

4. rész: Java Enterprise Edition bevezető

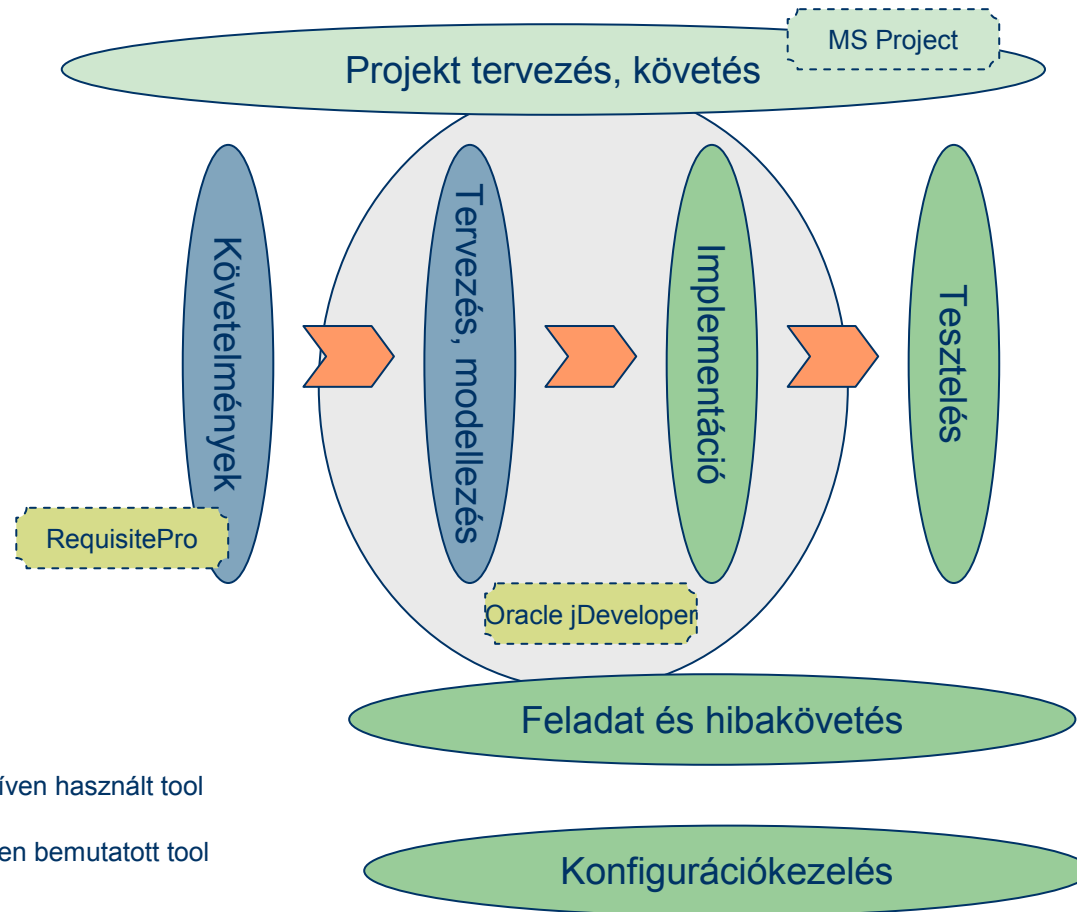
Bakay Árpád dr.

NETvisor kft

(30) 385 1711

arpad.bakay@netvisor.hu

Hol tartunk?



Tartalom

- Alkalmazás architektúrák
 - Komponensek
 - Többrétegű alkalmazások
 - Java Enterprise Edition
- Áttekintés: a Java Standard Edition érdekes szolgáltatásai

1. Komponensek

- Szoftver modulok, amelyek
 - Önálló logikai funkciót töltenek be
 - Explicit, kényelmes interfészekon csatlakoznak a környezethez
 - Koherencia, kohézió
- *„egy darab szoftver, amelyet változtatás nélkül használnak más, a komponens íróitól függetlenül elkészített alkalmazások,”*
 - Újrafelhasználhatók
 - Termékként, katalógusból értékesíthetők

- Martin Fowler, www.martinfowler.com

Mi a Java Enterprise Edition?

