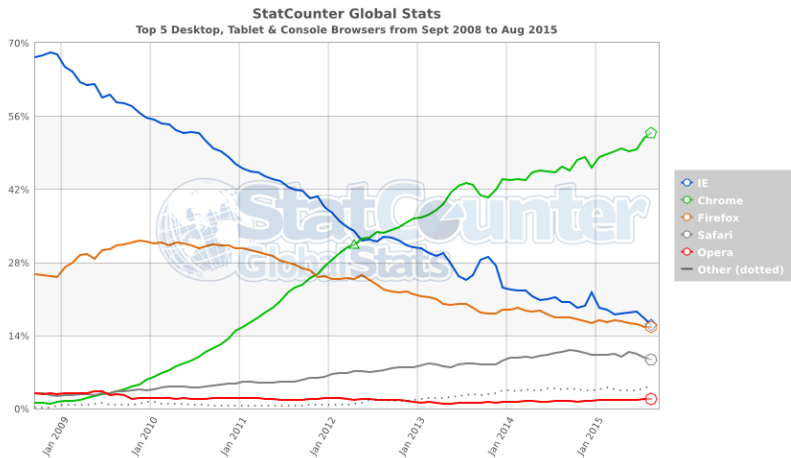
„HyperText is a way to link and access information of various kinds as a web of nodes in which the user can browse at will. It provides a single user-interface to large classes of information (reports, notes, data-bases, computer documentation and on-line help). We propose a simple scheme incorporating servers already available at CERN... A program which provides access to the hypertext world we call a browser...”

- Tim Berners-Lee: [World Wide Web](#) (WWW), 1990
  - 1990, [CERN httpd](#) prototípus implementáció, WYSIWYG böngészés és tartalomszerkesztés (WorldWideWeb, Nexus)
  - 1992, új böngészők: Line Mode Browser, ViolaWWW, Erwise, MidasWWW, ...
  - 1993, NCSA (National Center for Supercomputing Applications) Mosaic
  - 1994, a [W3C](#) megalapítása, Marc Andreessen → Netscape Communications Corporation, Netscape Navigator
  - 1995, Internet Explorer 1.0



Böngészők piaci részesedése az első böngészőháború idején

# Böngésző háborúk



Böngészők piaci részesedése napjainkban

# A Web megoldása

A web legfontosabb technológiái:

- **URL** (Uniform Resource Locator), a hiperhivatkozások megvalósítása
- **HTTP** (HyperText Transport Protocol), a webet felépítő oldalak hálózaton keresztüli továbbítása
- **HTML** (HyperText Markup Language), az oldalak leírására

Kliens-szerver architektúra (**böngésző**, **webkiszolgáló**)



Tim Berners-Lee



# URL: egységes erőforrás-meghatározó

## Az URL felépítése

`scheme://username:password@domain:port/path?query_string#fragment_id`

Részei:

**scheme** Séma, vagy protokoll. Pl. http, ftp, mailto, ...

**username** felhasználónév

**password** jelszó

**domain** a tartomány, annak a gépnek a neve, amin az oldal megtalálható

**port** portszám, szolgáltatás-azonosító (web: 80, 443)

**path** az oldal elérési útja és neve a kiszolgálón (lehet virtuális)

**query\_string** lekérdező karakterlánc, különféle paraméterek átadása

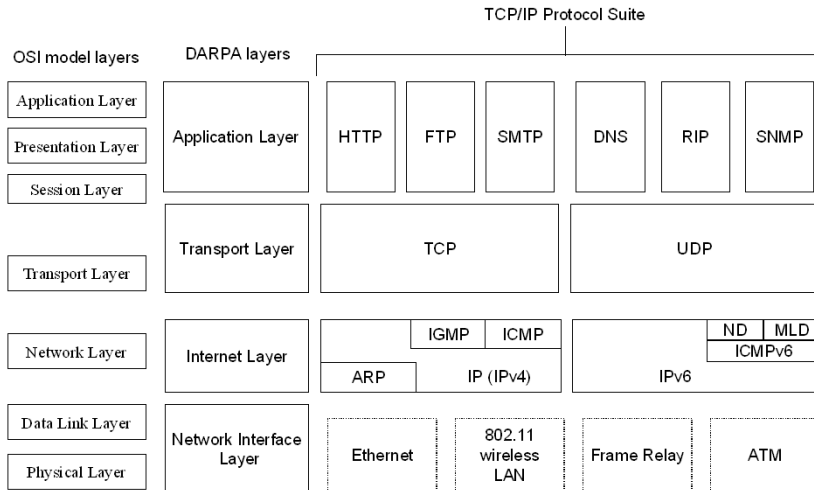
**fragment\_id** oldalon belüli hivatkozás, horgony

