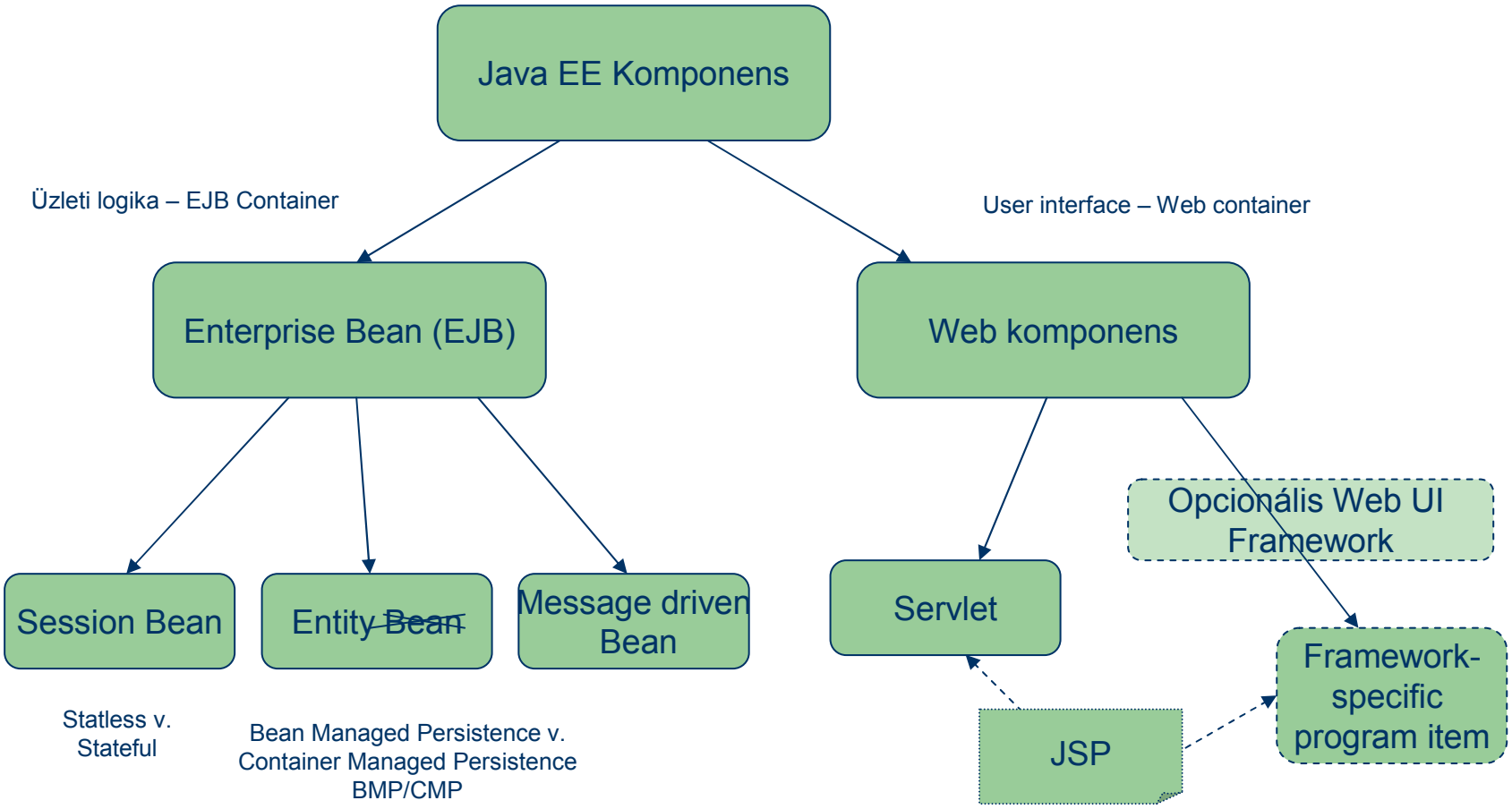
5. rész: A Java EE és az Enterprise Bean réteg

Bakay Árpád dr.
NETvisor kft
(30) 385 1711
arpad.bakay@netvisor.hu

Java EE Komponensek „családfája”



