

---

# Sisteme distribuite – Tehnologii

## 5. Servicii Web

---

---

# Definitii de la W3C

- WS este un sistem software proiectat pentru a suporta *interactiunea interoperabila masina la masina* peste o retea
  - Sistemul software
    - Este identificat prin URI (accesibil Web printr-un URL)
    - A carui interfete publice si legaturi sunt definite si descrise utilizand artefacte XML (precum Web Services Definition Language - WSDL)
  - Alte sisteme software
    - Pot descoperi definitia sa & si pot interactiona cu serviciul Web intr-o maniera prescrisa de definitia sa
    - Pot sa-l acceseze prin protocoale bazate pe XML precum Simple Object Access Protocol (SOAP) expediat peste protocoale acceptate de Internet, precum HTTP
-

---

# Arhitectura WS (WSA) ca si arhitectura de componente distribuite (DCA)

- Scop DCA: permite programelor dintr-un mediu sa comunice si sa partajeze date/continut cu alte programe din alt mediu.
  - Clasic: programatorii trebuie sa
    - Spuna unui program unde poate gasi un alt program cooperant - aplicatii "strans cuplate"
    - Mentina aceste legaturi programatice pe parcusul vietii aplicatiilor pe care le-au scris
    - Crearea acestor legaturi "hard-wired" este complicata si greoaie
  - WSA – o noua DCA ?
    - Aplicatiile inele pot gasi automat programe cu care sa coopereze
    - Procesul de comunicare program-la-program este numit "cuplat slab."
-

---

## Definitia cf. unit.cercetare Gartner - 2001

- Def: WSurile sunt componente software slab cuplate livrate peste tehnologii standard de Internet
  - WSurile sunt aplicatii care se descriu pe sine si modularare...
    - ... care expun logica afacerii ca servicii peste Internet...
    - ... prin interfete programabile si utilizand protocoale de Internet...
    - ... cu scopul de oferi odalitati pentru a gasi, subcrie si invoca aceste servicii.
  - WSurile pot fi dezvoltate utilizand orice limbaj de programare, orice protocol si orice platforma
-

---

# Evolutie istorica: inceputul

- Spre sfarsitul anilor 1990, Microsoft & si alte companii au gandit un RPC bazat pe XML care poate lucra peste HTTP.
    - Termenul *SOAP* a fost elaborat in 1998.
    - Prima versiune SOAP 1.0 publicat in Dec 1999.
    - Suportul pentru comunitatea comerciala si open-source au condus la 2001, SOAP 1.2.
  - Dupa 2000 Microsoft, IBM, si Sun Microsystems imping WSurile ca noua tehnologie care permite dezvoltatorilor sa creeze obiecte la distanta mai usor
    - Tehnologiile obiectelor la distanta precum COM+ si CORBA, au fost dificil de implementat si au avut costuri mari de mentinere
-

---

## Evolutie istorica: WS ca rezultat a evolutiei Web

1. Initial, Webul a constat din situri care erau doar pagini HTML
  2. Mai tarziu, aplicatiile Web genereaza dinamic aceste pagini HTML
    - Aplicatiile Web sunt limitate de capabilitatile GUI a paginilor HTML
    - Permite interactiunea intre un utilizator si un site Web
  3. WSurile depasesc aceasta limitare pentru ca separa siteul Web sau aplicatia (serviciul) de GUIul HTML.
    - Serviciul este reprezentat in XML si este disponibil via Web ca XML.
    - Permite comunicarea aplicatie-la-aplicatie peste Internet
  - Exemplul unui site Web cu o harta:
    1. Site Web: numai linkuri statice la numeroase orase si localuri
    2. Aplicatie Web: ofera directii de condus, harta customizata, etc
    3. Functionalitate existinsa pentru a oferi un serviciu Web: fiecare companie mentionata poate oferi directii catre locatiile birourilor lor inetegrate cu sisteme de localizare globala
-