További részletek: [RFC3305](#)

# URL: egységes erőforrás-meghatározó

Név	Rendeltetés	Példa
http	Hiperszöveg (HTML)	<a href="http://rs1.sze.hu/~hatwagnf/">http://rs1.sze.hu/~hatwagnf/</a>
https	Hiperszöveg biztonságos átvitele	<a href="https://www.otpbank.hu/">https://www.otpbank.hu/</a>
ftp	Fájlátvitel (FTP)	<a href="ftp://ubuntu.sth.sze.hu/ubuntu/">ftp://ubuntu.sth.sze.hu/ubuntu/</a>
file	Helyi állomány	<a href="file:///home/hatwagnf/WWW/index.html">file:///home/hatwagnf/WWW/index.html</a>
mailto	E-mail	<a href="mailto:miklos.hatwagner@sze.hu">mailto:miklos.hatwagner@sze.hu</a>
rtsp	Valós idejű média letöltés	<a href="rtsp://youtube.com/montypython.mpg">rtsp://youtube.com/montypython.mpg</a>
sip	Multimédia hívás	<a href="sip:eve@adversary.com">sip:eve@adversary.com</a>
about	Böngésző információ	<a href="about:plugins">about:plugins</a>

# HTTP: hiperszöveg-átviteli protokoll



A HTTP protokoll elhelyezkedése az OSI és TCP/IP modellekben

# HTTP 1.x: hiperszöveg-átviteli protokoll

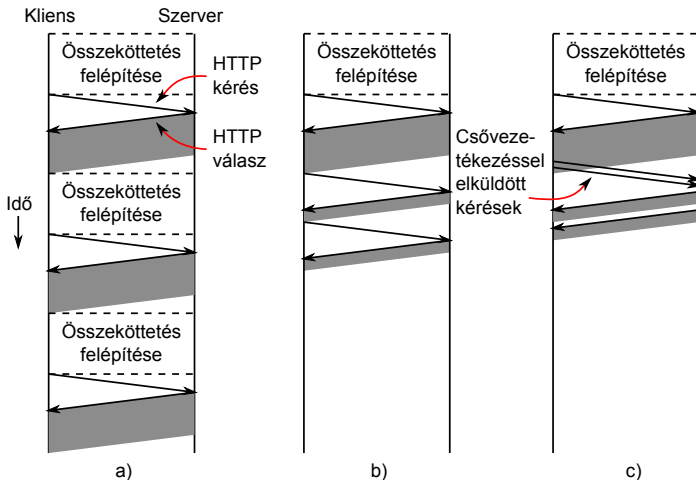
- Eredeti cél: információcsere webszerver és kliens között
- HTTP 1.1: [RFC2616](#)
- ASCII-alapú, kérdés-válasz protokoll, tipikusan TCP felett, tartalom típusa MIME-mal adott
- Új cél: „szállítási protokoll”, pl.
  - médialejátszó információkat kér le
  - víruskereső frissítése
  - szoftverfejlesztők projektállományokat töltenek le
  - [SOAP](#) (távoli eljáráshívás HTTP felett)

# HTTP: hiperszöveg-átviteli protokoll

## Kapcsolatok

- HTTP 1.0: egy összeköttetésben egy kérdés-válasz pár → átlagosan 40 további objektum oldalanként!
- HTTP 1.1:
  - tartós kapcsolatok (persistent connection) / kapcsolat újrahasználás (connection reuse)
  - csővezeték (pipeline) módszer
  - gyorsulás okai:
    - új összeköttetés kiépítése időigényes
    - kevesebb lassú TCP indítás (torlódáskezelés)
  - kapcsolat lezárása:
    - időkorlát
    - túl sok nyitott kapcsolat
- HTTP 1.1 előtt: párhuzamos összeköttetés (parallel connection)
  - késleltetési idő jelentős részét elrejtí, de
  - torlódáskezelés → versengő kapcsolatok, csomagvesztés

