



# Webfejlesztés alapjai

**KnowCamp**

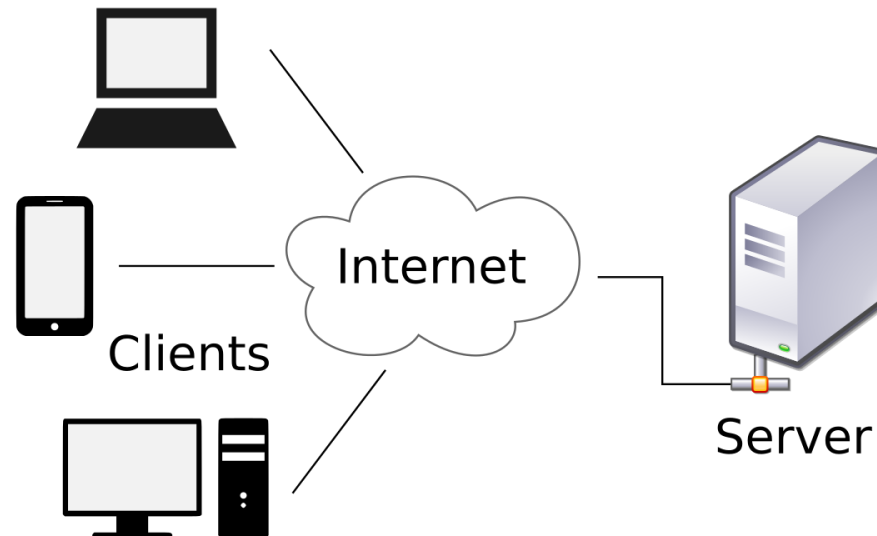
# A WEB MŰKÖDÉSE

**Az internet működése**

**A web születése**

**HTTP protokoll**

- **Internetezés:** Távoli számítógépekkel (kiszolgálókkal) való információcsere.
- **Kliens:** Az adatot kérő felhasználó számítógépe
- **Szerver:** Az adatot biztosító távoli számítógép

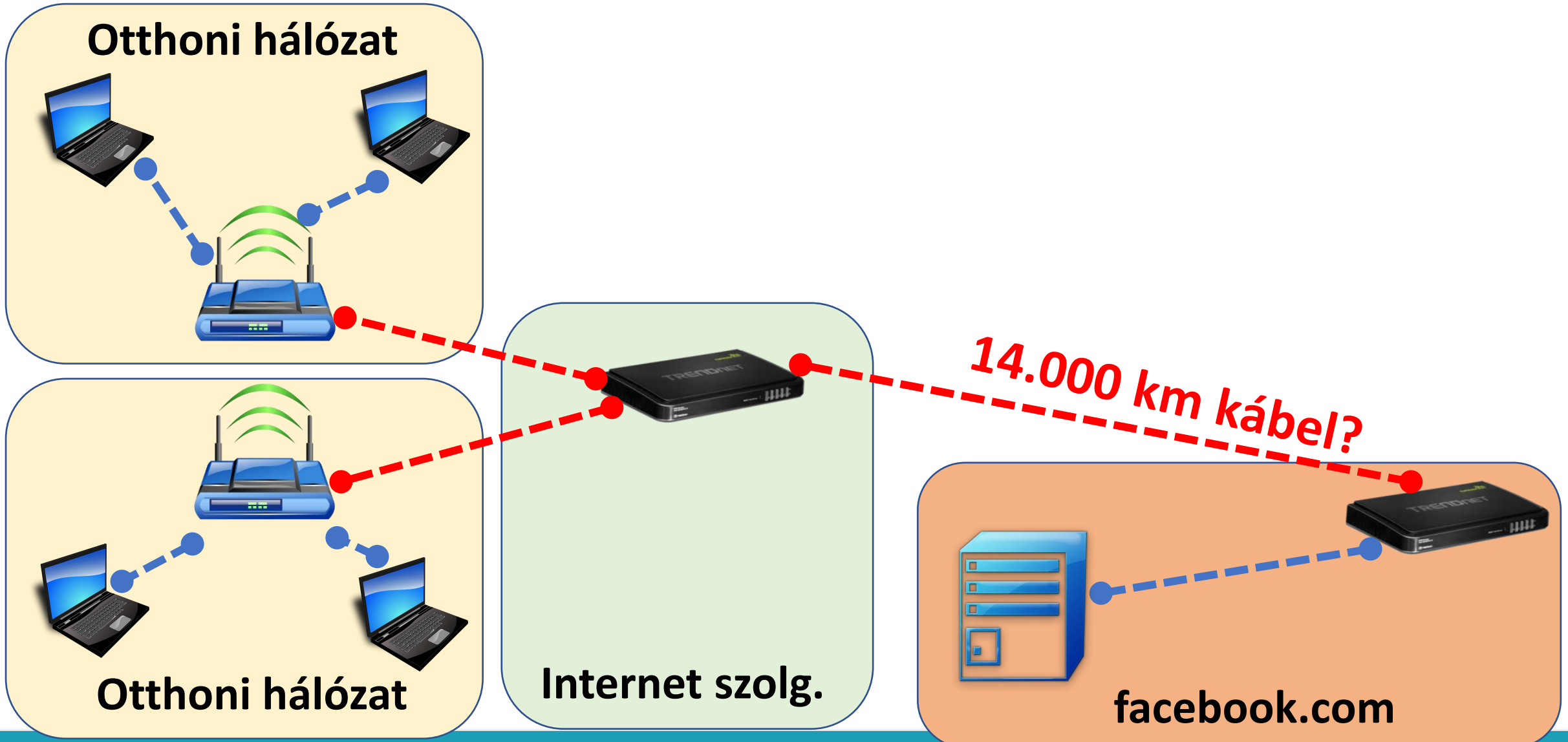


- **Domain Name System (DNS):** A távoli számítógépek IP címét nem szükséges ismernünk, elég ha egy rövid nevet megjegyzünk (domain név).
- **Domain név felépítése:** facebook.com (cím – pont – végződés)
- **Végzódések:** **.com** (kereskedelem), **.net** (hálózati szolgáltatások), **.org** (civil szervezetek), **.hu** (magyar oldalak), **.sk** (szlovák oldalak), **.de** (német oldalak)
- **Aldomain:** pl: mail.google.com, drive.google.com, www.google.com (adott domainen belül más-más IP címre mutathatnak a különböző aldomainek)
- **Rövid történelme**
  - Minden kliens letöltött egy katalógust egy központi helyről
  - Egy központi szervernek elküldi minden kliens az ismert nevet és visszakapja tőle az IP címet
  - Egyre több kérést kell a szervernek kiszolgálnia -> több szerver
  - Globális elosztott rendszer
  - Root DNS szerverek a fő katalógusok, tőlük vesznek át adatot más szerverek

