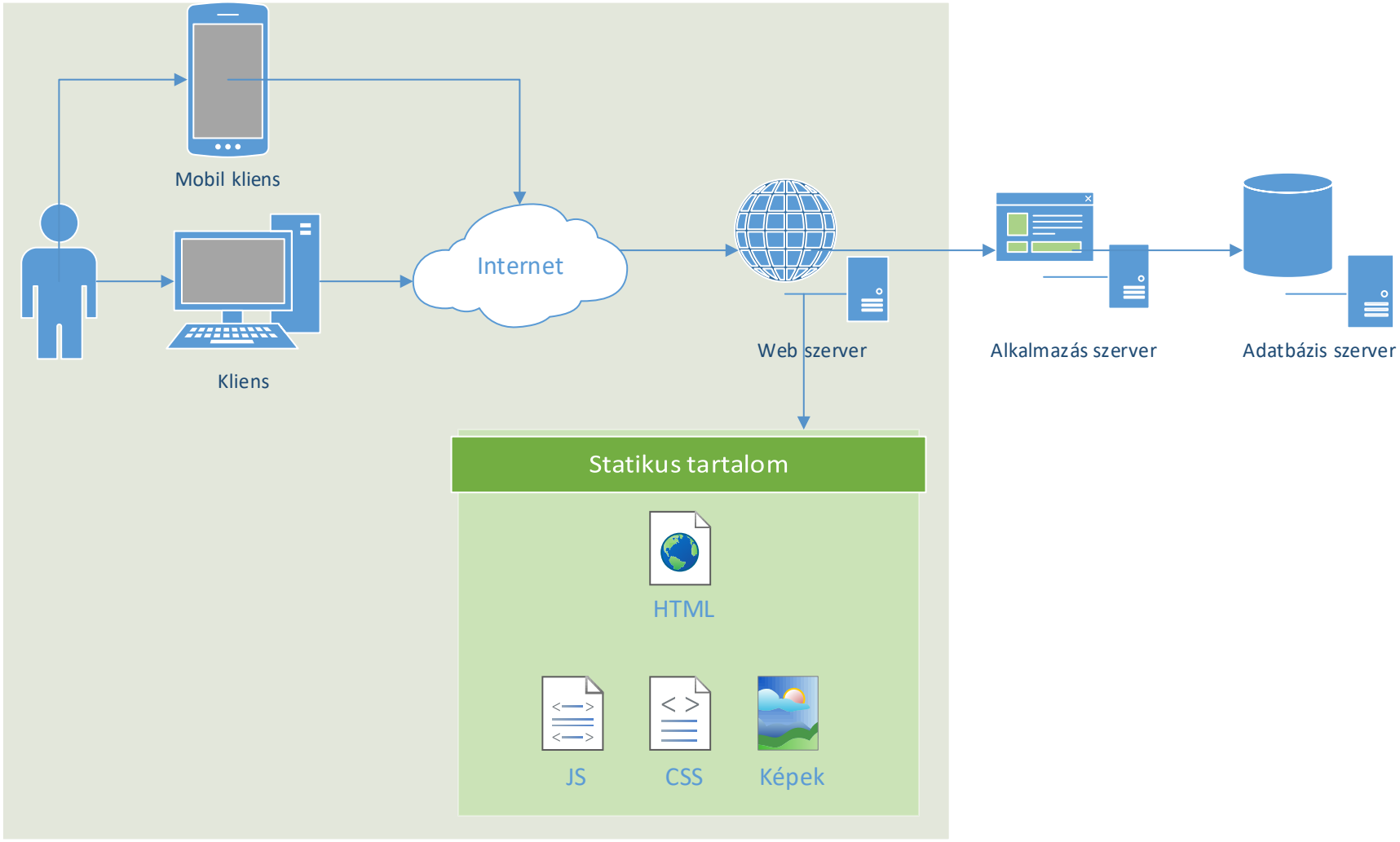
Mobil- és webes szoftverek

Bevezetés a webes technológiák világába

BME AUT
Automatizálási és Alkalmazott Informatikai Tanszék

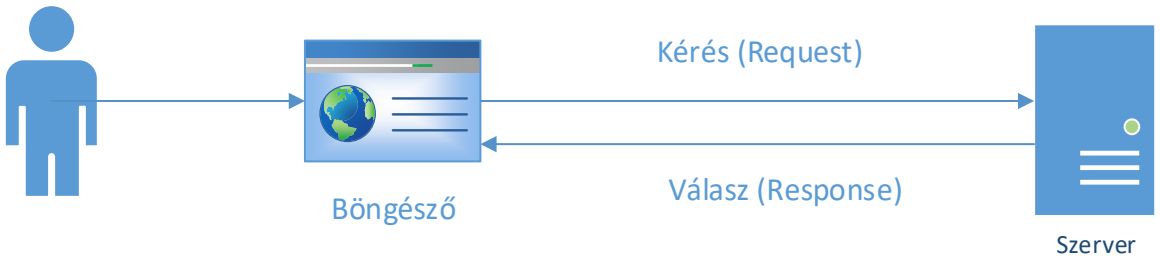
Gincsai Gábor
gincsai@aut.bme.hu

Webes architektúra áttekintése



Kérés-válasz (request-response)

- A kommunikációt mindig a kliens kezdeményezi, a szerver csak válaszol (pull model).
- **User agent**: a kliens általános megnevezése, bármilyen alkalmazás, amely HTTP kérést tud küldeni.
  - pl. leggyakrabban web böngésző, RSS olvasó, mobil kliens

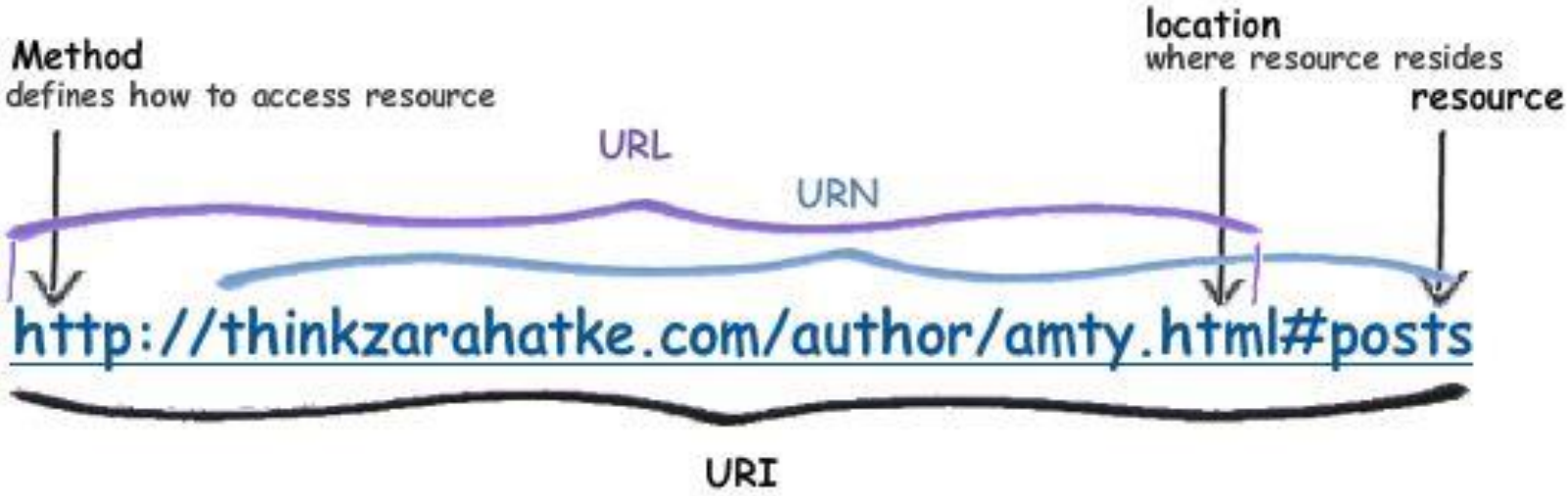


Kapcsolat- és állapotmentes

- A válasz után a szerver bontja a kapcsolatot.
  - HTTP 1.1: a socket alapértelmezés szerint nyitva marad, hacsak nem jön **Connection: Close** header.
- A kérések között nincs állapotmegőrzés.
  - HTTP szinten nem alakul ki felhasználói munkamenet.
    - **Session** (munkamenet): egy felhasználó első és utolsó kérése között lezajló kérés-válasz tranzakciók.
      - Időkorlátos, például 20 perces csúszó ablak (sliding timeout).
  - Állapotmegőrzésre használható pl. cookie, rejtett mező, URL paraméter

Hogyan kezdeményez a kliens?

