

# **CORBA bevezetés**

Paller Gábor

2004.10.08

# CORBA

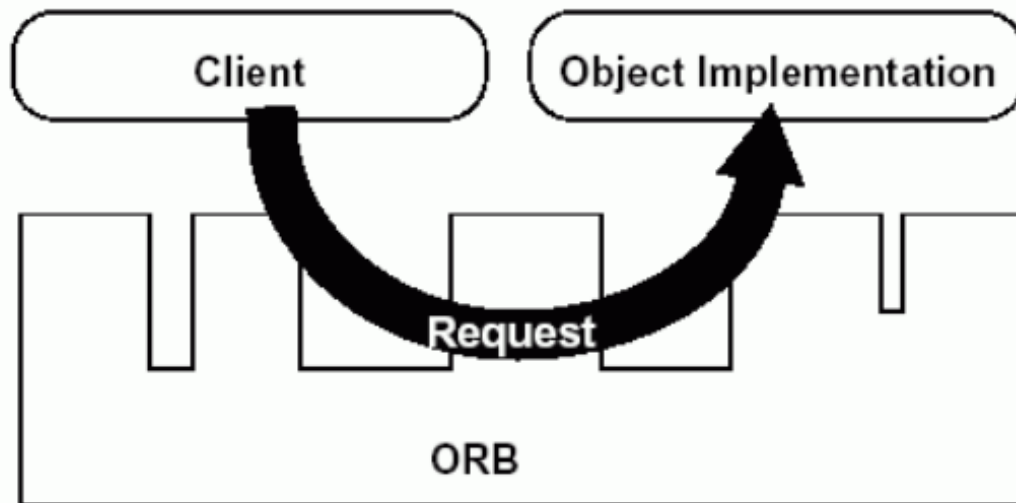
- Common Object Request Broker Architecture
- Az Object Management Group (OMG) felügyeli (ugyanaz, mint az UML-t)
- A specifikáció célja alkalmazások együttműködésének biztosítása. Ezeket az alkalmazásokat tetszőleges nyelven, operációs rendszerre lehet fejleszteni és közöttük levő hálózati kapcsolat sem számít.
- A specifikáció tartalma:
  - Objektumorientált, nyelvfüggetlen elosztott alkalmazások készítését lehetővé tevő keretrendszer
  - Szabványos CORBA szolgáltatások
- Jelentősége a menedzsmentben: menedzsment interfészeket gyakran implementálnak CORBA modellben.

# CORBA alapelvek

- A CORBA lehetővé teszi, hogy az alkalmazások objektumokat tehessenek más alkalmazás által elérhetővé
- A CORBA objektumok nem kell megfeleljenek semmilyen objektumnak az objektum funkcionalitását megvalósító programban
- Ezek az objektumok az Object Request Broker (ORB) ellenőrzése alatt vannak, az ORB hozza létre, adminisztrálja és pusztítja el őket
- Az alkalmazás csak a minimálisan szükséges számú objektumot osztja az ORB-n keresztül, az ORB-n keresztül történő hívás elég drága.

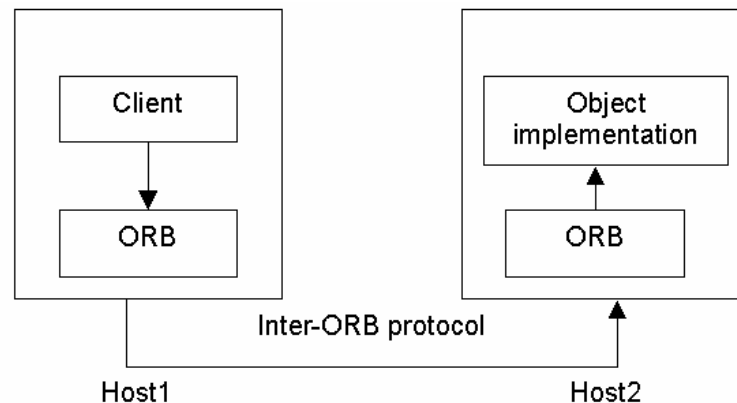
# CORBA alapelvek (2)

- Az elosztott objektumokat a saját ORB-n keresztül érjük el, így az ORB-k együttműködése elfedi a megcélzott objektum implementálási nyelvét, az eléréshez használt protokolt, stb.



