

---

# Service Oriented Architectures e Web Services

## Corso di **Applicazioni Telematiche**

A.A. 2006-07 – Lezione n.17

Prof. Roberto Canonico

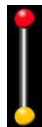
Università degli Studi di Napoli Federico II  
Facoltà di Ingegneria

---

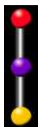
---

## Evoluzione delle applicazioni in rete

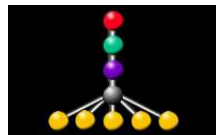
Client-  
Server



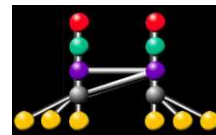
3-Tier



Web  
Application



Web  
Services



## Web Computing

- Programming with distributed components on the Web:
  - Heterogeneous
  - Distributed
  - Multi-language

## Cos'è un Web Service?

- Le evoluzioni delle tecnologie Internet hanno due obiettivi principali:
  - soddisfare i bisogni degli utenti
  - integrare tra loro sistemi informativi tra loro eterogenei e sempre più complessi
- Web Service:
  - Componenti software distribuiti ed accoppiati in modo lasco, che forniscono un servizio ben definito e sono accessibili da programmi mediante protocolli Internet standard
  - *Servizio* inteso non necessariamente come un servizio finale ma come un *componente indipendente* che può essere usato per fornire un servizio finale

---

## Web Service: Definitions

- Component for Web Programming
  - Self-contained, self-describing, modular component that can be published, located, and invoked across the Web
- 

---

## Caratteristiche dei Web Service

- Riutilizzabili
  - Indipendenti da:
    - piattaforma (Unix, Windows)
    - implementazione (VB, C#, Java...)
    - architettura sottostante (.NET, J2EE,...)
  - Accessibili mediante un'interfaccia standard *self-describing*
  - Combinano gli aspetti migliori dello sviluppo basato sulle componenti e gli standard web
-

## Benefici nell'utilizzo dei WS

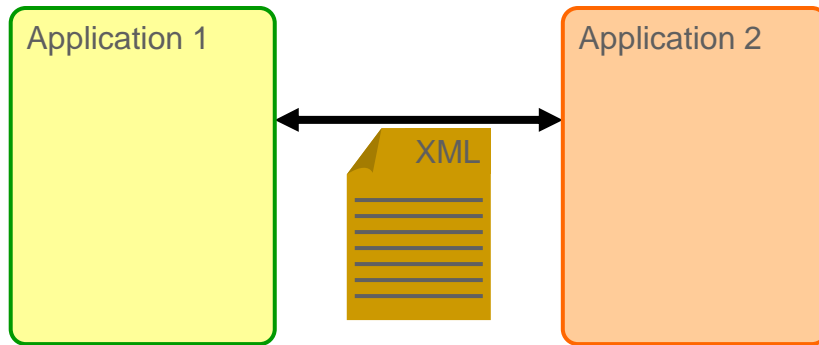
- **Loose Coupling ( accoppiamento indipendente )** : ciascun servizio esiste indipendentemente dagli altri servizi che costituiscono l'applicazione. Questo permette alle singole parti di un'applicazione di essere modificate senza toccare le aree non interessate.
- **Utilizza architetture orientate ai servizi**: i processi business che costituiscono un'applicazione sono separati in componenti indipendenti detti servizi. Questi servizi, interagendo con processi e macchine, riescono a creare una soluzione per i business problem. Le architetture SOA garantiscono una grande flessibilità perché è possibile modificare dinamicamente una parte di esse senza ridefinire l'intero sistema.
- **Facilita l'integrazione**: l'integrazione tra business partners o ambienti diversi è garantita dall'utilizzo di formati standard per lo scambio dei msg. Tali standard sono, inoltre, comprensibili all'uomo e disponibili pubblicamente, e permettono ad uno sviluppatore di vedere esattamente cosa sta accadendo nel sistema.
- **Facilmente accessibile**: WS utilizzano il protocollo di trasporto già esistente e notevolmente utilizzato come http, facendo leva su infrastrutture esistenti e permettendo alle info di essere richieste e ricevute in tempo reale.

## SOA System

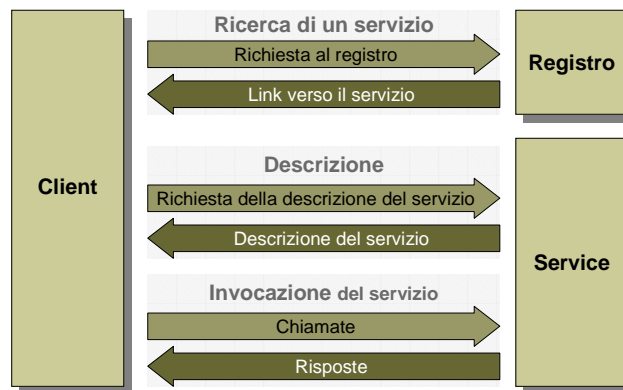
I WS si basano sulla Service Oriented Architecture (SOA). I tre componenti principali sono:

1. **Service Provider**, rende disponibile il servizio e pubblica il contratto che ne descrive l'interfaccia (tramite il broker).
2. **Service Requestor o Consumer**, effettua le queries al service broker e questo cerca il servizio compatibile.
3. **Service Registry o Broker**, da info al consumer su quale servizio utilizzare e dove trovarlo.