# HTTP: hiperszöveg-átviteli protokoll



HTTP a) több összeköttetéssel és egymást követő kérésekkel b) tartós kapcsolat és egymást követő kérések c) tartós kapcsolat és csővezeték módszerrel küldött kérések

# HTTP: hiperszöveg-átviteli protokoll

## Metódusok

- a protokoll műveletei
- kis- és nagybetűs alakra érzékenyek
- kérés első szavai

```
http://xenia.sze.hu/index.html
```

```
wajzy@wajzy-laptop:~$ telnet xenia.sze.hu 80
Trying 193.224.129.115...
Connected to xenia.sze.hu.
Escape character is '^]'.
GET /index.html HTTP/1.1
Host: xenia.sze.hu

HTTP/1.1 200 OK
Date: Tue, 03 Sep 2013 06:43:34 GMT
Server: Apache/2.2.16 (Debian) PHP/5.3.3-7+squeeze14 with Suhosin-Patch mod_python/3.3.1 ...
Perl/v5.10.1
Last-Modified: Wed, 13 Feb 2013 09:28:48 GMT
ETag: "1fa02-b1-4d597c4f4d400"
Accept-Ranges: bytes
Content-Length: 177
Vary: Accept-Encoding
Content-Type: text/html

<html><body><h1>It works!</h1>
<p>This is the default web page for this server.</p>
<p>The web server software is running but no content has been added, yet.</p>
</body></html>
Connection closed by foreign host.
wajzy@wajzy-laptop:~$
```

# HTTP: hiperszöveg-átviteli protokoll

Metódus	Leírás
GET	Weboldal olvasása
HEAD	Weboldal fejlécének olvasása
POST	Weboldalhoz történő „hozzáfűzés”
PUT	Weboldal tárolása
DELETE	Weboldal eltávolítása
TRACE	Bejövő kérés visszaküldése
CONNECT	Kapcsolódás proxy-n keresztül
OPTIONS	Egy oldal opcióinak lekérdezése

Beépített HTTP-kérés metódusok

Kód	Jelentése	Példák
1xx	Információ	100 = a kiszolgáló jóváhagyja az ügyfél kérését
2xx	Siker	200 = sikeres kérés
3xx	Átírányítás	301 = az oldal elköltözött
4xx	Ügyfél hibája	404 = az oldal nem található
5xx	Kiszolgáló hibája	500 = belső hiba a kiszolgálóban

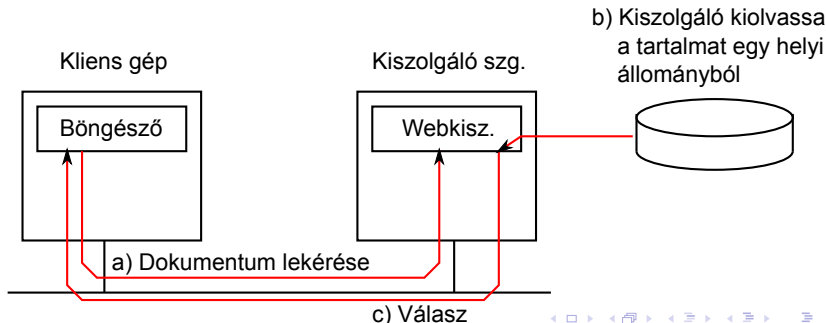
A [válasz állapot kódok](#) csoportjai

A HTTP fejlécek listája [itt](#) érhető el.



