Trường Cao Đẳng Lý Tự Trọng TP. HCM

Khoa Công nghệ thông tin

GIÁO TRÌNH

Công nghệ & Dịch vụ Web

GV: Trần Việt Khánh

Nội dung trình bày

- Chương 1: Công nghệ Web Service
- Chương 2: Kiến trúc Web Service
- Chương 3: Xây dựng Web service
- Chương 4: Khai thác Web service
- Chương 5: Bảo mật trong Web service

Chương 1: Công nghệ Web Service

1.1 Web Service là gì ?

1.2 Đặc điểm Web Service

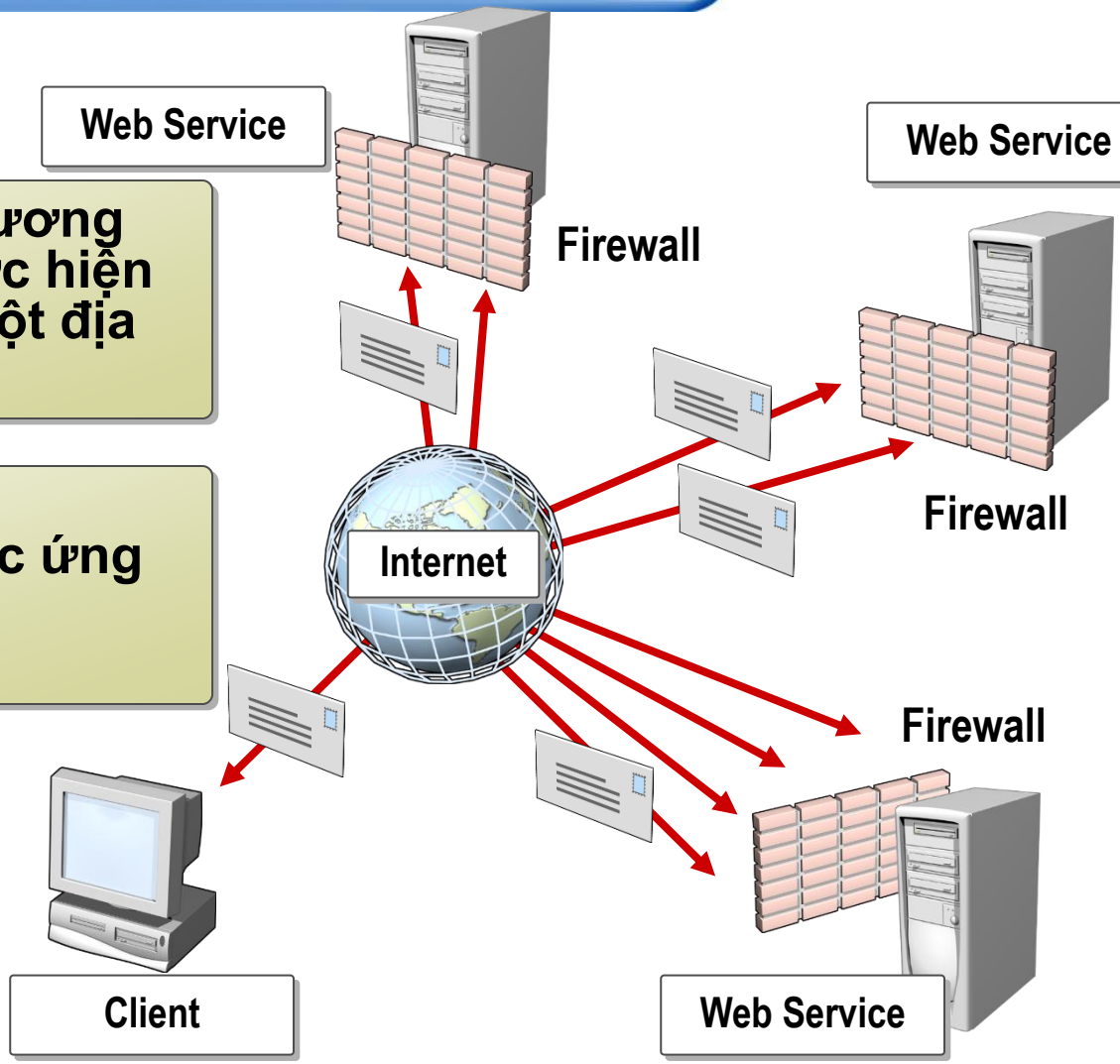
1.3 Nền tảng của web service

1.4 Các công nghệ của web service

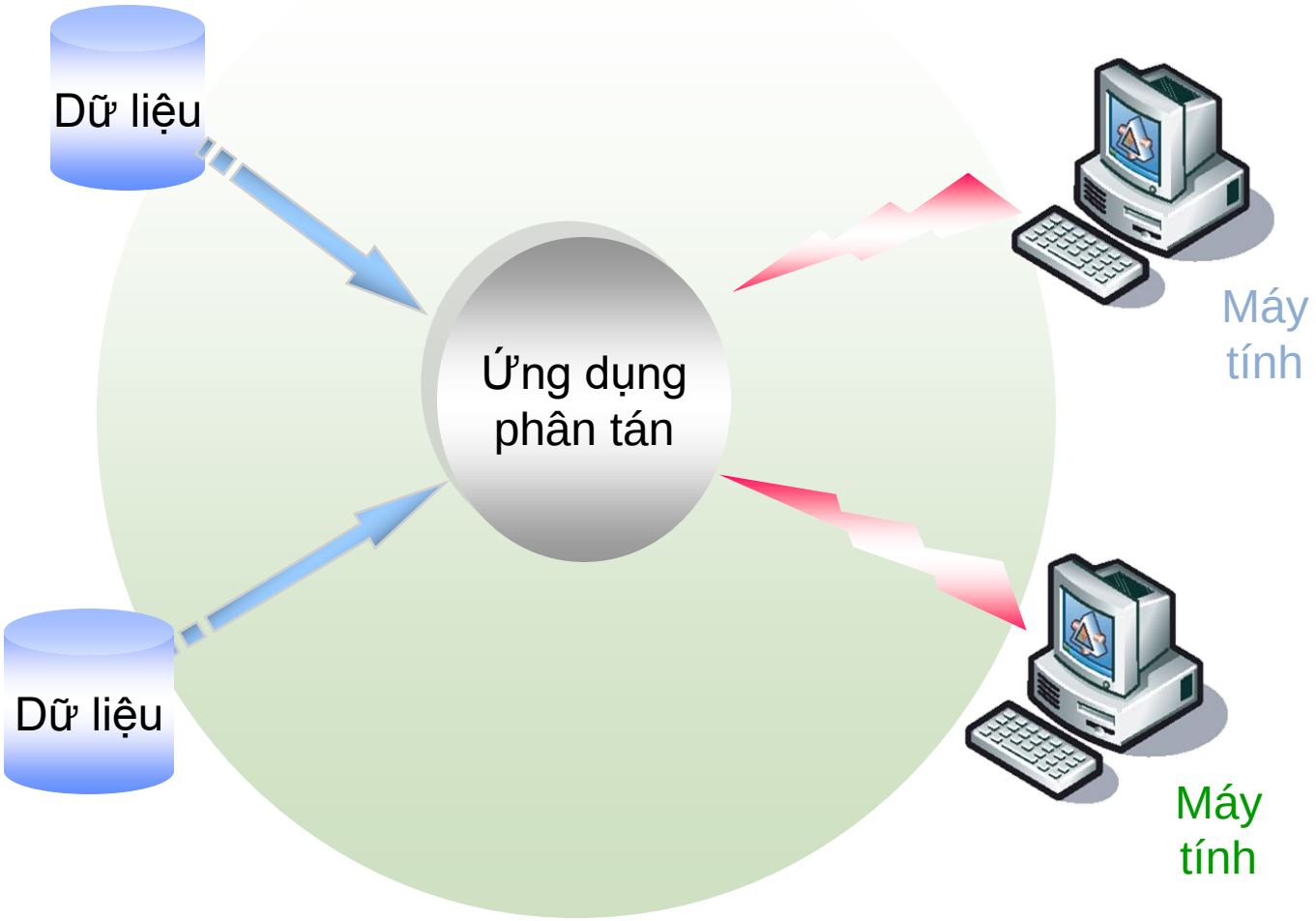
1.1 Web service là gì

1. Là một tập các phương thức được gọi thực hiện từ xa thông qua một địa chỉ URL

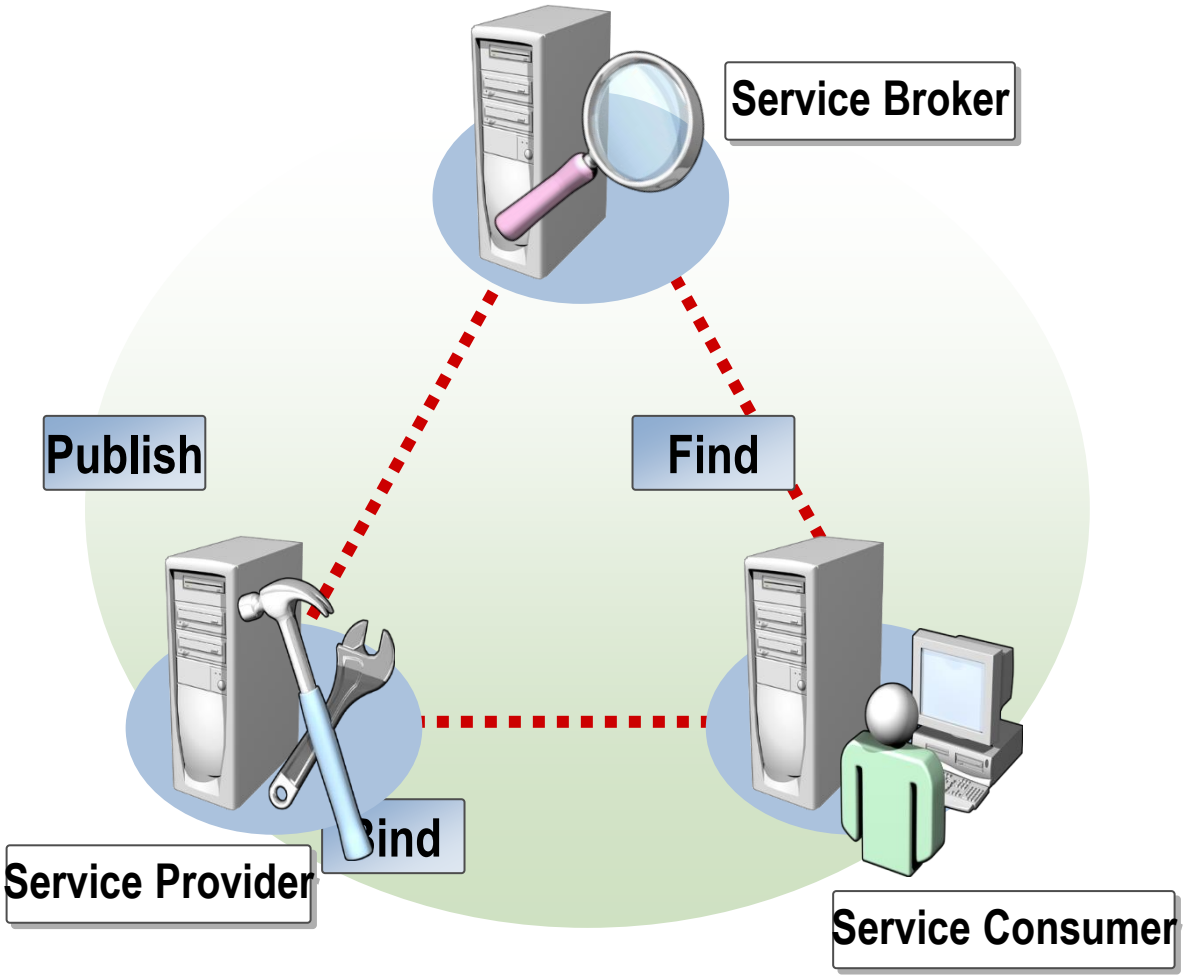
2. Sử dụng để tạo các ứng dụng phân tán



Các ứng dụng phân tán
Distributed Application



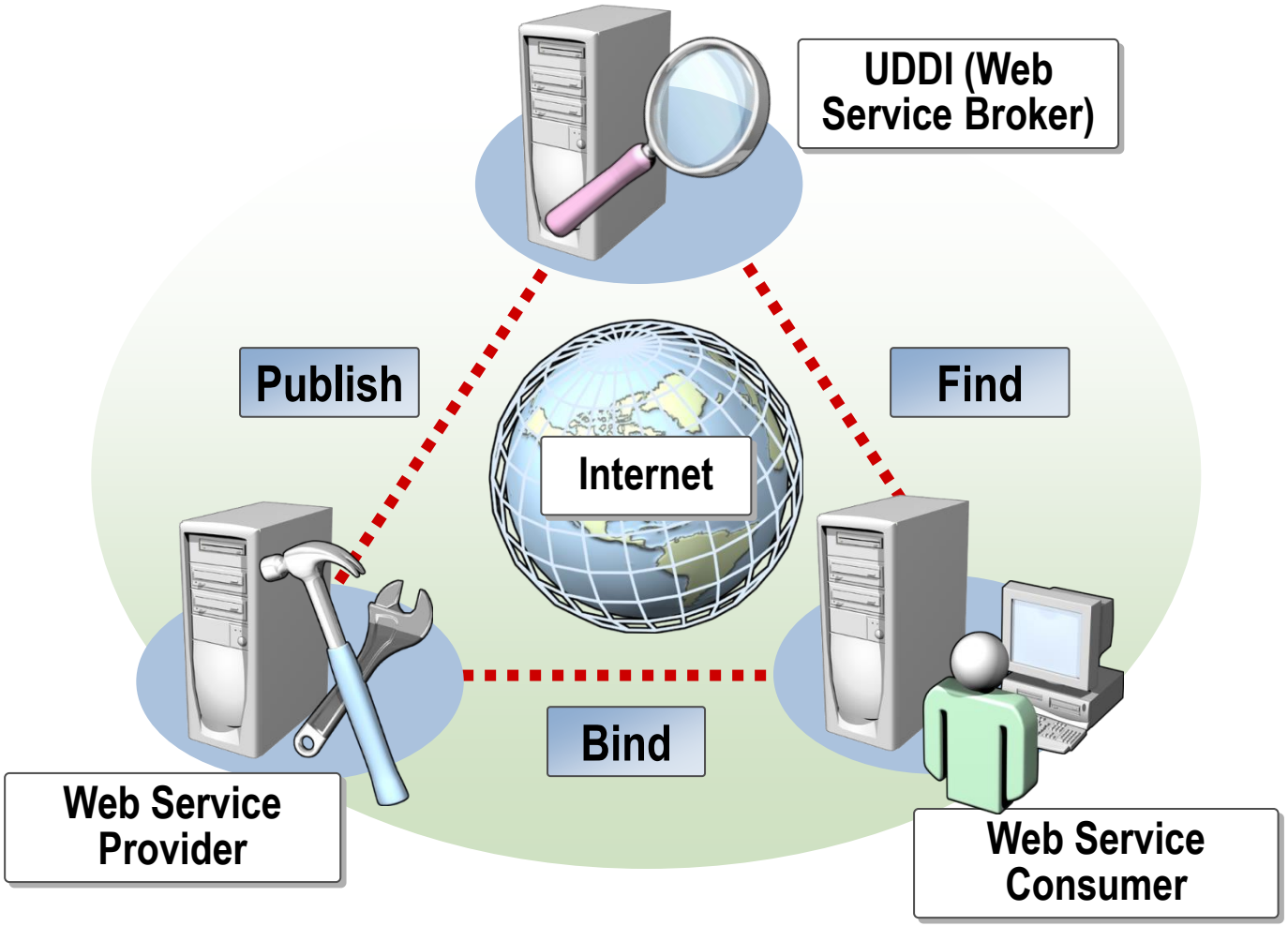
Kiến trúc hướng dịch vụ
Service-Oriented Architecture-SOA



1.2 Đặc điểm của Web Service

1. Không phụ thuộc vào ngôn ngữ lập trình
2. Truy cập bất cứ ứng dụng nào
3. Hỗ trợ thao tác giữa các thành phần không đồng nhất
4. Chi phí phát triển thấp
5. Dễ bảo trì

1.3 Nền tảng của Web Service

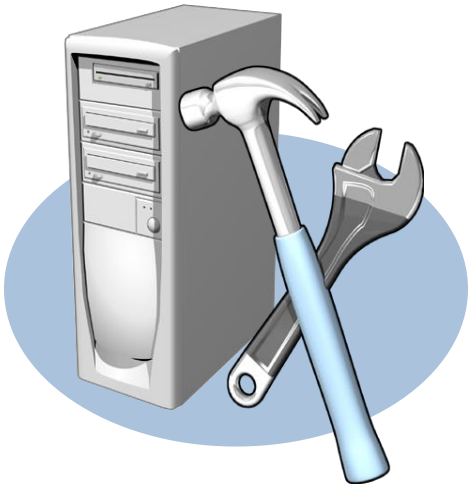


1.3 Nền tảng của Web Service

Web service provider

1. Web Server
2. Cung cấp Web service

IIS

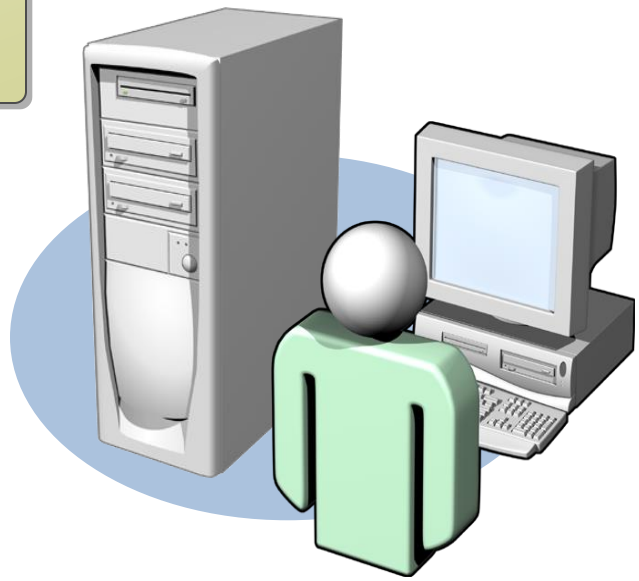


Yêu cầu

1. Hỗ trợ 1 protocol listenner
