

Web service

A cura di

Azzurra Ragone

RPC (Remote Procedure Call)

“A type of **protocol** that allows a program on one computer to execute a program on a server computer. Using RPC, a system developer need not develop specific procedures for the server. The **client** program sends a message to the server with appropriate **arguments** and the server returns a message containing the **results** of the program executed. Two newer object-oriented methods for programs to communicate with each other, CORBA and DCOM, provide the same types of capabilities as traditional RPCs” **Webopedia**.

Web service

Un Web Service è un servizio disponibile in rete ad uso di altri programmi, che può essere pubblicato, localizzato e invocato attraverso il Web. A sua volta, un servizio espone una o più funzionalità. Una funzionalità offerta da un Web Service può andare da una semplice richiesta/risposta elementare a complesse elaborazioni.

I Web Service sono debolmente accoppiati e comunicano direttamente con altri Web Service attraverso Internet usando tecnologia basata esclusivamente su standard (XML e HTTP) .

Web service: What is this?

“A Web service is an *application* that exposes functionalities accessible via the Internet, using **standard Web protocol**”
(**Casati F.**, ECAI 2002)

“A Web service is a *software system* designed to support **interoperable machine-to-machine interaction** over a network. It has an **interface** described in a **machine-processable format** (specifically WSDL). Other systems *interact* with the Web service in a manner prescribed by its description using **SOAP-messages**, typically conveyed using **HTTP** with an **XML** serialization in conjunction with other *Web-related standards*.
(**W3C**, <http://www.w3.org/TR/ws-gloss/>)

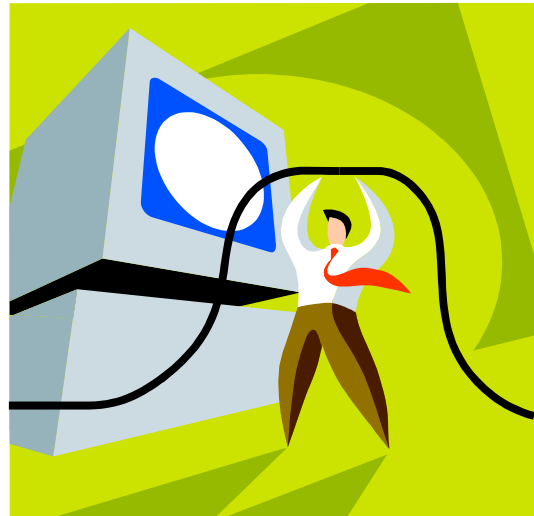
Web service : What is this?(2)

“The term Web services describes a **standardized** way of *integrating Web-based applications* using the XML, SOAP, WSDL and UDDI open standards over an Internet protocol backbone. **XML** is used to tag the data, **SOAP** is used to transfer the data, **WSDL** is used for describing the services available and **UDDI** is used for listing what services are available.”

Webopedia (<http://www.webopedia.com>)

Web service: ricapitolando

Se un'applicazione ha un'*interfaccia well-defined* (ad es. descritta in WSDL), e interagisce con un client scambiando messaggi XML incapsulati in pacchetti SOAP allora è un Web service!



Cosa non è un Web service!

• Una semplice risorsa web richiamabile tramite un URL (ad es. un semplice CGI script)

• Un programma accessibile sul web tramite delle API e contenente delle informazioni descrittive in una qualche directory

Web service (1)

Un WS deve essere “*defined, described, and discovered*”, in modo che possa essere invocato da un client.

I WS sono componenti che *devono* poter essere integrati in complesse *applicazioni distribuite*.

L'uso di tecnologie standard riducendo l'eterogeneità, facilita l'integrazione delle applicazioni

Web service (2)

È una tecnologia che consente alle applicazioni di comunicare vicendevolmente in maniera che sia indipendente:

- dalla **piattaforma** (hardware e software)
- dal **linguaggio di programmazione**



Web service e XML

Un WS è un'interfaccia software che descrive un insieme di **operazioni**, accessibili tramite la rete attraverso messaggi standard di tipo XML

I WS usano protocolli XML-based per descrivere le operazioni da eseguire e i dati da scambiare con altri WS.



Web service = e-service?

Ciascuna compagnia può esporre le funzionalità implementate tramite WS come *servizi*.

Ma un Web service non è un servizio!

Service: procedure, method, or object with a stable, published interface that can be invoked by a client.

(Alonso et al.,2003)

Sia la *richiesta* che *l'esecuzione* del servizio impone che vi sia un programma che “chiama” un altro programma!

Web service e SOA (1)

I WS sembrano essere la *tecnologia candidata* per la realizzazione delle Service-Oriented Architecture (SOA)

SOA is an *architectural style* whose goal is to achieve loose coupling among interacting software agents. A **service** is a **unit of work** done by a *service provider* to achieve desired end results for a *service consumer*. Both provider and consumer are roles played by software agents on behalf of their owners.

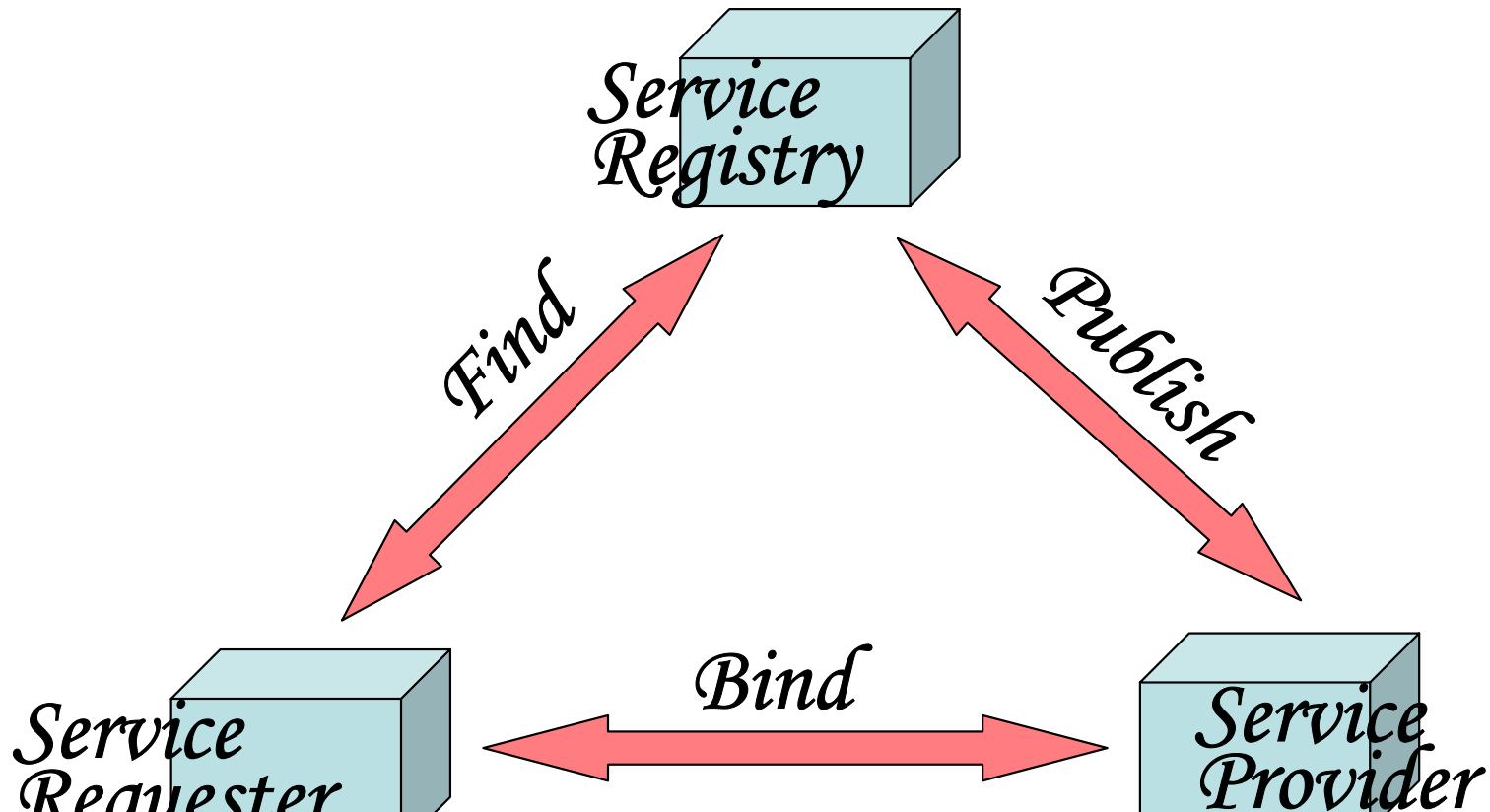
(<http://webservices.xml.com/>)

Web service e SOA (2)

“An *application architecture* in which all functions, or services, are defined using a description language and have invokable interfaces that are called to perform *business processes*. Each *interaction* is independent of each and every other interaction and the interconnect protocols of the communicating devices (i.e., the infrastructure components that determine the communication system do not affect the interfaces). Because interfaces are platform-independent, a client from any device using any operating system in any language can use the service.” Webopedia (<http://www.webopedia.com>)

Web service e SOA (3)

Le SOA sono ambienti in cui i servizi devono poter “*ricercarsi*” e “*scoprirs*” dinamicamente l’un l’altro, per realizzare “una qualche applicazione” e ricombinarsi in vario modo.



Web service: scopo

Integrare applicazioni software che utilizzano diversi:

- sistemi operativi (Windows, Unix)
- linguaggi di programmazione (C++, Java, Python, Perl, Basic, ecc.)
- piattaforme hardware

Business process e Web service

Un WS può essere visto come una singola *unit work* che gestisce una specifica task

Ciascuna *task* può essere combinata in modo da gestire particolari *business operational task*

I WS colmano il **gap** esistente tra la conoscenza tecnica e la conoscenza gestionale in un'impresa.



