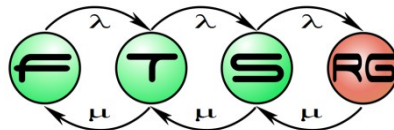


Nagyvállalati SOA infrastruktúra (ESB, szolgáltatástárak)

Szolgáltatásintegráció előadás

Huszerl Gábor, Gönczy László (BME MIT)



Kérdések

- Szolgáltatás-orientáltság platform függetlenül?
 - A WS is csak egy platform
 - A SOA alatt lehet más is
 - A szolgáltatás „örök”, a platform változik
- Hogyan érik el a szolgáltatások egymást?
 - kell egy platform (topológia is!)
 - Java API, WS világ
 - spagetti architektúra, Bábel
 - kommunikációs infrastruktúra (üzenetsín)

Szolgáltatás-orientált rendszerek

- Alapfogalmak
 - Szolgáltatás, alkalmazás, rendszer
 - Interfész
 - kiajánlott felület
 - Referencia
 - elvárt/használt felület
 - Megvalósítás
 - kibontás vagy technológiához kötés
 - Üzleti objektumok
 - üzenet/hívás adatstruktúrák

Service Component Architecture

- *Open SOA Collaboration* tervezési modellje

- SCA 1.0 2007 márciusában
- Formális szabványosítás: OASIS
- Jobbára Java környéki cégek

BEA, Fiorano, IBM, Iona, Oracle, Red Hat, SAP, Siemens, Sun, Sybase, Tibco, ...

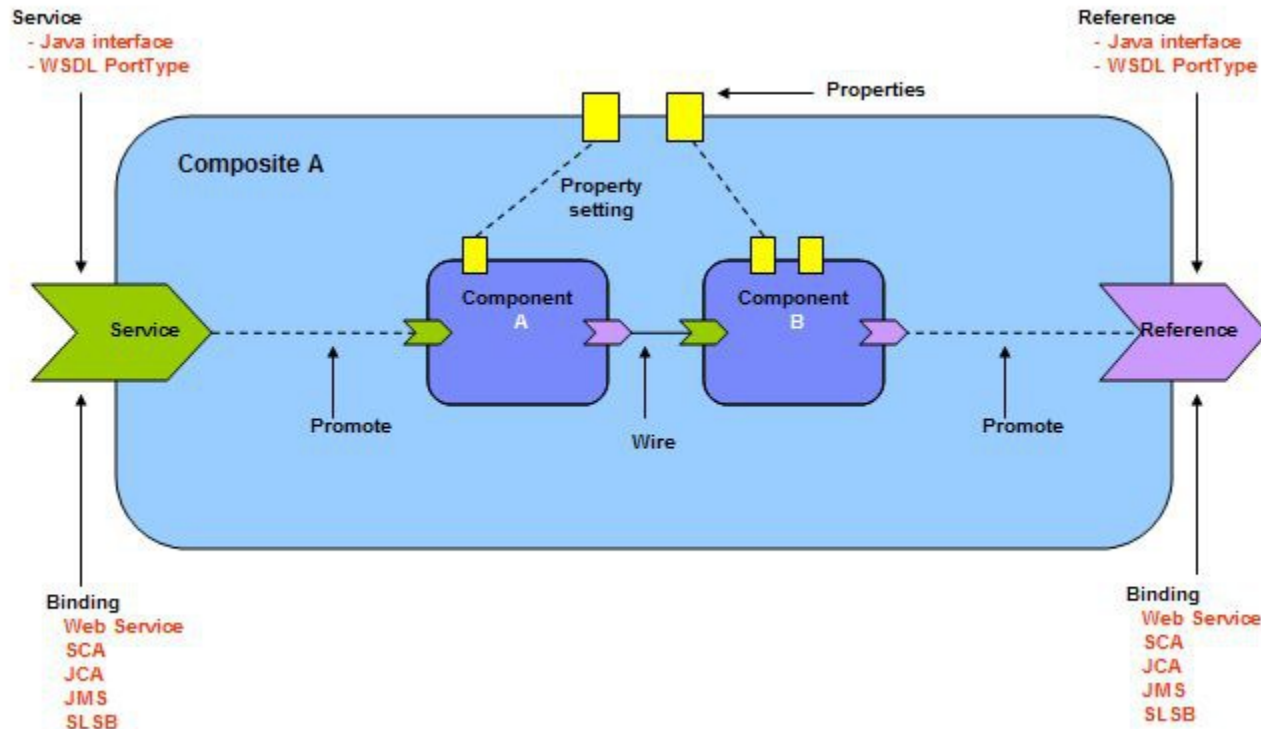
- Microsoft hasonló koncepciója:

Windows Communication Foundation

- Inkább komm. platform, mint tervezési modell

- Portolhatóságot tűz ki

SCA referencia modell



Kompozit: komplex, szolgáltatást nyújt, igénybe vesz

Szolgáltatás: publikus (absztrakt) interfész (+ huzalozás!)

