Phát triển phần mềm phân tán sử dụng các dịch vụ Web

Cao Tuấn Dũng
Trịnh Tuấn Đạt

Nội dung

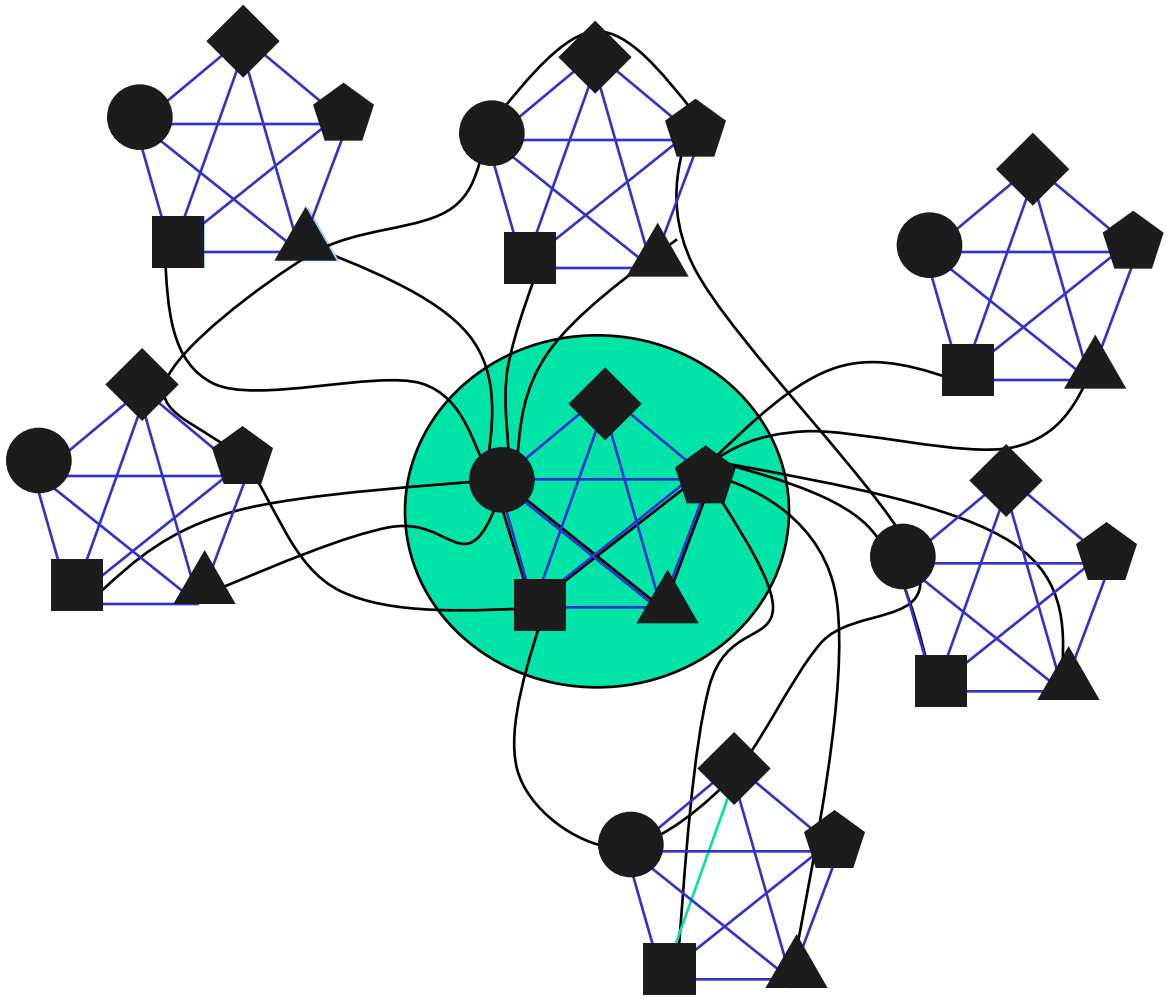
- 1. Tổng quan về Web Services
- 2. Các công nghệ sử dụng
  - 2.1. XML
  - 2.2. SOAP
  - 2.3. WSDL
  - 2.4. UDDI
- 3. Phát triển ứng dụng Web Services với Java

1. Tổng quan về Web Services

1. Ngữ cảnh

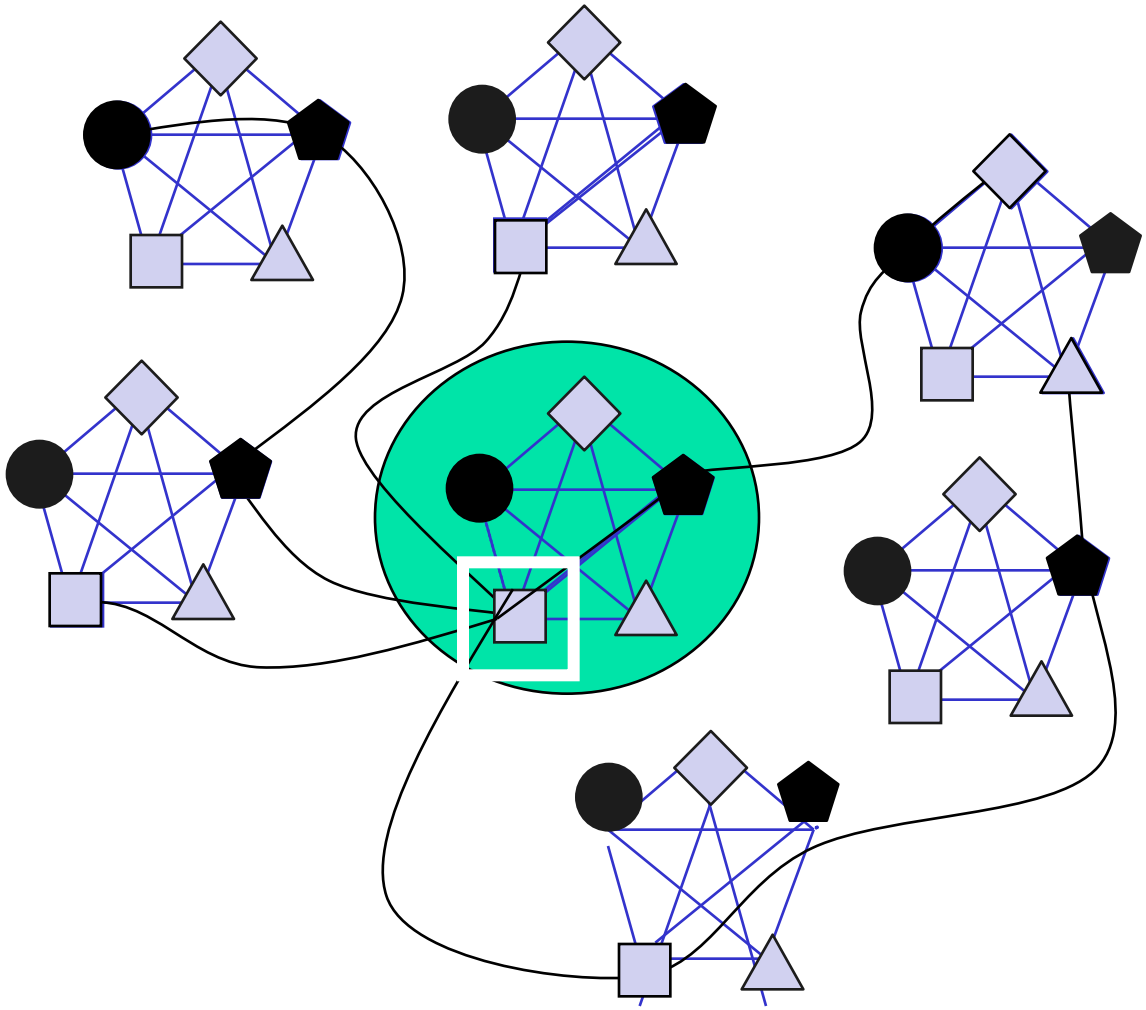
- Kinh doanh ngày nay:
  - Thách thức với CEO
    - Sửa đổi, nâng cấp dịch vụ khách hàng
    - Cải tiến quy trình sản xuất
    - Quyết đóan hơn
  - Thách thức với CIO
    - Tái định hướng kiến trúc IT
    - Kết nối với nhiều hơn với các thành phần bên ngoài
    - Mở rộng các xử lý với các thành phần bên ngoài

1. Ngữ cảnh



Các tiến trình thương mại trên Internet mở rộng các công ty

1. Tổng quan về Web Services



Các công nghệ phát triển cũng phải đáp ứng theo xu hướng

Yêu cầu đặt ra

- Các Công ty, Nhà cung cấp, Khách hàng cần có thể làm việc cùng nhau:
  - Nhanh hơn bao giờ hết
  - Trên Internet
  - Hoặc sẽ đối mặt nguy cơ "bị cô lập"

Các giải pháp

- Tính toán phân tán truyền thống
- Web sites (portals)
- Web Services

Tính toán phân tán

- Mô hình Client/server
  - Khó mở rộng
  - Không an toàn
- Mô hình Distributed object
  - Components: Được đóng gói và có thể cộng tác
  - Remoting: remote method invocation
  - COM, CORBA, Java RMI và EJB
    - Không dễ triển khai
    - Khó khăn trong cộng tác: chưa đề ra các chuẩn
    - Vẫn khó mở rộng

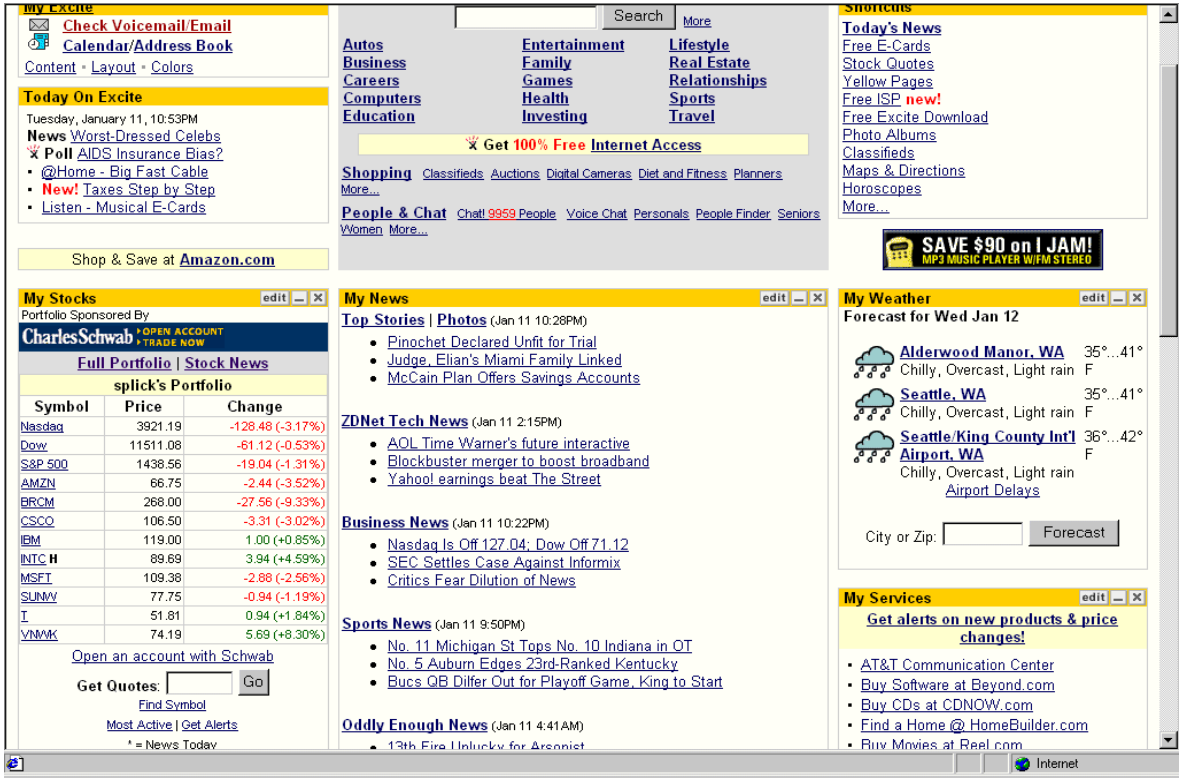
Portal

Ads

Mail

Calendar

Finance



Other Svcs

Weather

News

Hạn chế của portal

- Không có cách thức chuẩn tắc để thể hiện các chức năng
- Tích hợp: đòi hỏi chi phí cao, dễ gặp lỗi
- Khó outsource
- Không được thiết kế để có thể sử dụng ngoài mục đích ban đầu

Web Service là gì?

- Giải pháp là Web Services!
- Web Service cung cấp chức năng cho khách hàng:
  - Qua Internet hoặc intranet, thông qua 1 URL
- Dựa trên các chuẩn Web
  - HTTP, XML, SOAP, WSDL, UDDI
- Có thể được cài đặt với bất kỳ nông ngữ nào, trên bất kỳ platform nào
- Black boxes
  - Dễ dàng tái sử dụng, như 1 component
- Web service: đơn thuần là 1 dịch vụ cung cấp sẵn trên Web

Web Service là gì?

- Web Service kết hợp các đặc tính tốt nhất của distributed computing và portals & loại bỏ đi các nhược điểm:
  - Cung cấp cơ chế gọi phương thức từ xa
  - Sử dụng các chuẩn Web (vd: HTTP, XML)

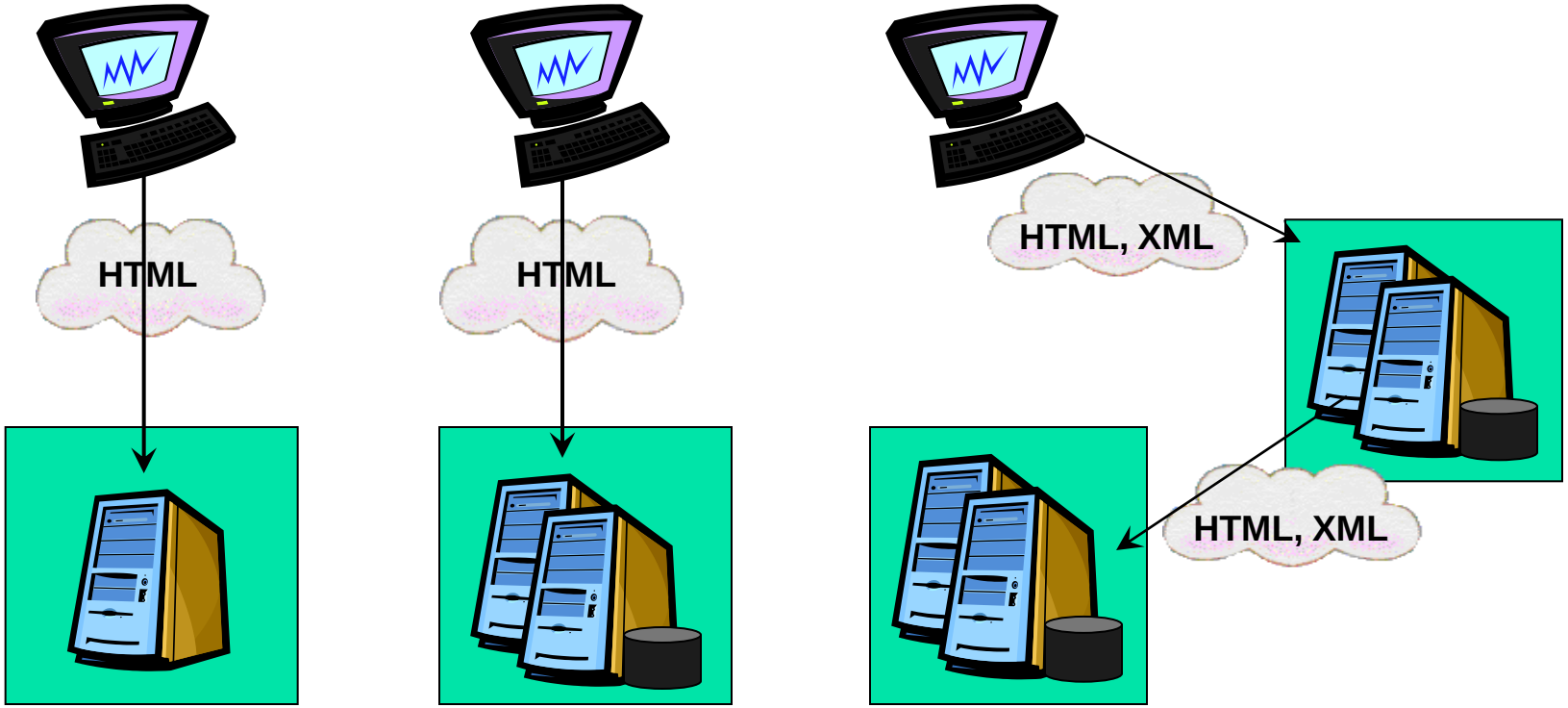
Web Service là gì?

- Web Services cho phép kết nối:
  - Các công ty khác nhau
  - Nhiều & bất cứ thiết bị nào
  - Các Applications
  - Các clients khác nhau
    - Không chỉ browsers
- Phân phối và tích hợp các application logic
- Cho phép programmable Web
  - Không chỉ đơn thuần là interactive Web
- Web Services là loosely coupled

Web Service là gì?

- Mô thức mới cho sự phát triển của internet
- Chuyển giao ứng dụng như là các dịch vụ: SaaS
- Internet thế hệ thứ 3

Tiến hóa của Web

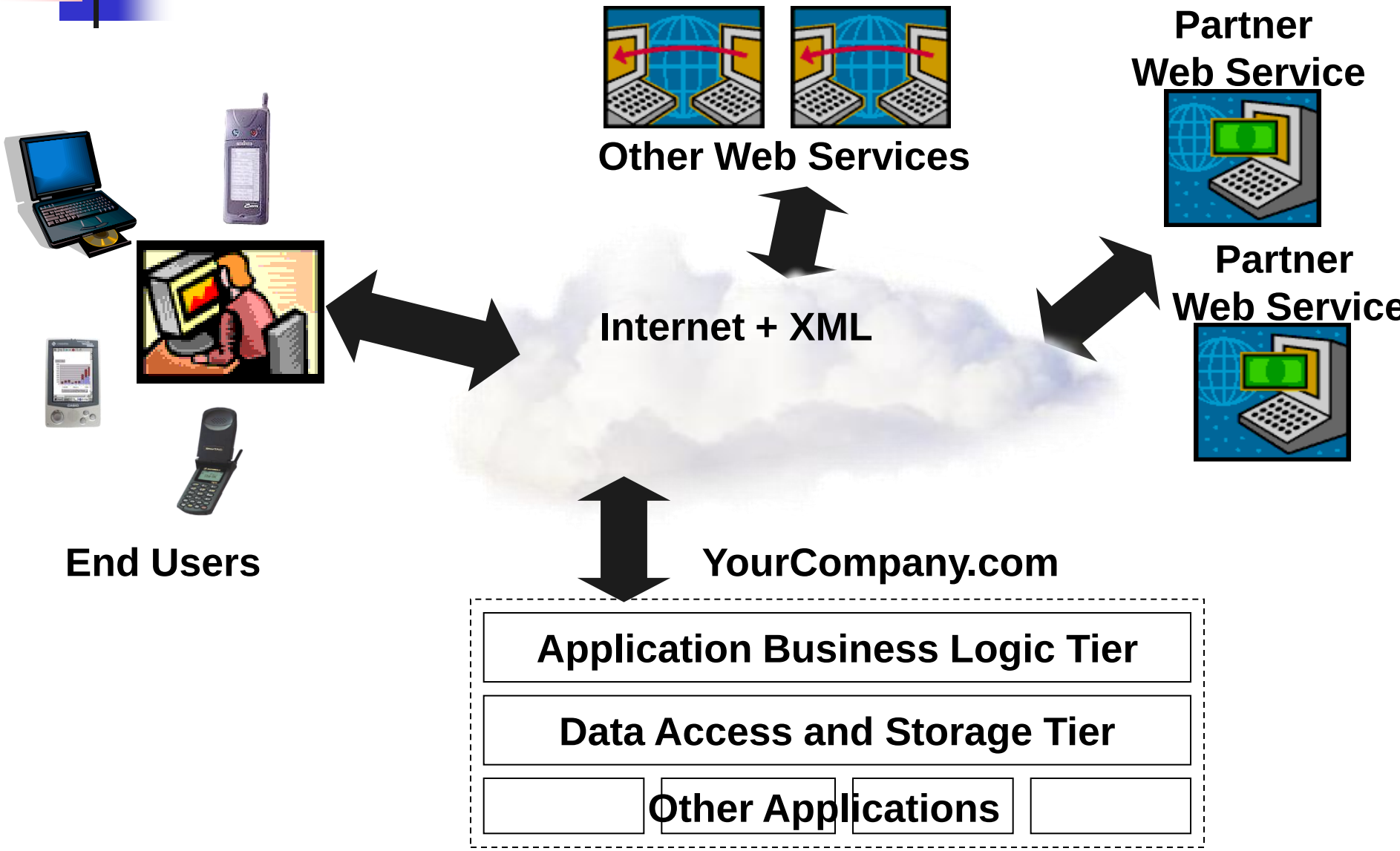


Generation 1
Static HTML

Generation 2
Web Applications

Generation 3
Web Services

Mô hình ứng dụng



Ví dụ về web services

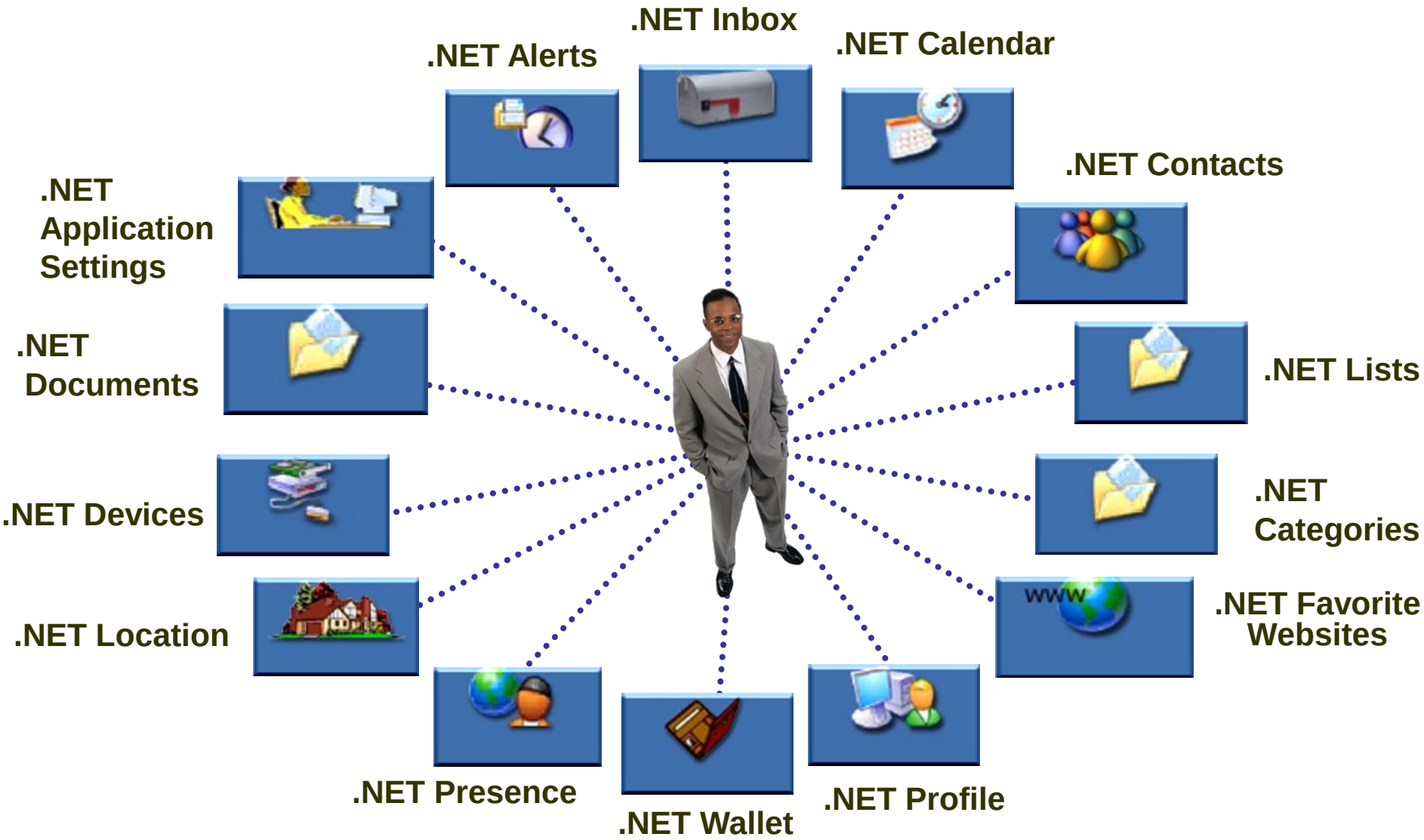
- E-commerce: order sách, đồ dùng văn phòng, sản phẩm khác, …
- Track packages: UPS, FedEx
- Thời tiết
- Bản đồ