# Weboldalak letöltése

- 1 Böngésző beolvassa az URL-t (pl. `http://xenia.sze.hu/index.html`)
- 2 Megkérdezi a `xenia.sze.hu` IP-címét a **DNS**-től
- 3 Válasz: `193.224.129.115`
- 4 Böngésző TCP kapcsolatot létesít a webkiszolgáló 80-as portjával
- 5 Lekéri az `/index.html` oldalt
- 6 Kiszolgáló HTTP válaszban küldi az oldalt
- 7 Ha az oldal további megjelenítendő tartalmak URL-jeit tartalmazza, azokat is hasonlóan lekéri
- 8 Böngésző megjeleníti az oldalt
- 9 A TCP összeköttetést lebontják

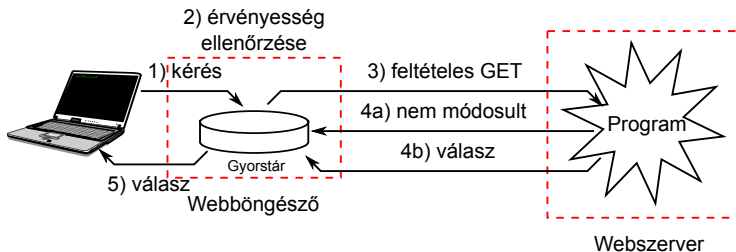


- Miért van rá szükség?
  - Egyre több forrásból származó információ építi fel az egyre nagyobb weboldalakat
  - TCP: egyszerre egy kérést tud hatékonyan kiszolgálni, a probléma megkerülése nem elég hatékony (egymással versengő TCP-összeköttetések, erőforrásokkal visszaélés, csővezeték blokkolódása, duplikált forgalom, stb.). „Hackelésre” többé semmi szükség.
- A [SPDY/2](#) (Google) alapján hozta létre az [IETF egyik munkacsoportja](#) (httpbis)
- Fő eltérések az 1.x-hez képest:
  - bináris formátum (1.x: szöveges)
  - teljesen multiplexelt (1.x: blokkok, sorba rendezés) → egy kapcsolattal is lehetséges a párhuzamos letöltés
  - fejlécek tömörítettek ([HPACK](#), Huffman-kód)
  - szerver olyan tartalmat is küldhet a kliensnek, amiről az még nem is tudja, hogy hamarosan szüksége lesz rá (push)

- Titkosítás elvileg nem kötelező, de a böngészők gyakorlatilag csak [TLS](#) felett engedik a [HTTP/2](#) használatát
- A titkosítás miatt a nyomkövetés nehézkes, de [megoldható](#)
- [Specifikáció](#), [FAQ](#), [teszt](#)

# Gyorstárazás (caching)

- Azonos oldal sokszori újratöltése pazarló → mentsük le!
- Változások detektálása
  - URL nem alkalmas (pl. híroldal)
  - 1. mo.: oldal érvényesítés (*Expires* fejléc, ha van, különben heurisztika *Last-Modified* felhasználásával; veszélyes, pl. tőzsde)
  - 2. mo.: feltételes GET (*Last-Modified* és *If-Modified-Since*, vagy *ETag* és *If-None-Match* → jó pl. többnyelvű oldalaknál)
  - De a *Cache-Control* **direktívái** mindent felülírnak



- Helyettes gyorstárazás (Proxy cache)

- Probléma: egymástól független oldalletöltések
- Fizetős oldalak, testre szabható portálok, ...?
- IP-cím figyelés nem megoldás: [DHCP](#), [NAT](#)
- Megoldás: süti (NetScape, 1994, [RFC2109](#))

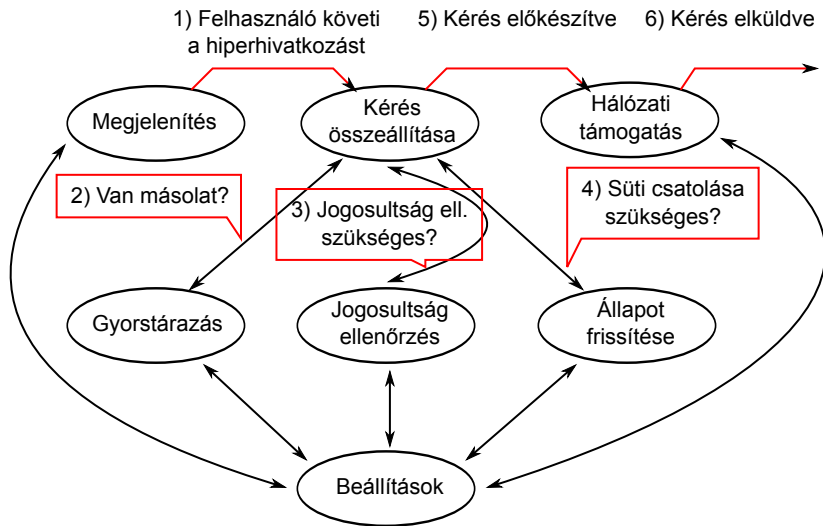
## Tulajdonságok:

- a kért oldallal együtt küldi a kiszolgáló
- max. 4kB méretű karakterlánc, max. 20 süti/tartomány/felhasználó
- korlátozott idejű megőrzés (tartós/nem tartós süti)
- biztonságos átvitel támogatott ([SSL/TLS](#))
- legközelebbi letöltésnél a böngésző mellékeli a tartomány sütiit

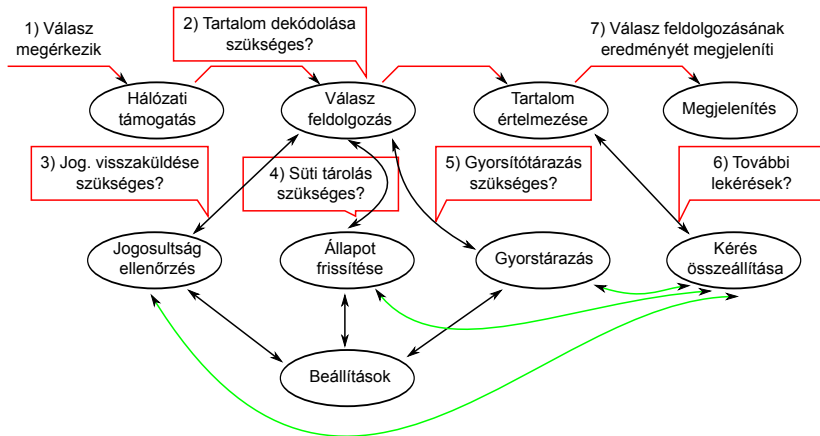
Tartomány	Útvonal	Tartalom	Lejárat	Biztonság
toms-casino.com	/	CustomerID=297793521	15-10-10 17:00	igen
jills-store.com	/	Cart=1-00501;1-07031;2-13721	11-1-11 14:22	nem
aportal.com	/	UserID=4627239101	31-12-19 23:59	nem

Új probléma: [webes nyomkövetés](#)

# Kérés előállítás folyamata



# Válasz feldolgozása





# MIME-típusok

Különféle típusú dokumentumok kezeléséhez:

[application/vnd.ms-powerpoint](#)

Alkalmazás, gyártóspecifikus formátum, PowerPoint

[application/pdf](#)

PDF-dokumentum

[image/x-photoshop](#)

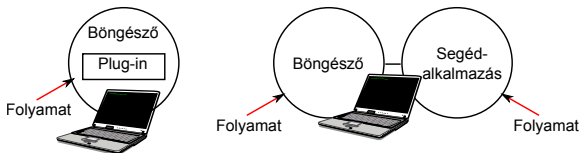
PhotoShop kép

[video/mpeg](#)

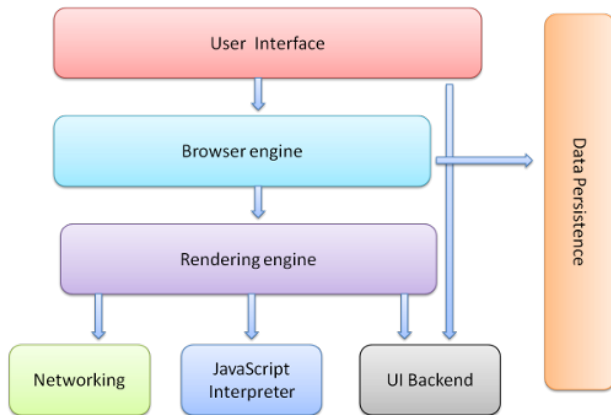
MPEG videó, stb.