## What are Web Services?



## Scenario di richiesta di un Servizio



---

## Is this New?

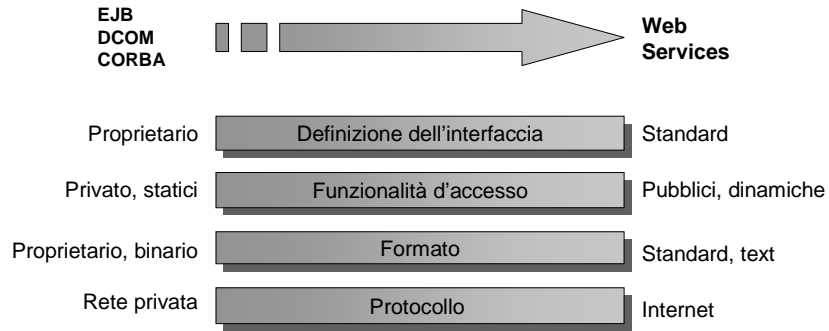
- Sun RPC (1985)
  - CORBA (1992)
  - DCE / RPC (1993)
  - Microsoft COM (1993)
  - Microsoft DCOM (1996)
  - Java RMI (1996)
- 

---

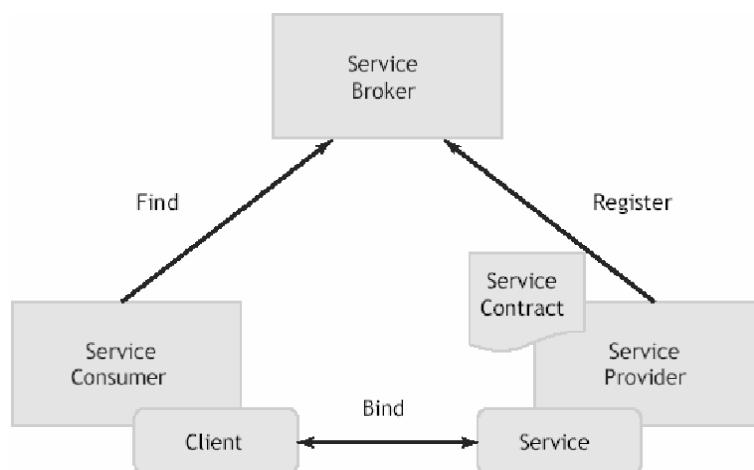
## Is this Different?

- Platform neutral
  - Open Standards
    - Interoperable
  - Based on ubiquitous software
    - XML Parsers
    - HTTP Server
-