Java EE komponensek csomagolása és telepítése

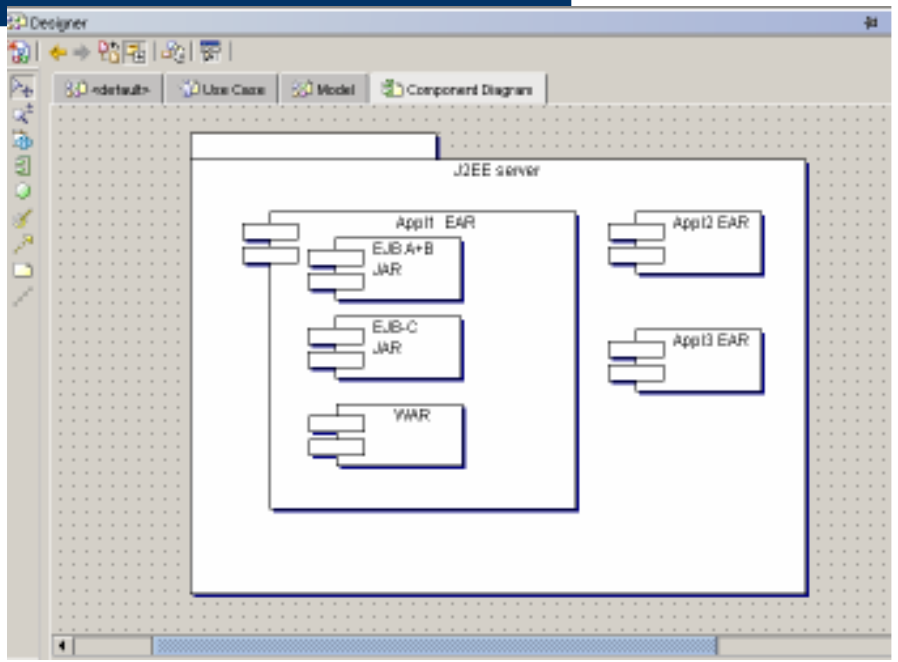
- **Alkalmazás telepítése egyetlen file-ban!!**
  - egymásba skatulyázott jar (zip) fileok
  - EJB JAR: EJB komponens
    - Interfész , implementáció, segéd osztályok
  - WAR (web archive): Web komponensek
    - Servletek, JSP-k, segéd osztályok
  - EAR (enterprise archive): EJB JAR-ok és WAR-ok
    - tehát egy komplett alkalmazás
    - saját class loader a hierarchiában
- xml formátumú komponens descriptorok
  - A komponensek fontos tulajdonságai tartlamazza
    - Exportált és importált bean-ek, tranzakciós, security paraméterek, etc.
  - + szerver-specifikus kiegészítő descriptorok
  - **az EJB 3.0-ban ez már csak opcionális!**

Példa a telepített csomagokra

JBOSS App szerver példány

- **App1.ear**
  - AandB.jar
    - ABean.class, A.class
    - BBean.class, B.class
    - **ejb-jar.xml + jboss.xml**
  - C.jar
    - CBean.class, C.class
    - **ejb-jar.xml + jboss.xml**
  - Gui.war
    - index.jsp
    - main.jsp
    - **web.xml + jboss-web.xml**
  - **application.xml + jboss-application.xml**
- **App2.ear**

  …
- **SimpleBean.jar**
- **SimpleWebApp.war**



J2EE alkalmazásfejlesztés - munkafolyamat, szerepek

- Komponens készítők
  - EJB készítő
    - EJB source (.java) + deployment descriptor (.xml)
    - EJB .class-ok + dd .xml -> EJB komponens (.jar)
  - Web komponens készítő
    - Servlet-ek (.java) + JSP-k (.jsp) + fix content (.html+képek) + deployment descriptor (.xml)
    - servlet .class-ok + .jsp + .html + képek +dd .xml -> web appl. arhívum (.war)
- Alkalmazás integrátor
    - EJB .jar-ok, és web .war-ok integrálása
    - .jar + .war + appl depl descriptor -> enterprise archívum .ear
- Alkalmazás telepítő és üzemeltető
    - Telepíti és felügyeli az alkalmazást egy J2EE futtatókörnyezetbe