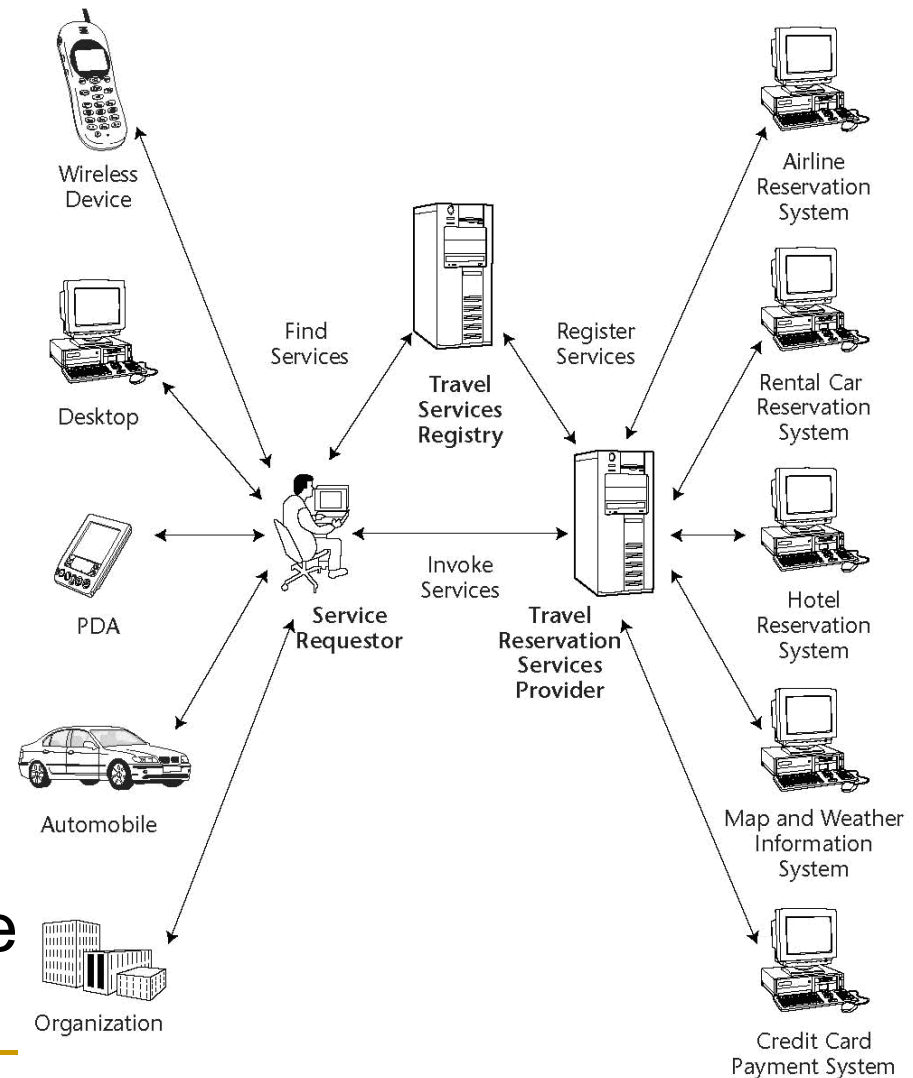
---

## Ratiuni tehnice in favoarea alegerii serviciilor Web in locul aplicatiilor Web

- WSurile pot fi invocate pe baza de mecanisme RPC bazate pe XML peste firewalluri.
  - WSurile ofera o solutie pentru platforme si limbaje multiple bazata pe mesagerie XML
  - WSurile faciliteaza integrarea aplicatiilor utilizand o infrastructura simpla fara a afecta scalabilitatea
  - WSurile permit interoperabilitatea intre aplicatii eterogene
-

# Exemplu: servicii de rezervare pt. calatorii

- Expune aplicatii de afaceri ca servicii Web
- Suporta o varietate de clienti si aplicatii ale acestora
- Aplicatiile de afaceri sunt oferite de diferite organizatii de calatorii care exista in diferite retele si in locatii diferite





---

# Obiect reutilizabil - concept fundamental in WS

Exemplu:

- Un programator scrie o singura data un program gen calculator ca si WS
  - Disponibilitate pentru (incorporare, cuplare sau reutilizare) in:
    - Un program tip spreadsheet,
    - Un program de tranzavtii,
    - Un program de calcul a amortizarii,
    - Sau orice alt program care necesita un calculator.
-