Ví dụ về web services

- Google Web Services:
  - Tìm kiếm 1 page
  - Kiểm tra chính tả cho 1 từ
  - Lấy ra cache của 1 web page
  - http://www.google.com/apis

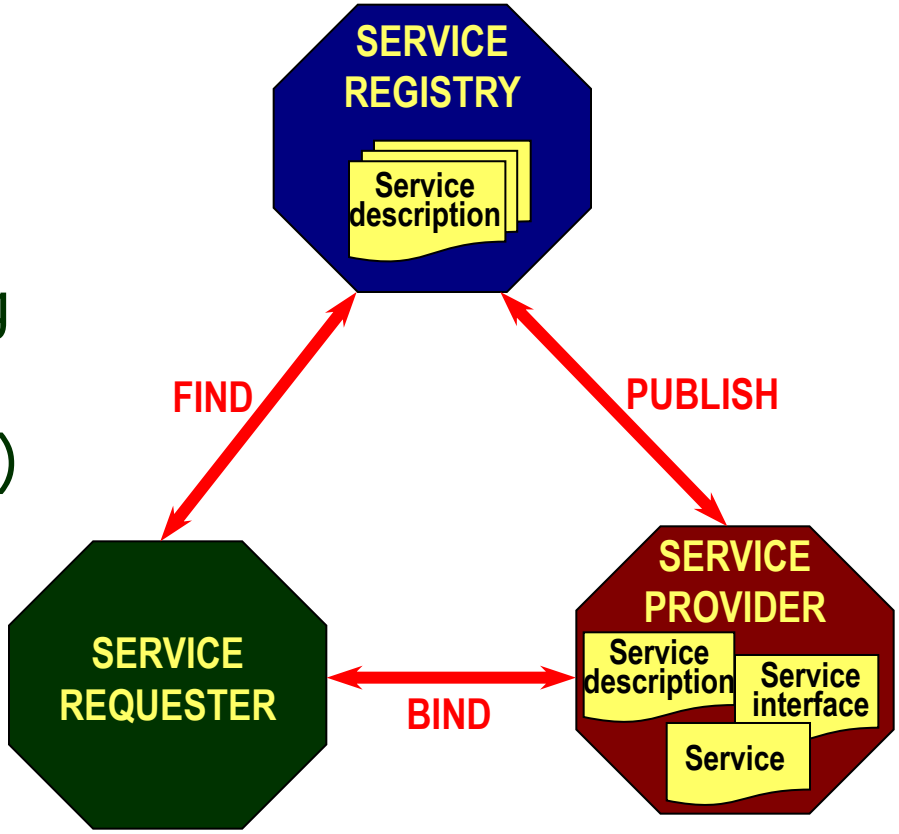
- Amazon Web Services
  - Cho phép các Web sites liên kết với amazon.com và lấy được các khoản phí.
  - Quản lý bên bán hàng, nhà cung cấp third-party
  - www.amazon.com/gp/aws/landing.html

Ví dụ về web services



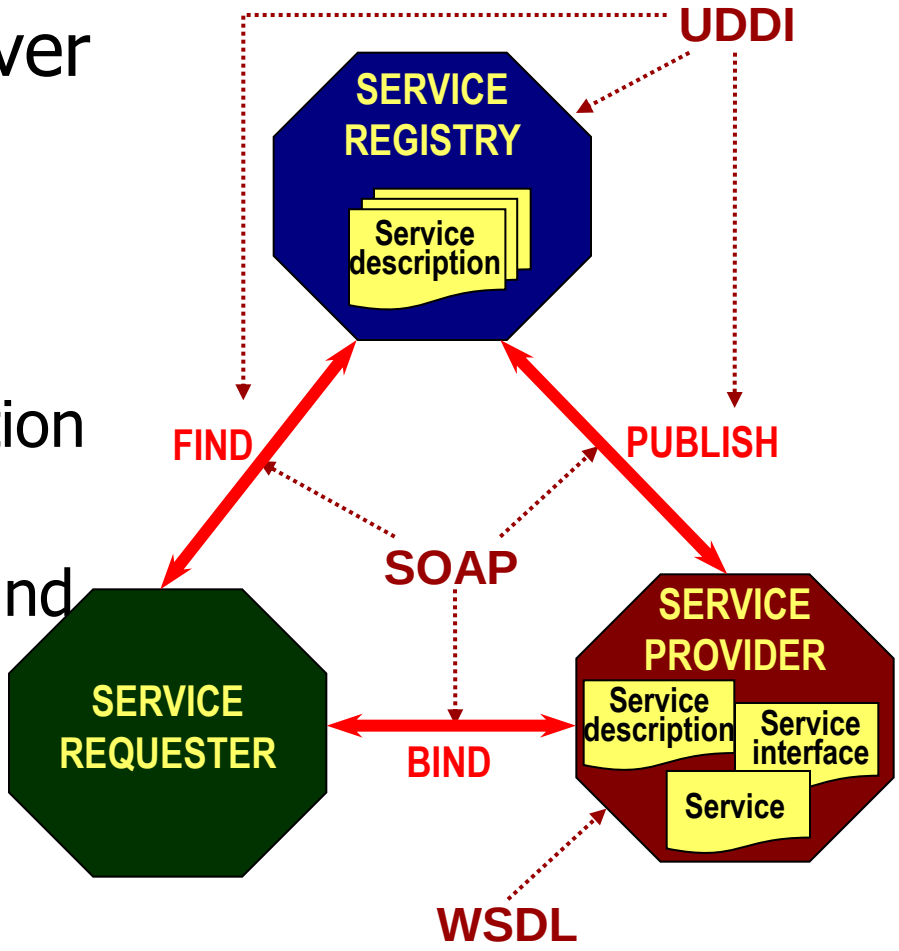
Kiến trúc Web services

- Được khởi xướng bởi IBM; dựa trên 3 thành phần chính:
  - 1. Service **requester** (service client) –Khách hàng của một dịch vụ
  - 2. Service **provider** (server) – tổ chức cài đặt service và cung cấp cho requester
  - 3. Service **registry** – kho liệt kê tất cả các services; cho phép providers quảng bá services của mình, requesters tìm kiếm các services



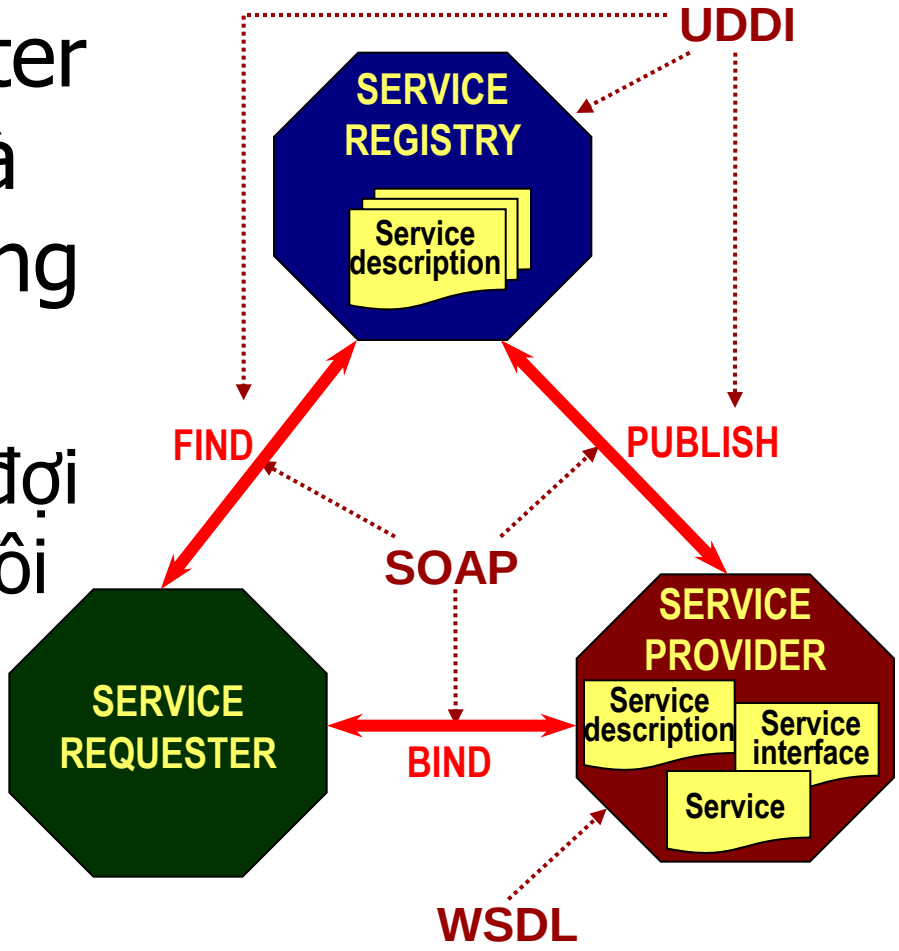
Kiến trúc Web services (2)

- Sử dụng mô hình Client-Server
- Đặc tả của:
  - SOAP (Simple Object Access Protocol)
  - WSDL (Web Services Description Language)
  - UDDI (Universal Description and Discovery Protocol)



Kiến trúc Web services (3)

- Tương tác giữa requester và provider luôn luôn là synchronous, dù sử dụng bất kỳ giao thức nào
  - Service consumer phải đợi đến khi nhận lại phản hồi



Ví dụ 1 scenario sử dụng Web service

Buy Harry Potter (5)

Harry Potter (5)
Price: 20.95
Copies in Stock: 4

How Much?

Book Store

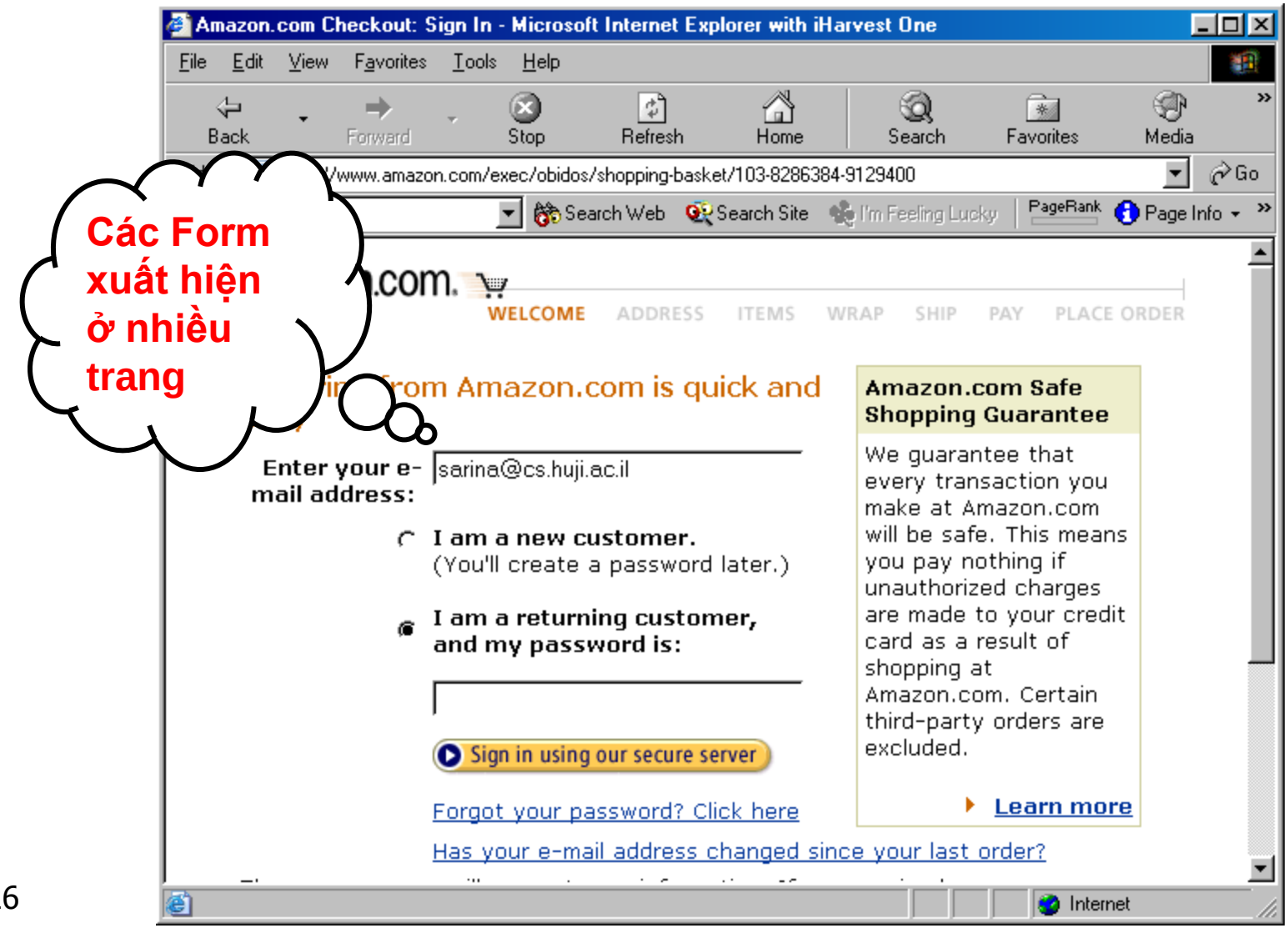
Book Store

20.95

Harry Potter (5)
Price: 25.95
Copies in Stock: 0

Khó khăn khi sử dụng 1 web sites như 1 web service

Khó khăn khi sử dụng 1 web sites như 1 web service (2)-Điền form



Các Form xuất hiện ở nhiều trang

Mong muốn

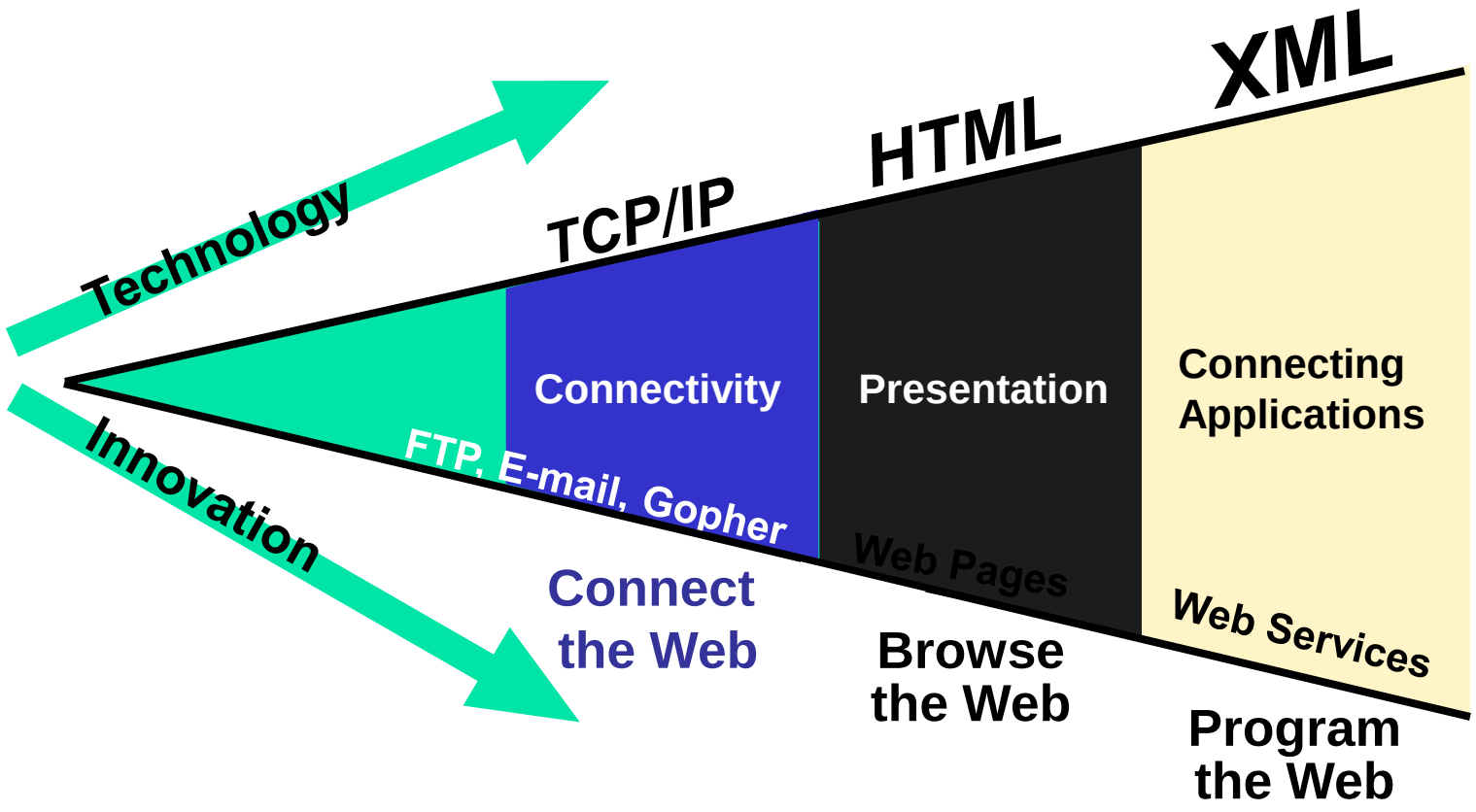
- Gọi các hàm như:
  - Amazon.getPrice("Harry Potter")
  - Amazon.buyBook("Harry Potter", myId)
- Ngôn ngữ chúng ta sử dụng không phụ thuộc vào ngôn ngữ Amazon sử dụng
- Sử dụng các giao thức chuẩn (HTTP, FTP, SNMP)

2. Các công nghệ liên quan

2. Các công nghệ liên quan

- 2.1. XML
- 2.2. SOAP
- 2.3. WSDL
- 2.4. UDDI

2.1. XML là "keo liên kết"



2.1. Sử dụng XML và XML schema

- XML:
  - Dễ dàng mô hình dữ liệu
  - Bao gồm thẻ và dữ liệu
  - Dễ dàng trao đổi thông tin

```
<streetaddress>123 State Street</streetaddress>
```

- XML Schema:
  - Cho phép định nghĩa và tạo ràng buộc cho nội dung của 1 tài liệu XML
  - Cho phép định nghĩa kiểu đơn giản và phức tạp

```
<element name = "streetaddress"
  type="string"/>
```

2.1. Ví dụ 1 tài liệu XML

```
<?xml version="1.0" encoding ="UTF-8" ?>
  <!-- Namespace declarations removed for brevity -->
<portType name="StockQuoteService">
  <operation name="getQuote">
    <input message="impl:getQuoteRequest"
           name="getQuoteRequest"/>
    <output message="impl:getQuoteResponse"
           name="getQuoteResponse"/>
  <operation>
</portType>
```