- DNS zónafájl

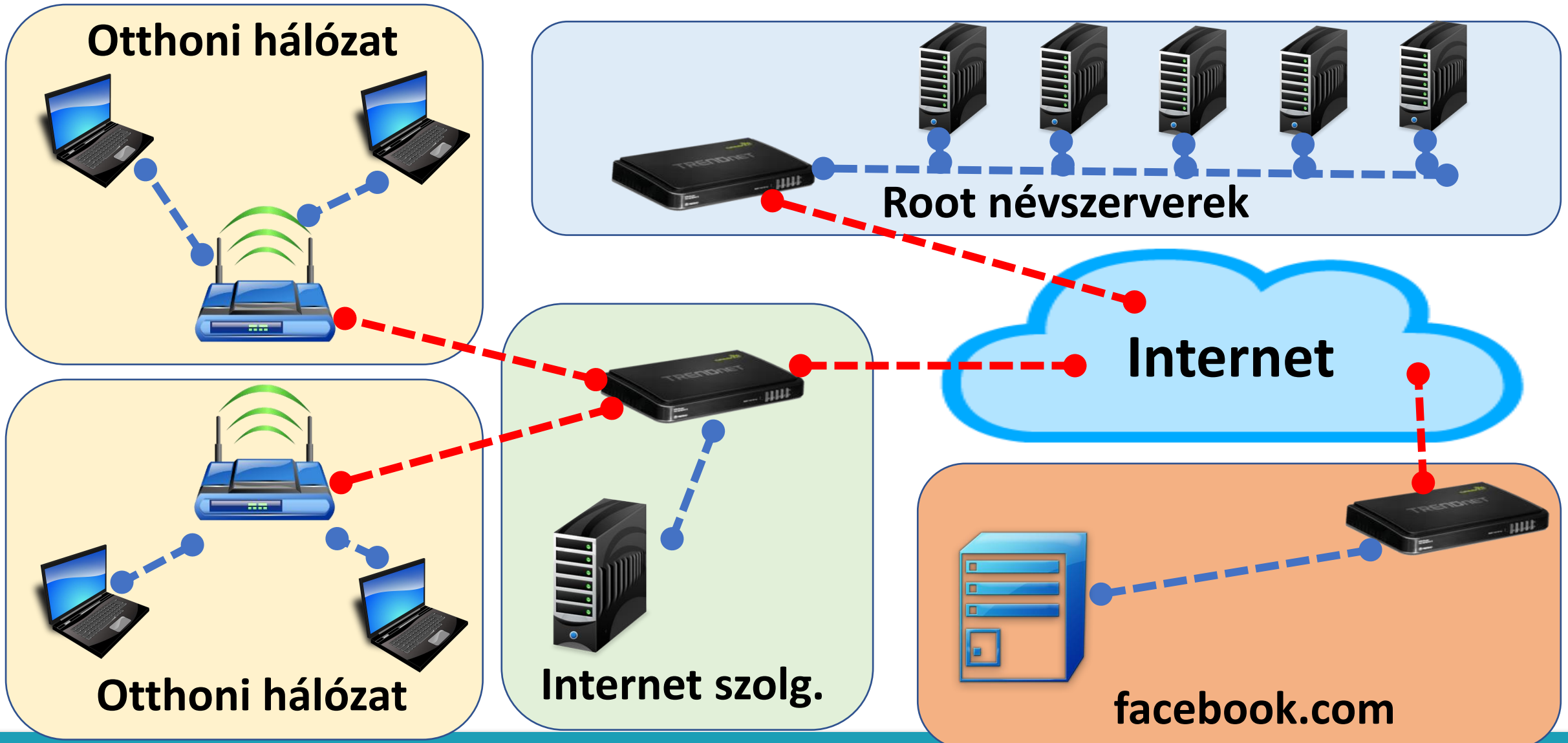
\$TTL 1d	⇒	Default TTL of 1 day
\$ORIGIN example.com.	⇒	Default FQDN to attach
@ IN SOA ns1.example.com. admin.example.com. ( 2013091200 ; se = serial number 12h ; ref = refresh 15m ; ret = refresh retry 3w ; ex = expiry 2h ; nx = nxdomain ttl )	⇒	SOA (Start of Authority)
IN NS ns1.example.com. IN NS ns2.example.net.	⇒	NS record
3w IN MX 10 mail.example.com. IN MX 20 mail.example.net.	⇒	MX record
ns1 IN A 172.16.140.41 mail IN A 172.16.140.42 joe IN A 172.16.140.43 www IN A 172.16.140.44	⇒	A record
ftp IN CNAME ftp.example.net.	⇒	CNAME record