---

# Domeniul WS

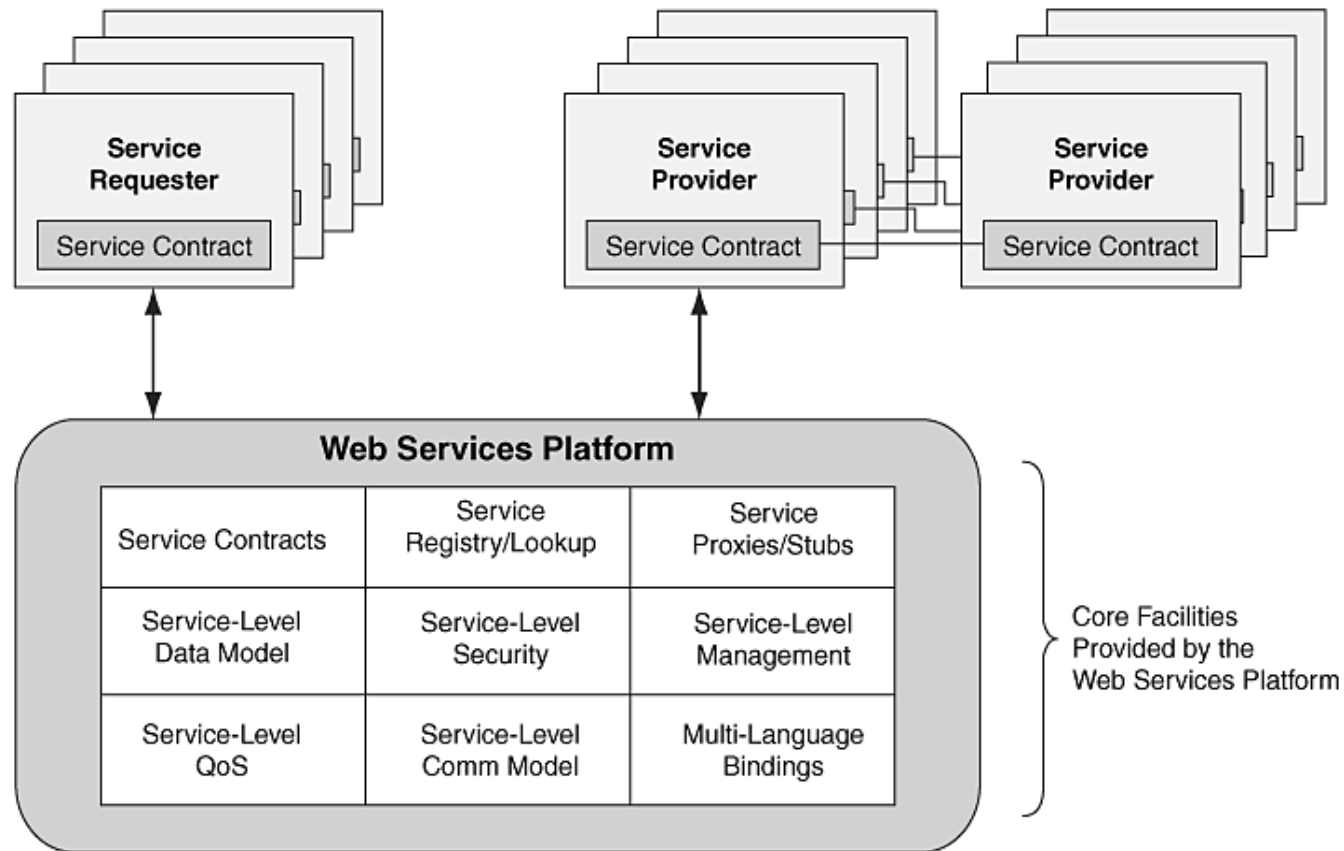
- ! Componente software care sunt accesibile programatic peste protocoale Internet standard
    - “Internetul este SO”!
  - WSul expune o interfata standard care este independenta de platforma si tehnologie.
  - Context: necesitatea crescanda pentru comunicare aplicatii-la-aplocatii si interoperabilitate.
    - WSuri ofera modalitati de comunicare intre aplicatiile software
      - Ruleaza pe platforme diferite
      - Sunt scrise in diferite limbaje de dezvoltare a aplicatiilor
      - Prezinta informatia catre utilizator intr-un context dinamic
-

---

# Beneficiile de baza ale WS

- Interoperabilitate in mediu eterogen
  - Servicii de afaceri peste Web
  - Integrare cu sistemele existente
  - Libertate in alegere
  - Suporta diverse tipuri de clienti
  - Productivitate in programare
-

# Elementele unei platforme WS (1/4)



---

# Elementele unei platforme WS (2/4)

## 1. **Contractul serviciului**

- neambinguu, interfata bine definita a serviciului in WSDL.
- trebuie sa fie citibil de om si masina.