Riusabilità dei Web service

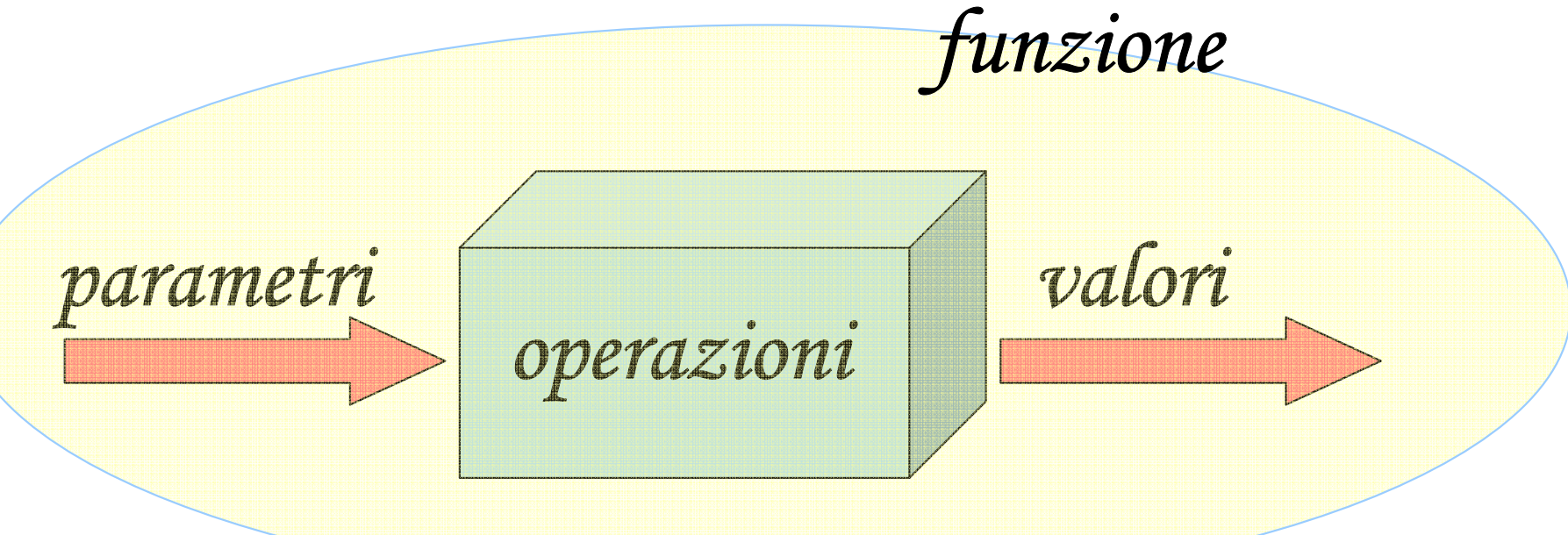
Specificare in maniera *well-defined* le interfacce e le task permette di riutilizzare in diverse applicazioni tali task.

I singoli moduli che rappresentano le task possono essere così facilmente riusati e riaggregati in diversi modi.

Web services and objects (1)

Prima di pensare a “*come combinare*” più servizi in un’unica applicazione dobbiamo occuparci di “*come costruire*” tali servizi.

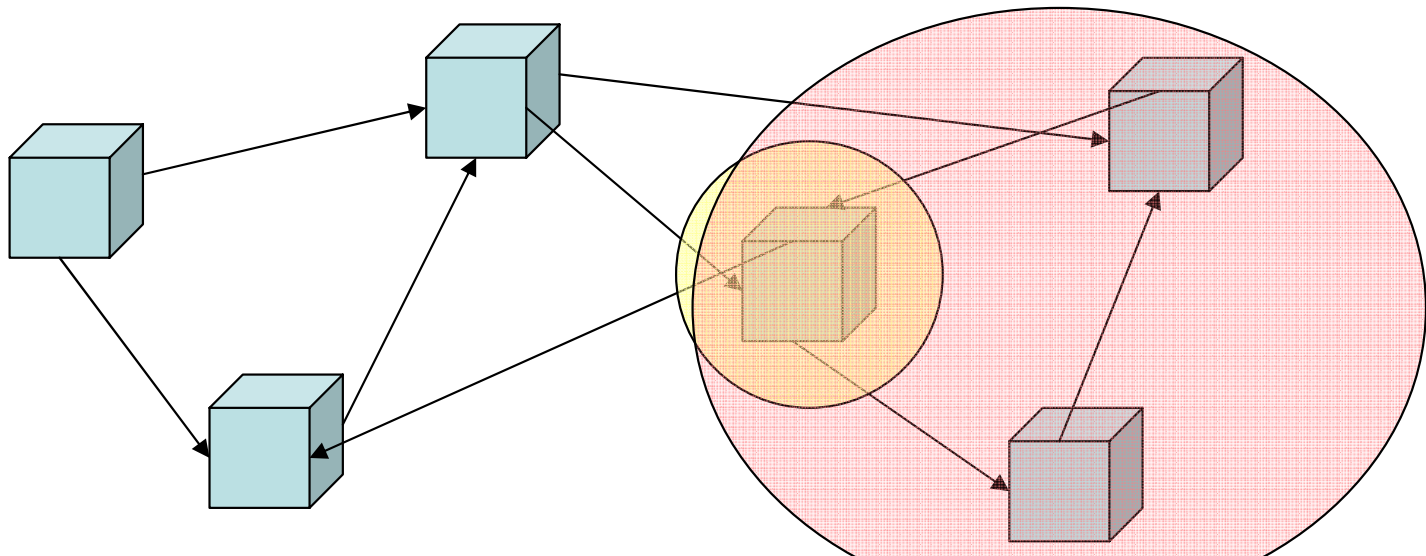
Oltre l’idea di funzione:



Web services and objects (2)

Il paradigma dei WS si basa invece sul concetto di **oggetto**

Ogni oggetto è una collezione di operazioni con una *interfaccia universale*, che consente ad altri oggetti di invocarlo anche se sono scritti in altri linguaggi di programmazione e “*girano*” su altri sistemi operativi.



Caratteristiche distintive dei WS

In sintesi i WS consentono:

interazioni tra servizi indipendentemente dalla piattaforma e dal linguaggio.

inglobare le funzioni in task (**riusabilità**)

elevata **flessibilità** (*loosely-coupled*)

adattabilità ai cambiamenti e ai bisogni dei consumatori

inglobare le applicazioni software esistenti

introdurre funzioni di gestione (**affidabilità, sicurezza**)

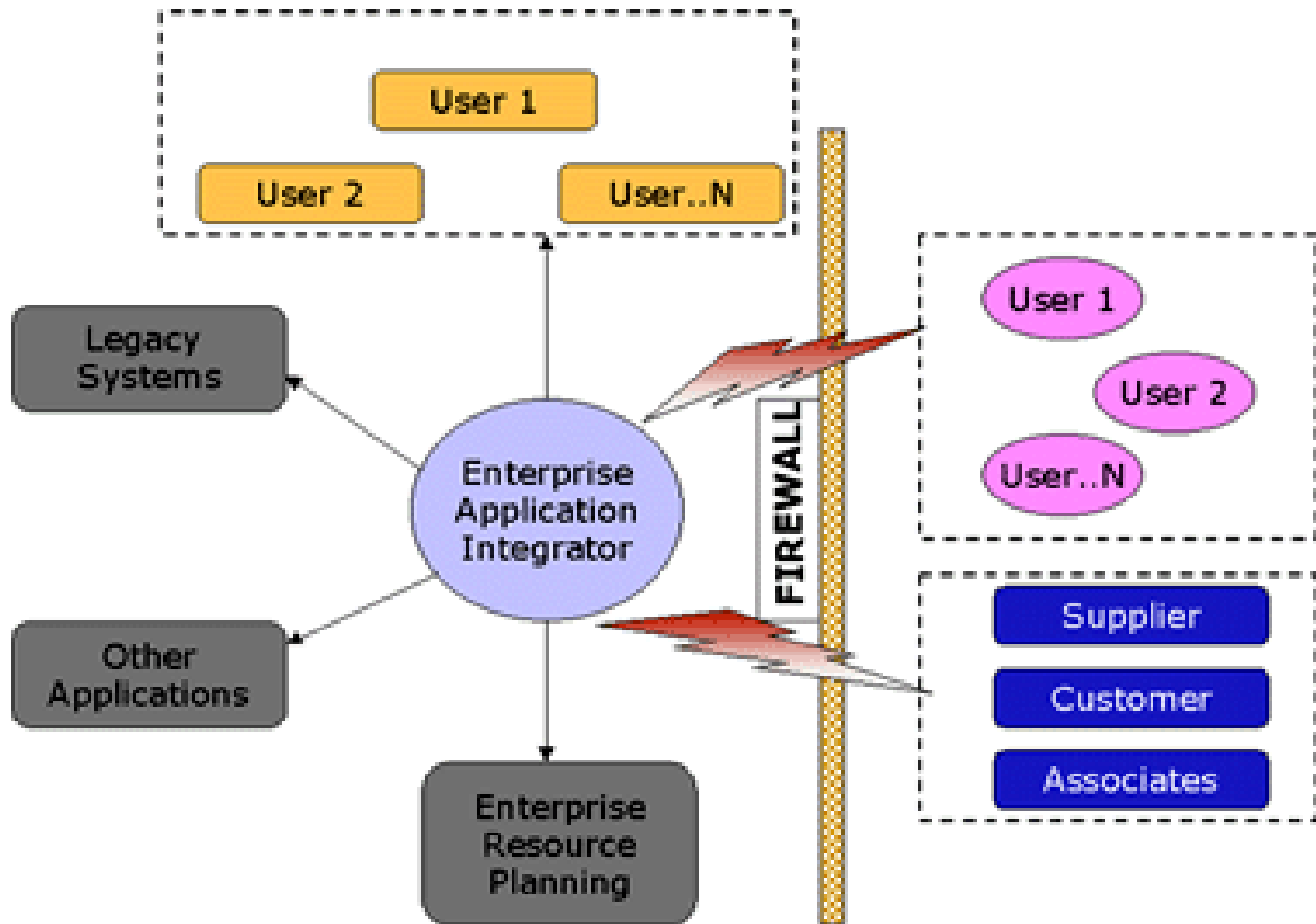
Web service e EAI (1)

I WS semplificano l'integrazione delle piattaforme EAI (Enterprise Application Integration), riducendo il bisogno di integrazione, in quanto tutti parlano lo stesso linguaggio o, al limite, usano la stessa grammatica (XML).

