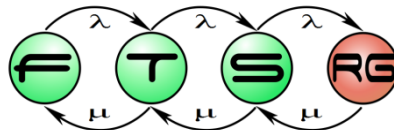


Kommunikációs middleware megoldások

Szolgáltatás integráció előadás

Huszerl Gábor (BME MIT)



Kérdések

- Hogyan kommunikálhat két alkalmazás?
 - Szinkron és aszinkron megoldások
 - Ilyet én is tudok írni. Sőt, jobbat is!
- Melyik megoldás hogyan működik?
 - Melyik mire jó?
 - Mi a különbség?
 - Hol találni ilyet?
- Aszinkron megoldások olcsón?

Middleware

- Hol van az a középén?
 - Operációs rendszer felett, alkalmazás alatt
 - Alkalmazásokból „alászálló” funkciók
 - néha tovább az OS-be
- Mit csinálnak?
 - Kommunikáció, HA, UI, skálázás, grafika, játék, ...
 - A továbbiakban itt mindig kommunikációs MW
- Minek?
 - Alkalmazás integráció, szolgáltatás integráció
 - Komponensközi kommunikáció

Preklasszikus middleware fajták

- Adatcsere az alkalmazások között, ahogy lehet
 - Fájl átvitel
 - Adatbázisok
 - Elektronikus levelezés
 - Weboldalak
 - Sockets, Pipes
 - Házilagos megoldások
 - alkalmazásokból kiemelve

Korszerű middleware technikák

- Házilagos megoldások
 - Házon belüli fejlesztés házon belüli igényekhez
- Távoli eljáráshívás (RPC, ORB) 1988
 - Szinkron elosztott alkalmazásokhoz
- Üzenetsorok (MQ) 1994
 - Üzenet alapú, nagy megbízhatóságú komm.-hoz
- Publish-Subscribe (P/S) 1998
 - Üzenet alapú, valós idejű kommunikációhoz
- Egyéb aszinkron technikák
 - Üzenet alapú, olcsóbb
 - Rövid érvényességű információhoz

Middleware-ek feladatai

- Kliens-szerver kapcsolat (tág értelemben)
 - Komponensközi kapcsolat, hívások/üzenetek
- Platformfüggetlenség (HW-től és OS-től)
- Hálózatfüggetlenség (hálózati protokolltól is)
- Publikus API (Mennyire publikus?)
- Nyelvfüggetlenség (programozási nyelvtől)
- Adattárolás függetlenség
 - Adatbázis rendszerektől
 - Üzenettárolásnál, jogosultságoknál, címeknél is
- Egységes alkalmazásfejlesztői platform

Extra MW feladatok 1.

- Közös felhasználó azonosítás
 - Egységes felhasználói jogosultságok (különböznek!)
 - Egyszeri belépés (single sign on)
- Tranzakció azonosítás (összetartozó üzenetek)
- Biztonság
 - Titkosított adat- és vezérlési forgalom
 - Komm. titkosítása + üzenetek titkosítása
- Elhelyezkedés-függetlenség
 - Hol fut a másik?
 - Ha „mozog” a másik

Extra MW feladatok 2.

- Adatbázis-orientált szolgáltatások
 - Elosztott lekérdezések/beillesztések/törlések
 - RDBMS szolgáltatások
- Alkalmazás-orientált szolgáltatások
 - Bármilyen, ami ott éppen kell
 - Pl. atomi tranzakciók, óra szinkron, ...
- Menedzsment szolgáltatások
 - MW felügyelete (SNMP, Unicenter, Tivoli, ... ágens)
 - Konfigurációs eszközök
- ... és még sokan mások

Korszerű middleware technikák

- Házilagos megoldások
- Távoli eljáráshívás (RPC, ORB)
- Üzenetsorok (MQ)
- Publish-Subscribe (P/S)
- Egyéb aszinkron technikák

Házilagos MW megoldások

- Zöld mezős, házon belüli megoldások
- Házon belüli célokra (fejlesztendő alkalmazás(család)hoz)
- Fejleszteni kell hozzá
 - Mindenféle fejlesztő és tesztelő eszköz
 - Saját hálózati protokoll
 - Saját API, belső működés, technológiák
- Szakember igény (kereskedelmi termék adoptálásához kevesebb/gyengébb is elég)
 - Rendszermérnök, programozó, tesztelő
 - Hálózati mérnök, hálózati adminisztrátor
 - Projekt menedzser

Házilagos MW megoldások

- A várható igények felmérése (rendszerfejlesztők)
- Meghozandó fejlesztői döntések
 - Kommunikáció (szinkron, aszinkron)
 - Hálózat (TCP, UDP, IPX, ...)
 - Információáramlás (üzenetek/hívások, egyirányú/kétirányú, 1-to-1/1-to-n/n-to-n)
 - Technológiák (C, C++, Java, C#, XML, címzés, ...)
 - Teljesítmény (sávszélesség, késleltetés, kliensek száma, üzenetek száma, ...)
 - Megbízhatóság, biztonság, ...

Házilagos MW megoldások

- Előnyök
 - Jobb testre szabhatóság, kritikus paramétereik jobbak lehetnek
 - Extrém körülmények között megoldást nyújt
- Hátrányok
 - Fejlesztés költsége, ideje, szakember igénye
 - Folyamatos karbantartás és fejlesztés
 - Támogatás később is csak házon belülről
 - cég erős függése alkalmazottaitól

Házilagos MW megoldások

- Tipikus alkalmazási területek
 - Valós idejű kómm. speciális igényekkel
 - pl. TTTech
 - Speciális hálózati protokollok felett
 - kevésbé támogatott technológiákhoz
 - Örökölt, kritikus, nehezen integrálható alkalmazásokhoz
 - ha már a kereskedelmi MW házilagos adaptere sem megoldás
- „Csináld magad!”