A böngésző ezeket

- közvetlenül kezeli (HTML, XML, CSS)
- beépülő modullal (plug-in)
- segédalkalmazással (helper application)



# Böngészők felépítése

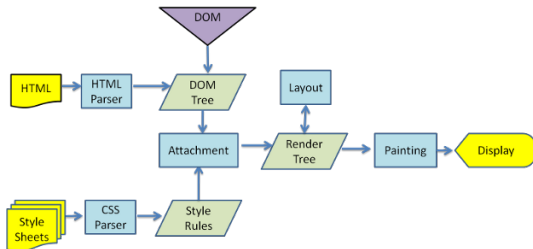


## Webböngészők fontosabb komponensei

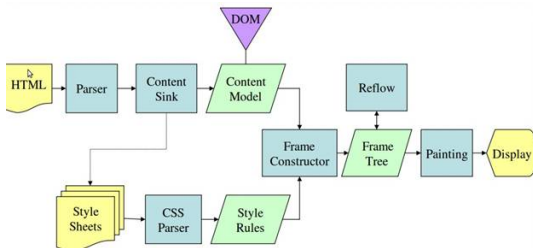


## Renderelés folyamata

# Böngészők felépítése

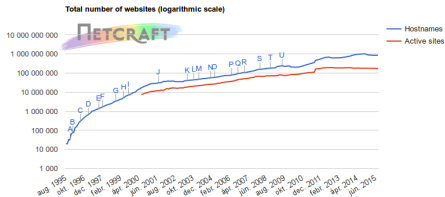


## WebKit motor működésének vázlata

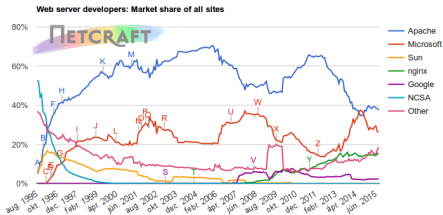


## Mozilla Gecko motor működésének vázlata

# Webszerverek piaca



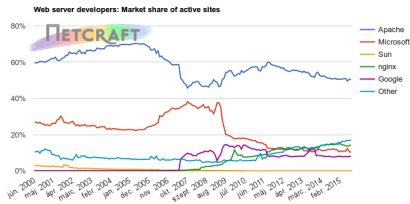
## Weboldalak számának alakulása



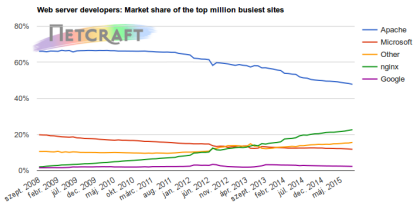
## Webszerverek piaci részesedése

Forrás: [Netcraft](#)

# Webszerverek piaca



Webszerverek aránya az aktív weboldalak között



Webszerverek aránya az 1 000 000 legforgalmasabb oldal között

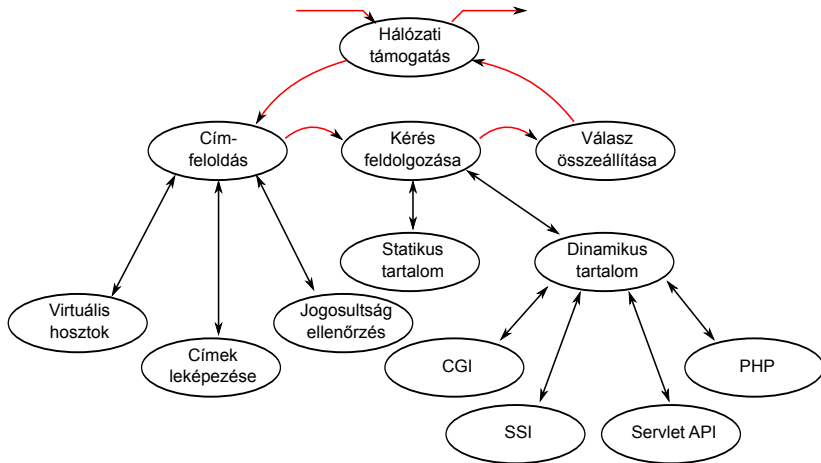
Forrás: [Netcraft](#)