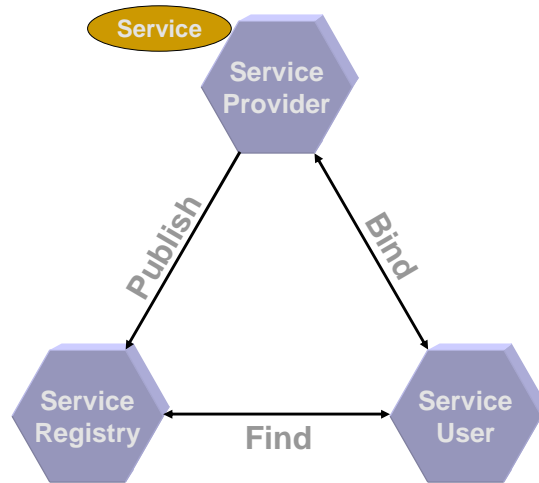
## Evoluzione verso i Web Services



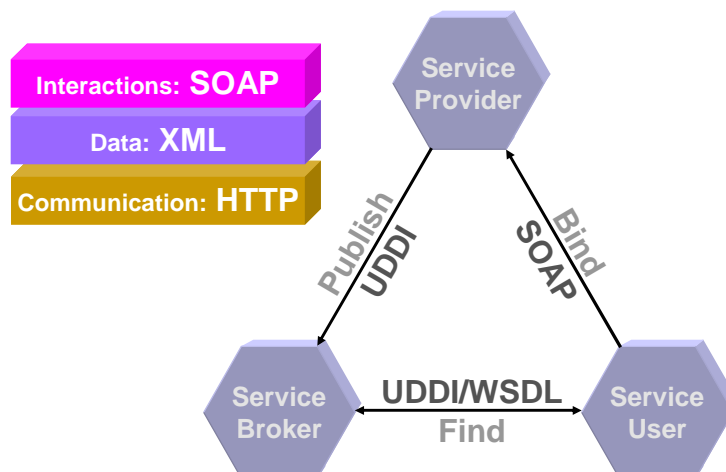
## Web Services roles



## Service-Oriented Architecture



## Web Service Architecture





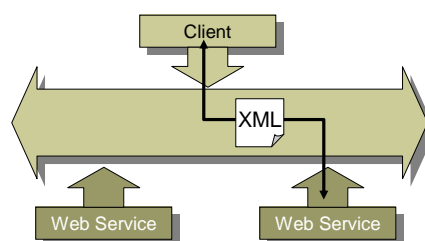
## Web Service Scenario

- Provider builds and defines the service in WSDL
- Provider registers the service in UDDI
- User finds the service by searching UDDI registry
- User application binds to the Web service and invokes its operations via SOAP

## XML e i Web Services

### ■ Un Servizio Web utilizza in modo sistematico XML con i namespace e la specifica XML Schema

- Descrizione dei meccanismi di trasporto
- Le richieste
- I risultati
- Gli errori

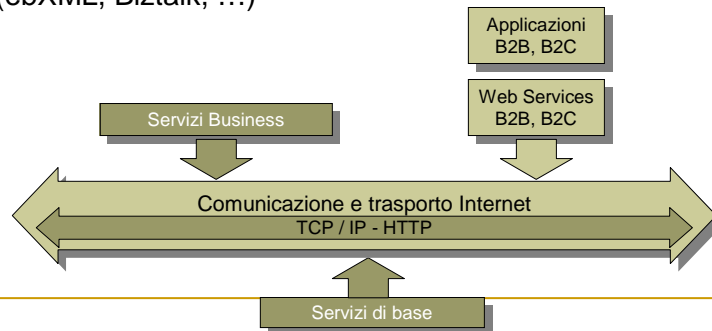


### ■ Svantaggi

- Complessità del documento XML e il costo del suo trattamento
- Tempo di elaborazione per il trattamento dei dati XML
- XML Parser

## Classificazione dei servizi Web

- 3 classi di servizi
  - I servizi di comunicazione e di trasporto (SOAP)
  - I servizi di base: per l'indicizzazione e ricerca dei servizi (UDDI; WDSL)
  - I servizi business: servizi specifici per certi settori d'attività (ebXML, Biztalk, ...)



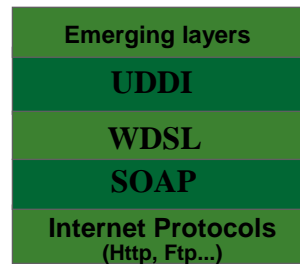
## Stack di tecnologie per i web services

**UDDI (*Universal Description, Discovery and Integration*):** le 'pagine gialle' dei servizi Web

**WSDL (*Web Services Description Language*):** descrizione dei messaggi

**SOAP (*Simple Object Access Protocol*):** protocollo per lo scambio di messaggi

**XML (*eXtensible Markup Language*):** formato per lo scambio dei dati



## Web Services Protocols



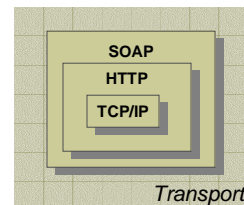
## Infrastructure Elements

- Directories
  - central location to locate Web Services
  - provided by other organizations (e.g. UDDI registry)
- Discovery
  - locating WSDL for a particular Web Service
- Description
  - defines what interactions the Web Service supports
- Wire Formats
  - enable universal communication (e.g. SOAP)

## Servizi di comunicazione e trasporto

### ■ Strato di trasporto:

- XML è usato per descrivere la struttura dei messaggi scambiati tra servizi Web
- Il messaggio è costituito da:
  - L'identificativo
  - Il destinatario
  - Una lista di argomenti eventuali
  - Il nome dell'operazione invocata
  - Una lista di valori di ritorno attesi
  - Altri parametri



## Transport

- HTTP POST is most common
- But other protocols such as
  - FTP
  - SMTP
  - HTTP GET
- And other exotic ones:
  - Jabber
  - BEEP

---

## SOAP

Wire-protocol based on XML and HTTP that consists of:

- an envelope for describing what is in a message and how to process it
  - a set of encoding rules for expressing instances of application-defined data types
  - a convention for representing remote procedure calls and responses
- 

---

## SOAP

- Protocollo leggero di scambio dei dati e strutture di dati tra i nodi della rete
  - nato con l'obiettivo di creare un meccanismo di RPC in ambiente distribuito, in modo trasparente al sistema operativo o linguaggio di programmazione
  - Il payload di un messaggio SOAP è in formato XML
  - SOAP lavora su protocolli applicativi esistenti (ad es. HTTP e SMTP)
-