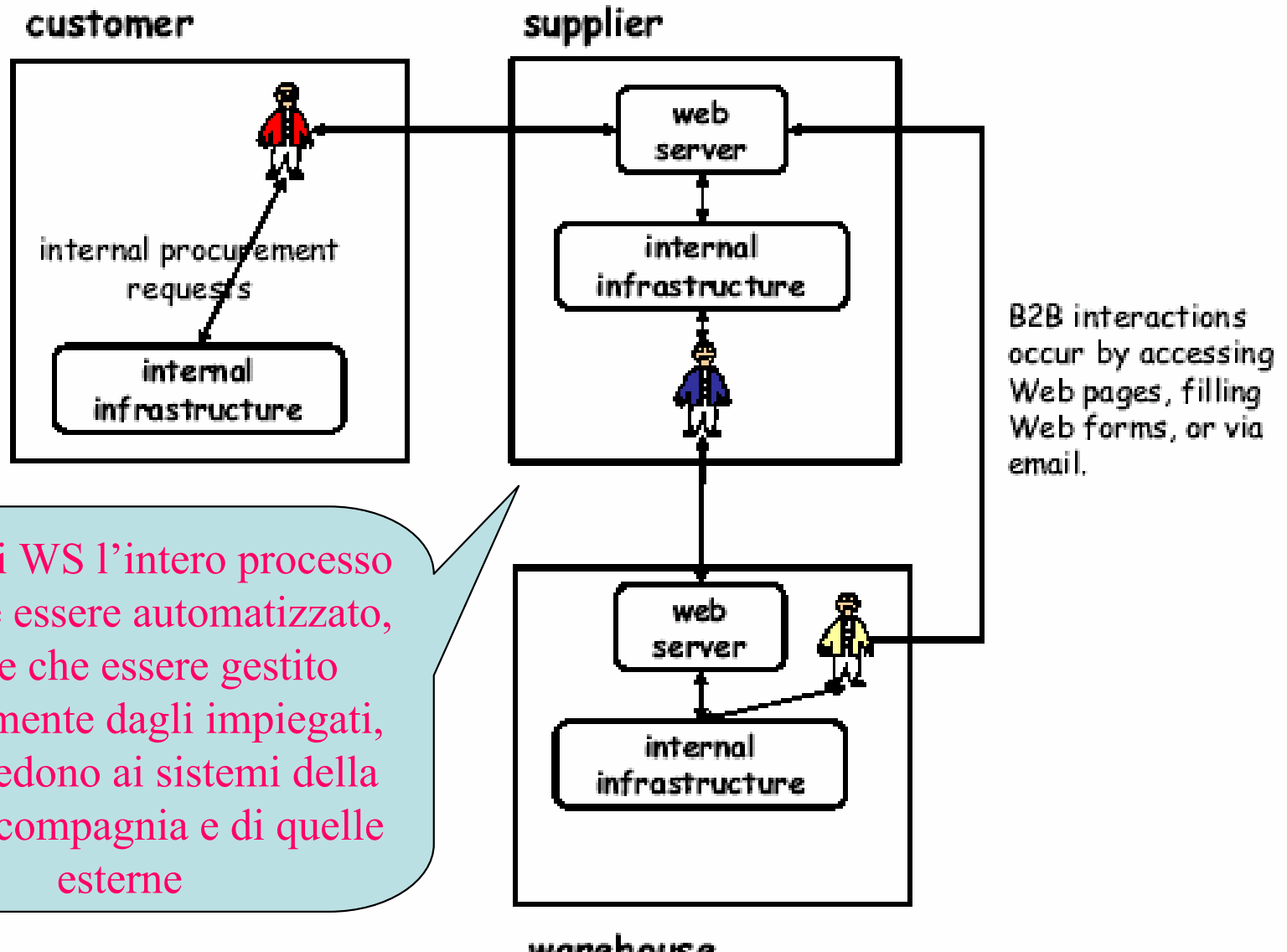
Rendono possibile l'automazione:

- delle **supply chain**
- dei **business process**

Web service e EAI (2)

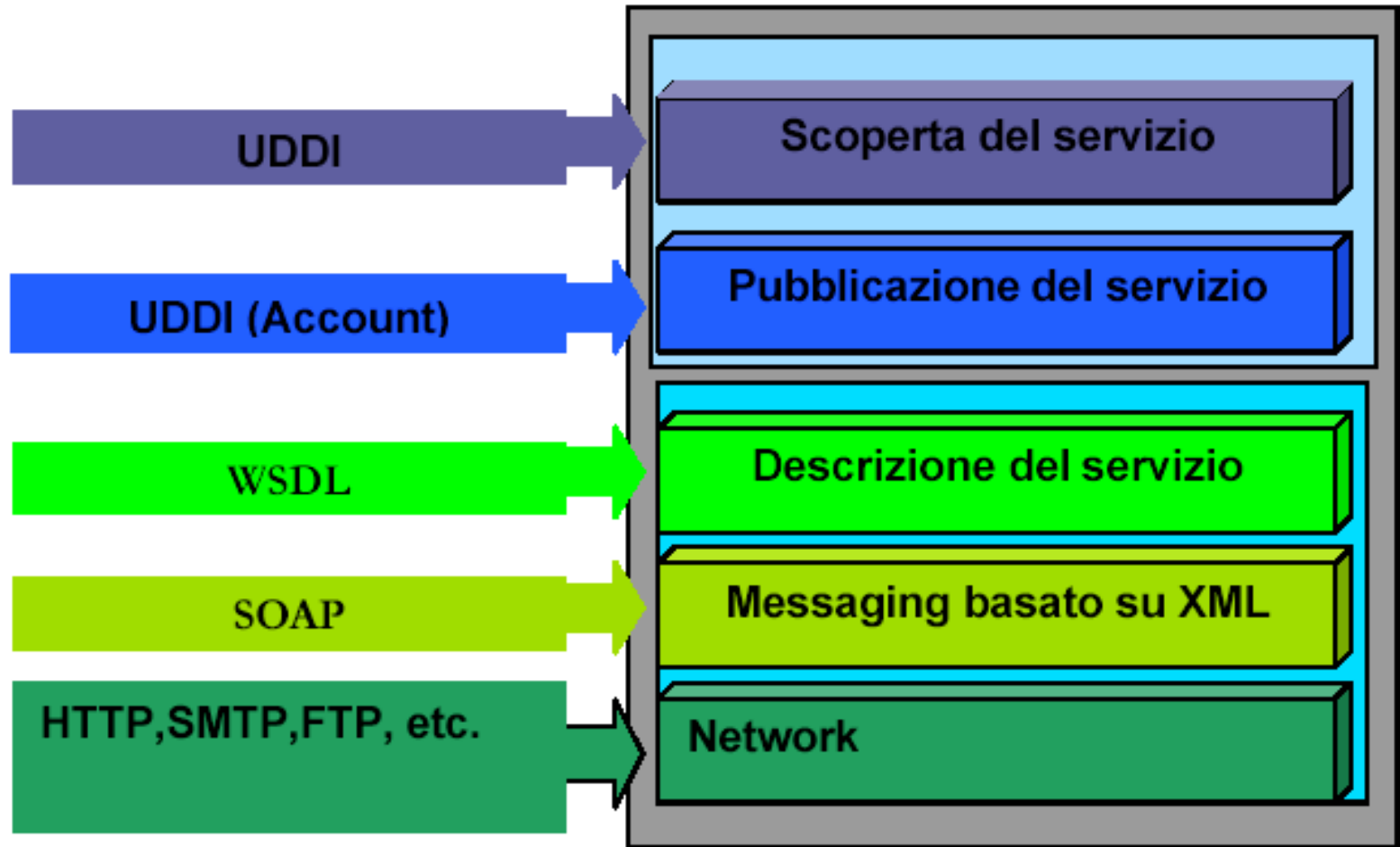


Web service e EAI: esempio



Grazie ai WS l'intero processo potrebbe essere automatizzato, invece che essere gestito manualmente dagli impiegati, che accedono ai sistemi della propria compagnia e di quelle esterne

Stack tecnologico dei WS



Basic Web Services Technologies

- **SOAP** (Simple Object Access Protocol)

- **WSDL** (Web Services Description Language)

- **UDDI** (Universal Description Discovery and Integration)

SOAP

The first version (1.0) 1999, latest version (1.2) June 2003

Canon, IBM, Microsoft and Sun

“A lightweight **XML**-based messaging protocol used to *encode* the information in Web service **request** and **response** messages before sending them over a network. SOAP messages are independent of any **operating system** or **protocol** and may be transported using a variety of **Internet protocols**, including SMTP, MIME, and HTTP”. Webopedia

SOAP: what is this?

