

---

Sisteme distribuite – Tehnologii

5. Arhitecturi orientate spre servicii

---

---

# SOA

- Termenul a fost introdus de Yefim Natis intr-o lucrare de cercetare din 1994:  
*SOA este o arhitectura software care porneste de la o definitie de interfata si construiește întreaga topologie a aplicatiei ca o topologie de interfețe, implementari de interfețe si apeluri de interfețe*
- Desi termenul a fost introdus de multi ani, SOA a devenit buzzword numai la inceputul anilor 2000.
- Odata cu aparitia serviciilor Web si proceselor de business care respecta WSDL, SOA a devenit popular in randurile entuziastilor tehnologiei

---

# SOA – un stil arhitectural

- SOA este un *stil de design care ghideaza* toate aspectele crearii si utilizarii serviciilor prin tot ciclul lor de viata (de la conceptie la retragere).
- SOA este o modalitate de a defini si oferi o infrastructura IT pentru a *permite aplicatiilor diferite* sa schimbe date si sa participe la procese,
  - indiferent de sistemul de operare si
  - indiferent limbajele de programare care stau la baza acestor aplicatii
- O abordare de constructie a sistemelor IT in care primeaza necesitatea pentru business
  - Abordarile anterioare tind sa fie dependente de functionalitatile mediului de dezvoltare (vezi OO)

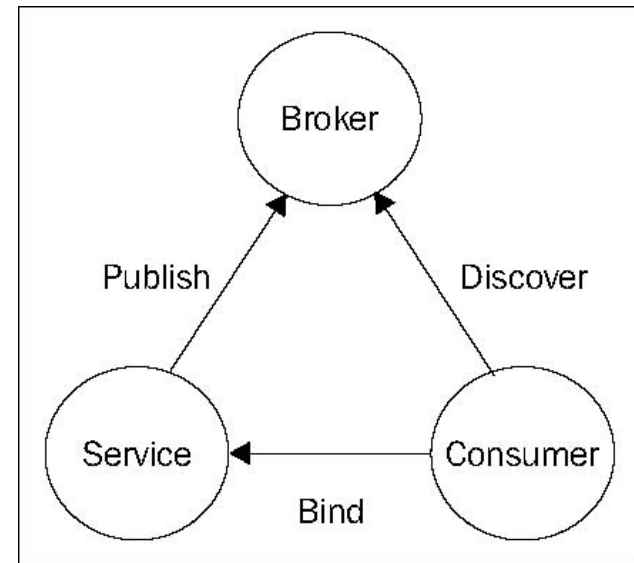
---

# SOA promoveaza reutilizarea software-ului

- Conceptul nu este nou:
    - Arhitecturile OO promoveaza reutilizarea prin reutilizare clase si obiecte
      - Obiectele sunt adesea prea rafinate pentru o utilizare efectiva
    - Arhitecturile bazate pe componente au aparut din necesitatea de a reutiliza entitati
      - Aceste componente consta intr-o multime de clase relationate, resursele lor si informatii de configurare
      - Nu adreseaza temele aparute in mediile curente ale companiilor:
        - Astazi mediile companiilor sunt complexe datorita utilizarii unei varietati mari de platforme software si hardware, comunicare bazata pe Internet, etc.
  - SOA utilizeaza serviciul ca si componenta reutilizabila
    - Serviciile au o granularitate mai mare decat componentele
    - Comunicarea dintre servicii si cu clientii se realizeaza pe baza unor interfete bine definite
-

