WEBFEJLESZTÉS 2. – MUNKAMENET-KEZELÉS, HITELESÍTÉS

Horváth Győző

Egyetemi adjunktus

1117 Budapest,
Pázmány Péter sétány 1/C, 2.420
Tel: (1) 372-2500/1816

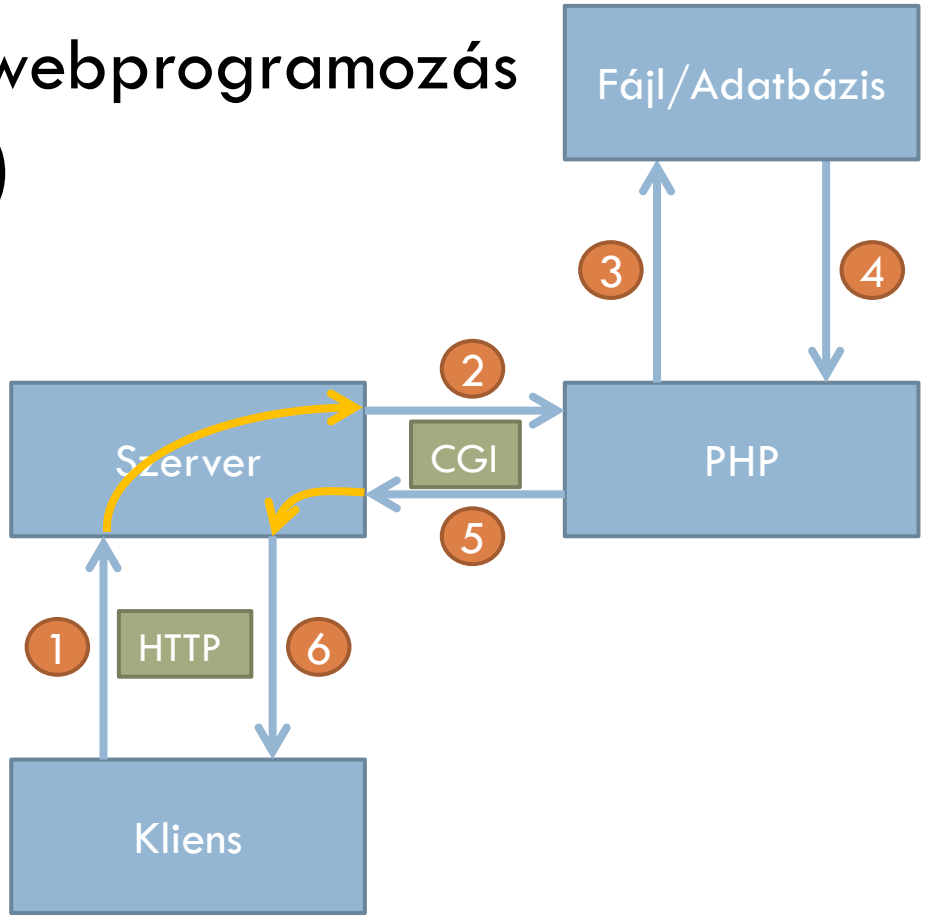
PHP beadandó

- Honlapról elérhető
- Labirintus-játék szerveroldali funkcionalitása
  - tárolás (fájlban)
  - listázás
  - szerkesztés
  - AJAX
- Határidő: 2014. május 25. éjfél

Ismétlés

Ismétlés

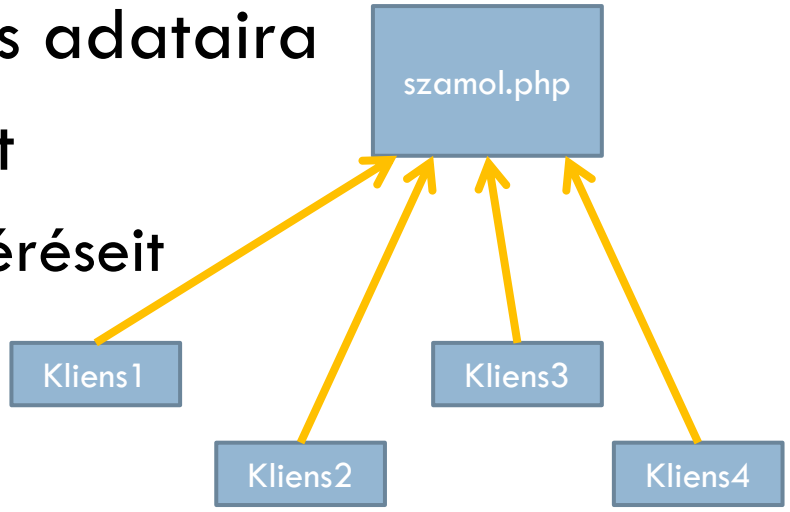
- Dinamikus szerveroldali webprogramozás
- Output (HTML generálás)
- Input (link, űrlap)
- Adattárolás: fájlok
  - Fájlműveletek
    - alacsony szintű műveletek
    - magas szintű műveletek
  - Fájlszerkezet-vezérelt
  - Adatszerkezet-vezérelt
    - Sorosítás
- Kódszervezés (logikailag, fizikailag)

