

# WEB SERVICES

# word wide web (www)

- In the past years the www had a very large success basically for two reasons:
  - Simplicity
  - Ubiquity
- For a service provider is simple to contact a great number of users.
- For a user is simple to access a service everywhere it is located.
- However, www is strongly oriented to the interaction among users and systems.

# Interaction among distributed computing applications

- The evolution of the internet and web utilization made necessary the demand of interactions among applications in different scenarios:
  - A program of business management must be able to integrate the local informations with those made available on Internet by suppliers, banks or public administrations.
  - An application used by estate agents must access real time informations coming from worldwide financial markets.
  - The library management system can look for a book either in the local catalog or in those of a larger library management system (University, local government,...)

# Web limitations

- The web model is not suitable to the management of the interactions among applications.
- It is based on a very simple model:
  - the user, using a browser, sends a URL to a server.
  - the web server returns a HTML page that is graphically displayed by the browser.
- In a interaction between two applications this scheme presents two critical aspects:
  - The URL are a very simple tool in order to express complex and articulated requests.
  - HTML is a language basically used to describe how documents may be displayed.

# A new model

- In order to overcome the previously described difficulties a new model has been developed: **web services**.
- An evolutive approach has been chosen:
  - HTTP as application protocol in order to reuse the previous infrastructure and to maintain the interaction model simplicity.
  - XML as a tool able to better describe the fundamental elements of the interaction, the request and the response.

## **web service definition**

*"A WS is a software application described by a URI whose public interfaces and relative bindings are defined and described in XML. Its definition may be used by different software applications.*

*These applications can interact with the WS following the directives present in the service definition, using XML messages and internet transport protocols. "*

*("Web Services Architecture"*

*<http://www.w3c.org/TR/2002/WD-ws-arch-20021114>)*

- web services work on assumption that the functionality made available by a company will be exposed as a *service*.
- In middleware terms, a service is a procedure, method, or object with a a stable, **published interface** that can be invoked by clients.
- The invocation, *and this very important*, is made by a program. Thus, requesting and executing a service involves a program calling another program.
- Web services assume that services are loosely coupled, since in general they are defined, developed and managed by different companies