- Egy URI címre küld megfelelően formázott csomagot



Hogyan válaszol a szerver?

- A beérkező kérést a webszerver feldolgozza és előállítja a szöveges HTTP válaszüzenetet.
  - **Statikus** tartalmakat (fájlokat) szolgál ki jellemzően
    URI → fájlrendszer megfeleltetés alapján.
  - **Dinamikus** tartalmat állít elő a kérés paraméterei
    és az alkalmazás állapota (memória, DB) alapján.

Statikus vs dinamikus kiszolgálás

Statikus kiszolgálás

- Egyszerű
- Olcsó
- Hatékony

A tartalom csak a szerveren található fájlok manipulációjával frissíthető.

Dinamikus kiszolgálás

- Bonyolult
- Drága
- Lassú

A tartalom újraindítás/ telepítés nélkül frissíthető.

Statikus kiszolgálás

- A statikus kiszolgálás a kérések feldolgozásának **egy** lehetséges módja

- **Statikus kiszolgálás ≠ statikus weboldal!**
  - Pl.: statikus JS kódból módosítjuk a tartalmaz
- Statikus weboldal = csak statikus kiszolgálás
  - Pl.: egyszerű HTML fájlok letöltése
- Dinamikus weboldal ≠ csak dinamikus kiszolgálás
  - Pl.: single page application-ök.

A kérés és a válasz felépítése

- Kérés általános formája:

```
Method RequestURI HTTP-Version <CR><LF>
header <CR><LF>
<CR><LF>
body
```

- Válasz általános formája:

```
HTTP-Version Status-Code Reason-Phrase <CR><LF>
header <CR><LF>
<CR><LF>
body
```


(a) Request message
Request start line (command)
GET /tools.html HTTP/1.0
Request headers
User-agent: Mozilla/4.75 [en] (Win98; U)
Host: www.joes-hardware.com
Accept: text/html, image/gif, image/jpeg
Accept-language: en
No request body
Client
www.joes-hardware.com
(b) Response message
Response start line (status)
HTTP/1.0 200 OK
Response headers
Date: Sun, 01 Oct 2000 23:25:17 GMT
Server: Apache/1.3.11 BSafe-SSL/1.38 (Unix)
Last-modified: Tue, 04 Jul 2000 09:46:21 GMT
Content-length: 403
Content-type: text/html
Response body
<HTML>
<HEAD><TITLE>Joe's Tools</TITLE></HEAD>
<BODY>
<H1>Tools Page</H1>
<H2>Hammers</H2>
<P>Joe's Hardware Online has the largest selection of hammers on the earth.</P>
<H2><A NAME=drills></A>Drills</H2>
<P>Joe's Hardware has a complete line of cordless and corded drills, as well as the latest in plutonium-powered atomic drills, for those big around the house jobs./<P>...
</BODY>
</HTML>

A kérés és a válasz elemei

- Metódusok (methods, verbs)
  - GET, POST, HEAD, OPTIONS, DELETE, TRACE, PUT
- A kért erőforrás (resource)
  - URI pl.: http://www.aut.bme.hu
- Fejléc mezők
  - Szerverre, tartalomra, biztonságra és gyorsítótárazásra vonatkozó extra információk
- Hibakódok (Status-Code) + Hibaüzenetek (Reason-Phrase)
  - Pl.: 404 – Not Found

Metódusok

- **GET**: a kért erőforrás letöltése a szerverről.
- **POST**: adatot (pl űrlap tartalmát) küld fel a szerverre a kérés body részében.
- **HEAD**: meta információk lekérdezése a megadott erőforrásról.
  - pl. méret, típus, utolsó módosítás dátuma
- **OPTIONS**: visszaadja a szerver által támogatott HTTP metódusok listáját.
- **DELETE**: törli a megadott erőforrást.
- **TRACE**: visszaküldi a kapott kérést (diagnosztika).
- **PUT**: feltölti a megadott erőforrást a szerverre.