# Fundamentele SOA

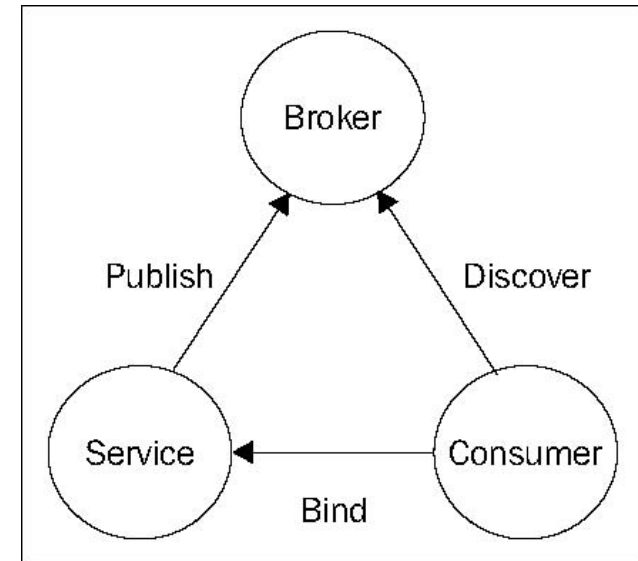
- Bazate de: Serviciu, Mesaj si Descoperire dinamica.
- In SOA exista:
  1. **Serviciul** care implementeaza logica si expune logica prin interfete bine definite
  2. **Registru** unde serviciul publica interfetele pentru a permite clientilor sa descopere serviciul
  3. **Clientii** (incluzand clientii care pot fi servicii!) care descopera serviciul utilizand registru si acceseaza serviciul direct prin interfetele expuse



# Componentele de baza

La nivel inalt, SOA este format din trei componente de baza:

1. Furnizor de serviciu (Serviciu) - ofera procese in forma de servicii
2. Consumator de serviciu (Consumator) – serviciile oferite de furnizor sunt apelate de conumator
3. Servicii director (premise de Broker) – intre furnizor si consumator



- Serviciul care este disponibil consumatorului este publicat in serviciile director la broker.
- Cosumatorul descopera servicii de la broker
- Daca serviciul este gasit, se ofera legatura la serviciu si se executa logica de procesare

---

# Abstractizarea serviciului

- Metadatele serviciului specifica:
  - Localizarea in retea (adresa de retea a serviciului)
  - Descrierea intr-un format citibil de masina a mesajelor pe care le receptioneaza si optional returneaza.
  - Defineste ce sabloane suporta in termeni de schimb de mesaje
  - O schema pentru datele continute in mesaj este utilizata ca parte a contractului (adica descrierii) stabilite intre solicitantul de serviciu si furnizorul de servicii
  - Operatiile suportate
  - Cerintele suplimentare precum cele de securitate
- Implementarea serviciului poate fi in orice mediu de executie pentru care exista suport pentru servicii

---

# Agentul executabil si stratul de mapare

- Implementarea serviciului este numita si *agent executabil*.
    - Ruleaza intr-un mediu de executie
    - Descrierea serviciului este separata de agentul executabil:
      - O descriere poate avea mai multi agenti executabili asociati cu aceasta
      - Un agent poate suporta descrieri multiple
  - *Stratul de mapare* (numit si strat de transformare):
    - Este adesea implementat prin proxy-uri sau stub-uri.
    - Este responsabil pentru acceptarea mesajului
    - Transforma descrierea datei la un format nativ
    - Expediaza data la un agent executabil
-