# Az Apache webszerver architektúrája

- Apache 1.3
  - folyamat alapú (stabilitás  $\leftrightarrow$  teljesítmény)
  - Windows támogatás (nem hatékony)
  - moduláris (további funkciókkal bővíthető)
- Apache 2.0
  - Multi Processing Modules, MPM (tisztán folyamat alapú  $\leftrightarrow$  tisztán szál alapú)
  - protokoll modulok (pl. POP3 felhasználók azonosíthatók a webes hitelesítő rendszerrel)
  - kompatibilitási problémák
  - hordozható (Apache Portable Runtime  $\rightarrow$  Windows-on is hatékony)

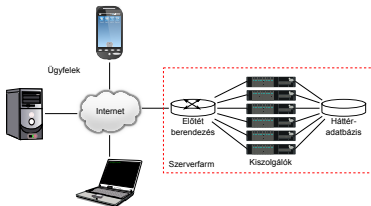
Forrás: [TLDP](#)

# Webszerverek működése



Egyetlen logikai gépnek kell tűnnie

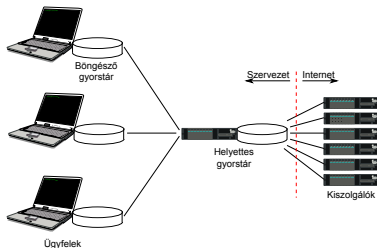
- ❶ DNS, IP-címek körforgó listája
- ❷ Előtét-berendezés (front end) használata
  - Ált. adatkapcsolati rétegbeli kapcsoló vagy IP útválasztó
  - Magasabb szintű protokollok fejléceit használja, hogy a webkérés minden minden csomagját ua. a kiszolgáló kaphassa meg → veszélyes, de hasznos
    - ❶ üzenetszórás: a szerver eldönti, mire válaszol
    - ❷ terhelés-kiegyenlítés (load balancing)
  - Miért kukucskálunk? Pl. gyorsítótárzás, terheléskiegyenlítés (TCP), felhasználó azonosítás, munkamenetek (HTTP, sütik) miatt



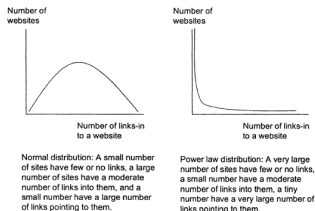


# Webhelyettesek (web proxy)

- sokak által lekért tartalmakat érdemes (forgalom csökkentés → költségcsökkentés, válaszidő csökk.) közös helyen tárolni, pl. képek
- több szint is kialakítható
- kb. 100 főig hatásos, átl. egy szint elég
- tartalomszűrés
- anonimitás
- de hasztalan: titkosított forgalom, hitelesítést igénylő oldalak, állandóan frissülő tartalom esetén



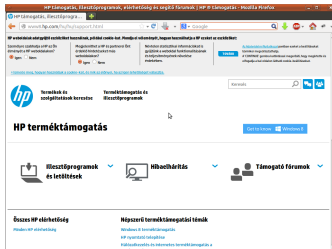
- Link / URL (tartomány, könyvtár) szerkezet
- Skálafüggetlen gráf ( $\approx$ viszonylag sok nagy fokszámú csomópontjuk van, és a csomópontok fokszámeloszlása méretfüggetlen)
- Kis (átmérőjű) világok, lokalitás
- Kapcsolódások dinamikusak (népszerűség  $\rightarrow$  hatványfüggvény)
- Irányított kapcsolatok



Normál és hatványfv. szerinti eloszlás (forrás)

- Tudás hasznosítás
  - információ visszakeresés,
  - újra elrendezés,
  - nyújtás
- Navigáció
  - böngészés
  - célzott keresés (a kereső az „új URL”)