- Futtató környezet komponensek számára
 - Az alkalmazás logikája szabványos architektúrájú komponensekben
 - Java EE Tervezési minták – design patterns alapján
 - Premium szolgáltatások a környezettől
 - Szinte teljesen szabványos,
 - azaz egy alkalmazás bármely Java EE szerverben futtatható
- A Java Standard Edition-ra épül
 - Java SE: OO környezet -- Java EE: komponens környezet
- Logikusan következik belőle egy alkalmazás architektúra
 - Leginkább általános üzleti alkalmazásokhoz használható
 - Pl. real-time-hoz kevésbé.
 - Legfőbb jellegzetesség: „többrétegű” alkalmazások

A többrétegű architektúra kialakulása

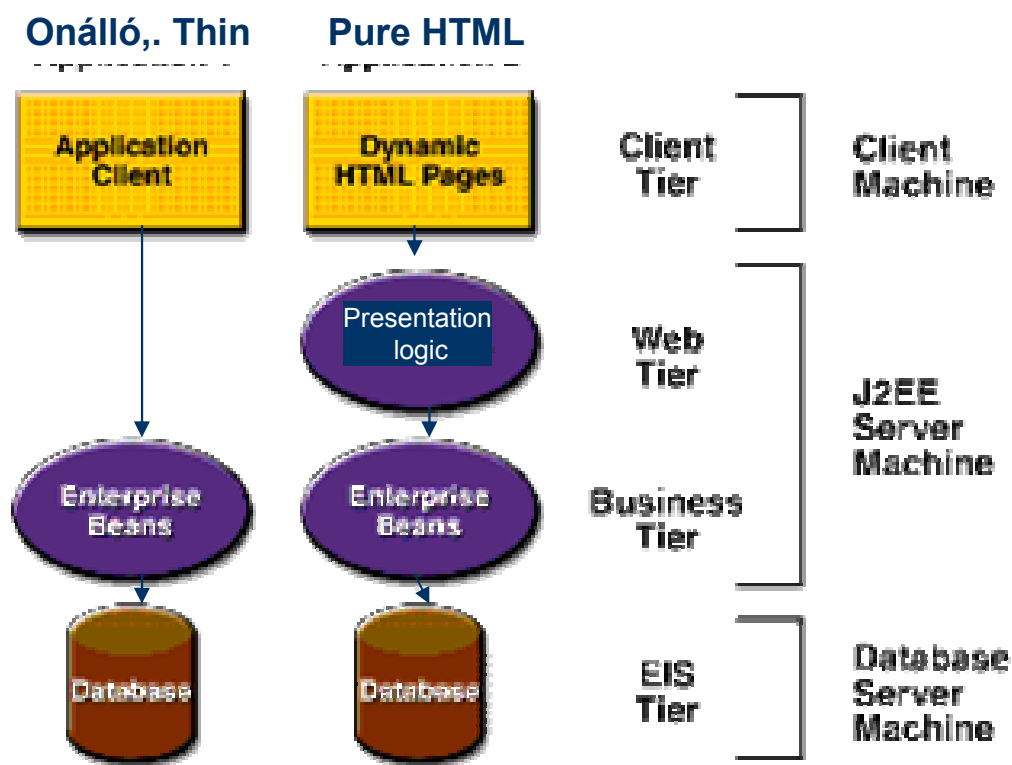
- 1960-1990: monolitikus alkalmazás
 - Több felhasználó multiuser OS segítségével
 - Egyetlen nagy gép, buta terminálokkal
- 1990-2000 : „kliens-szerver” architektúra
 - A szerver rendszerint egy adatbázisszerver (pl. Oracle)
 - A kliensek rendszerint PC-k
 - Nagy és bonyolult kliens-program (
 - „fat client”: az üzleti logika is itt van
 - Installálási, verziófrissítési problémák)
 - Teljesítmény-problémák
 - a sokszoros párhuzamos hozzáférés miatt
- 2000- A „többrétegű” architektúra (általában 3 réteget jelent):
 - Adatbázis „tier” (*ejtsd: tír*)
 - csak általános adatkezelés, lehetőleg semmi speciális logika
 - „Middle” tier
 - Az „üzleti logika” helye
 - Egy vagy több ún. alkalmazáserveren fut
 - Client tier
 - Lehetőleg minél egyszerűbb, csak GUI kezelés