Referencia: elvárt szolgáltatások (absztrakt) interfésze

Tulajdonság: paraméterek, konfiguráció

Komponens: a szolgáltatáslogika összetevője

Kompozit

- Komplex, szolgáltatást nyújt, igénybe vesz
 - Milyen elemekből? (top-down jelleg)
 - Melyik szolgáltatást ki nyújtja?
 - Ki kivel kommunikál?
- Lehet „telepítési” egység (deployment package)
- Lehet „azonos környezetben futási” egység
 - A belső kommunikáció nem *távoli* jellegű
- Lehet „láthatósági” egység (*visibility scope*)
 - „include”-olt elemek mindig látnak és láthatóak

Szolgáltatás, referencia

- Nyújtott szolgáltatás publikus (abszt.) interfésze
- Elvárt szolgáltatások (absztrakt) interfésze
- Leírás nyelve tetszőleges
 - Ha *távoli kommunikációt* feltételez, akkor *WSDL*-be fordíthatónak kell lennie
- Hosszú távú kapcsolódást külön jelölni
 - Session, conversation
- Implementáció:
 - WSDL, Java interfész/osztály, JMS binding, ...

Komponens

- A szolgáltatáslogika összetevője
- Alacsonyabb szintű kompozit (rekurzivitás)
- Java osztály, BPEL folyamat
- Implementáció:
 - BPEL, JSP, POJO, egyéb nyelvű program, selector, human task, business rule, business state machine, ...

SCA kötése hordozóprotokollhoz

- SCA platformfüggetlen, megvalósítás nem
 - Üzleti logika leválasztása az elérés megvalósításáról (de az is kell valahol!)
- Szolgáltatások/referenciák kötése (binding)
 - Hogyan hívható, hogyan akar hívni
 - SCA sokféle kötést ismer (WS, EJB, JMS, JCA, ...)
 - SCA alapú eszköz ezt bővítheti
- Vezetékezés megvalósítása: SCA motorra bízva
 - Mit köt össze a vezeték, mit ismer a motor
 - Ha interoperabilitás kell → WS

Adatáramlás szolgáltatások között

- Referenciák és szolgáltatások összehuzalozása
 - Szép, de önmagában nem elegendő
 - Adatstruktúra egyezést deklarál
 - Eltérő platformú szolgáltatások között probléma

- „Business Object on the Wire”
 - Java osztály
 - Service Data Object

Mire jó az SCA/SDO?

- Referenciamodell
- Eszközök építhetőek rá
 - Modellező eszközök
 - Fejlesztő eszközök (Java5 annotációk)
 - Végrehajtó motorok
- Szolgáltatás-orientált fejlesztés
- Meglévő alkalmazások beillesztése
 - Bármilyen SCA komponensbe csomagolható

Kérdések

- Szolgáltatás-orientáltság platform függetlenül?
 - A WS is csak egy platform
 - A SOA alatt lehet más is
 - A szolgáltatás „örök”, a platform változik
- **Hogyan érik el a szolgáltatások egymást?**
 - kell egy platform (topológia is!)
 - Java API, WS világ
 - spagetti architektúra, Bábel
 - kommunikációs infrastruktúra (üzenetsín)

A SOA infrastruktúra problémája

- Nagyvállalati informatikai rendszerek
 - sok funkció, sok alrendszer, szervezeti hierarchia
 - heterogén technológiák, evolúció (üzlet, technológia)
 - több száz/ezer szolgáltatás
 - ezek kommunikációja → szolgáltatássínek
 - ezek közben tartása → szolgáltatástárak

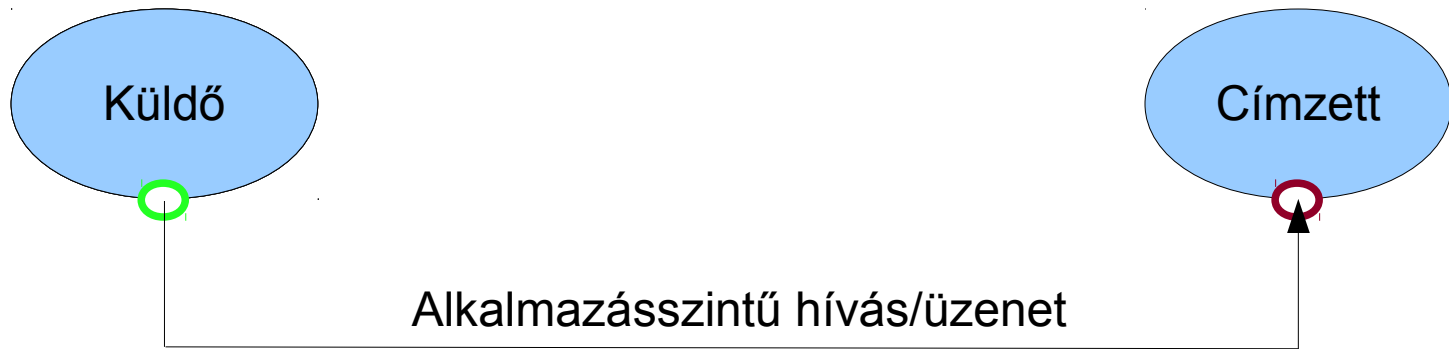
Platformfüggetlenség

- Szabványosítás (nemzetközi, állami, iparági/ipari)
- Cégek közötti megállapodás (piaci részesedés)
 - Nem kötelező érvényű
 - „Közös érdek”, közös álláspontot tükröz
 - Korlátozza a vevő „röghöz kötés”-ét
 - Korlátozza a gyártó innovációs szabadságát
- Többféle szinten lehetséges
 - Pl. portolhatóság, interoperabilitás