Metódusok tulajdonságai

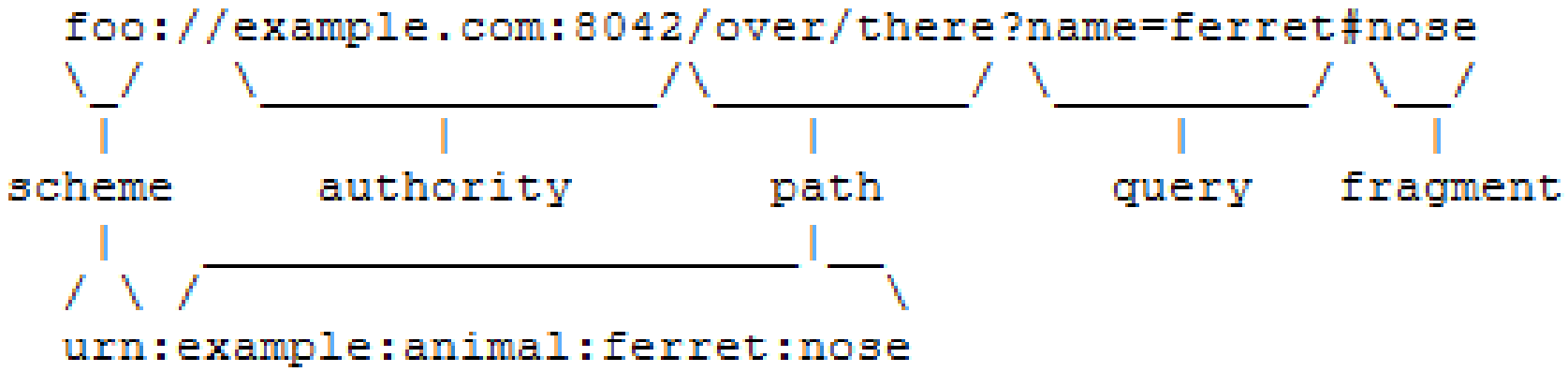
- **Biztonságos (safe) metódus**
  - Csak információ letöltésére szolgál, nincs mellékhatása, nem változtat állapotot a szerveren
    - pl. GET, HEAD, OPTIONS, TRACE.
  - A kliens újra próbálkozhat.
- **Idempotens (idempotent) metódus**
  - Többszöri végrehajtása ugyanazt a hatást váltja ki, mint az egyszeri. (pl. PUT, DELETE)
    - DELETE nem dobhat hibát, hogy az erőforrás nem található.
  - Minden biztonságos metódus egyben idempotens is.
  - POST tipikusan nem (pl. fórum hozzászólás felküldése).
    - POST-Redirect-GET (PRG) pattern.

A kért erőforrás (resource)

- **Uniform Resource Identifier (URI)** azonosítja
- *„A Uniform Resource Identifier (URI) is a compact sequence of characters that identifies an abstract or physical resource.”* (RFC 3986, 61.old)
- **[uri_scheme]:[scheme specific part]**
  - tel:+36 1 4633714
  - mailto:John.Doe@example.com
  - http://www.bme.hu
- Az „URI” elnevezés javasolt az „URL” helyett.

Uniform Resource Locator (URL) RFC 3986

- Speciális URI weboldalak címzésére.
- Meghatározza az erőforrás elérését (location) is.

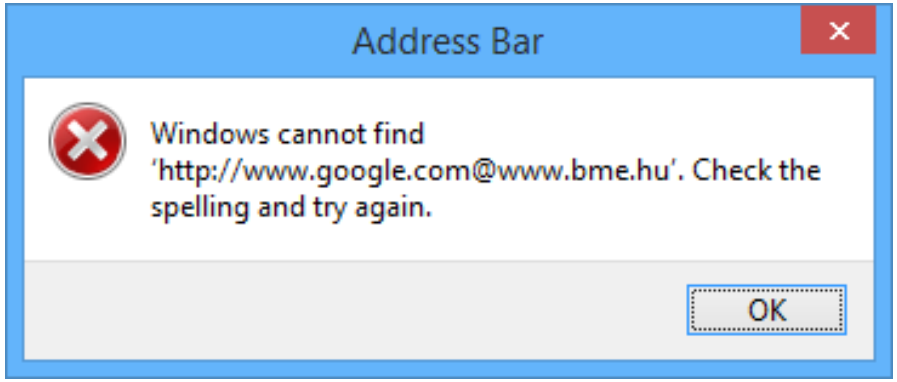


- A fragment nem jut el a szerverre, a kliens dolgozik vele.
- Gyakorlatban használt általános forma:

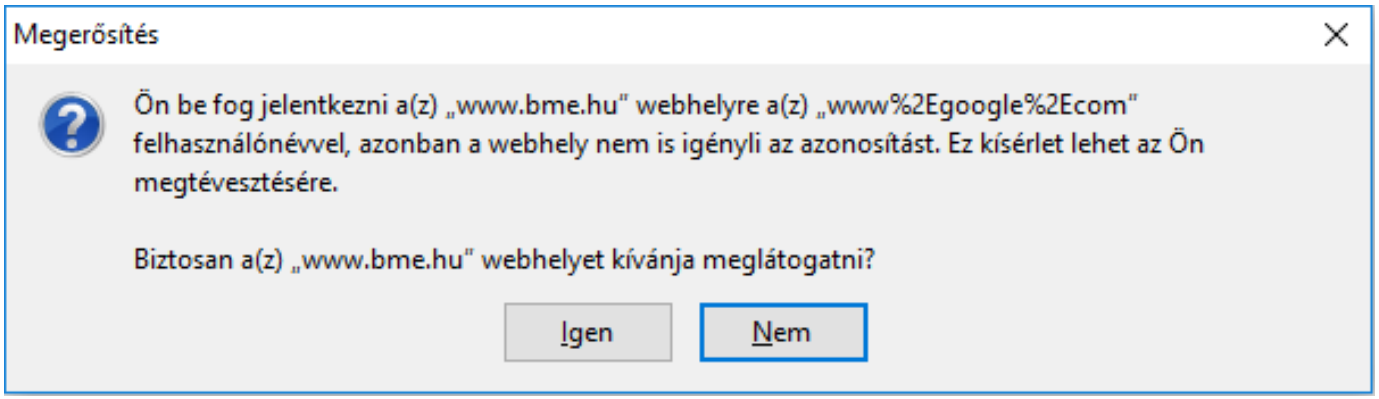
protocol://username:password@FQDN:port/path/file
?variable1=value1&variable2=value2#name

Adathalász (phishing) támadások

- URI: http://www.google.com@www.bme.hu
- Chorme betölti a www.bme.hu-t
- Internet Explorer



- FireFox



URL fajtái

- **Absolute URL**: mindentől függetlenül egyértelműen meghatározza az erőforrást
  - http://www.bme.hu/hirek
- **Relative URL**: az aktuális dokumentumhoz vagy a szerver gyökeréhez (root relative) képest relatív út
  - /Oktatas/Lists/Szakiranyok
  - Image%20Library/BulletinImage.jpg
- Általában case-sensitive
  - szerver beállítás és kódolás kérdése

Fejléc mezők (RFC 2616 Section 14)

- **Szerverrel** kapcsolatos mezők
  - `Date: Wed, 21 Aug 2013 08:41:30 GMT`
  - `Server: Apache`
- **Tartalommal** kapcsolatos mezők (például):
  - `Accept: text/html, image/jpeg`
  - `Accept-Encoding: gzip, deflate`
  - `Accept-Language: en-US,hu-HU;q=0.5`
  - `Content-Length: 3495`
  - `Content-Type: text/html`
  - `Content-Disposition: mentendő fájl neve`
  - `Content-Encoding: gzip`

Fejléc mezők (RFC 2616 Section 14)

- **Gyorsítótárral** (cache) kapcsolatos mezők (például):
  - `Cache-Control: no-cache`
  - `Expires: dátum`
  - `If-Modified-Since: dátum`
  - `Last-Modified: dátum`
  - `ETag: verzió`
- **Biztonsággal** kapcsolatos mezők (például):
  - `Authorization: Basic` `TXlEb21haW5cTXlDb21wdXRlcjpTdXBlclNlY3JldFBhc3N3b3Jk`
  - `WWW-Authenticate: Basic realm="MyComputer"`
  - `X-Frame-Options: SAMEORIGIN`
  - `DNT: 1`