„Üzleti tudás”

„GUI tudás”

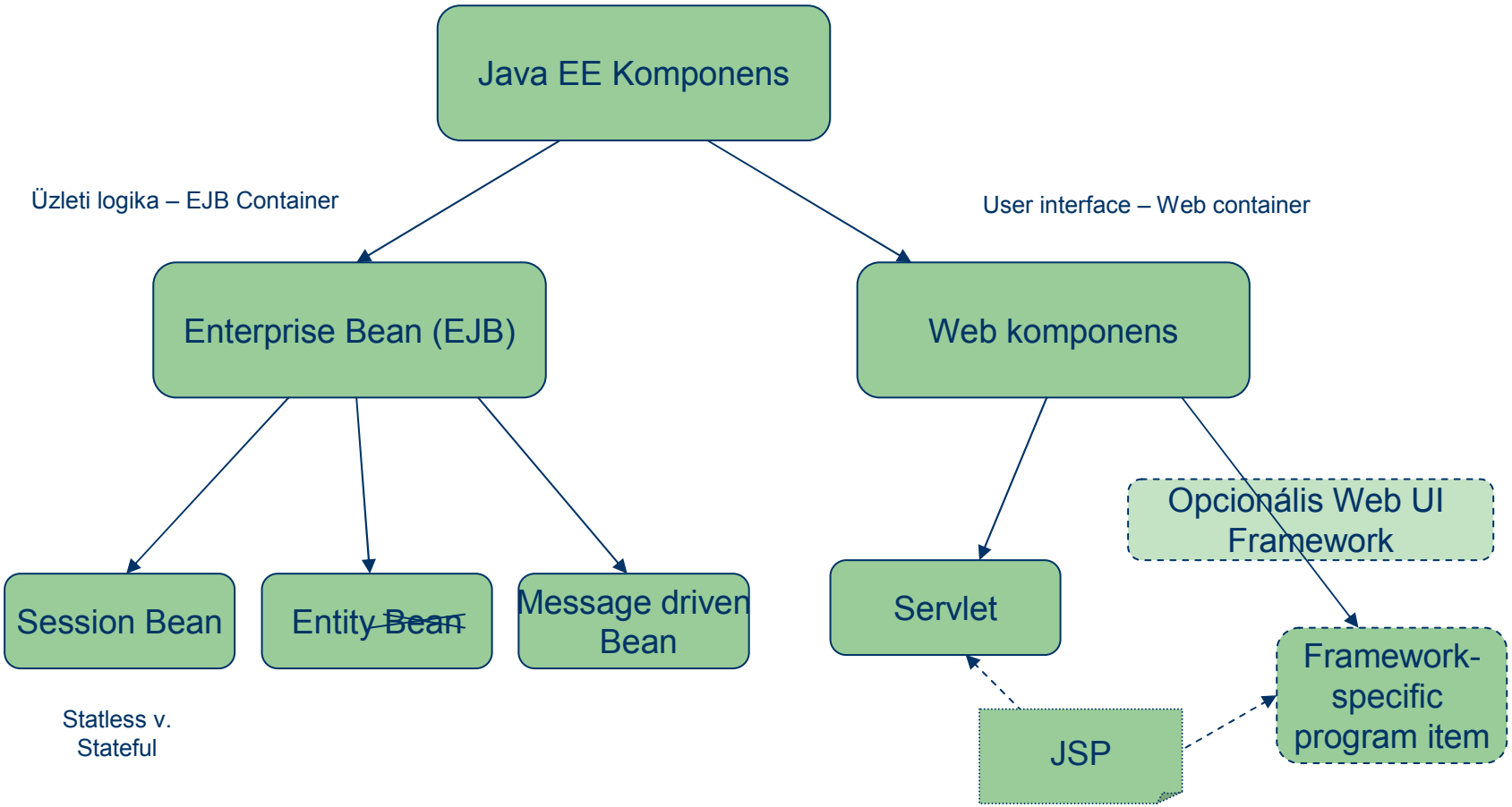
Rendszerintegrátor

Operátor, adminisztrátor

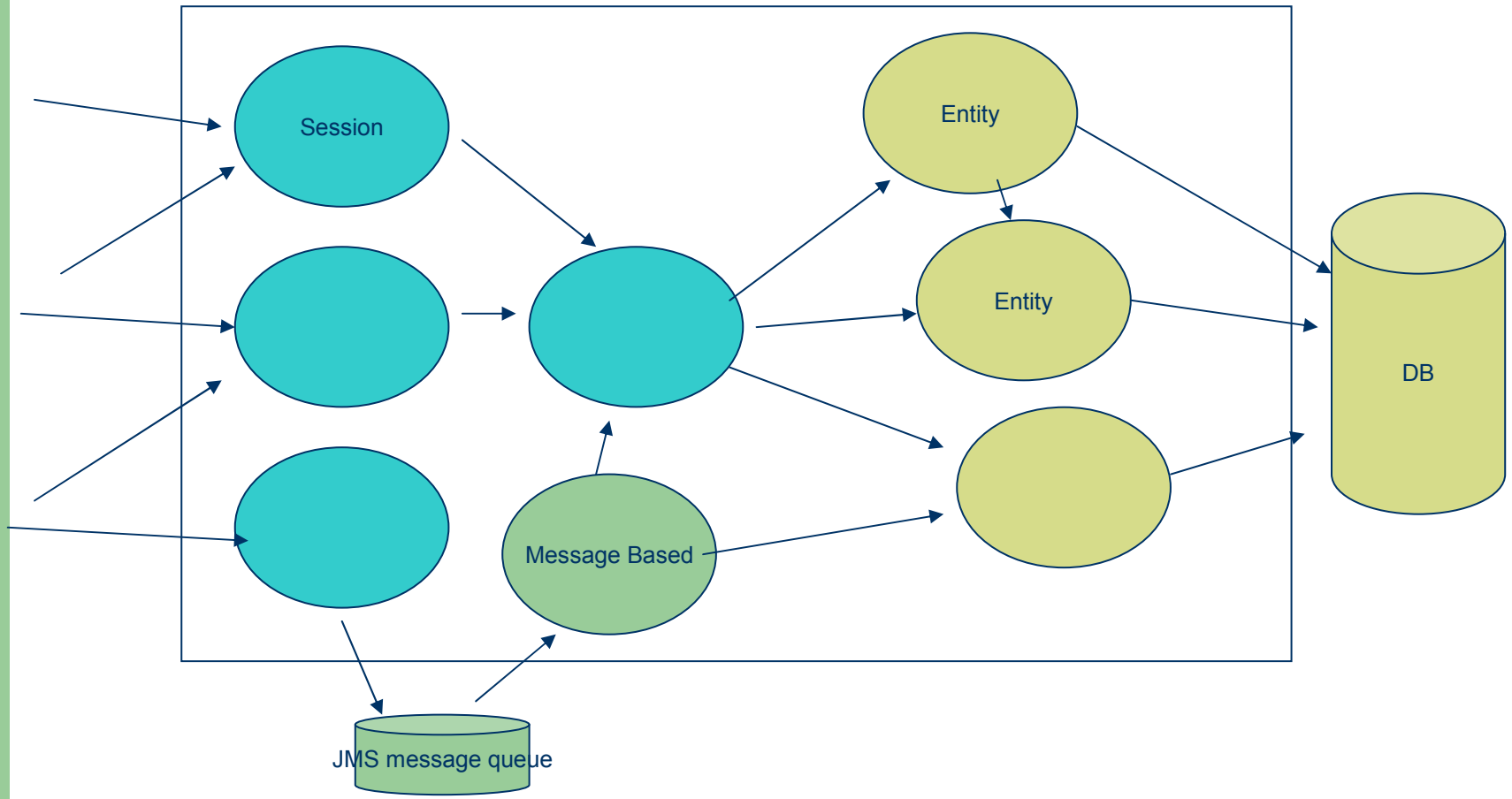
Az EJB réteg

- EJB típusok
- Az RMI-től az EJB-ig
- EJB technológia értékelése

Ismét:
Java EE Komponensek „családfája”