# ORB absztrakció

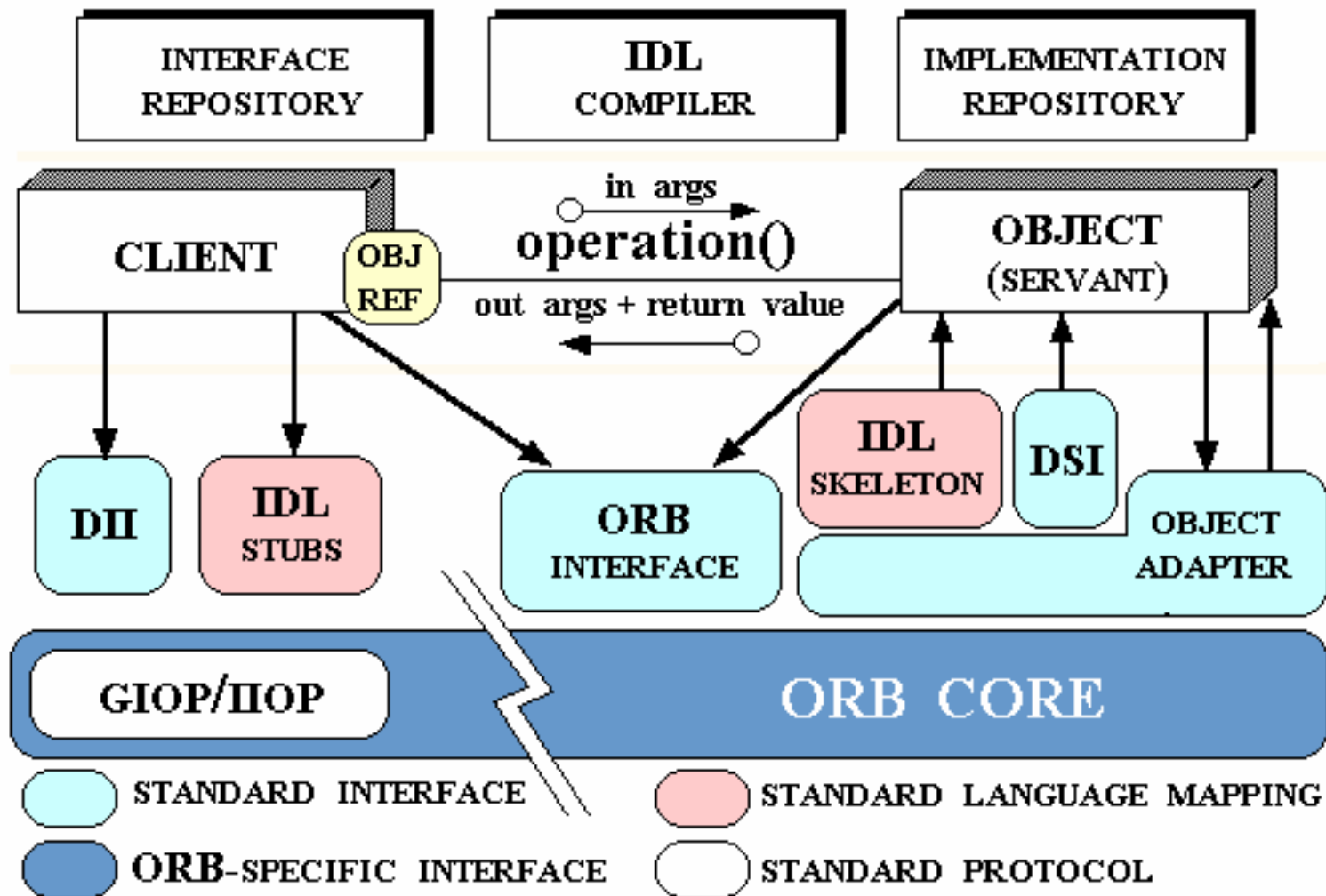
- Az alkalmazások csak saját ORB-jüket látják, ez a CORBA következő fontos tulajdonságait teszi lehetővé:
  - A kliens és a szerver tetszőleges nyelven lehetnek írva mindaddig, amíg az alkalmazás és az ORB együttműködése megfelel a CORBA által megkövetelt szemantikának. Az implementációs nyelv nem is kell objektum-orientált legyen.
  - A kliens és a szerver lehetnek ugyanazon a gépen vagy lehetnek más gépeken is, melyek között hálózati kapcsolat van. Ha az ORB-k megértik egymást, a két ORB tetszőleges protokollal használhat egymás között
  - Terhelésmegosztási vagy hibatűrési megfontolásokból a szervert át is lehet helyezni egy másik gépre és a kliens alkalmazás ezt nem veszi észre, mert az ORB-n keresztül ugyanazt a szolgáltatást látja.