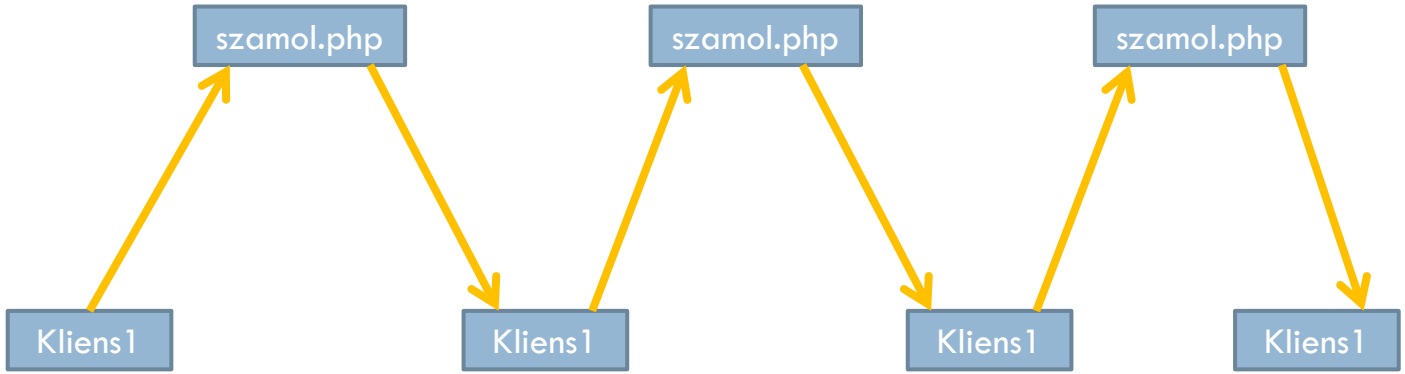


Munkamenet-kezelés

HTTP állapotmentesség

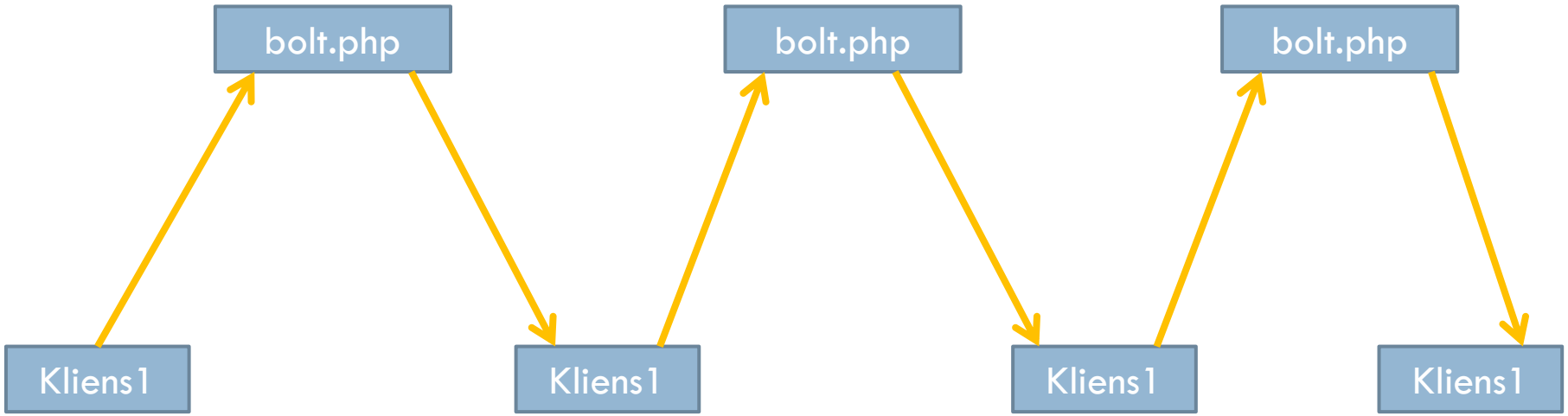
- A HTTP állapotmentes protokoll
- Nem emlékezik az előző kérés adataira
- Függetlenül kezeli a kéréseket
  - Ugyanazon kliens különböző kéréseit
  - Különböző kliensek kéréseit



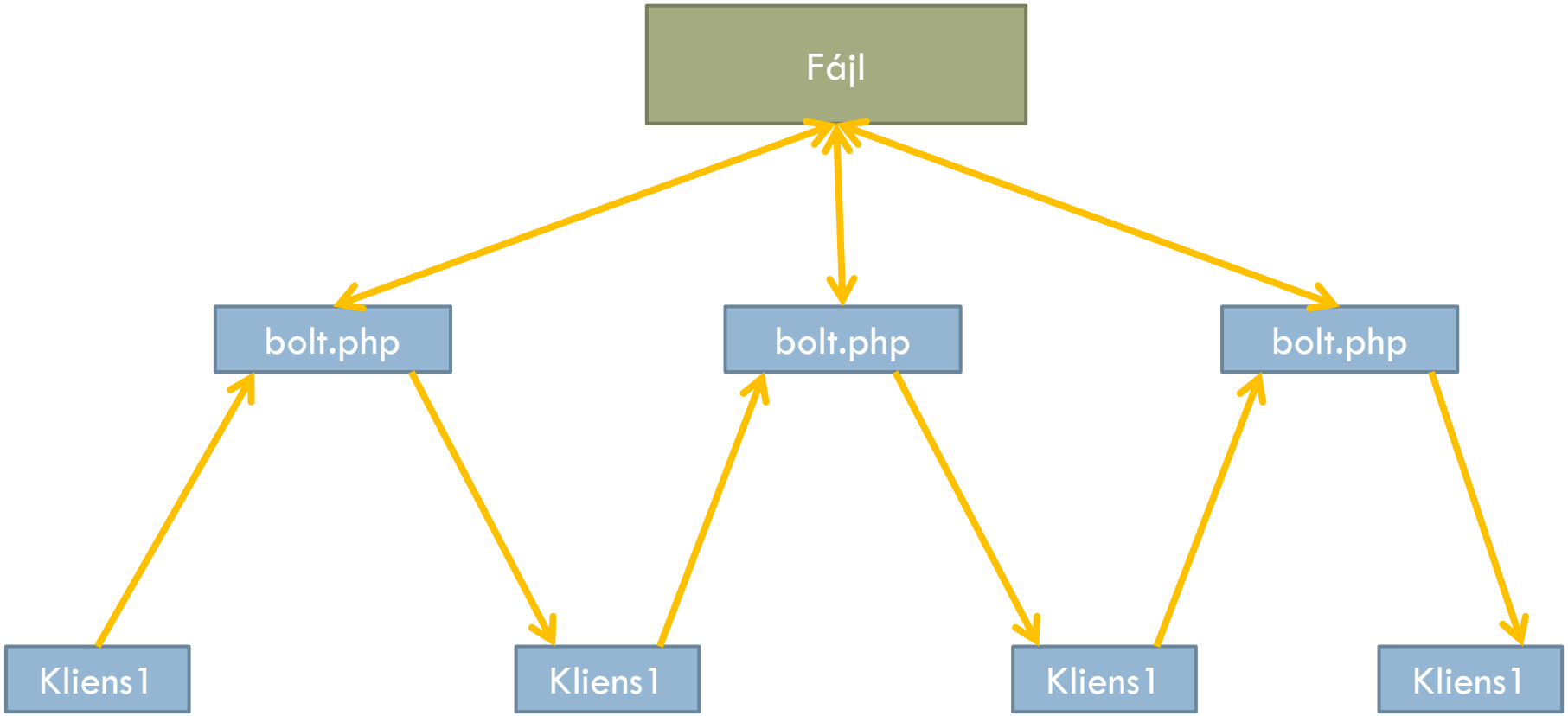


1. probléma

- Ugyanazon kliens több kérése között az állapot megtartása
- Pl. kosár
- HTTP kéréstől külön tárolni

