

1. modul - Webalkalmazások fejlesztése

ANTAL Margit

Sapientia - EMTE, Pannon Forrás „Egységes erdélyi felnőttképzés a Kárpát-medencei hálózatban”

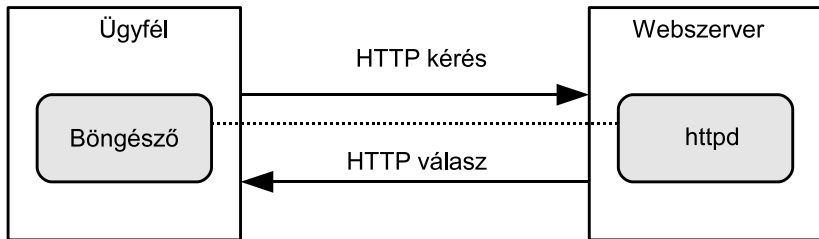
2010

- [Antal] Java alapú webtechnológiák, Scientia, Kolozsvár, 2009.
- [Sun] Sun Microsystems, Web Component Development with Servlet and JSP Technologies. SL-314, 2007.
- [Hunter] Jason Hunter, Java szervletek programozása, Kossuth kiadó, 2002.
- [Bergsten] Hans Bergsten, Java Server Pages, Kossuth kiadó, 2001.
- [Crawford] William Crawford, Jonathan Kaplan, J2EE Design Patterns, O'Reilly, 2003.

- Webalkalmazások fejlesztése
- Szervletek
- Menetek kezelése
- Eseménykezelők és szűrők
- JSP lapok
- JSP elemkönyvtárak
- Adatbázisok használata webalkalmazásokban
- A Struts keretrendszer
- Webalkalmazások biztonsága

- Webalkalmazás technológiák
- A Java EE 5 platform
- Webalkalmazások végrehajtása
- Szervlet technológia
- JSP technológia
- Háromrétegű webalkalmazások
- MVC architektúra

- HTTP, Hypertext Transfer Protocol
- HTML, Hypertext Markup Language



- Web hely:
 - HTML és más médiafájlok gyűjteménye
 - webserveren van elhelyezve
 - komplex könyvtárstruktúra jellemezheti
- Webalkalmazás:
 - web hely + programvégrehajtás szerver oldalon
 - dinamikus tartalom

- Java:
 - Programozási nyelv
 - Futtatási környezet (JRE)
- Java platformok:
 - Java SE - Java Standard Edition
 - Java EE - Java Enterprise Edition
 - Java ME - Java Mobile Edition

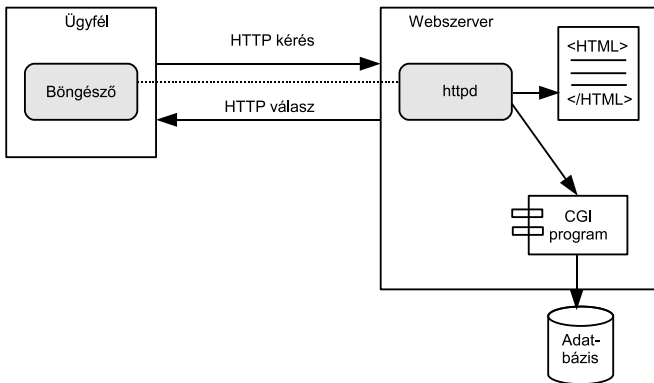
- Alkalmazások komponens alapú
 - tervezése - design
 - fejlesztése - development
 - összeállítása - assembly
 - telepítése - deployment
- többretegű osztott alkalmazásmodell
- újrafelhasználható komponensek
- egységesített biztonsági modell
- tranzakcióvezérlés
- web-szolgáltatások
- platformfüggetlenség

Webszerverek és kliensek

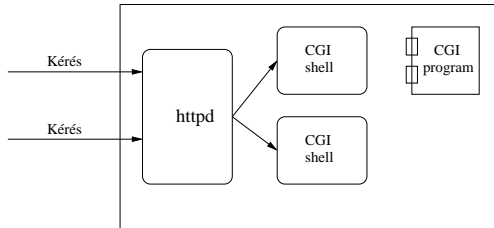
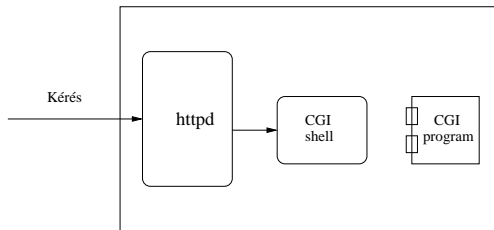
Cég	Szerver	Kliens
Apache	httpd	
Netscape	Netscape Server	Netscape Navigator
Microsoft	IIS	Internet Explorer
Mozilla		Firefox