Korszerű middleware technikák

- Házilagos megoldások
- Távoli eljáráshívás (RPC, ORB)
- Üzenetsorok (MQ)
- Publish-Subscribe (P/S)
- Egyéb aszinkron technikák

Két rokon middleware technika

- RPC (Remote Procedure Call)
 - Tradicionális progr. techn. mintájára
 - Szabványos eljárásívás szerint
- ORB (Object Request Broker)
 - OO technológia mintájára
 - Metódushívás „kiterjesztése”
- Mindkettő
 - Request/reply szinkron kommunikáció
 - Hely és platform transzparens
 - interoperabilitás (gép, nyelv, OS között)
 - heterogén elosztott rendszerek
 - skálázhatóság (egy → több gép, kis → nagy gép)

RPC/ORB

- Feladatai
 - Hívás elkapása, hívott fél megkeresése
 - Paraméterek átvitele
 - Szerver eljárásának/metódusának meghívása
 - Eredmény visszajuttatása (vezérlés visszaadása)
- Technológiák
 - RPC: régi, tisztán nem fordul elő már
 - ORB: különböző technológiák
 - CORBA (OMG szabvány)
 - DCOM (eredetileg Windows alá)
 - RMI (csak Java alá)

RPC/ORB

- Előnyök
 - Szabványos technológiák elosztott rendszerekhez
 - Alkalmazások központosítható menedzselése
 - Tipikusan objektum-orientált megközelítés
 - Hagyományos alkalmazások „webesíthetőek”
- Hátrányok
 - Igazán nagyra rosszul skálázódik
 - Gyakran szakértőt igénylő architektúrák
 - Inkompatibilis ORB implementációk
 - Nehéz hibakeresés és adminisztráció

RPC/ORB

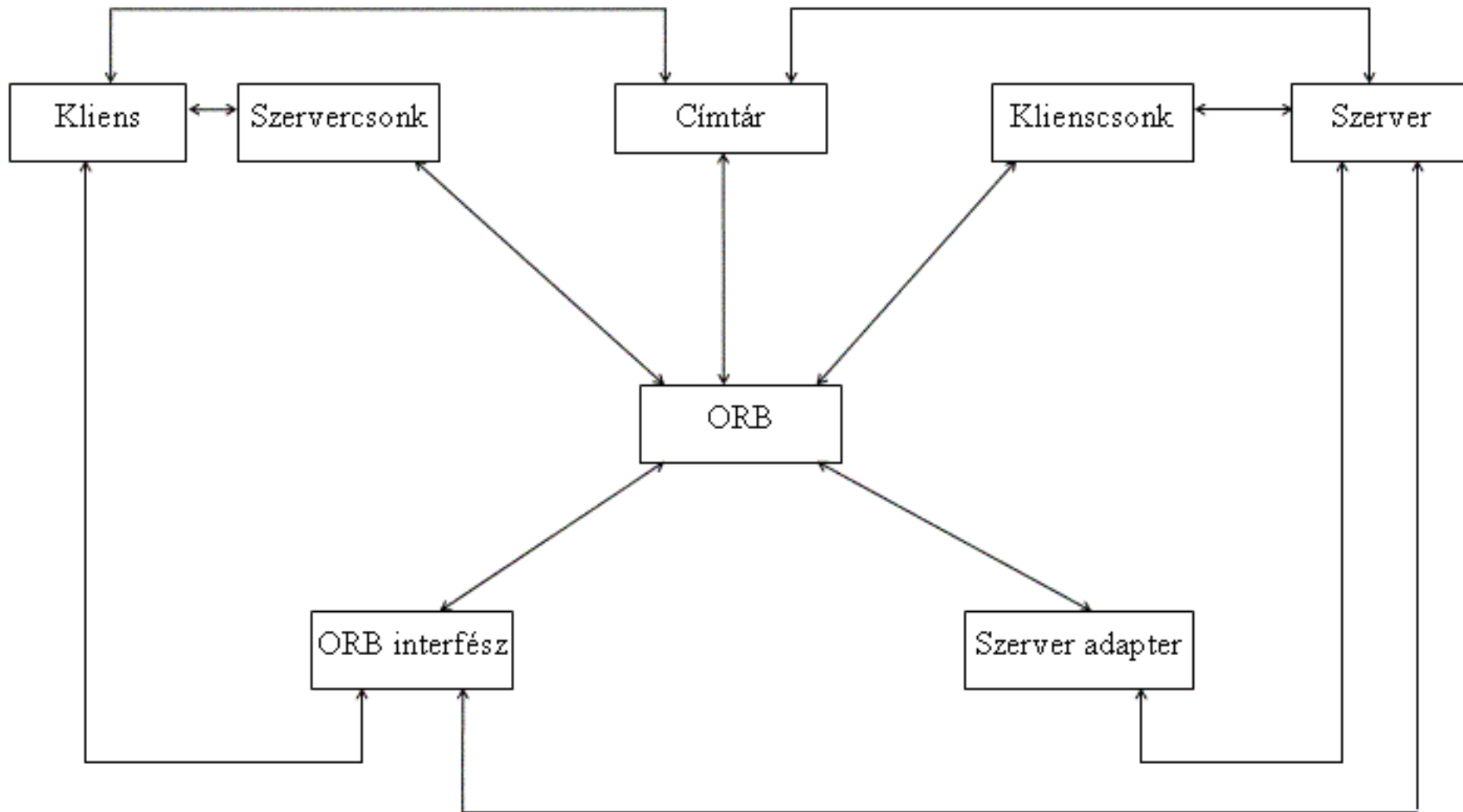
- Tipikus alkalmazási területek
 - Help desk alkalmazások
 - Számla lekérdező rendszer
 - Hagyományos szerverek webes felülete

- Jellemző: A kliens ügyis kénytelen megvárni

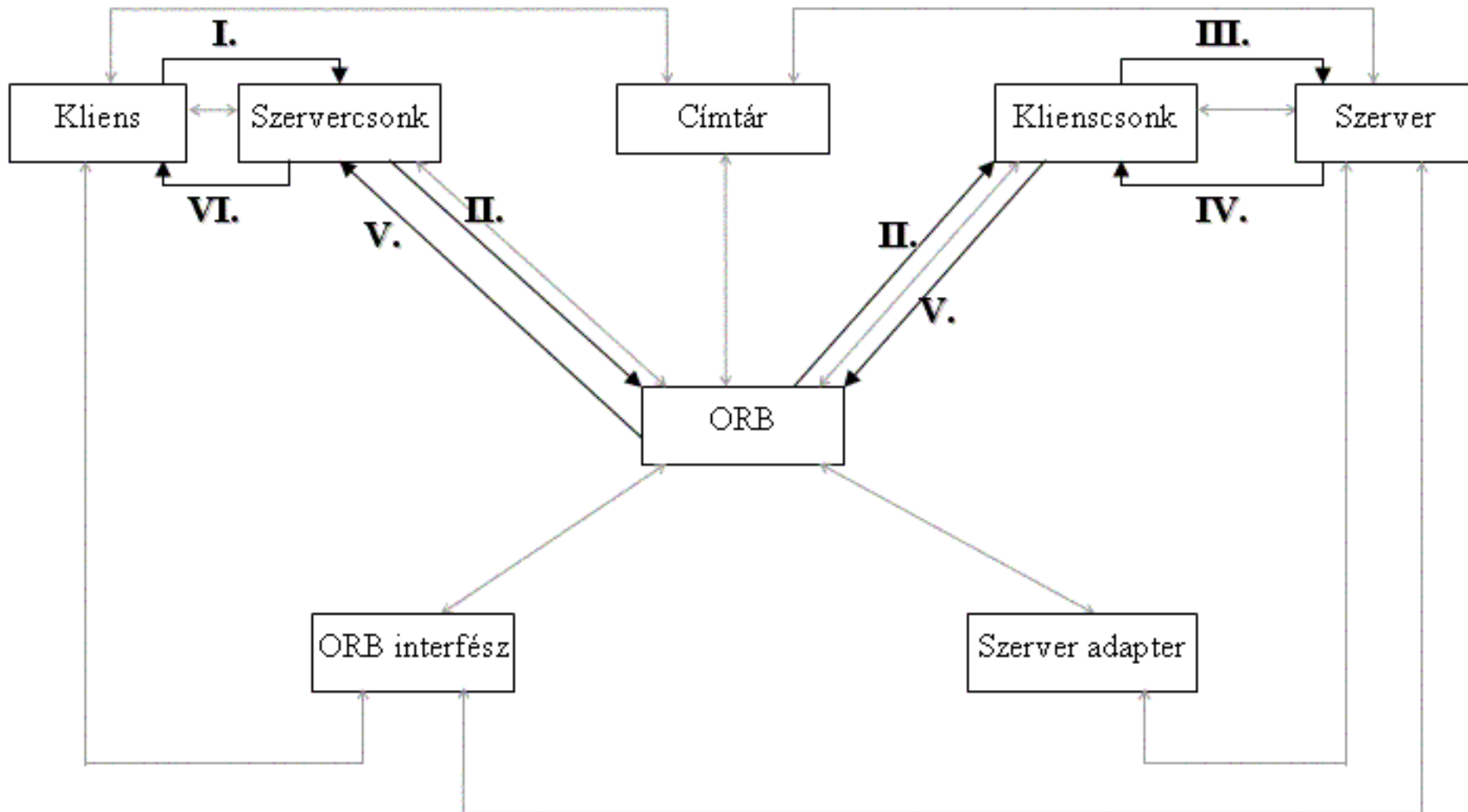
ORB architektúra modellek

- Szabó Péter: Távoli eljáráshívás alapú middleware rendszerek modellezése (Diplomaterv, BME MIT, 2003)
- Általános modell
- CORBA
- DCOM
- RMI