# SOAP

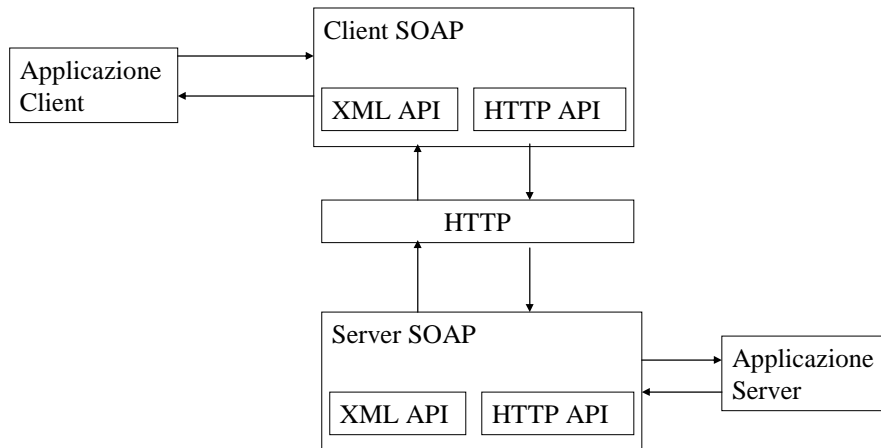
- La specifica SOAP definisce
  - La busta SOAP (struttura del messaggio)
  - Le regole di codifica dei tipi di dati (come rappresentare un intero, una tabella, un grafo, ecc.)
  - Le convenzioni per la rappresentazione della chiamate a procedure remote e per l'invio dei risultati o di messaggi d'errore.
  - Le regole per il trasporto dei messaggi SOAP sul protocollo HTTP
  - Gli URL per l'indirizzamento ed il routing
  - Dialogo tra mittente e destinatario

# Messaggio SOAP

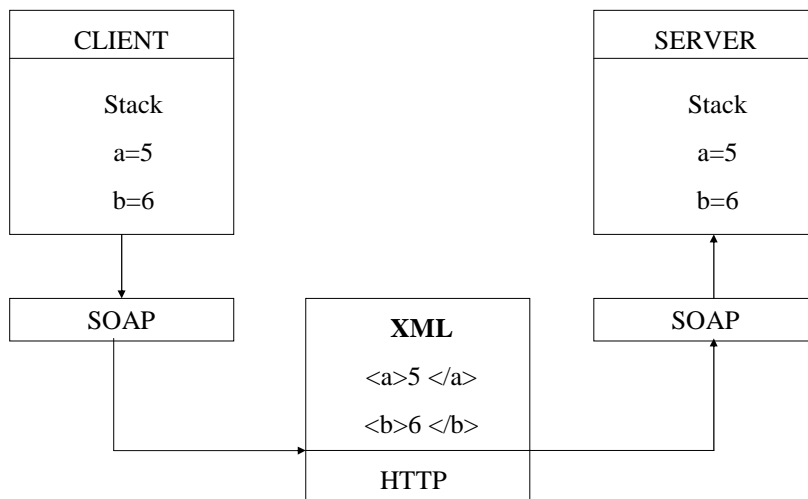
- La struttura dei messaggi SOAP
  - La busta SOAP
  - L'header SOAP

```
Busta SOAP      <SOAP:Envelope
                    xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
Header SOAP    <SOAP:Header>
                    <example:header xmlns:example="www.y.com"></example:header>
                    </SOAP:Header>
Corpo del      <SOAP:Body>
messaggio SOAP <example:body xmlns:example="www.y.com"></example:body>
                    </SOAP:Body>
                    </SOAP:Envelope>
```

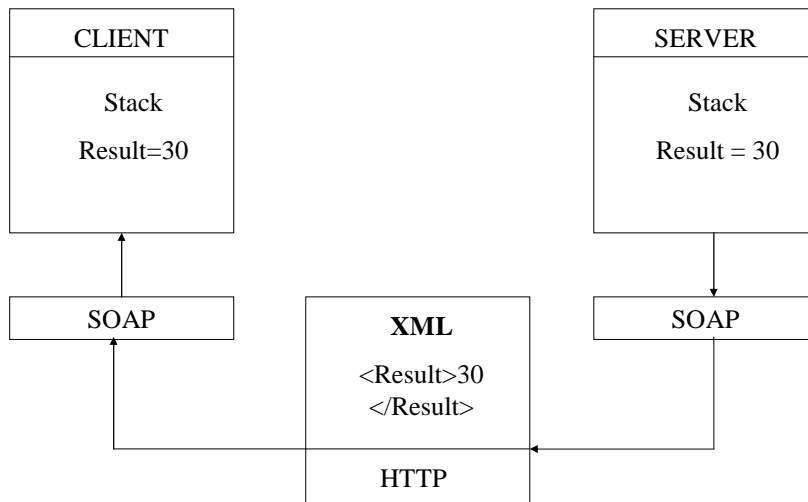
## Schema della comunicazione SOAP



## Chiamata al metodo remoto



## Invio del messaggio di risposta



## Sample SOAP request

```
POST /CurrencyServer/CurrencyExchange.jws HTTP/1.1
Host: theseus
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: http://ws.x.com/webservices/Euro
```