Aldomain és IP cím  
összerendelések



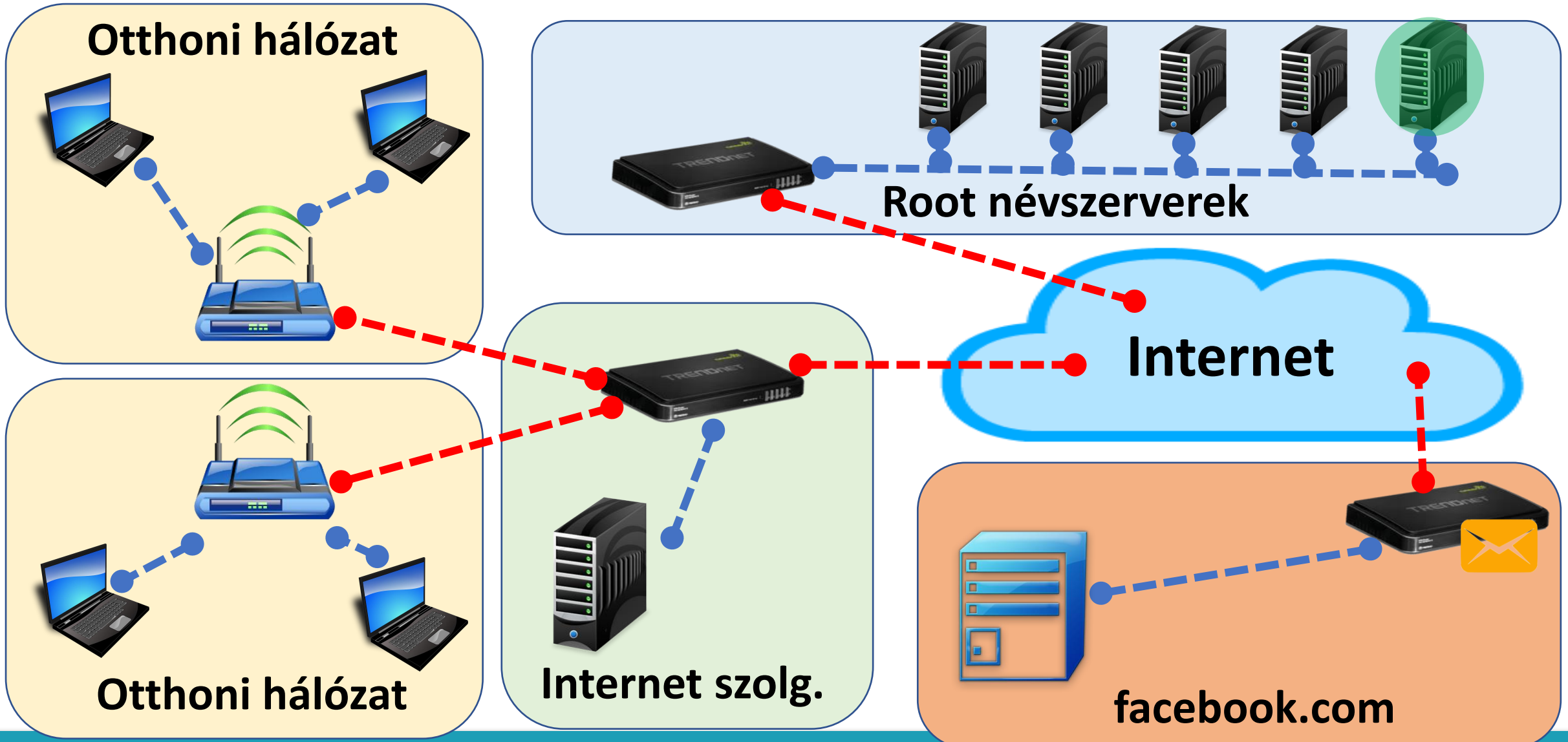
# Az internet működése

7



- **Címkerés folyamata I. (példa)**
- A facebook.com tulajdonosai vásárolnak egy szerver számítógépet és előfizetnek egy internetkapcsolatra
- Statikus IP címet igényelnek (mindig ugyanaz lesz az IP címük)
- A facebook.com tulajdonosai beregisztrálják a domain nevüket és IP címüket (facebook.com) a Root DNS szerverekbe.
- Amíg meg nem szűntetik, addig a bejegyzés megmarad





- **Címkerés folyamata II. (példa)**
- Egy felhasználó fel szeretne menni a facebook.com weboldalra
- Megpróbál csatlakozni
- A facebook.com IP címét a számítógépe először a saját DNS gyorsítótárában keresi