CGI-Common Gateway Interface

- Szerveroldalon futtatott alkalmazás: szkript vagy végrehajtható állomány
- A szerver külön folyamatként futtatja

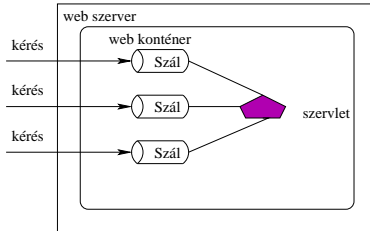
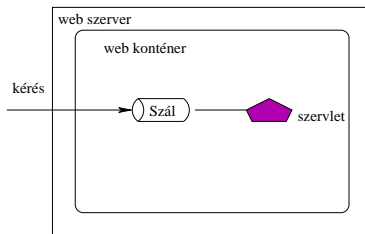


CGI programok



- Előnyök
 - Bármilyen programozási nyelvben elkészíthető (Perl a legelterjedtebb)
 - Hibás CGI program nem befolyásolja a webservert működését
 - Kényelmes a hívása - egy sor a weboldalon
 - Nincs konkurencia probléma - minden CGI program saját burokból fut
 - Minden szolgáltató támogatja
- Hátrányok
 - Időigényes - a burok létrehozása miatt
 - Nem skálázható
 - A CGI programnyelvek nem mindig biztonságosak
 - A szkript nyelvek néha platformfüggők is lehetnek

- **függőleges:** Ugyanazon egység kapacitásának növelése (pl. memória bővítése, új processzor hozzáadása, merevlemez hozzáadása)
- **vízszintes:** Több logikai egységből összeállítani egy olyan egységet, amely a felhasználók fele egy egységként látszik. Pl. klaszter, osztott fájlrendszer stb.



- Előnyök
 - Gyorsabb mint a CGI
 - Skálázható
 - Hibatűrő \Leftarrow Java nyelvben íródott
 - Platformfüggetlen \Leftarrow Java nyelvben íródott
 - Naplózási lehetőség
- Hátrányok
 - Keveredik a Java kód (logika) HTML kóddal (megjelenítés)
 - A weblap módosítása maga után vonja a szervlet újrafordítását és telepítését
 - Konkurencia problémákat kell kezelni

- A szervletek a szervletkonténerben futnak (szervlet motornak is nevezik)
- Webkonténer = Szervletkonténer = JVM + Servlet API

Java szervletek

Java kódba beágyazott HTML kód

JavaServer Pages - JSP

HTML oldalba beágyazott Java kód

- A JSP oldal szervletként fut
- Ami szervletként megvalósítható az megvalósítható JSP-ben is
- A JSP a prezentációs részre fókuszol

Egyszerű JSP lap

```
<table>
<% for( int i=0; i<10;++i){ %>
  <tr><td>
    <%=i%><%=i*i%><br>
  </td></tr>
<% } %>
</table>
```

- JSP - Java Server Pages
- PHP - ..HyperText Preprocessor
- ASP - Active Server Pages
- Ruby on Rails

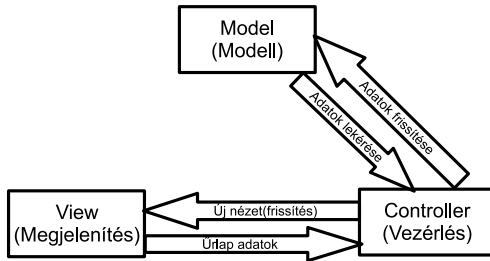
- Előnyök
 - Hatékony (szálak)
 - Platformfüggetlen
 - Kényelmes
- Hátrányok
 - Nehézkes hibakeresés
 - Az alkalmazás logika és megjelenítés nehezen választható szét
 - Konkurencia kezelés

Cél: az alkalmazás rugalmasságának növelése

- Ügyfél réteg
- Üzleti logika
- Adat (perzisztens)

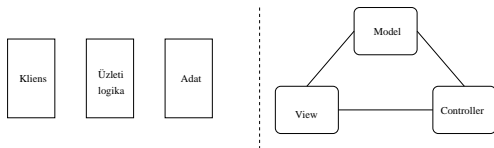
Model-View-Controller (MVC) architektúra