Fejléc mezők (RFC 2616 Section 14)

- Referer: http://www.google.com/?k=szó
  - Helyesen "referrer", de így került a specifikációba.
- User-Agent: Mozilla/4.0+(compatible;+MSIE+6.0;+Windows+NT+5.1;+SV1)
  - **User agent sniffing**: más tartalom megjelenítése különböző klienseknek
    - pl. mobil/desktop böngésző
  - **User agent spoofing**: a User-Agent mező meghamisítása

Hibakódok (Status-Code) RFC 2616 Sec. 10

- Teljes lista: http://support.microsoft.com/kb/943891
- **1xx: Information**
  - 100 Continue
  - 101 Switching protocols (ld. WebSocket)
- **2xx: Successful**
  - 200 OK
  - 201 Created (ld. REST)
  - 204 No content
- **3xx: Redirect**
  - 301 Moved permanently
  - 302 Found (temporary move)
  - 304 Not modified

Hibakódok (Status-Code) RFC 2616 Sec. 10

- **4xx: Client Error**
  - 400 Bad request
  - 401 Unauthorized
  - 403 Forbidden
    - 403.5: SSL required
    - 403.6: Forbidden: IP address rejected
  - 404 Not found
  - 405 Method not allowed
  - 410 Gone
  - 413 Request entity too large
  - 414 Request URI too long
- **5xx: Server Error**
  - 500 Internal server error
  - 503 Service unavailable

Hibaüzenetek (Reason-Phrase)

- A szabványban csak javaslatok vannak.
- A szerver küldhet vissza egyedi hibaoldalt is.
- A böngésző jeleníthet meg barátságos üzenetet.
  - Internet Options → Advanced → Browsing → Show friendly HTTP error messages
  - IIS Manager: Error Pages

Állapotkezelés

A HTTP állapotmentes, de webes szükség van a kérések közötti állapot megőrzésére…

Probléma

- HTTP állapotmentes (stateless)
  - Az egyes kérés-válasz párok között a protokoll nem biztosít állapotmegőrzést.
  - Nem alakul ki munkamenet (session).
- Miért van szükség állapotkezelésre?
  - elég egyszer bejelentkezni egy webalkalmazásba.
  - webáruházban megmarad a kosár tartalma.
  - testreszabási beállítások megmaradnak.
  - „memory for websites”

Megoldási lehetőségek (kliens)

A munkamenethez (session) tartozó információk minden kérésnél és válasznál utaznak a böngésző és a szerver között.

- Előny: Nem igényel szerver oldali erőforrást
  - sok felhasználóra jól skálázódik.
- Hátrányok:
  - A tárolható adatok mérete korlátozott
    - adatmennyiségre nem jól skálázódik.
  - Az adatok mindig utaznak a hálózaton
    - sávszélesség pazarló.
  - Az adatok láthatóak egy MITM támadó számára
    - nem biztonságos.

Állapot tárolási lehetőségek (kliens)

- URL paraméterben.

```
http://www.aut.bme.hu?page=2
```

- Rejtett mezőben.

```
<input type="hidden" name="id" value="2">
```

- **Cookie**-ban.
- HTML 5 esetén:
  - Local storage és session storage.
  - IndexedDB.
  - File system.

Mire kell figyelni? (kliens oldalon)

- Támadó látja az adatokat (eavesdropping)
  - Megoldás: HTTPS.
- Támadó megváltoztathatja az adatokat (tampering)
  - Megoldás: digitális aláírás.
- Az adatok elveszhetnek (pl. elszáll a böngésző vagy a felhasználó manuálisan törli az adatokat)
  - Megoldás: fallback mechanizmus.
- Korlátozott méretek.
- HTML 5 technológiákat nem minden böngésző támogatja vagy nem ugyanúgy támogatja
  - Folyamatosan változik.

Megoldási lehetőségek: kliens és szerver

A munkamenethez tartozó **információk a szerveren tárolódnak**, csak a munkamenet azonosítója (session ID) utazik a böngésző és a szerver között.