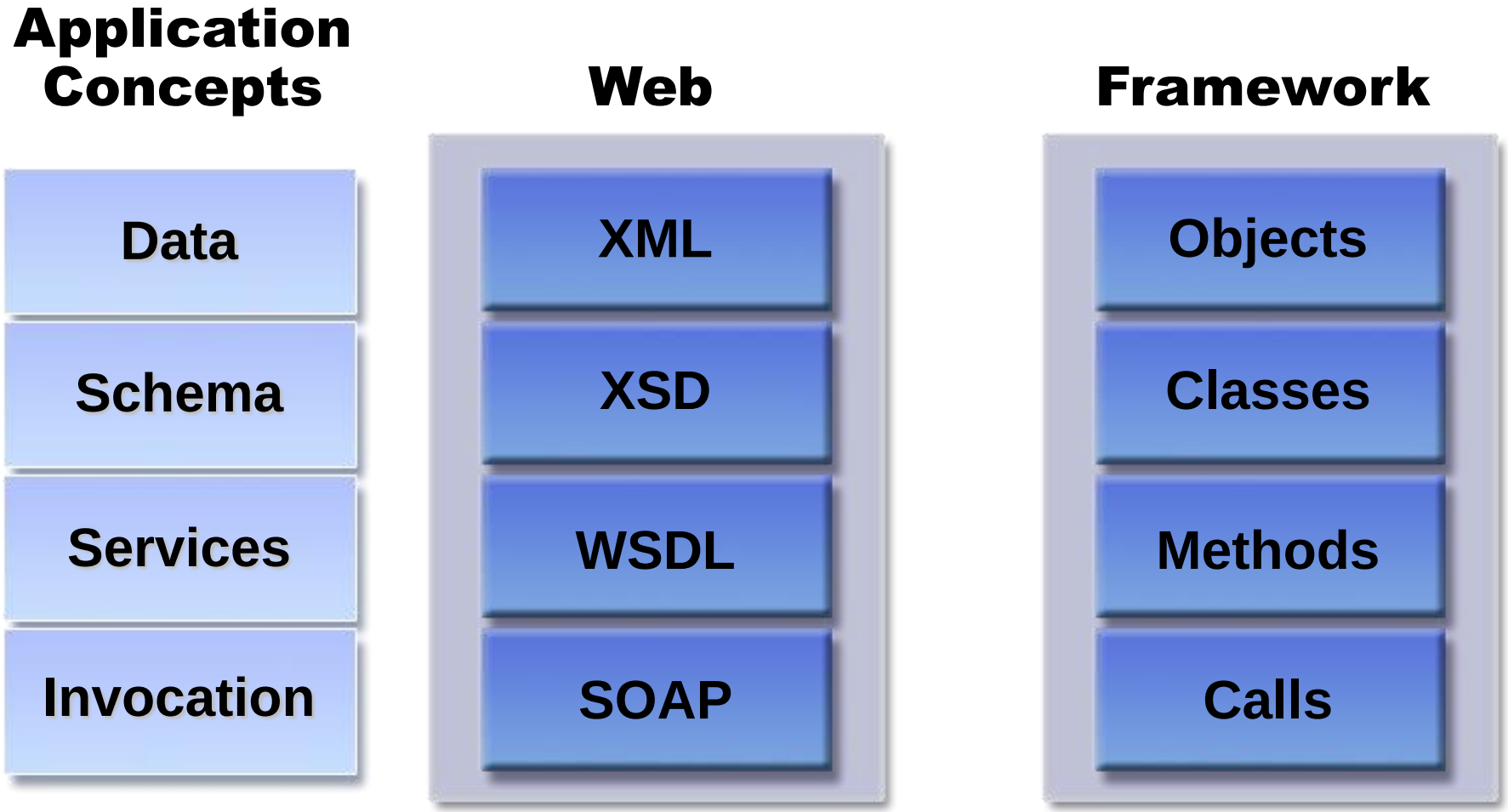
Element

Attribute

XML và Web services

- XML nằm trong 1 phần định nghĩa của Web services
- XML: là nền tảng của rất nhiều chuẩn Web services
- Đặc tính của XML đặc biệt phù hợp với các yêu cầu tích hợp dịch vụ

XML - Web Services



2.2. SOAP

Web Service Wire Format

- Web Service Wire Format: chỉ ra cách thức trao đổi các thông điệp
  - HTTP-GET
  - HTTP-POST
  - SOAP
- HTTP-GET và HTTP-POST
  - Hỗ trợ hạn chế các kiểu dữ liệu
- SOAP cung cấp giao diện HTTP/XML mạnh mẽ hơn
  - Mở rộng hỗ trợ các kiểu dữ liệu

SOAP là gì?

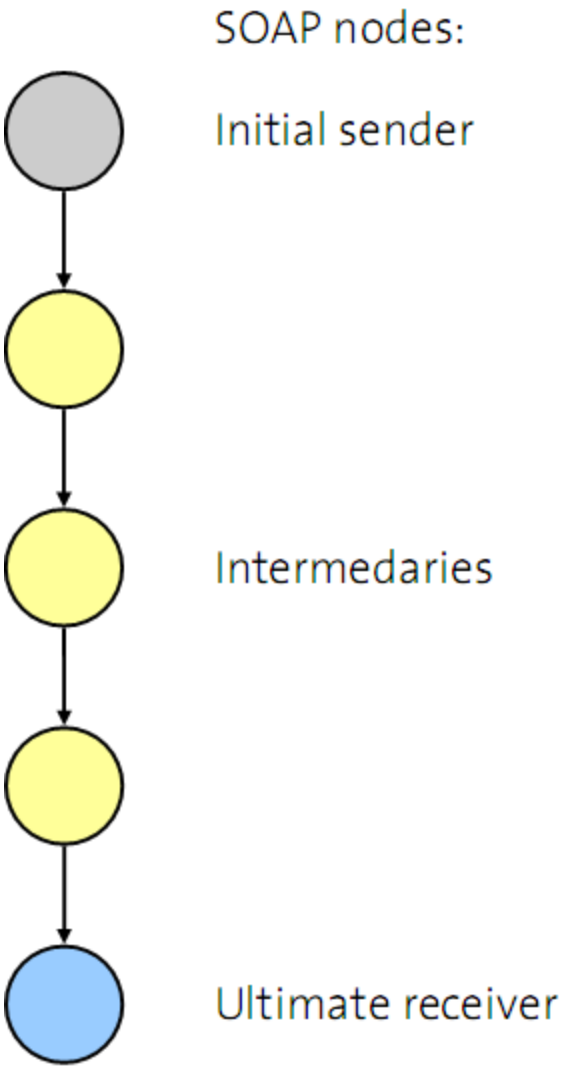
- SOAP: Simple Object Access Protocol
- Giao thức hạng nhẹ dùng để chuyển giao thông điệp trong môi trường phân tán.
- Được sử dụng cho "Remote Procedure Calls"
  - IIOP (cho Corba), ORPC (cho DCOM), RMI (cho Java)
- W3C
  - SOAP : XML – phương tiện mang – HTTP: kênh vận chuyển (Ngoại lệ: SMTP, SIP..)

SOAP là gì? (2)

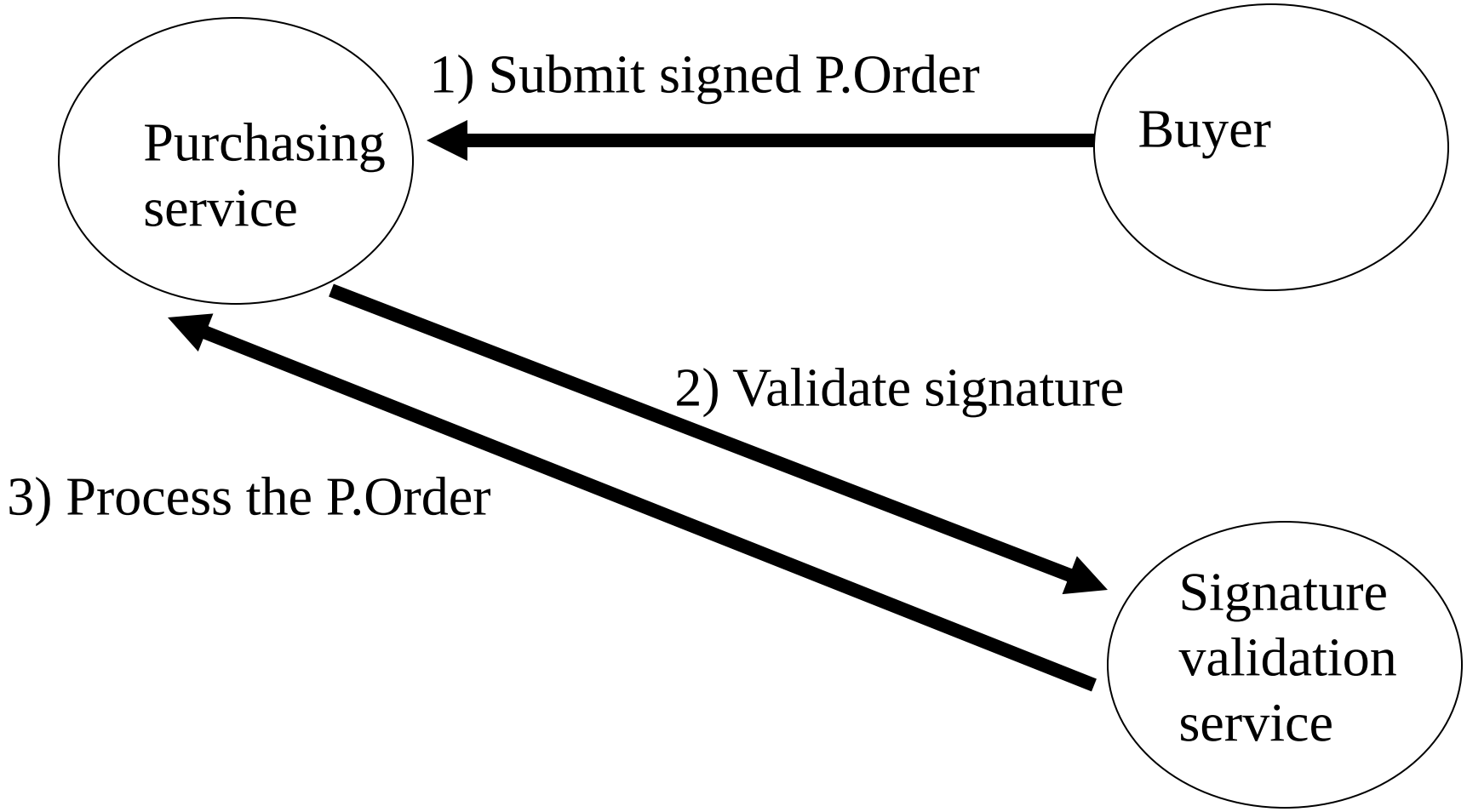
- Đặc tả SOAP định nghĩa:
  - Định dạng thông điệp SOAP
  - Mô tả về cách thức 1 SOAP message được truyền trên HTTP (cho Web based interaction) hoặc SMTP (cho e-mail based interaction)
  - Tập các luật xử lý các SOAP message.
  - Tập các quy ước:
    - Chuyển 1 lời gọi RPC thành 1 thông điệp SOAP & ngược lại
    - Cách thức thực thi tương tác RPC

Đường truyền thông điệp SOAP (SOAP message path)

- 1 thông điệp SOAP có thể truyền qua nhiều phần trung gian từ điểm gửi tới nơi nhận cuối cùng
- Các thành phần liên quan trong việc truyền thông điệp gọi là các SOAP nodes
- Các SOAP intermediaries sẽ forward các thông điệp và có thể xử lý chúng
- Mỗi node có 1 role nhất định, quyết định cách xử lý thông điệp tại node đó



Ví dụ về Intermediary



Thông điệp SOAP - Purchase order

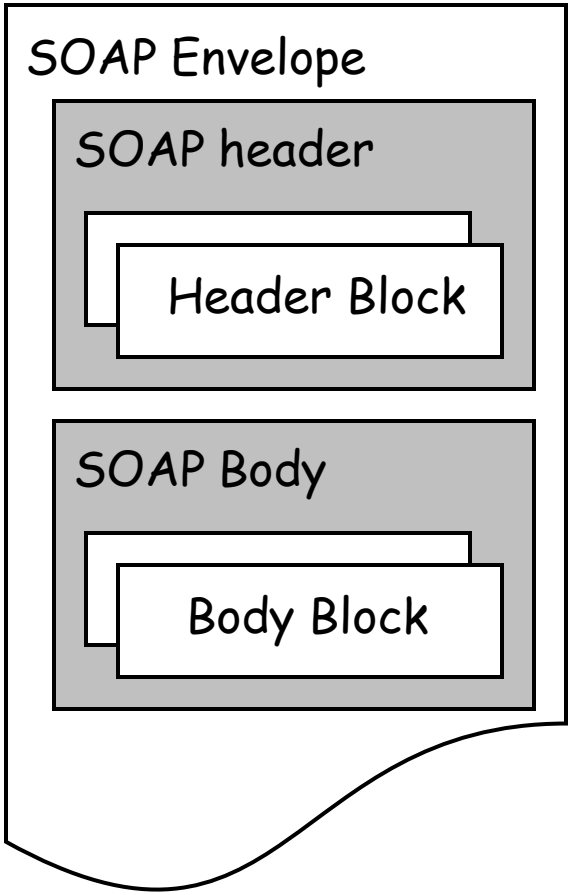
```
<s:Envelope xmlns:s=http://www.w3.org/2001/06/soap-envelope>
  <s:Header>
    <m:transaction xmlns:m="soap-transaction" s:mustUnderstand="true">
      <transactionID>1234</transactionID>
    </transaction>
  </Header>
  <s:Body>
    <n:purchaseOrder xmlns:n="urn:OrderService">
      <from><person>Christopher Robin</person>
        <dept>Accounting</dept></from>
      <to><person>Pooh Bear</person>
         <dept>Honey></dept></to>
      <order><quantity>1</quantity>
        <item>Pooh Stick</item></order>
    </n:purchaseOrder>
  </s:Body>
</s:Envelope>
```

Ưu điểm của SOAP

- Giúp tăng tính liên tác hệ thống
- Sử dụng HTTP – nâng cao tính tương thích và co dãn.
- Linh hoạt cho việc phát triển nhờ các thuộc tính của XML
- SOAP: cung cấp cách thức giao tiếp giữa các applications trên:
  - Các hệ điều hành khác nhau
  - Các công nghệ khác nhau
  - Các ngôn ngữ lập trình khác nhau

Thông điệp SOAP

- Giao thức SOAP: thực hiện dựa trên việc trao đổi các message
- Các messages: như các phong bì, ứng dụng đính dữ liệu vào và gửi đi
- Một message gồm 2 phần chính, mỗi phần có thể gồm nhiều blocks:
  - header: tùy chọn
  - body: bắt buộc
- Vai trò:
  - body: Dữ liệu mức ứng dụng
  - header: Dữ liệu mức hạ tầng mạng

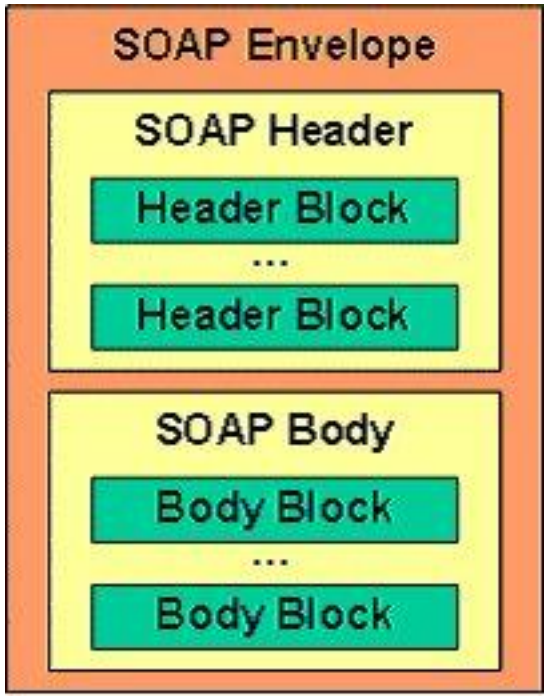


Các thành phần thông điệp

- Envelope (bắt buộc)
  - Phần tử ở đỉnh tài liệu XML biểu diễn thông điệp.
- Header (tùy chọn)
  - Xác định cách thức nơi nhận thông điệp SOAP xử lý thông điệp.
  - Bổ sung các tính năng: chứng thực, quản lý giao dịch, định tuyến, …
- Body (bắt Buộc)
  - Thông tin nội dung
  - Sử dụng cho RPC và thông báo lỗi.

Cấu trúc thông điệp SOAP biểu diễn dưới dạng XML

```
<?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:Header>
      ...
    </soap:Header>
    ...
    <soap:Body>
      ...
            <soap:Fault>
              ...
            </soap:Fault>
    </soap:Body>
</soap:Envelope>
```



Ví dụ

```
<Envelope>
    <Header>
        <transId>345</transId>
    </Header>
    <Body>
        <Add>
            <n1>3</n1>
            <n2>4</n2>
        </Add>
    </Body>
</Envelope>
```

c = Add(n1, n2)

SOAP Request

```
<SOAP-ENV:Envelope
    xmlns:SOAP-ENV=“http://schemas.xmlsoap.org/soap/envelope/”
    SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/”>
    <SOAP-ENV:Header>
        <t:transId xmlns:t=“http://a.com/trans”>345</t:transId>
    </SOAP-ENV:Header>
    <SOAP-ENV:Body>
        <m:Add xmlns:m=“http://a.com/Calculator”>
            <n1>3</n1>
            <n2>4</n2>
        </m:Add>
    </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

SOAP Request

namespace mô tả SOAP envelope

```
<SOAP-ENV:Envelope
   xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
   SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
   ...etc...
</SOAP-ENV:Envelope>
```

Thiết lập kiểu mã hóa

SOAP Request

```
...etc...
<SOAP-ENV:Header>
   <t:transId xmlns:t="http://a.com/trans">1234</t:transId>
</SOAP-ENV:Header>
<SOAP-ENV:Body>
   <m:Add xmlns:m="http://a.com/Calculator">
      <n1>3</n1>
      <n1>4</n2>

</SOAP-ENV:Body>
...etc...
```

Định nghĩa phương thức

SOAP Response

```
<SOAP-ENV:Envelope
   xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
   SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
   <SOAP-ENV:Header>
      <t:transId xmlns:t="http://a.com/trans">345</t:transId>
   </SOAP-ENV:Header>
   <SOAP-ENV:Body>
      <m:AddResponse xmlns:m="http://a.com/Calculator">
         <result>7</result>
      </m:AddResponse>
   </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Tên phương thức với hậu tố "Response" appended

Ví dụ 1 request loại kết nối SOAP HTTP

```
POST /InStock HTTP/1.1
Host: www.stock.org
Content-Type: text/xml; charset="utf-8"
Content-Length: nnn
SOAPAction: "Some-URI"

<?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 xmlns:m="http://www.stock.org/stock">
    <m:GetStockPrice>
      <m:StockName>IBM</m:StockName>
    </m:GetStockPrice>
  </soap:Body>
</soap:Envelope>
```

Chú ý: có dòng trống

Ví dụ 1 response loại kết nối SOAP HTTP

```
HTTP/1.1 200 OK
Content-Type: text/xml; charset="utf-8"
Content-Length: nnn

<?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 xmlns:m="http://www.stock.org/stock">
    <m:GetStockPriceResponse>
      <m:Price>34.5</m:Price>
    </m:GetStockPriceResponse>
  </soap:Body>

</soap:Envelope>
```

Chú ý: có dòng trống

SOAP Header

- Header: chứa các thông tin độc lập với ứng dụng (application có thể không xử lý header)

- Mục đích sử dụng chính của header là: thông tin điều phối, định danh (ví dụ trong các giao dịch transactions), thông tin security (ví dụ các certificates), …

Ví dụ actor header

```
<s:Envelope xmlns:s=http://www.w3.org/2001/06/soap-envelope>
  <s:Header>
    <x:signature actor="uri:SignatureVerifier">
      …
    </x:signature>
  <s:Body>
    <abc:purchaseOrder>…</abc:purchaseOrder>
  </s:Body>
</s:Envelope>
```

SOAP Header (2)

Soap header cung cấp cơ chế để chỉ ra:

- Ai sẽ làm việc với 1 header block (thuộc tính SOAP **actor**) – có thể là:
  - **None** – Truyền thông tin không cần xử lý.
  - **Next** – node nhận thông có thể xử lý block đó.
  - **ultimateReceiver** – chỉ ra rằng header dành cho node nhận message cuối cùng
- Làm gì với 1 header block (thuộc tính ***mustUnderstand***):
  - Có giá trị ***true*** hoặc ***false***
  - Nếu 1 node có thể xử lý 1 message (thuộc tính **actor**), thuộc tính **mustUnderstand** quyết định việc có bắt buộc xử lý hay không

SOAP Header (3) - Xử lý thông điệp

- SOAP chỉ ra cách thức xử lý các thông điệp
  - Mỗi SOAP node trên message path sẽ lấy các role kết hợp với từng phần của thông điệp
  - Có 3 roles chuẩn tắc: none, next, hoặc ultimateReceiver; nhưng các ứng dụng có thể tự định nghĩa và sử dụng các role của chúng
  - role quyết định ai chịu trách nhiệm với từng phần của 1 message
- Nếu 1 block không chỉ ra role, mặc định sẽ là ultimateReceiver
- Nếu thuộc tính mustUnderstand là *true*, node ứng với role chỉ ra phải xử lý phần tương ứng của message, hoặc phải sinh ra lỗi và không forward tiếp message
- SOAP 1.2 có thêm thuộc tính relay. (Chỉ ra một node phải forward nếu không xử lý phần tương ứng đó của message)

Node	intermediary	ultimate receiver
mustUnderstand		
"true"	must process	must process
"false"	may process	may process
absent	may process	may process

SOAP Body

- Chứa các dữ liệu cụ thể cho ứng dụng trong 1 message
- Một body entry (hay body block) có cú pháp tương tự với 1 header entry, với thuộc tính role = ultimateReceiver và mustUnderstand = 1
- Khác với headers, SOAP chỉ rõ nội dung của 1 body entries:
  - Biểu diễn thống nhất các RPC requests & responses bằng 1 tập các SOAP body block
  - Fault entry
    - Chứa thông tin về lỗi khi xử lý 1 SOAP message
    - Để nhận biết 1 SOAP message có mang thông tin lỗi, SOAP message đó chứa 1 phần tử SOAP fault duy nhất-con duy nhất của SOAP body

SOAP Body (2)

- Fault Entry: có 4 thành phần (SOAP 1.1)
  - fault code: chỉ ra loại lỗi (4)
  - fault string: phần giải thích về lỗi dành cho con người (không được xử lý tự động)
  - fault actor: ai gây ra lỗi
  - detail: thông tin chi tiết của ứng dụng về lỗi
- Fault code:
  - Version mismatch: namespace không đúng trong SOAP envelope
  - Must Understand: không hiểu được header block với thuộc tính mustunderstand được thiết lập "true"
  - Client: không hiểu được message (định dạng hoặc nội dung)
  - Server: có vấn đề với server, khiến message không thể được xử lý