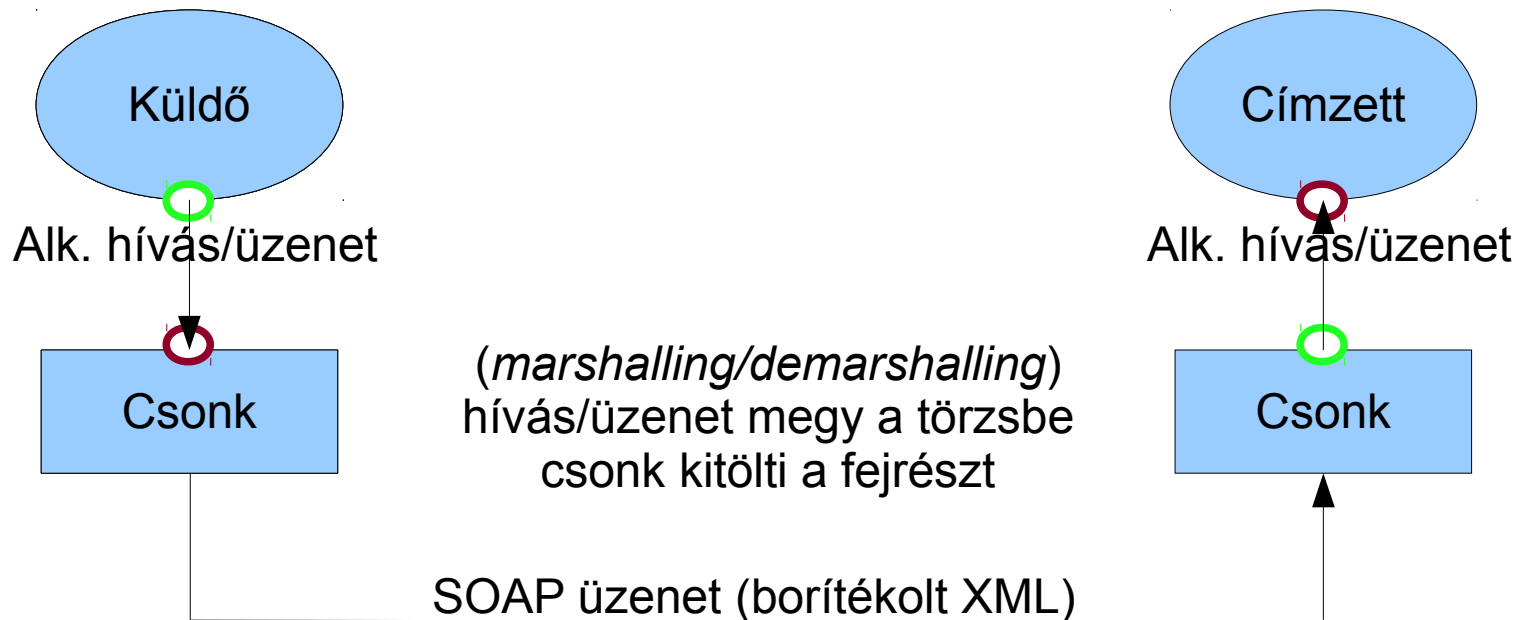
Szolgáltatás-orientált rendszerek & Java

- Java kiegészítése szolg. alapú integrációhoz?
 - JSR-208 (2005) Java Business Integration 1.0
 - JSR-312 (?) Java Business Integration 2.0
- WS + Java alapú EAI/B2B jellegű integráció (ESB)
- Java szabvány → csak egy API
- Nem vetélytársa az SCA/SDO-nak
- Implementációk
 - Apache ServiceMix, Fuse ESB, JBoss ESB, OpenESB, Sun Glassfish, TIBCO ActiveMatrix, Oracle Fusion, ...