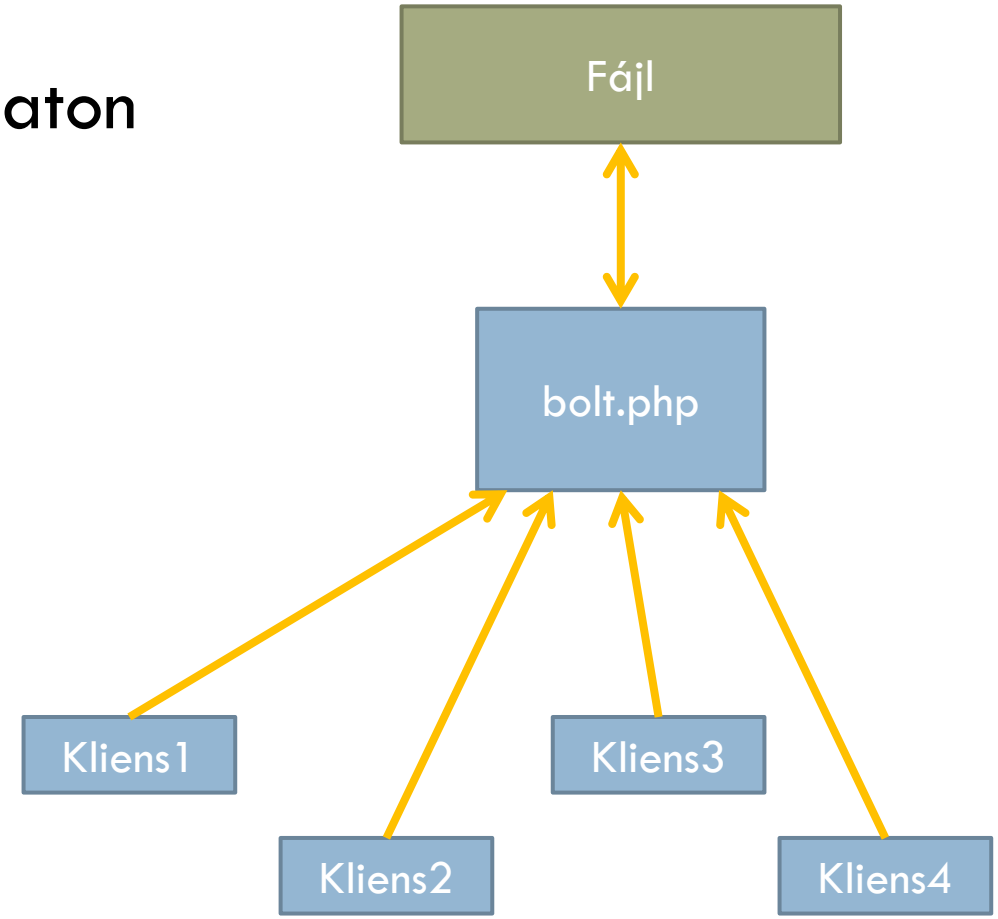


1. probléma


Fájl
bolt.php
bolt.php
bolt.php
Kliens1
Kliens1
Kliens1
Kliens1

2. probléma

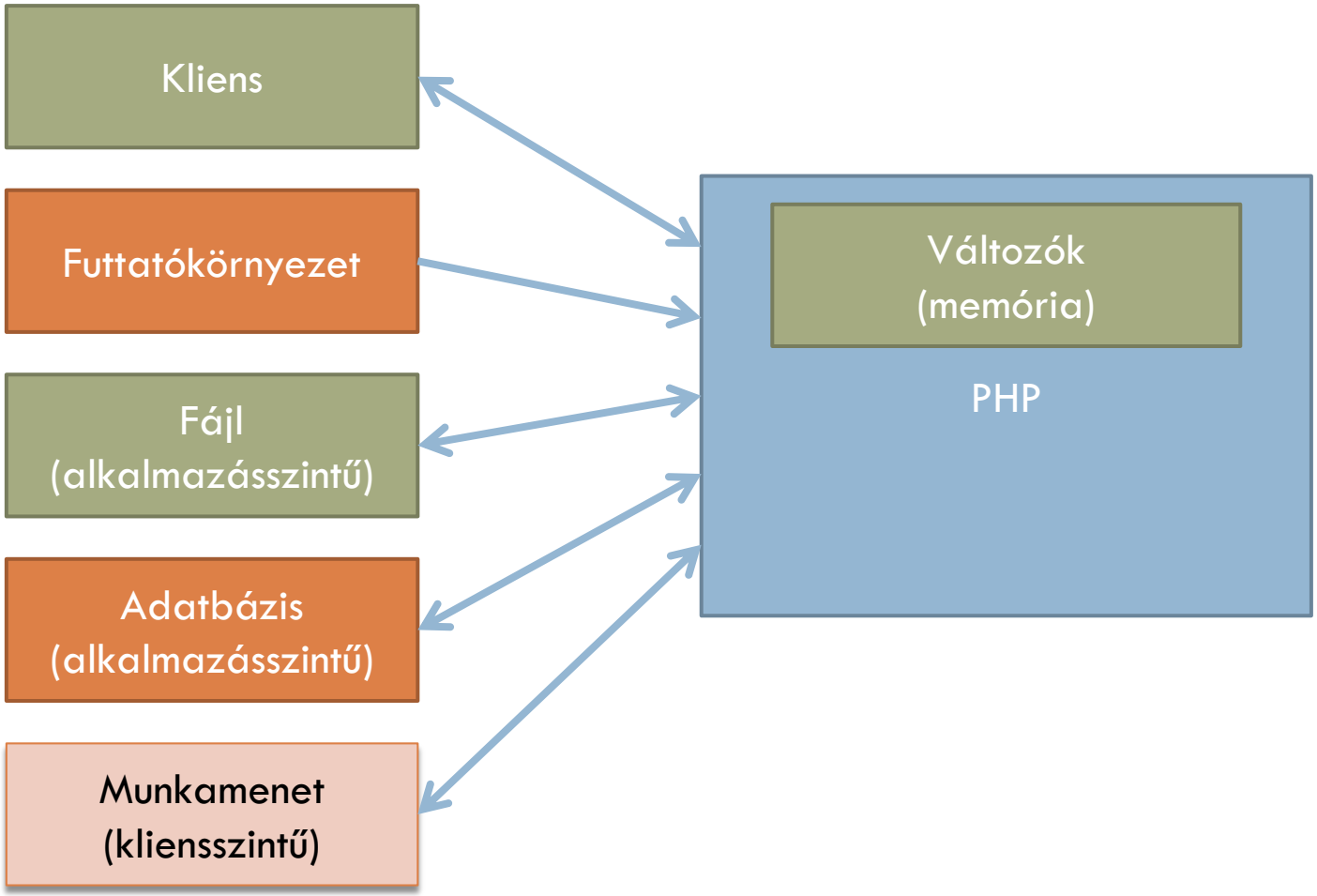
- Mindegyik kliens ugyanazon az adaton osztozkodik



Munkamenet-kezelés

- Kliensek megkülönböztetése
- Kliensenkénti adattárolás
- Példák:
  - levelezés, dokumentumok
  - internetbank
  - webáruház kosara, online szerkesztők (Doodle)
- Megoldás
  - kliens oldalon
  - szerver oldalon