2. Có các cơ chế bảo mật
3. Cung cấp đúng service mà consumer yêu cầu

Web service consumer

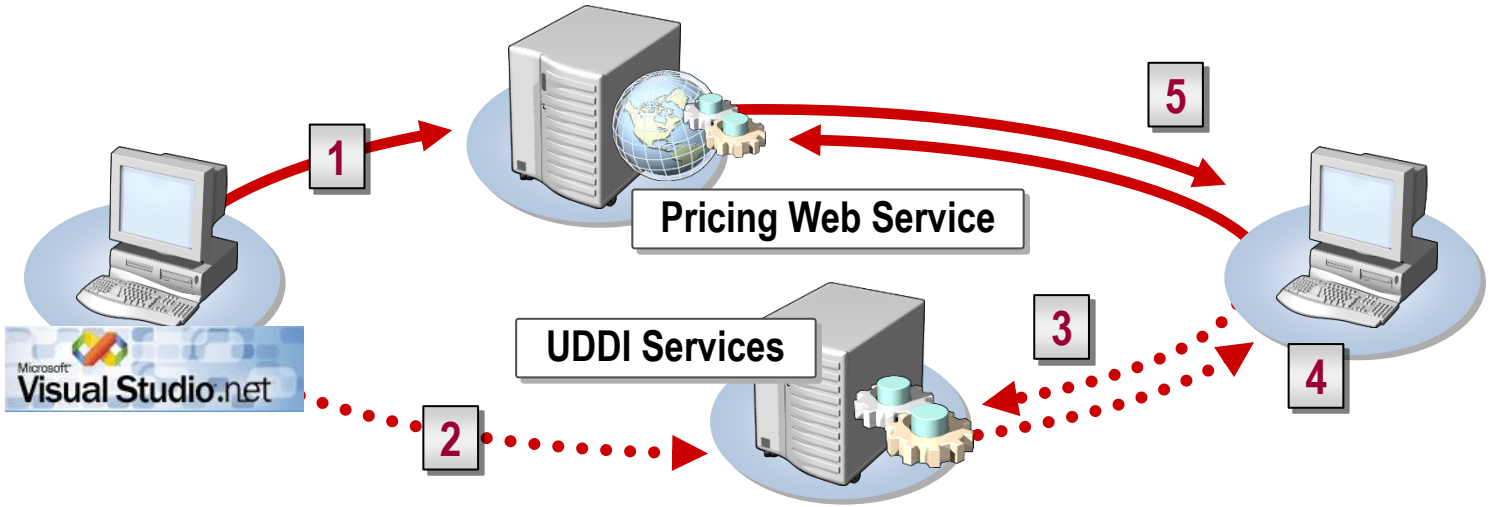
Sử dụng Web service do WS Provider cung cấp để xây dựng ứng dụng



Web service broker

1. Là lớp trung gian giữa consumer và provider
2. Tương tác với provider : để lấy các thông tin về web service
3. Tương tác với consumer :cung cấp cho consumer địa chỉ các Web service
4. Sử dụng UDDI registries (Universal Description, Discovery and Integration)

Mô hình tương tác giữa các thành phần



1. **Web service developer** builds and deploys a pricing Web service
2. **Web service developer** registers and categorizes the Web service
3. **Web service consumer** queries UDDI for "pricing" services
4. **Web service consumer** determines the most appropriate "pricing" service
5. **Web service developer** builds a solution that directly consumes the Web service data

1.4 Các công nghệ của Web Service

HTTP – Hyper text transfer protocol

HTTP Request

```
POST /TheStockExchange/Trading/GetStockPrice.asp HTTP/1.1
Host: localhost
Content-Type: application/x-www-form-urlencoded
Content-Length: 11

Symbol=MSFT
```

HTTP Response

```
HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: 75

<?xml version="1.0" encoding="utf-8"?>
<stock symbol="MSFT" Price="71.50" />
```