- **M**odel - Modell - Az adatok, illetve az ezeknek megfelelő adatstruktúrák, objektumok
- **V**iew - Megjelenítés - A modell megjelenítése
- **C**ontroller - Vezérlés - Események, amelyek hatására tevékenységet kell végezni a modellen



Megjegyzés

A három rétegű architektúra \neq MVC



- 1. Modell: A kérést egy JSP lap vagy egy szervlet feldolgozza, elvégzi az adatok érvényesítését, kezeli az alkalmazás logikáját és előállítja a kérésre a választ
- 2. Modell: Célja a feladatok jobb szétválasztása;
Megvalósítás: MVC architektúra;
 - **Model:** Java osztályok, megvalósítják az alkalmazás logikáját
 - **View:** JSP lap, előállítja a választ HTML formátumban; szükség esetén kommunikál a modellel adatok elérése céljából; az előállított űrlap megteremti az interaktivitás lehetőségét
 - **Controller:** A vezérlést egy szervlet végzi. Feladata a beérkező adatok ellenőrzése, a modell és a megjelenítés frissítése

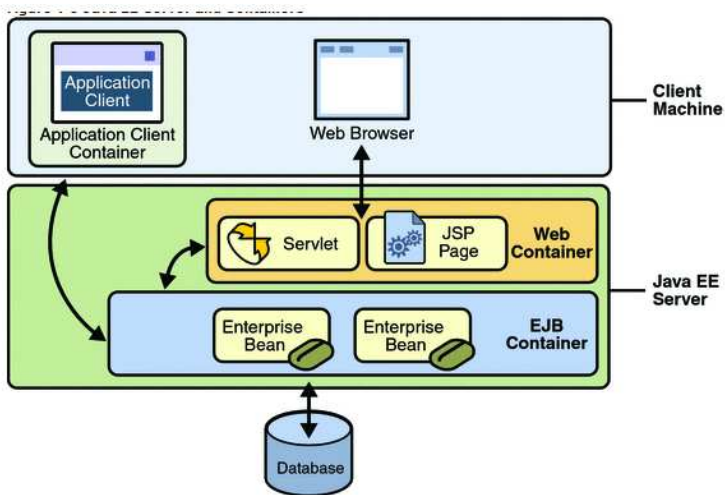
2. modell típusú létező keretrendszerek

- Struts - Jakarta
- JavaServer Faces - Sun
- Velocity - Apache

- EJB konténer - Nagyvállalati babok futtatása - A Java EE szerveren fut
- Webkonténer - JSP lapok és szervletek futtatása - A Java EE szerveren fut
- Kliens alkalmazás konténer - Kliens komponensek futtatása - Kliens oldalon fut
- Applet konténer - Appletek végrehajtása - Java kompatibilis böngésző kliens oldalon

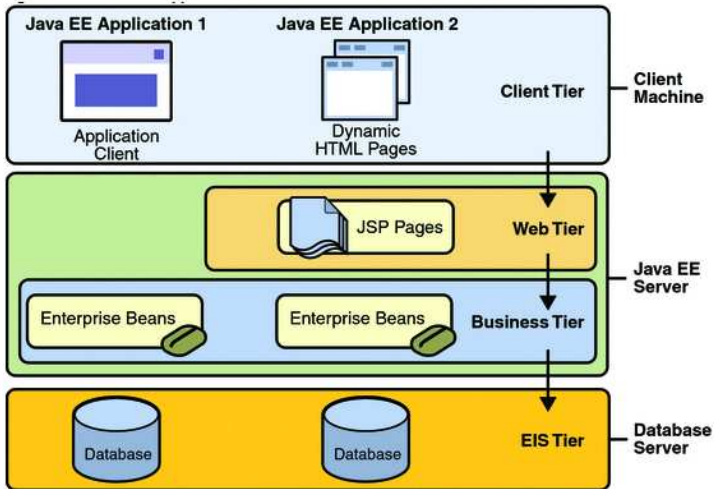
Java EE konténerek

[Java EE 5 Tutorial]



Java EE alkalmazások

[Java EE 5 Tutorial]



- Java Servlet 2.5
- JavaServer Pages 2.1
- JavaServer Pages Standard Tag Library
- JavaServer Faces 1.2

- tranzakciókezelés
- szálkezelés
- névszolgáltatás

- Webcentrikus:
 - Statikus HTML
 - HTML alapvető szervletekkel és JSP lapokkal
 - JSP JavaBeans komponensekkel
- Java EE centrikus: Strukturált webalkalmazások moduláris komponensekkel és Enterprise Bean-ekkel