## 2. **Repozitoriu de contracte de servicii**

- Poate include taxonomii in UDDI pentru a categorisi serviciile si pentru cautare
- Trebuie de fie disponibil si replicat.

## 3. **Inregistrare de serviciu si lookup**

- Un serviciu de numre pentru localizarea instantelor de servicii si resurselor
- Daca repozitoriu este utilizat pentru cautare de contracte, acesta este utilizat pentru gasirea instantelor serviciilor in rulare

## 4. **Securitatea la nivel de serviciu**

- Autentificarea solicitantilor de servicii
  - Control al accesului bazat pe roluri
  - Single-sign-on, privat, integritate, ne-repudiere.
-

---

# Elementele unei platforme WS (3/4)

## 5. **Administrarea datelor la nivel de serviciu**

- Utilizare XML Schema pentru validare date, transformare, mapare date intre diferite structuri de mesaje incluzand filtre de date sau agregare

## 6. **Comunicare la nivel de serviciu**

- Suport pentru sabloane multiple de interactiune si stiluri de comunicare utilizand SOAP

## 7. **Protocol multiplu si suport la transport**

- Infrastructura pentru transmitere de mesaje suporta o varietate de protocoale pentru numerosi clienti servere si platforme

## 8. **Calitatea serviciilor**

- Suport pentru ordonarea mesajelor, livrare garantata sau livrare cel mult o data
  - Facilitati pentru administrarea tranzactiilor pentru definirea si executia tranzactiilor incluzand two-phase commit
  - Facilitati speciale incluzand clusterizare, tratarea esecurilor, restart automatic, balansarea incarcarii, lansare de servicii
-

---

# Elementele unei platforme WS (4/4)

## 9. Administrarea la nivel de serviciu

- Suport pentru lansarea, pornirea, oprirea si monitorizarea serviciilor
- Suport pentru servicii de versiuni.
- Suport pentru audit a utilizarii serviciilor
- Suport pentru metrice si plata pentru utilizarea serviciilor
- Monitorizare servicii, status serviciu, reactivitatea serviciilor, deviere de la contract

## 10. Suport pentru limbaje multiple de programare

- Servicii de generare de proxy si schelete de servicii pentru limbajele de programare suportate

## 11. Interfete de programare a serviciilor

- Oferă interfețe pt. ca dezvoltatorii să acceseze facilitățile platformei WS din limbajele lor favorite de programare
-

---

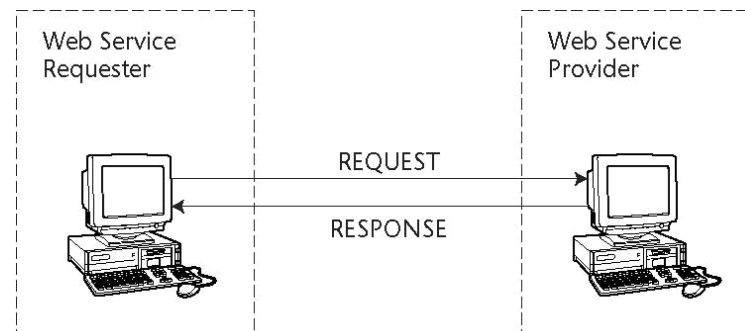
# Tipuri de arhitecturi WS (WSA)

- Diferente cum realizeaza aceste sarcini
  - Cele mai comune WSA:
    1. Remote Procedure Call (RPC)
      - XML-RPC ofera o multime de unelte de baza pentru crearea apelurilor RPC peste platforme multiple, utilizand HTTP ca fundatie
      - WSurile incapsuleaza RPC cu XML ca si impachetare a datelor.
    2. Service-Oriented Architecture (SOA)
      - Combina tehnologiile SOA cu WSurile conducand la:
        - Stiva de protocoale ale WS,
        - O colectie de protocoale de retea care sunt utilizate pentru a defini si implementa cum WSurile interactioneaza intre ele
    3. Representational State Transfer (REST)
-