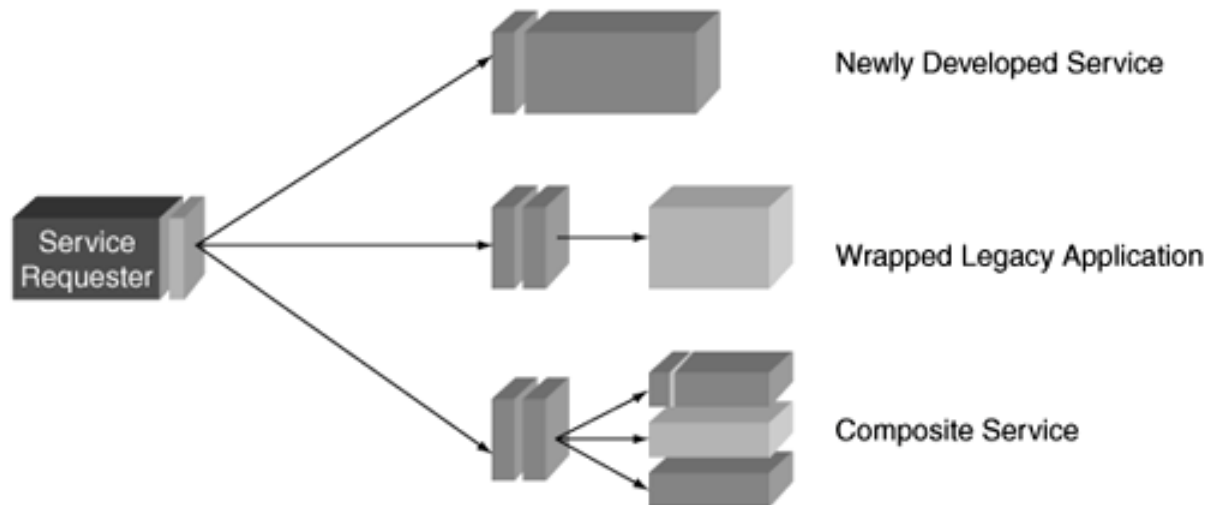
---

# Handler-ul (minuitor) de serviciu

- Serviciile sunt publicate de 'furnizor' si sunt legate de 'consumator' prin 'handler'-ul de serviciu.
- Se comporta ca un agent de colaborare intre furnizor si consumator
- Handler-ul contine logica de realizare
- Cand serviciul este solicitat, se cauta intre numeroase cai de transmitere de mesaje si prin handleri multipli
- Handler-ul in mod uzual ruteaza mesajele la sistemul tinta sau uneori efectueaza o anumita procesare inainte de forwardarea mesajului la sistemul tinta

# Solicitant si furnizor

- Un solicitant (consumator) poate fi de asemenea furnizor & vice versa
  - un agent de executie poate juca ambele roluri
- Unul dintre cele mai mari beneficii ale abstractizarii serviciilor este abilitatea de a accesa usor o varietate de tipuri de servicii, incluzand
  - Servicii noi dezvoltate
  - Aplicatii existente incastrate
  - Aplicatii compuse din alte servicii (noi sau existente).



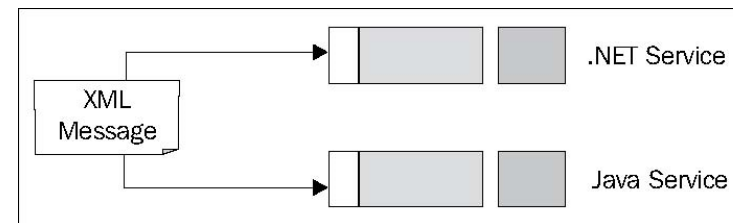
# Obiectivele SOA (1/2)

## 1. Cuplarea slaba:

- ❑ Descompunerea in servicii independente ajuta la scaderea dependentei de un singur proces

## 2. Neutralitatea de platforma:

- ❑ Transmiterea de mesaje bazate de exemplu pe XML creste capacitatea de a atinge neutralitatea de platforma.



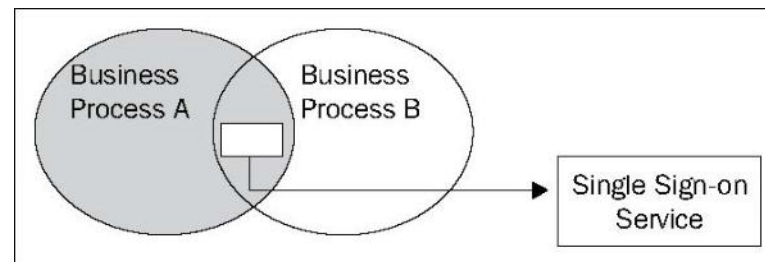
## 3. Standarde:

- ❑ Scurgerea mesajelor este bazata pe standarde acceptate
- ❑ Serviciul trebuie sa depinda numai de descriere

# SOA Objectives (2/2)

## 4. Reutilizarea:

- Logica aplicatiei fiind divizata in unitati logice mici, serviciile pot fi usor reutilizate



## 5. Scalability:

- Cum procesele sunt descompuse in unitati mici, adaugarea de noi logici este usor de realizat
- Noua logica poate fi adaugata ca o unitate extinsa a unui serviciu curent sau poate fi construita ca un serviciu nou