# CORBA architektúra részletesen (2)

- Object – CORBA programozási egység, ami identitásból, interfészből és implementációból (servant) áll
- Servant – célnyelven írt program, ami a távoli interfészt implementálja
- Client – a távoli interfészt hívó program
- ORB – Object Request Broker
- ORB interface – a szabványos CORBA ORB interfész implementációspecifikus megvalósítása
- CORBA IDL stub és skeleton – a nyelvfüggetlen interfészből fordított csatoló kód az ORB és az alkalmazás között
- Dynamic Invocation Interface (DII) – a stubok megkerülésével közvetben lehet elérni az ORB szolgáltatásait. Pl. nem csak RPC hívás lehetséges, pl. egyirányú vagy szétválasztott kérdés-válasz
- Dynamic Skeleton Interface (DSI) – a DII szerveroldali párja. Lehetővé teszi, hogy az ORB IDL skeleton nélkül is továbbíthasson kéréseket az alkalmazásnak.
- Object adapter – az ORB az objektum élelciklusát vezérli ezen az interfészen keresztül

# CORBA architektúra részletesen



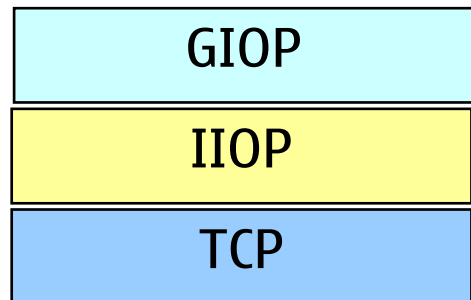
# CORBA specifikáció elemei

- ORB specifikáció, hogyan működik együtt az ORB és az alkalmazások
- Inter-ORB protokoll, ami biztosítja az ORB-k együttműködését, még akkor is, ha az ORB-k különböző gyártóktól származnak
- Interface Definition Language, ami lehetővé teszi az elosztott objektumok interfészeinek nyelvfüggetlen leírását
- Számos szabványos CORBA szolgáltatás



# GIOP és IIOP

- Az ORB-k közötti protokoll szabványosítva van, hogy különböző gyártóktól származó ORB-k is együttműködhessenek
- Szabvány protokollok
  - General Inter-ORB protocol (GIOP): transzport-protokolltól független protokoll ORB-k között
  - Internet Inter-ORB protocol (IIOP): GIOP TCP felett
- IIOP stack:



# GIOP protokollüzenetek

- Request: CORBA metódushívás
- Reply: CORBA metódushívás válasz
- CancelRequest: CORBA metódushívás megszakítása
- LocateRequest: megvizsgálja, hogy a cél ORB képes-e kezelni egy CORBA metódushívást. Erre a Request is használható lenne, de a LocateRequest nem viszi az (esetlegesen hosszú) hívási paramétereket
- LocateReply: Válasz a LocateRequest-re
- CloseConnection: Bármelyik fél küldheti, ha le akarja zárni a kapcsolatot. A lebontó fél összes kint levő kérése törlődik
- MessageError: Az üzenetet fogadó hibásnak találta a GIOP üzenetet

# Példa: Request GIOP üzenetstruktúra

- Request ID (32 bites üzenetazonosító)
- Response flags (mikor kapja vissza a kliens a vezérlést (amikor az ORB elfogadta a kérését, amikor sikerült elküldeni, amikor a cél ORB visszajelzi, hogy a kérést sikeresen megkapta ...))
- Target – a célobjektum azonosítója a cél ORB-on belül. Kezdetben ez az érték teljesen ORB specifikus volt (objektumazonosító, objektumcím, stb.). A későbbi CORBA szabványokban bevezették az Interoperable Object Reference (IOR) fogalmát. Az IOR struktúra megmondja az objektum típusát, hogy null-e, milyen transzportokon elérhető, stb.
- A meghívandó metódus nevét
- Hívási paraméterek binárisan kódolva, egymás után pakolva