PHP programok adatforrása



Munkamenet-kezelési lehetőségek

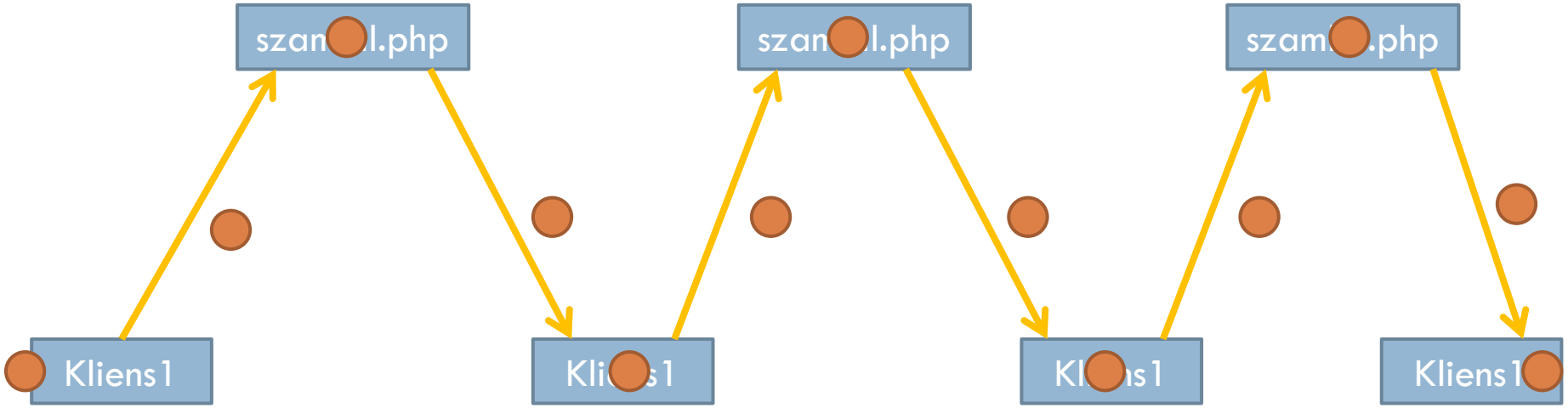
Példa

- Tároljuk egy számláló értékét felhasználónként, és minden kérésnél növeljük a számláló értékét eggyel!

Kliens oldali állapottartás

- Az adatot a kliensen tároljuk
- Minden kérésnél felküldjük a szerverre
- A szerver visszaadja a kliensnek

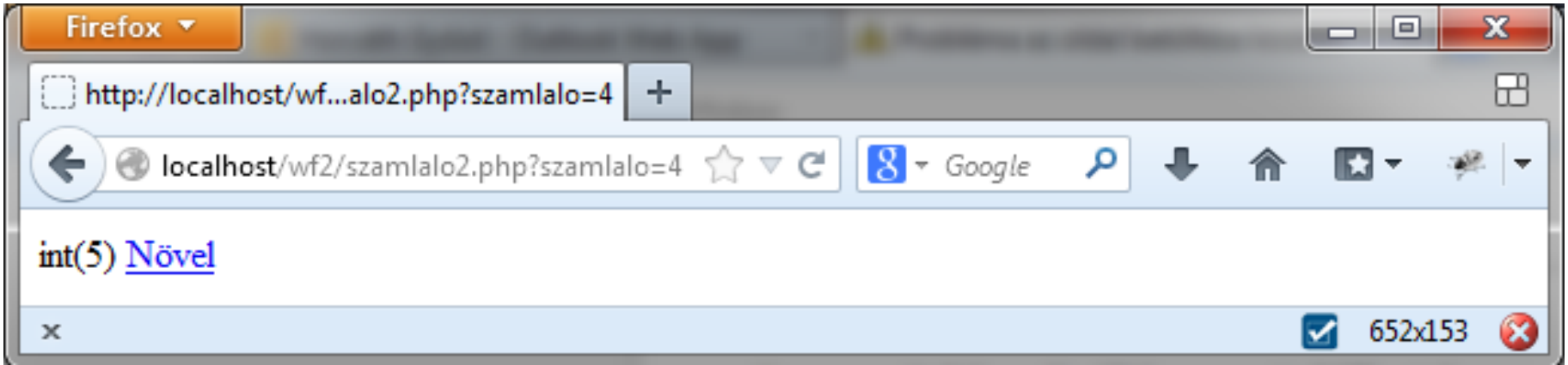
- Kliens oldali technológiák
  - URL
  - Rejtett mező
  - Süti



Állapottartás URL-ben

szamol.php?szamlalo=1

```php
<?php
$szamlalo = 0;
if (isset($_GET['szamlalo'])) {
  $szamlalo = $_GET['szamlalo'];
}
$szamlalo += 1;

var_dump($szamlalo);
?>
<a href="szamlalo2.php?szamlalo=<?php echo $szamlalo; ?>">Növel</a>
```



Állapottartás URL-ben

- Hátránya:
  - Minden linkhez oda kell generálni
    - Ha egyről is lemarad, elvész az adat
  - Sok adat nem fér el benne
    - URL hossza legfeljebb 2kB
  - Feltűnő (zavaró)
  - Könnyen átírható
  - Könyvjelzőzhető (nem mindig hátrány)

Állapottartás rejtett mezőben

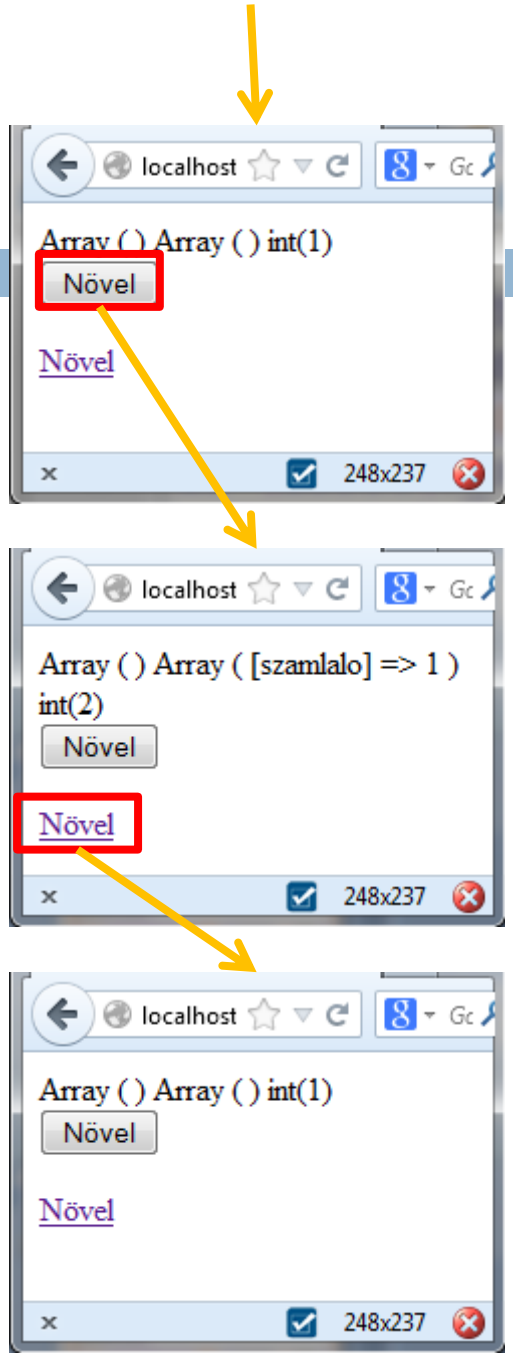
- URL: kevés adat, feltűnő, manipulálható
- → űrlap rejtett mezője

```
<input type="hidden" name="szamlalo" value="4">
```

- Előny
  - sok adat
  - nem feltűnő
- Hátrány
  - manipulálható
  - csak űrlapok esetén
  - normál linkeknél JavaScript kell