**Saját gyorsítótárban keresés**

↓ nincs meg

**Otthoni routerhez fordulás**

↓ nincs meg

**Internet szolgáltatóhoz fordulás**

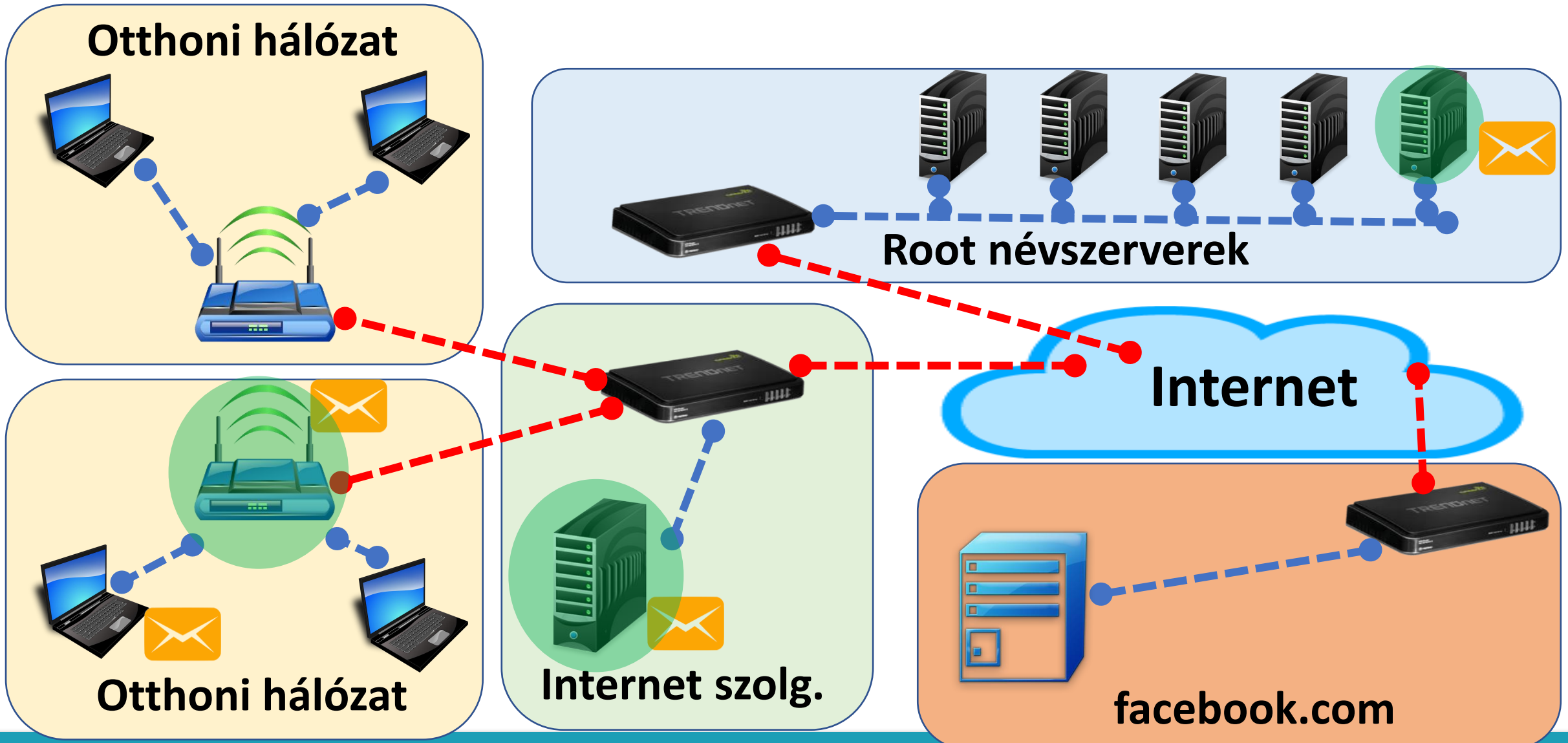
↓ nincs meg

**Root DNS szerverekhez fordulás**

A router visszaküldi a választ a számítógépnek (és elmenti)

A szolgáltató visszaküldi a választ az otthoni routernek (és elmenti)

A Root DNS szerverek visszaküldik a választ a szolgáltatónak (és elmenti)



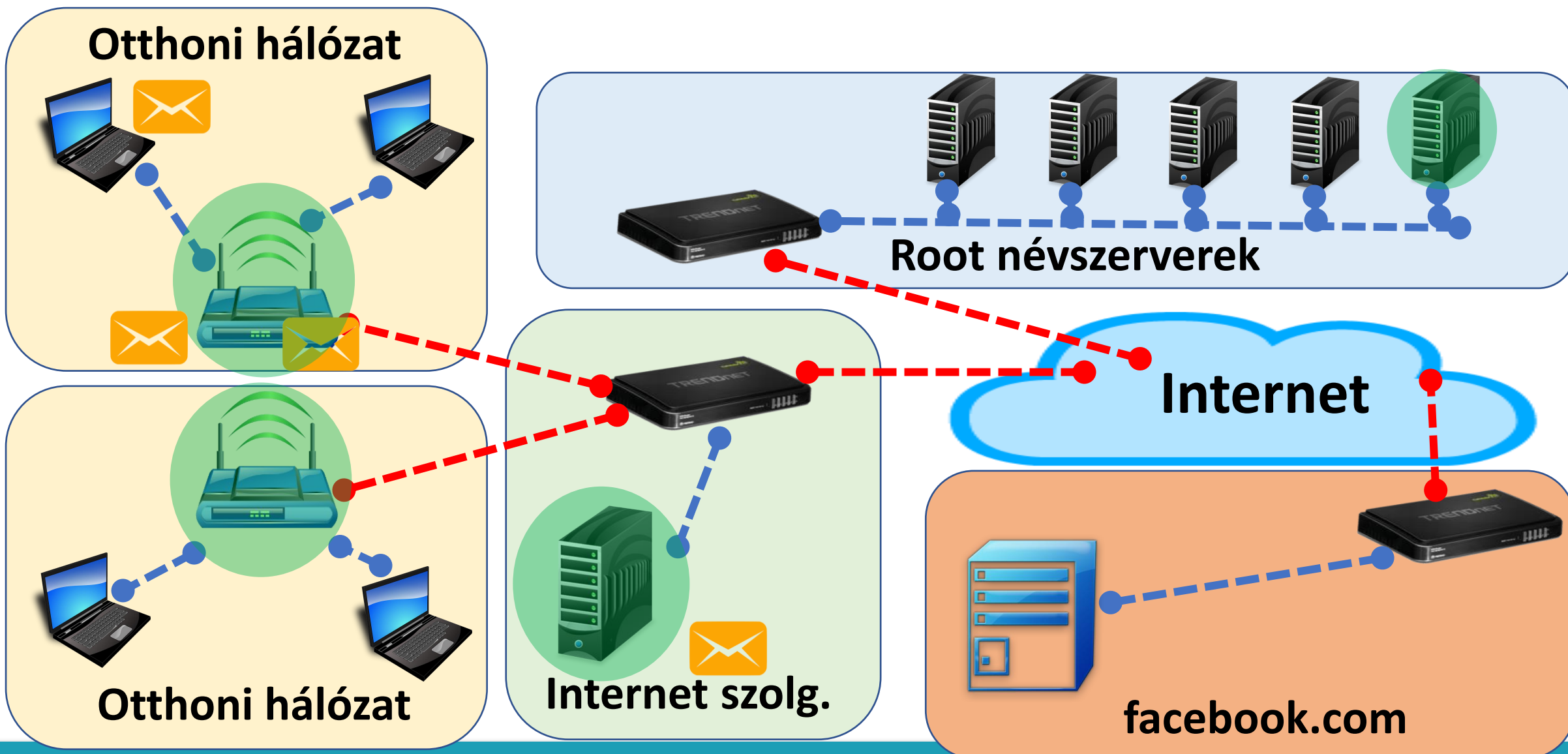
- **Lekérési lánc végén**

- A szolgáltató DNS szervere is megismerte a címet
- Az otthoni hálózat routere is megismerte a címet

- Tehát innentől nekik nem kell továbbadniuk a kérést, hanem saját adatbázisból kiszolgálják a kéréseket

- **Gyakorlatilag:**

- Hogyha a Kossuth u. 43-ból lekértük a facebook.com oldalt, és az internetszolgáltató névszervere eltárolta a címet, akkor a Kossuth u. 58-ból érkező kérésre már a szolgáltató névszervere tud válaszolni.
- Ugyanez a folyamat zajlik egy lakáson belüli hozzáférés esetében is