SOAP alapú kommunikáció



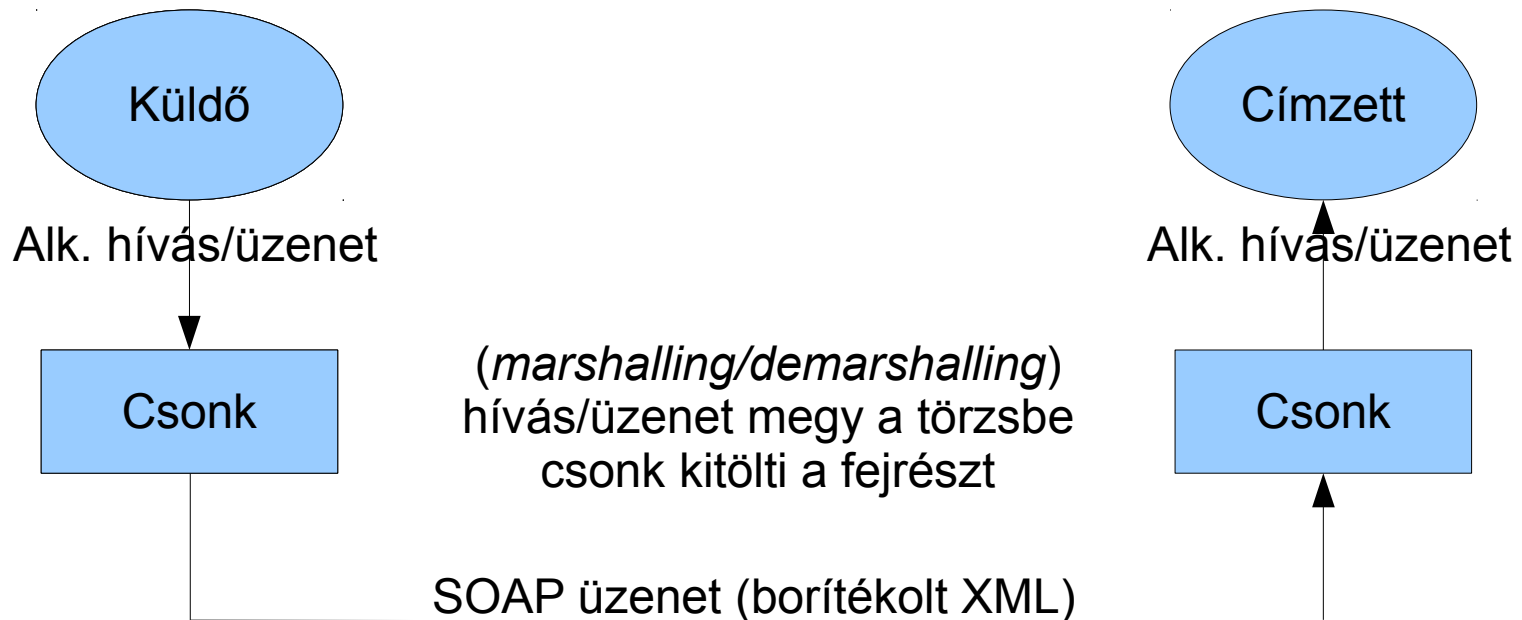
SOAP alapú kommunikáció



- **Marshalling**

- kódolt üzenetek (SOAP Section 5)
(Programnyelvi típusok kódolása SOAP szabvány szerint. Ma ritka.)
- betű szerinti üzenetek (literal)
(Kódolás a WSDL-beli XSD szerint, akár komplex típusok. Általános.)