Jellemző EJB architektúra



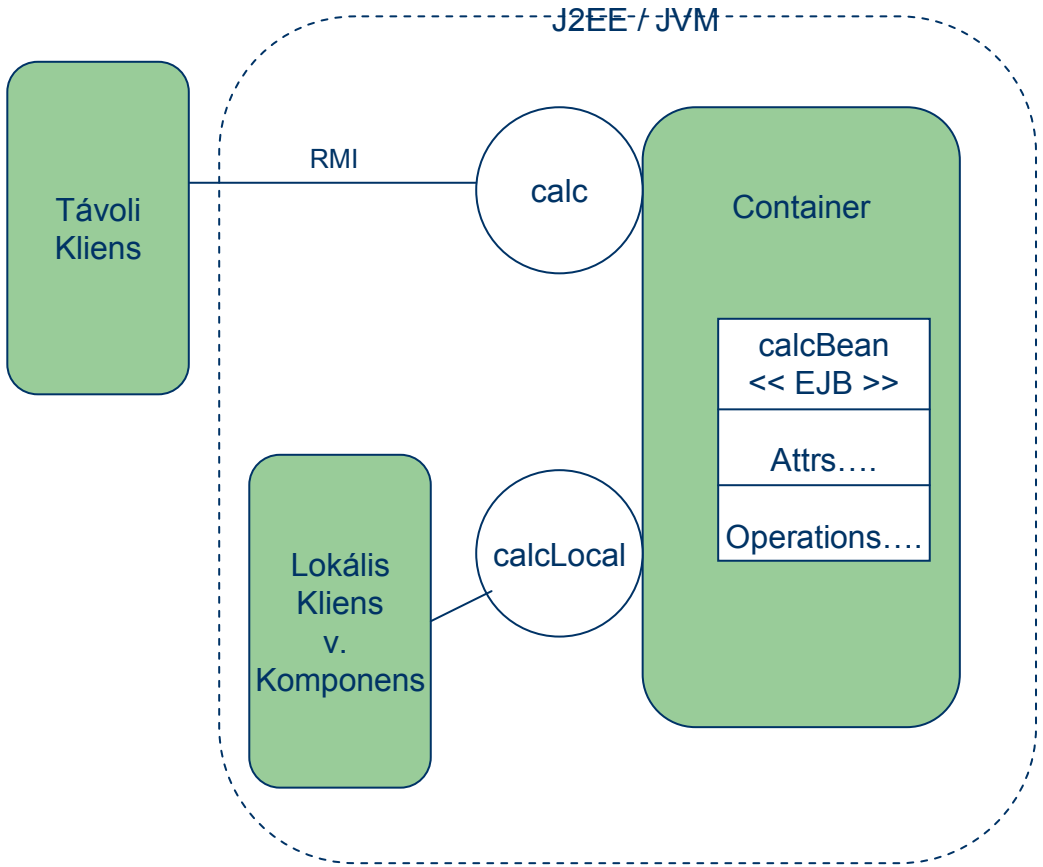
A legegyszerűbb bean fajta: Session Bean

- Az RMI koncepció továbbfejlesztése
  - Alapötlet: N objektummal M ügyfelet kiszolgálni, N << M
- Egy bean instance – jellemzően egy kliens
  - Egyszerre mindig csak egy -> szinkronizálás
  - Ha a kliens elengedi – vége van
- Általános üzleti funkciókra,
  - Pl. shoppingCart, orderManager
  - vagy nem OO jellegű API: pl. arithmetics

Stateful és Stateless Session bean-ek

- Stateless -- „közös használatú”
  - Állapot megőrzése *csak a hívás idejéig*
  - Felcserélhető instance-ok
    - a J2EE szerver egy object pool-ból gazdálkodik
  - A gyakorlatban lényegesen gyorsabb
- Stateful – „állapotos”
  - Lehet hívások között megőrzött állapota
    - *de csak amíg valakik fogják a referenciát!!!*
    - Többfázisú műveleteket támogat (conversation/protocol)
  - A Bean instance-ek nem felcserélhetők
  - Aktív és „ki-swappelt” mód
    - ejbActivate(), ejbPassivate() -> swappelés elő-és utómunkálatok
  - Create-nél akár paraméterezhető