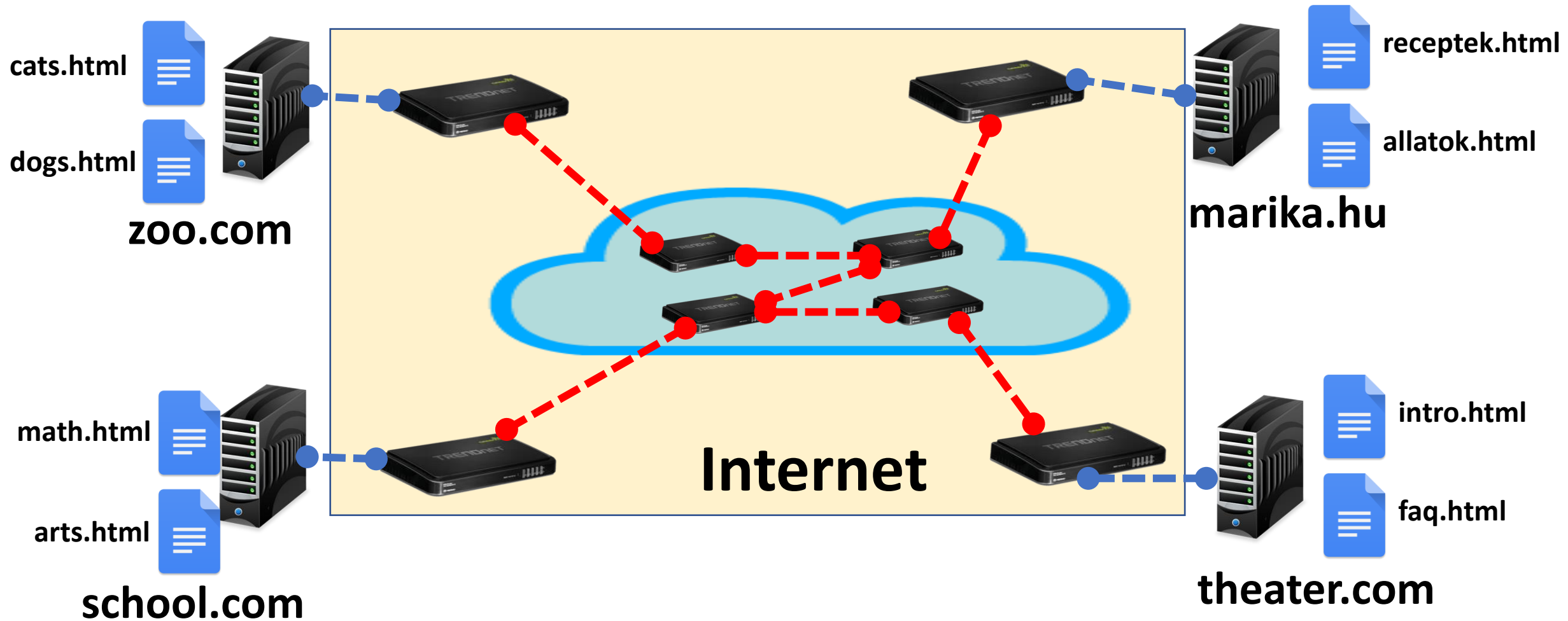


- **Eddigi ismereteink:** A számítógépünk képes kapcsolatba lépni IP vagy domain név alapján egy távoli számítógéppel. (Ez az internet – hálózatok hálózata)
- **World Wide Web**
  - Hivatkozásokkal összekötött dokumentumok rendszere
  - Megalkotója: **Tim Berners-Lee** (CERN részecskefizikai kutatóközpontjának munkatársa)
  - Elsődleges célja a webnek: kutatói munka támogatása, eredmények automatizált megosztása egymás között