Ngôn ngữ XML – Extend Markup Language

1. Sử dụng để mô tả Web service interface
2. SOAP Message là một tài liệu XML
3. Sử dụng để serialize các đối tượng thành một tài liệu XML (System.Xml.Serialization)

Công nghệ SOAP (1)

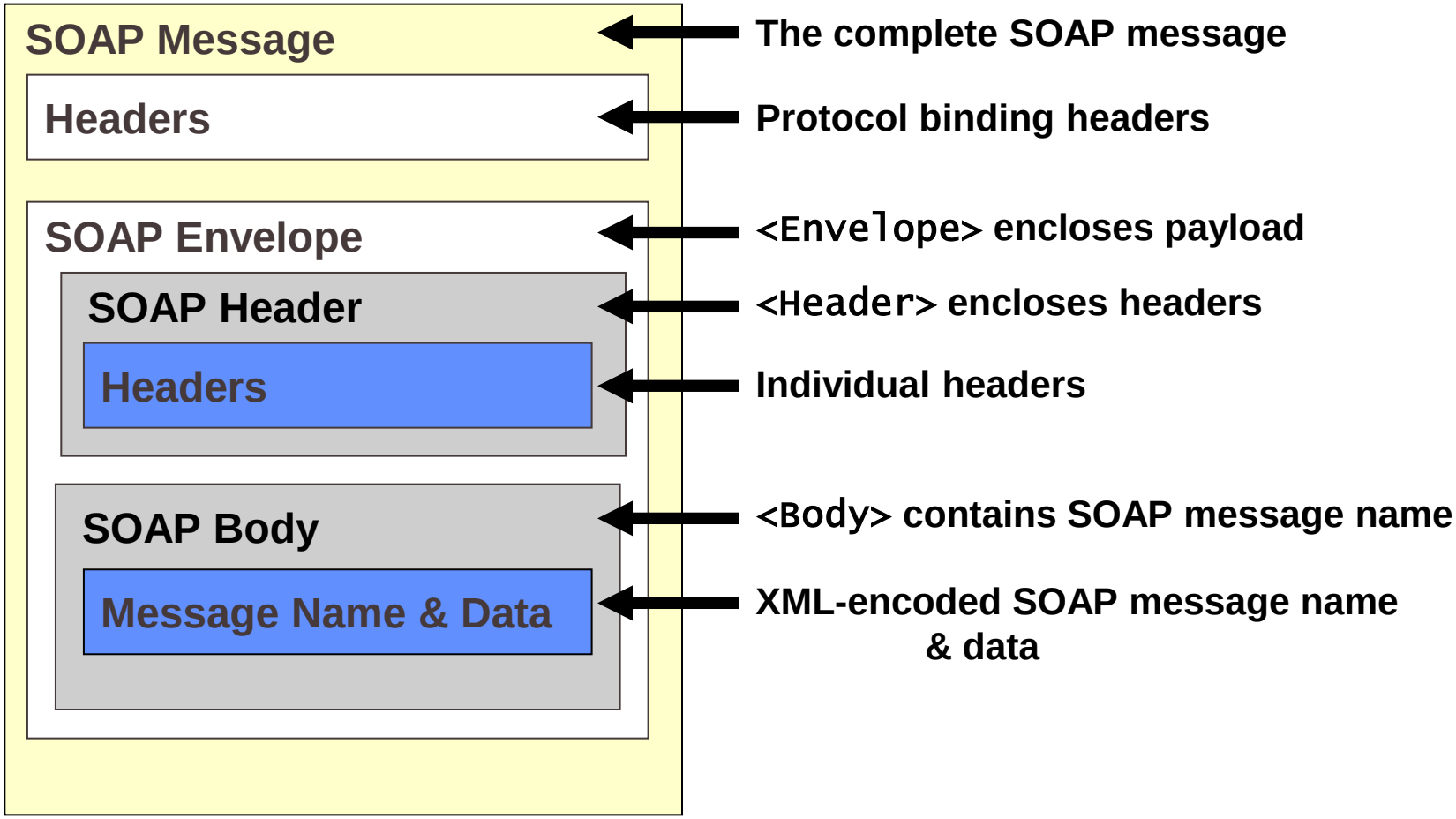
1. SOAP – Simple Object Access Protocol
2. Protocol for message-based communication
3. XML-Based protocol
   - An envelope for handling extensibility and modularity
   - An encoding mechanism for representing types within an envelope
4. SOAP ≈ HTTP + XML
5. Submitted to W3C

SOAP (2)

6 SOAP Request : HTTP POST Request

```
POST /WebCalculator/Calculator.asmx HTTP/1.1
Content-Type: text/xml
SOAPAction: “http://tempuri.org/Add”
Content-Length: 386

<?xml version=“1.0”?>
<soap:Envelope ...>
  ...
</soap:Envelope>
```

SOAP(3) - Message Structure



SOAP(4) - Example of a SOAP 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>
```

SOAP(5) - Example of a SOAP Response

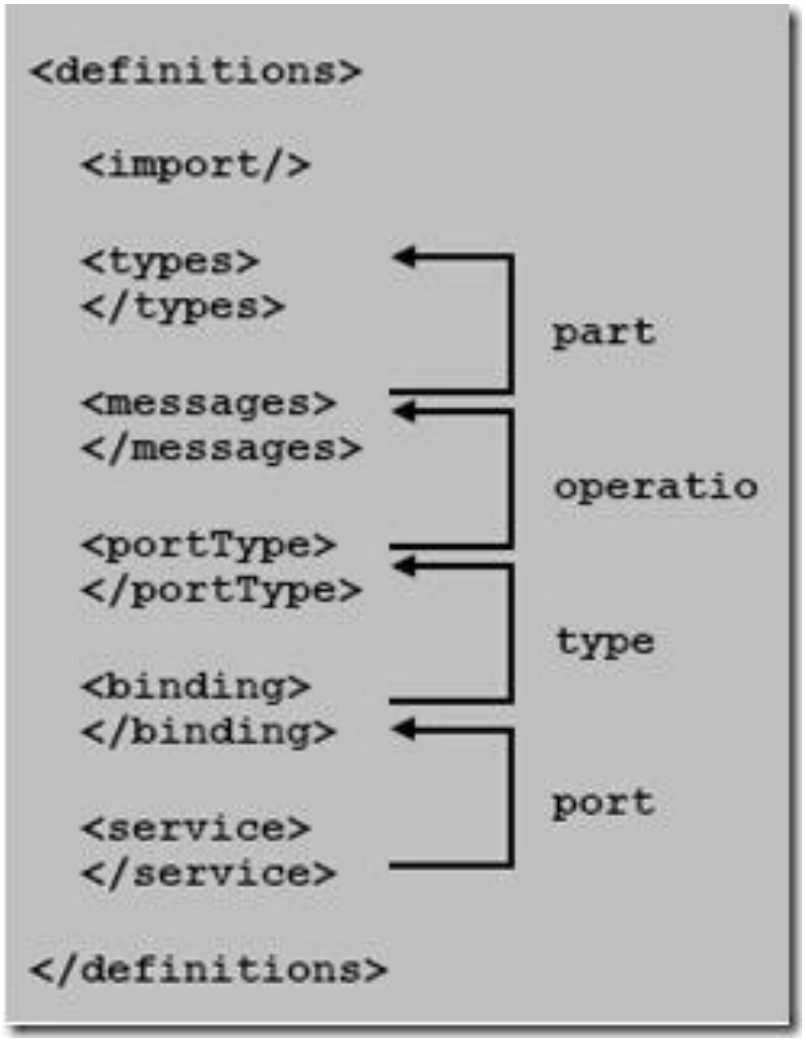
```
HTTP/1.1 200 OK
Content-Type: text/xml;
charset="utf-8"
Content-Length: nnnn