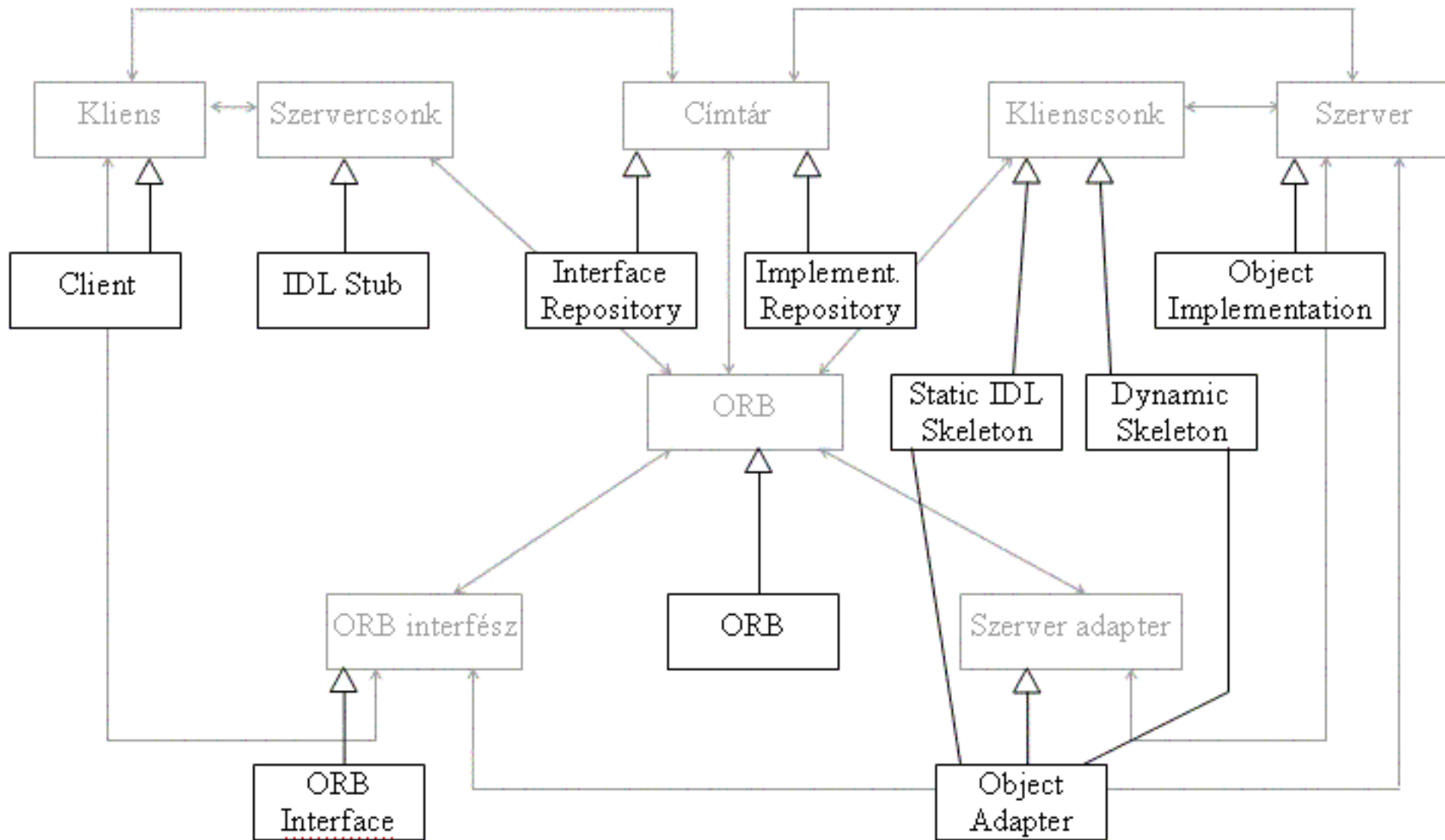
Általános ORB architektúra



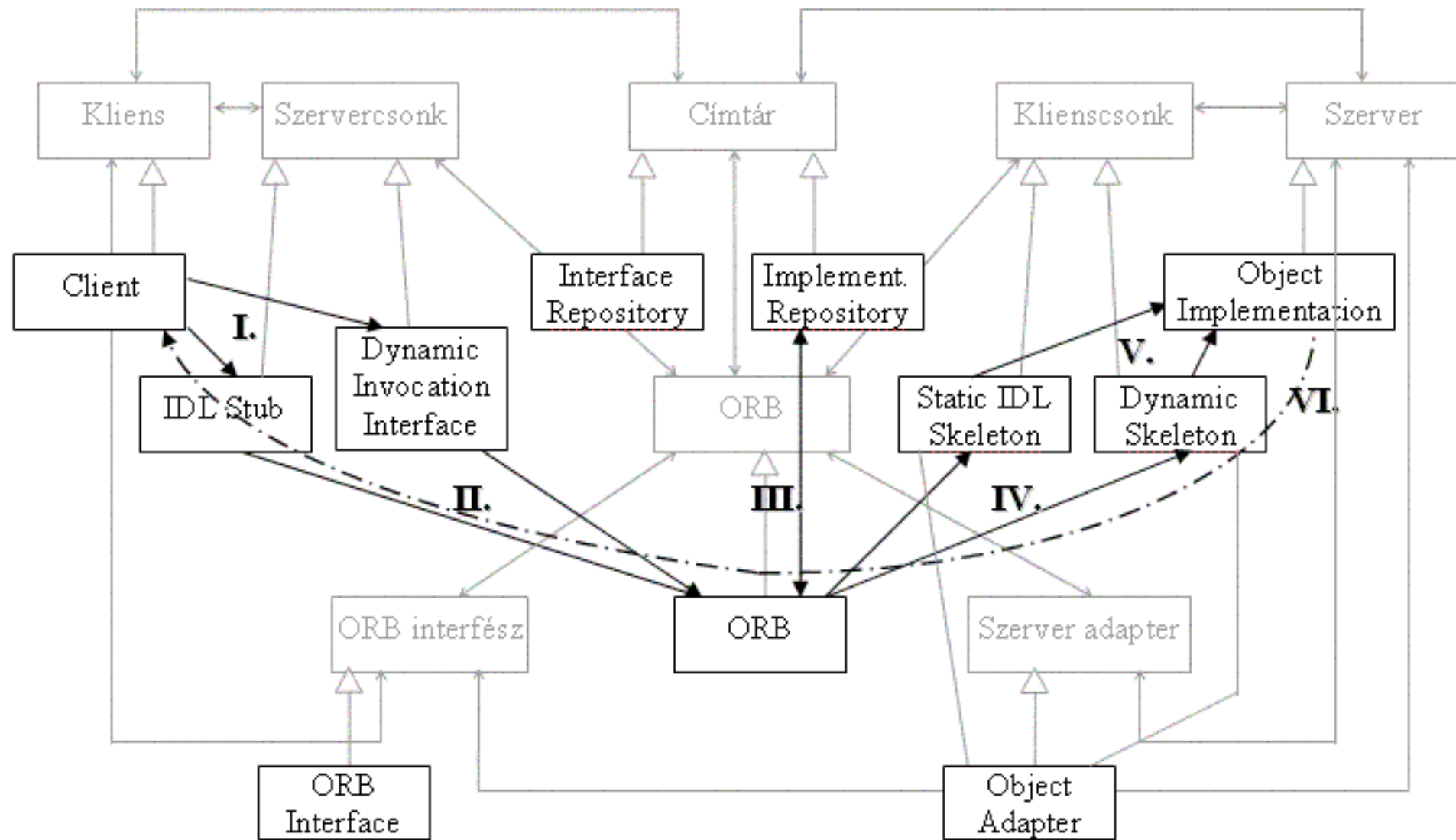
Általános ORB kommunikáció



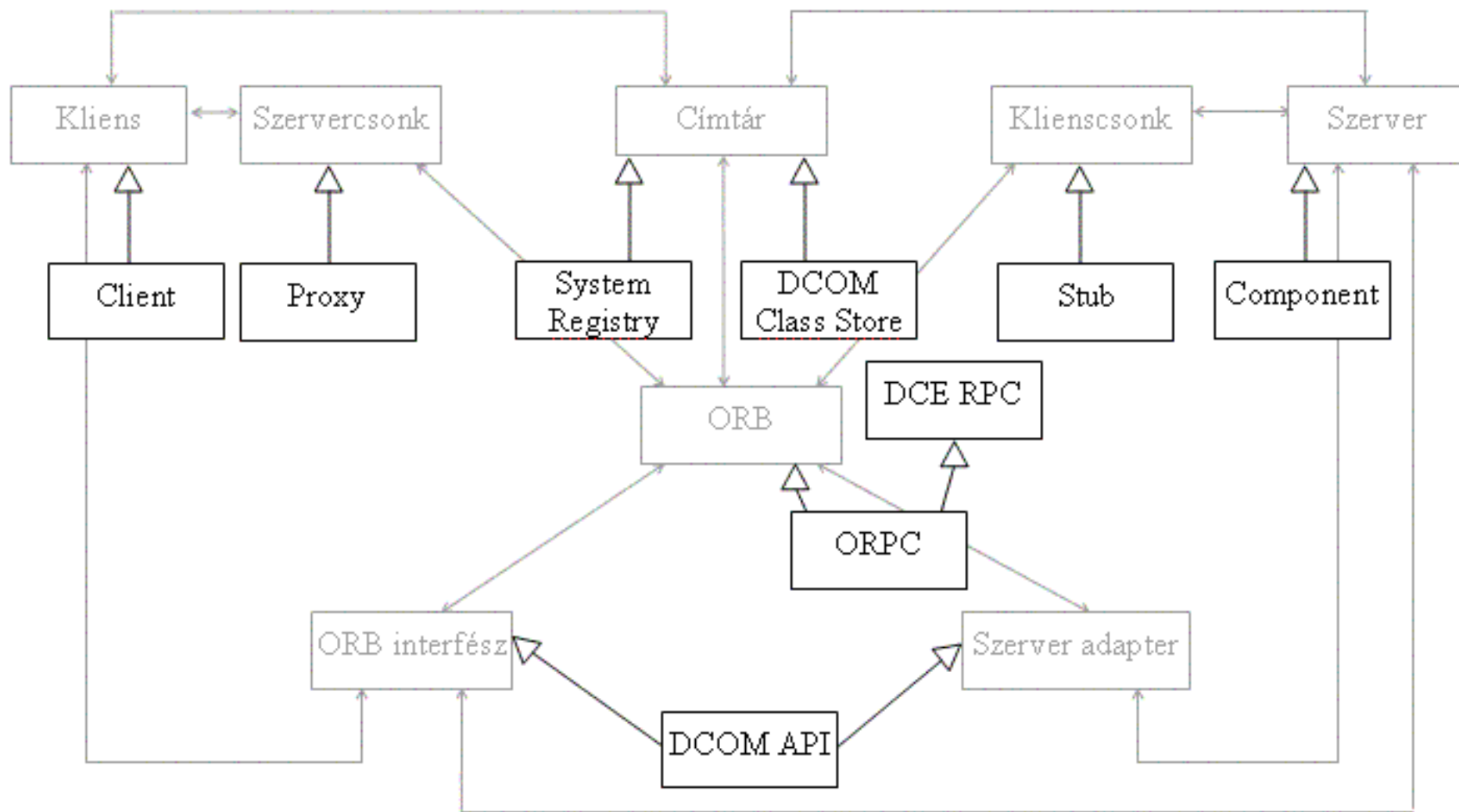
CORBA architektúra



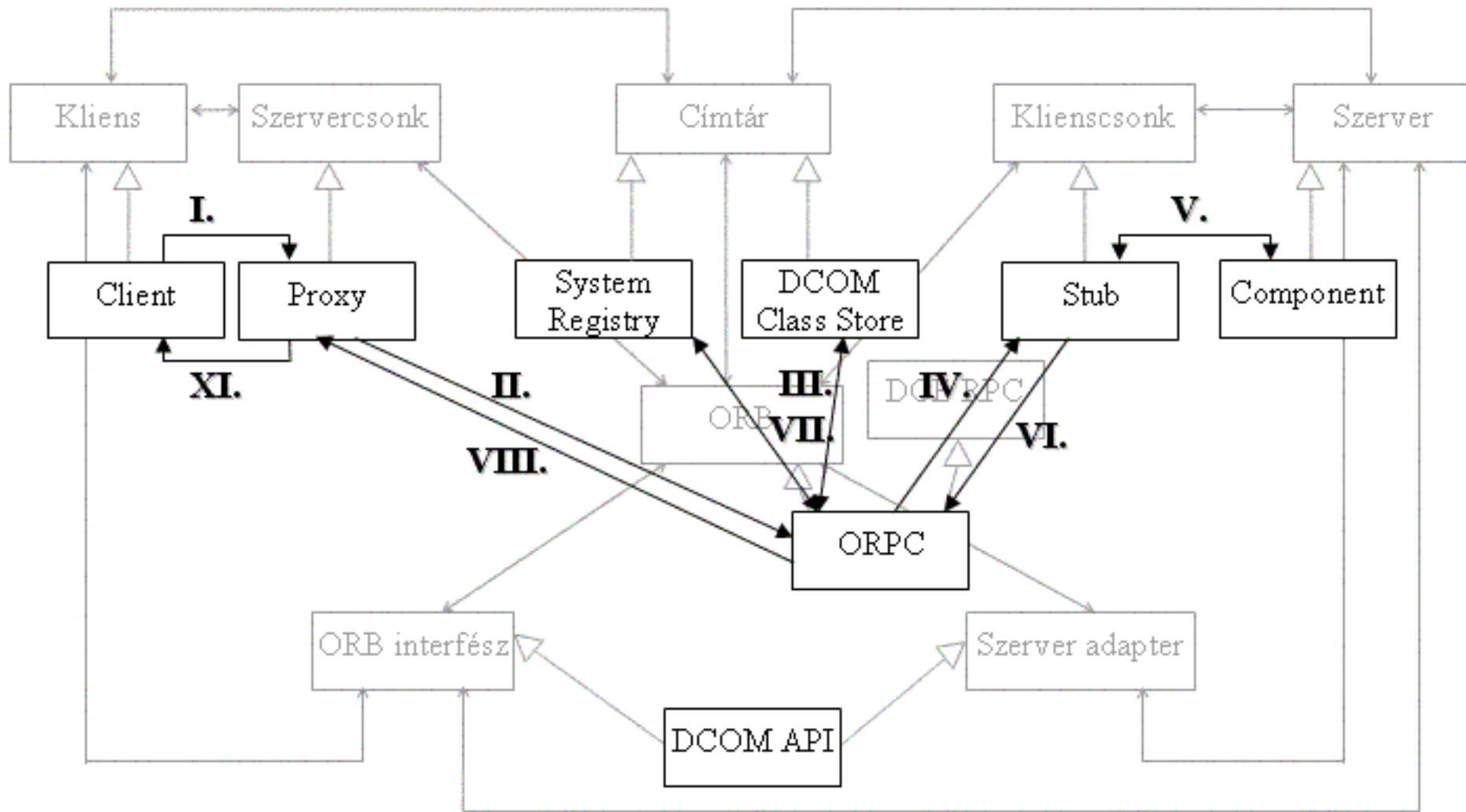
CORBA kommunikáció



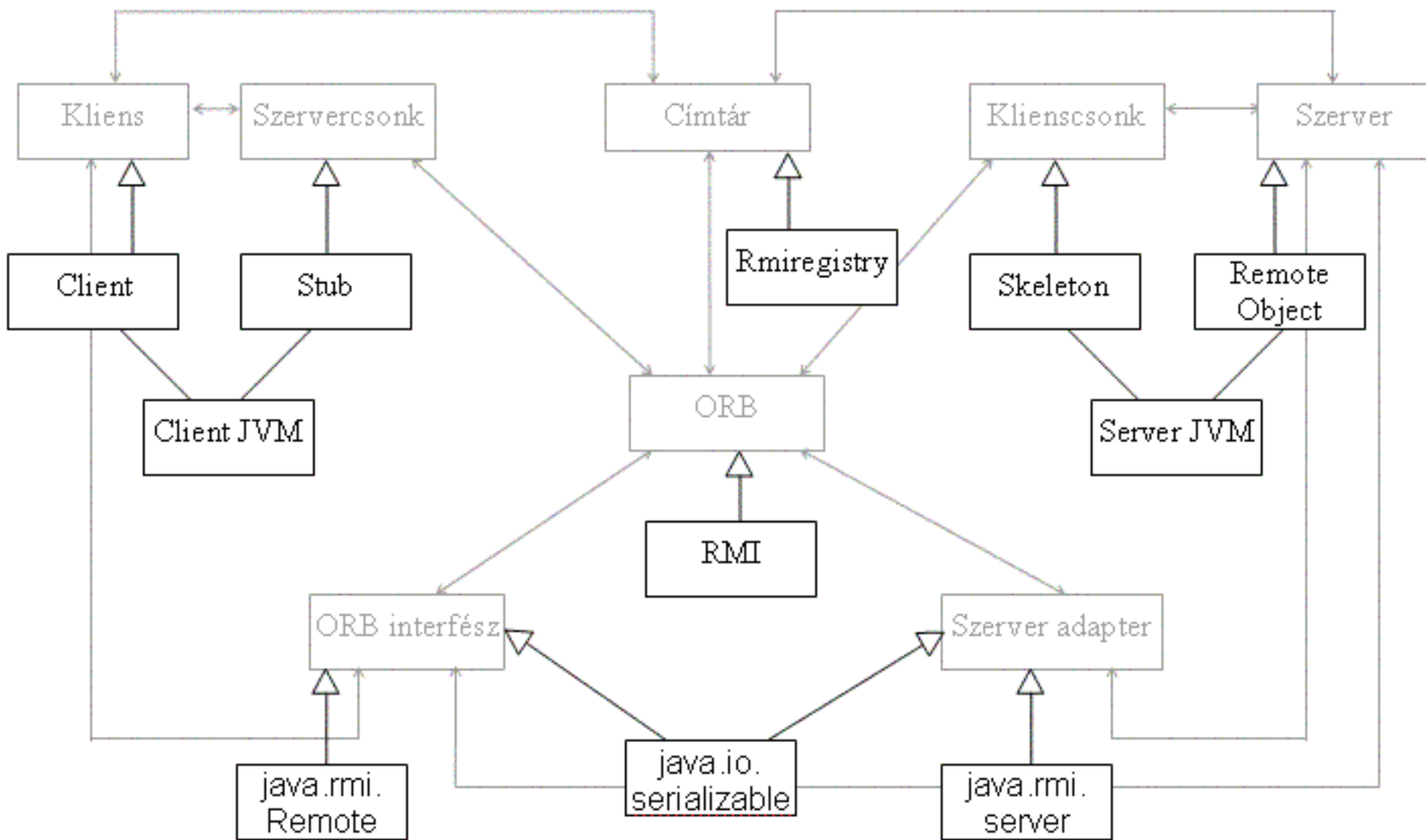
DCOM architektúra



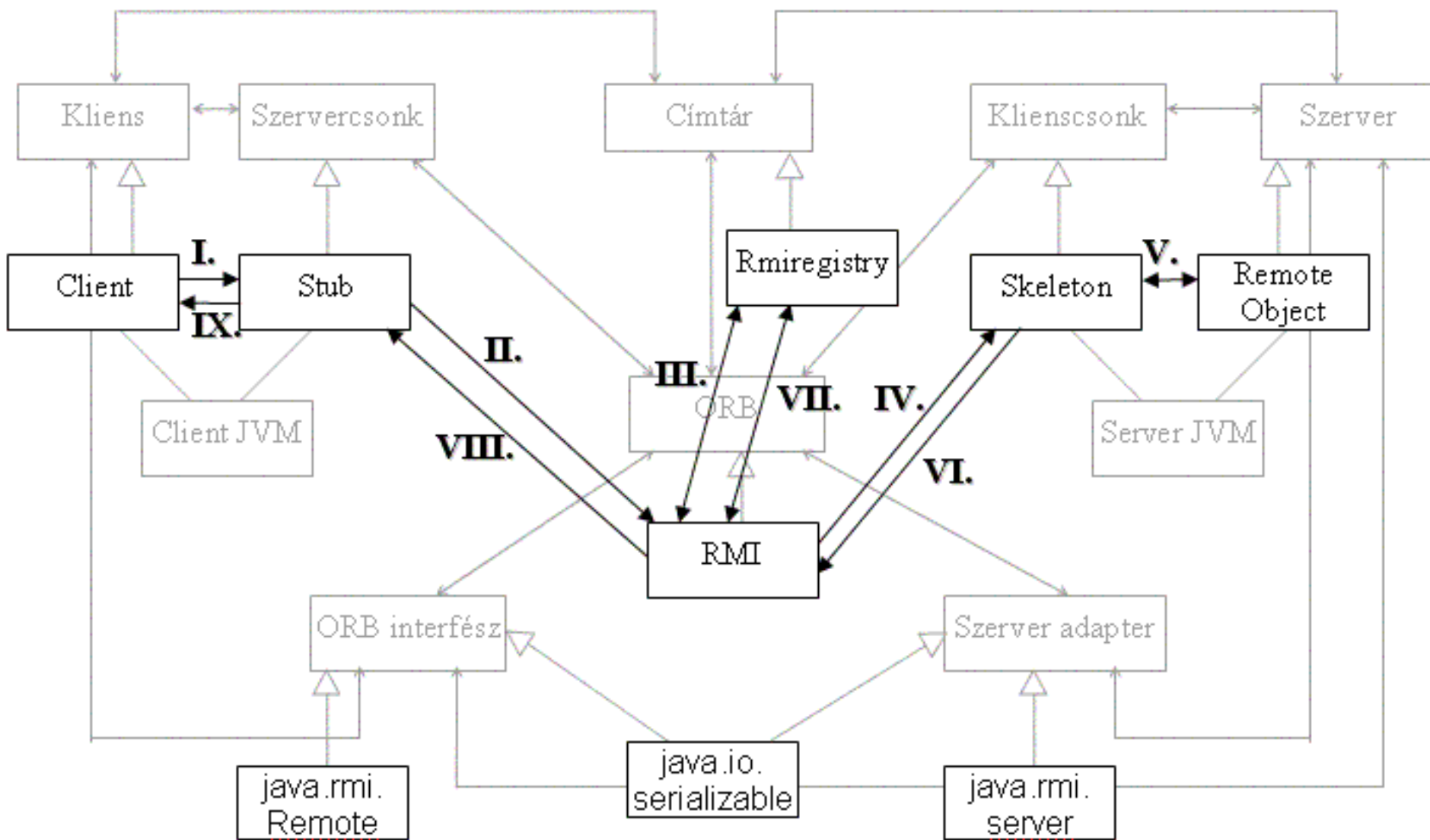
DCOM kommunikáció



RMI architektúra