# Internet vs. WEB

16

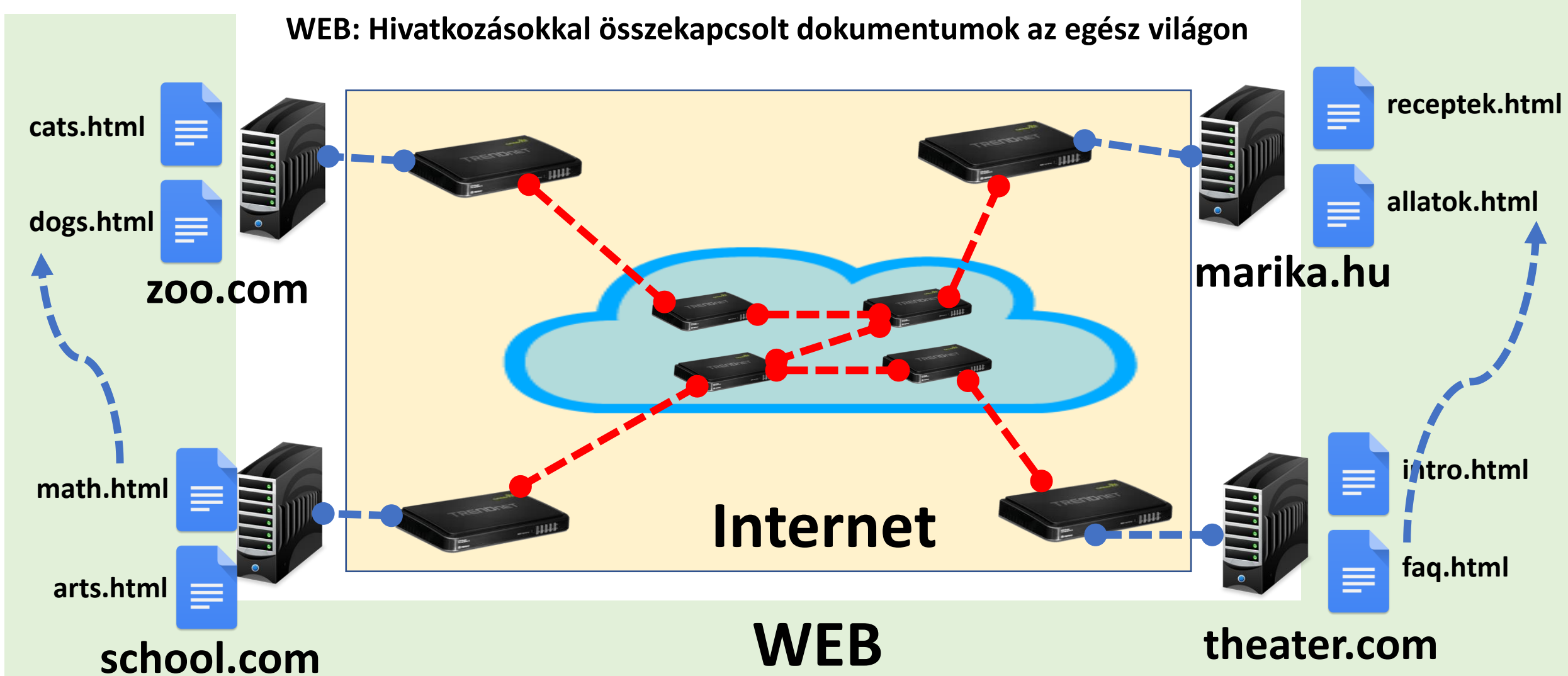




# Internet vs. WEB

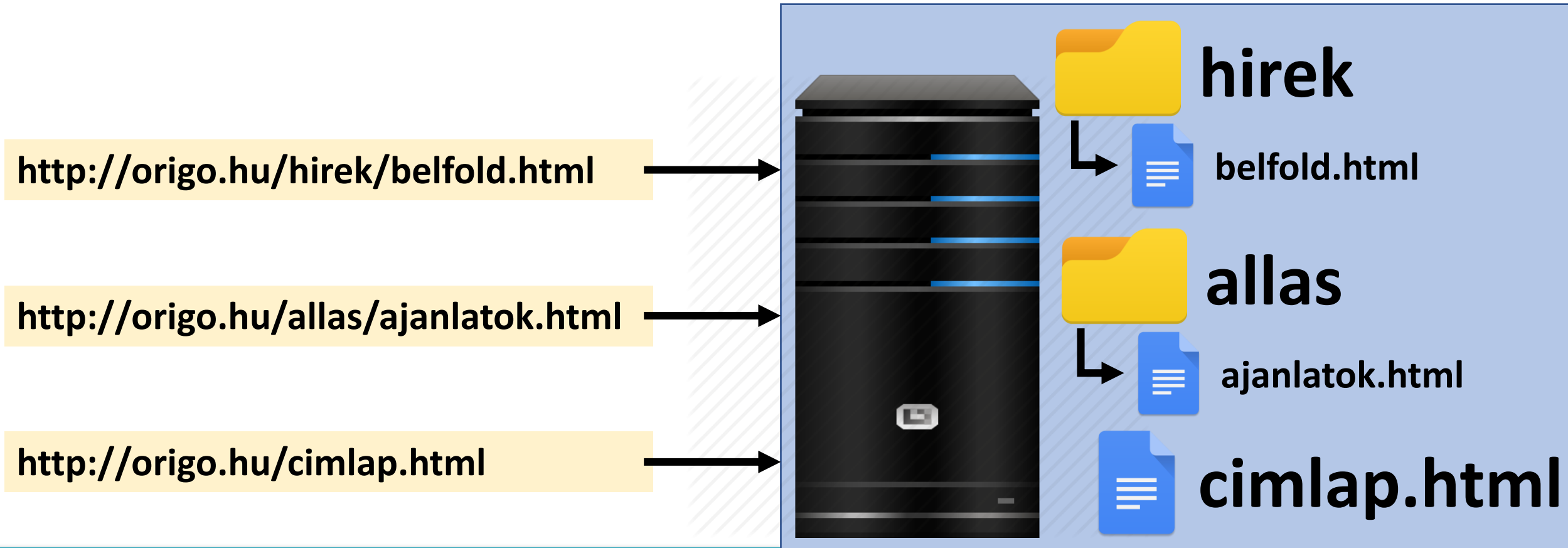
17

WEB: Hivatkozásokkal összekapcsolt dokumentumok az egész világon



- **URL (Uniform Resource Locator)**

- Az összetartozó dokumentumok egyedi címmel való ellátása
- Hierarchia szervezése a dokumentumokból



- **HTML (Hyper Text Markup Language)**

- Séma a dokumentumok elkészítésére
- Egységes, szabványos leíró nyelv

eredmeny.html  
tartalma

```
<table border = "1">  
  <tr>  
    <th>I. hely</th>  
    <th>II. hely</th>  
    <th>III. hely</th>  
  </tr>  
  <tr>  
    <td>Béla</td>  
    <td>Gizi</td>  
    <td>Saci</td>  
  </tr>  
</table>
```



I. hely	II. hely	III. hely
Béla	Gizi	Saci

Ezt látjuk a böngészőben

- **HTTP (Hyper Text Transfer Protocol)**

- A weboldalak tartalmának átvitelét biztosító protokoll
- Kérés-válasz alapú, **mindig a kliens kezdeményez** valamit, a szerver csak válaszol
- Küldünk egy parancsot és paramétereket (adatok), kapunk egy választ (adatok)

```
GET http://users.nik.uni-obuda.hu/sima/201617springnotesPRAMSc.htm
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/we
Referer: http://users.nik.uni-obuda.hu/sima/oktatas.htm
Accept-Encoding: gzip, deflate
Accept-Language: hu-HU,hu;q=0.8,en-US;q=0.6,en;q=0.4
Cookie: SESS5271c37b9aa6f51d903b4e8f7292bcf0=cmcp0o54c97moao5ve9a79v77
```

```
HTTP/1.1 200 OK
Date: Tue, 22 Aug 2017 16:23:52 GMT
Server: Apache
Last-Modified: Mon, 08 May 2017 05:48:50 GMT
ETag: "7a6024-af4-54efccb1db480"
Accept-Ranges: bytes
Content-Length: 2804
Keep-Alive: timeout=15, max=96
Connection: Keep-Alive
Content-Type: text/html
```