<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>
```

Ngôn ngữ mô tả Web Service – WSDL

Web Sevice Description Language (WSDL) định nghĩa một tài liệu XML mô tả giao diện của các dịch vụ web. Tài liệu WSDL này được sử dụng cho bên yêu cầu dich vụ (service requester). Bên yêu cầu dịch vụ sẽ sử dụng các thông tin về giao diện định nghĩa trong lược đồ WSDL để triệu gọi (invoke) dịch vụ web.

Phần tử gốc của tất cả các tài liệu WSDL luôn là phần tử <definitions>. Nó chứa bên trong sáu thành phần chia thành hai nhóm: Thông tin trừa tượng và thông tin cụ thể.

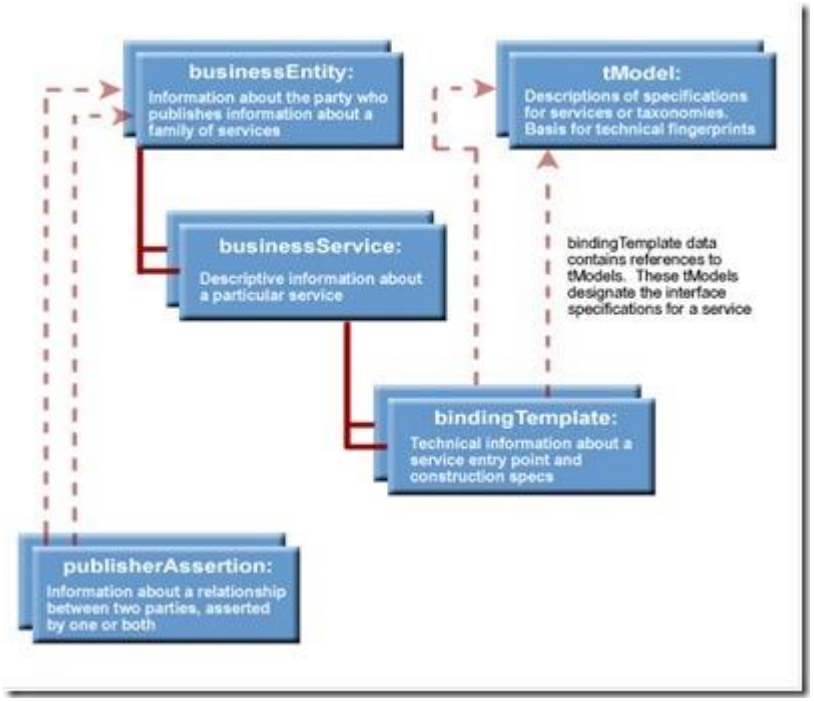
Ngôn ngữ mô tả Web Service – WSDL



UDDI – Universal Description , Discovery and Intergration

Universal Description, Discovery, and Intergration (UDDI) là một tập các quy tắc đăng ký và tìm kiếm thông tin các Web Service. Nó đóng vai trò như service broker cho phép người sử dụng dịch vụ tìm đúng nhà cung cấp dịch vụ cần tìm.

UDDI định nghĩa bốn loại thông tin cơ bản để kết nối đến Web service.



UDDI – Universal Description , Discovery and Intergration

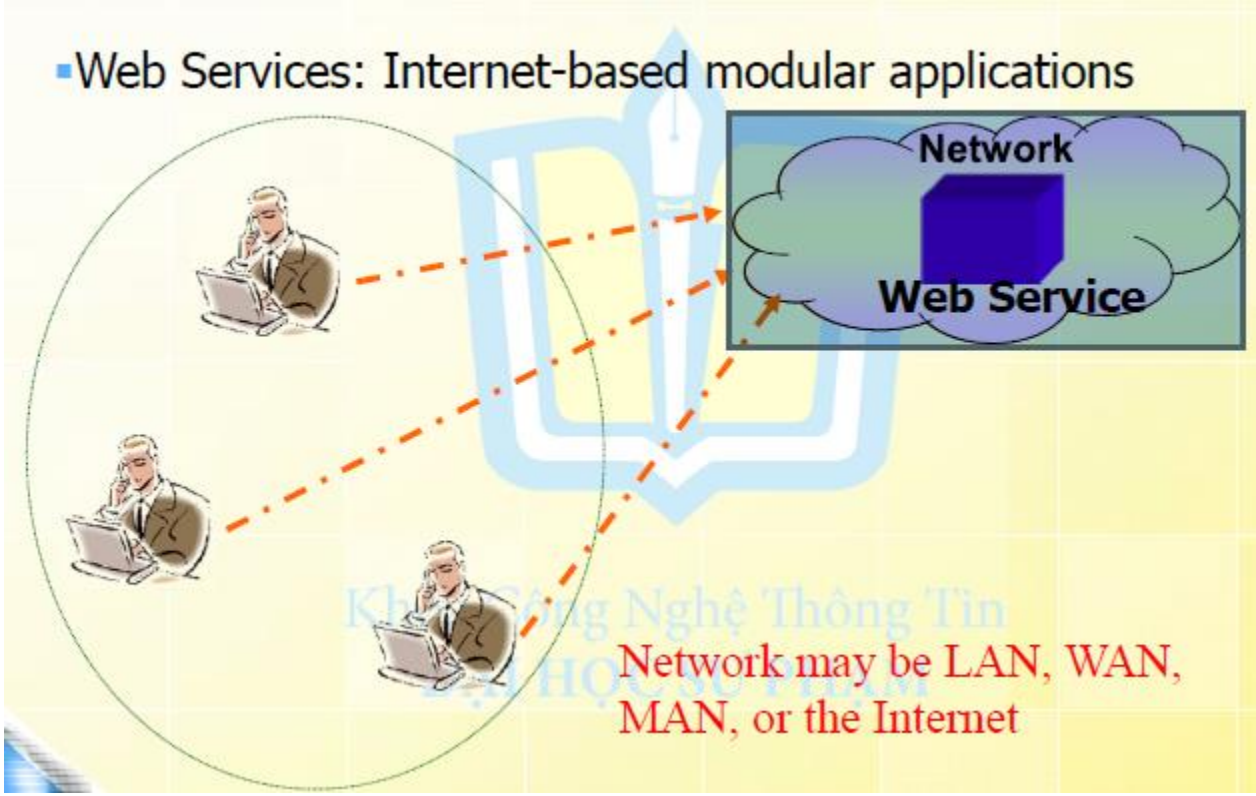
Cấu trúc thông tin bao gồm :

Business entity: Một business entity chứa thông tin về công bao gồm tên công ty, một đoạn mô tả ngắn gọn và một vài thông tin liên lạc cơ bản (địa chỉ, số điện thoại, email, v.v….).
Mỗi doanh nghiệp được cấp một định danh duy nhất, ví dụ như số D-U-N-S.

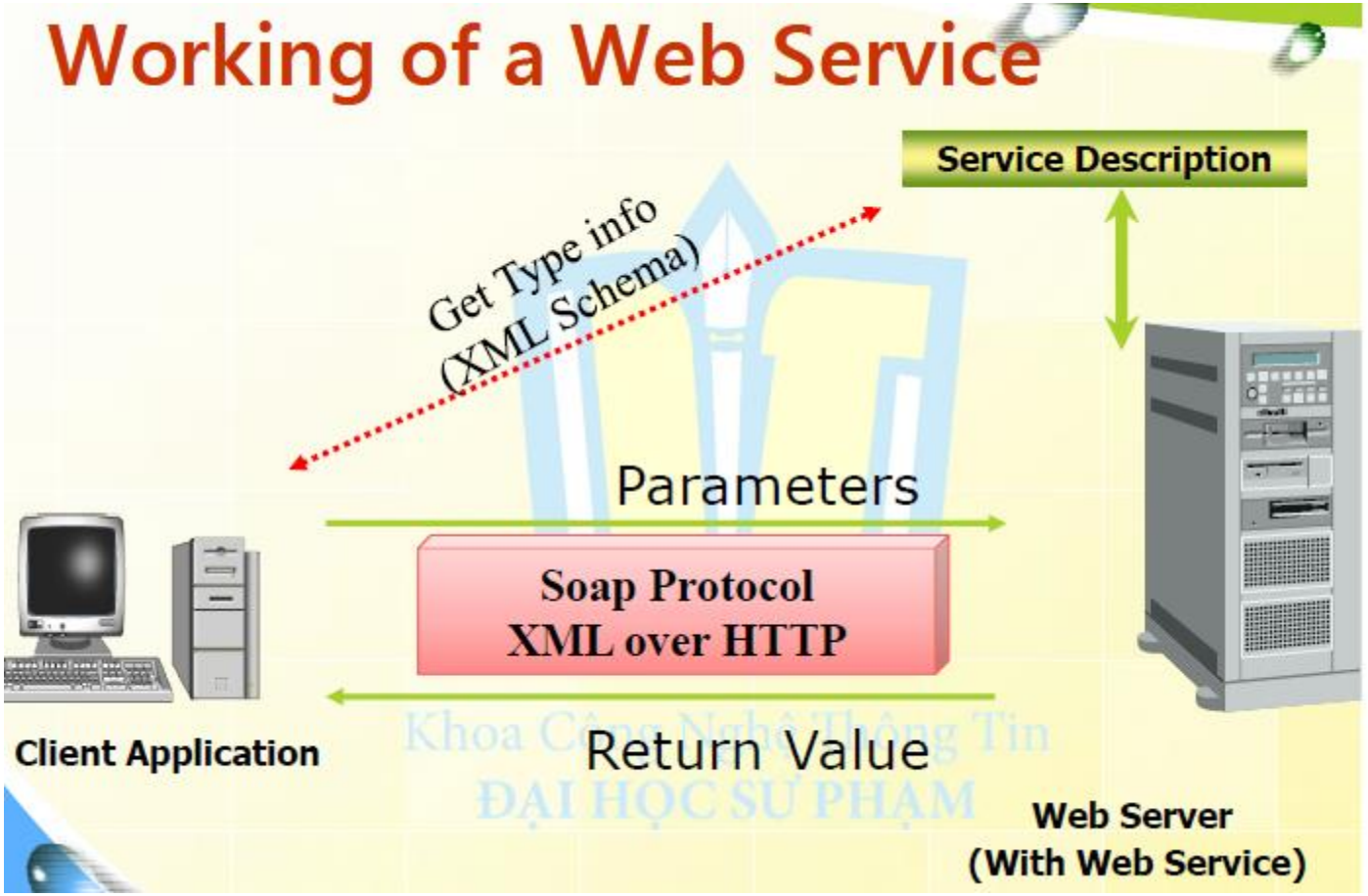
Business service: Liên kết với mỗi business entity là một danh sách các business service cung cấp bởi business entity đó. Mỗi thành phần chứa thông tin mô tả về dịch vụ, về thông tin phân loại của dịch vụ và danh sách các binding template liên quan đến thông tin kỹ thuật của dịch vụ. Mỗi business service cần có ít nhất một binding template.

Binding template: Gắn với mỗi business service là một danh sách các binding template cung cấp thông tin về địa điểm có thể tìm thấy Web Service và làm cách nào để sử dụng nó.

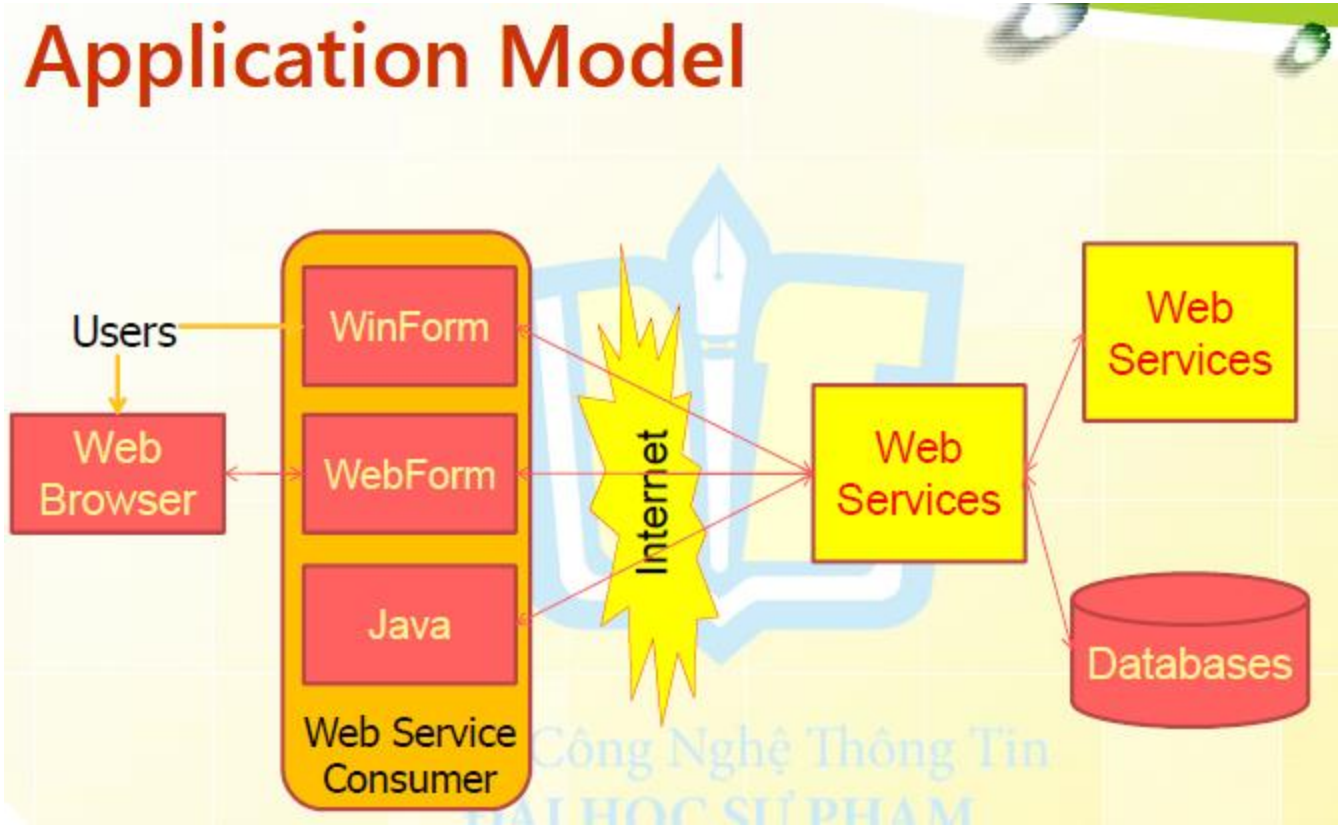
Chương 2: Kiến trúc Web Service



2.1 Cơ chế hoạt động



Cơ chế hoạt động



2.2 Kiến trúc phân tầng của Web Service

2.3 Kiến trúc hướng dịch vụ SOA

Chương 3: Xây dựng Web Service

1. Các vấn đề khi xây dựng Web Service
2. Xây dựng Web Service với ASP.NET
3. Xây dựng Web Service với Java

3.1 Các vấn đề khi Lập trình Web service (1)

1. Loại Project : ASP.NET Web Service
2. Các name space thường dùng
   - System
   - System.Data
   - System.Web
   - System.Web.Webservices
   - System.Xml
3. Các thành phần
   - File *.asmx
   - File global.asax
   - File web.config
   - Thư mục Bin

Lập trình Web service (2)

4 Khai báo Web service method

Thêm chỉ thị [Web Method]

Ví dụ