---

# Avantajele SOA

1. Permite **dezvoltarea de aplicatii slab-cuplate** care pot fi distribuite si accesibile in retea
  2. **Integrarea:**
    - O solutie bazate pe SOA este bazata uzual pe inter-operabilitate
    - Cosuturi reduse de integrare prin solutii de compunere
  3. **Agilitate in afaceri:**
    - Beneficii in termeni de *bunuri software* pot fi derivate din abilitatea de reutilizare si simplificare a integrarii
      - Perioada de dezvoltare se scurteaza.
      - Adaptare mai usoara la schimbari => solutia dureaza mai mult in timp
    - Beneficii in termeni de *hardware* derivate din cuplarea slaba ce permite delegarea in domenii multiple
      - Balansarea incarcarii cu procese intr-o organizatie
  4. **Reutilizarea bunurilor**
  5. **ROI (Return-of-Investment) crescut**
-

---

# Tranzitia la SOA

- Principala problema in implementarea SOA este complexitatea solutiilor
    - Descompunerea proceselor curente in servicii mici este o provocare mare in sine
  - Abordari:
    1. **Top-down:**
      - Sunt create cazuri de utilizare si se dau specificatiile pentru crearea serviciilor
      - Unitatile functionale sunt descompuse in procese mici si apoi dezvoltate
    2. **Bottom-up:**
      - Sistemele curente dintr-o organizatie sunt studiate si
      - Procese de business adecvate sunt identificate pentru conversie la servicii
-

---

# SOA vs. OO si CBD

- SOA este o imbunatire a dezvoltarilor de tip **orientat-obiect (OO)** sau **dezvoltare bazate pe componente (CBD)**.
  - Retine elementele cele mai importante de la fiecare
  - Procesele sunt constituite din bucati mici de software interpretate ca si 'componente'.
  - Logica in interiorul componentelor este bazate pe principiile si programarii OO

---

# SOA este implementata utilizand o varietate de tehnologii

- Obiecte distribuite CORBA, J2EE, COM/DCOM.
- Middleware orientat spre mesaje (MOM)  
WebSphere MQ, Tibco Rendezvous.
- Monitoare CICS, IMS, Encinia, Tuxedo.
- Platforme B2B precum ebXML, RosettaNet.
- Servicii Web
- ...



---

# WebSphere MQ

- Numeroase organizatii au implementat SOA utilizand WebSphere MQ
  - Studiu de caz: AXA Financial
    - Companie de servicii de asigurari si financiare
    - Utilizeaza WebSphere MQ ca un strat de trasnmitere de mesaje si integrare pentru a conecta sisteme software existente cu aplicatii de interfatare
    - AXA a inceput dezvoltarea arhitecturii in 1989.
    - Arhitectura de integrare SOA trateaza astazi peste 600 000 de tranzactii pe zi.
- Numai o fractiune mica de sisteme bazate pe WebSphere MQ sunt orientate spre servicii.

---

# CORBA (1/2)

- De ce CORBA pentru SOA?
    - Este un standard deschis.
    - Suporta RMI (adica RPC), transmitere de mesaje asincrona, si comunicare de grup
    - Oferă securitate, servicii de numire, administrarea tranzactiilor si mesagerie de incredere
    - Suporta limbaje de programare multiple
    - Oferă CORBA IDL utilizat ca un limbaj de definire a serviciilor
    - Obiectele pot fi expuse ca servicii Web deoarece OMG a definit o mapare de la CORBA IDL la WSDL
  - Anumite limitari in implementarea SOA:
    - Perceptut ca fiind complex.
    - Cere ca atat solicitantul cat si furnizorul sa utilizeze CORBA.
    - Nu ofera suport explicit pentru XML
    - Nu ofera suport pentru schimb asincron de documente peste Internet
-