# Interface Definition Language

- CORBA programokat sokféle nyelven lehet implementálni
- Az interfészek leírására kell egy közös nyelv, hogy pl. egy C++ kliens megértse, milyen interfészt nyújt egy Jáva objektum
- Az IDL nyelv egy ilyen közös interfészleíró nyelv
- Feladata: interfészek, közöttük levő öröklődési relációk, az interfészeken keresztül elérhető metódusok definiálása
- IDL leírásokból általában az ORB IDL fordítója IDL stubokat és skeletonokat fordít, közvetlenül az IDL interfészleírót nem használják

# IDL példa

- `typedef sequence<long> LongVect;`

```
interface AdderInterface {  
    long add( in long n1, in long n2 );  
    long addArray( in LongVect nums );  
    string reverse( in string str );  
};
```

- A típusok (long, string) IDL típusok. Minden célnyelvnek definiálnia kell a leképezést a saját típusaira.
- Pl: CORBA long: 32 bites előjeles egész szám. Ennek Jávában az `int` Jáva típus felel meg.
- `sequence<long>`: CORBA long-ok változó hosszúságú tömbje. Ennek Jáva megfelelője: `int`-ek tömbje, `int[]`

# IDL metódusok paraméterei

- in – a metódushívás bemeneti paramétere. Más nyelveken ez megfelel az érték szerinti hívásnak
- inout – a metódushívásnak ez egy bemeneti paramétere, de a metódus megváltoztathatja a bemeneti értéket és a metódus futásának a végén a hívó visszakapja a módosított értéket. Más nyelveken ez megfelel a referencia szerinti hívásnak
- out – a metódusnak ez egy kimeneti paramétere. A hívás során nem adódik át bemeneti érték, a metódushívás végén a paraméter értéke visszaadódik a hívónak. Más nyelveken ezt gyakran referencia szerinti hívással implementálják

# Öröklődés

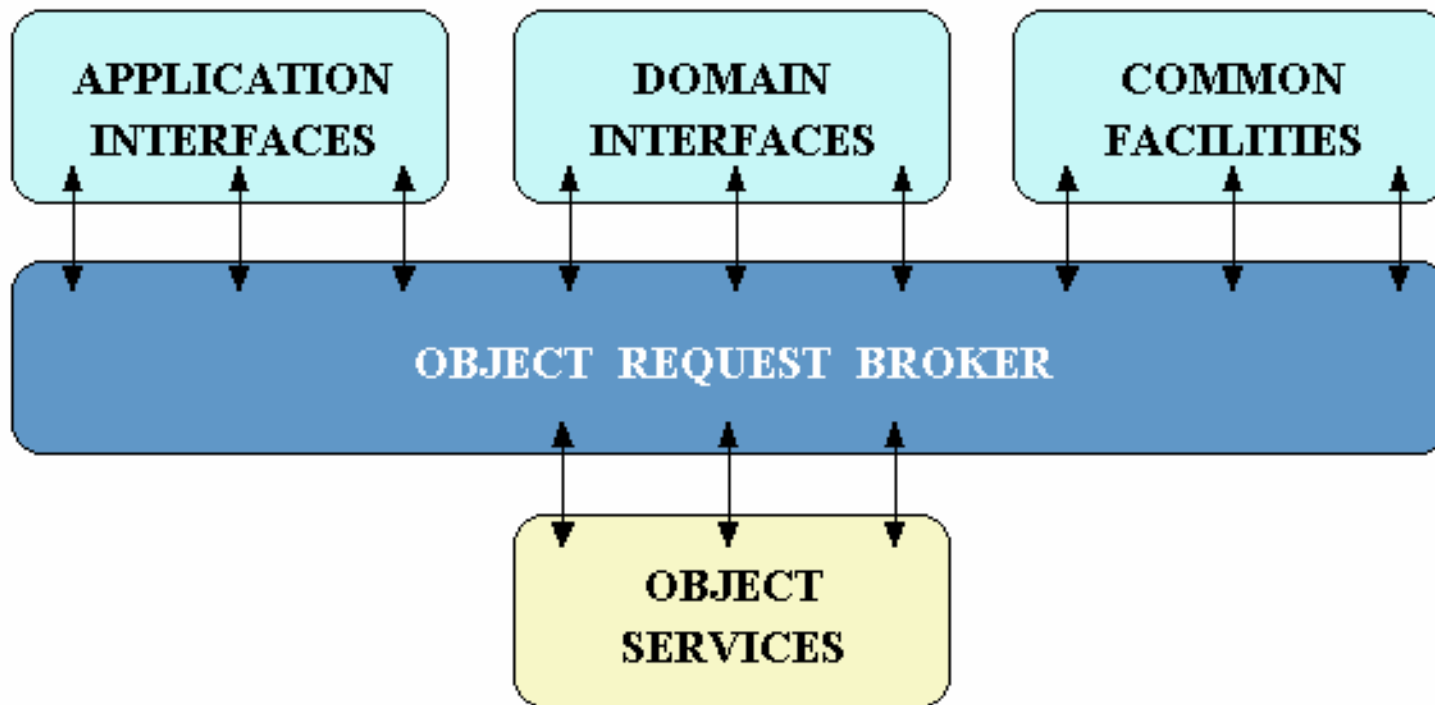
- Interfészek öröközhetnek egy másik interfészből
- Többszörös öröklődés is lehetséges
- ```
interface A { ... }  
interface B: A { ... }  
interface C: A { ... }  
interface D: B, C { ... }
```

# Interface repository

- Egy ORB által szolgáltatott interfészek az interface repository-ban vannak, innen keresi ki az ORB, amikor hívást kap
- Általában nem az IDL fájlokat tárolják, hanem azokból fordított modulokat



# CORBA interfészek

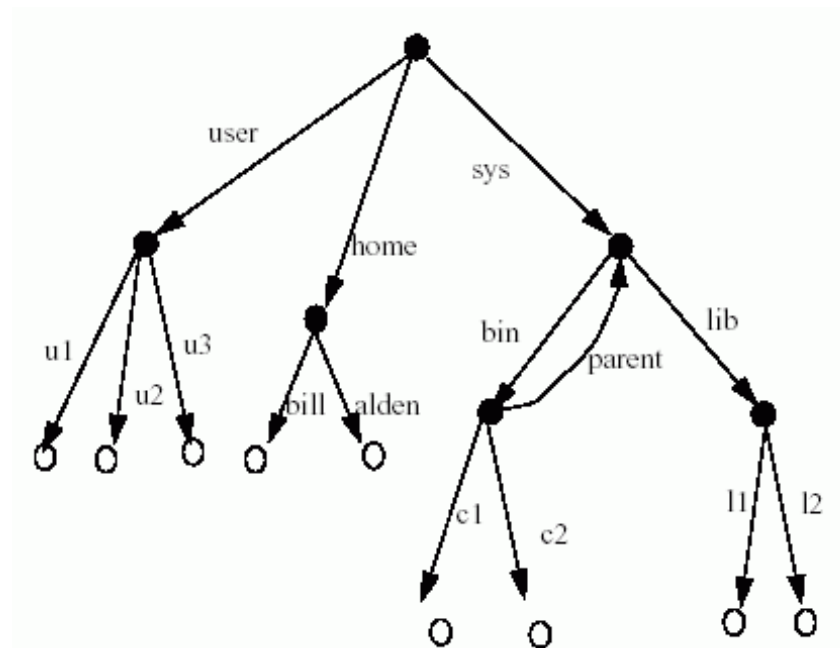


# CORBA interfészek (2)

- Object services – alkalmazástípustól független interfészek, amelyek általánosan szükségesek elosztott programokhoz
  - Naming service – objektumokat név szerint lehet megtalálni vele
  - Trading service – objektumokat tulajdonságaik alapján lehet megtalálni vele
- Common facilities – alkalmazástípustól független interfészek, amelyek főleg végfelhasználói alkalmazásoknak hasznosak. Pl. Distributed Document Component Facility (DDCF).
- Domain interfaces - alkalmazástípusok részére kifejlesztett interfészek, pl. telekommunikáció, orvosi, pénzügyi, stb.
- Application interfaces – kifejezetten az alkalmazás részére kifejlesztett interfészek