Ví dụ (SOAP 1.1)

```
<SOAP-ENV:Envelope
   xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
   SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
   <SOAP-ENV:Body>
      <SOAP-ENV:Fault>
         <faultcode>SOAP-ENV:Server</faultcode>
         <faultstring>Internal Application Error</faultstring>
         <detail xmlns:f="http://www.a.com/CalculatorFault">
            <f:errorCode>794634</f:errorCode>
            <f:errorMsg>Divide by zero</f:errorMsg>
         </detail>
      </SOAP-ENV:Fault>
   </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

SOAP Body (3)

- Fault entry (SOAP 1.2)
  - Bắt buộc có 2 thành phần:
    - Code: chứa 1 giá trị Mã lỗi & có thể có subcode (dành cho thông tin cụ thể của ứng dụng)
    - Reason: Như fault string trong SOAP 1.1
  - Và 1 số thành phần tùy chọn
    - Details: Như trong SOAP 1.1
    - Node: Định danh của node sinh ra lỗi (Mặc định sẽ là node cuối trong message path)
    - Role: vài trò của node sinh ra lỗi

Ví dụ về phần tử Fault (SOAP1.2)

```
<?xml version='1.0' ?>
<env:Envelope xmlns:env="http://www.w3.org/2003/05/soap-envelope"
        xmlns:rpc='http://www.w3.org/2003/05/soap-rpc'>
 <env:Body>
  <env:Fault>
   <env:Code>
    <env:Value>env:Sender</env:Value>
    <env:Subcode>
     <env:Value>rpc:BadArguments</env:Value>
    </env:Subcode>
   </env:Code>
   .....
```

Ví dụ về phần tử Fault (2)

```
....
<env:Reason>
 <env:Text xml:lang="en-US">Processing error</env:Text>
 <env:Text xml:lang="bg">Грешка при обработка</env:Text>
</env:Reason>
<env:Detail>
 <e:myFaultDetails
      xmlns:e="http://travelcompany.example.org/faults">
  <e:message>Name does not match card number</e:message>
  <e:errorcode>999</e:errorcode>
 </e:myFaultDetails>
</env:Detail>
</env:Fault>
</env:Body>
</env:Envelope>
```

SOAP RPC

RPC Request

SOAP Envelope

SOAP header

Transactional context

SOAP Body

Name of Procedure

Input param 1

Input param 2

RPC Response (one of the two)

SOAP Envelope

SOAP header

Transactional context

SOAP Body

Return parameter

SOAP Envelope

SOAP header

Transactional context

SOAP Body

Fault entry

SOAP RPC (2)-Ví dụ

- Request:

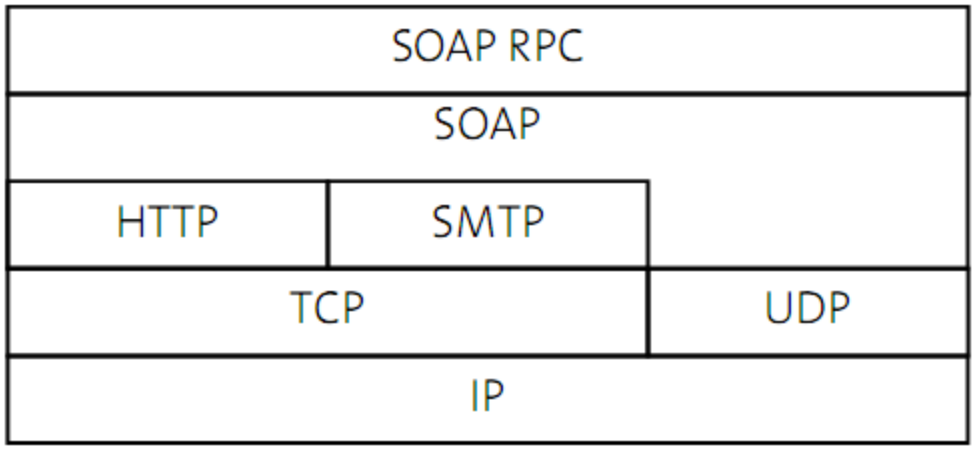
```
<SOAP-ENV:Body>
  <m:GetLastTradePrice xmlns:m="Some-URI">
    <symbol>DIS</symbol>
  </m:GetLastTradePrice>
</SOAP-ENV:Body>
```

- Response:

```
<SOAP-ENV:Body>
  <m:GetLastTradePriceResponse xmlns:m="Some-URI">
    <Price>34.5</Price>
  </m:GetLastTradePriceResponse>
</SOAP-ENV:Body>
```

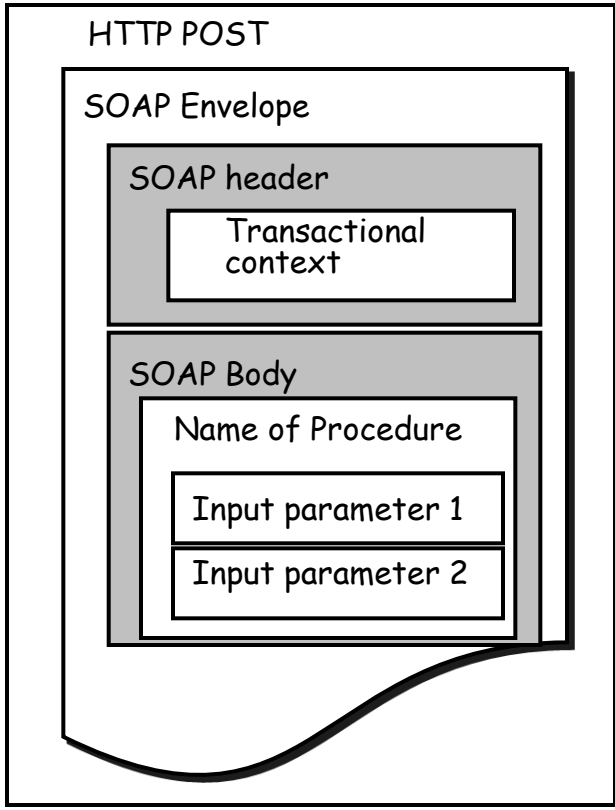
Liên kết SOAP với giao thức

- Thông điệp SOAP có thể được truyền với bất kỳ giao thức nào
- Binding SOAP với 1 giao thức truyền thông: chính là mô tả cách truyền 1 thông điệp SOAP trên giao thức đó



SOAP & HTTP

- Loại kết nối phổ biến của SOAP là HTTP. Ở đó, SOAP có thể sử dụng phương thức GET hoặc POST:
  - GET: request không phải là 1 SOAP message nhưng response là 1 SOAP Message
  - POST: cả request và response đều là SOAP messages
- SOAP sử dụng cùng loại lỗi & các mã status như HTTP →HTTP responses có thể được xử lý trực tiếp bởi một SOAP module



SOAP-RPC trên HTTP POST Request

```
POST /Reservations HTTP/1.1
Host: travelcompany.example.org
Content-Type: application/soap+xml; charset="utf-8"
Content-Length: nnnn
<?xml version='1.0' ?>
<env:Envelope xmlns:env="http://www.w3.org/2003/05/soap-
   envelope" >
 <env:Header>
  <t:transaction
   xmlns:t=http://thirdparty.example.org/transaction
   env:encodingStyle="http://example.com/encoding"
   env:mustUnderstand="true" >5
  </t:transaction>
 </env:Header>
 ......
```

SOAP-RPC trên HTTP POST Request (2)

```
.....
 <env:Body>
  <m:chargeReservation
    env:encodingStyle="http://www.w3.org/2003/05/soap-encoding"
        xmlns:m="http://travelcompany.example.org/">
   <m:reservation
    xmlns:m="http://travelcompany.example.org/reservation">
        <m:code>FT35ZBQ</m:code>
   </m:reservation>
    <o:creditCard xmlns:o="http://mycompany.example.com/financial">
    <n:name xmlns:n="http://mycompany.example.com/employees"> Åke
    Jógvan Øyvind </n:name>
    <o:number>123456789099999</o:number>   <o:expiration>2005-
    02</o:expiration>
    </o:creditCard>
   </m:chargeReservation>
  </env:Body>
</env:Envelope>
```

SOAP-RPC trên HTTP POST Request (3) Response OK

```
HTTP/1.1 200 OK
Content-Type: application/soap+xml; charset="utf-8"
Content-Length: nnnn

<?xml version='1.0' ?>
<env:Envelope xmlns:env="http://www.w3.org/2003/05/soap-
   envelope" >
 <env:Header>
      ...
      ...
 </env:Header>
 <env:Body>
      ...
      ...
 </env:Body>
</env:Envelope>
```

SOAP-RPC trên HTTP POST Request (4)
Response Bad

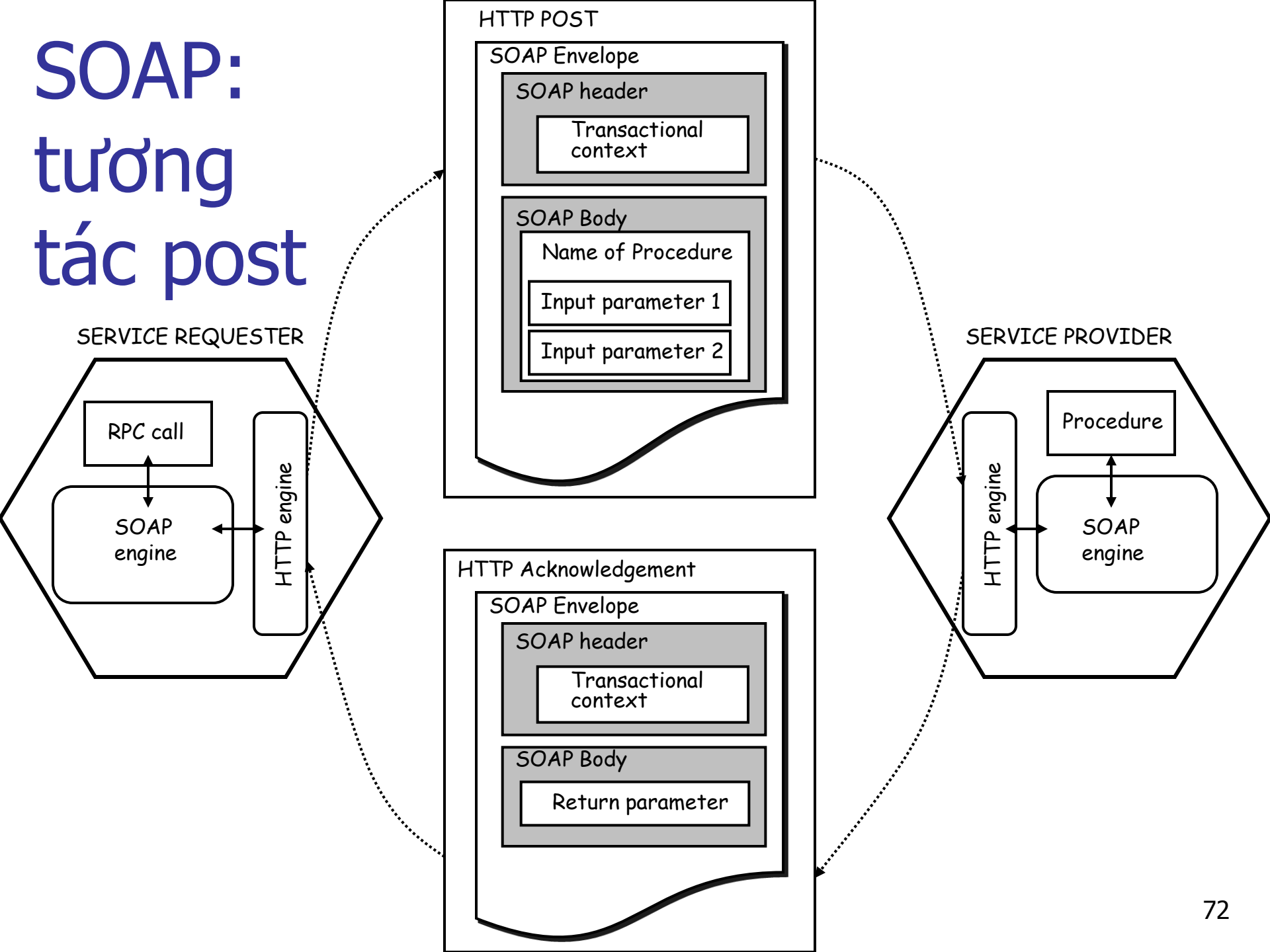
```
HTTP/1.1 500 Internal Server Error
Content-Type: application/soap+xml; charset="utf-8"
Content-Length: nnnn

<?xml version='1.0' ?>
<env:Envelope xmlns:env="http://www.w3.org/2003/05/soap-
   envelope">
 <env:Body>
  <env:Fault>
   <env:Code>
    <env:Value>env:Sender</env:Value>
    <env:Subcode>
   <env:Value>rpc:BadArguments</env:Value>
   </env:Subcode>
   </env:Code>
   ....
```

SOAP-RPC trên HTTP POST Request (5) Response Bad (tiếp)

(cont.)

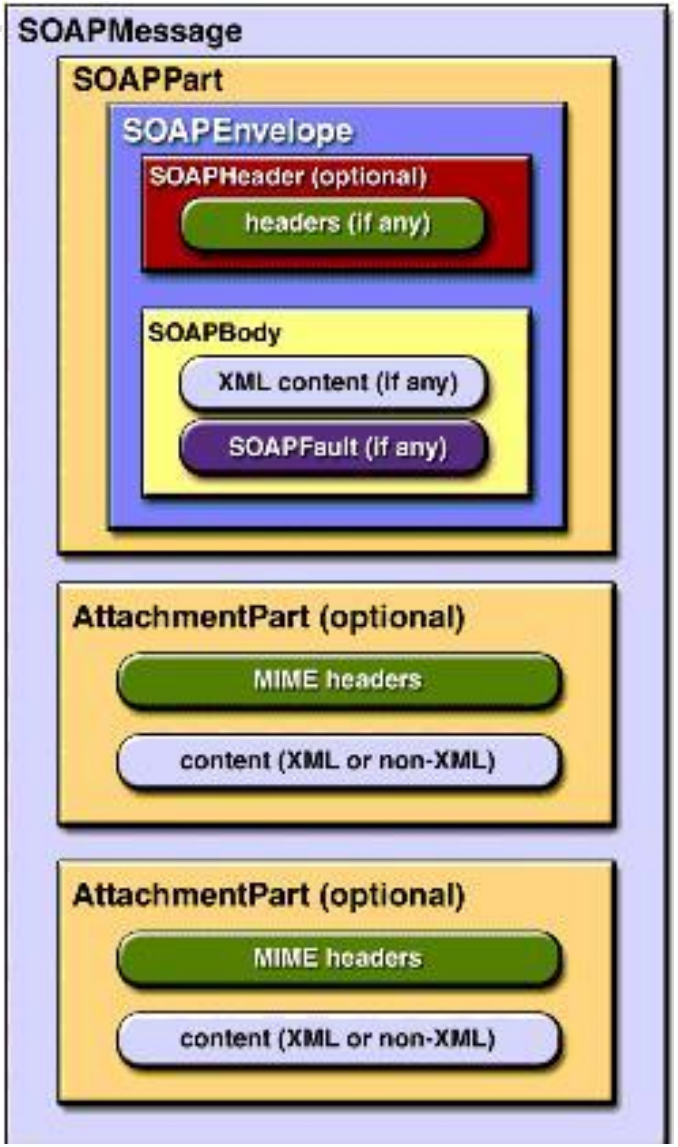
```
……
    <env:Reason> <env:Text xml:lang="en-US">Processing
   error</env:Text> </env:Reason>
    <env:Detail>
     <e:myFaultDetails
       xmlns:e="http://travelcompany.example.org/faults" >
       <e:message>Name does not match card
   number</e:message>
       <e:errorcode>999</e:errorcode>
     </e:myFaultDetails>
    </env:Detail>
  </env:Fault>
 </env:Body>
</env:Envelope>
```

SOAP: tương tác post



SOAP với Attachment

- Một thông điệp SOAP có thể có 1 hoặc nhiều attachments.
- Mỗi AttachmentPart có 1 **MIME** header chỉ ra loại dữ liệu mà nó chứa.
- Khi 1 SOAPMessage có nhiều AttachmentPart, thành phần SOAPPart có thể chứa hoặc không nội dung thông điệp.
- http://www.w3.org/TR/2002/WD-soap12-af-20020924/



Các gói phát triển phần mềm với SOAP

- Apache SOAP cho Java
- Apache Axis cho Java
- Idoox WASP cho C++
- Microsoft SOAP Toolkit (là một thành phần trong .net framework)
- SOAP::Lite cho Perl
- http://oreilly.com/catalog/progwebsoap/chapter/ch03.html

Ví dụ: Cài đặt 1 SOAP Processor

Java Reflection

```
// Không dùng reflection
Foo foo = new Foo ();
foo.hello ();

// Dùng reflection
Class cl = Class.forName ("Foo");
Method method = cl.getMethod ("hello",
  null);
method.invoke (cl.newInstance (), null);
```

SOAP processor

- SOAP processor: Xử lý các SOAP message
- Cách thức cài đặt bộ SOAP Processor?

- Sử dụng Java reflection để cài đặt minh họa 1 SOAP Processor
  - Java reflection: lấy ra các class/method/field của 1 đối tượng

SOAP processor

- Trong SOAP, thông tin về RPC được đóng gói trong tài liệu XML
- Trong ví dụ này, thông tin chi tiết nằm trong các tham số của HTTP request
  - Cho phép user gọi bất kỳ phương thức nào của 1 lớp bất kỳ (Giả thiết phương thức chỉ có các tham số String)
- Tất cả cho chúng ta hiểu hơn về cách thức hoạt động của SOAP processor

Lấy các thông tin (về invocation) từ Client (HTML Form)

```
<form name="info" method="get"
   action="servlet/SOAPImitation">
Class Name: <input type="text" name="class"><br>
Method Name: <input type="text" name="method"><br>
<input type="button" value="Set Number of Arguments"
   onClick="setArguments()">
 <input type="hidden" name="paramNum" value="0">
```

```
<p id = "argumentP"></p>
<input type="submit">
</form>
```

Lấy các thông tin (về invocation) từ Client (Java script)

```
function setArguments() {
  num = prompt("Enter the number of arguments to the method", 0);
  p = document.getElementById("argumentP");
  str = ""
  for (i = 0 ; i < num ; i++) {
    str += "Argument " + i + ": " +
            "<input type='text' name='param"  + i + "'><br>";
  }
  p.innerHTML=str;
  info.paramNum.value=num
}
```

SOAP Processor-Servlet

```
import java.io.*; import javax.servlet.*; import javax.servlet.http.*;
import java.lang.reflect.*;

public class SoapImitation extends HttpServlet {

    public void doGet(HttpServletRequest req,
                      HttpServletResponse res)
        throws ServletException, IOException  {
    res.setContentType("text/html");
    PrintWriter out = res.getWriter();

    try{
        Class objClass = Class.forName(req.getParameter("class"));
        int numParams=
          Integer.parseInt(req.getParameter("paramNum"));
```

```
    Class[] paramClasses = new Class[numParams];
    Object[] paramObjs = new Object[numParams];
    for (int i = 0 ; i < numParams ; i++) {
      paramClasses[i] = Class.forName("java.lang.String");
      paramObjs[i] = req.getParameter("param" + i);
    }