RMI kommunikáció



Korszerű middleware technikák

- Házilagos megoldások
- Távoli eljáráshívás (RPC, ORB)
- **Üzenetsorok (MQ)**
- Publish-Subscribe (P/S)
- Egyéb aszinkron technikák

Üzenetsorok

- Jellemzők
 - Üzenet-orientált, aszinkron
 - Inkább csak „... az egynek” kommunikáció
 - Laza csatolás
 - nincs közvetlen kapcsolat
 - nem szinkronizálódnak
 - nem fogják vissza, nem rántják le egymást
 - Nagy megbízhatóságú → nem gyors
- Alapfogalmak
 - Üzenetek: átküldeni szánt információ adag (msg.)
 - Sorok: üzenetek elosztói, tárolói (queues)

Sorok feladatai

- Üzenetek fogadása a küldőtől
 - Aszinkronitás (vezérlés visszaadása a lehető leghamarabb)
 - Tárolás, míg címzett át nem veszi
- Postafiók rendszerű működés
 - Címzett változhat, ha a postafiók marad
- Üzenet nem vehet el, amíg a sor él
 - Alkalmazásokat, MW-t futtató gépek leállása
- Üzenetek transzformációja
 - Interpretálhat, konvertálhat
 - Csak egyszerűbb átalakításokra van idő, kapacitás

QoS

- Garanciák (külön be kell állítani, ha lehet)
 - üzenet nem veszhet
 - üzenet nem duplikálódhat
 - sorrend nem cserélődhet fel
- Szintek (másik megközelítés)
 - 0 – „best effort”
 - 1 – kézbesítés legalább egyszer
 - 2 – kézbesítés pontosan egyszer