# A Web felhasználása



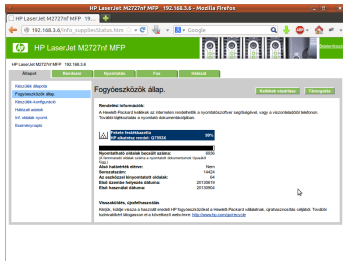
Terméktámogatás

Hírportal

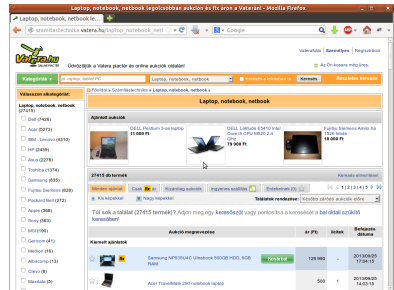


E-learning oldal

# A Web felhasználása



Beágyazott rendszerek

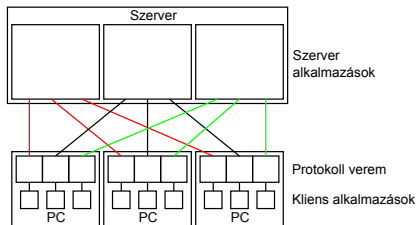


E-kereskedelem

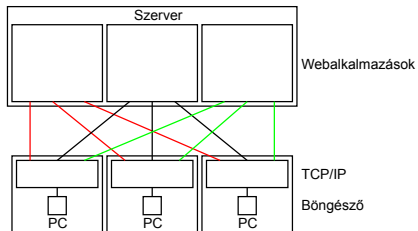
- Hirdetési felület
- Kereskedelmi csatorna (biztonság)
- Kommunikációs megoldás (→ E-mail, Facebook, Twitter, stb.)
- Szórakoztatás (interaktív játékok, webcast, podcast, stb.)

- Web-technológia belső vállalati célokra
  - tűzfalak
  - azonosított felhasználók
- Alkalmazás
  - üzleti kapcsolattartás, oktatás (→ extranet)
  - tudás menedzsment
    - információ szigetek összekapcsolása
    - portálok
  - előtét vállalati alkalmazásokhoz

# Webalkalmazások



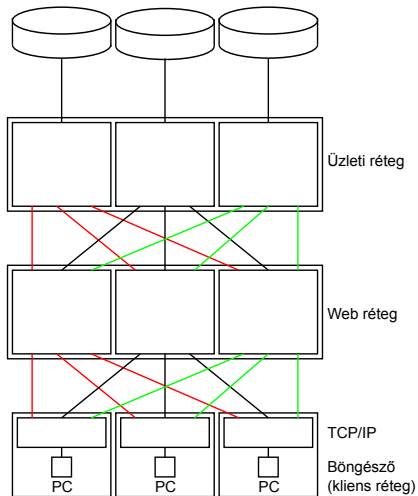
Hagyományos kliens-szerver alkalmazások (vastag kliens, fat client)



Webalkalmazások (vékony kliens, thin client)



# Több rétegű (multi-tier) webalkalmazások



- Elosztott számítógépes környezet
- Automatikusan indított tranzakciók nem csak böngészőkből
- Intelligens ügynökök, piacterek, árverések, ...
  - nyílt rendszerek
  - webszolgáltatások (web service)

- Összetett üzleti információk
  - időjárás, hírek, menetrendek, árfolyamok
  - csekk és kártya ellenőrzés, árverés
- Tranzakciós szolgáltatások
  - helyfoglalások, bérletek, megrendelések
  - beszállítói láncok
- Üzleti folyamatok megnyitása
  - munkafolyamat szintű összekapcsolás
  - folyamat szintű integráció

- az elvárt szolgáltatást nyújtja
- interfész specifikációi
  - teljesek, jól definiáltak
  - publikusan hozzáférhetőek és
  - közmegegyezéssel karbantartottak
- az implementációja megfelel a specifikációinak

- Interoperabilitás (gépek, nyelvek, operációs rendszerek között)
- Heterogén elosztott rendszerek megvalósítása
- Skálázhatóság (egy  $\rightarrow$  több gép, kis  $\rightarrow$  nagy teljesítményű rendszer)

Andrew S. Tanenbaum, David J. Wetherall

[Számítógép-hálózatok](#)

Panem, Bp., 2013.

HTML5 Rocks Tutorials

[How Browsers Work: Behind the scenes of modern web browsers](#)