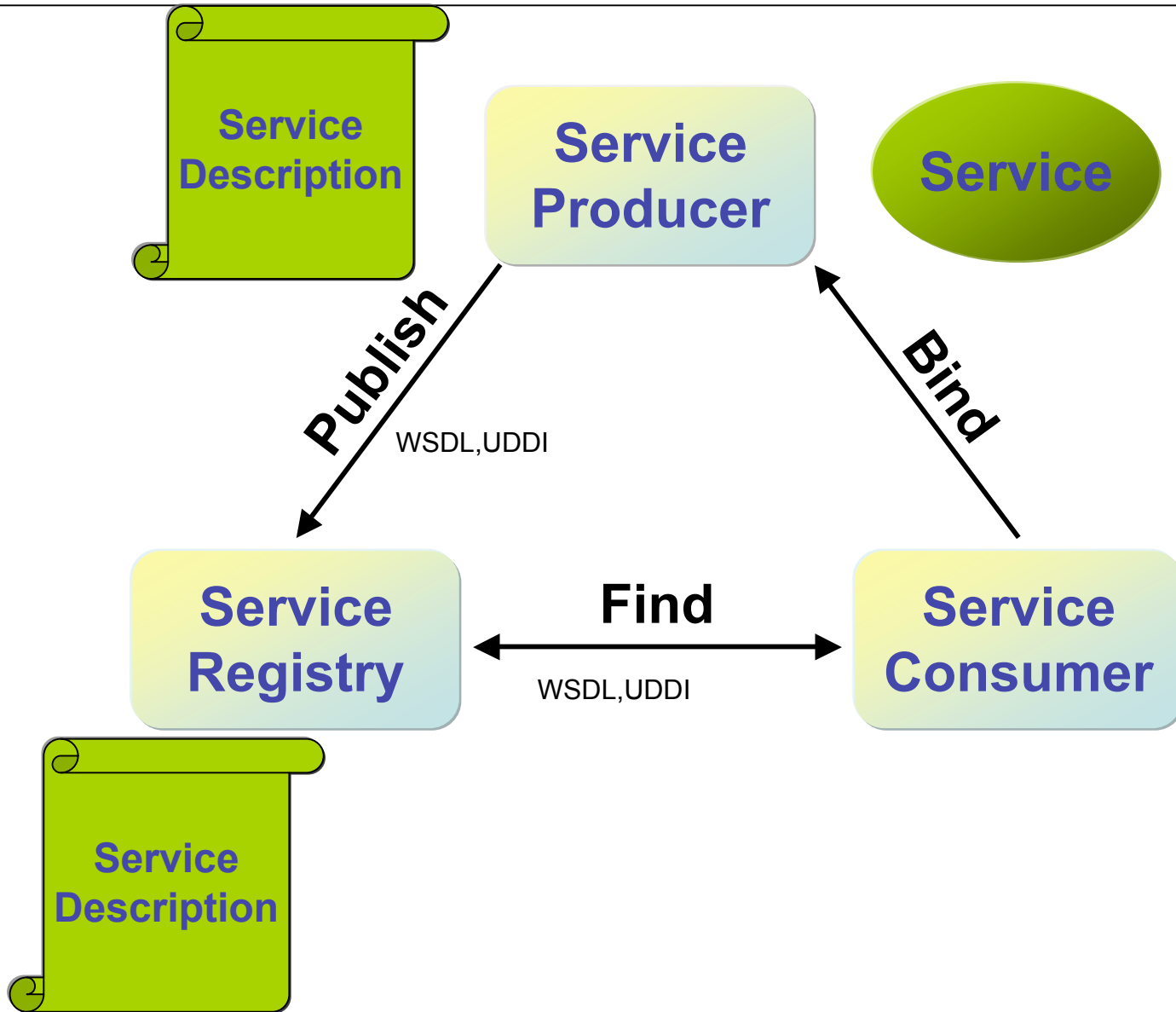
- The interface hides the implementation details of the service (hardware, software, programming language).
- A web service corresponds to a specific action or to a set of actions.
- It may be used by oneself or with other web services to implement a complex transaction.



## **A web service is defined as:**

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

# The Web Services Model -



## **Roles in a web services architecture**

### *Service producer.*

It is the service owner. From the architectural point of view it represents the node containing the service.

### *Service consumer*

It is the application which is calling the service . A service consumer may be a browser or an application (ex., another web service)

### *Service register*

It is a register which contains the descriptions of the services published because the request of a producer and requested by the service consumers.

# Operations in a web services architecture

- *Publish*

To be accessible a service description must be published allowing a service consumer to find.

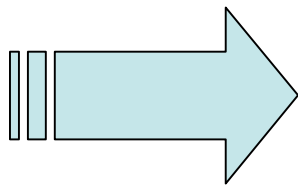
- *Find*

The find operation may happen in two different phases: during the design of the service to look for the interface description, at run time to implement the binding of the request service.

- *Bind*

It is the operation by which the service is localized

- The service consumer links up to a UDDI register (internet or intranet)
- It looks up (by categories, providers, ecc...) the requested service (example: web service providing stock rates)
- It finds the service description ( its functional aspects, the address of the server containing the service, how it is possible to call it)
- It connects himself to the provider server and utilizes the service.
- The service consumer is a software application



The web services are web applications that provide services to others web applications