    Method method =
      objClass.getMethod(req.getParameter("method"),
                                    paramClasses);
    Object result = method.invoke(objClass.newInstance(),
                                          paramObjs);
    out.println("<HTML><BODY><h1>Method Result</h1>" +
                  result + "</BODY></HTML>");
  } catch (Exception e) {
    out.println("<HTML><BODY>Error!</BODY></HTML>");
  }  }  }
```

Method Invocation Page - Microsoft Internet Explorer with iHarvest One

File Edit View Favorites Tools Help

Back Forward Stop Refresh Home Search Favorites

Address http://pita:8080/soap-test/MethodInvocation.html Go

Google Search Web Search Site I'm Feeling Lucky PageRank

Imitating SOAP Processor!

This form enables your to specify a class, a method of the class and any number of string arguments. You will see the result of invoking the method of the class on the arguments.

Class Name: hello.HelloServer

Method Name: sayHelloTo

Set Number of Arguments

Argument 0: John

Submit Query

Done Local intranet

http://pita:8080/soap-test/servlet/SoapImitation?class=hello.HelloServer&method=sayHe...

File Edit View Favorites Tools Help

Back Forward Stop Refresh Home Search Favorites

Address http://pita:8080/soap-test/servlet/SoapImitation?class=hello.HelloServer&method=sayHelloTo& Go

Google Search Web Search Site I'm Feeling Lucky PageRank

Method Invocation Result

Hello John, How are you doing?

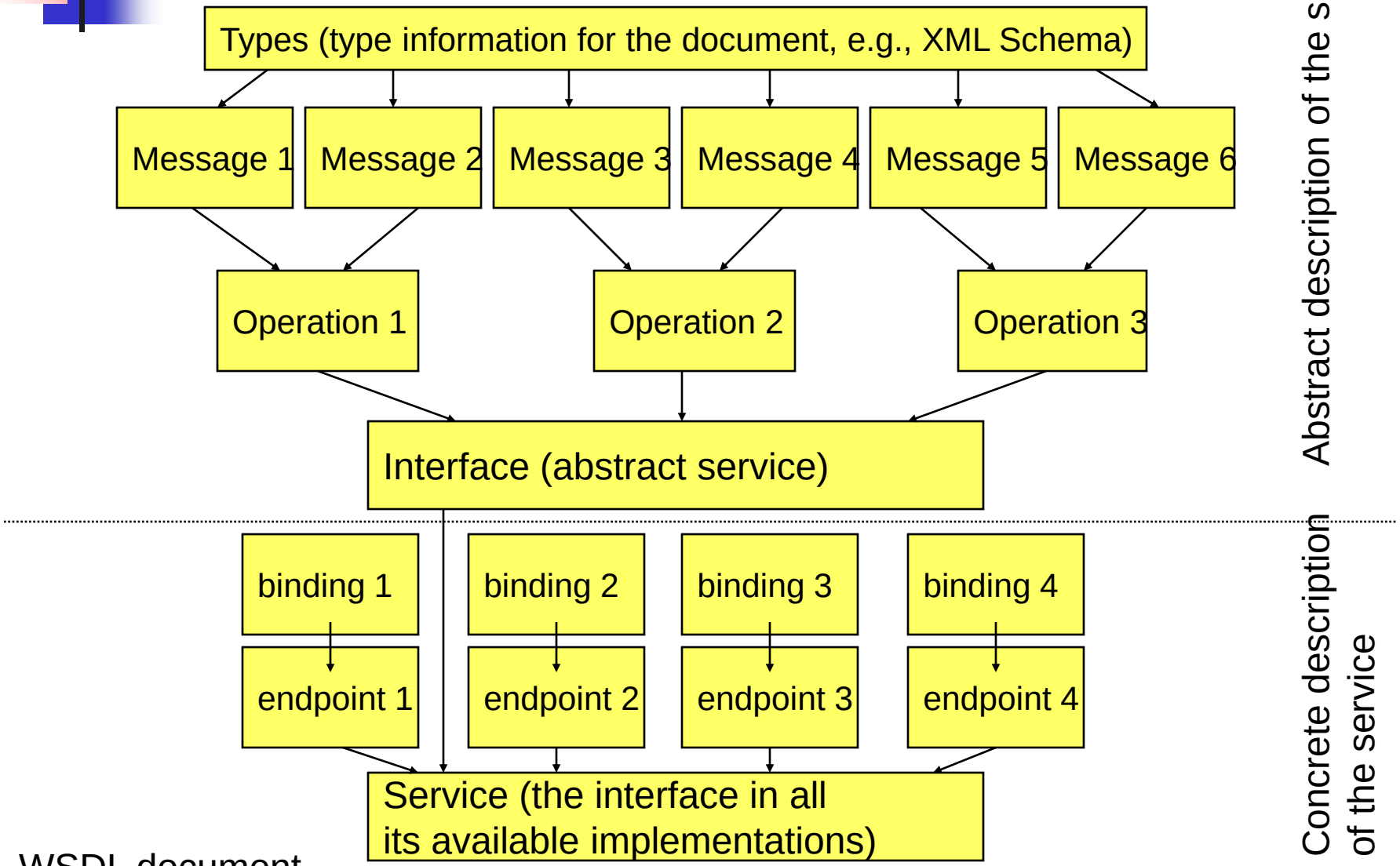
Done Local intranet

2.3. WSDL

WSDL: Web Service Description Language

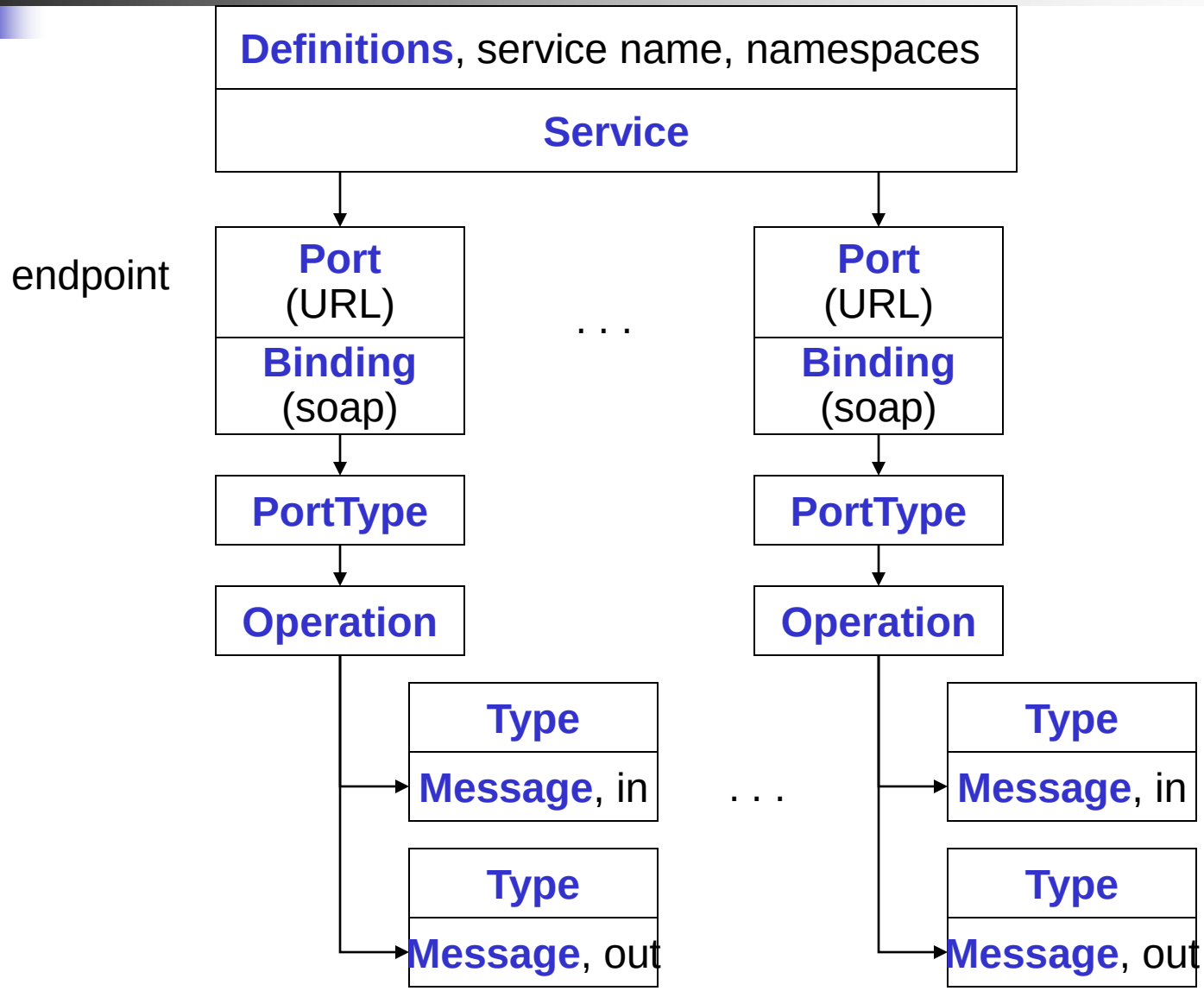
- WSDL được sử dụng để mô tả các Web services, bao gồm 4 phần chính:
  - Chức năng của dịch vụ trong bộ thuật ngữ chuẩn;
  - Các kiểu dữ liệu của tham số và giá trị trả về của các hàm (service);
  - Thông tin kết nối (Binding information) về giao thức truyền thông được sử dụng, thường là SOAP;
  - Thông tin địa chỉ (Address information) xác định cho 1 service cụ thể.
- 3 thành phần cuối có thể được sinh tự động.
- Web services được mô tả bằng WSDL có thể được tìm kiếm theo yêu cầu.
- Web services được mô tả bằng WSDL cung cấp chi tiết lời gọi từ xa

Các phần tử trong WSDL



WSDL document

Cấu trúc các phần tử trong 1 tài liệu WSDL



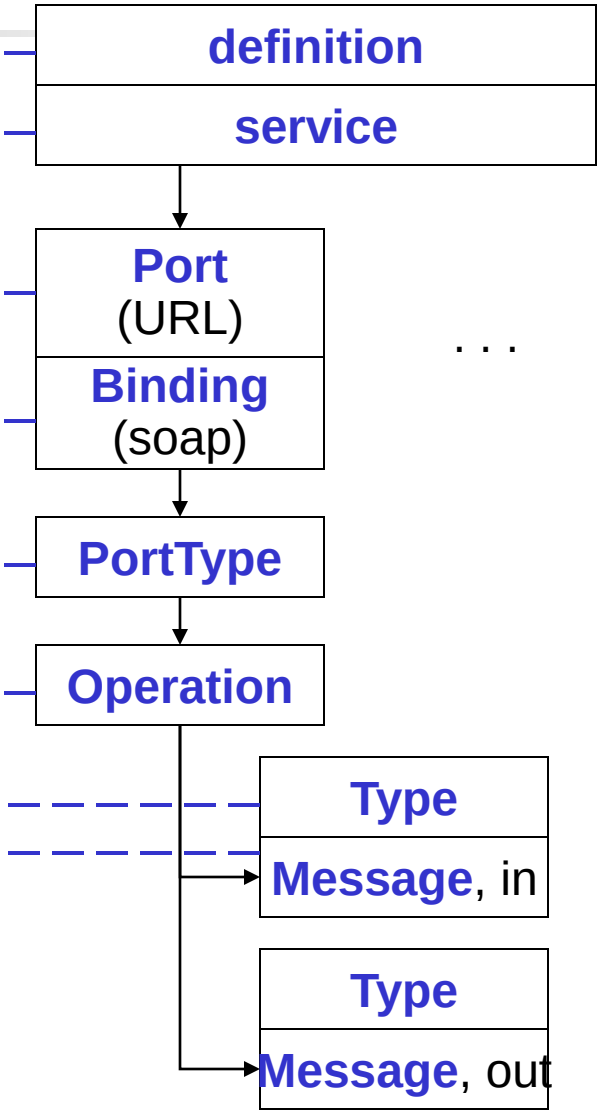
Ví dụ WSDL: Tra giá cổ phiếu

- Ví dụ về tài liệu WSDL cho dịch vụ đơn giản: tra giá cổ phiếu.
- Dịch vụ cung cấp 1 **operation** duy nhất: GetLastTradePrice, sử dụng giao thức SOAP 1.1 trên HTTP.
- Request:
  - Input: một biến string (**type**)
  - Output: trả về giá kiểu float (**type**).

Ví dụ WSDL: Tổ chức code

http://www.w3.org/TR/wsdl

```
<?xml version="1.0"?>
<definitions name="StockQuote"
    (namespaces used)
    <types> ... </types>
    <messages> … </messages>
     <portType>
        <operation> … </operation>
    </portType>
    <binding> … </binding>
    <service>
        <port> … </port>
    </service>
</definitions>
```



Ví dụ WSDL

```
<?xml version="1.0"?>
<definitions name="StockQuote"
    targetNamespace="http://example.com/stockquote.wsdl"
    xmlns:tns="http://example.com/stockquote.wsdl"
    xmlns:xsd1="http://example.com/stockquote.xsd"
    xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
    xmlns="http://schemas.xmlsoap.org/wsdl/">
    <types>
        ...
    </types>
```

Ví dụ WSDL (2)

Defined in “xmlns:xsd1="http://example.com/stockquote.xsd”

```
<types>
    <schema targetNamespace="http://example.com/stockquote.xsd"
        xmlns="http://www.w3.org/2000/10/XMLSchema">
        <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>
```

Ví dụ WSDL (3)

```
<message name="GetLastTradePriceInput">
   <part name="body"
                        element="xsd1:TradePriceRequest"/>
</message>
<message name="GetLastTradePriceOutput">
   <part name="body" element="xsd1:TradePrice"/>
</message>
<portType name="StockQuotePortType">
   <operation name="GetLastTradePrice">
      <input message="tns:GetLastTradePriceInput"/>
      <output message="tns:GetLastTradePriceOutput"/>
   </operation>
</portType>
```

Ví dụ WSDL (4)

```
<binding name="StockQuoteSoapBinding" type="tns:StockQuotePortType">
    <soap:binding style="document" transport=
                                "http://schemas.xmlsoap.org/soap/http"/>
    <operation name="GetLastTradePrice">
        <soap:operation soapAction=
                                "http://example.com/GetLastTradePrice"/>
        <input> <soap:body use="literal"/></input>
        <output> <soap:body use="literal"/></output>
    </operation>
  </binding>
  <service name="StockQuoteService">
    <documentation>My first service</documentation>
    <port name="StockQuotePort" binding="tns:StockQuoteBinding">
        <soap:address location="http://example.com/stockquote"/>
    </port>
  </service>
</definitions>
```

Tên của các phần tử

Definition: name="StockQuote“, namespaces

Service: name="StockQuoteService">

Port
name="StockQuotePort" binding="tns:StockQuoteBinding"> <**soap**:address location="…"

Binding
name="StockQuoteSoapBinding”

PortType: name="StockQuotePortType”

Operation: name="GetLastTradePrice”

Type: TradePriceRequest

Message, name="**GetLastTradePriceInput**”>

Type: TradePrice

Message, name="**GetLastTradePriceOutput**">

Các Namespaces được sử dụng trong WSDL và SOAP

- Tất cả các phần tử WSDL đều nằm trong WSDL namespace, định nghĩa trong: http://schemas.xmlsoap.org/wsdl/
- WSDL SOAP binding được định nghĩa trong http://schemas.xmlsoap.org/wsdl/soap/
- WSDL HTTP GET và POST binding, được định nghĩa trong http://schemas.xmlsoap.org/wsdl/http/
- WSDL MIME binding (MIME: Multipurpose Internet Mail Extension - RFC-2045 to RFC-2049) định nghĩa trong http://schemas.xmlsoap.org/wsdl/mime/

Các phần tử trong WSDL

WSDL documents sử dụng các phần tử sau:

- Phần tử gốc **definitions**:
  - Định nghĩa tên của web service,
  - Khai báo không gian tên sử dụng trong tài liệu
  - Chứa tất cả các phần tử service được mô tả dưới đây
- Phần tử **service:** là 1 tập các endpoints (ports). Định nghĩa địa chỉ để gọi 1 service cụ thể.

Các phần tử trong WSDL (2)

- Phần tử **types**: chứa các kiểu dữ liệu, sử dụng hệ thống kiểu dữ liệu nhất định (Vd: DTD hoặc XSD).
  - Mô tả tất cả các kiễu dữ liệu sử dụng giữa client và server.
  - Không bắt buộc, nhưng mặc định, WSDL sử dụng XML Schema Definition (XSD).
  - Nếu service sử dụng kiểu đơn giản sẵn có của XSD (strings hoặc integers), kiểu của phần tử là không bắt buộc.

Phần tử Types

PURCHASE ORDER TYPE

```
<element name="PO" type="tns:POType"/>
<complexType name="POType">
   <all>
      <element name="id" type="string"/>
      <element name="name" type="string"/>
      <element name="items">
         <complexType>
            <all>
               <element name="item" type="tns:Item" minOccurs="0" maxOccurs="unbounded"/>
            </all>
         </complexType>
      </element>
   </all>
</complexType>
```

ITEM TYPE

```
<complexType name="Item">
   <all>
      <element name="quantity" type="int"/>
      <element name="product" type="string"/>
   </all>
</complexType>
```

INVOICE TYPE

```
<element name="Invoice" type="tns:InvoiceType"/>
<complexType name="InvoiceType">
   <all>
      <element name="id" type="string"/>
   </all>
</complexType>
```

Các phần tử trong WSDL (3)

- Phần tử **port:** là 1 endpoint, được định nghĩa là kết hợp của 1 binding và 1 network address
- Phần tử **binding:** là 1 giao thức cụ thể và 1 đặc tả định dạng dữ liệu cho 1 port type cụ thể.
  - Mô tả chi tiết cách service được truy cập từ xa
- Phần tử **portType:**
  - Phần tử quan trọng nhất, định nghĩa 1 Web service, các **operations** có thể thực hiện và các **messages** liên quan
  - Tương đương với khái niệm function library trong lập trình truyền thống. Mỗi **operation** tương đương với function

Các phần tử trong WSDL (4)

- Phần tử **operation:** mô tả 1 hành động được hỗ trợ bởi service

- Các loại **operation**:
  - **One-way**: operation nhận 1 message nhưng không trả về 1 response
  - **Request-response**: operation nhận 1 request vàwill trả về 1 response
  - **Solicit-response**: operation gửi 1 request và chờ 1 response
  - **Notification**: operation gửi 1 message nhưng không chờ 1 response
  - Phiên bản 1.2: **request-multiple response** …

Ví dụ về loại **One-way** và **Notification** Operation

```
<portType name="RegisterPort">
   <operation name="register">
      <input name="customerInfo" message="RegInfo"/>
   </operation>

   <operation name = "register Response">
      <output name = "response" message="ResponseInfo"/>
   </operation>
</portType >
```

Ví dụ về loại **One-way** và **Request-Response** Operation

ONE-WAY:

```
<wsdl:operation name="Purchase">
  <wsdl:input name="Order" message="PO"/>
</wsdl:operation>
```

REQUEST-RESPONSE:

```
<wsdl:operation name="Purchase">
  <wsdl:input name="Order" message="PO"/>
  <wsdl:output name="Confirm" message="Conf"/>
  <wsdl:fault name="Error" message="POError"/>
</wsdl:operation>
```

Các phần tử trong WSDL (5)

- Phần tử **message**: định nghĩa phần tử dữ liệu của 1 **operation**
  - Có thể là 1 request hoặc response message 1 chiều
  - Mỗi message và chứa 0 hoặc nhiều **part**, (mỗi part tương đương với các tham số của 1 function trong lập trình truyền thống)

Message

```
<message name="PO">                                  1.0
  <part name="po" element="tns:PO"/>
  <part name="invoice" element="tns:Invoice"/>
</message>
```

Ví dụ WSDL: Phép toán cộng (1)

```
<definitions namespaces defined here>
<types>
    <s:schema elementFormDefault="qualified” targetNamespace="http://tempuri.org/">
      <s:element name="Add">
        <s:complexType>
          <s:sequence>
            <s:element minOccurs="1" maxOccurs="1" name="x" type="s:int" />
            <s:element minOccurs="1" maxOccurs="1" name="y" type="s:int" />
          </s:sequence>
         </s:complexType>
      </s:element>

      <s:element name="AddResponse">
        <s:complexType>
          <s:sequence>
            <s:element minOccurs="1" maxOccurs="1" name="AddResult“ type="s:int" />
        </s:sequence>
       </s:complexType>
     </s:element>
...
</types>
...</definitions>
```

Ví dụ WSDL: Phép toán cộng (2)

```
<definitions namespaces defined here>
  ...
// <message> maps the messages with data types defined in <types>
// element
    <message name="AddSoapIn">
        <part name="parameters" element="s0:Add" />
    </message>
    <message name="AddSoapOut">
        <part name="parameters" element="s0:AddResponse" />
    </message>

// <portType> element specifies what methods Web service provides,
// and the input and output messages for these methods
<portType name="Service1Soap">
  <operation name="Add">
    <input message="s0:AddSoapIn" />
    <output message="s0:AddSoapOut" />
   </operation>
</portType>
...
</definitions>
```

Ví dụ WSDL: Phép toán cộng (3)

```
<definitions namespaces defined here>
  ...
// <binding> element specifies message protocols accepted by Web service
<binding name="Service1Soap">
  <operation name="Add">
    <input message="s0:AddSoapIn" />
    <output message="s0:AddSoapOut" />
   </operation>
</binding>

// <service> element specifies the Web service's endpoint for each binding
<service name="Service1">
  <port name="Service1Soap" binding="s0:Service1Soap">
    <soap:address location="URL of Web service" />
  </port>
</service>