Példa – számláló


```php
<?php
$szamlalo = 0;
if (isset($_POST['szamlalo'])) {
  $szamlalo = $_POST['szamlalo'];
}
$szamlalo += 1;

var_dump($szamlalo);
?>
<form action="szamlalo3.php" method="post">
  <input type="hidden" name="szamlalo"
         value="<?php echo $szamlalo; ?>">
  <input type="submit" value="Növel">
</form>

<!-- Ez nem működik -->
<a href="szamlalo3.php">Növel</a>
```



Állapottartás sütivel

- Rejtett mező: macerás, manipulálható → süti
- HTTP kérés és PHP

```
Cookie: név1=érték1; név2=érték2; név3=érték3
```

```
$_COOKIES
```

- HTTP válasz és PHP:

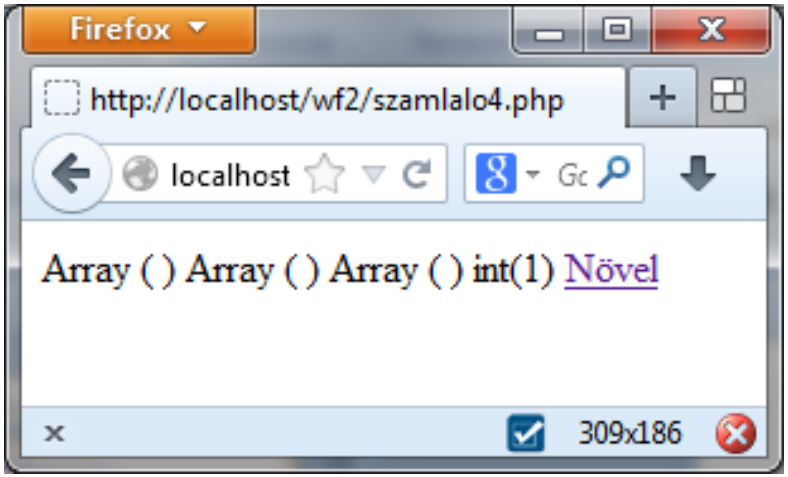
```
Set-Cookie: név=érték[; expires=dátum][; domain=domain][;
path=path][; secure]
```

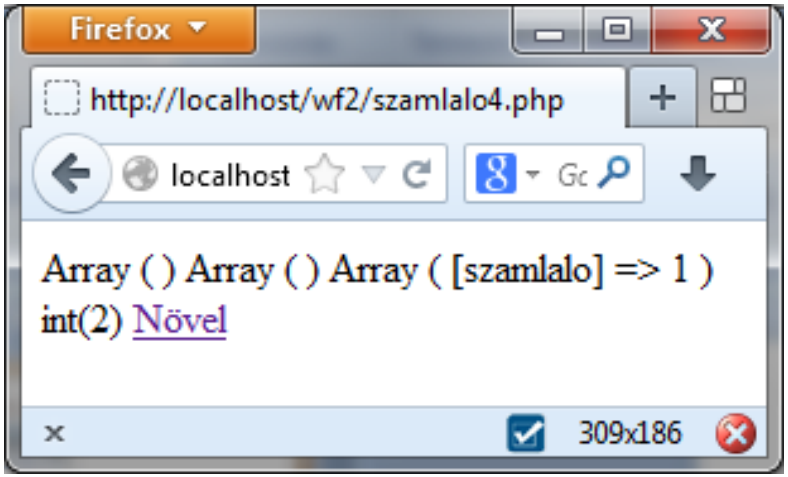
```
//Általános formája
$siker = setcookie($név[, $érték [, $expires = 0 [, $path [, $domain
[, $secure = false]]]]]);
//Néhány példa
setcookie('alma', 'piros');
setcookie('körte', 'sárga', time() + 60); //lejárat 60 mp múlva
```

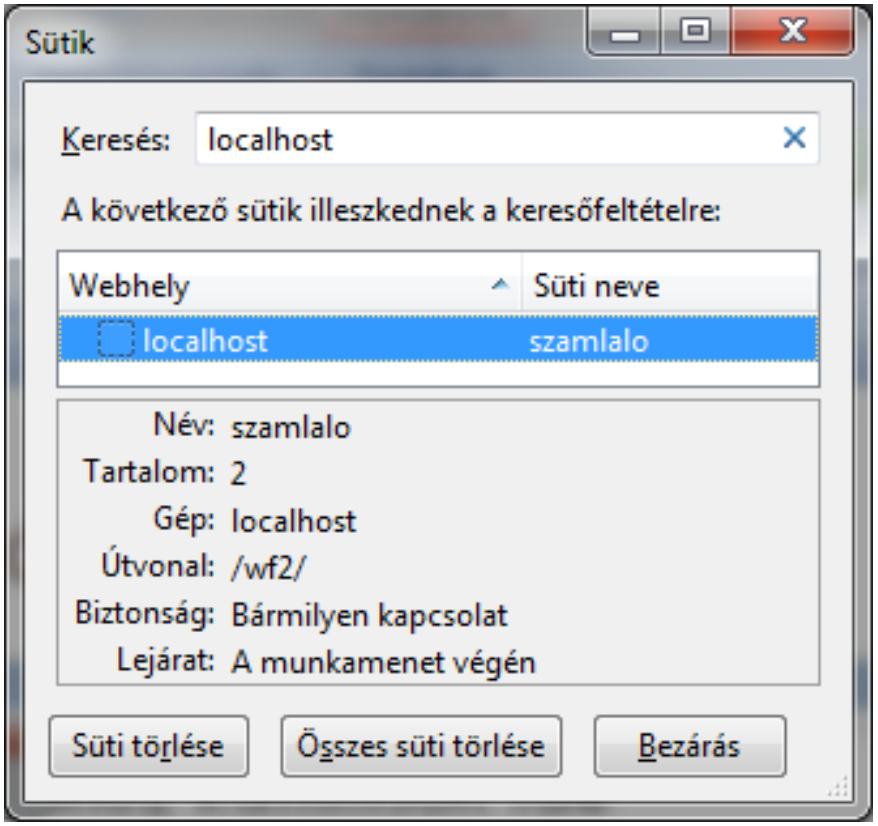
Példa – számláló


```php
<?php
$szamlalo = 0;
if (isset($_COOKIE['szamlalo'])) {
  $szamlalo = $_COOKIE['szamlalo'];
}
$szamlalo += 1;
setcookie('szamlalo', $szamlalo);

print_r($_GET);
print_r($_POST);
print_r($_COOKIE);

var_dump($szamlalo);
?>
<a href="szamlalo4.php">Növel</a>
```