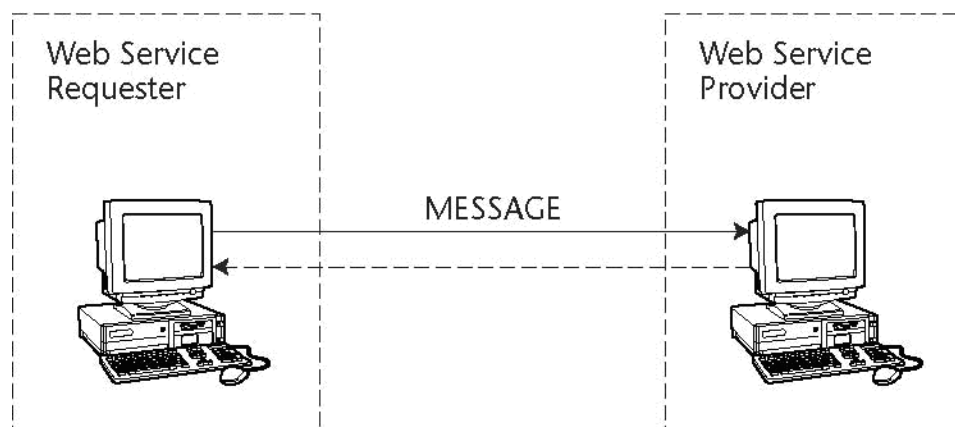
# Modele de comunicare WS: 1 -modelul RPC

- ❑ definește o comunicare sincronă bazată pe cerere/răspuns
- ❑ WS bazate pe RPC sunt strâns cuplate și sunt implementate prin obiecte la distanță în aplicația client
- ❑ Atât furnizorul cât și solicitantul de servicii se înregistrează și descoperă servicii



# Modele de comunicare WS – 2 Bazat pe mesaje

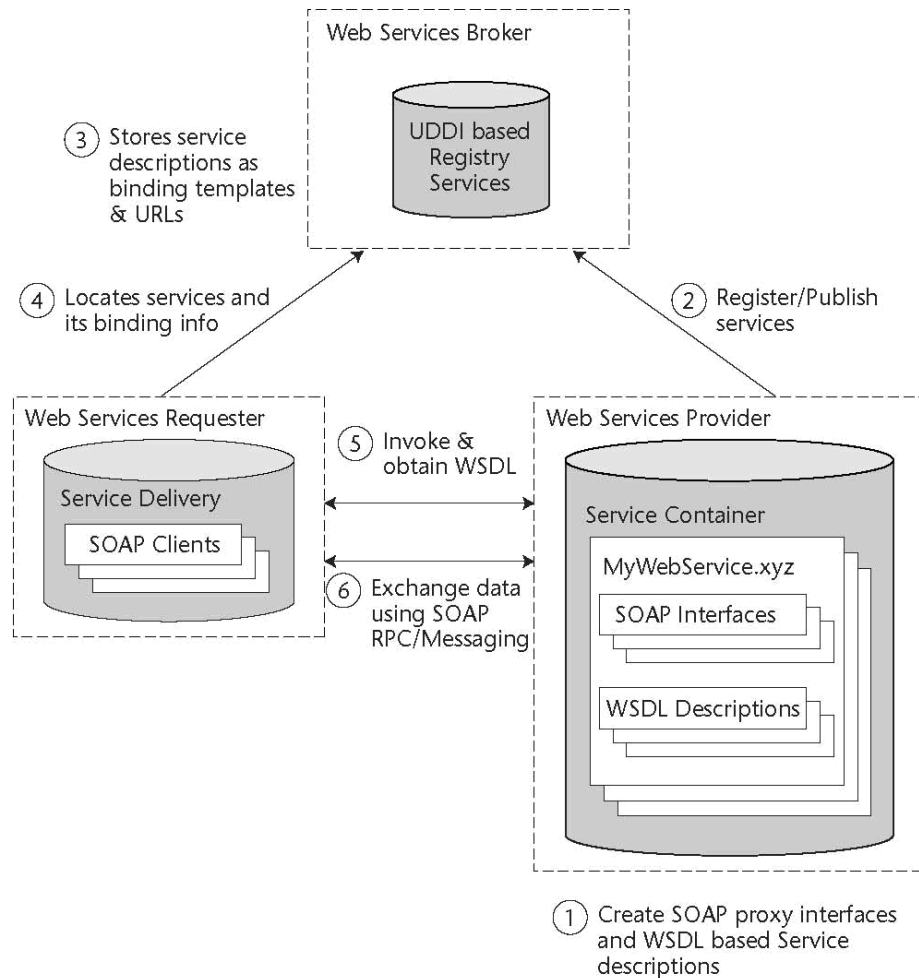
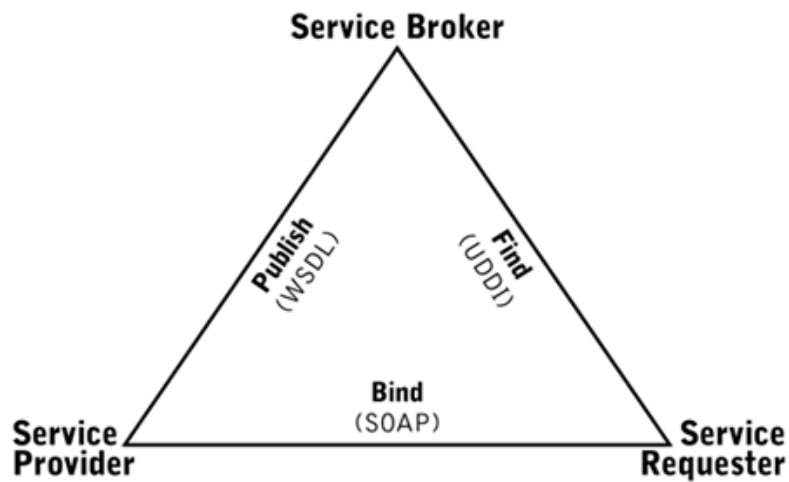
- Comunicare slab cuplata si orientata spre documente
- Solicitantul nu asteapta raspunsul
- Expeditorul trimite un intreg document in locul unor parametrii
- Furnizorul poate sau nu sa returneze un mesaj