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope
  xmlns:xsi=http://www.w3.org/2001/XMLSchema-instance
  xmlns:xsd=http://www.w3.org/2001/XMLSchema
  xmlns:soap=http://schemas.xmlsoap.org/soap/envelope>
  <soap:Body>
    <Euro xmlns=http://ws.x.com/webservices>
      <currency>string</currency>
    </Euro>
  </soap:Body>
</soap:Envelope>
```



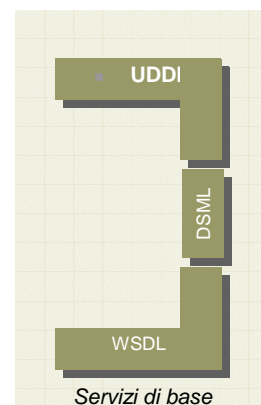
## Sample SOAP reply

```
HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: length
<?xml version="1.0" encoding="utf-8"?>

<soap:Envelope
  xmlns:xsi=http://www.w3.org/2001/XMLSchema-instance
  xmlns:xsd=http://www.w3.org/2001/XMLSchema
  xmlns:soap=http://schemas.xmlsoap.org/soap/envelope>
  <soap:Body>
    <EuroResponse xmlns=http://ws.x.com/webservices>
      <EuroResult>double</EuroResult>
    </EuroResponse>
  </soap:Body>
</soap:Envelope>
```

## Servizi di base

- Gestiscono gli scambi tra servizi ed applicazioni
- XML è usato per:
  - Descrivere ed identificare i servizi
  - Rendere sicuro ed autenticare l'accesso ai servizi
  - Esprimere le relazioni tra i servizi
  - Classificare e ricercare i servizi negli annuari
- Gli standard:
  - WSDL (Web Service Description Language) per definire un servizio Web
  - UDDI (Universal Description, Discovery and Integration) per creare ed interrogare dei registri
  - DSML (Directory Process Modelling Language) per creare dei registri di risorse interne all'impresa in XML (compatibile con LDAP)

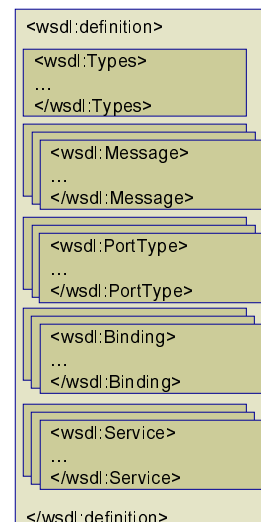


## WSDL

- WSDL: documento XML che contiene la descrizione dell'interazione client-server
- Standard W3C basato su XML Schema
- Obiettivo: fornire
  - L'elenco dei servizi offerti da un servizio web
  - I dati necessari al servizio
  - I dati di risposta
  - Le modalità di invocazione del servizio
- Il linguaggio descrive il servizio a due livelli diversi
  - Descrizione astratta (abstract view)
    - Descrive un servizio sulla base delle operazioni che questo mette a disposizione
    - Il servizio è identificato da un gruppo di *portType*, che a loro volta sono costituite da un insieme di *operation*
  - Descrizione concreta (concret view)
    - Specializzazione, binding ports, services

## WSDL

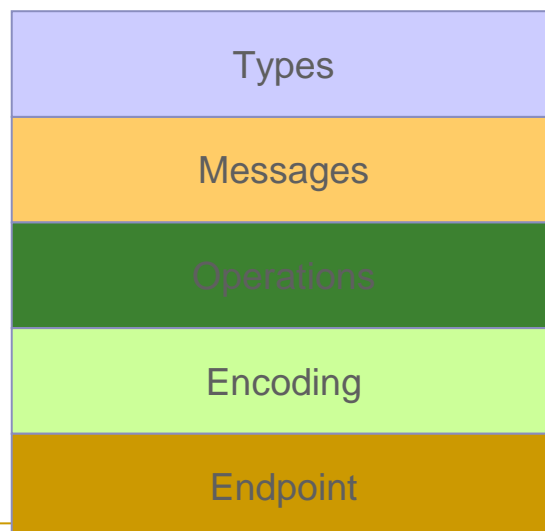
- WSDL (Web Service Description Language) definisce le seguenti entità
  - Tipi: definizione dei tipi di dati del Web Service (uso di XMLSchema)
  - Messaggio: definizione dei messaggi che fanno riferimento ai tipi
  - Tipo di porta: insieme di operazioni (action) implementate da un Web Service che fanno riferimento ai messaggi. Una porta è un punto di terminazione identificato in maniera unica da un'identificativo (e.g. URL) e un binding.
  - Binding WSDL/SOAP: un protocollo concreto d'accesso a una porta e un formato dei messaggi e dei dati per questa porta
  - Servizio: una collezione di porte (le istanze delle porte WSDL)



## WSDL Structure

Types	Data type definitions
Message	Signature of request and reply for each method ( $\approx$ IDL)
Port Type	$\langle$ service, protocol $\rangle \Leftrightarrow$ operations
Operation	method $\Leftrightarrow$ messages
Binding	Protocol and data-format specification
Service	{ Port $\Leftrightarrow$ binding }
Port	Address ( $\approx$ URL)

## Description – WSDL



## Types