## HTTP kérés

- Dokumentum neve
- Elvárt formátum
- Előzőleg látogatott oldal
- Elvárt nyelv

## HTTP válasz

- Dátum
- Utolsó módosítás
- Hossz
- Formátum
- Maga a dokumentum törzse

- HTTP utasítások

**GET** /kovacs.andras/webprog/hello.html **HTTP/1.0**

**GET parancs**

A megadott dokumentum tartalmát küldje vissza a szerver

**Dokumentum neve**

Felépítése: név és kiterjesztés, ahogyan az operációs rendszeren is

**Verziószám**

A HTTP protokoll melyik verziójában értelmezendő a kérés

- HTTP utasítások

Utasítás	Leírás
<b>HEAD</b>	Ugyanaz, mint a GET, de csak az adott dokumentum jellemzőit kéri le, a dokumentumot nem.
<b>GET</b>	Adott dokumentum letöltése
<b>POST</b>	Adatok felküldése a szerverre (pl. űrlapba írt adatok)
<b>PUT</b>	Dokumentum feltöltése a szerverre
<b>DELETE</b>	Dokumentum törlése a szerverről

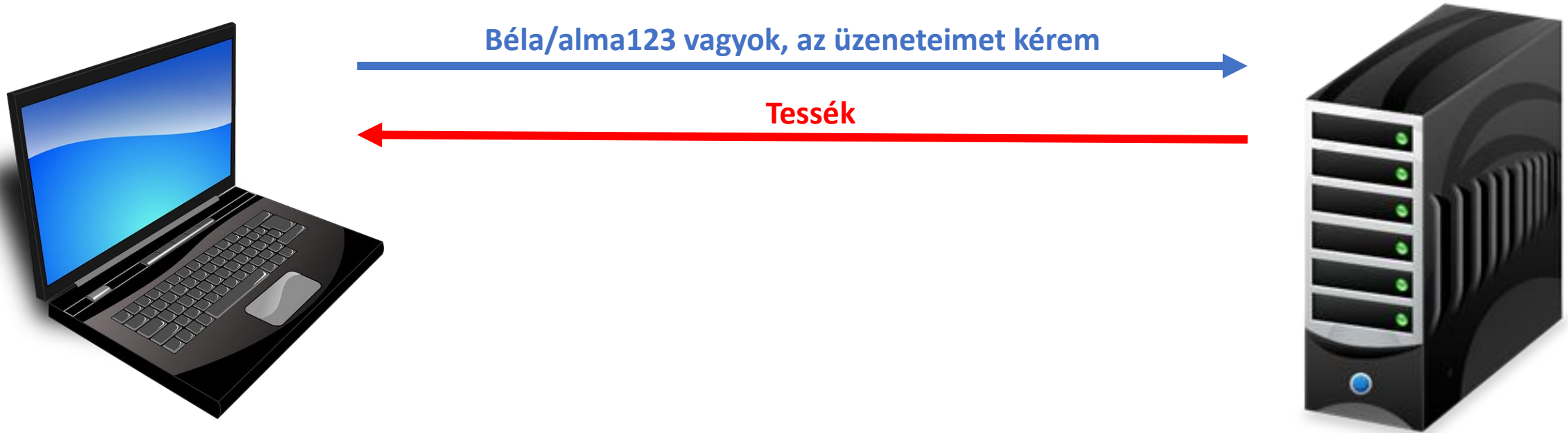
- **HTTP állapotmegőrzés**

- A dokumentum lekérések között nincs kapcsolat
- Tehát amikor böngészünk egy weboldalt, akkor minden hivatkozás megnyitása egy-egy újabb kérés



- **HTTP állapotmegőrzés**

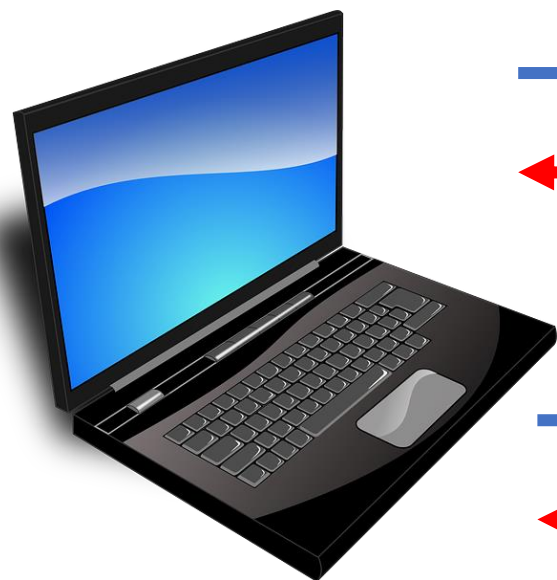
- A dokumentum lekérések között nincs kapcsolat
- Tehát amikor böngészünk egy weboldalt, akkor minden hivatkozás megnyitása egy-egy újabb kérés





- **HTTP állapotmegőrzés**

- A dokumentum lekérések között nincs kapcsolat
- Tehát amikor böngészünk egy weboldalt, akkor minden hivatkozás megnyitása egy-egy újabb kérés



Béla/alma123 vagyok, az üzeneteimet kérem

Tessék

Béla/alma123 vagyok, hányan like-olták a képet?

123-an



- **HTTP állapotmegőrzés**

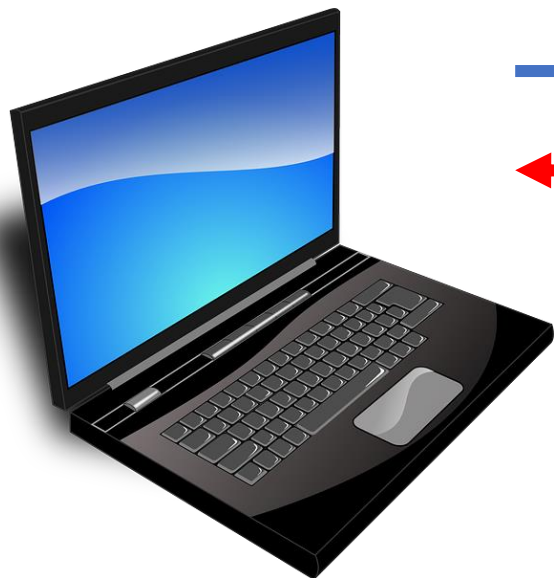
- Akkor miért nem kell facebookra mindig bejelentkezni?
- Cookie/süti biztosítja a két kérés közti kapcsolatot