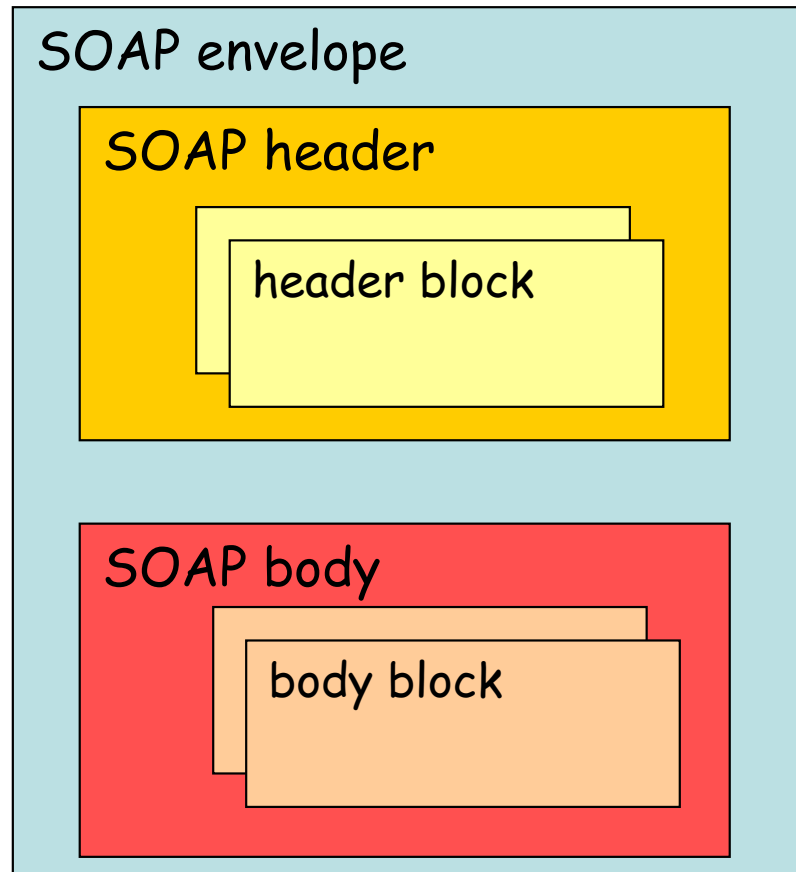
- stands for **Simple Object Access Protocol**
- is a **communication protocol** (stateless e one-way)
- is for communication **between applications**
- is a **format** for sending messages
- is designed to communicate via Internet
- is **platform independent**
- is **language independent**
- is based on **XML**
- is **simple** and **extensible**
- allows you to get around **firewalls**
- will be developed as a **W3C** standard

Firewall

Firewall: A system designed to prevent **unauthorized access** to or from a private network. Firewalls can be implemented in both hardware and software, or a combination of both. Firewalls are frequently used to prevent unauthorized Internet users **from accessing private networks** connected to the Internet, especially intranets. All messages entering or leaving the intranet pass through the firewall, which examines each message and **blocks** those that do not meet the specified security criteria.

Struttura di un messaggio SOAP

Un messaggio SOAP funge da *envelope*, in cui sono racchiuse tutte le informazioni che client e service si scambiano



Struttura di un messaggio SOAP (2)

Un messaggio SOAP ha un mittente e un ricevente e può avere una serie di intermediari (*node*), che processano e reindirizzano il pacchetto.

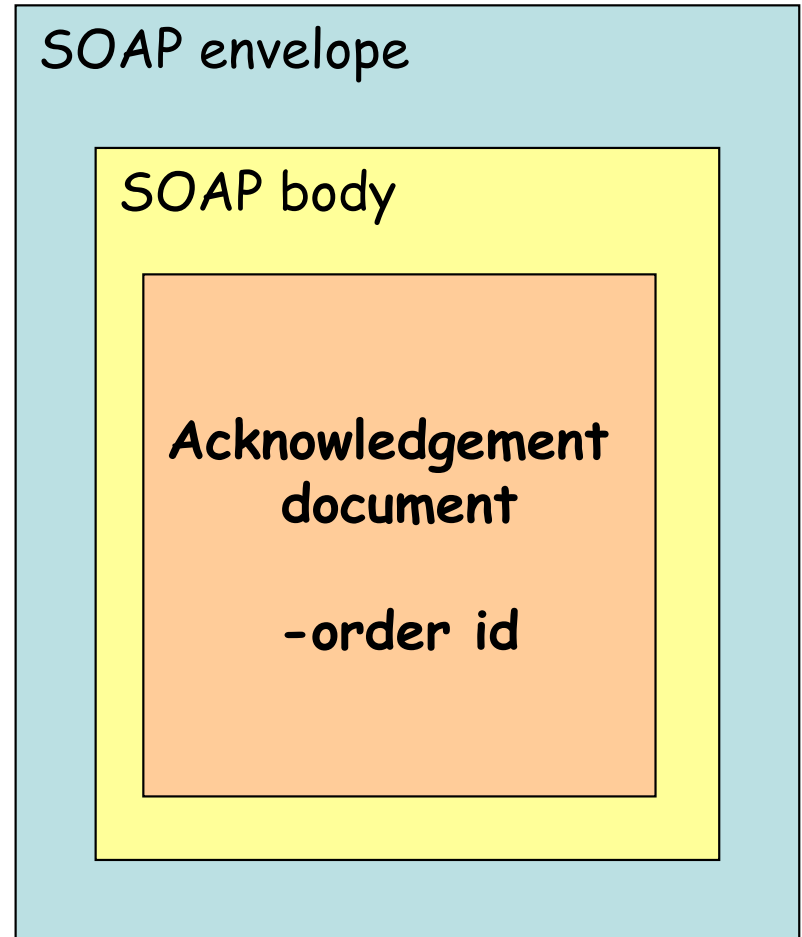
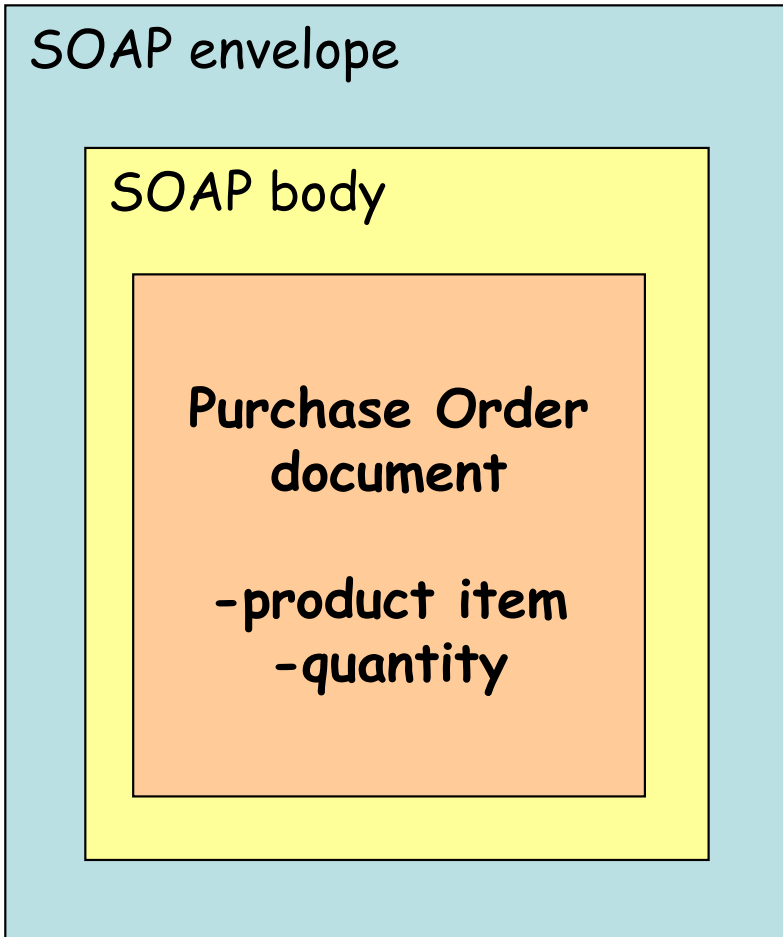
Le informazioni relative agli intermediari sono poste nell'*header* (opzionale), la parte *core* del messaggio (payload) è nel *body* (obbligatorio).