```
[WebMethod]
public int Sum(int a, int b)
{
    return a + b;
}
```

Lập trình Web service (3)

5 Các thuộc tính của [WebMethod]

- BufferResponse
- CacheDuration
- Description
- EnableSession
- TransactionOption

Lập trình Web service (3)

Ví dụ

```
[WebMethod(BufferResponse=false)]
public Transactions
GetTransactionHistory()
{
    //implementation code
}
```

```
[WebMethod (Description="This method using for converting
.." )]
public double ConvertTemperature(double dFahrenheit)
{
    return ((dFahrenheit - 32) * 5) / 9;
}
```

Lập trình Web service (4)

5 XML Serialization

Dùng để xác định cấu trúc tài liệu XML mong muốn để biểu diễn đối tượng của một lớp

Namespace : System.Xml.Serialization

- XmlRoot
- XmlElement
- XmlAttribute
- XmlArray
- XmlArrayItem

Lập trình Web service (5)

Ví dụ 1

```
[XmlRoot("account")]
public class Acct
{
        [XmlElement("description")]
        public string Description;
        [XmlElement("number")]
        public string Number;
        [XmlElement("type")]
        public string Type;
        [XmlElement("balance")]
        public decimal Balance;
        [XmlAttribute("status")]
        public string Status;
}
```

```
<account status="123">
  <description>Hello</description>
  <number>10  </number>
  <type>C</type>
<account>
```

Lập trình Web service (6)

Ví dụ 2

```
…
[return:XmlArray("AccountList")]
[return:XmlArrayItem("Account")]
public Acct[] GetAllAccounts()
…
```

```
<AccountList>
  <account>
  ……..
  <account>
  <account>
  ….
  </account>
<AccountList>
```

Lập trình Web service (7)

Ví dụ 3

```
…
[return:XmlArrayItem(ElementName="savingsAcct", Type=typeof(SavingsAcct))]
[return:XmlArrayItem(ElementName="creditCardAcct", Type=typeof(CreditCardAcct))]
[WebMethod]
public Acct[] GetAllAccounts()
{
...
```

```
<AccountList>
        …
        <SavingAcct>
        </SavingActt>
        …
        <CreditCardAcct>
        </CreditCardActt>
        …
</AccountList>
```

Lập trình Web service (8)

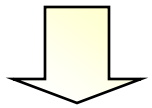
6 Kiểu dữ liệu trong web service method

- Simple data types
- Input and output parameter (in, ref,…)
- Variable length parameter list (*homogeneous, heterogeneous*)
- Complex data type
  - Classes and structures
  - Array, Collections
  - DataSets

Lập trình Web service – provider side(9)

Trường hợp lưu ý

Nếu tham số web service method là 1 mảng các đối tượng có kiểu không rõ ràng



→ Phải khai báo tường minh tất cả các kiểu dữ liệu có thể có trong mảng

→Sử dụng :XmlInclude
Name space System.Xml.Serialization

Lập trình Web service – provider side(10)

Ví dụ

```
public class Acct
{
    public string Description;
    public string Number;
    public string Type;
    public decimal Balance;
    public string Status;
}
public class SavingsAcct : Acct
{
    public decimal MinimumBalance;
}
public class CreditCardAcct : Acct
{
    public int PayPeriod;
}
```