...
</definitions>
```

2.4. UDDI

Publish các Web Services

- Web services có thể được publish qua Microsoft's DISCO hoặc qua UDDI
  - DISCO: công nghệ publishing/discovery của Microsoft tích hợp trong .NET
    - Web service được mô tả trong tài liệu **.disco**
    - **.disco** chứa các tham chiếu tới các tài liệu mô tả (WSDL+docs+links)
    - Tài liệu DISCO có thể có tham chiếu tới các sites khác và tài liệu DISCO khác
  - UDDI: industry-wide publishing/discovery initiative

"A Telephone Book"

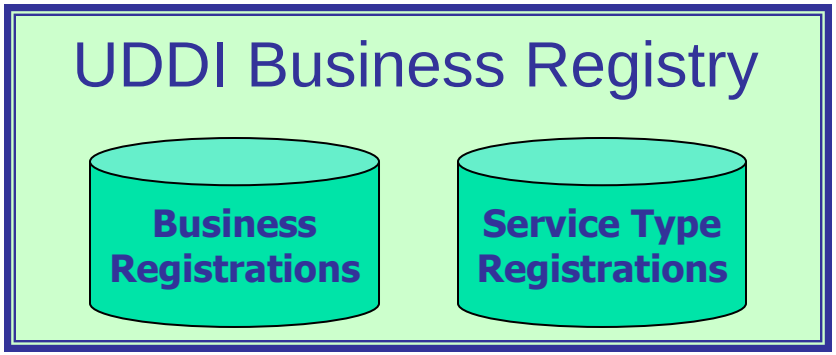
- Làm cách nào để tìm ra 1 Web service?
- Làm cách nào để đăng ký Web service để người khác có thể tìm thấy nó?
- UDDI là một chuẩn để publish và tìm kiếm các Web services
- UDDI ≈ telephone book

Cách thức UDDI hoạt động?

1.Software companies, standards bodies and programmers populate the registry with descriptions of different

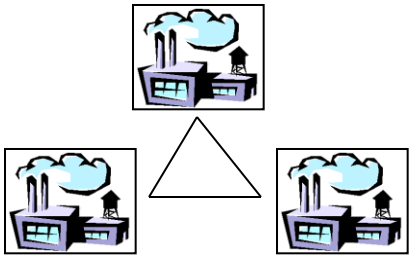
service specifications

4. Marketplaces, search engines and business apps query the registry to discover services at other companies



2. Businesses populate the registry with descriptions of the services they support

3. UDDI assigns a universally unique identifier (UUID) to each registry record



5. Businesses use this data to facilitate easier integration with each other over the Web

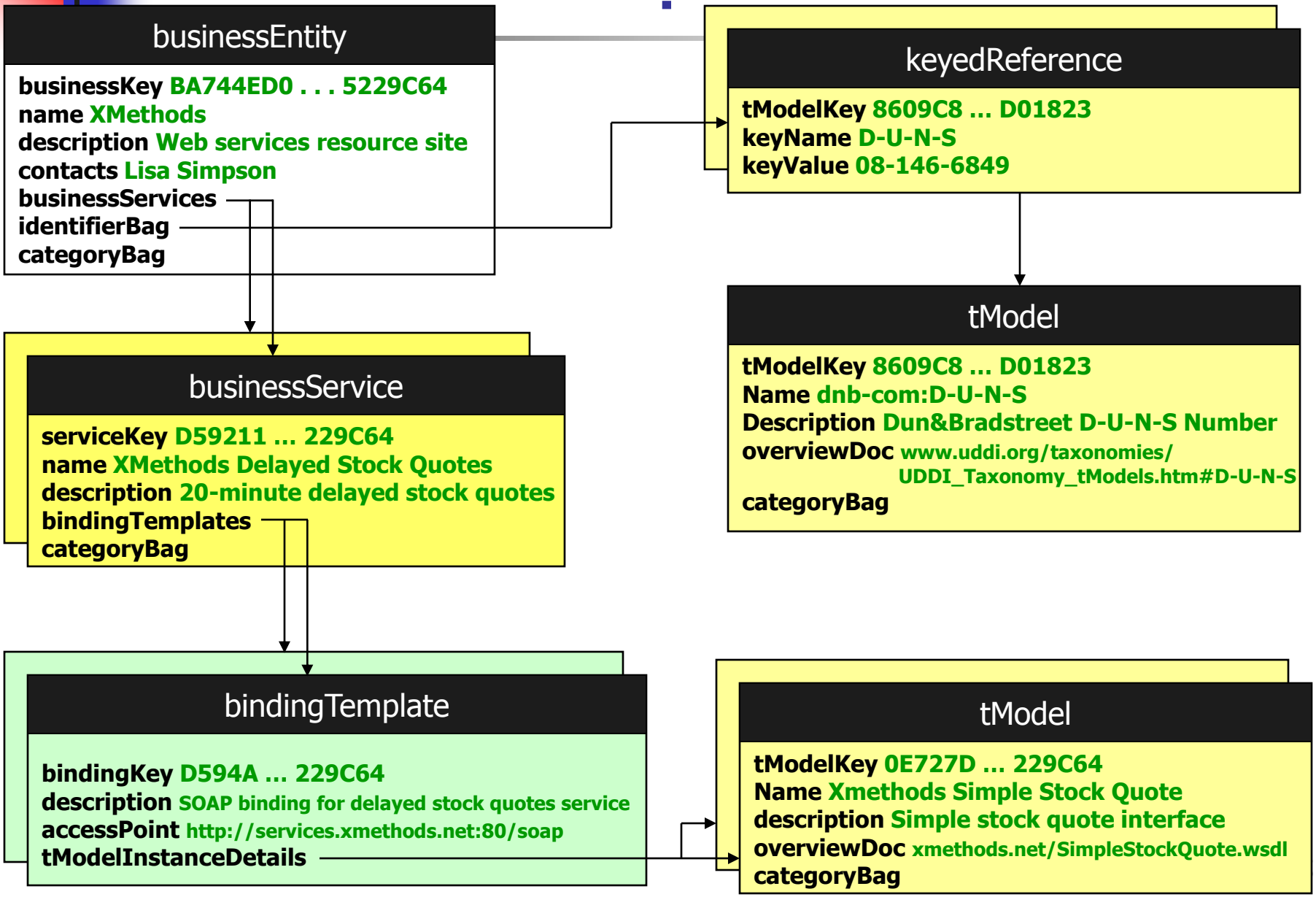
Các thành phần trong UDDI registry

- 1 entry trong UDDI registry là 1 tài liệu XML, gồm nhiều phần tử, quan trọng nhất là:
  - businessEntity: Mô tả về tổ chức cung cấp dịch vụ.
  - businessService: Danh sách các dịch vụ cung cấp bởi business entity
  - Binding template: Mô tả các khía cạnh về công nghệ của dịch vụ cung cấp
  - tModel ("technical model"): cung cấp thêm thông tin về dịch vụ, đặc biệt thông tin công nghệ: cách sử dụng, điều kiện sử dụng, giao thức, định dạng, …

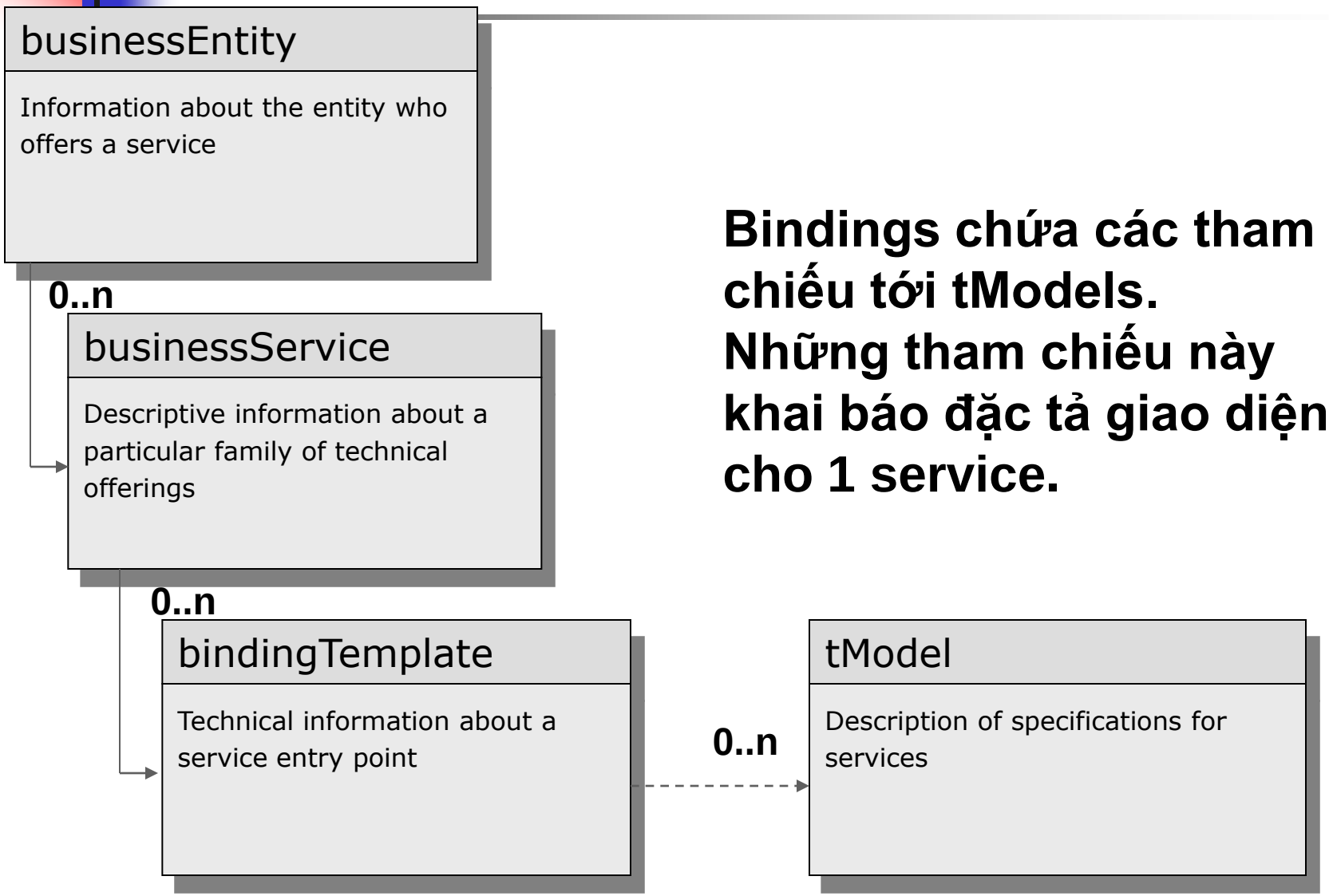
Thông tin cung cấp được chia thành 3 loại trang

- White Pages:
  - Thông tin cơ bản: tên doanh nghiệp, địa chỉ, …
  - Cho phép tìm kiếm dựa trên identification
- Yellow Pages:
  - Mô tả các dịch vụ Web theo các category
  - 3 loại chính trong v1.
    - Industry: NAICS (Industry codes - US Govt.)
    - Product/Services: UN/SPSC (ECMA)
    - Location: Geographical taxonomy
- Green Pages:
  - Các thông tin kỹ thuật về cách sử dụng service, các phương thức cung cấp của các Web services
  - Chứa đường dẫn đến tài liệu WSDL tương ứng

Mô hình dữ liệu UDDI

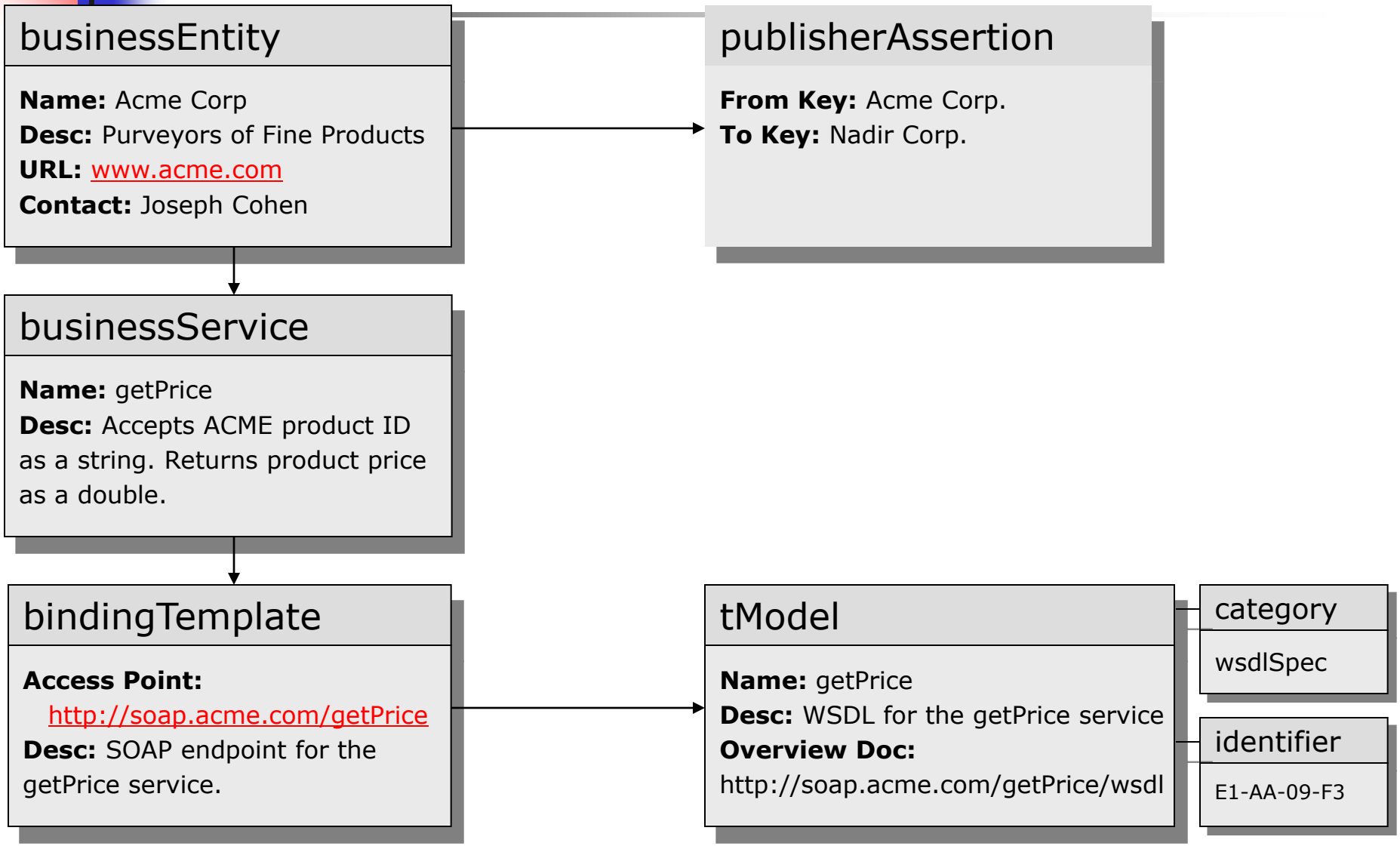


Key Entities Description

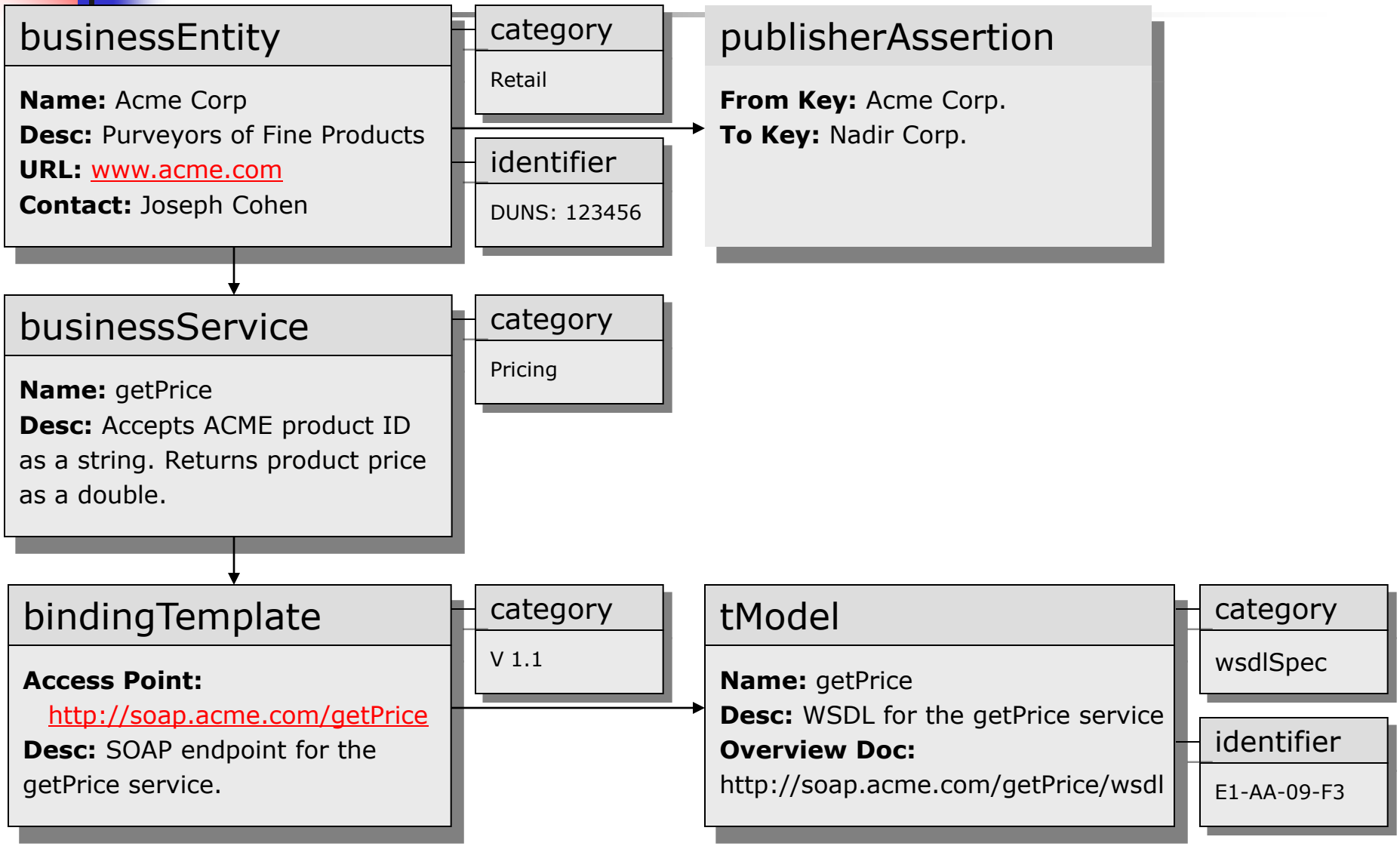


Bindings chứa các tham chiếu tới tModels. Những tham chiếu này khai báo đặc tả giao diện cho 1 service.

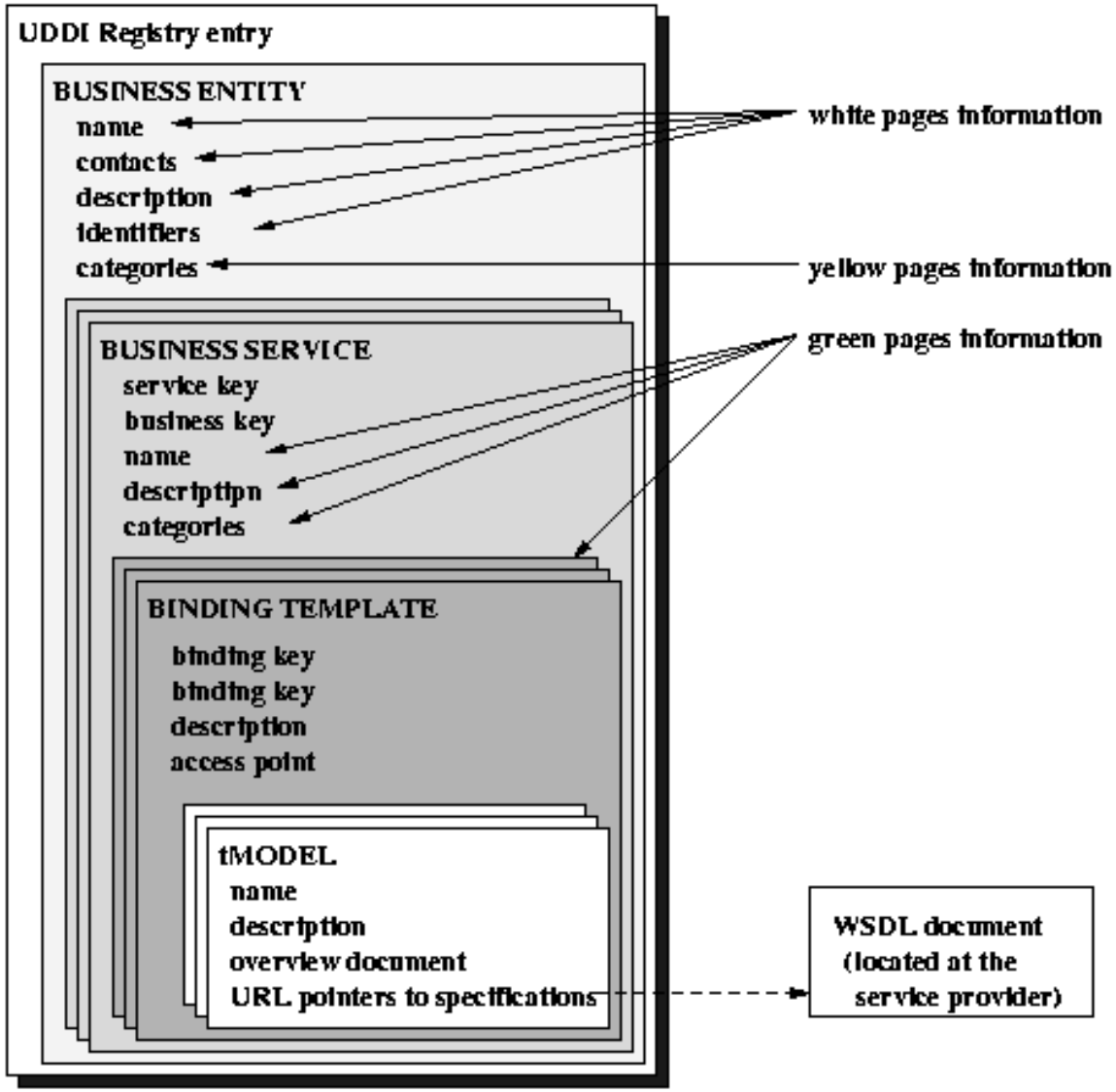
Key Entities Example



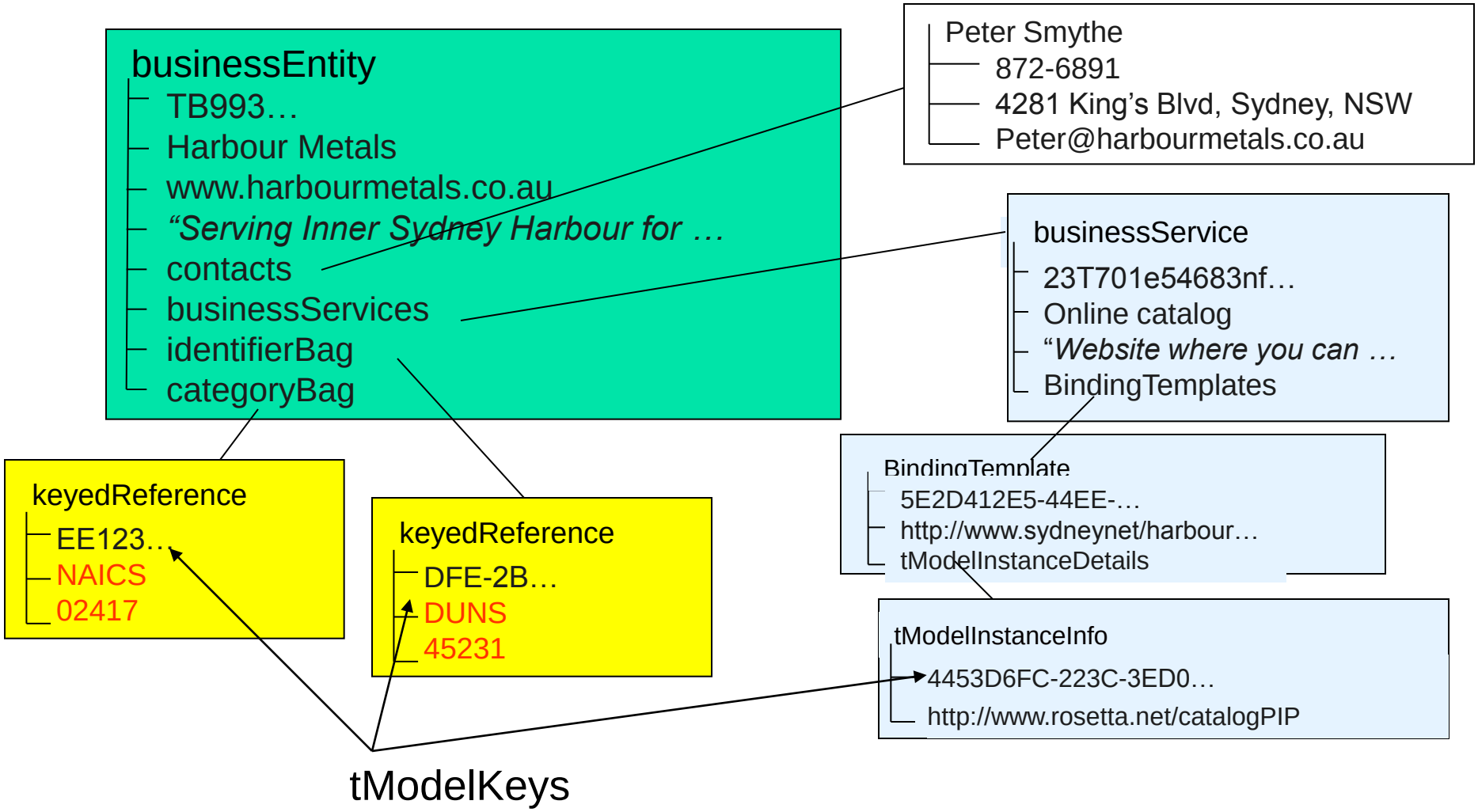
Categorizing Entities



Khái quát dữ liệu trong UDDI



Ví dụ về 1 bản đăng ký



Business Service XML