Interazioni SOAP

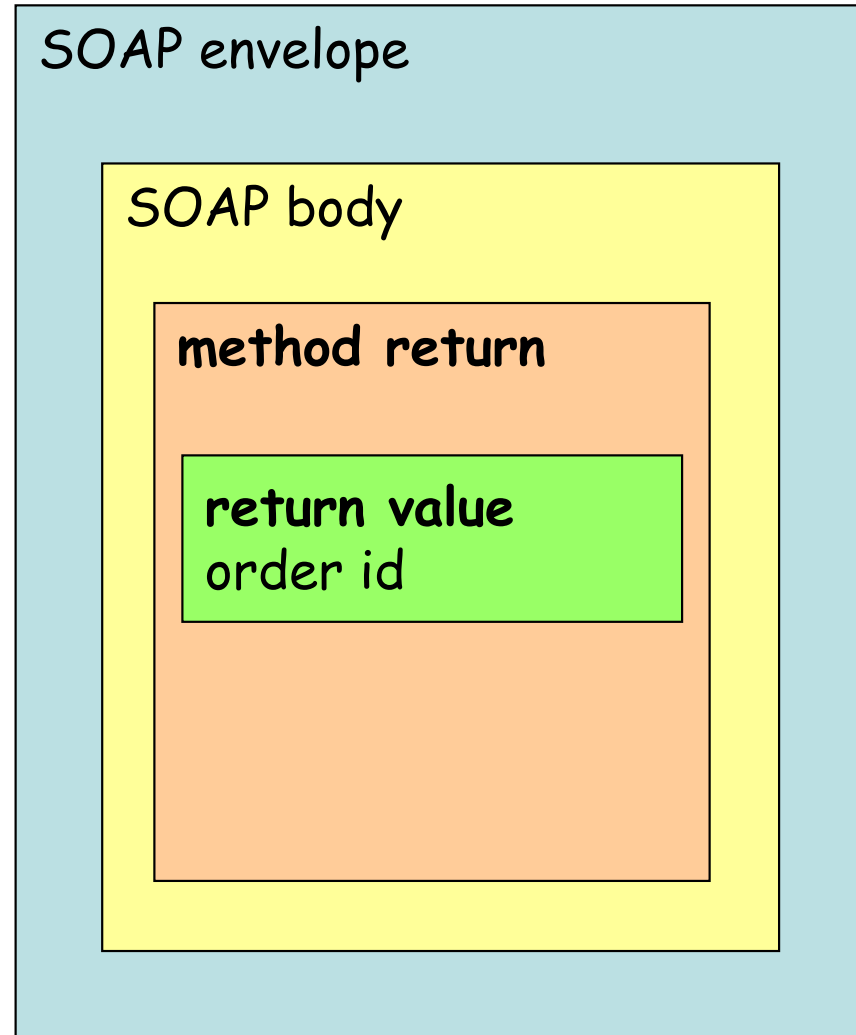
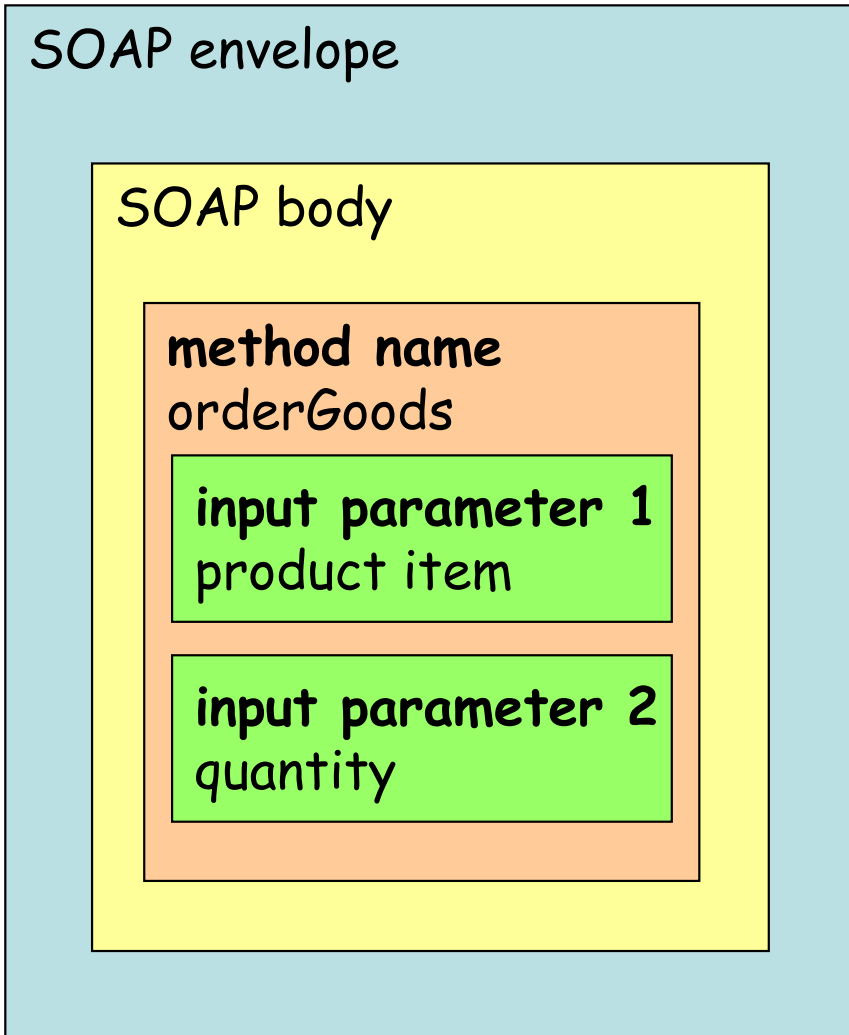
Esistono due stili di interazioni SOAP:

- **Document-style**: le due applicazioni che interagiscono sono d'accordo sulla struttura dei documenti scambiati. I messaggi SOAP trasmettono questi documenti da una applicazione all'altra.
- **RPC-style**: i messaggi SOAP contengono chiamate a metodi, sono specificati il nome del metodi, i parametri passati in input (*request message*) e quelli ritornati come output (*response message*).

Document-style interaction



RCP-style interaction



Messaggio di richiesta SOAP (1)

```
<?xml version="1.0" encoding="UTF-8"?>
SOAP-ENV:Envelope
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:SOAP-
ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <SOAP-ENV:Body>
    <ns1:getCAP xmlns:ns1="http://ejbean:8080/soap/">
      <arg xsi:type="xsd:string">Avellino</arg>
    </ns1:getCAP>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Messaggio di risposta SOAP (1)

```
<?xml version="1.0" encoding="UTF-8"?>
<SOAP-ENV:Envelope
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:SOAP-
ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <SOAP-ENV:Body>
    <ns1:getCAPResponse xmlns:ns1="
      http://ejbean:8080/soap/">
      <result xsi:type="xsd:string">Hello!</result>
    </ns1:getCAPResponse>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Regole di codifica

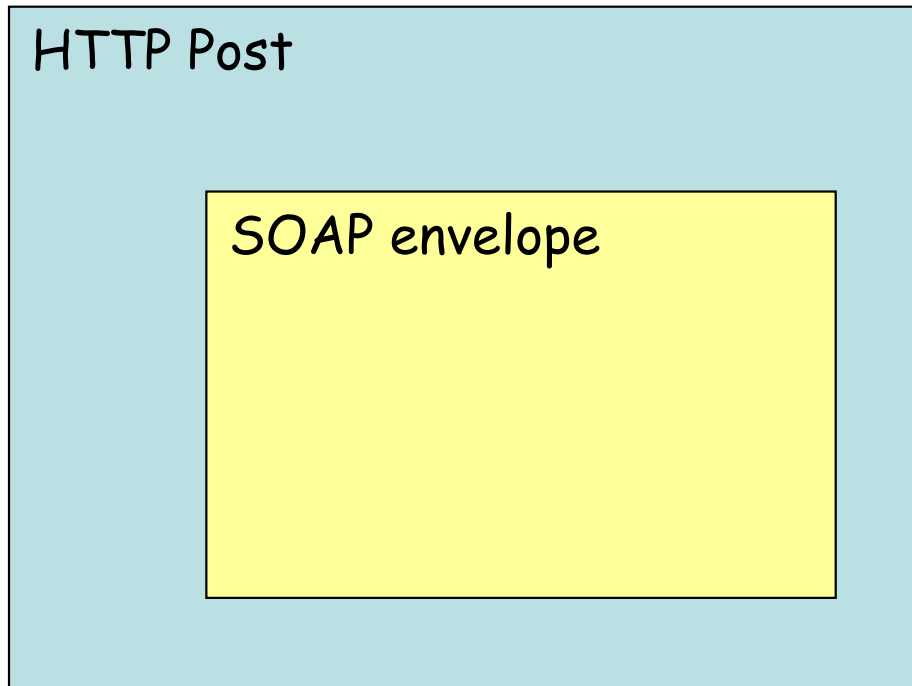
La struttura di un messaggio SOAP è determinata anche dalle regole di codifica, che definiscono come una struttura dati o un'entità è rappresentata in XML.

SOAP 1.2 definisce una particolare forma di codifica chiamata *SOAP encoding*, che definisce come i tipi base (interi, stringhe, ecc) e i tipi complessi (array e tipi composti) possono essere serializzati in XML.

SOAP lascia libertà di usare altri tipi di codifica (*literal encoding*).

```
<?xml version="1.0"?>
<soap:Envelope xmlns:soap="http://www.w3.org/2001/12/soap-
envelope"
soap:encodingStyle="http://www.w3.org/2001/12/soap-encoding">
<soap:Body >
...
</soap:Body>
</soap:Envelope>
```

SOAP e HTTP



Messaggio di richiesta SOAP (2)

Request

```
POST /StockQuote HTTP/1.1
Host: www.stockquoteserver.com
Content-Type: text/xml
Content-Length: nnnn
SOAPAction: "Some-URI"
```

Messaggio
HTTP-Post

```
<SOAP:Envelope
xmlns:SOAP="urn:schemas.xmlsoap.org:soap.v1">
<SOAP:Header>
<t:Transaction xmlns:t="URI"
mustUnderstand="1">5</t:Transaction>
</SOAP:Header>
<SOAP:Body>
<m:GetLastTradePrice
xmlns:m="URI">
<symbol>DIS</symbol>
</m:GetLastTradePrice>
</SOAP:Body>
</SOAP:Envelope>
```

Messaggio
SOAP

Messaggio di risposta SOAP (2)