```
[WebMethod]
[XmlInclude(typeof(CreditCardAcct))]
[XmlInclude(typeof(SavingsAcct))]
[return:XmlArray("AccountList")]
[return:XmlArrayItem("Account")]
public Acct[] GetAllAccounts()
{
SavingsAcct a = new SavingsAcct();
CreditCardAcct cc = new
CreditCardAcct();
// populate the accounts
Acct [] sa = new Acct[2];
sa[0] = a;
sa[1] = cc;
return sa;
}
```

Triển khai Web service

1. Tự chép các file cần thiết lên Web server

2. Sử dụng Windows Installer Files
   Tạo 1 web setup project trong VS.NET

Web Service Files	Needed	Remove
.sln, .vbproj, .csproj, .vsdisco, .webinfo		✓
.resx		✓
.vb, .cs		✓
.xsd		✓
\Web References folder and files		✓
\bin directory and .dll(s)	✓	
.asmx	✓	
Web.config	✓	
Global.asax	✓	
.xml	✓	

Web Service Consumer

1. WSDL Documents
2. Proxy class
3. Consuming Web service in VS.NET 2010

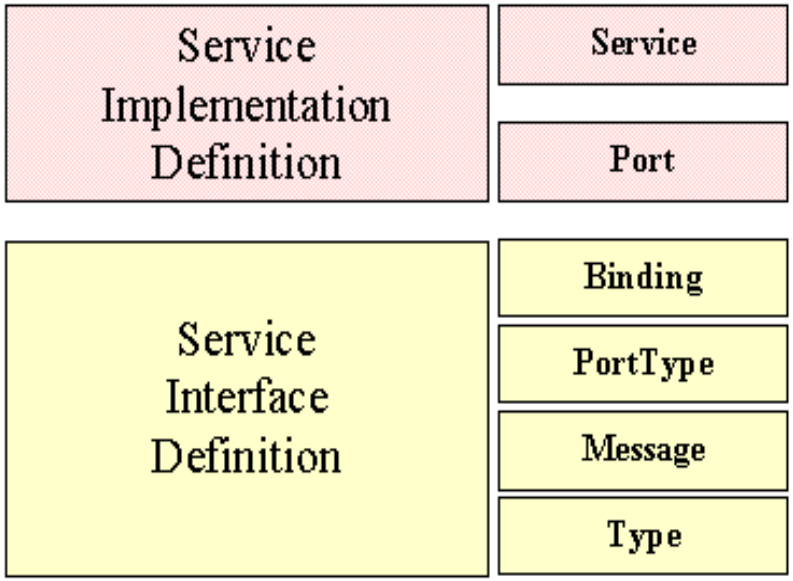
WSDL Documents

1. **Web Service Description Language**

2. **XML schema : Mô tả giao diện của web service**

3. **Sử dụng đề phát sinh proxy class**

4. **Xem WSDL Document của 1 web service**
   http://localhost/TestService/service.asmx?wsdl



Proxy class

1. Proxy : Ủy nhiệm
2. Được phát sinh từ WSDL Document
3. Được sử đụng đề giao tiếp với Web service

3.2 Xây dựng Web service trong .NET

1 Tạo lớp proxy để giao tiếp với web service

Sử dụng wsdl.exe

wsdl [*options*] {*URL* | *Path*}

wsdl http://www.woodgrovebank.com/services/bank.asmx?wsdl

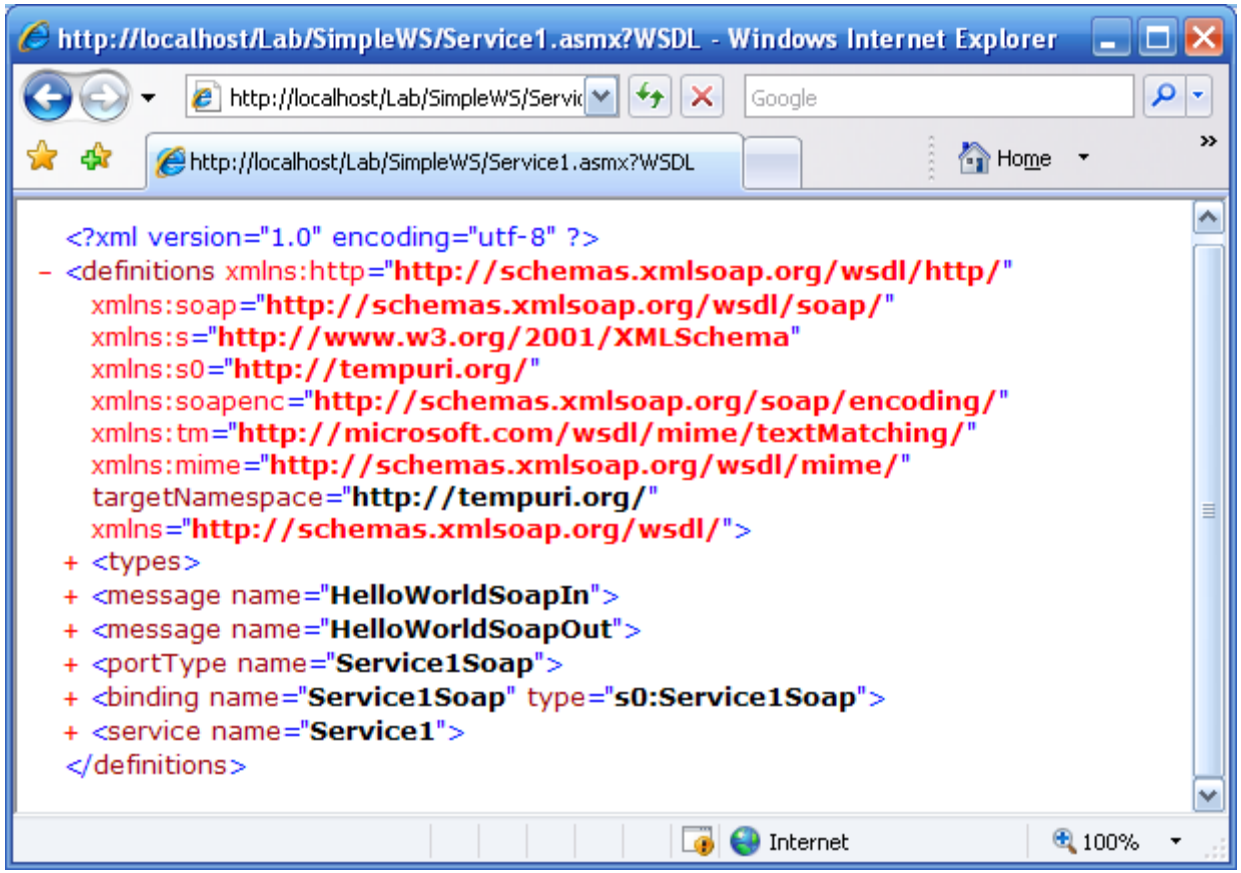
Sử dụng VS.Net

Add Web References …

2 Sử dụng lớp proxy để gọi các phương thức của web service

Gọi bình thường như các phương thức khác

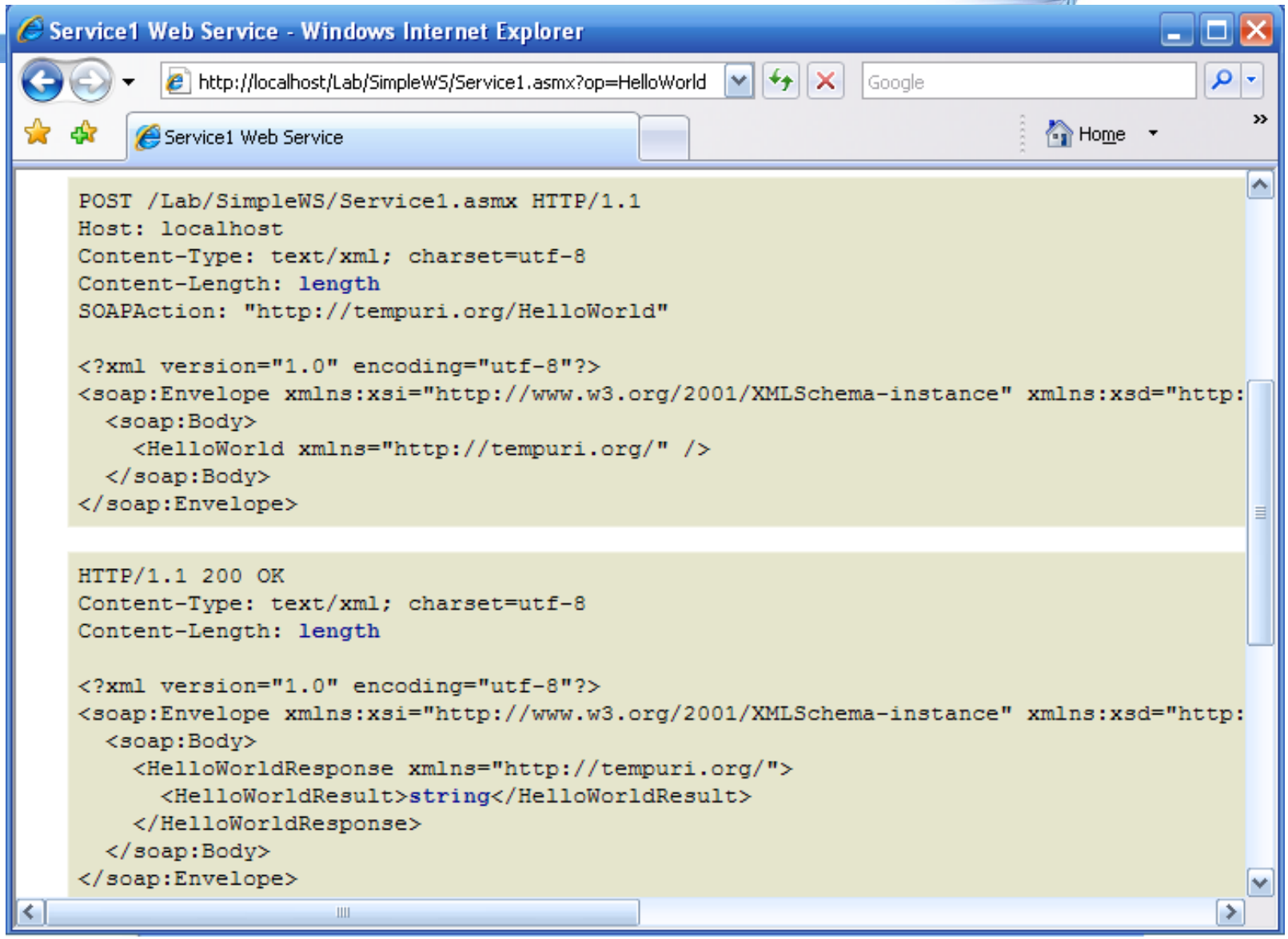
WSDL Document



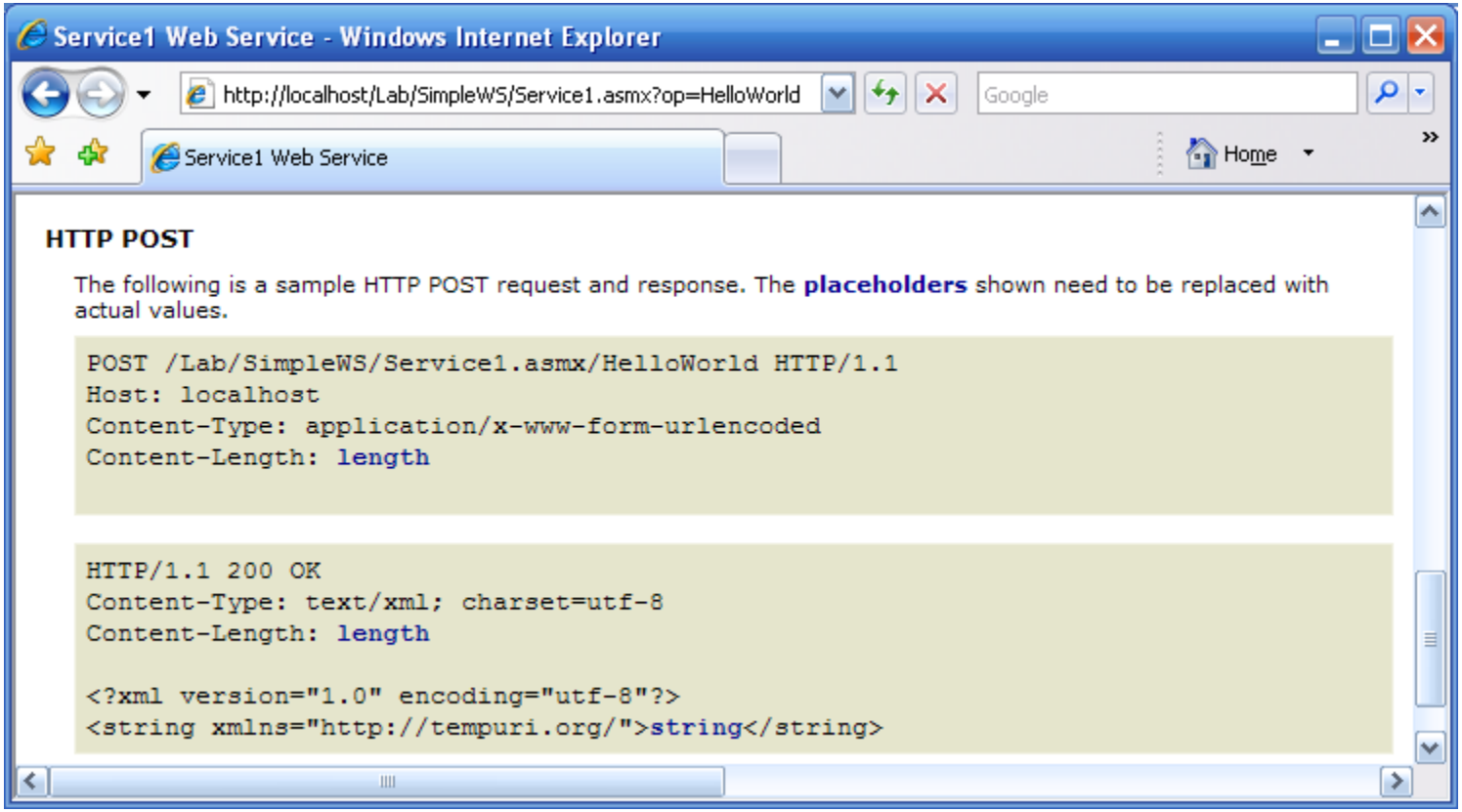
Proxy Class

```
[System.Diagnostics.DebuggerStepThroughAttribute()]
[System.ComponentModel.DesignerCategoryAttribute("code")]
[System.Web.Services.WebServiceBindingAttribute(Name="Service1Soap", Na
public class Service1 : System.Web.Services.Protocols.SoapHttpClientPro

    /// <remarks/>
    public Service1() {
        this.Url = "http://localhost/Lab/SimpleWS/Service1.asmx";
    }

    /// <remarks/>
    [System.Web.Services.Protocols.SoapDocumentMethodAttribute("http://
    public string HelloWorld() {
        object[] results = this.Invoke("HelloWorld", new object[0]);
        return ((string)(results[0]));
    }