Példa – számláló





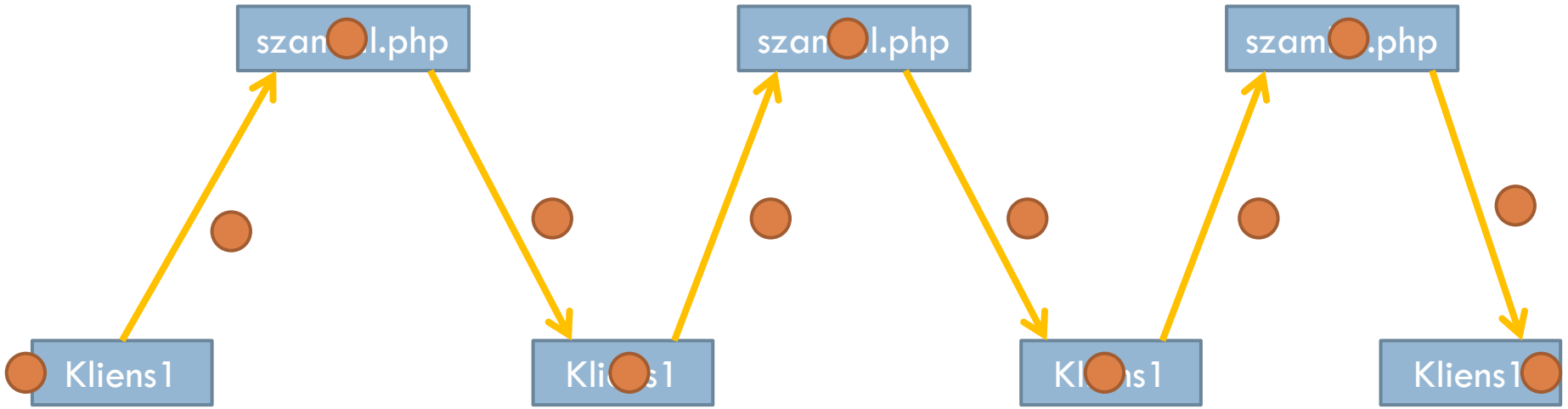


Állapottartás sütivel

- Előny
  - nem feltűnő
  - automatikus küldése
- Hátrány
  - manipulálható (kódolható)
  - limitált adatmennyiség
  - letiltható

Kliensoldali megoldások

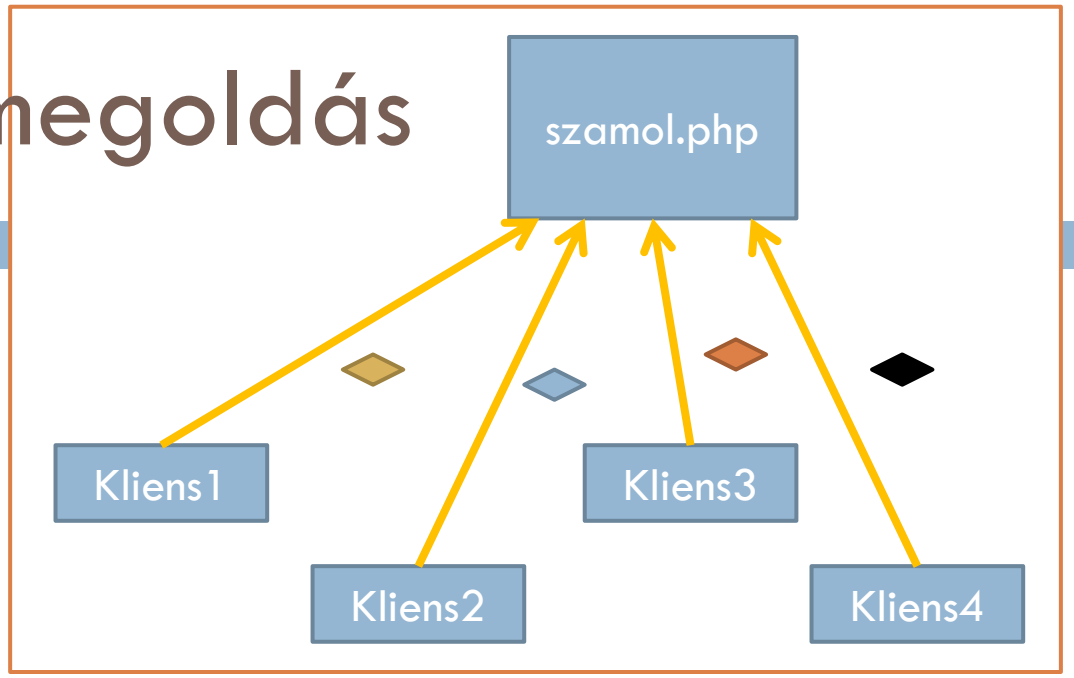
- Adat a kliensen van
- Manipulálható
- Sok adat esetén feleslegesen sok adat megy oda-vissza a kliens és szerver között

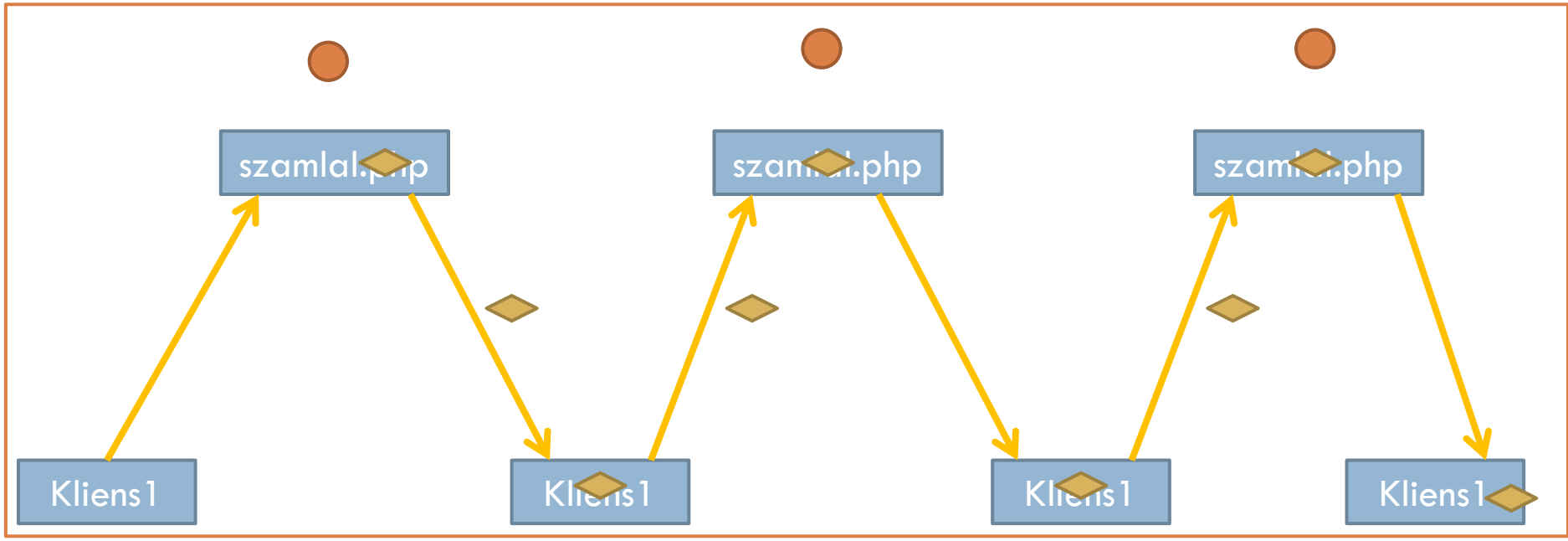


Szerveroldali megoldások

- Tároljuk az adatot a szerveren
  - → nem manipulálható kliens oldalon
  - → nem kell sok adatot küldözgetni
- A kliens megkülönböztetése továbbra is szükséges
- → tokent kap, amivel azonosítja magát és hozzáfér a tokenhez tartozó adatokhoz
- Token kliensoldali megoldással közlekedik
  - süti (alapértelmezett)
  - URL (ha nincs süti)