Response

HTTP/1.1 200 OK

Content-Type: text/xml

Content-Length: nnnn

<SOAP:Envelope

xmlns:SOAP="urn:schemas.xmlsoap.org:soap.v1">

<SOAP:Header>

<t:Transaction xmlns:t="URI"

xsi-type="xsd:int"

mustUnderstand="">5</t:Transaction>

</SOAP:Header>

<SOAP:Body>

<m:GetLastTradePriceResponse

xmlns:m="URI">

<return>34.5</return>

</m:GetLastTradePriceResponse>

</SOAP:Body>

</SOAP:Envelope>

WSDL

The first version (1.0) September 2000, latest version (2.0) August 2004

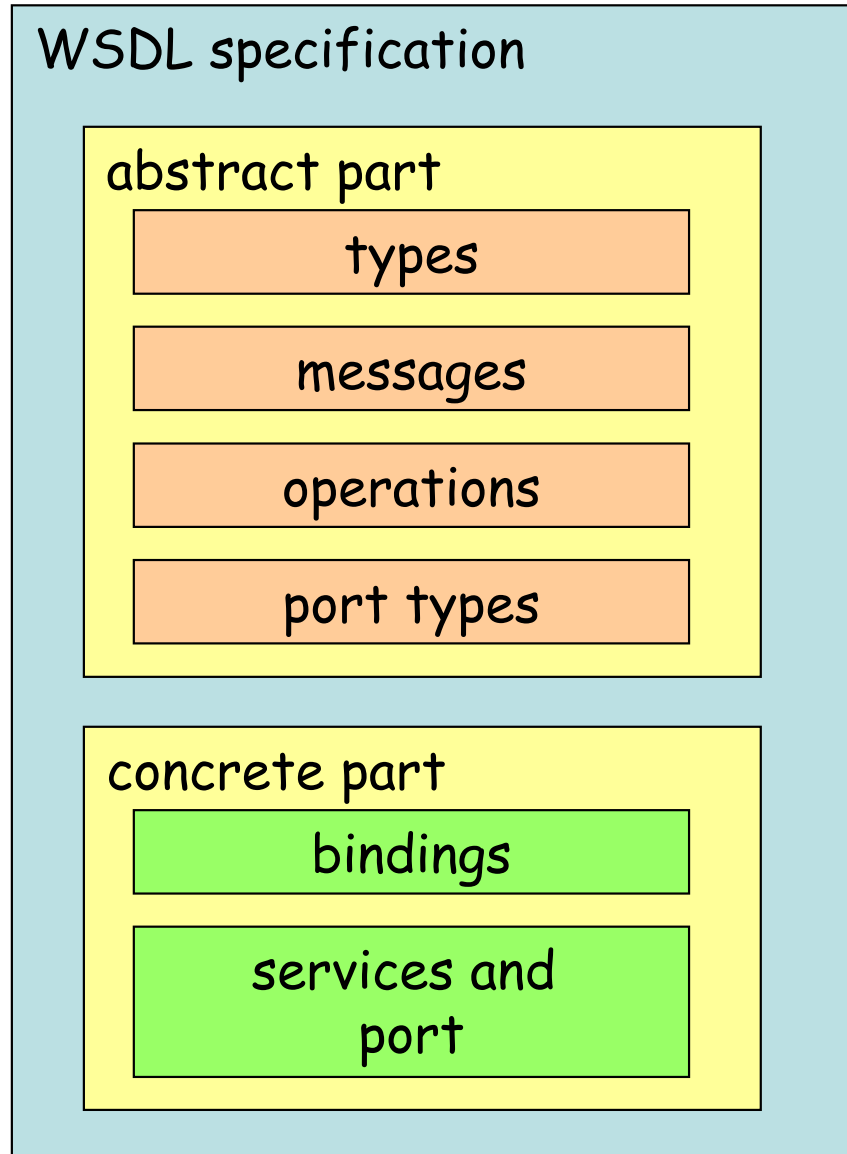
IBM, Microsoft, Ariba

È un linguaggio **XML-based** per la definizione delle **interfacce** dei WS.

L'interfaccia WSDL permette di specificare:

- I **metodi** supportati dal WS;
- Per ogni metodo gli **input** e gli **output**;
- Come accedere al WS (*location*)

Struttura di un'interfaccia WSDL



Abstract part

- **types**: fornisce informazioni su tipi di dati complessi, tale sezione non è presente quando vengono utilizzati solo tipi di dato semplici.
- **message**: una definizione *astratta* dei tipi di dato che saranno scambiati.
- **operation**: una descrizione astratta delle azioni supportate dal servizio.
- **portType**: un astratto set di operazioni supportato da uno o più endpoint.

Concrete part

- **binding**: describe come invocare le operazioni specificando il protocollo e il formato dei dati delle operation e dei message.
- **port**: specifica l'indirizzo di rete che implementa la specifica funzionalità relativa alla port type.
- **service**: specifica gli indirizzi delle porte per il binding; è un insieme di network endpoint o porte.

Esempio di interfaccia WSDL

```
targetNamespace="http://example.com/procurement/definitions"
xmlns:tns="http://example.com/procurement/definitions"
xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
xmlns="http://schemas.xmlsoap.org/wsdl/" >
```

```
<message name="OrderMsg">
  <part name="productName" type="xs:string"/>
  <part name="quantity" type="xs:integer"/>
</message>
```

**abstract
part
messages**

```
<portType name="procurementPortType">
  <operation name="orderGoods">
    <input message = "OrderMsg"/>
  </operation>
</portType>
```

**operation and
port type**

```
<binding name="ProcurementSoapBinding" type="tns:procurementPortType">
  <soap:binding style="document"
    transport="http://schemas.xmlsoap.org/soap/http"/>
  <operation name="orderGoods">
    <soap:operation soapAction="http://example.com/orderGoods"/>
    <input>
      <soap:body use="literal"/>
    </input>
    <output>
      <soap:body use="literal"/>
    </output>
  </operation>
</binding>
```

**concrete
part
binding**

```
<service name="ProcurementService">
  <port name="ProcurementPort" binding="tns:ProcurementSoapBinding">
    <soap:address location="http://example.com/procurement"/>
  </port>
```

**port and
service**

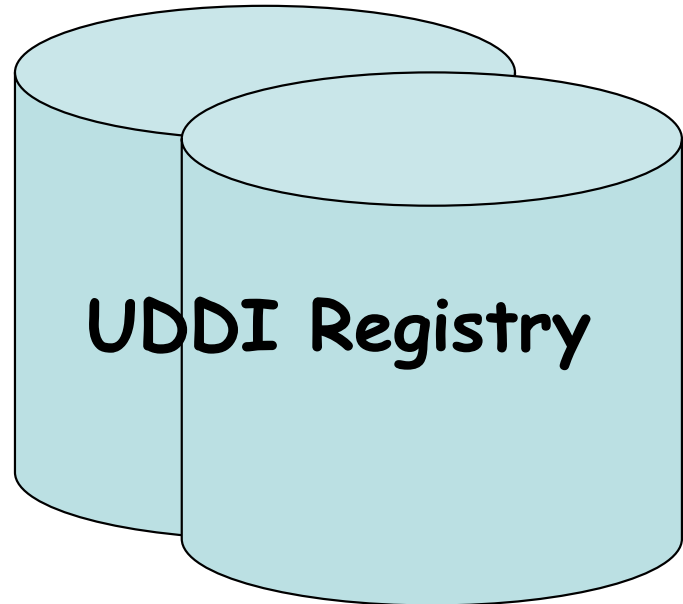
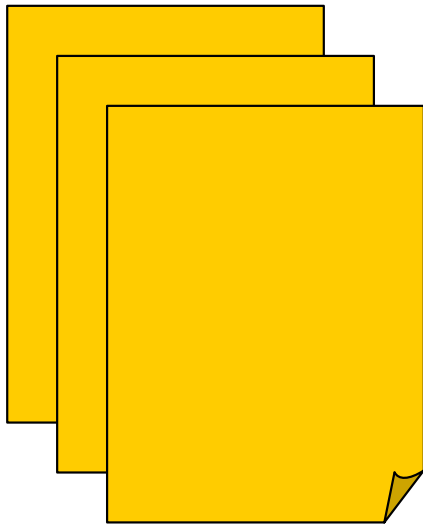
Tipi di operazioni

- one-way
- request-response
- solicit-response
- notification

UDDI

The first version (1.0) September 2000, latest version (3.0.2) October 2004.

IBM, Ariba e Microsoft, oggi il progetto è gestito dal consorzio OASIS



UDDI: scopo

“UDDI is the specification of a framework for **describing** and **discovering** Web services”

[Alonso et al. 2003]

Pubblicare la descrizione del servizio nel registro UDDI

Interrogare il registro per cercare un particolare servizio

Informazioni contenute nel registro UDDI

Possiamo classificare le informazioni contenute nel registro rispetto allo scopo per cui ciascuna informazione è utilizzata:

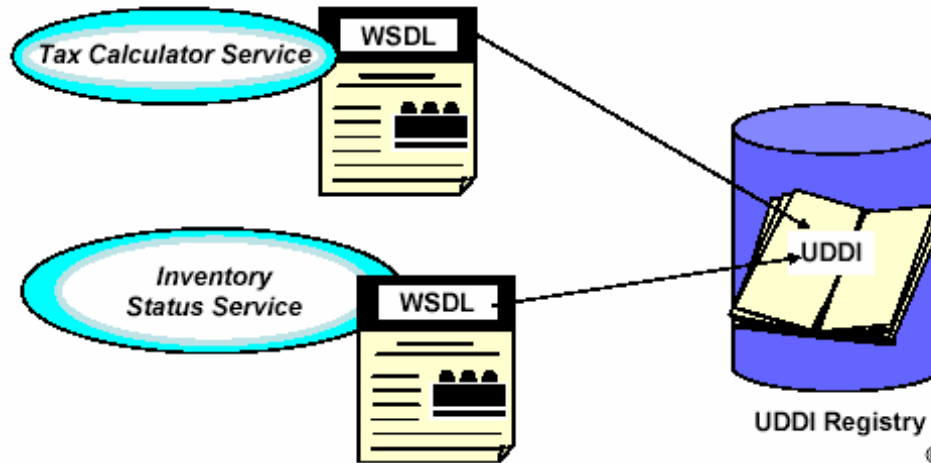
pagine **bianche**: elenco contenente il nome dell'organizzazione, i contatti (ad es. telefono o indirizzo e-mail) e il/i servizi offerto/i. La ricerca del WS è fatta in base alla società che lo fornisce.