    /// <remarks/>
    public System.IAsyncResult BeginHelloWorld(System.AsyncCallback cal
        return this.BeginInvoke("HelloWorld", new object[0], callback,
    }
```

SOAP Request and Response

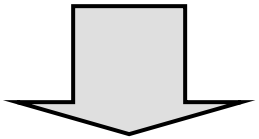


HTTP Request and Response



Web service and State Management

1. ASP.NET Web service Application → Web Applicaton
2. Web service Application : Stateless



Sử dụng đối tượng Application và Session đề quản lý trạng thái của web service

Ứng dụng của Web Services

1. WS là một tập hợp các chức năng được thực thi qua môi trường mạng, được các ứng dụng khác sử dụng
2. Một WS có thể kết hợp với các WS khác để đưa ra tính năng cao hơn
3. Mô hình ứng dụng trong tương lai sẽ là sự kết hợp giữa các service
4. …

3.3 Xây dựng Web Service với Java

Giới thiệu và Demo

Demo bài tập xây dựng Web Service với ASP.NET (C#)

Chương 4: Khai thác Web Service

- 1. Ứng dụng Window Form kết nối tới Web Service
- 2. Ứng dụng Java Swing kết nối tới Web Service .
- 3. Ứng dụng Web ASP.NET kết nối tới Web Service

4.1 Ứng dụng Window Form kết nối tới Web Service

Demo

Ứng dụng Web ASP.NET kết nối tới Web Service

- Demo

4.2 Ứng dụng Java Swing kết nối tới Web Service

- Demo

4.3 Ứng dụng Web ASP.NET kết nối tới Web Service

Demo

Chương 5: Bảo mật trong Web Service

- 1. Tổng quan về vấn đề bảo mật.
- 2. Một số kiểu giả mạo, đánh cắp thông tin và cách phòng chống.
- 3. Bảo mật trong Web Service.
- 4. Thực hiện bảo mật trong Web Service

1. Tổng quan về vấn đề bảo mật.

- Cùng với sự phát triển không ngừng của Internet, hệ thống thông tin ngày một phát triển vượt bậc. Khái niệm business không còn giới hạn ở bên trong mà đã phát triển cả ở trên Internet. Khái niệm thương mại điện tử ra đời là sự minh chứng cho sự phát triển đó.Tuy nhiên, đi cùng với sự phát triển đó thì vấn đề an toàn của nó cũng là một vấn đề nóng và cấp bách.Tuy không có khái niệm về sự an toàn tuyệt đối nhưng cũng phải cần một cơ chế an toàn thích hợp để khai thác thương mại trên Internet.Các giải pháp hiện nay đang được sử dụng như mã hóa khóa, chữ ký số có thể đảm bảo ở một mức nào đó. Nhưng cũng phải cần không ngừng hoàn thiện và phát triển các giải pháp đó để đảm bảo phù hợp hơn trên môi trường Internet. An toàn thông tin trên Internet là một vấn đề chung hiện nay.Sự an toàn web service lại càng cần được sự quan tâm hơn nữa, khi các thông tin nhạy cảm như tài khoản cá nhân ở ngân hàng có thể bị đánh cắp.

1. Tổng quan về vấn đề bảo mật.

- Điều gì xảy ra nếu sử dụng business service mua hàng, chứng khoán,chuyển tiền mà không có một sự bảo đảm an toàn cần thiết. Đây là một chuẩn an toàn chung cần thiết cần được đáp ứng:
- Identification: định danh được những ai truy cập tài nguyên hệ thống.
- Authentication: chứng thực tư cách truy cập tài nguyên của người muốn sử dụng.
- Authorization: cho phép giao dịch khi đã xác nhận định danh người truy cập.
- Mục tiêu:
- Hiểu về vấn đề bảo mật trên Internet và Web Service đáp ứng chuẩn an toàn cơ bản
- Hiểu về một số kiểu giả mạo, đánh cắp thông tin và cách phòng chống
- Thực hiện bảo mật Web Service với thư viện WSE 3.0 do Microsoft cung cấp.
- Xây dựng dịch vụ Web bảo mật với Visual Studio - ASP.NET, triển khai và công bố dịch vụ Web.
- Đồng thời xây dựng ứng dụng để khai thác và sử dụng chuẩn kết nối bảo mật vừa xây dựng

1. Tổng quan về vấn đề bảo mật.

- Integrity: toàn vẹn thông tin trên đường truyền. - Confidentiality: độ an toàn, không ai có thể đọc thông tin trên đường đi. - Auditing: kiểm tra, tất cả các giao dịch đều được lưu lại để kiểm tra. - Non-repudiation: độ mềm dẻo, cho phép chứng thực tính hợp pháp hóa của thông tin đến từ một phía thứ ba ngoài 2 phía là người gửi và người nhận. Những yêu cầu trên giúp cho hệ thống an toàn hơn, tránh được phần nào những truy cập không hợp lệ. HTTP (HyperText Transfer Protocol) là giao thức thường sử dụng nhất cho việc trao đổi thông tin trên Internet, tuy nhiên không an toàn, bởi vì tất cả thông tin được gửi dưới dạng văn bản trong mạng ngang hàng không an toàn
- HTTP thuộc về nhóm của những nghi thức như SMTP, telnet, và FTP, được thiết kế trong giai đoạn đầu của Internet khi mà vấn đề an toàn chưa được quan tâm đến nhiều. Một phát triển của HTTP là HTTPS, nó là một chuẩn an toàn cho HTTP. HTTPS cho phép chứng thực client và server qua những chứng thực giữa client và server. Trước khi có web servives security (WS-Security) thì ý nghĩa thông thường của an toàn web service là bảo mật kênh truyền dữ liệu. Nó được thực hiện cho những SOAP/HTTP dựa trên cơ chế truyền thông điệp bằng cách sử dụng giao thức HTTPS. Không giống sự an toàn mức thông điệp, HTTPS cung cấp sự an toàn tới toàn bộ gói dữ liệu HTTP. Bởi vậy, chúng ta không có một tùy chọn nào để áp dụng sự an toàn có chọn lọc chỉ trên những thành phần của một thông điệp. Mặc dầu HTTPS không bao phủ tất cả các khía cạnh trong chuẩn an toàn chung nhưng nó cũng đã cung cấp một mức bảo chứng đầy đủ với định danh và chứng thực, sự toàn vẹn thông điệp, và độ tin cậy. Tuy nhiên, authentication, auditing, and nonrepudiation chưa được cung cấp. Bên cạnh đó, HTTPS là một giao thức nên khi thông điệp đi qua HTTP server thì lại không an toàn.

2. Một số kiểu giả mạo, đánh cắp thông tin và cách phòng chống.

- Message Replay Attack Message Replay là một kiểu tấn công trên mạng kiểu bắt và gửi lại gói tin mà client đã gửi cho server.
- Trong quá trình này Attacker như một trung gian của client và server, nó bắt gói tin client và sao chép gói tin đó,chỉnh sửa và gửi lại cho server.Và nếu server có gửi lại cho client thì Attacker cũng có thể bắt được. Ta phải ngăn chặn việc gửi lại gói tin kiểu này bởi vì các gói tin gửi đi sẽ gây ra không nhất quán dữ liệu, các thông tin gửi đi bị sai lệch ảnh hưởng đến client,việc gửi replayed message liên tục và một cách tự động sẽ làm chết server.
- Giải pháp cho vấn đề này là sử dụng cache lưu lại tên định danh cho message và server sẽ loại bỏ các message có định danh bị trùng.Như vậy các message có một tên định danh duy nhất được gửi đi và chắc chắn rằng các message không bị giả mạo trong quá trình truyền. Web service security đã cung cấp việc sử dụng Usernametoken trong đó có thẻ username và password. <wsse:UsernameToken wsu:Id="Example-1"> <wsse:Username>... </wsse:Username> <wsse:Password Type="...">... </wsse:Password> <wsse:Nonce EncodingType="...">... </wsse:Nonce> <wsu:Created>... </wsu:Created> </wsse:UsernameToken> Các giá trị Nonce được tạo ra ngẫu nhiên cùng với Created là thời gian mà message được gửi đi.Các giá trị này đều được Hash để che dấu thông tin. Giá trị thời gian mà message phải được đồng bộ với thời gian của server.Khi gói tin đến server thì server sẽ giới hạn khoảng thời gian mà message đến,nếu vượt ra khoảng thời gian giới hạn gói tin bị loại bỏ

2. Một số kiểu giả mạo, đánh cắp thông tin và cách phòng chống.

- Web Spoofing Web spoofing là một kiểu lừa đảo trên web phổ biến hiện nay.Có 4 kiểu web spoofing là IP spoofing,ARP spoofing,Web spoofing,DNS spoofing. a. Ip spoofing Chứng minh mình thực sự tin cậy với một máy khác nhằm hợp pháp việc truy cập máy đó và có thể lấy được username, password (có máy giả mạo giả mạo server hoặc client). Máy khác trở thành đồng phạm vì đã để cho máy giả mạo sử dụng IP giả mạo trùng với IP của máy đó.Kiểu tấn công này thường xảy ra ở các máy client khi độ bảo mật an toàn ở các máy đó không được cao. Để tránh kiểu tấn công này thì server và client đều có định danh xác thực lẫn nhau.
- Web spoofing: Người giả mạo tạo ra bản sao của một trang web và đánh lừa người dùng click vào. Cách thực hiện :đặt trang web giả mạo liên kết với trang web thông dụng. Khi sử dụng mail kích hoạt web mail,attacker gửi mail đến giới thiệu link đến web giả. Kẻ giả mạo còn đánh lừa máy tìm kiếm web để nó trỏ vào web giả. Khi user request đến server và sever trả về url của web đó. Người giả mạo có thể hướng người dùng truy cập đến trang web của họ để dễ dàng đánh cắp thông tin. Đây là kiểu tấn công thường xuyên và phổ biến hiện naykhi lừa một cách trực tiếp đến người dùng cuối.Cách này thông thường kẻ giả mạo đánh lừa người dùng cuối chứ khó có thể giả mạo client thực sự của web service.
- DNS spoofing Cách hoạt động của DNS server : Giao thức Domain Naming System (DNS) như được định nghĩa trong RFC 1034/1035 có thể được xem như là một trong những giao thức quan trọng nhất được sử dụng trong Internet. Nói ngắn ngọn để dễ hiểu, bất cứ khi nào bạn đánh một địa chỉ web chẳng hạn như http://www.google.com vào trình duyệt, yêu cầu DNS sẽ được đưa đến máy chủ DNS để tìm ra địa chỉ IP tương xứng với tên miền mà bạn vừa nhập. Các router và các thiết bị kết nối Internet sẽ không hiểu google.com là gì, chúng chỉ hiểu các địa chỉ chẳng hạn như 74.125.95.103. Máy chủ DSN làm việc bằng cách lưu một cơ sở dữ liệu các entry (được gọi là bản ghi tài nguyên) địa chỉ IP để bản đồ hóa tên DNS, truyền thông các bản ghi tài nguyên đó đến máy khách và đến máy chủ DNS khác. Kiến trúc máy chủ DNS trong toàn doanh nghiệp và Internet là một thứ khá phức tạp. Như một vấn đề của thực tế, bạn có thể hình dung chúng như các quyển sổ chuyên dụng cho kiến trúc DNS. Người giả mạo cố tình cung cấp sai thông tin DNS để lừa nạn nhân truy cập vào 1 địa chỉ mà attacker chỉ định nhằm đánh cắp các thông tin như TK ngân hàng, … Giải pháp cho vấn đề này là cung cấp một DNS server tin cậy và khi sử dụng dịch vụ DNS phải có chứng thực

2. Một số kiểu giả mạo, đánh cắp thông tin và cách phòng chống.

- ARP spoofing Arp là giao thức hoạt động ở tầng 2 cho phép ánh xạ địa chỉ MAC. Giao thức ARP được thiết kế để phục vụ cho nhu cầu thông dịch các địa chỉ giữa các lớp thứ hai và thứ ba trong mô hình OSI. Lớp thứ hai (lớp data-link) sử dụng địa chỉ MAC để các thiết bị phần cứng có thể truyền thông với nhau một cách trực tiếp. Lớp thứ ba (lớp mạng), sử dụng địa chỉ IP để tạo các mạng có khả năng mở rộng trên toàn cầu. Lớp data-link xử lý trực tiếp với các thiết bị được kết nối với nhau, còn lớp mạng xử lý các thiết bị được kết nối trực tiếp và không trực tiếp. Mỗi lớp có cơ chế phân định địa chỉ riêng và chúng phải làm việc với nhau để tạo nên một mạng truyền thông. Với lý do đó, ARP được tạo với RFC 826, “một giao thức phân định địa chỉ Ethernet - Ethernet Address Resolution Protocol”. Quá trình truyền ARP bao gồm request và respone. Khi gói Arp request được gửi đi kẻ giả mạo bắt được thông tin đó và gửi trả về cho người gửi và bảng ARP của máy đó sẽ cập nhật địa chỉ của giả mạo. Cách phòng chống vấn đề này là đảm bảo mức an toàn ở mạng nội bộ. Vẫn còn một số kiểu lừa đảo qua mail nhưng kiểu giả mạo lừa đảo đó thì phải cần sự cảnh giác của client
- SSL spoofing: SSL và HTTPS Phá hủy cơ chế HTTPS: HTTPS là một kênh thông tin an toàn.Khi đã thiết lập được một kết nối từ máy chủ đến máy khách thì rất khó để có thể phá vỡ được kết nối đó.Tuy nhiên mọi việc đều có thể xảy ra khi máy khách vẫn chưa thật sự kết nối được với máy chủ. Kết nối HTTPS an toàn tuy nhiên ta có thể phá vỡ cấu trúc đó trước khi kết nối được thiết lập bằng phương pháp “man in the middle”.Chặn request từ client đến server để trở thành kẻ thứ ba trong phiên kết nối. Quá trình thực hiện như sau:

2. Một số kiểu giả mạo, đánh cắp thông tin và cách phòng chống.

- Lưu lượng giữa máy khách và máy chủ đầu tiên sẽ bị chặn. Khi bắt gặp một HTTPS URL, sslstrip sẽ thay thế nó bằng một liên kết HTTP và sẽ ánh xạ những thay đổi của nó. Máy tấn công sẽ cung cấp các chứng chỉ cho máy chủ web và giả mạo máy khách. Lưu lượng được nhận trở lại từ website an toàn và được cung cấp trở lại cho máy khách. Quá trình làm việc khá tốt, máy chủ có liên quan vẫn nhận lưu lượng SSL mà không hề biết về sự khác biệt này. Nếu không để ý thì rất khó phát hiện điều dị thường khi kết nối. 3. Bảo mật trong web service Ngày nay công nghệ web services đã và đang được triển khai và ứng dụng trong rất nhiều lĩnh vực khác nhau bao gồm cả những lĩnh vực nhạy cảm, đòi hỏi tính an toàn cao như tài chính, ngân hàng,… Do đó web service cần cung cấp một mức an toàn đủ để hỗ trợ những công việc như thế. Bên cạnh mặt được của công nghệ web services mang lại thì việc đảm bảo an toàn, tin cậy, toàn vẹn thông tin trao đổi trên web service cũng là một điều rất quan trọng trong quá trình xây dựng web services, bằng việc sử dụng ws security và các thành phần của nó giúp cho thông tin trao đổi trên web services trở nên an toàn hơn. Trước hết chúng ta xem xét những nhân tố rủi ro ảnh hưởng đến mức an toàn của những ứng dụng dựa trên web service.

2. Một số kiểu giả mạo, đánh cắp thông tin và cách phòng chống.

- Chúng ta sẽ sử dụng một kịch bản rút tiền ngân hàng qua mạng để xem xét vấn đề. Đây là một ứng dụng client/ server đơn giản mô tả một người rút tiền (client) kết nối tới trung tâm dữ liệu của ngân hàng để sử dụng một ứng dụng web service để thực hiện yêu cầu của mình. Nếu không có sự an toàn nào đã được áp dụng, thì có ba nhân tố mạo hiểm chính: - Những giao dịch không hợp pháp (Unauthorized transactions) : một người nào đó không có quyền nhưng vẫn yêu cầu rút tiền. Giao dịch này không hợp pháp. Chúng ta cấm vấn đề này bằng cách sử dụng cơ chế chứng thực của WS - Security. Một ví dụ của sự chứng thực bao gồmphải có một kết hợp user ID/ password trong SOAP message. - Những thông báo không mã hóa (Readable messages in clear text-no encryption): số hiệu tài khoản và số dư tài khoản trong gói SOAP rất dễ bị đọc lén trên mạng. Việc lộ thông tin này là do thông tin tài khoản và số dư được gửi qua mạng dưới định dạng văn bản.Để giải quyết vấn đề này, thông tin này phải được mã hóa ở mức kênh chuyển thông điệp hoặc ở mức thông điệp (WS - Security). - Những thông điệp bị thay đổi hoặc mất mát (SOAP message susceptible to modification-no integrity).Trong quá trình chuyển thông tin từ người rút tiền đến trung tâm dữ liệu, nó có thể bị chặn. Và những thông tin này có thể bị thay đổi, ví dụ như số tài khoản là 1234 thì bị thay đổi thành số 9876

2. Một số kiểu giả mạo, đánh cắp thông tin và cách phòng chống.

- Vấn đề này dẫn đến thiếu sự toàn vẹn. Những ví dụ trên, chúng ta đã mô tả sự an toàn liên quan tới những yêu cầu của sự chứng thực, tính bí mật, và sự toàn vẹn thông tin. Trước khi có WS-Security, sự an toàn kênh chuyển thông điệp rất thường được sử dụng. Sự an toàn kênh chuyển thông điệp ở chỗ là nó mã hóa toàn bộ thông điệp, dẫn đến sử dụng CPU cao hơn. Tuy nhiên với WS-Security, nó cung cấp những cách tối ưu hóa những thao tác an toàn, mà yêu cầu ít thời gian sử dụng CPU hơn. Dựa vào mức an toàn cần thiết mà một hoặc nhiều hơn những cơ chế an toàn này có thể được áp dụng cho một ứng dụng. Tuy nhiên việc chọn cơ chế an toàn cho web service phải đòi hỏi sao cho người dùng không cảm thấy quá phức tạp tạo một sự gò bó, do đó việc chọn cơ chế an toàn nào trong ws security thì phụ thuộc nhiều vào loại service và những tính năng mà servive này cung cấp, ví dụ như service về giao dịch tài chính ngân hàng phải có cơ chế an toàn hiệu qủa hơn so với service chọn lọc và phân loại tin tức hay service cho biết tỷ giá trao đổi giữa các loại ngoại tệ,…. Bên cạnh đó còn một điểm cần quan tâm đó là sự an toàn không chỉ phụ thuộc vào những giải thuật, những tiêu chuẩn, và những cơ chế mà ws security mang lại, mà nó còn tùy vào thái độ của các công ty có hiểu rõ tầm quan trọng của an toàn thông tin khi triển khai các ứng dụng, giao dịch trên mạng hay không cũng rất cần thiết. Có hai hình thức bảo mật, đó là bảo mật trên kênh truyền và bảo mật ở mức thông điệp.Hiện nay hầu hết các dịch vụ đều kết hợp cả hai hình thức để tối ưu cho việc bảo mật. Bảo mật ở mức kênh truyền : trên kênh truyền phải bảo đảm được thông điệp an toàn và toàn vẹn.Ở mức này thì ta thường dùng kênh truyền an toàn như HTTP + SSL để tạo kết nối an toàn đến client.An toàn ở mức này đòi hỏi cơ sở hạ tầng mạng phải tốt. Bảo mật ở mức thông điệp : để tăng mức độ an toàn cho thông điệp,ta sử dụng thêm WS security cung cấp mức an toàn cho thông điệp.Các dữ liệu được mã hóa và được sử dụng chữ kí số để tránh bị đánh cắp thông tin.Cả hai bên server và client đều sử dụng các key để có thể chứng thực lẫn nhau

3. Bảo mật trong Web Service.

- **Web Service Security là một chuẩn an toàn cho SOAP và cả những phần mở rộng của SOAP, nó được dùng khi muốn xây dựng những web service toàn vẹn và tin cậy. Web Service Securty đảm bảo cho tính an tovcậy của thông điệp**

- **Chứng thực trong một ứng dụng**
  - Phía máy khách: Máy khách sẽ cung cấp một dấu hiệu an toàn trong tập tin mô tả cũng như phải chỉ rõ một Callback handler để lấy tài khoản và mật khẩu trong thông điệp SOAP và gửi tới máy chủ.
  - Phía máy chủ: Máy chủ chỉ rõ một Callback handler để đọc dấu hiệu an toàn trong SOAP máy kháchvà xác nhận nó.

Ví dụ bảo mật trong Web Service.

- **Thiết lập quyền truy cập và sử dụng Web Service**
- **Web Service cũng như Web Page, một số cho phép truy cập tự do, một số yêu cầu phải có quyền truy cập. Việc thiết lập quyền truy cập một Web Service được thực hiện như đối với Web Page, và thường được cấu hình trong web.config. Đoạn mã sau thiết lập quyền truy cập Administrators cho tất cả các trang trong thư mục admin:**