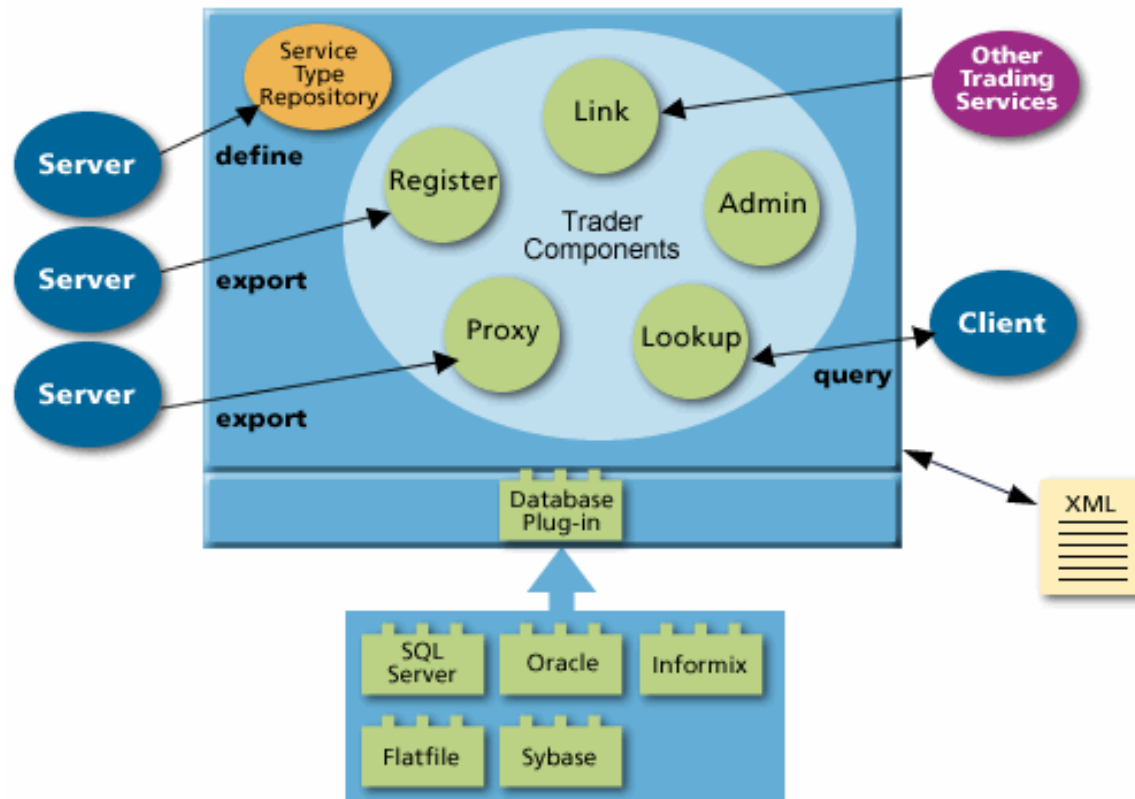
# CosNaming service

- Ez a CORBA alapértelmezett elnevezési szolgáltatása
- Egy CORBA objektumot az IOR azonosít, ez azonban változhat, pl. amikor az ORB-t újraindítják
- A CosNaming egy CORBA szolgáltatás (szabályos IDL által leírt interfész), ami IOR-ok névhez rendelését ill. névhez tartozó IOR-ok lekérdezését teszi lehetővé
- A neveket egy hierarchikus struktúrában tárolja



# CosTrading service

- "Telefonkönyv" szolgáltatás elosztott objektumok részére
- Az objektumot különböző tulajdonságaival együtt regisztrálni lehet a trader adatbázisban. A tulajdonságok alapján aztán bonyolult kereséseket lehet végezni



# Szabványos CORBA szolgáltatások

- Tucatjával vannak a Common Facilities és Domain Interfaces kategóriában
- Common Facilities példák:
  - Collection Service: Lehetővé teszi a leggyakoribb collection típusok (vektorok, hashtáblák, halmazok, sorok) egységesített manipulációját
  - Event Service: objektumok közötti események küldését és fogadását teszi lehetővé
  - Externalizing Service: lehetővé teszi egy objektum állapotának bájt sorozattá alakítását, majd ebből az objektum helyreállítását
  - Time Service: Megszerezhető az aktuális idő, az időmérés hibájának becslésére is alkalmas
  - ...
- Domain Interface példák:
  - Air Traffic
  - Audio/Visual streams
  - Genetikai adatok cseréje
  - Tervezési és gyártórendszerek adatainak kezelése (Platform Data Management)
  - Munkafolyamat (workflow)
  - ...