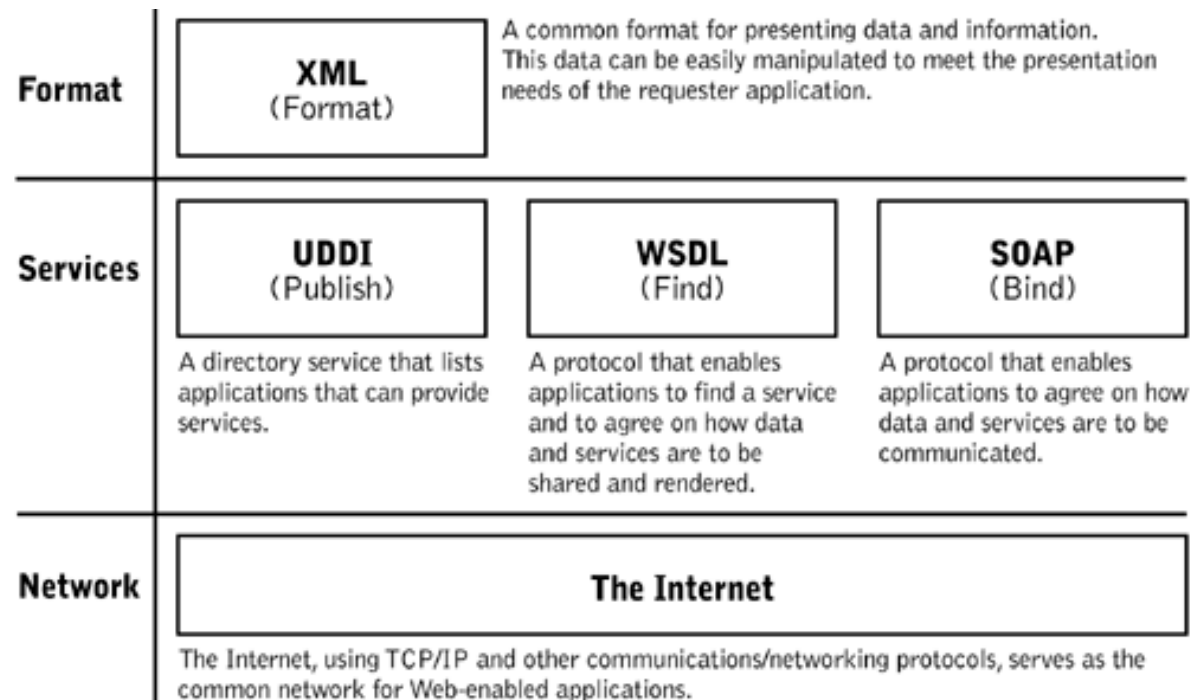
- Adoptarea unui model de comunicarea depinde de furnizorul de servicii
- Versiunile curente de SOAP si ebXML Messaging suporta aceste modele de comunicare