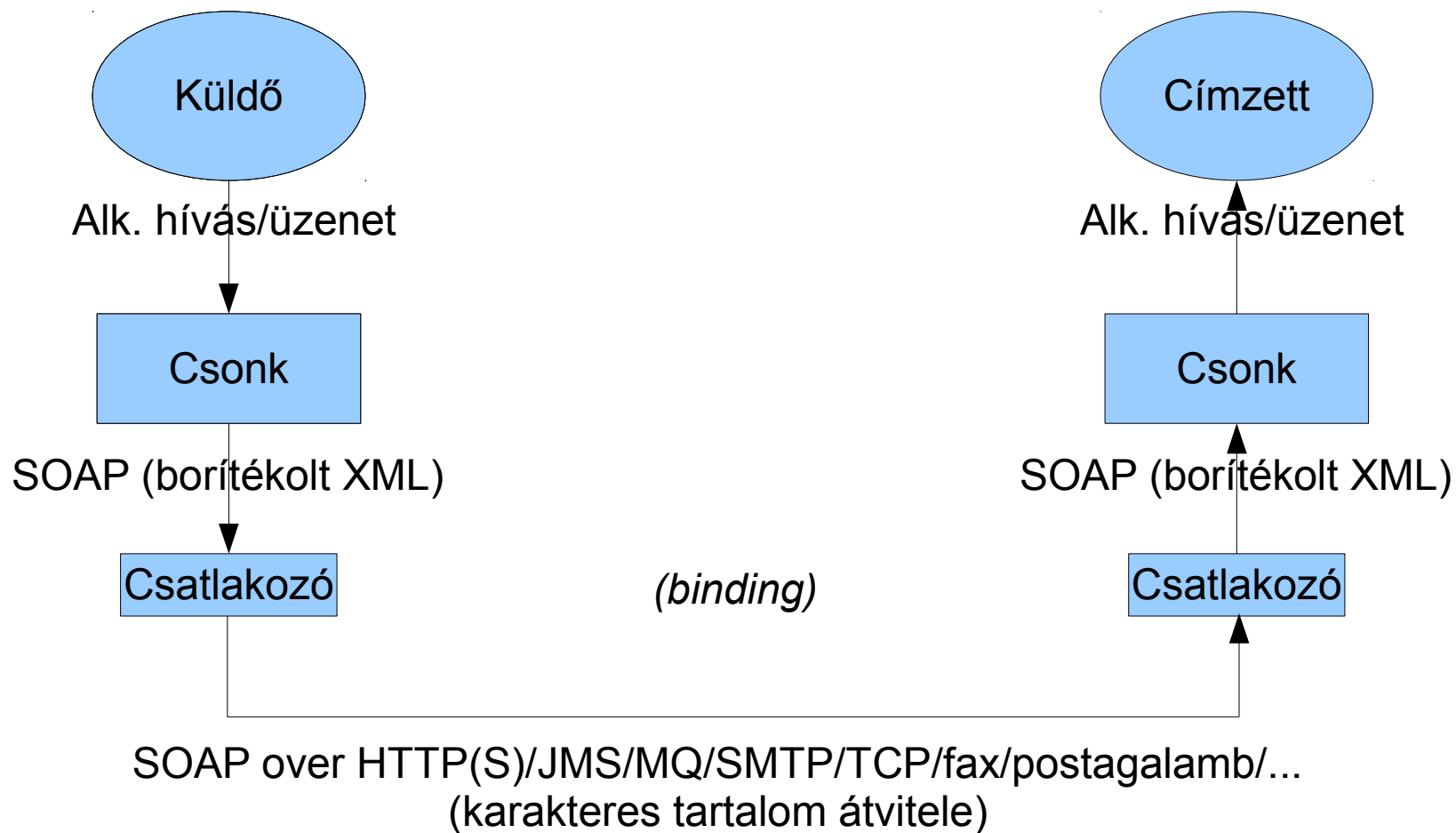
SOAP alapú kommunikáció



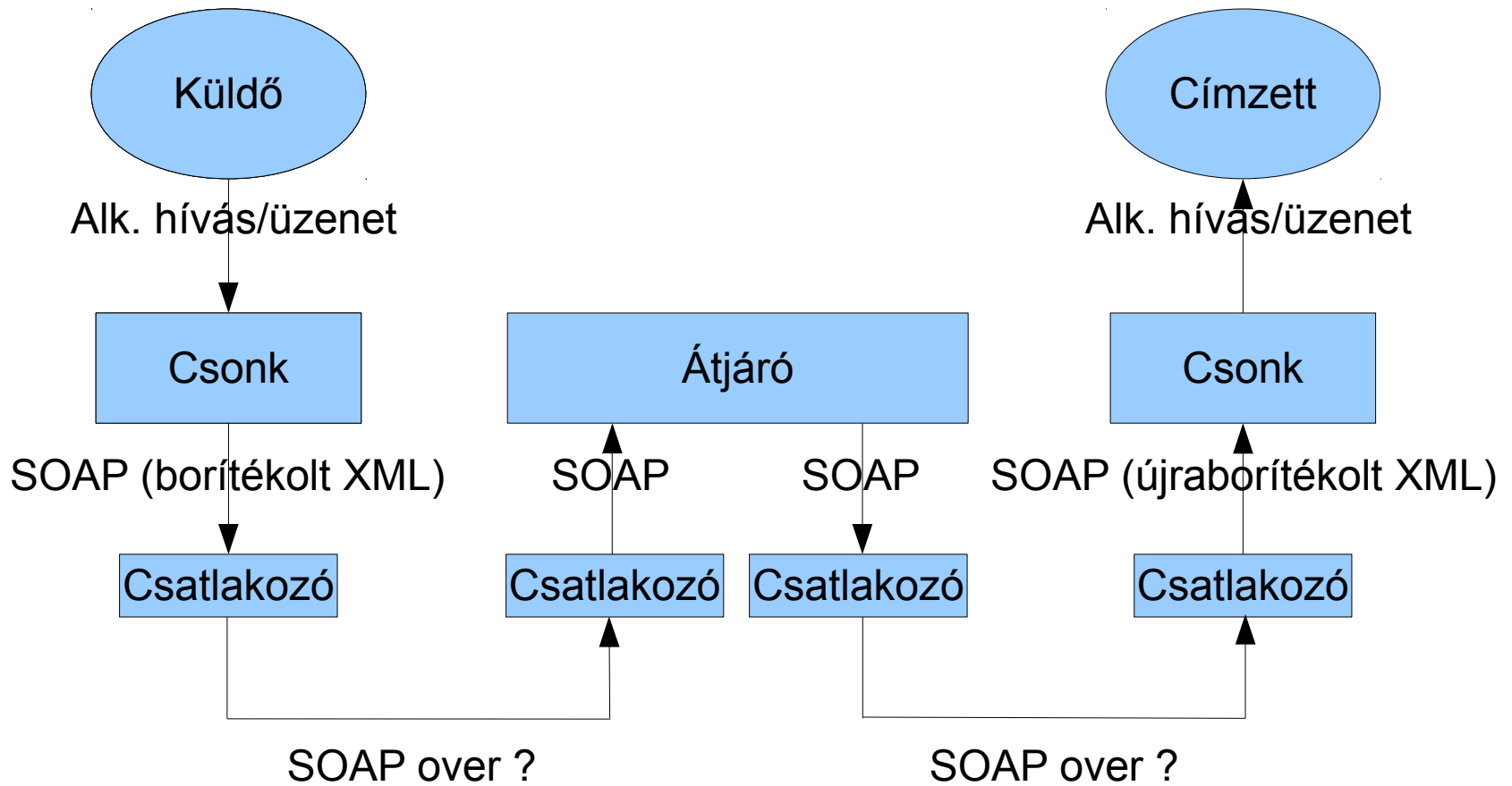
- Marshalling

- XML-RPC (SOAP Section 7) SOAP használata RPC-re (*metódusnév, metódusnévResponse, Fault*; kódolt üzenetek)
- Document (lehet kódolt vagy betű szerinti)