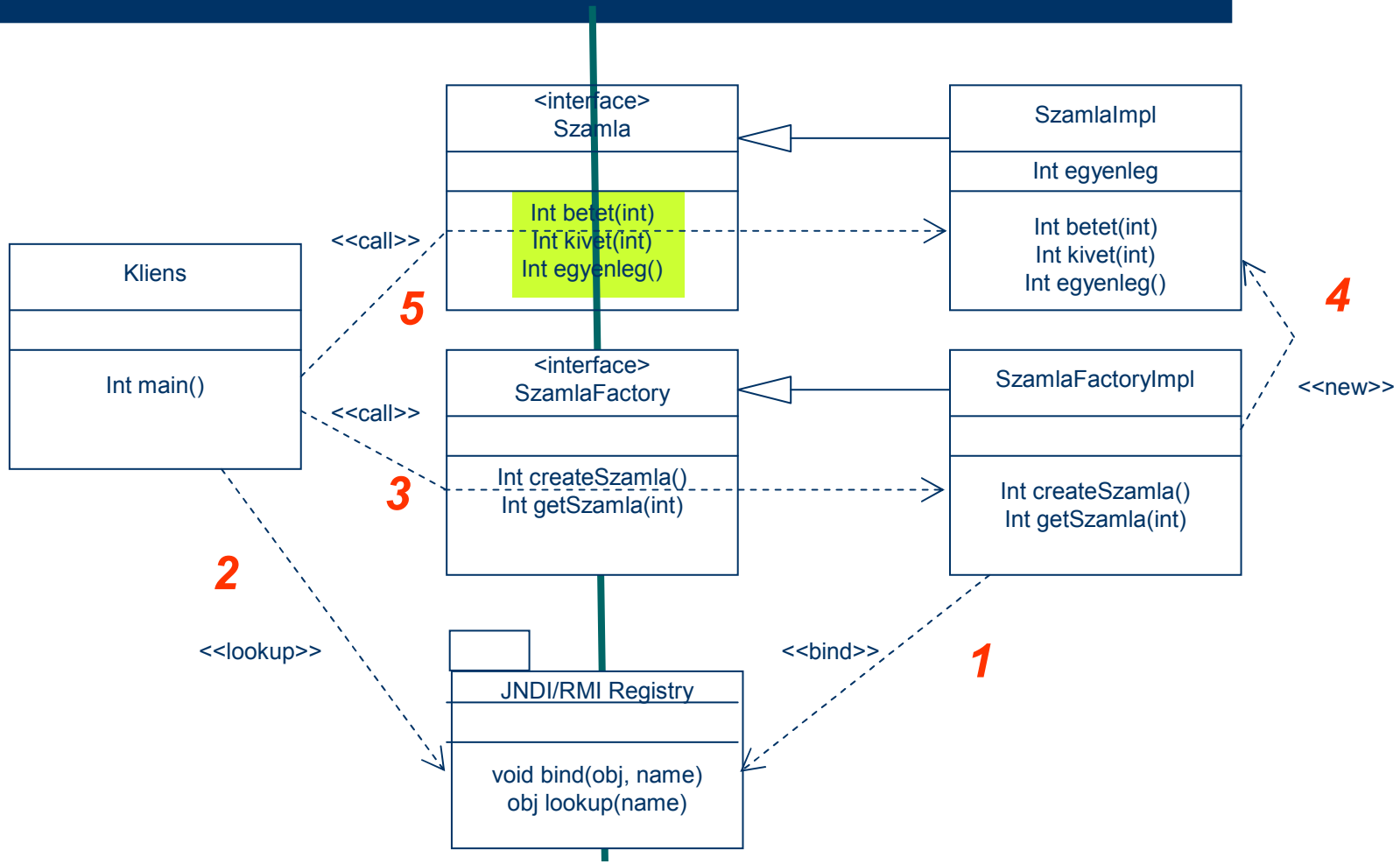
Normál (remote) és local Bean interfészek és használatuk



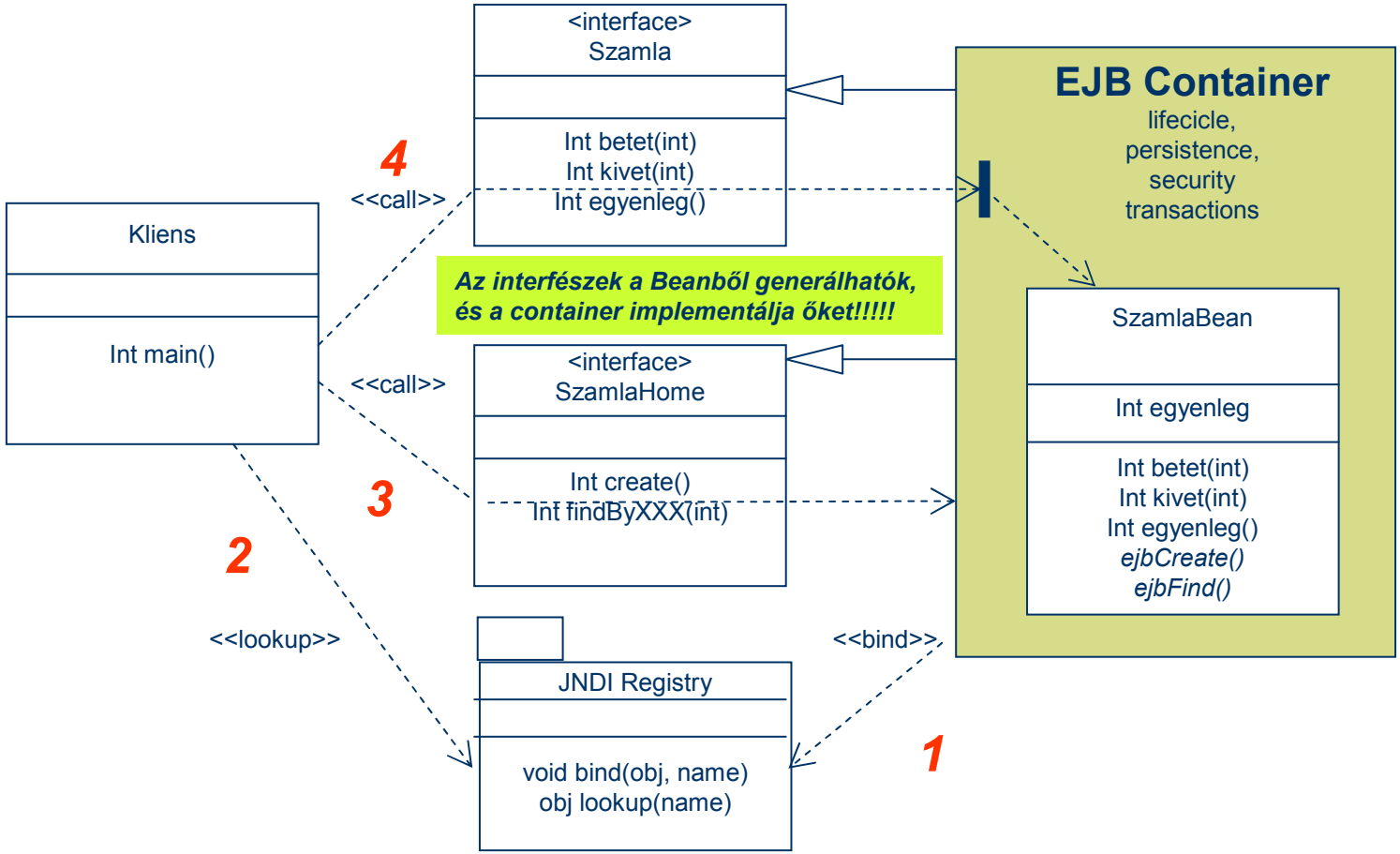
Remote interface
Előnyök:
•Mobilitás,
•Izoláció
•Failover,
•Load-Balancing

Local interface
Előnyök:
Sebesség
Nincs RemoteException

Hagyományos távoli komponens factory patternnel (pl: RMI)



EJB architektúra

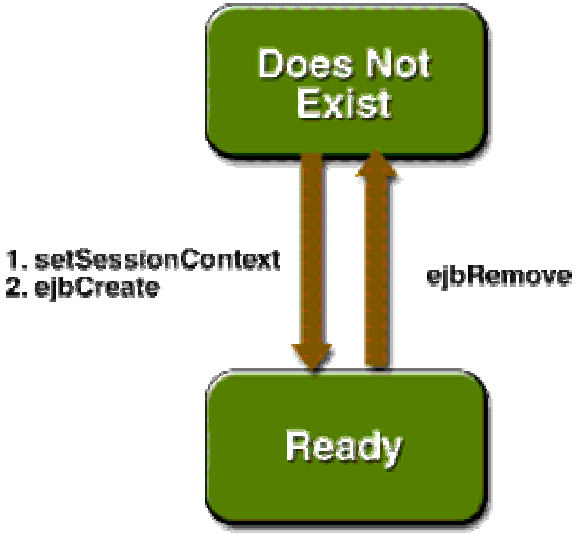


Milyen szolgáltatást kap a Bean a containertől?

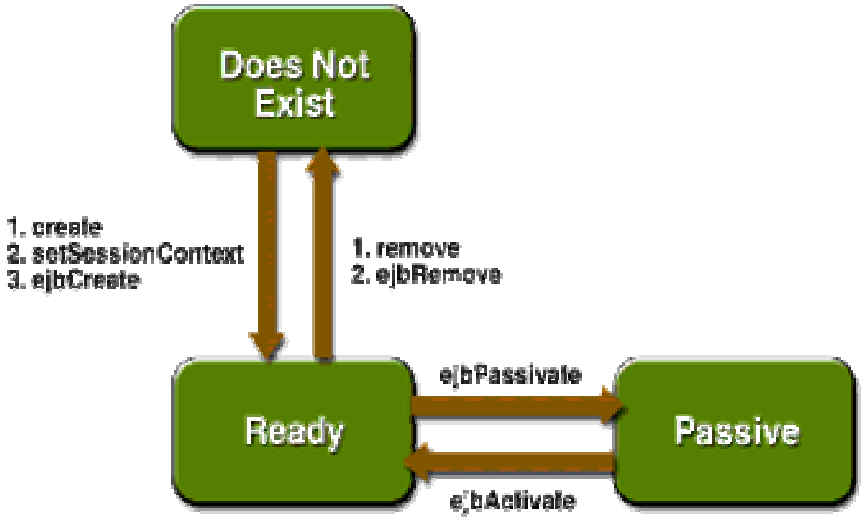
- JNDI regisztráció és létesítés
- Thread reentrancy szigetelés
- Remote invocation
- Tranzakciós szigetelés – ha kér
- Hibernálás és felélesztés (stateful bean)
- Database perzisztencia (entity bean)
- Authentikáció és authorizáció – ha kér
- Pooling, load balancing (stateles bean)
- Multi-homing (biztonság!)

Session bean életciklusok

Stateless
PI: CurrencyConverter



Stateful
PI: shoppingCart



Entity

- Az objektumok és a relációs rekordok közötti áthidalás elsődleges eszköze.
  - Entity általába valami passzív fogalmat ír le: számla, ügyfél, áru, vásárlás, stb.
- Enterprise Bean, „persistence” szolgáltatással
  - az állapotát adatbázisban tárolja
  - hosszú életű, „multi-homed” objektum
- Ki biztosítja az adatbázis-műveleteket?
  - A Bean írója: Bean-managed persistence BMP
  - A J2EE container: Container Managed P. CMP
- Oriási változás a 2.0 és 3.1 Entityk között
  - Hibernate rendszer került alkalmazásra

Message-Driven Bean

- Aszinkron hívásokhoz
  - Pl. SMS-ek feldolgozása
- Egyetlen metódusa az onMessage(msg)

- Nem fogjuk használni

Az EJB 2.1 technológia értékelése

- Előnyök:
  - Tiszta, izolált funkciók
    - EJB design patterneket kell alkalmazni
  - Maximálisan védett komponensek
    - Konkurrencia, tranzakciók, biztonság, high-availability, felügyelet
- Hátrányok:
  - Néhol túl szigorú architekturális kötöttségek
  - Komoly teljesítmény-problémák
  - Egy beanhez 3-6 file-t kellett megírni.

EJB 3.0

- Kevesebb filet kell írni (1 bean- 1 file)
- Korszerűsített perzisztencia réteg

- Kb. egy éve megjelentek az EJB3-kompatibilis EJB containerek.

Köszönöm a figyelmet!