Üzenetsorok

- Előnyök
 - Nagy megbízhatóságú hálózati kommunikáció
 - Mobil (off-line) partnerek kommunikációja
 - Új és hagyományos rendszerek laza csatolása
- Hátrányok
 - Nehézkes inicializálás és adminisztráció
 - Lassú, ha a sor hosszú
 - „Sok soknak” komm. nehezen megvalósítható

Üzenetsorok

- Tipikus alkalmazási területek
 - Webes megrendelés felvétel
 - feldolgozás hagyományos alkalmazásokkal a háttérben, lassan, az ügyfelet elengedve
 - Ügynöki kiszolgáló rendszer
 - ügynökök off-line szakaszainak tolerálása
 - Tranzakciós rendszer felhasználói felülete
 - Felület omlása ne vigye magával a tranzakciót

Korszerű middleware technikák

- Házilagos megoldások
- Távoli eljáráshívás (RPC, ORB)
- Üzenetsorok (MQ)
- **Publish-Subscribe (P/S)**
- Egyéb aszinkron technikák

- Jellemzők
 - Üzenet-orientált, aszinkron
 - Jó „... a soknak” kommunikáció
 - Laza csatolás (mint az MQ-nál)
 - Rugalmas komm. vegyes hálózati környezetben
 - Terjesztő transzformálhatja az üzeneteket
- Alapfogalmak
 - Üzenetek: átküldeni szánt információ adag (msg.)
 - Terjesztő: fogadó és elosztó hálózat
(publishing service, publishing network)
 - Előfizetők: terjesztőnél regisztrált címzettek
(subscribers)

Terjesztő feladatai

- Üzenetek fogadása a küldőtől
 - Aszinkronitás (vezérlés visszaadása a lehető leghamarabb)
 - Címzettek azonosítása, útvonalválasztás
- Üzenetszórás a regisztrált címzettek felé
 - Terjesztő hálózat optimális kihasználása
- Regisztrációs lista folyamatos, központosított karbantartása
- Feliratkozási rendszerek (alap típusok)
 - Kategória alapú
 - Kulcsszavas
 - Mintaillesztős

- Előnyök
 - Komm. folyamatosan változó partnerekkel
 - Jól skálázódó „... soknak” kommunikáció
 - Időre érzékeny hagyományos rendszerek összekötése
- Hátrányok
 - Nehezen tranzakciósítható
 - Nehézkes üzemeltetés és hibakeresés
 - Robusztus, teljesítőképes hálózat kell alá

- Tipikus alkalmazási területek
 - Valós idejű árverező és tőzsdei rendszerek
 - bárki bármikor bármire fel-/leiratkozhat
 - bármelyik tag küldhet
 - Időjárás-jelentő rendszer
 - hírügynökségek feliratkoznak (terület, esemény)
 - jelentések folyamatosan mennek ki
 - jelentés készítése és előfizetés kezelése szétválik
 - Szolgáltatás-orientált rendszerek
 - „Nem tudom, kinek a dolga, de valaki csinálja meg”
 - bizonyos funkciókra mindig van előfizető
 - Hálózati riasztórendszer
 - hálózat gépei be-/kikapcsoláskor fel-/leiratkoznak

Korszerű middleware technikák

- Házilagos megoldások
- Távoli eljáráshívás (RPC, ORB)
- Üzenetsorok (MQ)
- Publish-Subscribe (P/S)
- **Egyéb aszinkron technikák**

Egyéb aszinkron megoldások

- Aszinkron kommunikáció sokszor hasznos
- MQ és P/S infrastruktúrája költséges
 - Időben és pénzben is
- Sokszor nem a nagy megbízhatóság a lényeg
- Sokszor fix, ismert a címzett

- Fire and forget (FF)
- Ajánlott üzenetek (Sync with server)
- Lekérdezés (Polling)
- Visszahívás (Call back)

Fire and forget

- Szerver nem ad visszatérési értéket
 - Pl. egyoldalú értesítések mennek
- Hibaüzenetek sincsenek
 - Pl. megismételt üzenet már nem lenne aktuális (külvilág nem állítható meg, pörgethető vissza)
- Üzenetvesztés elfogadható
 - Pl. nem kritikus a szolgáltatás vagy „úgyis csak frissen jó”
- Példák:
 - Loggolás
 - Model – View – Controller

Fire and forget

- Megoldás
 - Lokális csonkkal szinkron kommunikáció
 - Csonk üzen távolra, de nem blokkol
 - új szálon fut,
vagy nem blokkoló kommunikációt használ

Ajánlott üzenetek

- Szerver nem ad visszatérési értéket
 - Pl. egyoldalú értesítések mennek
- Hibaüzenetek sincsenek, de visszaigazolás kell
 - Feldolgozás előtt, csak a kézbesítésről
- Megoldás:
 - Kliens oldali csomópont visszaigazolásig blokkol
 - Hálózati hibát detektál, szerver oldalit alig
 - Szerver oldali csomópont nem a feldolgozás szálán fut

Lekérdezés

- Aszinkron kommunikáció, de kell az eredmény
 - De nem kell azonnal
 - Szerverrel párhuzamosan dolgozó kliens
 - pl. a kért ID generálása közben a kliens létrehoz, konfigurál, kitölt (amit ID nélkül is lehet)
- Megoldás:
 1. Kliens oldali csomópont pollozza a szervert
 - kliens blokkolva, aszinkron ez?
 2. Kliens oldali csomópont blokkolva, kliens dolgozhat
 - kliens pollozza a csomópont egy szálát
- Hosszú pollozás → drága; rövid → szinkron jobb

Visszahívás

- Aszinkron kommunikáció, de kell az eredmény
 - Amikor polozni hosszú lenne
- Megoldás:
 - Kliens oldali csonk blokkolt szála az eredménnyel visszahívja a klienst
 - kliens call-back interfésze? ennek címe?
 - Több szálas kliens kell
 - nem transzparens módszer