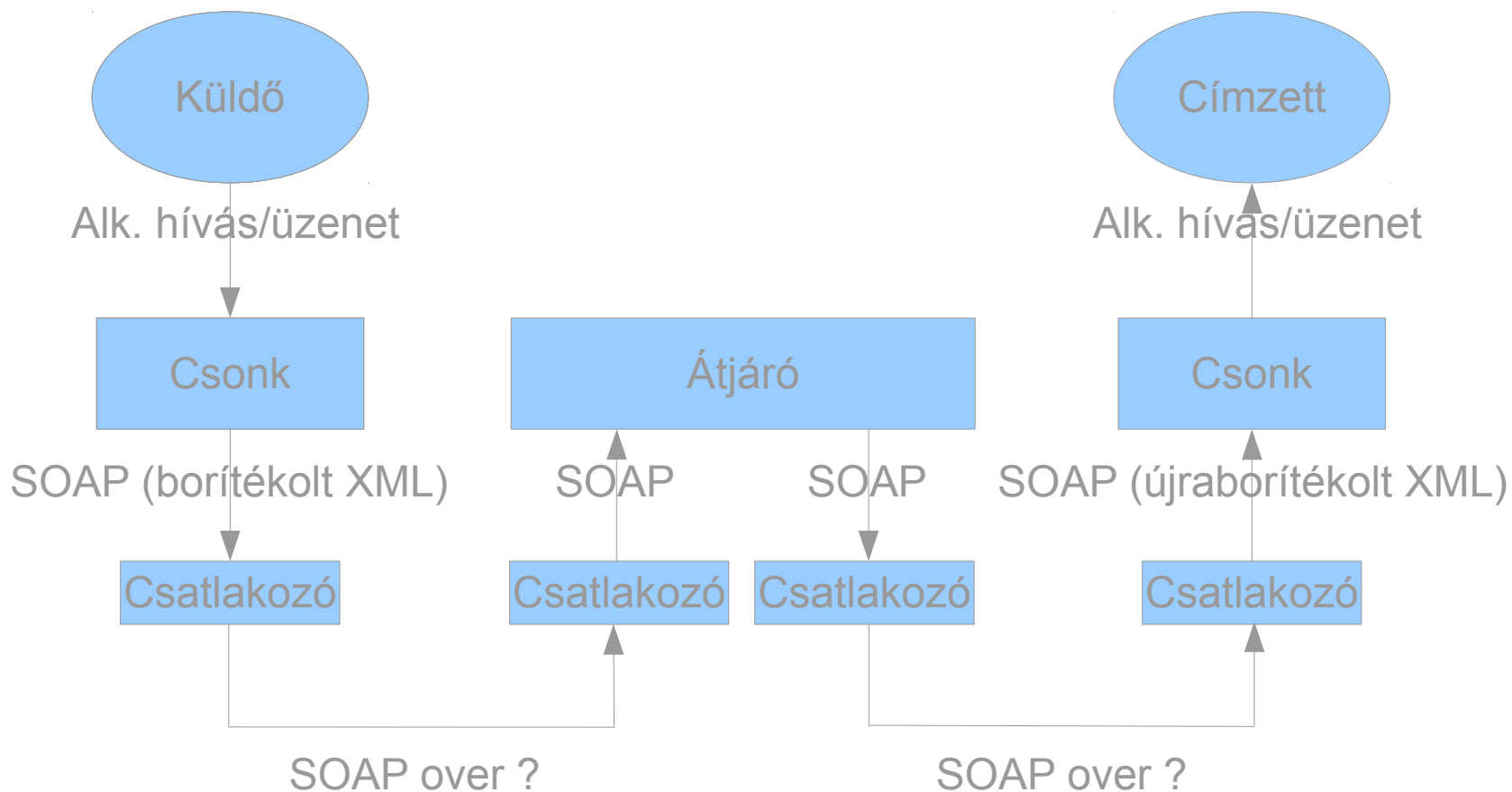
SOAP alapú kommunikáció



SOAP alapú kommunikáció



SOAP alapú kommunikáció

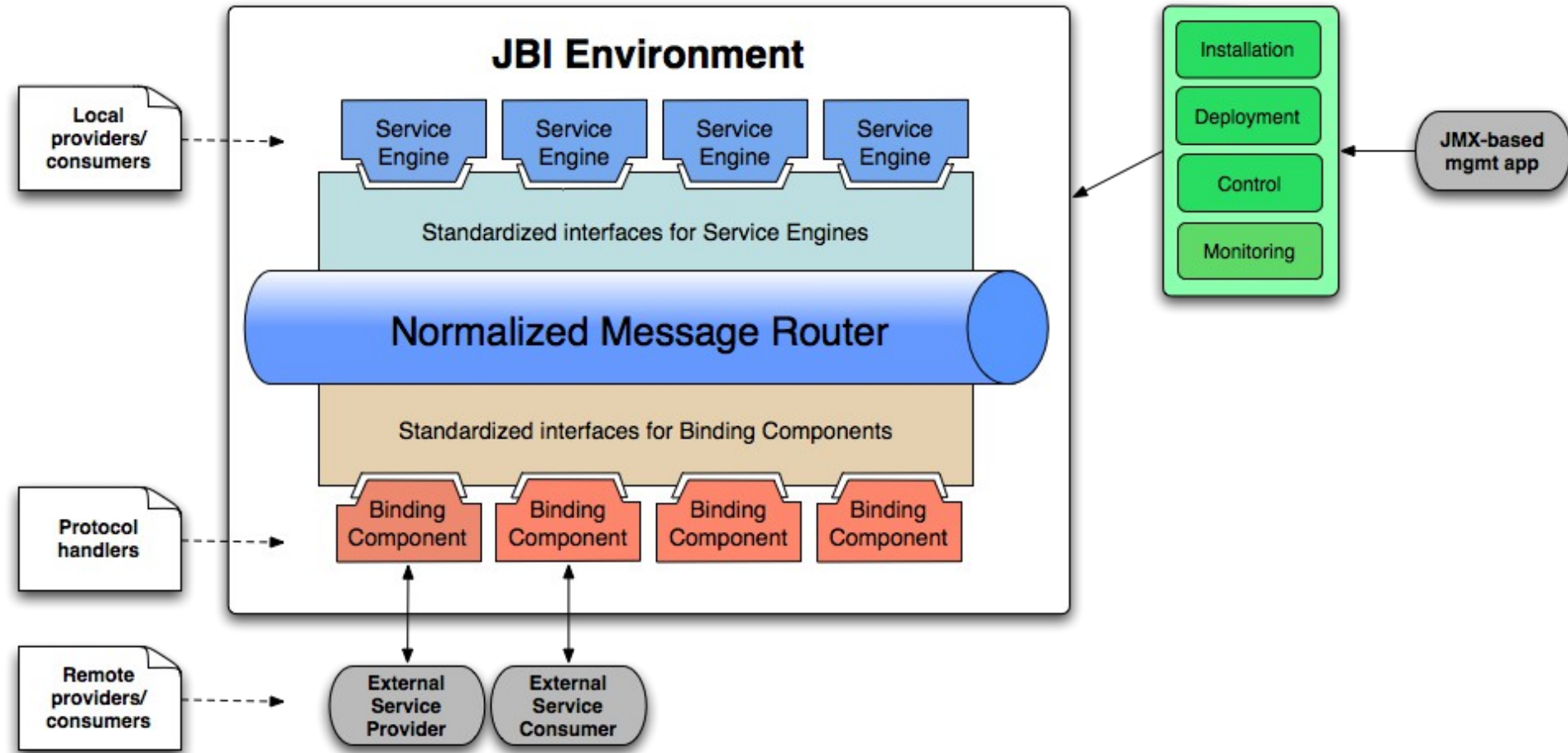


- Ezért nem bízhatunk mindent a hordozó protokollra
 - WS-* (WS-Addressing, WS-RM, WS-Security, ...)

Kommunikációs topológiák

- „Hagyományos” WS:
bedrótzott pont-pont topológia
 - minden változáskor kód módosítás
 - módosítás hatása beláthatatlan
 - kereskedelmi termékekben lehetetlen
 - teljes gráf topológiához tart
 - rugalmatlan, üzemeltethetetlen
 - minden küldő minden címzethez külön csontot tart (esetleg több átviteli protokollhoz többet)
- Sín topológia jobban kézben tartható

JBI környezet



ESB – Enterprise Service Bus

- Nagyvállalati szolgáltatásain
- Kommunikációs infrastruktúra
 - nem pont-pont kapcsolat
 - aszinkron kommunikáció (üzenet alapú) (laza csatolásért)
 - eseményvezérelt működés
 - üzenetek intelligens szűrése, átalakítása
 - üzenetek intelligens irányítása (útvonalválasztás)
 - protokoll átjáró (HTTP, MQ, SMTP, ...)

ESB definíciók 1.

Gartner Group:

EAI rendszerek olcsó, egyszerű alternatívája

IDC:

a jövő nyílt, szabványos összekapcsolási gerince

ZapThink:

szolgáltatás-orientált interfésszel rendelkező
üzenetsín

ESB definíciók 2.

Kisebb gyártók:

WS, JMS alapú middleware termékek

Nagyobb gyártók:

MQ, workflow alapú EAI termékek/rendszerek

Elmélet:

Nagyvállalati információs rendszerek SOA alapú összekapcsolási infrastruktúrája

ESB elődei

- Nagyvállalati üzenetkezelés (enterprise messaging)
 - nagy sebességű, aszinkron, bárki-bárkivel
 - nagy megbízhatóság, jó skálázhatóság
 - akár saját formátum
 - nem szolgáltatás-orientált
- Üzenetközvetítők (message brokers)
 - laza integráció, P/S
- Web Services technológiák
 - nyílt szabványok, platformfüggetlenség

Első generációs ESB-k

- Csak nyílt szabványokra építenek
- Olcsó megvalósítás
- Senki nem kompatibilis velük