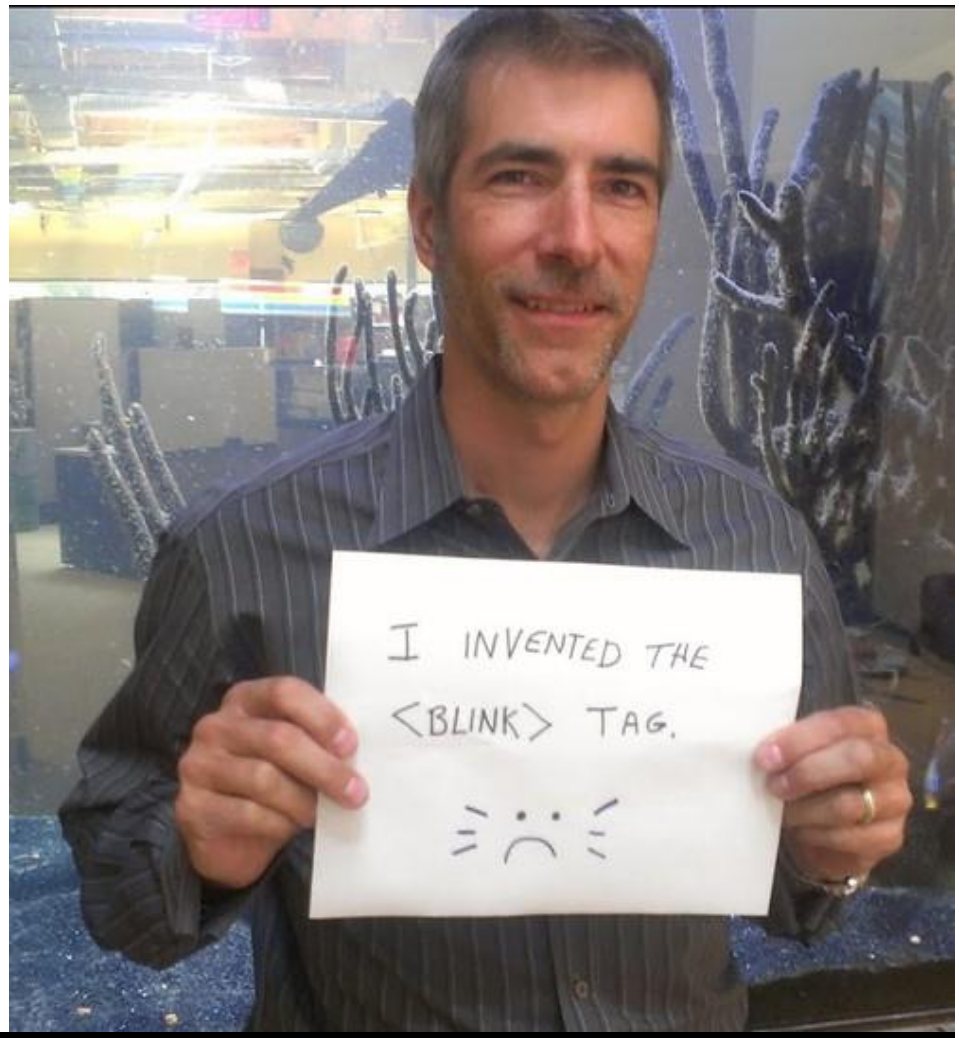
- Előny: ami a kliensoldali megoldásnál hátrány volt.
- Hátrány:
  - Memória igény
    - sok felhasználóra nem jól skálázódik.
  - Szerver farm esetén
    - vagy intelligens terheléselosztás (server affinity) kell,
    - vagy state server → single point of failure.

Mire kell figyelni? (szerver oldal)

- Skálázódási problémák sok felhasználó esetén
  - > nehéz tesztelni.
- A webalkalmazás vagy a webszerver bármikor újraindulhat (pl. process crash, OS upgrade).
- Szerver farm esetén a terheléselosztás problémái (server affinity vagy state server).