Szerveroldali megoldás





Szerveroldali megoldás

- Adatok tárolása
  - fájlban (ld. PHP)
  - adatbázisban
- Plusz erőforrás a szervertől
- A tokenre nagyon kell vigyázni!
  - ellopható
  - kilépés

Munkamenet-kezelés PHP-ban

Munkamenet-kezelés PHP-ban

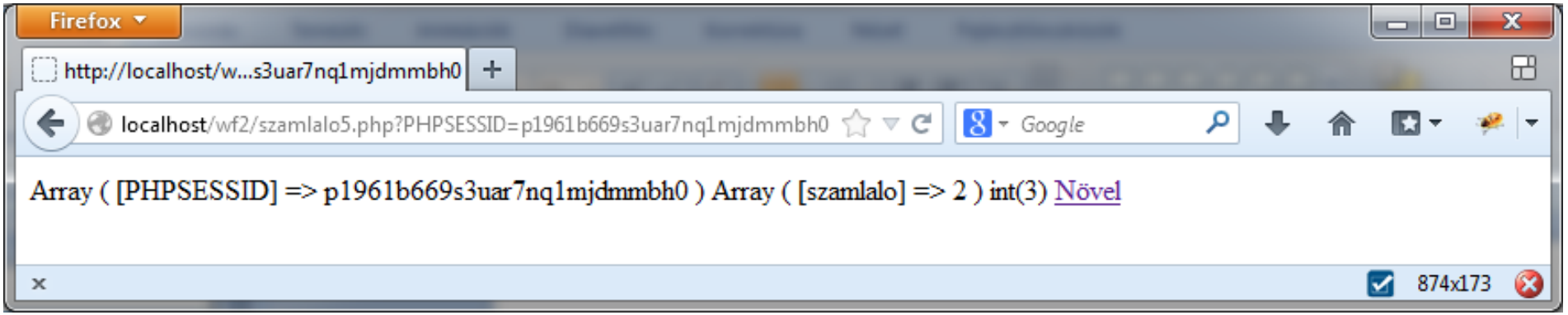
- Munkamenethez tartozó adatok:
  - $_SESSION
- Munkamenet-kezelő függvények:
  - session_start()
  - session_destroy()
  - SID konstans (PHPSESSID=token)

Példa – számláló


```php
<?php
session_start();

print_r($_GET);
print_r($_SESSION);

$szamlalo = 0;
if (isset($_SESSION['szamlalo'])) {
  $szamlalo = $_SESSION['szamlalo'];
}
$szamlalo += 1;
$_SESSION['szamlalo'] = $szamlalo;

var_dump($szamlalo);
?>
<a href="szamlalo5.php?<?php echo SID; ?>">Növel</a>
```

Példa - számláló

- Süti letiltása → token az URL-ben



Példa – számláló

- Munkamenet megszüntetése

```php
<?php
session_start();

print_r($_GET);
print_r($_SESSION);

$_SESSION = array();
session_destroy();

print_r($_SESSION);
?>
<a href="szamlalo5.php">Vissza</a>
```

Hitelesítés

Hitelesítés

- Ki használja az alkalmazást?
- → azonosított felhasználó
  - felhasználónév
  - teljes név
- → névtelen felhasználó
  - vendég
- Bizonyos oldalak csak azonosított felhasználók számára érhetőek el

Technológiák

- .htaccess, .htpasswd
  - könyvtár alapú védelem
- WWW-Authenticate
  - Módok
    - Basic
    - Digest (titkosított)
  - HTTP része
  - PHP-ból kiolvasható
- Ezek PHP-tól független technológiák
- Plusz adat tárolására nem alkalmasak
- Csak hitelesítésre

Hitelesítés munkamenettel

- Azonosított felhasználó munkamenetében egy speciális kulcsot helyezünk el
- Ezzel jelezzük, hogy már azonosítottuk
- Folyamat
  - beléptető űrlap (login form)
  - sikeres belépés esetén → kulcs
  - minden oldalon: ha ez a kulcs megvan, akkor azonosított
  - ettől függően más logika, más nézet lehet

Regisztráció

- Űrlap

```
<?php print_r($hibak); ?>
<form action="reg.php" method="post">
  Felhasználónév:
  <input type="text" name="felhnev"> <br>
  Jelszó:
  <input type="password" name="jelszo"> <br>
  <input type="submit" name="reg" value="Regisztrál">
</form>
```

Regisztráció


```php
$hibak = array();
if ($_POST) {
  $felhnev = trim($_POST['felhnev']);
  $jelszo = $_POST['jelszo'];
  $jelszavak = fajlbol_betolt('jelszavak.txt');
  if (strlen($felhnev) == 0) {
    $hibak[] = 'Nincs felhnev!';
  }
  if (strlen($jelszo) == 0) {
    $hibak[] = 'Nincs jelszo!';
  }
  if (array_key_exists($felhnev, $jelszavak)) {
    $hibak[] = 'Letezo felhnev!';
  }
  if (!$hibak) {
    $jelszavak[$felhnev] = md5($jelszo);
    fajlba_ment('jelszavak.txt', $jelszavak);
    header('Location: login.php');
    exit();
  }
}
```

Beléptetés

- Űrlap

```
<?php print_r($hibak); ?>
<form action="login.php" method="post">
  Felhasználónév:
  <input type="text" name="felhnev"> <br>
  Jelszó:
  <input type="password" name="jelszo"> <br>
  <input type="submit" name="belep"
value="Belép">
</form>
```

Beléptetés


```php
$hibak = array();
if ($_POST) {
  $felhnev = trim($_POST['felhnev']);
  $jelszo = $_POST['jelszo'];
  $jelszavak = fajlbol_betolt('jelszavak.txt');
  if (!(array_key_exists($felhnev, $jelszavak) &&
        $jelszavak[$felhnev] == md5($jelszo))) {
    $hibak[] = 'Nem jó!';
  }
  if (!$hibak) {
    $_SESSION['belepve'] = true;
    $_SESSION['felhnev'] = $felhnev;
    header('Location: szamlalo5.php');
    exit();
  }
}
```

Hitelesítés ellenőrzése

- munkamenet indítása
- létezik-e a kulcs a munkamenet adatai között
- kulcs értéke indifferens
- külön fájlba tehető

```
session_start();
function azonositott_e() {
  return isset($_SESSION['belepve']);
}
```

Kijelentkezés

- Kulcs kivétele a munkamenetből
- Általában az egész munkamenet megszüntetése (feladatfüggő)

```
unset($_SESSION['belepve']);
```