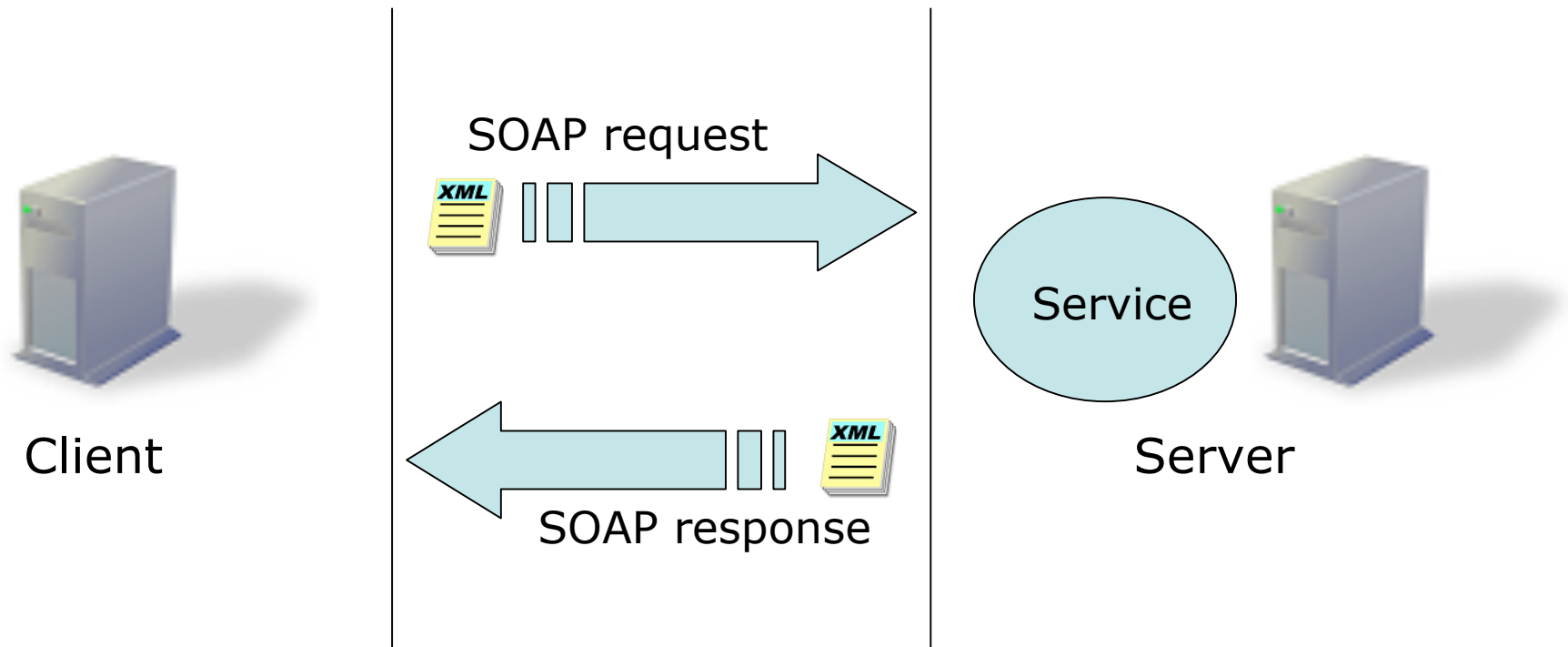
## Web Services – Technologies

---

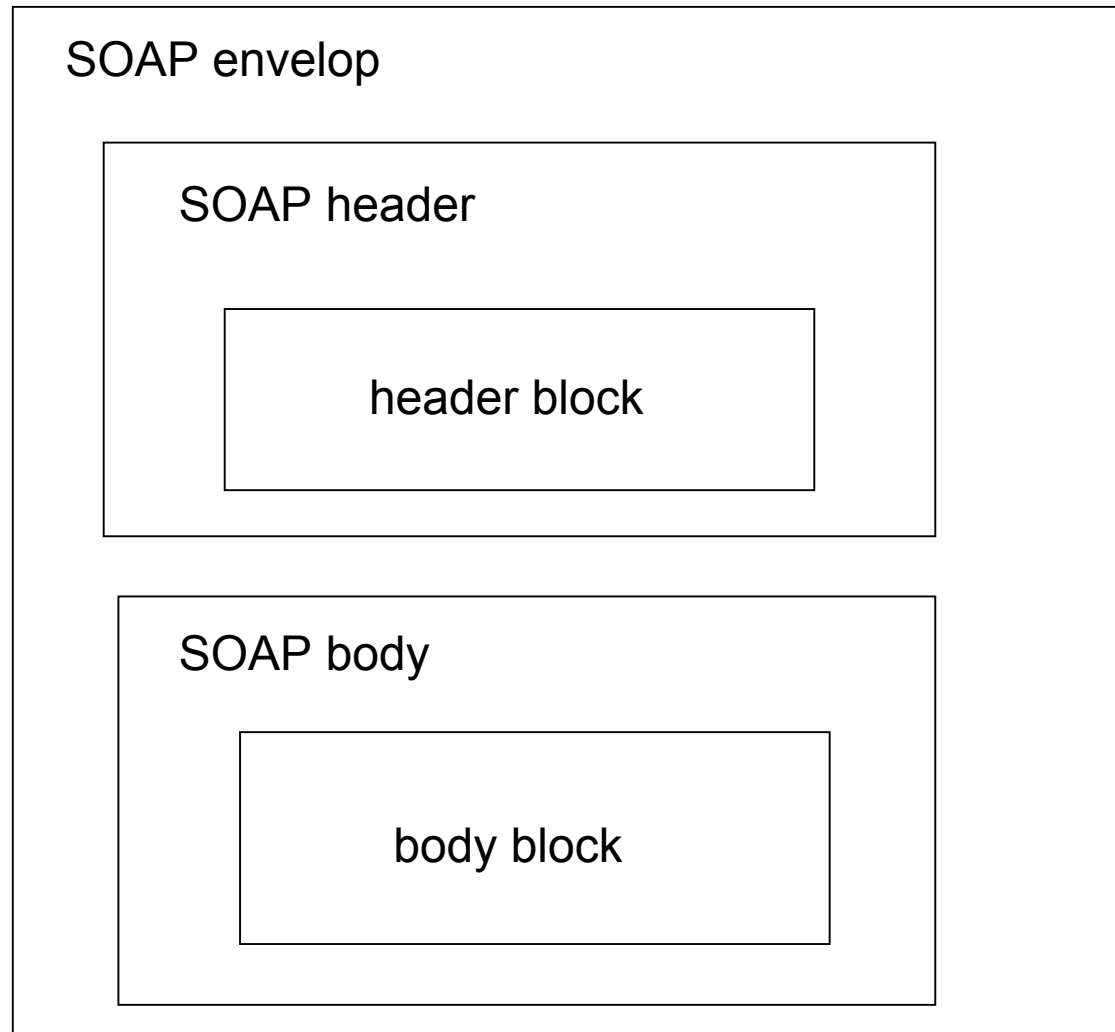
- SOAP – Simple Object Access Protocol – Provides a standard packaging structure for transporting XML documents over a variety of standard Internet technologies, including HTTP. SOAP provides a simple structure for doing RPC:document exchange.
- WSDL – Web Services Definition Language – XML thechnology that describes the interface of a web service in a standardized way. WDSL standardizes how input and output parameters can be represented and the function’s structure. WDSL allow disparate clients to automatically understand how to interact with a web service.
- UDDI – Universal Description, Discovery and Integration – UDDI provides a worldwide registry of web services. UDDI is used to discover available web services by searching for names, identifiers, categories.

- The client queries a UDDI registry for the service either by name, category, identifier. Once located, the client obtains information about the corresponding WSDL document.
- The WSDL document contains information about how to contact the web service and the format of request messages in XML schema.
- The client creates a SOAP message in accordance with the XML schema found in WSDL and sends a request to the host (where the service is).