 - fokozatos átállás lehetséges csak
- Kiforratlan szabványok
 - WS, WS-*, JCA, ...
- Teljesítmény és funkcionális korlátok

ESB – Elvárások 1.

- Kommunikációs middleware
 - szinkron/aszinkron, request-reply, one-way, call-back, ...
 - P/S funkciók (üzenetek, témák, előfizetés, elosztás)
 - QoS (biztonság, megbízhatóság, teljesítmény, tranzakciók)
 - szabványos API-k és protokollok
- Hívások/válaszok feldolgozása menet közben
 - fordítás, átalakítás, szűrés, kiegészítés (XSLT)
 - naplózás, (üzleteti) monitorozás
- Szabványos kapcsolódási lehetőség

ESB – Elvárások 2.

- Lazán csatolt komponensek és együttműködésük menedzselése
- Érvényesség és jogosultság ellenőrzés
 - üzenet formátum (tartalom) érvényessége
 - legalább XML validálás
 - küldő identitása és jogosultsága, üzenetváltások
 - címzett nem „látja” a küldőt
 - jogosultságok központi karbantartása

Érdekes ESB megoldások 1.

- P2P alapú ESB
 - Minden gépen egy-egy P2P szerver
 - teljesítmény, skálázhatóság
 - P2P szerverek hálózata
 - belül RPC+MQ megoldások, nem érdekes
 - „super peer” központi funkciókkal (menedzsment, biztonság)
 - WS interfész a helyi üzleti komponensek felé

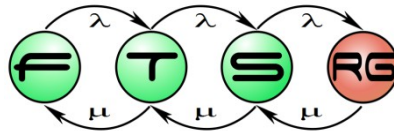
Érdekes ESB megoldások 2.

- Útiterv alapú irányítás
 - Tartalom alapú helyett
 - Feladó minden érintendő komponenst előre megad (mint a poggyásznál a repülőn, azután „letépkedés”)
 - ESB komponens transzformálható, kifelé akciózható, de belül továbbad
 - Nem kell központi komponens („mediáció”)

SOA infrastruktúra (tárak)

Szolgáltatásintegráció előadás

Huszerl Gábor, Gönczy László (BME MIT)



Kérdések

- Mi legyen, ha sok szolgáltatás van?
 - Hogyan érik el a szolgáltatások egymást?
 - spagetti architektúra, Bábel
 - kommunikációs infrastruktúra (üzenetsín)
 - Milyen szolgáltatásaink vannak már?
 - szolgáltatások életciklusa, verziók
 - szolgáltatás táruk