```
<location path="Admin">
<system.web>
        <authorization>
                <allow roles=”Administrators”/>
                <deny users=”*”/>
        </authorization>
</system.web>
</location>
```

4. Thực hiện bảo mật trong Web Service.

- Các bước tạo sự an toàn thông tin trong một ứng dụng

Phía máy khách:

- Chỉ rõ những thành phần của thông điệp mà phải có chữ ký hay một dấu hiệu chứng thục nào đó (nằm ở phần thân thông điệp)thông điệp

- Chỉ rõ một khóa trên hệ thống tập tin mà sẽ ký lên thông điệp. Chỉ những máy khách đã được cấp quyền mới có quyền sở hữu khóa này.

- Chỉ rõ những giải thuật sẽ được sử dụng bởi khóa để ký lên thông điệp.

Phía máy chủ:

- Chỉ rõ những thành phần của thông điệp cần được ký. Nếu thông diệp đến không có một chữ ký hợp lệ, thì yêu cầu sẽ thất bại.

- Chỉ rõ giải thuật mà khóa sử dụng để bảo đảm tính toàn vẹn của thông điệp gửi đến .Thông điệp phản hồi phải được ký và cung cấp thông tin chữ ký khi phản hồi.

Ví dụ thực hiện bảo mật trong Web Service.

- Đăng nhập

- Để có thể sử dụng các phương thức có thiết lập quyền thực hiện, cần phải đăng nhập với quyền tương ứng. Phương thức đăng nhập có thể được định nghĩa như sau:

```
[WebMethod] public bool LogIn(string userName, string password) {
        if (System.Web.Security.Membership.ValidateUser(userName, password)) {
                FormsAuthentication.SetAuthCookie(userName, true); return true; }
        else { return false; } }
```

- Việc kiểm tra tính hợp lệ của tài khoản được thực hiện bằng phương thức ValidateUser của Membership và đặt trạng thái đăng nhập bằng phương thức SetAuthCookie của FormsAuthentication. Để trạng thái đăng nhập được lưu lại phục vụ cho việc thực hiện các phương thức khác, phía client phải bật Cookie.

Ví dụ thực hiện bảo mật trong Web Service.

- Đăng xuất

- Phương thức đăng xuất được định nghĩa như sau:

```
[WebMethod] [PrincipalPermission(SecurityAction.Demand, Authenticated = true)]
public void LogOut() {
        if (Context.User.Identity.IsAuthenticated) FormsAuthentication.SignOut();
}
```

- Sau khi đăng xuất, trạng thái đăng nhập bị xoá và để thực hiện được các phương thức có thiết lập quyền thì phải đăng nhập trở lại.

- Để xác định trạng thái đăng nhập hiện thời, hoặc xem tài khoản đăng nhập hiện thời có nằm trong nhóm nào đó hay không, có thể định nghĩa các phương thức sau:

```
[WebMethod] public bool IsAuthenticated() {
        return Context.User.Identity.IsAuthenticated;
}
[WebMethod] public bool IsInRole(string role) {
        return System.Web.Security.Roles.IsUserInRole(role);
}
```

Tài liệu tham khảo

[1]. Bob Breedlove, *Web Programming Unleashed*. Sams Publishing.
[2]. Michael Girdley, Kathryn A. Jones, *Web Programming with Java*. Sams Publishing
[3]. Al Williams*, Active Server Pages*, Coriolis Group
[4]. *Beginning Active Server Pages*, Wrox Press
[5]. Các địa chỉ website :
www.oracle.com/technetwork/java/javase/downloads/
www.netbeans.org/downloads/
www.visualstudio.com/downloads,
www.microsoft.com/en-us/download/,www.truli.vn
[6] http://www.glib.hcmus.edu.vn (Thư viện Đại học Khoa Học Tự Nhiên TP. HCM)