*Minket főleg
ez érdekel!!*

Kliens oldali alternatívák

- „fat client”
 - Üzleti logikát tartalmaz, nem illik a 3-rétegű szemléletbe
- Önállóan futtatható, de „thin” kliens alkalmazás
 - A telepítési problémák továbbra is fennállnak
 - J2EE szerver esetén leginkább ez is Java (a RMI miatt)
- Java applet „thin” kliens
 - Automatikusan letöltődik a browserbe
 - Viszont: funkcionális és teljesítmény-problémák
- Pure .html kliens
 - A GUI prezentációs logika is a szerver oldalra kerül
 - Egy új réteg: „web tier”
 - Összességében ezzel van a legkevesebb probléma!!!!
 - Legújabb: AJAX: JavaScript extenzív használata, ennek köszönhetően reszponzív, jól kezelhető UI.

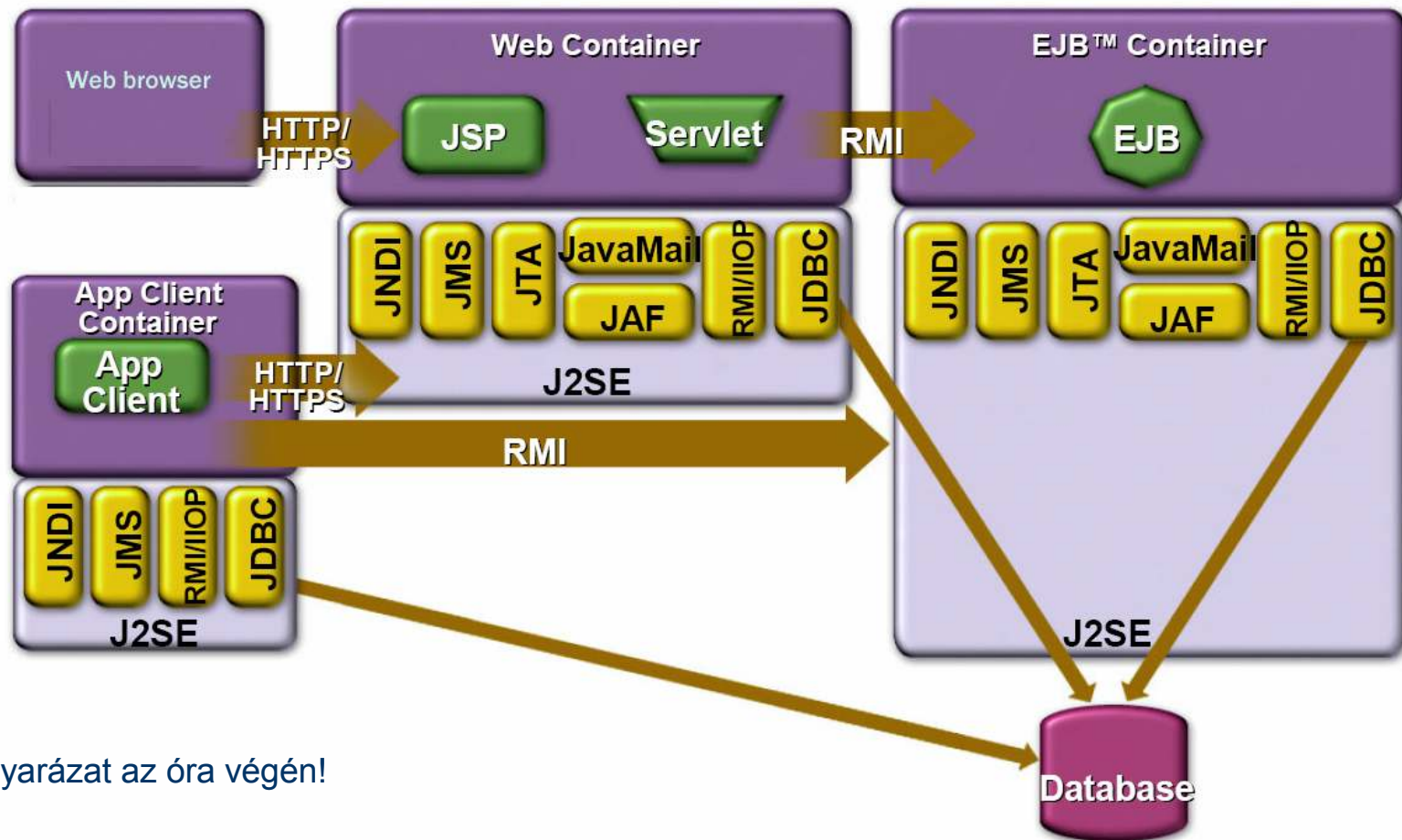
Többrétegű alkalmazások egyszerű nézete



- Két alternatíva
 - Önálló GUI
 - v. java Applet
 - Web GUI (browser)
- Számunkra a lényeg a middle tier!
- De: az igazi kincs a data rétegben van!