```
<types>  
<schema targetNamespace=" IMessageService.xsd"  
  xmlns=".../XMLSchema" xmlns:SOAPENC=".../soap/encoding"/>  
</types>
```

## Messages

```
<message name="purchase">  
  <part name="item" type="xsd:string"/>  
  <part name="quantity" type="xsd:integer"/>  
</message>
```

## Operations

```
<operation name="setMessage">
  <input name="setMessageRequest"
    message="tns:setMessageRequest"/>
  <output name="setMessageResponse"
    message="tns:setMessageResponse"/>
</operation>
```

## Encoding

```
<soap:operation soapAction="" style="rpc"/>
  <input name="setMessage0Request">
    <soap:body use="encoded"
      namespace="MessageService"
      encodingStyle="../../../soap/encoding"/>
  </input>
```

## Endpoint

```
<service name="MessageService">
  <port name="MessageServicePort"
        binding="tns:MessageServiceBinding">
    <soap:address location="http://localhost:8080/setMessage/" />
  </port>
</service>
```

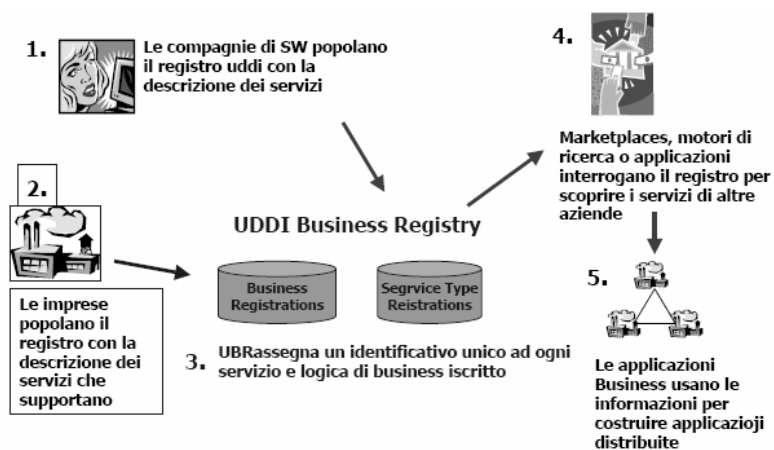
## WSDL example

```
<message name="RateSoapIn">
  <part name="parameters" element="s0:Rate" /></message>
<message name="RateSoapOut">
  <part name="parameters" element="s0:RateResponse" /></message>
<message name="EuroSoapIn">
  <part name="parameters" element="s0:Euro" /></message>
<message name="EuroSoapOut">
  <part name="parameters" element="s0:EuroResponse" /></message>
<message name="RateHttpGetIn">
  <part name="from" type="s:string" />
  <part name="to" type="s:string" /></message>
<message name="RateHttpGetOut">
  <part name="Body" element="s0:double" /></message>
<message name="EuroHttpGetIn">
  <part name="currency" type="s:string" /></message>
<message name="EuroHttpGetOut">
  <part name="Body" element="s0:double" /></message>
<message name="RateHttpPostIn"><part name="from" type="s:string" />
  <part name="to" type="s:string" /></message>
<message name="RateHttpPostOut">
  <part name="Body" element="s0:double" /></message>
<message name="EuroHttpPostIn">
  <part name="currency" type="s:string" /></message>
<message name="EuroHttpPostOut">
  <part name="Body" element="s0:double" /></message>
...
```

## UDDI

- UDDI (*Universal Description, Discovery and Integration*): insieme di specifiche che definiscono un modo di pubblicare e cercare servizi attraverso una repository centralizzata di servizi
- L'obiettivo è fornire un elenco mondiale di servizi che permetta di ricercare i Servizi Web sullo stesso principio delle pagine gialle
  - Definizione delle informazioni da fornire per ciascun servizio e tipo di codifica
  - query e update API che descrivono come si può accedere e aggiornare le informazioni
- La riuscita di UDDI richiede che i diversi fornitori di Servizi Web si accordino sulla definizione di criteri comuni e di determinate categorie di servizi
  - Operatori: Microsoft, IBM, SAP et HP
  - Problematico

## Standard UDDI



## Standard UDDI

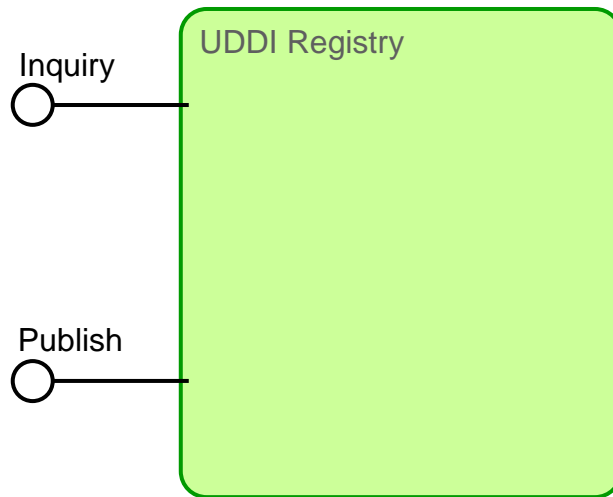
- Le informazioni sono definite con XML Schema
- Quattro tipi fondamentali di informazioni
  - Informazioni di business (chi fa cosa)
  - Informazioni sul servizio (cosa fa per gruppo o categorie)
  - Informazioni sul binding (come invocare)
  - Informazioni sulla descrizione dei servizi (cosa invocare)

## Discovery – UDDI

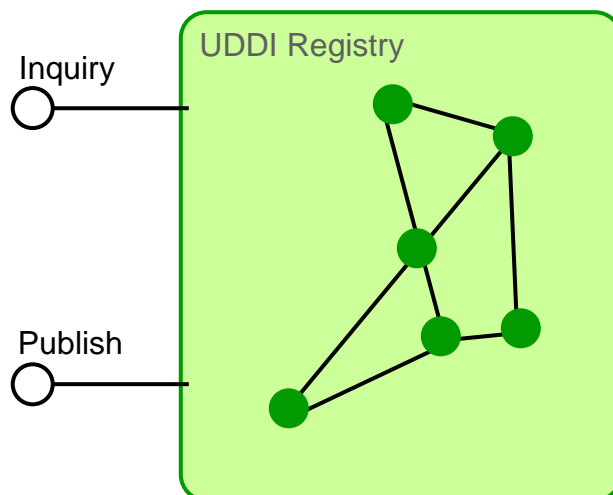
- Demonstration:  
<https://uddi.ibm.com/ubr/registry.html>



## Discovery – UDDI



## Discovery – UDDI



# UDDI

Le specifiche UDDI definiscono le tre parti costituenti il registry

Pagine bianche

Identità del fornitore, indirizzo fisico ed elettronico, qualificazioni che fanno riferimento a tassonomie industriali standard (e.g. DUNS)

Pagine gialle

La descrizione dei servizi offerti

Pagine verdi

Informazioni sui modelli di accesso al servizio e i differenti modelli di dati sottostanti

## Discovery – UDDI

- **Tipo di servizio:** si definisce il servizio e si assegna a questo un identificatore unico chiamato tModel. Un tModel punta a una specifica che definisce una risorsa. Oltre al servizio si definisce un'interfaccia astratta.

### *Pagine bianche*

- **Service providers** registra i businesses e tutti i servizi che offrono. Tramite il costrutto Template fornisce info sul binding e sul punto di accesso.

### *Pagine verdi*

- **Categorizzazione:** si usa una varietà di categorie per classificare le entità in base alla localizzazione geografica, al codice prodotto ecc.

### *Pagine gialle*

## Discovery – UDDI

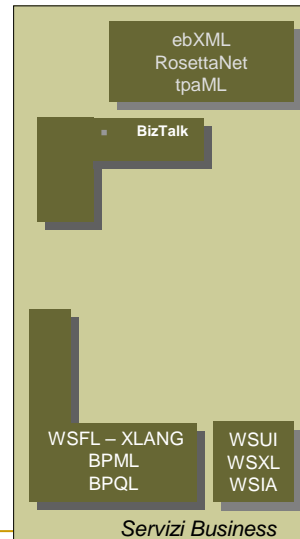
- **Ricerca.** I service consumer possono ricercare il servizio effettuando queries al registro Uddi, questa ricerca può avvenire tramite tipo di servizio o service provides. Nel caso di una intranet, il broker può cercare prima i service provider interni e poi altri broker su Internet se tali servizi non esistono localmente.