pagine **gialle**: la classificazione è fatta rispetto ad una tassonomia (standardizzata o definita dall'utente). Si ricerca il servizio in base alla categoria a cui appartiene, in accordo con lo schema di classificazione.

pagine **verdi**: informazioni che descrivono come un servizio può essere invocato

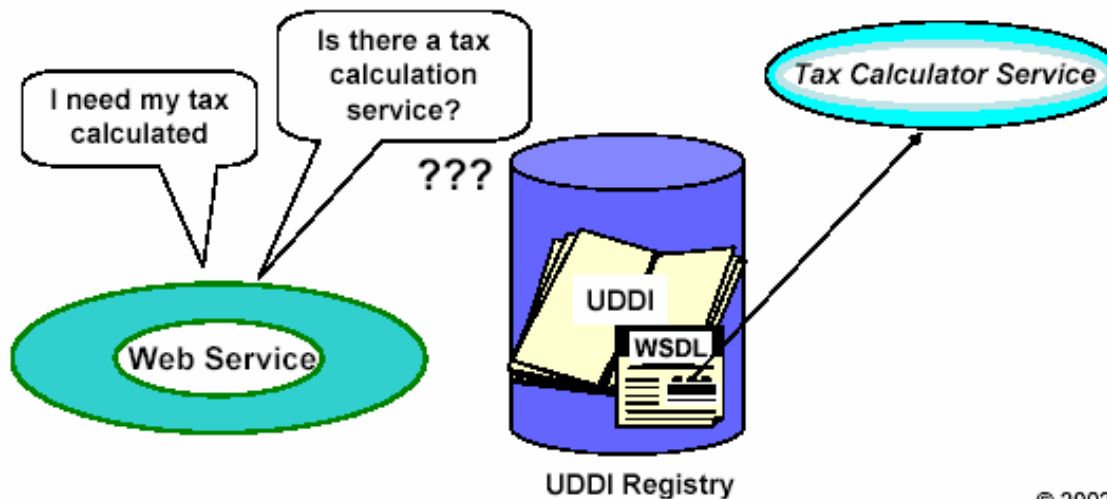
UDDI: esempio

How Web Services Are Registered



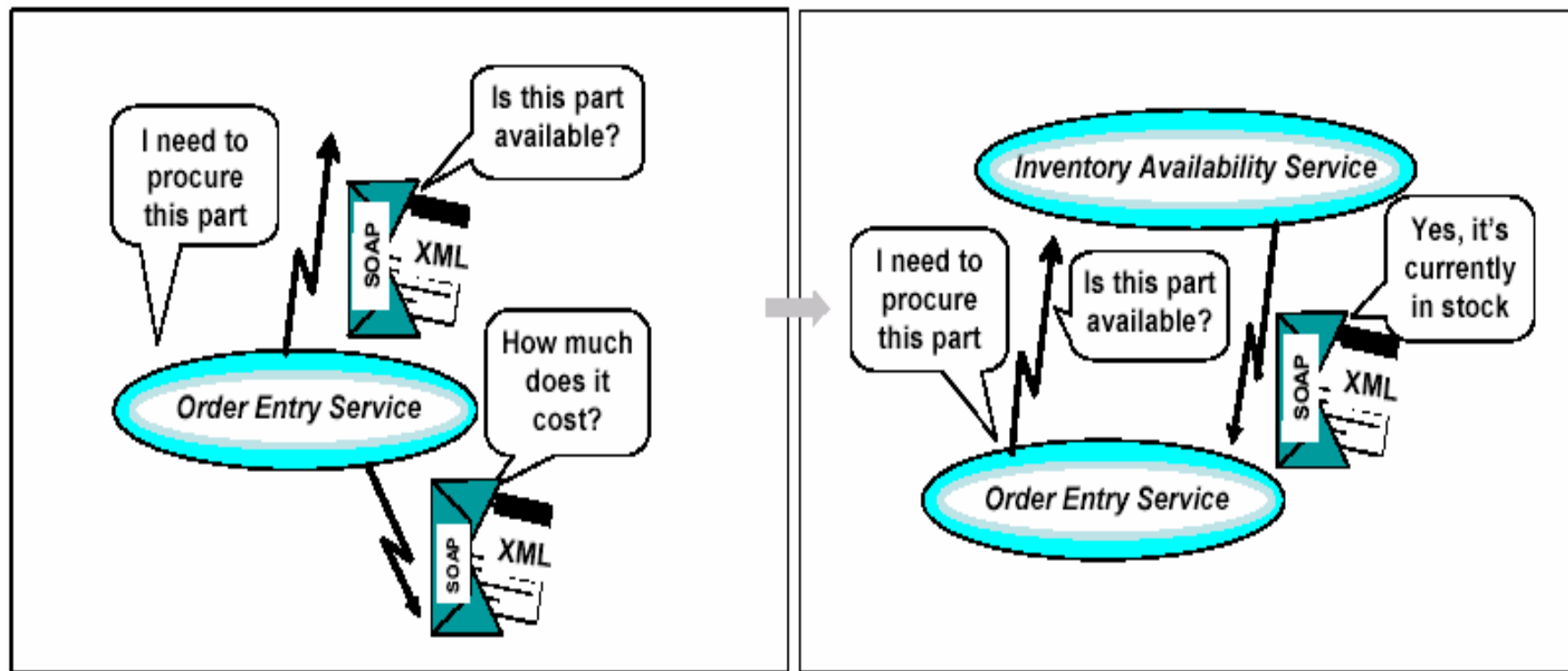
© 2002 Patricia Seybold Group, Inc.

How Web Services Are Discovered

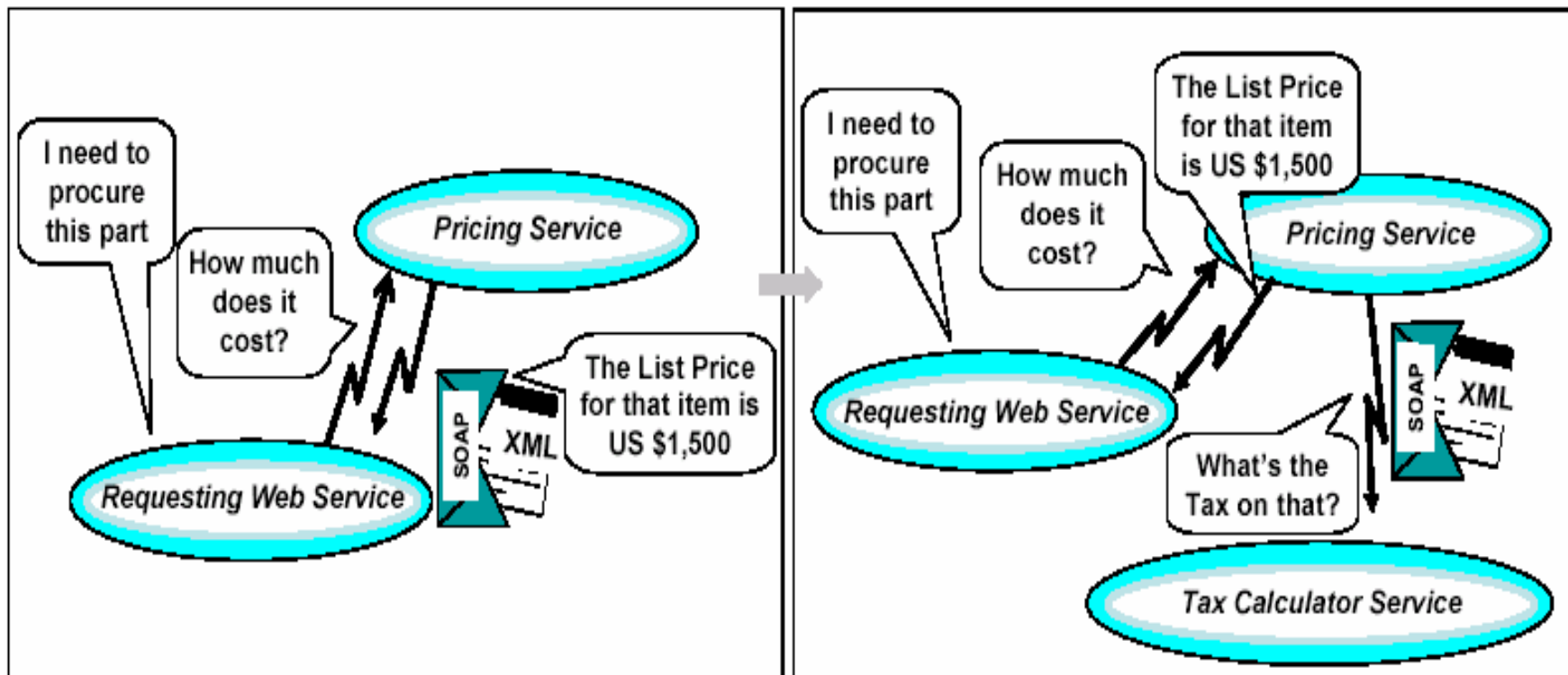


© 2002 Patricia Seybold Group, Inc.