- **HTTP állapotmegőrzés**

- Akkor miért nem kell facebookra mindig bejelentkezni?
- Cookie/süti biztosítja a két kérés közti kapcsolatot

Béla munkamenet azonosítója:  
**34?!dg53?+%+=+ffd**



Béla/alma123 vagyok, belépnék az oldalra

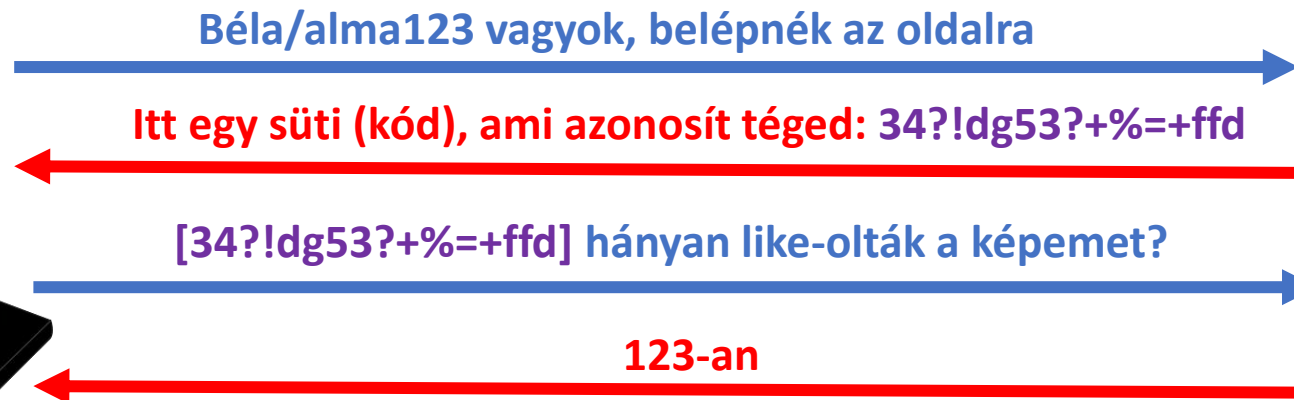
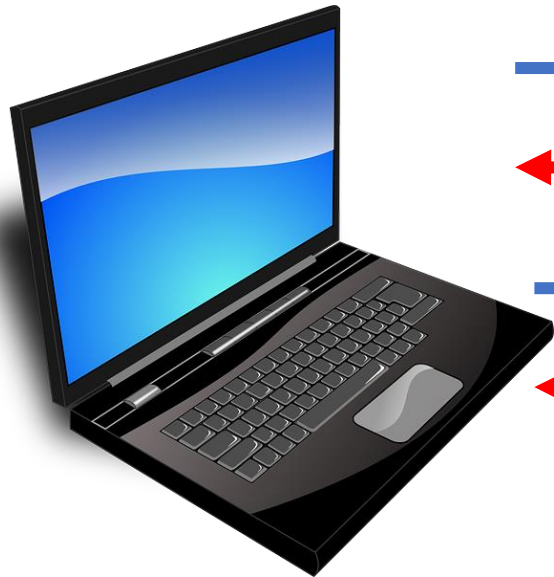
Itt egy süti (kód), ami azonosít téged: **34?!dg53?+%+=+ffd**



- **HTTP állapotmegőrzés**

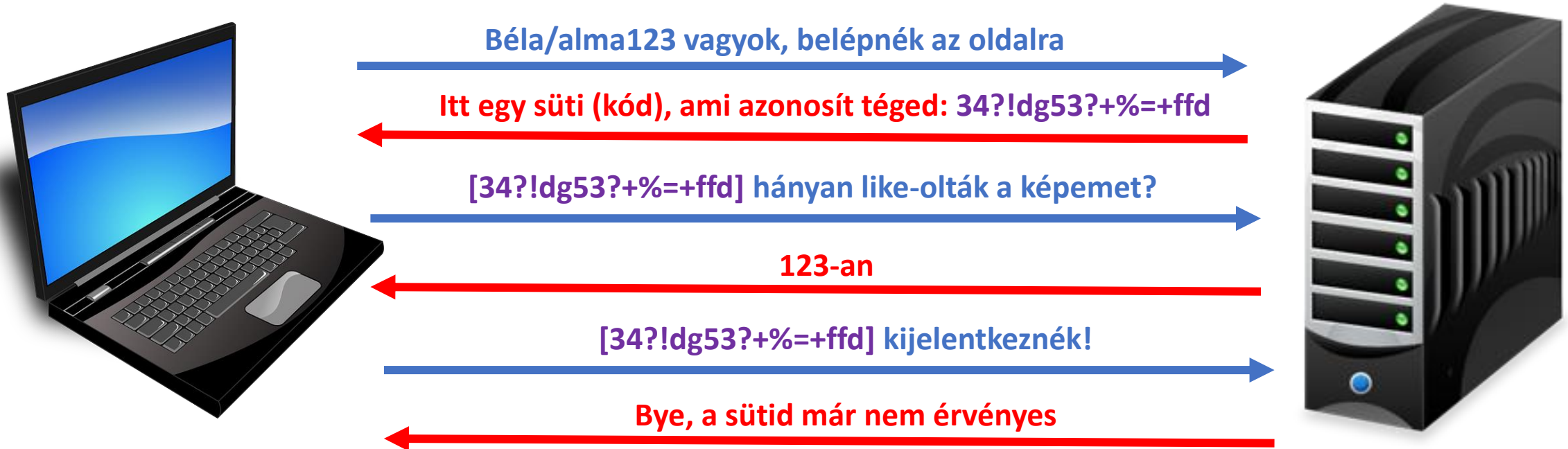
- Akkor miért nem kell facebookra mindig bejelentkezni?
- Cookie/süti biztosítja a két kérés közti kapcsolatot

Béla munkamenet azonosítója:  
34?!dg53?+%+=+ffd



- **HTTP állapotmegőrzés**

- Akkor miért nem kell facebookra mindig bejelentkezni?
- Cookie/süti biztosítja a két kérés közti kapcsolatot



Köszönöm a figyelmet!