# Cum se lucreaza cu WS

- Publicare, gasire si legare



# Standarde WS (vezi curs urmator!)



## Necesitati:

1. Limbaj comun de markup pentru comunicare
2. Format comun al mesajelor pentru schimbul de informatii
3. Format comun pentru specificarea serviciilor
4. Modalitati comune pentru cautarea de servicii

---

# Stiva protocolurilor WS – 4 nivele de baza

- Transport:
    - HTTP sau HTTPS, SMTP, si FTP.
  - Mesagerie:
    - XML-RPC si SOAP.
  - Descriere
    - Format WSDL format
  - Descoperite
    - Protocol UDDI
-

---

# Legarea serviciilor

- Este diferita la SOA bazata pe WS comparativ cu SOA bazata pe J2EE sau CORBA:
    - J2EE, CORBA: legatura se face via pointeri sau nume,
    - WSurile sunt legate pe baza descoperirii care poate fi dinamica
  - Daca solicitantul serviciului poate intelege
    - WSDLul si
    - Fisierile cu politici asociate furnizate de furnizormesajele SOAP pot fi generate dinamic pentru a executa serviciul furnizorului
-

---

# Implementari WS

- 2 parti majore in industria tehnologiei WS.
    1. Microsoft.
      - A avut un avantaj la start pentru ca a dezvoltat standardul *SOAP* si apoi l-a oferit comunitatii open-source.
      - Bazandu-se pe avansul in *SOAP*, Microsoft a pornit dezvoltarea in limbaje de programare precum *C#* si *Visual Basic.NET* pentru crearea de implementari proprietar.
    2. In jurul Java,
      - Sun Microsystems, creator al Java, lansarea tehnologiilor.
      - Altii precum BEA, Cape Clear Software, IBM, etc.
      - Numeroase biblioteci WS care sunt disponibile liber si usor de obtinut cun sunt cele de la Sun sau grupul Apache
  - Produse tri-partite disponibile care permit WSurilor sa fie integrate cu CORBA, COBOL, C++ si alte sisteme
-

---

# Implementarea Microsoft .NET

- Oferă un număr mare de unelte pentru crearea și utilizarea de WS-uri.
  - Include
    - Generarea automată de WSDL,
    - Unelte de descoperire precum disco (care caută servere care au servicii Web .NET),
    - Testare și descoperire de metode bazată pe browser
    - Creare de WS-uri în limbajele suportate de Microsoft.
  - Dar toate tehnologiile WS ale Microsoft se bazează pe serverul Web *Internet Information Server* (IIS)
    - are probleme de securitate
-



---

# Implementari Java

- Sun si IBM livreaza produse WS.
  - Grupul Apache ofera o biblioteca *SOAP* pentru a accesa WSuri.
  - Avantajele utilizarii WS in Java
    - numerosi vendori implementeaza WSuri in Java
    - Java WSurile lucreaza peste
      - atat Java Server Pages (JSP) cast si servleturi,
      - *Tomcat* care este un server Java ce se integreaza usor cu *Apache*, este oferit liber de la grupul *Apache*.
      - Serverele IBM *Websphere*, BEA *WebLogic*, si Sun *iPlanet* ofera variante comerciale care permite mai multe optiuni cand dezvoltati WSuri.
-

---

# Implementari diferite in Java

- SUN SDK: porneste de la interfata

```
import java.rmi.Remote;
import java.rmi.RemoteException;
public interface AccountIF
    extends Remote {
    public void deposit (int amount)
        throws RemoteException;
    public void withdraw (int
        amount) throws
        RemoteException;
    public int balance () throws
        RemoteException;
}
```

- BEA's WebLogic Server porneste de la o clasa Java

```
public class Account implements
    com.bea.jws.WebService
{
    static final long
        serialVersionUID = 1L;
    /**
     * @common:operation
     */
    public void deposit (int amount);
    {
        ...
    }
}
```