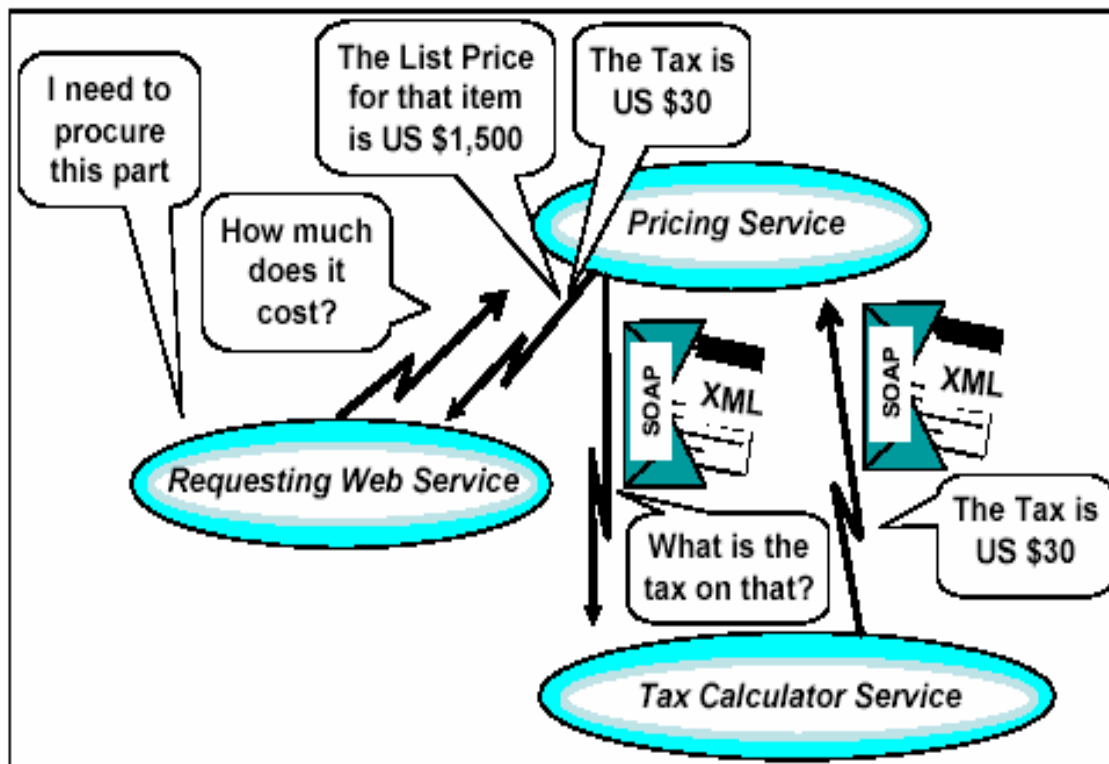
WS: esempio di interoperabilità (1)



WS: esempio di interoperabilità (2)

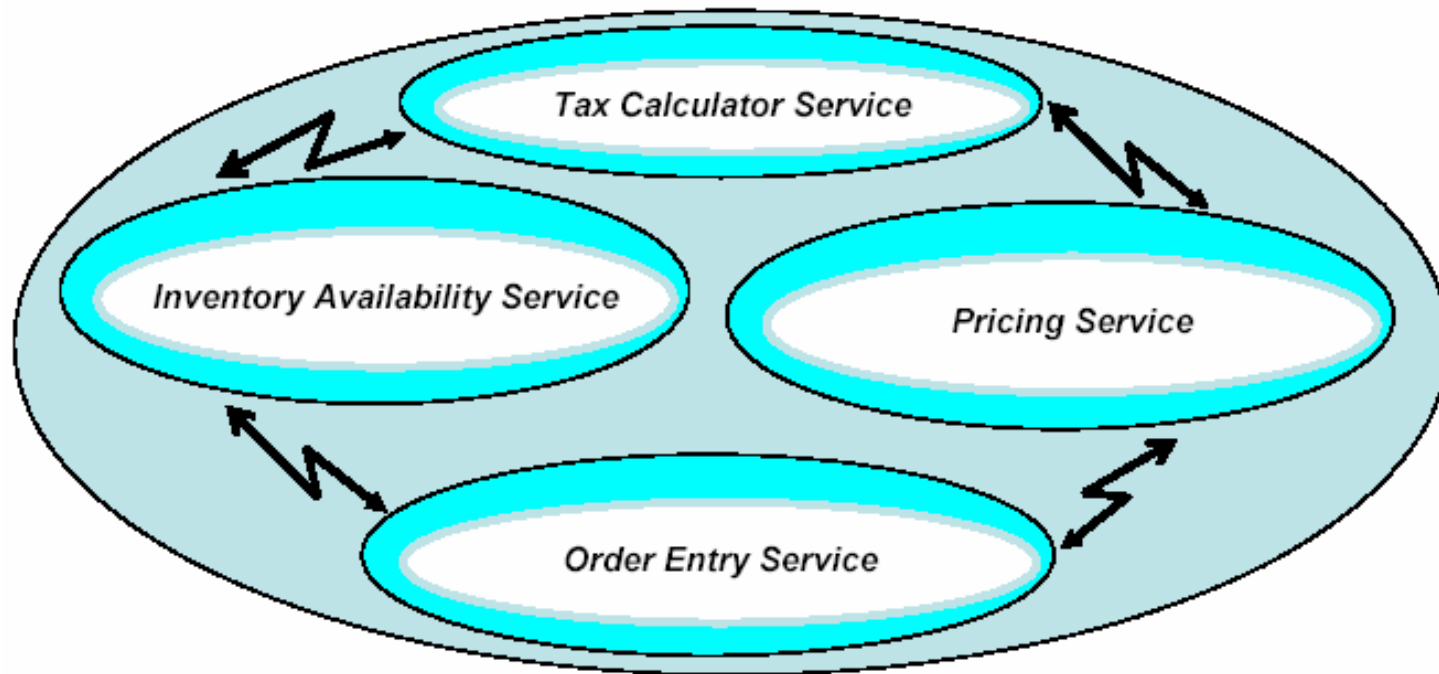


WS: esempio di interoperabilità (3)



(Web) service composition (1)

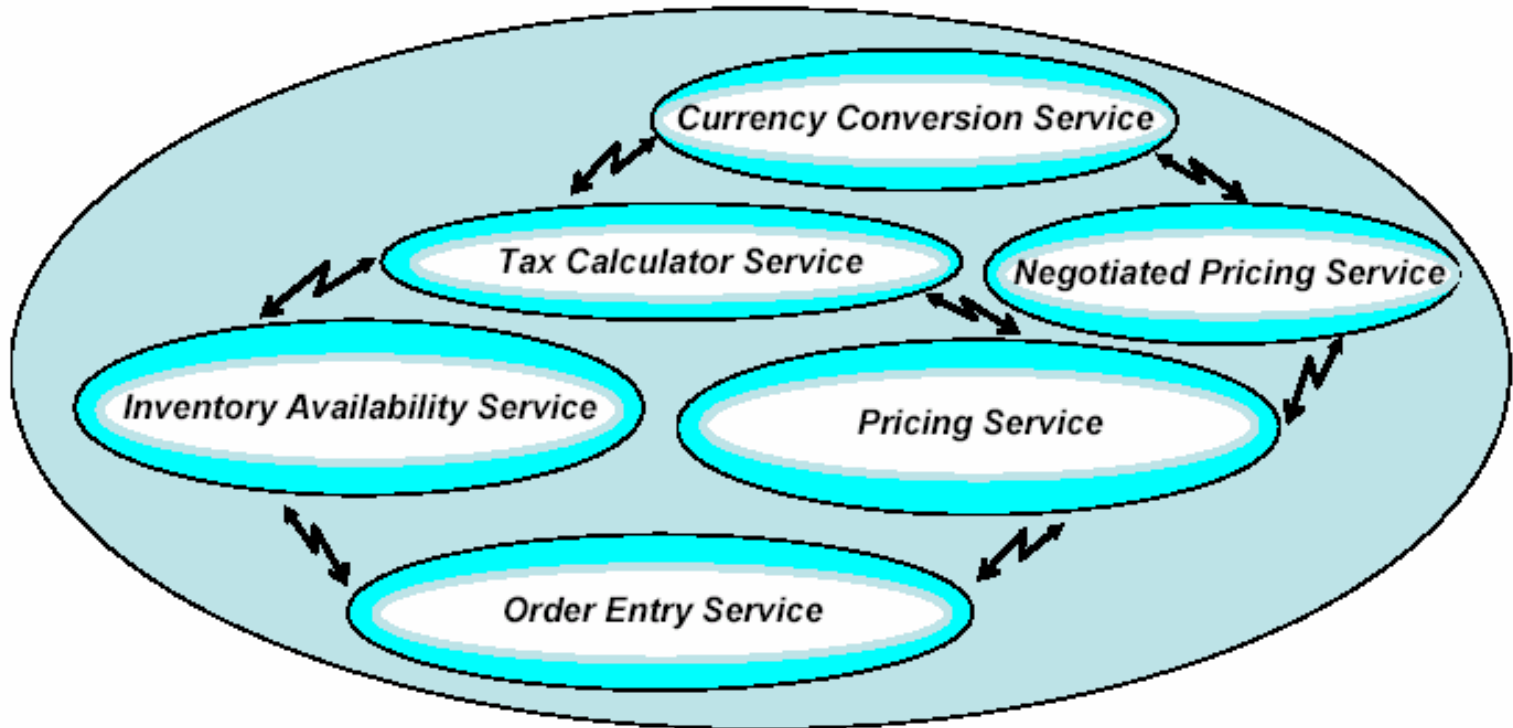
How Web Services Can Be Combined to Create a New Web Service



“Ordering a Line Item” Service

(Web) service composition (2)

How Web Services May Be Recombined Dynamically at Runtime



Ordering a Line Item in a Foreign Currency for a Customer with a Volume Discount

Bibliografia

• Alonso et al., Web services, Springer, 2002.

• Dal sito IBM:

<http://www-106.ibm.com/developerworks/webservices/newto/websvc.html>