A SOA infrastruktúra problémája

- Nagyvállalati informatikai rendszerek
 - sok funkció, sok alrendszer, szervezeti hierarchia
 - heterogén technológiák, evolúció (üzlet, technológia)
 - **több száz szolgáltatás**
 - ezek kommunikációja → szolgáltatássínek
 - ezek **kézben tartása** → szolgáltatástárak

Szolgáltatástárak

- Service registry (nyilvántartás, nem repository!)
- Klasszikus WS technológia
 - Csak szolgáltatások közvetítésére
 - Táruk használata megvalósításkor
 - Elkészít → bejegyez
Megkeres → letölt → csonkot épít → használ
 - UDDI, WSIL szabványok
- SOA elképzelés
 - Teljes életciklus kezelés (ötlettől felszámolásig)
 - Tervezéskor, megvalósításkor, futás közben

UDDI

- Universal Description, Discovery and Integration
 - Szolgáltatások szabványos regisztrációja és felderítése
 - Megosztott, kereshető web alapú tár (telefonkönyv)
 - telefonkönyv színek (fehér, sárga, zöld)
 - Eredetileg tár az ebXML-hez (2000-2002 környékén)
 - Publikus, privát, osztott/félprivát táarak

Universal Business Registry

- UDDI Business Registry (UBR, UDDI alapú szolgáltatástár)
- Felépítése:

Business Entity (cég, cég rész)

↳ Business Service (szolgáltatás)

↳ Binding Template → t_Model (adott című interfész megvalósítás milyen szabványoknak/specifikációknak felel meg)

↳ Binding Template → t_Model

↳ Business Service

↳ Binding Template → t_Model

Business Entity

↳ Business Service

↳ Binding Template → t_Model

UDDI hátrányai

- Bejegyzések 2/3-a használhatatlan
 - Karbantartás, megszűnő szolgáltatások törlése
 - Publikus tárban kinek a felelőssége?
- Publikus tárok üzleti modellje?
- Szó sincs SLA-ról, QoS-ről
 - A szolgáltatások szemantikájáról sincs
 - gyártóspecifikus megoldások
 - open source

http://www.tutorialspoint.com/uddi/uddi_implementations.htm

WSIL – WS Inspection Language

- **WS Szemlélnyelv** (szemlélő/inspektor, szemrevételező)
 - IBM, MS (2001 vége) UDDI táruk helyett
- **„Bárki bárkivel” technikailag lehetséges**
 - Nem műszaki okok miatt mégsem
 - „Bárki bárki ismerőssel” – off-line kapcsolat fontos
 - Nem telefonkönyv kell, hanem névjegyek
 - Elérhetőségi adatok decentralizált publikációja
 - mindenki a saját adatait a saját webserveren
- **Fontos infók:**
 - Szolgáltatások/interfészleírások elérhetősége
 - Elérhetőségek elérhetősége

WSIL szokások

- Fix WSIL leírás címek
 - Minden WS-t nyújtó webszervernek illene
 - Humán és robot keresést is egyszerűsíti
- WSIL leírások egymásra hivatkozása
 - Hierarchikus elrendezés, szervezeti hierarchia
 - Szolgáltatások csoportosítása funkció vagy szolgáltató szerint
 - Karbantartás felelőssége tiszta

WSIL & UDDI

- WSIL szerint a szolgáltatás leírás helye lehet
 - WSDL fájl egy webserveren
 - Adott kulcsú bejegyzés egy UDDI tárban
- WSIL hivatkozhat
 - WSIL fájlra
 - UDDI „Business Entity”-re (egy cég UDDI-ban)
- UDDI-ban Business Entity adatai között
 - Központi WSIL leírása

SOA és szolgáltatástárak

SOA alapú megoldások

→ egy cégen belül is sok szolgáltatás

- A „káosz” uralása → governance (irányítás)
 - útmutatások, ellenőrzés és kikényszerítés
fejlesztés, üzemeltetés, rendtartások, metaadatok, eljárások,
szerepek, erőforrások, technológiák, bevált megoldások,
portfólió- és életciklus-kezelés, konzisztencia, teljesítmény
 - minőségbiztosítás (bürokrácia és áttekinthetőség)
 - ez már inkább „repository”
- A „káosz” uralása → menedzsment
(felügyelet)
 - folyamatok, életciklus, gazdálkodás, monitorozás

Szolgáltatástárak tartalma

- **Meglévő szolgáltatások listája**
 - Mi az, ami már van? Ki használja? (verziók, függőségek)
 - Elérhetőségek (felderítés), szemantika leírás (doksik)
 - Metaadatok (+szabvány/törvény konformancia)
- **Készülő szolgáltatások listája**
 - Nem csak azt nem kell legyártani, ami már van, hanem azt sem, ami már készül
- **Előfizetett szolgáltatások listája**
- **„Jó” külső szolgáltatások listája**

Hatékony újrahasznosítás!!!

Szolgáltatások metaadatai

- Szolgáltatás választás akár futási időben

- Metaadatok alapján

funkcionalitás, tervező, gondozó, jogosultságok, rendtartások, konfigurációs beállítások, üzleti paraméterek (business rules)

teljesítmény, rendelkezésre állás (mikor), SLA

verziók, életciklus fázis, várható indítás/váltás/leállítás ideje

tanúsítványok, vonatkozó szabványok/törvények

kapcsolódó protokollok, formátumok, adatstruktúrák, modellek, kapcsolódó szolgáltatások

aktuális terhelés, megbízhatóság, elérhetőség (fut-e, elérhető-e)

költségek, biztonsági besorolás

statisztikák, jelentések