```
<businessService businessKey="..." serviceKey="...">
  <name>StockQuoteService</name>
  <description> (...) </description>
  <bindingTemplates>
      (...)
    <bindingTemplate>
      (...)
      <accessPoint urlType="http">
            http://example.com/stockquote
      </accessPoint>
      <tModelnstanceDetails>
            <tModelnstanceInfo tModelKey="...">
            </tModelnstanceInfo>
      <tModelnstanceDetails>
    </bindingTemplate>
  </bindingTemplates>
</businessService>
```

tModel XML

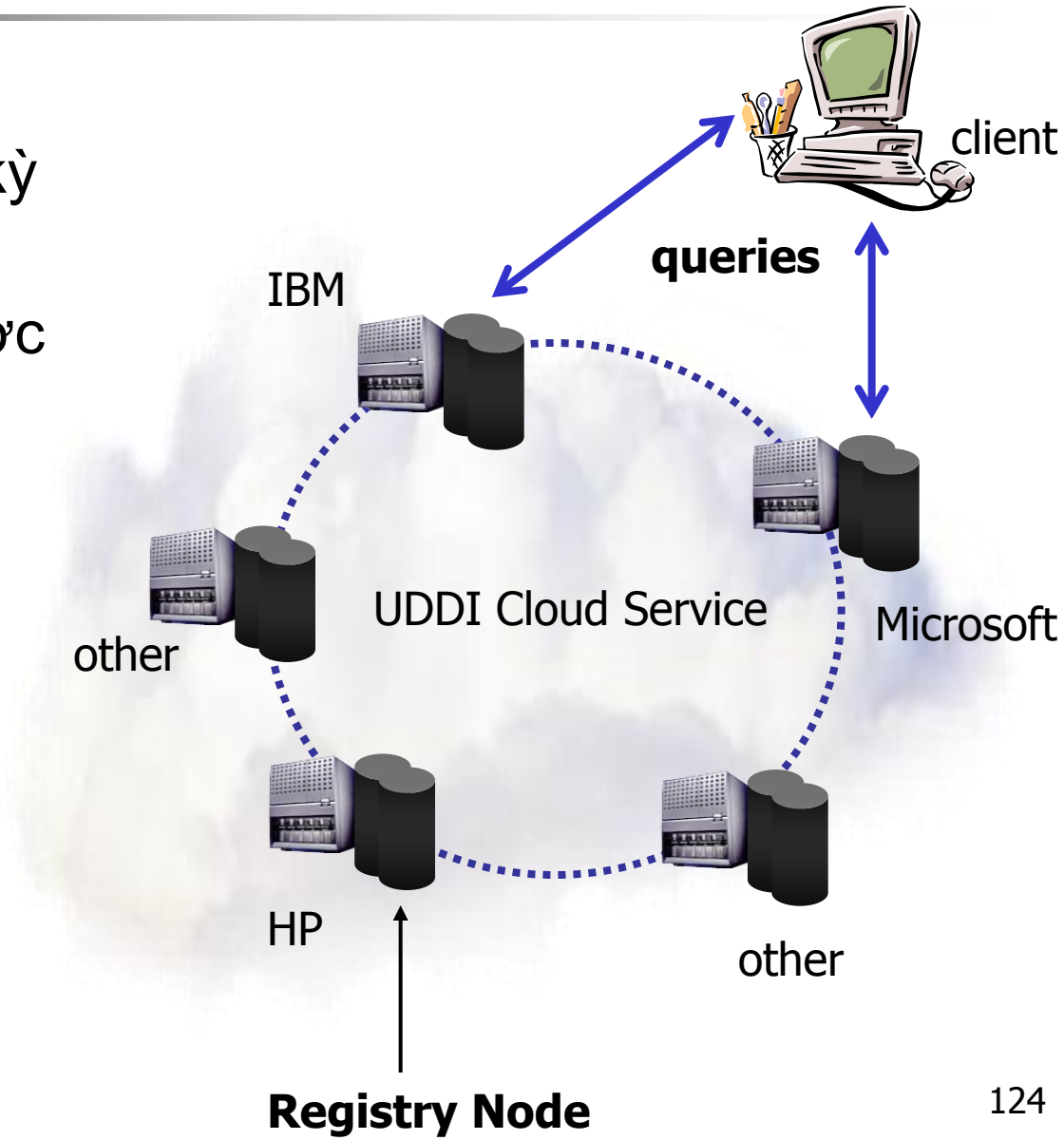
```
<tModel authorizedName="..." operator="..." tModelKey="...">
    <name>StockQuote Service</name>
    <description xml:lang="en">
        WSDL description of a standard stock quote service interface
    </description>
    <overviewDoc>
        <description xml:lang="en"> WSDL source document.</description>
        <overviewURL> http://stockquote-definitions/stq.wsdl</overviewURL>
    </overviewDoc>
    <categoryBag>
        <keyedReference tModelKey="UUID:...“ keyName="uddi-org:types"
        keyValue="wsdlSpec"/>
    </categoryBag>
</tModel>
```

UDDI Business Registry (UBR), Public Cloud

- Các Nodes chứa tất cả thông tin UDDI
- Các Nodes được đồng bộ hóa, nên có dữ liệu giống nhau
- Có thể truy vấn trên bất kỳ node nào
- Có thể thêm thông tin UDDI vào 1 node bất kỳ, nó sẽ được sao lưu sang các node khác

Registry Nodes Operation

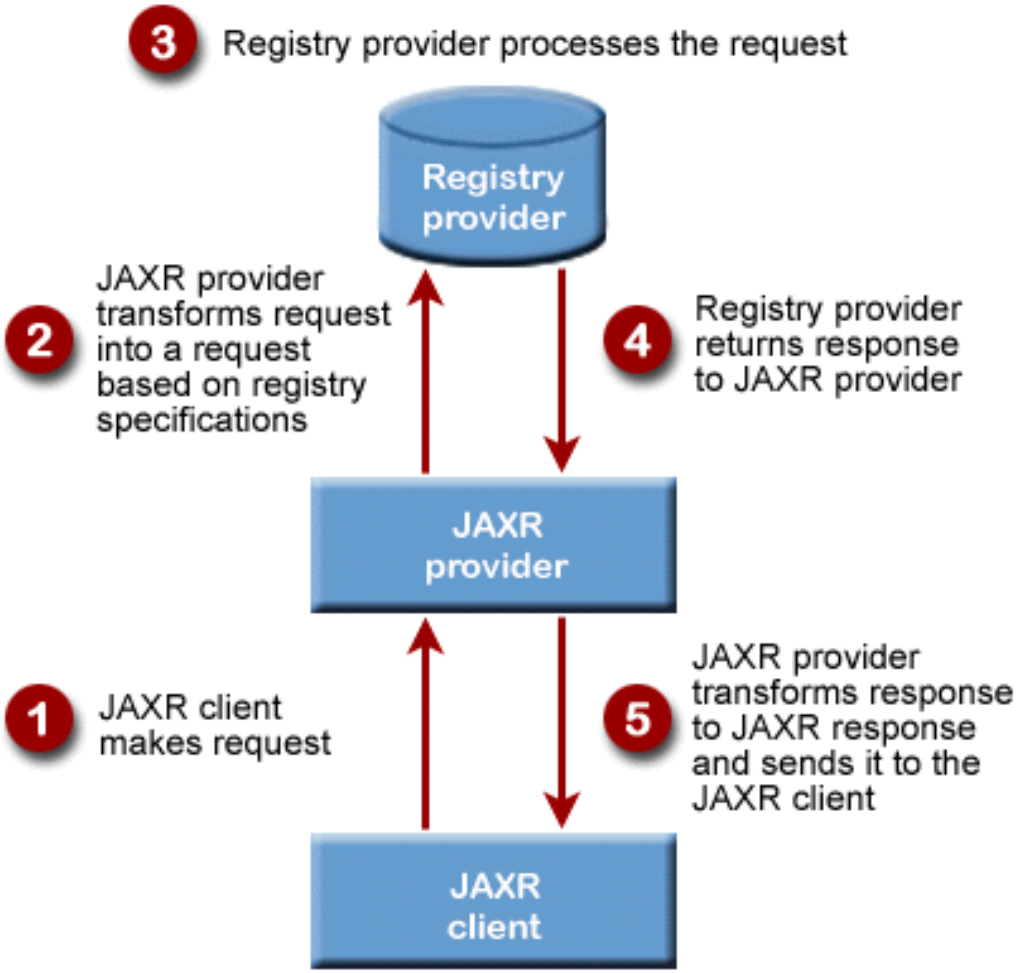
- Một nghiệp vụ có thể được đăng ký ở bất kỳ node nào
- Các bản đăng ký được nhân bản hàng ngày
- Hoạt động như DNS: tập trung về logic, phân tán về vật lý



Tương tác với UDDI

- **UDDI: bản chất là 1 Web service!!!**
- Tương tác qua SOAP messages
- Gói JAXR định nghĩa cách thức chuẩn tắc để tương tác với các bản ghi (có thể làm việc với các loại bản ghi khác như ebXML)
- 2 loại tương tác:
  - Inquiry: Không cần xác thực người dùng
  - Publish: Cần xác thực người dùng

JAXR



http://developer.java.sun.com/developer/technicalArticles/WebServices/WSPack/

UDDI API

- Inquiry API

  - find_business
  - find_service
  - find_binding
  - find_tmodel

  - get_businessDetail
  - get_serviceDetail
  - get_bindingDetail
  - get_tmodelDetail

- Publishing API

  - save_business
  - save_service
  - save_binding
  - save_tmodel

  - delete_business
  - delete_service
  - delete_binding
  - delete_tmodel

  - get_authtoken
  - discard_authtoken

UDDI Inquiry API

Name	Description
find_business	Locates information about one or more businesses.
find_service	Locates services within a registered businessEntity.
find_binding	Locates bindings within a registered businessService.
find_tModel	Locates one or more tModel information structures.
get_businessDetail	Gets businessEntity information for one or more businesses.
get_businessDetailExt	Gets extended businessEntity information.
get_serviceDetail	Gets full details for a set of registered businessServices.
get_bindingDetail	Gets bindingTemplate information for making service requests.
get_tModelDetail	Gets details for a set of registered tModels.

Ví dụ tìm kiếm 1 Business

```
<?xml version='1.0' encoding='utf-8'?>
<s:Envelope
  xmlns:s='http://schemas.xmlsoap.org/soap/envelope/'>
  <s:Body>
    <find_business generic="1.0" xmlns="urn:uddi-org:api">
      <name>Microsoft</name>
    </find_business>
  </s:Body>
</s:Envelope>
```

Kết quả tìm kiếm

```
<businessList generic="1.0" truncated="false"
  operator="Microsoft Corporation"
  xmlns="urn:uddi-org:api">
  <businessInfos>
    <businessInfo
     businessKey="0076B468-EB27-42E5-AC09-9955CFF462A3">
      <name>Microsoft Corporation</name>
      <description xml:lang="en">Empowering people through
great software...</description>
    <serviceInfos>
        <serviceInfo
         businessKey="0076B468-EB27-42E5-AC09-9955CFF462A3"
         serviceKey="1FFE1F71-2AF3-45FB-B788-09AF7FF151A4">
          <name>Web services for smart searching</name>
        </serviceInfo>
...
      </serviceInfos>
    </businessInfo>
  </businessInfos>
</businessList>
```

UDDI Publishing API

Name	Description
get_authToken	Requests an authentication token from an operator site.
get_registeredInfo	Requests information currently managed by user.
save_business	Registers/updates a businessEntity.
save_service	Registers/updates a businessService.
save_binding	Registers/updates a bindingTemplate.
save_tModel	Registers/updates a tModel.
delete_business	Deletes a businessEntity from registry.
delete_service	Deletes a businessService from registry.
delete_binding	Deletes a bindingTemplate from registry.
delete_tModel	Deletes a tModel from registry.
discard_authToken	Discards an existing authentication token.

Ví dụ lưu một Business

```
<?xml version='1.0' encoding='utf-8'?>
<s:Envelope
  xmlns:s='http://schemas.xmlsoap.org/soap/envelope/'>
  <s:Body>
    <save_business generic="1.0" xmlns="urn:uddi-org:api">
      <!-- retrieved from get_authToken API -->
      <authinfo>fd3c7a44-118f-413a-a2e3-473a35379993
      </authinfo>
          <businessEntity
        businessKey="ee3be846-d828-4a38-a5e4-3c33f931d122">
        <name>Developmentor</name>
        <description>Services the developer by...
        </description>
        <businessServices>
          <!-- service descriptions go here -->
        </businessServices>
      </businessEntity>
    </save_business>
  </s:Body>
</s:Envelope>
```

3. Phát triển ứng dụng Web Services với Java

IDE & API

- Ví dụ về Web service với nhiều ngôn ngữ lập trình khác nhau:
  - http://oreilly.com/catalog/progwebsoap/chapter/ch03.html
- Eclipse/My Eclipse/Netbean
- Java API:
  - Apache AXIS
  - JAX-WS
  - Xfire
  - Apache CXF

Mô hình ứng dụng

- Server Application:
  - Cài đặt các dịch vụ Web
- Client Application
  - Sử dụng các dịch vụ được cung cấp từ server
- Ví dụ cài đặt với Eclipse:
  - http://www.eclipse.org/webtools/community/tutorials/BottomUpAxis2WebService/bu_tutorial.html
- Ví dụ cài đặt với MyEclipse
  - http://www.myeclipseide.com/documentation/quickstarts/webservices_jaxws_overview/

3.1. Server Application

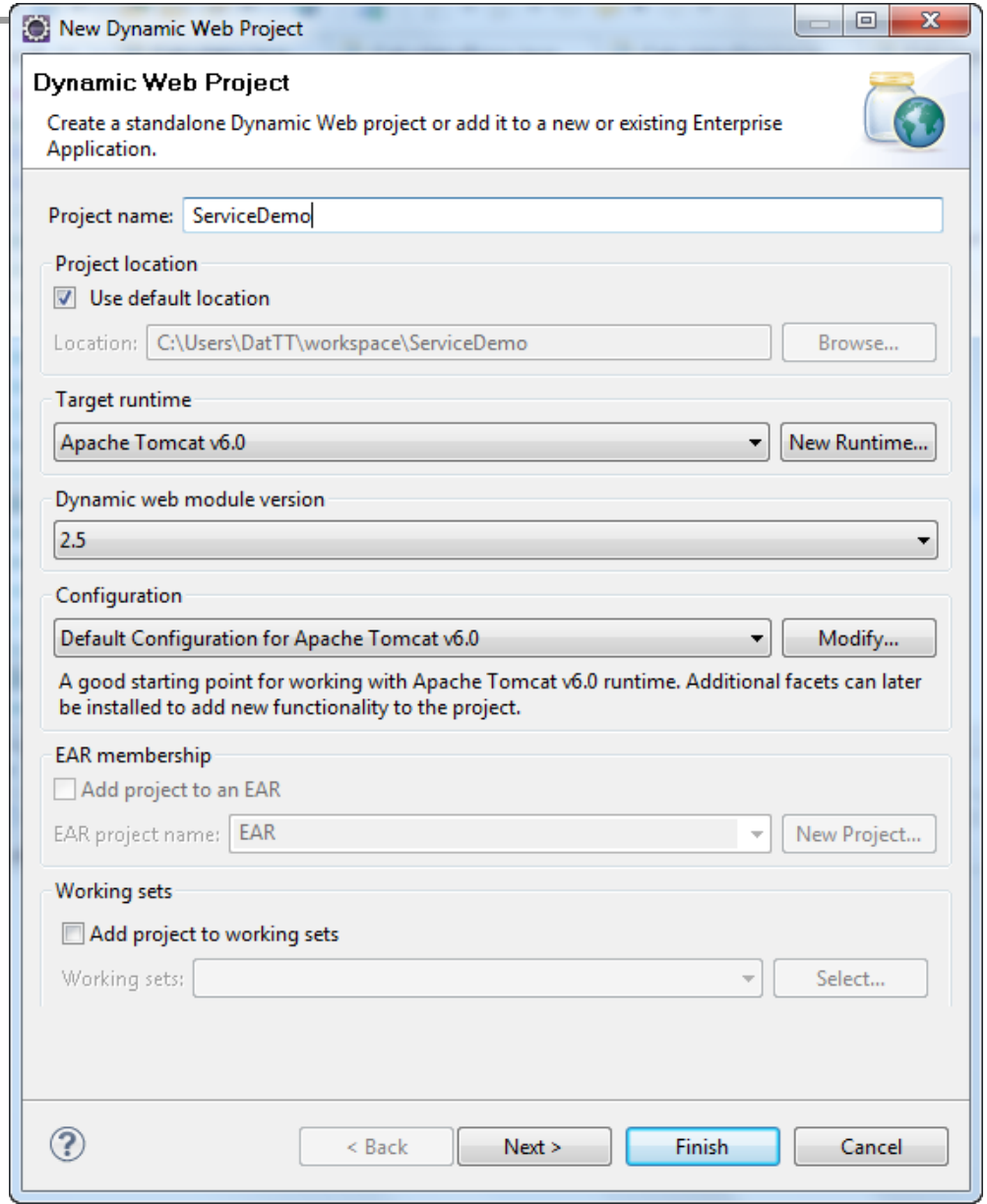
- Có thể tạo các Web services bằng 2 cách
  - Top-down:
    - Phức tạp hơn, nhưng là cách khuyên dùng, vì lập trình viên có thể kiểm soát mọi vấn đề
    - Thiết kế Web Service bằng cách tạo file WSDL trước (bằng các bộ soạn thảo WSDL)
    - Sinh tự động các lớp Java tương ứng
    - Lập trình viên code chi tiết cho các method/class
  - Bottom-up
    - Đơn giản hơn, dành cho người mới bắt đầu
    - Tạo các Java Bean hoặc EJB Bean trước
    - Sinh tự động WSDL và Web services.

3.1. Server Application

- Ứng dụng cần demo:
  - Cung cấp dịch vụ tính toán: các phép cộng, nhân
- Ví dụ với IDE Eclipse

3.1. Server Application

- Bước 1:
  - Tạo 1 dynamic Web Project: ServiceDemo



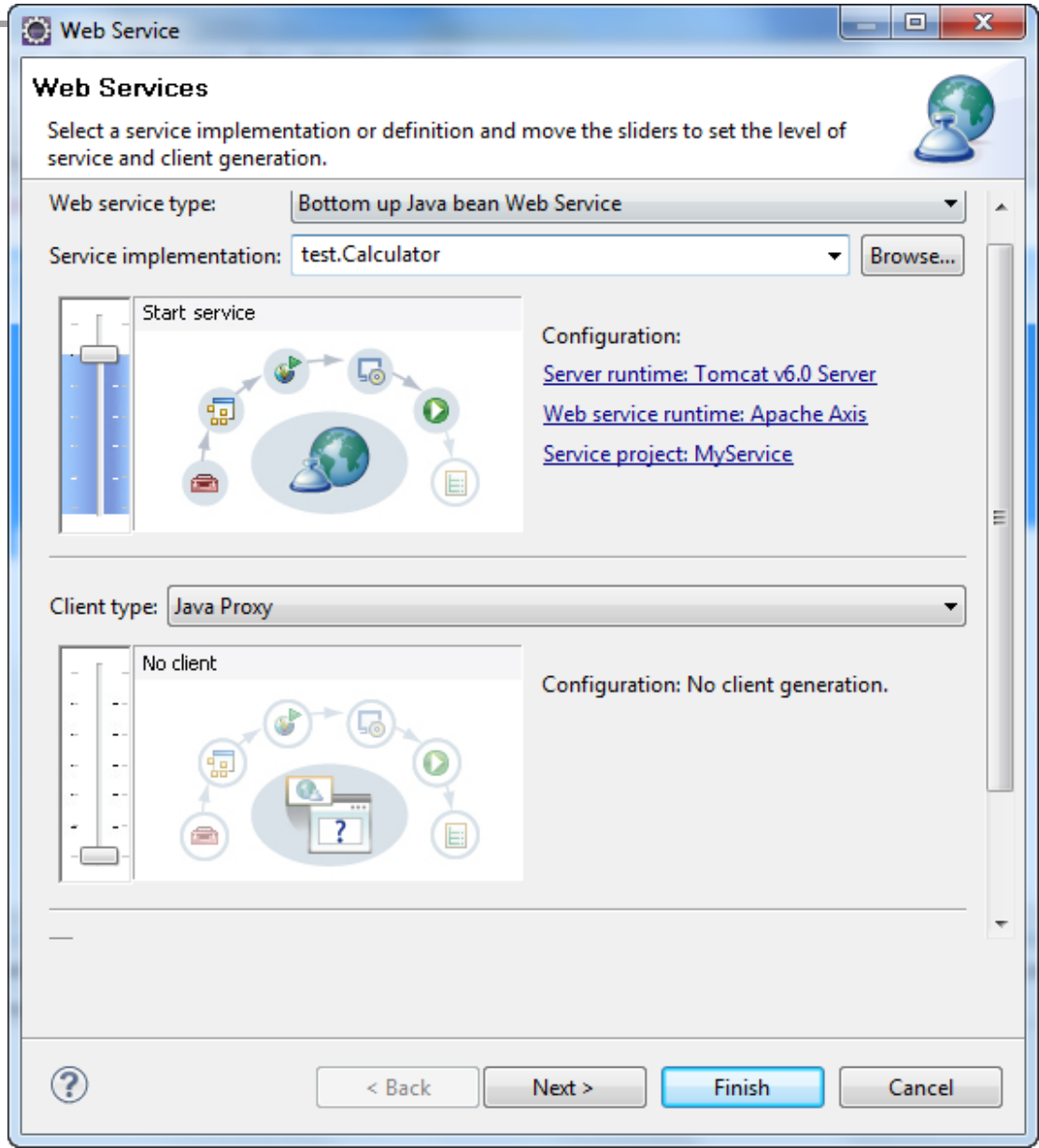
3.1. Server Application

- Bước 2:
  - Tạo lớp Java Bean Calculator