..de a valóság ennél komplexebb!

Példa: egy J2EE alkalmazásszerver vázlatja



Magyarázat az óra végén!

2. J2SE Hasznos szolgáltatások

- JAR-ok és class loaderek
- Reflection, JavaBeans
- Networking, RMI
- JDBC, JNDI
- Security:
 - Controlled execution,
 - Security Services (encryption, signatures, hashing)
 - PKI key management
- JMX
- Internationalization

JAR-ok és classloaderek

- JAR: Java binárisok (és más fileok) ZIP formátumban összegyűjtve és tömörítve
- MANIFEST.MF file a JAR-ban, pl:
 - *Main-Class*: a JAR default main osztálya
 - *Class-Path*: hivatkozás más JAR-okra
 - Verzió-azonosító adatok
- Classloader
 - Hierarchikus mechanizmus osztályok megkeresésére
 - Először a generikusabb helyekről tölt
 - 1. JAVA JRE, 2. Deployed extensions, 3 User-set classpath

Ld: még <http://java.sun.com/docs/books/tutorial/ext/index.html>
<http://java.sun.com/docs/books/tutorial/deployment/jar/index.html>

Reflection és JavaBeans

- Runtime típusvizsgálat
 - Lekérdezhetők a metódusok, és adatok, típusokkal és paraméterekkel
 - Hierarchia követése
 - Új: Annotations
- JavaBeans: speciális konvencióknak megfelelő, runtime vizsgálható (introspection) osztályok
 - Property-k, event mechanizmus
 - Perzisztencia
 - Scripting és build environment támogatás

<http://java.sun.com/docs/books/tutorial/reflect/index.html>

<http://java.sun.com/docs/books/tutorial/javabeans/index.html>

Networking és RMI

- Socketek (UDP és TCP servers/client), datagrammok.
- RMI: távoli eljárás hívás
 1. Szerver készít egy „remoteolható” objektumot
 2. Beregisztrálja a registrybe (bind)
 3. Kliens lookup-olja
 4. Kliens hívja a remote metódusokat
 - Remote és serializable objektumokkal

<http://java.sun.com/docs/books/tutorial/networking/index.html>

<http://java.sun.com/docs/books/tutorial/rmi/index.html>

JDBC, JNDI

- JDBC: adatbázis elérés
 - Connection with authentication
 - SQL utasítások végrehajtása
 - Adatok iterálása ResultSet-en
- JNDI
 - Címtárak (pl. DNS és LDAP elérése)
 - Az API elve a JDBC-hez hasonló