- **Binding :** il collegamento tra client e servizio può avvenire sia in fase di compilazione che di runtime.
  - **Statico:** La ricerca di un service viene effettuata una volta sola e se ne memorizza il risultato. Quindi la localizzazione dei servizi si conosce prima di iniziare l'esecuzione del programma e si tratta di indirizzi assoluti.
  - **Dinamico:** la ricerca viene effettuata in fase di runtime e di solito perché è stata modificata la posizione del servizio.
- **Scoperta dinamica.** Siccome Uddi è esso stesso un WS, un'applicazione può fare richieste al servizio, trovare dinamicamente il servizio, localizzare il suo punto di accesso, recuperare il WSDL e collegarsi ad esso tutto in runtime.

## Servizi Business

- Si tratta generalmente di funzioni legate al commercio elettronico
- Riproduzione in un mondo virtuale delle transazioni commerciali del mondo reale (transazioni, contratti, fatturazione, pagamenti, ecc.)
- I servizi Web Business mettono a disposizione degli sviluppatori un insieme di specifiche che facilitano lo sviluppo di applicazioni Web per settori applicativi specifici
- Per quanto riguarda gli aspetti tecnici, la specifica di alcuni Servizi Business è iniziata prima delle attività di standardizzazione del W3C

## Servizi Business

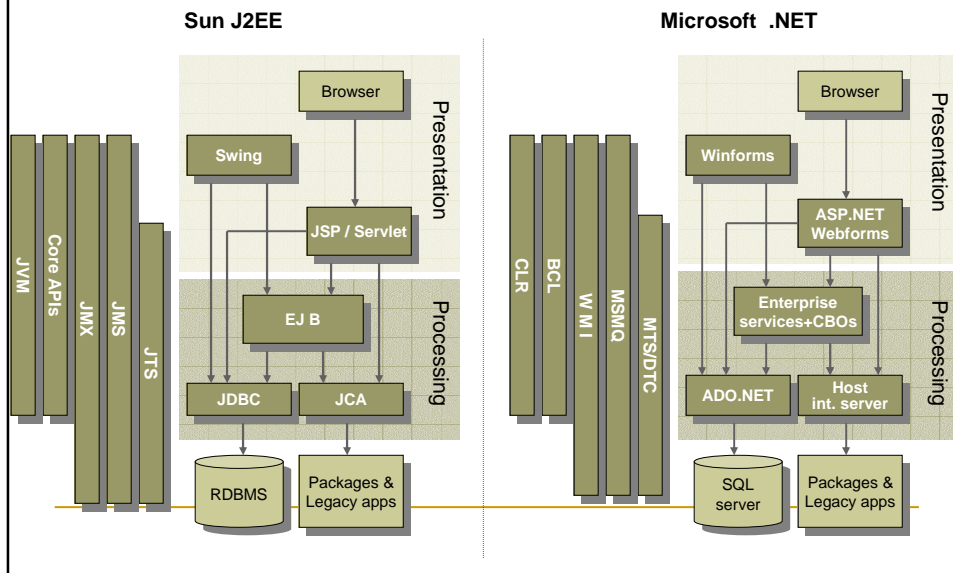
- **ebXML e RosettaNet**: per formalizzare un'infrastruttura completa per l'e-commerce
- **BizTalk** di Microsoft: formalizzazione dello scambio elettronico dei documenti professionali (fatture, ordini, ecc.) tra applicazioni Web distribuite
- **WSFL** (Web Services Flow Language) di IBM, **XLANG** di Microsoft e **WSCL** (Web Services Conversation Language) di HP per la specificazione della composizione di un'applicazione Web per l'esecuzione di processi business.
- **BPML** (Business Process Modeling Language) e **BPQL** del consorzio BPMI: una parte delle specifiche copre la sincronizzazione dei processi business in diverse aziende (ex: gestione delle relazioni con i clienti, logistica, ...)
- **WSUI** (Web Service User Interface) di Epicentric, **WSXL** (Web Services Experience Language) di IBM e **WSIA** (Web Services Interactive Applications) di OASIS. Gestione dell'accesso ai servizi Web



## Piattaforme di sviluppo e esercizio

- Web Service su piattaforma leggera
  - Server HTTP e Parser XML
  - Esempi: Apache SOAP, Apache Axis, SOAP::Lite (Perl), PHPSOAP (PHP), WhiteMesa SOAP (C++), SOAP for ADA, Smalltalk Web Services
- Web Service su Application Server
  - Valori aggiunti: gestione dell'accesso concorrente, gestione delle transazioni, sicurezza e autenticazione, infrastrutture, tool di sviluppo
  - Classificazione dei fornitori:
    - Basi di dati: DBMS tradizionale + infrastruttura XML per integrare l'architettura UDDI/WSDL/SOAP: Oracle, IBM, Sybase
    - Middleware: BEA, Viti, IBM, Progress
    - Sistemi operativi: SUN, IBM, Microsoft

## Piattaforme di sviluppo e esercizio



## Missing Pieces

- Security
  - Single Sign-on, credentials
- Transactions
- Quality of service
  - Timeliness guarantees
- Asynchronous operations
  - Co-ordination, workflow

## Real Examples

- Amazon Web Services API
- Google Web API
- HP & IBM online stores



## Conclusioni

- Web Service
  - Nuova forma di componenti logici sul piano tecnico e di economia di sviluppo. Si tratta di un'unità funzionale con interfaccia pubblica e implementazione privata. Meccanismi di assemblaggio di applicazioni sulla base delle interfacce dei servizi.
  - Caratteristiche di base: apertura, specifica basata su strati, meccanismi di estensione, riuso, autodescrizione, ricerca e classificazione dei servizi
  - Soap è uno standard stabile, WSDL e UDDI sono stati specificati, ma non ancora stabili
  - Gli altri standard sono in via di definizione