```
public class Calculator {
  public int add(int x, int y)
  {
   return x+y;
  }
  public int mul(int x, int y)
  {
   return x*y;
  }
}
```

3.1. Server Application

- Bước 3:
  - Tạo service (File →New→Web Service)
  - Chọn đến Java Bean cần cung cấp thành các services
  - Để các giá trị mặc định, nhấn finish

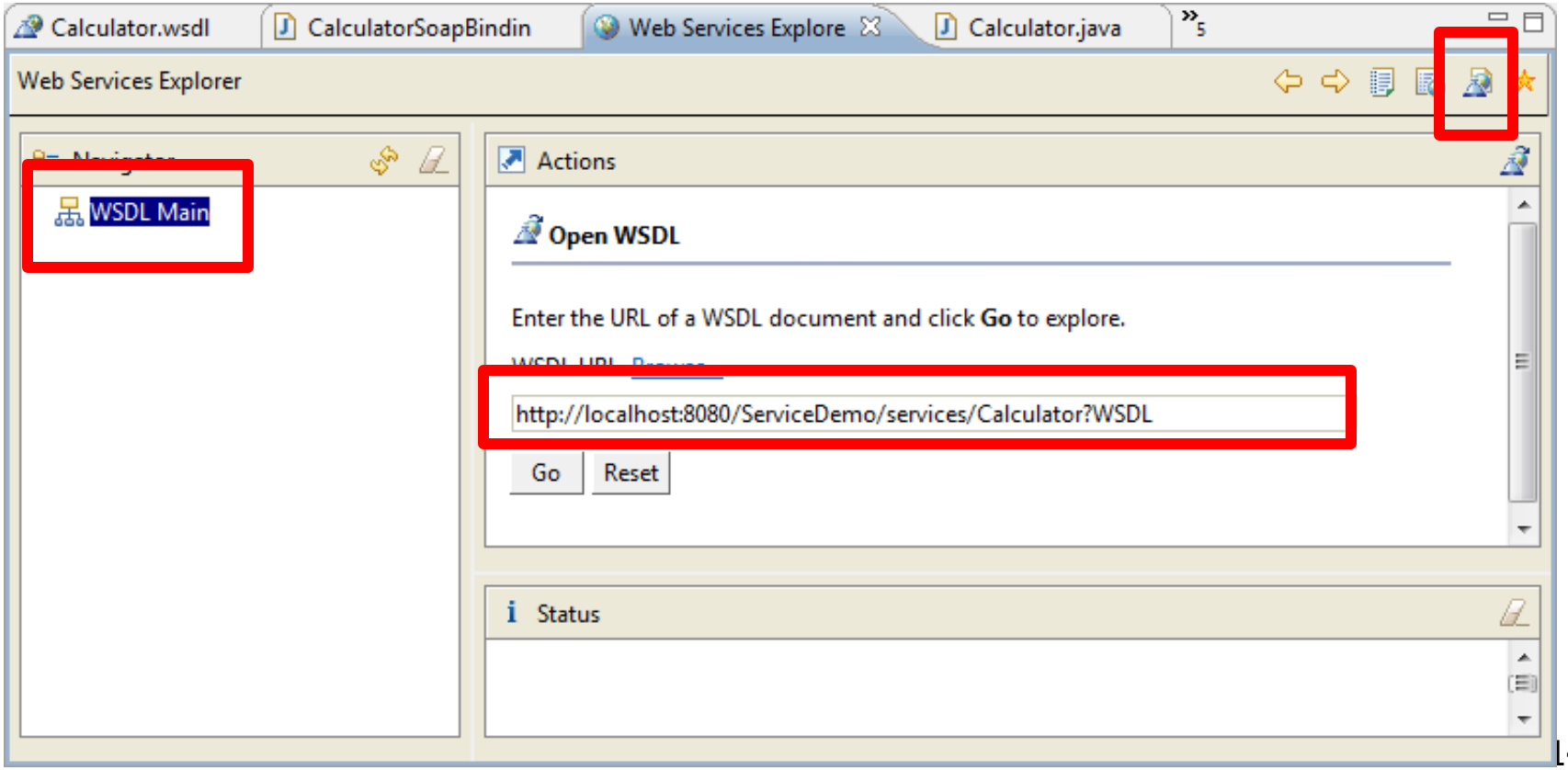


3.1. Server Application

- Bước 3 (tiếp)
  - IDE sẽ sinh tự động file wsdl và các file wsdd (web service deployment descritor) cần thiết
  - Lưu ý địa chỉ của service (Trong file Calculator.WSDL)
    - `<wsdlsoap:address location="http://localhost:8080/ServiceDemo/services/Calculator"/>`
- → Hoàn thành cài đặt phần server

3.2. Test service với uddi của Eclipse

- Mở View "Web Service Explorer" (Trên thanh toolbar) ()



3.2. Test service với uddi của Eclipse

Web Services Explorer

Navigator

- WSDL Main
  - http://localhost:8081/MySe
    - CalculatorService
      - CalculatorSoapBindi

Actions

WSDL Binding Details

Shown below are the details for this **SOAP** <binding> element. Click on an operation to fill in its parameters and invoke it or specify additional endpoints.

Operations

Name	Documentation
mul	--
add	--

Endpoints Add Remove

Endpoints
http://localhost:8081/MyService/services/Calculator

Go Reset

Status

IWAB0381I http://localhost:8081/MyService/services/Calculator?WSDL was successfully opened.

3.2. Test service với uddi của Eclipse

Web Services Explorer

Navigator

- WSDL Main
  - http://localhost:8081/MySe
    - CalculatorService
      - CalculatorSoapBindir
        - mul
        - add

Actions

Invoke a WSDL Operation Source

Enter the parameters for the WSDL operation "mul" and click **Go** to invoke.

Endpoints

http://localhost:8081/MyService/services/Calculator

Body

mul

x int

3

y int

5

Go Reset

Status

IWAB0381I http://localhost:8081/MyService/services/Calculator?WSDL was successfully opened.

3.2. Test service với uddi của Eclipse

Web Servi

Navi

WS

Status

Form

SOAP Request Envelope:

```
- <soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
    xmlns:q0="http://test" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  - <soapenv:Body>
    - <q0:mul>
        <q0:x>3</q0:x>
        <q0:y>5</q0:y>
      </q0:mul>
    </soapenv:Body>
  </soapenv:Envelope>
```

SOAP Response Envelope:

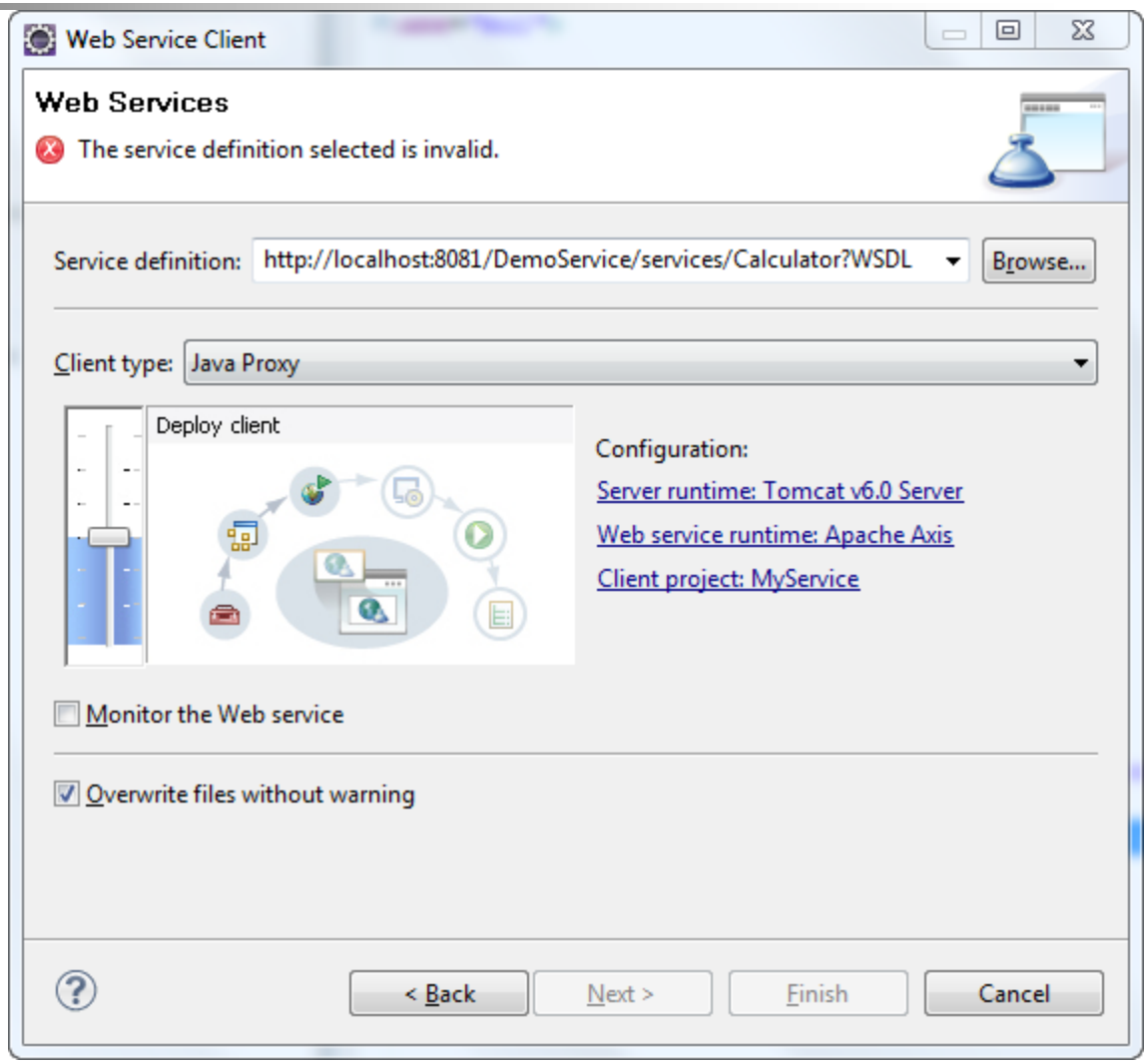
```
- <soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  - <soapenv:Body>
    - <mulResponse xmlns="http://test">
        <mulReturn>15</mulReturn>
      </mulResponse>
    </soapenv:Body>
  </soapenv:Envelope>
```

3.3. Client Application

- Bước 1:
  - Tạo 1 project dạng bất kỳ (Java Project, Web Project, …)

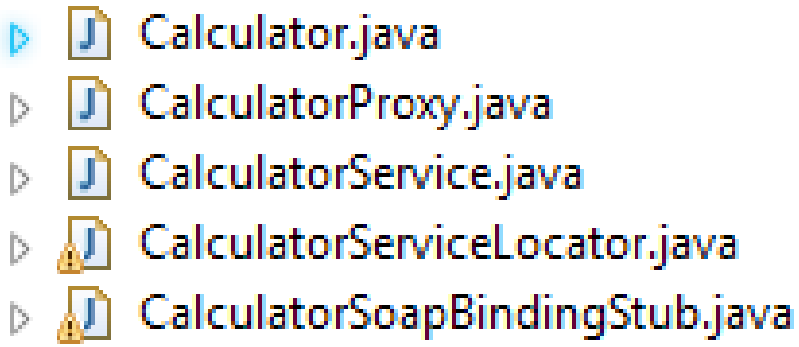
3.3. Client Application

- Bước 2:
  - Tạo Web Service Client (File→New→ Web Service Client)
  - Nhập vào địa chỉ của Service
  - Để mặc định và chọn Finish



3.3. Client Application

- Bước 2 (tiếp)
  - IDE sẽ sinh tự động 5 lớp
  - Chỉ cần quan tâm đến lớp Stub



3.3. Client Application

- Bước 3: Viết lớp gọi Service

```
public class DemoClient {
  public static void main(String[] args) {
    try {
      CalculatorSoapBindingStub stub
              = new CalculatorSoapBindingStub();
      System.out.println(stub.add(5, 6));
    } catch (AxisFault e) {
      e.printStackTrace();
    } catch (RemoteException e) {
      e.printStackTrace();
    }
  }
}
```

Tài liệu tham khảo

- Web Services Essentials
  - http://msdn.microsoft.com/library/default.asp?URL=/library/techart/webservicesessentials.htm
- W3Schools SOAP Tutorial
  - http://www.w3schools.com/soap/default.asp
- SOAP
  - http://msdn.microsoft.com/soap
- SOAP Specification
  - http://www.w3.org/TR/SOAP/
- Don Box on SOAP
  - http://msdn.microsoft.com/msdnmag/issues/0300/soap/soap.asp
- Introduction to SOAP
  - http://www.w3.org/2000/xp/Group/Admin/minutes-oct1100/soap-xp-wg_files/frame.htm

Tài liệu tham khảo (2)

- W3Schools WSDL Tutorial
  - http://www.w3schools.com/wsdl/default.asp
- WSDL Specification
  - http://www.w3.org/TR/wsdl
- A Quick Introduction to WSDL
  - http://msdn.microsoft.com/library/default.asp?url=/library/en-us/soap/htm/soap_overview_72b0.asp
- UDDI
  - http://www.uddi.org
  - http://uddi.microsoft.com
- Building Web Services with SOAP and ASP.NET
  - http://msdn.microsoft.com/msdnmag/issues/01/02/WebComp/webcomp.asp

Web Services Stack

- Discovery
  - Directory allows potential clients to locate relevant Web Services
    - UDDI (Universal Description, Discovery, and Integration)
  - Inspection allows you to locate documents about Web Services located on at a given URL
    - DISCO
  - A Description language defines the format of methods provided by a Web Service
    - WSDL

Web Services Stack

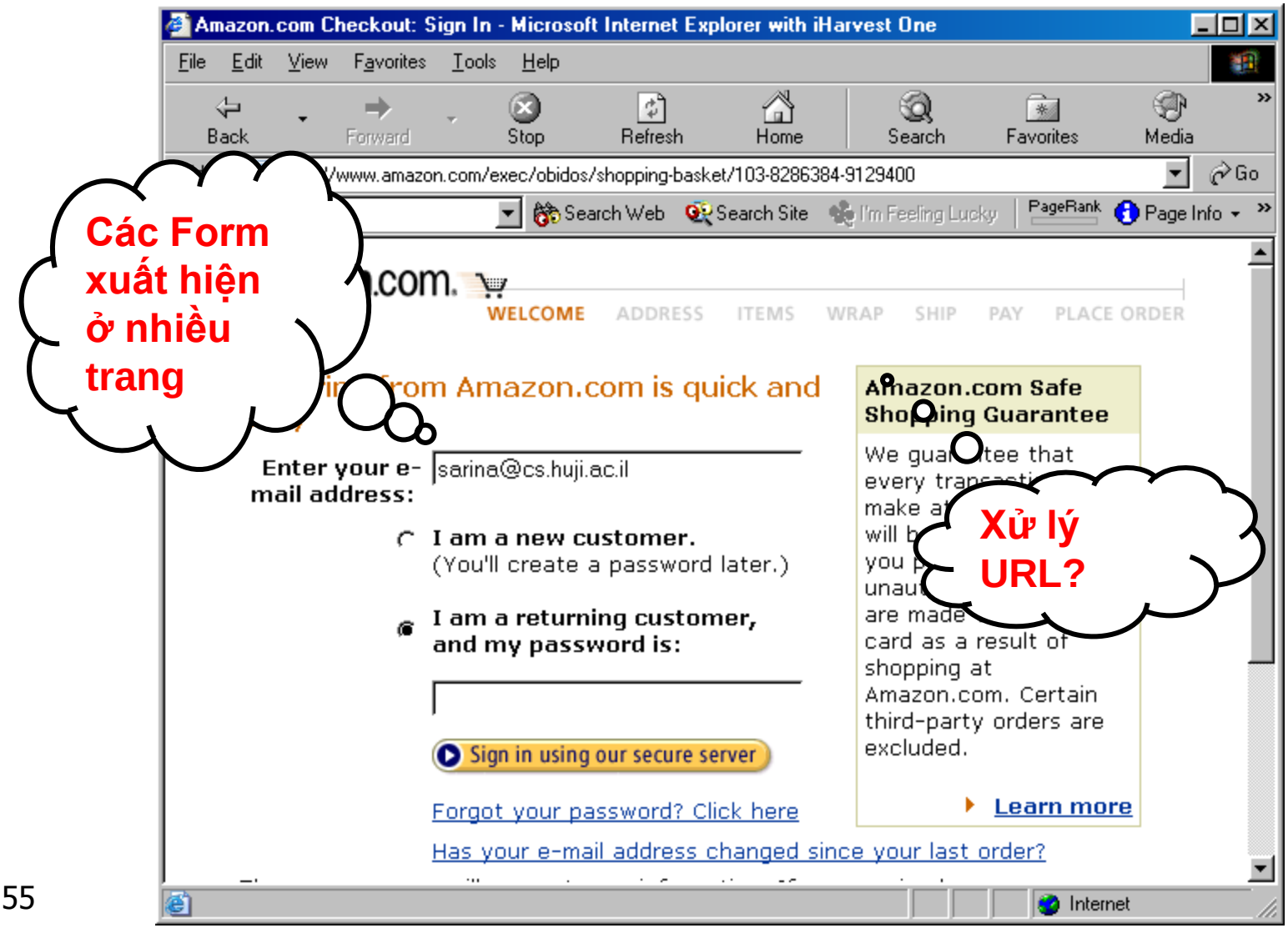
Directory: Publish & Find Services:	UDDI
Inspection: Find Services on server:	DISCO
Description: Formal Service Descriptions:	WSDL
Wire Format: Service Interactions:	SOAP
Universal Data Format:	XML
Everywhere Communications:	Internet

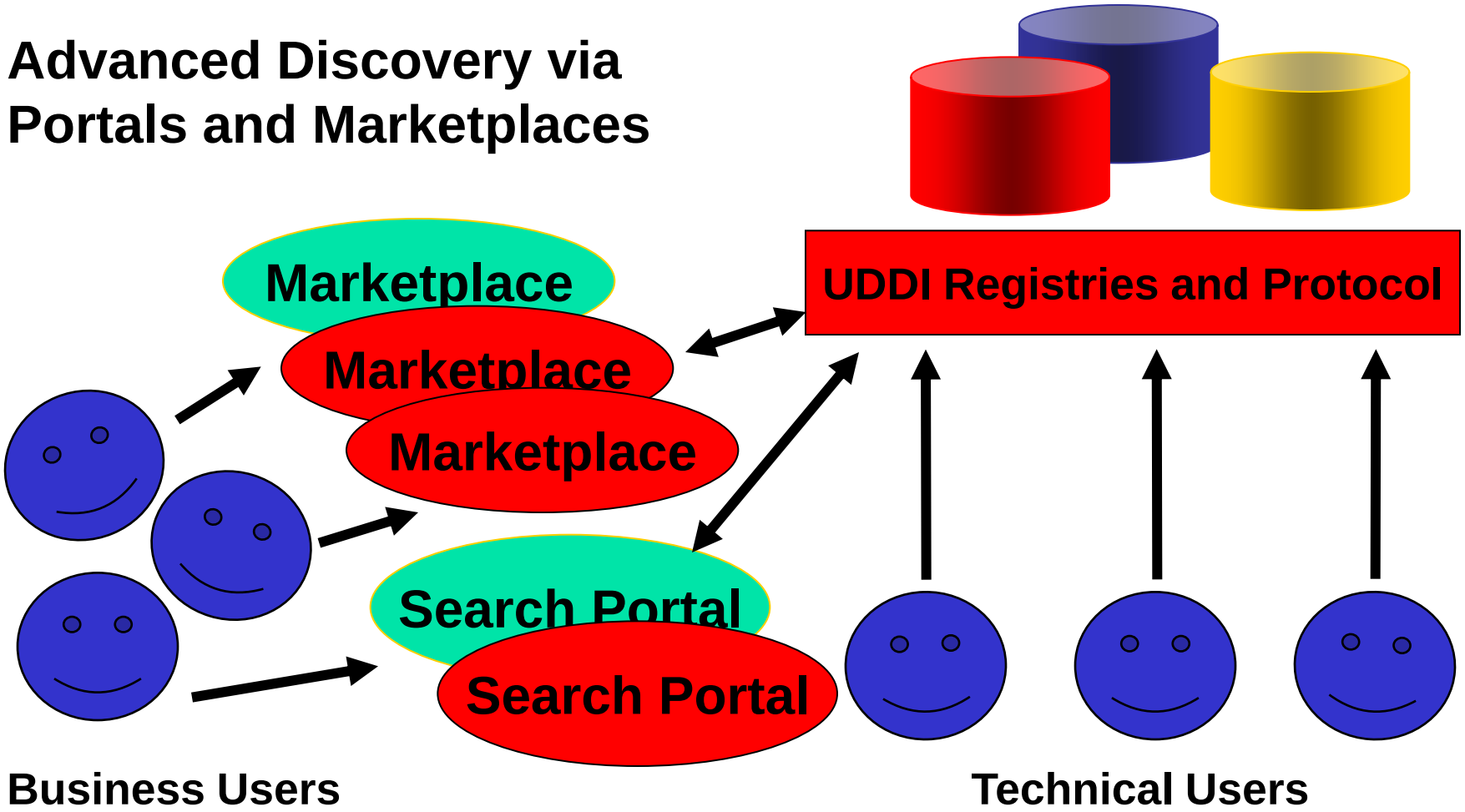
Simple, Open, Broad Industry Support

Web Services (In Practice)

Design-Time or Dynamic Runtime

Khó khăn khi sử dụng 1 web sites như 1 web service (2)-Điền form



UDDI - The Vision

Advanced Discovery via Portals and Marketplaces