Lou Montulli

- 1991-ben Lynx böngésző
- BLINK tag
- animáló GIF
- Második webkamera.



http://www.montulli-blog.com/2013/05/the-reasoning-behind-web-cookies.html

Cookie

RFC 6265: **HTTP State Management Mechanism**
(2011. április)

- Cél: memória a HTTP-hez.
- Ötlet volt, hogy a böngészőknek legyen egyedi azonosítója
  - de akkor nyomonkövethetők a felhasználók → elvetették.
- SessionId készítése, leküldése a böngészőnek, amit az visszaküld mindig a szervernek.
- Úgy készüljön, hogy cross site tracking-et ne engedje.
- A mai süti az akkori koncepció kb 95%-át megtartotta.
- A sütibe kerülhet egy random sessionID, username, és bármi, csak ne legyen nagy!
- Törlődjenek
  - ha bezárják a böngészőt, vagy újraindul a gép.

3rd party cookie

- Probléma lett 1996 körül
  - alapvetően a sütit nem arra tervezték, hogy nyomon kövesse a felhasználót oldalakon keresztül
- Két megoldás merült fel megoldási lehetőség
  - Továbbra is engedélyezni a 3rd party sütiket
    - így láthatóak maradnak a cégek, akik nyomon követik a felhasználói szokásokat
    - kormányzat az adatgyűjtést tudja szabályozni ha akarja.
  - Vagy letiltani az egészet
    - akkor majd kitalálnak valami mást, amit lehet, hogy nehéz lesz észrevenni is.

Cookie típusai

- **Session (in-memory/transient) cookie**
  - csak a munkamenet idejére létezik, a böngésző bezárásával törlődik.
  - Több böngésző ablak osztozik rajta.
- **Permanent (persistent) cookie**
  - diszkre mentődik.
  - „Remember me” checkbox a bejelentkező oldalakon.

Cookie tartalma

Szöveges tartalom, nem futtatható, de privacy problémát jelenthet.

- **Name**: süti neve, ezzel tudunk rá hivatkozni.
- **Value**: az eltárol érték sztring fomátumban.
- **Expiration date**: Süti lejárai ideje
- **Path**: URL-ben minek kell szerepelnie, hogy elküldje a sütit a böngésző.
  - Alapértelmezés szerint: "/"
- **Domain**: Melyik hostokra kell elküldeni.
  - Ha nincs megadva, akkor ahonnan letöltöttük az oldalt (subdomainek nélkül)
- **Secure**: Csak HTTPS-en keresztül használható.
- **HttpOnly**: Kapcsoló, hogy ne lehessen JS-ből módosítani.

Sütihez kapcsolódó HTTP fejlécek

- **Set-Cookie**
- **Cookie**

- Cookie törlésére nincs külön fejléc
  - felülírás üres tartalommal, elmúlt lejárati dátummal.
- A böngésző minden alkalommal visszaküldi a szerverre ha a domain és path egyezik.
  - Akkor is, ha az adott HTTP kéréshez nem kellene
    - pl. CSS → cookieless domain.

Biztonság

- Nyílt szövegként utazik
  - tartalom titkosítása, HTTPS + **Secure** flag beállítása
- Nyílt szövegként tárolhatja a kliens (privacy)
  - tartalom titkosítása
- Változtatható a tartalma
  - integritás ellenőrzés, digitális aláírás, HMAC
- Nem garantálható az eredete
  - Nem csak a szerver hozhatja létre
    - digitális aláírás.
  - Máshonnan is visszaküldhető (session hijacking)
    - Klienshez kötés, IP cím a **value** mezőbe?

Biztonság

- Script hozzáférhet és módosíthatja
  - XSS (Cross-site scripting) támadás → **HttpOnly** flag
- A perzisztens süti a böngésző bezárása után, a felhasználó akarata ellenére is eljuthat a szerverre
  - XSRF (Cross-site request forgery) támadás.
- Cookie store:
  - Több böngésző esetén több cookie store.
  - Több operációs rendszer felhasználó esetén több store.
  - „Private browsing”

shell:cookies

- IE: C:\Users\<USER>\AppData\Roaming\Microsoft\Windows\Cookies
- FF: C:\Users\<USER>\AppData\Roaming\Mozilla\Firefox\Profiles\<ID>\cookies.sqlite
- Chrome: C:\Users\<USER>\AppData\Local\Google\Chrome\User Data\Default\Cookies (SQLite)