The client creates a request (SOAP request), the server waits on a port, accepts the request, executes the service (service) and sends the result to the caller (SOAP response)







SOAP message schematic representation

## Web Services –SOAP example of a request

---

```
POST /StockQuote HTTP/1.1
Host: www.stockquoteserver.com
Content-Type: text/xml; charset="utf-8"
Content-Length: nnnn
SOAPAction: "Some-URI"
```

```
<SOAP-ENV:Envelope
  xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
  SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
  <SOAP-ENV:Body>
    <m:GetLastTradePrice xmlns:m="Some-URI">
      <symbol>DIS</symbol>
    </m:GetLastTradePrice>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

## Web Services –SOAP example of a response

---

HTTP/1.1 200 OK

Content-Type: *text/xml; charset="utf-8"*

Content-Length: *nnnn*

<SOAP-ENV:***Envelope***

xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"

SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding"/>

<SOAP-ENV:***Body***>

<m:**GetLastTradePriceResponse** xmlns:m="Some-URI">

<Price>**34.5**</Price>

</m:GetLastTradePriceResponse>

</SOAP-ENV:*Body*>

</SOAP-ENV:*Envelope*>

# WSDL

- In order to utilize a service it is necessary to know its characteristics.
- This function is implemented by the WSDL language.
- WDSL defines:
  - What a service is able to do (requests, responses and parameters)
  - Where it is located
  - How to call it

# WSDL: type description

- The first part of a WSDL document describes the types.

```
<types>
  <schema>
    <element name="TradePriceRequest">
      <complexType>
        <all>
          <element name="tickerSymbol" type="string"/>
        </all>
      </complexType>
    </element>
    <element name="TradePrice">
      <complexType>
        <all>
          <element name="price" type="float"/>
        </all>
      </complexType>
    </element>
  </schema>
</types>
```

# WSDL: messages and operations

- description of the messages:

```
<message name="GetLastTradePriceInput">
  <part name="body" element="xsd1:TradePriceRequest"/>
</message>
<message name="GetLastTradePriceOutput">
  <part name="body" element="xsd1:TradePrice"/>
</message>
```

- Description of the operations : each operation is constituted by a request message and a response message :

```
<portType name="StockQuotePortType">
  <operation name="GetLastTradePrice">
    <input message="tns:GetLastTradePriceInput"/>
    <output message="tns:GetLastTradePriceOutput"/>
  </operation>
</portType>
```

# WDSL: binding and service

- The binding è the link among the type of an operation (type) , the name of an operation (name) and the action to be performed (soapAction):

```
<binding name="StockQuoteSoapBinding"
  type="tns:StockQuotePortType">
  <soap:binding>
    <operation name="GetLastTradePrice">
      <soap:operation soapAction="..." />
      <input>...</input>
      <output>...</output>
    </operation>
  </binding>
```

- The last part of the document describes the service and its web address.

```
<service name="StockQuoteService">
  <port name="StockQuotePort"
    binding="tns:StockQuoteBinding">
    <soap:address
location="http://www.stockquote.com" />
  </port>
</service>
```

# UDDI

- UDDI roughly represents the equivalent of an on line book phone that enable the finding of web services
- UDDI offers 3 services whose names are typical of the telephony word.
  - White pages: finding a service by name
  - Yellow pages: finding a service by category
  - Green Pages: give technical informations on the services provided by a particular company.
- A UDDI service provider (IBM, Microsoft, SAP ecc.) manages an register called denominato UBR (UDDI Business Registry) that is accessible for publishing and finding the web services



# Web Services: summary

- Web services are autonomous applications, modular and self describing that can be published, retrieved and called.
- Allow to implement interactions among applications in Internet.
- A web service is constituted by three elements:
  - A mechanism of publication and research :UDDI
  - A mechanism of description: WSDL
  - A mechanism of calling: SOAP