Fájlfeltöltés

Űrlap és feldolgozás


```php
<?php print_r($hibak); ?>
<form action="" method="post" enctype="multipart/form-data">
  <input type="file" name="fajl">
  <input type="submit" value="Feltölt">
</form>
```

```php
$hibak = array();
if ($_FILES) {
  if (array_key_exists('fajl', $_FILES) &&
      $_FILES['fajl']['error'] == 0) {

    $honnan = $_FILES['fajl']['tmp_name'];
    $hova = 'fajlok\\' . $_FILES['fajl']['name'];
    move_uploaded_file($honnan, $hova);
  } else {
    $hibak[] = 'Hiba tortent!';
  }
}
```

$_FILES


```
Array
(
  [fajl] => Array
    (
      [name] => Desert_small.jpg
      [type] => image/jpeg
      [tmp_name] => C:\eltescorm\tmp\php1750.tmp
      [error] => 0
      [size] => 18218
    )
)
```

Könyvtárlista


```
<ul>
<?php foreach($fajlok as $f) : ?>
  <li><?php echo $f; ?></li>
<?php endforeach; ?>
</ul>
```

```
function fajlok($kvt) {
  $fajlok = array();
  $d = opendir($kvt);
  while (($f = readdir($d)) !== false) {
    $fajlok[] = $f;
  }
  closedir($d);
  return $fajlok;
}
$fajlok = fajlok('fajlok\\');
```

Hasznos függvények

Tömbműveletek

- in_array($mit, $tömb): eldöntés (logikai)
- array_search($mit, $tömb): lineáris keresés (kulcs)
- sort($tömb): rendezés (többféle van)
- shuffle($tömb): keverés
- array_keys($tömb): $tömb kulcsai tömbként
- array_values($tömb): $tömb értékeit tömbként
- array_key_exists($kulcs, $tömb): $kulcs szerepel-e a $tömb tömb kulcsai között.

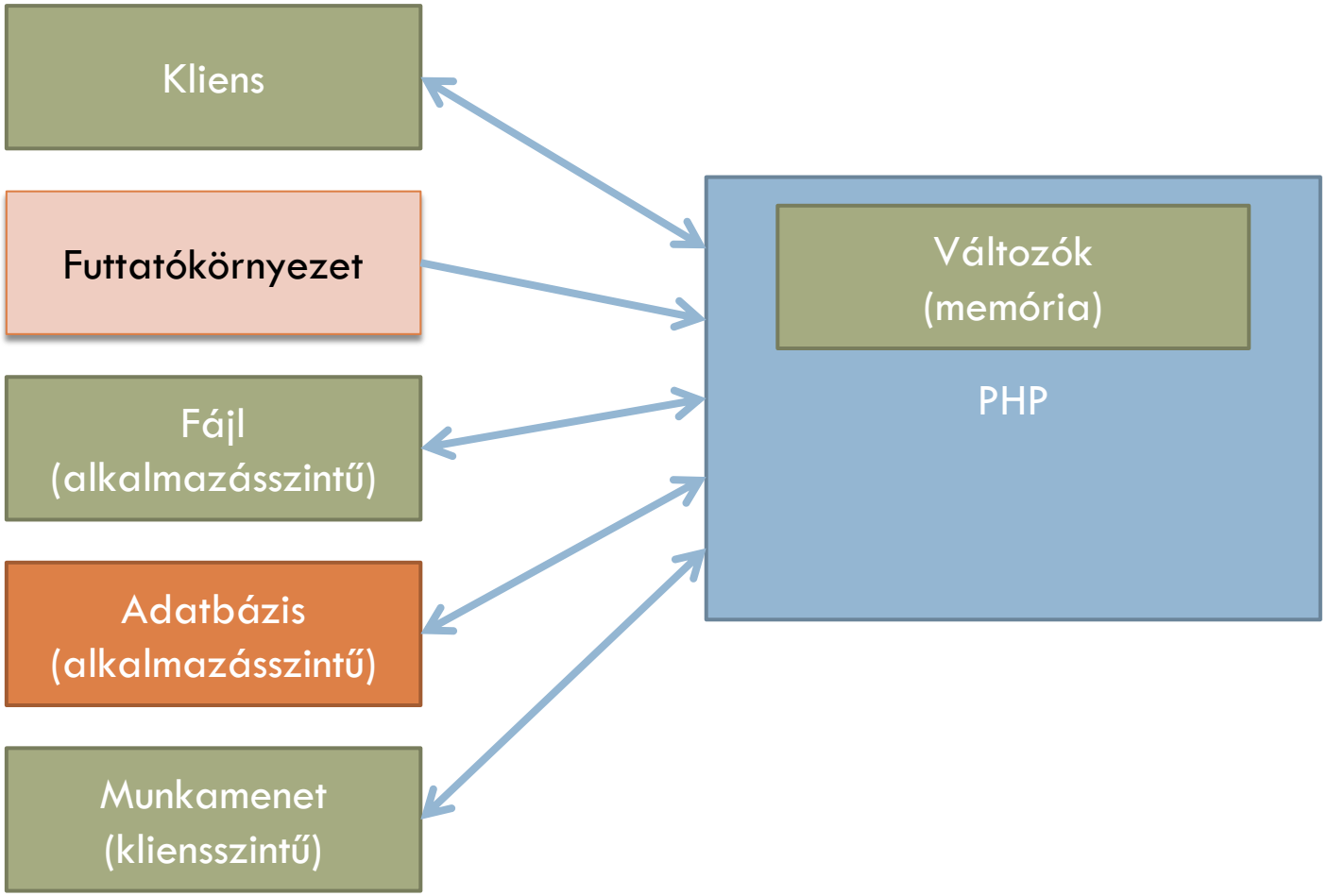
Szövegműveletek

- szöveg: karakterek tömbje
- explode($elválasztó, $s): elemekre bontás
- implode($elválasztó, $tömb): összefűzés
- substr($s, $kezdet, $hossz): részszöveg
- ltrim($s), rtrim($s), trim($s): fehérszóköz-eltávolítás.
- strstr($miben, $mit), strpos($miben, $mit), strrpos($miben, $mit): részszöveg keresése
- preg_match($minta, $s): regkif illeszkedése

Dátum és idő

- date($formátum): az aktuális idő kiírása a megadott formátumban.
- time(): Unix idő visszaadása másodpercben.
- strtotime($s): szövegként megadott dátum Unix időbe átalakítása.
- getdate(): dátuminformációk visszaadása tömbként.
- date_*: dátummal kapcsolatos további függvények

PHP programok adatforrása



További hasznos függvények

- header()
  - HTTP fejlécek leküldése
  - Pl.: header('Location: index.php');
    - átirányítás másik oldalra

Fejlesztői környezet

Fejlesztői környezet részei

- Szerver oldalon
  - webszerver
  - PHP
- Kliens oldalon
  - böngésző
- Fejlesztői környezet
  - Szerkesztő program (HTML, PHP, CSS)
  - SFTP, SCP kliens az állományok webszerverre töltéséhez

Otthoni fejlesztéshez

- XAMPP telepítőcsomag
  - http://www.apachefriends.org/en/xampp.html
  - Multiplatform
- WAMP Stack, LAMP Stack
  - http://bitnami.com/stacks/infrastructure