---

# CORBA (2/2)

- Numeroase organizatii au utilizat CORBA pentru SOA
    - Studiu de caz: Credit Suisse Group
      - Companie financiara leader mondial cu sediu in Zurich, Elvetia.
      - In 1997, Credit Suisse a pornit implementarea SOA numita Credit Suisse Information Bus (CSIB):
        - scopul CSIB este de asigura interoperabilitatea in timp real, securizata, de incredere si scalabila intre sistemele back-end si o varietate de aplicatii front-end bazate pe diferite platforme (J2EE, C++, SmallTalk, HTML, COM, si Visual Basic).
        - A inlocuit o infrastruktura de integrare bazata pe IBM WebSphere MQ care a devenit prea scumpa si dificil de intretinut
      - SOA de la Credit Suisse suporta mai multe de 100 000 utilizatori, incluzand 600 de servicii de afaceri in productie.
  - Numai un procent mic de sisteme CORBA sunt orientate spre servicii
-

---

# Java si tehnologiile J2EE (1/2)

- Au numeroase avantaje si dezavantaje similare cu CORBA in ceea ce priveste implementari SOA.
- Similaritati cu CORBA:
  - Ambele standarde deschise.
  - Ambele sunt tehnologii pentru obiecte distribuite care ofera suport excelent pentru invocarea de metode la distanta
  - Ambele cer ca solicitantul si furnizorul de serviciu sa utilizeze aceeasi tehnologie (J2EE respectiv CORBA).
  - Ambele ofera
    - Securitate integrata,
    - Servicii de numire (JNDI vs. CORBA Naming Service),
    - Administrarea tranzactiilor (JTA/JTS vs. Object Transaction Service),
    - Transmitere de incredere a mesajelor (JMS vs. CORBA Notification).
  - Ambele J2EE EJB-uri si obiectele CORBA pot fi expuse ca servicii Web.

---

# Java si tehnologiile J2EE (2/2)

- Diferente fata de CORBA:
  - CORBA suporta limbaje de programare multiple.
  - CORBA ofera CORBA IDL ca un limbaj explicit pentru interfete
  - Serviciile Web J2EE comunica nativ utilizand XML si SOAP, pe cand CORBA WSDL mapping comunica utilizand CDL si IIOP.
  - Comunitatea Java a definit o serie de APIuri pentru manipularea XML (ex: JAX-RPC, JAAS, JAX-B, etc).
  - J2EE are o comunitate mai mare si mai robusta de dezvoltatori
  - Implementari J2EE sunt disponibile de la furnizorii cei mai importanti de IT.
- Nu toate sistemele J2EE sunt orientate spre servicii
- Majoritatea aplicatiilor J2EE sunt strans cuplate

---

# Platforme B2B

- Exemple: ebXML si RosettaNet
- Sunt ideale pentru implementare SOA deoarece:
  - Sunt standarde deschise.
  - Sunt cuplate slab.
  - Sunt bazate pe XML.
  - Sunt bazate pe schimb asincron de documente (respectiv documente XML).
  - Oferă mecanisme integrate pentru
    - Inregistrarea serviciului,
    - Securitatea serviciului,
    - Monitorizarea si administrarea serviciilor
    - Administrarea proceselor de afaceri
    - Tranzactii
    - Trasmitere de incredere a mesajelor

---

# SOA utilizand Servicii Web

- Avantajul major in implementarea SOA utilizand serviciile Web:
  - WS sunt are pervasive, simple, si neutre la platforma
- Alte avantaje deriva din faptul ca se bazeaza pe succesul WWW:
  - Utilizarea document markup language precum HTML (sau XML) poate oferi o solutie de interoperabilitate puternica
  - Un protocol de transfer de document precum HTTP poate oferi un mecanism de transfer de date universal si efectiv
  - Pe Web nu conteaza
    - Daca SO este Linux, Windows, OS390, HP NonStop, sau Solaris.
    - Daca serverul Web este Apache sau IIS.
    - Daca logica este codata in Java, C#, COBOL, Perl, sau LISP.
    - Daca browser-ul este Netscape, Internet Explorer, Mozilla, sau W3C Amaya.
- WSurile pot intelege si procesa mesaje in format XML receptionate utilizand un mijloc de transport de comunicare suportat si returnarea nui rezultat daca este necesar este definita

# Platforma de servicii Web

Capabilitatile unui platforme complete de servicii Web:

