

REST, RMI, WS és CORBA

Készítette: Lukács Ferenc (feco911@gmail.com) 2013.06

Forrás: 2013-as Előadás diák.

Felelősséget nem vállalok, nem lehet hivatkozási alap.

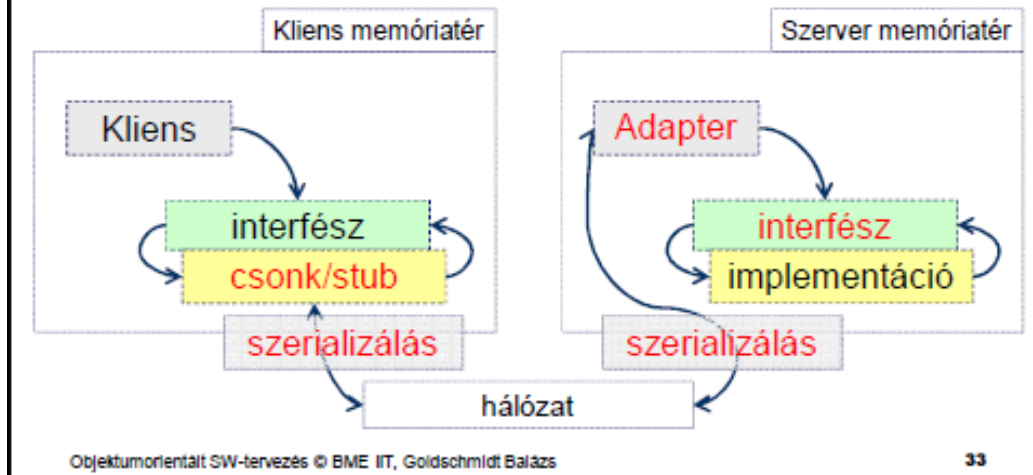
REST – Representational State Transfer

- http kiegészítése – GET POST PUT DELETE
- bemenő paraméterek
 - URL része, QUERY STRING, POST paraméter, BODY
- kimenő paraméter – BODY
- minden erőforráshoz URL-t rendel
 - egyértelmű helyinformációval
 - erőforrás pl: dokumentumok, adatok, szolgáltatások stb.
- működés erőforrásokon
 - CRUD – Create Read Update Delete
 - safe, idempotent
- állapotmentes
 - DE az alkalmazásnak lehet állapota, és ezt mindig visszaküldi
- kritikák
 - CRUD on kívül másra nem alkalmas
 - nincs interfészleíró
 - tervezési guidelineok hiánya
 - middleware funkciók hiánya
 - megbízható kommunikáció
 - titkosítás, digit. aláírás
 - tranzakciókezelés

RMI – Remote Method Invocation

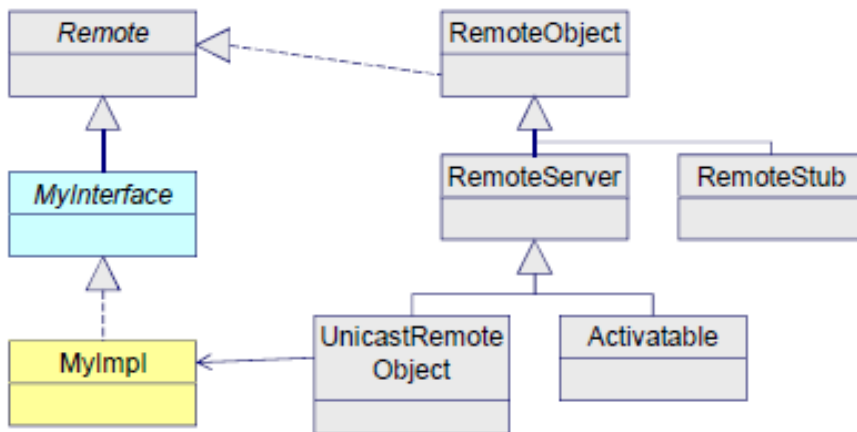
- Security Manager
 - felüldefiniálásával lehetséges szabályok létrehozása
 - checkXXX (read, write, access, connect) stb lekérdezés
 - **Elutasítás esetén Security Exceptiont dobnak**
 - engedélyek policy fájlban vannak

RMI architektúra



- Távoli interfész
 - Remote leszármazottja
 - minden metódusának RemoteException dobását kell jeleznie
 - visszatérési érték
 - primitív típus – érték szerint
 - Serializable megvalósító objektum deep copyval

Szerver oldal



- RemoteObject – távoli objektumok és csomók őssztálya
- RemoteSzerver – lehetővé teszi távoli objektumok létrehozását és exportálását
- UnicastRemoteObject – Távoli objektum exportálása, és a hozzá tartozó stub elérése
- Activatable – Perzisztens elérésű objektumokhoz

- RMI Registry
 - távoli objektumok névhez kötése
 - név – stub párokat tárol
 - LocateRegistry – megtalálja adott hoszton, porton, protokollal
 - Registry interfész – biztosítja a szolgáltatást

WS – SOAP – Simple Object Access Protocol

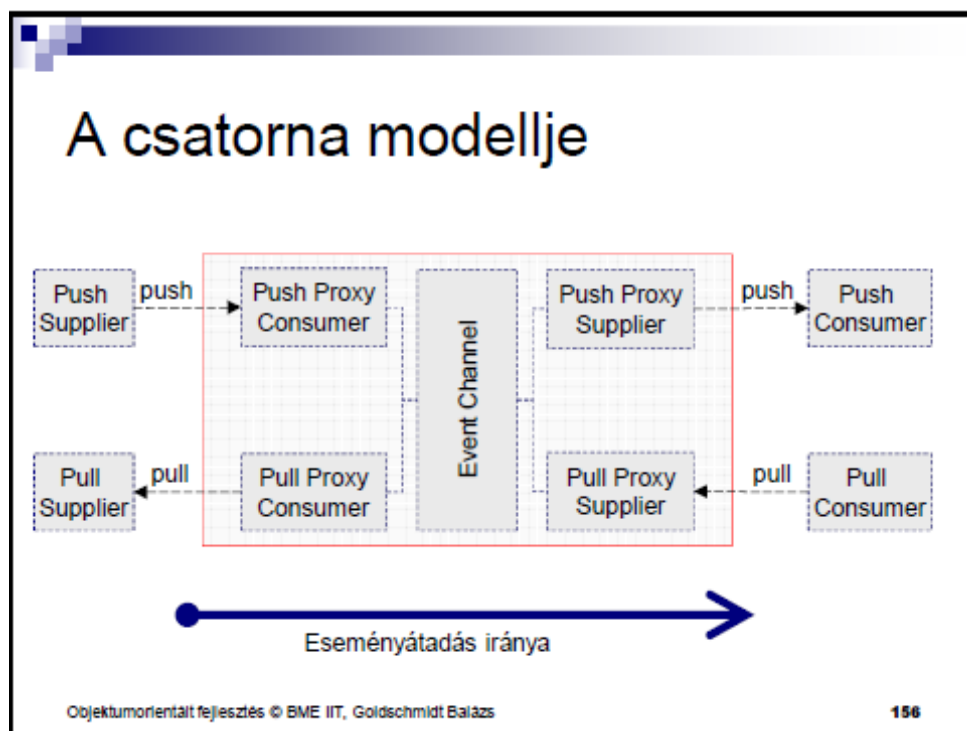
...soa.pdf

CORBA – Common Object Request Broker Architecture

- Nyelv független, oprendszer független
- elosztott objektum orientált architektúra
- cél: platform – és nyelvfüggetlen kommunikáció
 - szabványos adatátvitel, végpont elérés, implementációs csatlakozás
- hálózati protokoll: IIOP
- interfész leírás: IDL

- interfészek, metódusok, adatstruktúrák leírása
- **csak leírás, implementáció saját nyelven**
 - **hívás is natív nyelven**
 - IDL fordítása JAVA-ra
 - SHelper – segédfüggvények CORBA marshallinghoz
 - SHolder – segédosztály az out, és inout paraméterekhez
- névterek
- van primitív(double,long,boolean) és összetett típus(enum, string) is
- 3 féle paraméteradás (in,out,inout)
- Szerver oldal
 - POA – Portable Object Adapterből kell örököltetni a szervantot
 - célja, hogy a szervant újraírása nélkül lehessen ORB implementációt váltani
 - feladata a szervantok aktiválása, és értesítése
 - szabványos IDL specifikált metódusok
 - nyelvi sajátosságok nyelvi mappingban
- Kliens oldal
 - Az objektumot a STUB-on keresztül érjük el natív típusokkal, hívásokkal, kivételekkel
- CORBA szolgáltatások
 - Gyakran előforduló problémák
 - üzenetküldés, objektum keresés
 - Naming Service
 - Trading Service
 - Szolgáltatásokra tudunk keresni
 - ServiceType – szolgáltatás típusa => táblára képződik le
 - Offer – konkrét szolgáltatás => sorokra képződik le
 - Keresés SQL Query-vel
 - Ajánlatok kezelése
 - _ publikálás: „insert”
 - _ módosítás: „update”
 - _ törlés: „delete”
 - _ keresés: „select”
 - Különböző traderek összekapcsolása
 - az egyikben elindított keresés a többibe is kiterjed
 - A kapcsolat beállítása lehet
 - local_only: csak az aktuális traderben keresünk
 - if_no_local: ha az aktuálisban nincs, mehet tovább
 - always: minden elérhető traderben keresünk
 - Event és Notification Service
 - Szerepek
 - termelő (supplier) termeli az eseményeket
 - fogyasztó (consumer) dolgozza fel
 - Kommunikációs modell

- push modell
- pull modell
- Modellek implementálása
- Push modell
 - termelő, ha van új esemény, meghívja a fogyasztó push metódusát
 - a fogyasztó a push metódust implementálja és vár
- Pull modell
 - a fogyasztó, ha eseményt szeretne, meghívja a termelő pull metódusát
 - termelő implementálja a pull metódust és vár
- Azért, hogy a csatornához egyszerre többen is csatlakozhassanak proxy kell
 - plusz szolgáltatások
 - nincs közvetlen kapcsolat a termelő és fogyasztó között
 - proxy
 - interfészt nyújt (termelőként és fogyasztóként is viselkedhet)
 - tárolja a felgyülemlett objektumokat



A proxy-k

- A supplier proxy-k saját FIFO pufferrel rendelkeznek
 - ha tele van a puffer, az új esemény bekerül, a legrégebbit eldobja
- A **push consumer proxy** a *push* hatására a csatornához továbbítja az eseményt
- A **push supplier proxy** a következő eseményt push hívással adja át a fogyasztónak
- A **pull supplier proxy** a pull hatására vagy ad egy új eseményt a pufferből, vagy blokkol
- A **pull consumer proxy** pull hívással kér új eseményt a termelőtől

Elosztott technológiák

| | RMI | CORBA | WS | REST |
|------------------------|-----|-------|----|------|
| Elosztott | X | X | X | X |
| Szabványos | | X | X | X |
| Programnyelv-független | | X | X | X |
| Széles körű támogatás | | X | X | X |
| Egyszerű | X | | | X |
| Egyszerű API | X | | X | X |
| Gyors | X | X | | X |
| Biztonság, tranzakciók | | | X | |

Elosztott technológiák

| | RMI | CORBA | WS | REST |
|----------------|----------------|----------------|------|----------------|
| Protokoll | RMI | IIOP | SOAP | HTTP |
| Interfészleíró | Java interface | IDL | WSDL | nincs/ WADL |
| Katalógus | JNDI | Naming Service | UDDI | - |