<http://java.sun.com/docs/books/tutorial/jdbc/index.html>

<http://java.sun.com/docs/books/tutorial/jndi/index.html>

Security

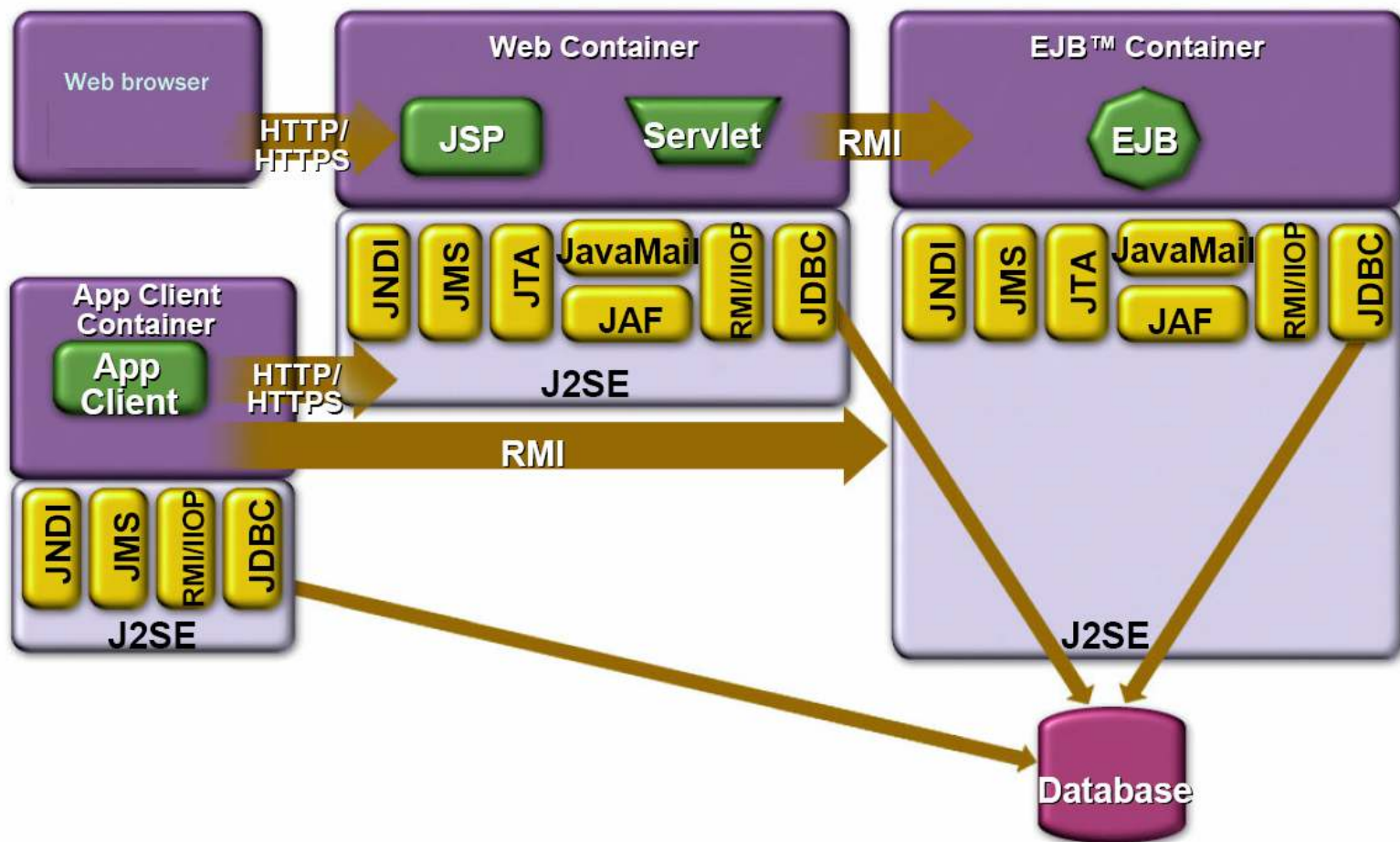
- Permission-controlled execution of sensitive ops
 - Permissions – mit tehet program
 - Definiálható a class helye vagy JAR signature alapján
- Security services
 - Encryption
 - Signatures creation and checking
 - Hashes, message digests
- PKI Key management
 - Kulcsok generálása és tárolása a VM által

<http://java.sun.com/docs/books/tutorial/security/index.html>

JMX, Internationalization

- JMX: kívülről, futás közben menedzselhető interfészek a VM-hez és a programokhoz
- Internationalization: több nyelvű user interfészre felkészített programok.
- <http://java.sun.com/docs/books/tutorial/jmx/index.html>
- <http://java.sun.com/docs/books/tutorial/i18n/index.html>

..de az élet nem ilyen egyszerű



Fontos API-k (és „TLA”-k)

- RMI – remote method invocation **Java SE**
 - Más java VM-ekben futó objektumok hívása
- JDBC – Java database connectivity **Java SE**
 - Database írása, olvasása
- JSP – Java Server Pages
- JTA (JTS) – Java transaction API és - Services
 - Nagy
- JMS – Java messaging services
- JNDI – Java naming and directory services **Java SE**
 - Objektumok keresése név stb.. szerint
- JAAS – Java authentication / authorization framew.
 - User jogosultság kezelésJavaMail
- JavaMail
- JAF -- Java activation framework????

Egyéb rövidítések

- EJB – Enterprise Beans
 - Egyfajta komponens, (nem JavaBeans!!!)
- JSP – Java Server Pages
 - Egyféle Web presentation technológia
 - Idén ezt tanuljuk

Ezekkel
részletesen
foglalkozunk
majd

Köszönöm a figyelmet!

