# CHƯƠNG 1: WEB SERVICE

**1.1 Giới thiệu về Service**

## 1.1.1 Khái niệm

Theo IBM: “*Service is a repeatable task within a business process*”[5]. Theo đó, Service là một ứng dụng với người dùng, một thao tác được thực hiện một hoặc nhiều lần trong một tiến trình và được thực hiện bởi một hay nhiều người.

Service là một hệ thống có khả năng nhận một hay nhiều yêu cầu xử lý và sau đó đáp ứng lại bằng cách trả về một hay nhiều kết quả. Quá trình nhận yêu cầu và trả kết quả về được thực hiện thông qua các giao diện đã được định nghĩa trước đó. Thông thường việc giao tiếp này được thực hiện trên các giao diện đã được chuẩn hóa và sử dụng rộng rãi.

Một hệ thống được thiết kế theo kiểu hướng Service là một hệ thống trong đó các chức năng của hệ thống được xây dựng dựa trên các service có độ kết dính thấp. Các service trong hệ thống giao tiếp với nhau thông qua việc gởi nhận các thông điệp.

## 1.1.2 Các đặc điểm chính của Service

Mỗi service được xây dựng dựa trên các giao diện chuẩn hóa đã được sử dụng rộng rãi. Chi tiết hiện thực của mỗi service sẽ không được thể hiện ra bên ngoài. Mỗi service chỉ công bố một số các giao diện của nó cho user có thể dùng để gởi các yêu cầu và nhận kết quả trả về.

Mỗi Service có tính độc lập cao, có thể được xây dựng và đưa vào sử dụng mà không phụ thuộc vào các service khác.

Trao đổi dữ liệu: các Service không truyền các *class* và *type*.Thay vào đó, các *class* và *type* sẽ được đặc tả hình thức.

**1.2. Tổng quan về Web Service**

## 1.2.1 Khái niệm Web Service

Web Service là một giao diện truy cập mạng đến các ứng dụng chức năng, được xây dựng từ việc sử dụng các công nghệ chuẩn Internet[5].

Thuật ngữ Web Service diễn tả một cách thức tích hợp các ứng dụng trên nền website lại với nhau bằng cách sử dụng các công nghệ XML, SOAP, WSDL,UDDI trên nền tảng các giao thức Internet với mục tiêu tích hợp ứng dụng và truyền thông điệp. XML được sử dụng để đánh dấu dữ liệu, SOAP được dùng để truyền dữ liệu, WSDL được sử dụng để mô tả các dịch vụ có sẵn và UDDI được sử dụng để liệt kê những dịch vụ nào hiện tại đang có sẵn để có thể sử dụng. Web Service cho phép các tổ chức có thể trao đổi dữ liệu với nhau mà không cần phải có kiến thức hiểu biết về hệ thống thông tin đứng sau Firewall kia.

Không giống như mô hình khách/chủ truyền thống,Web Service không cung cấp cho người dùng một giao diện đồ hoạ nào, Web Service đơn thuần chỉ là việc chia sẻ các dữ liệu logic và xử lý các dữ liệu đó thông qua một giao diện chương trình ứng dụng được cài đặt xuyên suốt trên mạng máy tính.

Web Service cho phép các ứng dụng khác nhau từ các nguồn khác nhau có thể giao tiếp với các ứng dụng khác mà không đòi hỏi nhiều thời gian lập trình, do tất cả các quá trình giao tiếp đều tuân theo định dạng XML, cho nên Web Service không bị phụ thuộc vào bất kì hệ điều hành hay ngôn ngữ lập trình nào.

Web Service cung cấp tính trừu tượng cho các giao diện chuẩn, cho nên sẽ không nảy sinh ra bất kì vấn đề gì trong quá trình tương tác. Web Service cho phép giao tiếp giữa các nền tảng khác nhau có thể hoạt động cùng nhau theo nguyên tắc tạo ra một nền tảng trung gian có liên quan.

=> Tóm lại: Web Service là:

* + - * Làm việc xuyên qua tường lửa và proxy
      * Sẵn sàng đối với các nền tảng máy trạm khác nhau
      * Một dịch vụ phần mềm được trình bày trên web thông qua giao thức SOAP, được mô tả bằng một tệp WSDL và được đăng ký trên UDDI.

## 1.2.2 Đặc điểm Web Service

Cho phép khách/chủ tương tác với nhau cả trong môi trường khác nhau.

XML và HTTP là nền tảng kỹ thuật chính. Phần lớn kỹ thuật của Web Service được xây dựng là những dự án nguồn mở cho nên độc lập và vận hành được với nhau

Web Service rất linh động: với UDDI và WSDL thì việc mô tả và phát triển Web Service có thể tự động hóa.

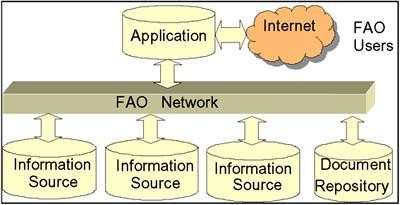
Web Service bao gồm nhiều mô đun và có thể công bố trên mạng Internet.

Web Service có thể chia sẻ và gọi thực hiện qua mạng và có độ an toàn riêng tư.

**1.3. Một số mô hình áp dụng Web Service**

## 1.3.1 Sử dụng để tổng hợp dữ liệu tại FAO

Chức năng quan trọng nhất của FAO: thu thập,phân tích,đánh giá các thông tin hỗ trợ các chính phủ chống đói nghèo và đạt được an ninh lương thực. Trong WAICENT, một lượng đồ sộ các dữ liệu được trình bày ở các định dạng hoàn toàn khác nhau, trên nhiều ngôn ngữ,không có các tiêu chuẩn cho việc trình bày. Điều quan trọng là chia sẻ dữ liệu giữa các hệ thống nhanh chóng và dễ dàng,các hệ thống đang tồn tại cần phải “nói chuyện” được với nhau. Bên trong tổ chức sử dụng hai công nghệ khác nhau (ASP và Java JSP/servlet) và không có tiêu chuẩn nào để quản lý các phương án ngôn ngữ văn bản hoặc kiến trúc dữ liệu[1].



*Hình 3.1: Mô hình kết nối CSDL của FAO*

Đòi hỏi công nghệ cần hỗ trợ tính tương hợp các nguồn dữ liệu sẵn có và quản lý các phương án đa ngôn ngữ mà không phải thay đổi kiến trúc của cơ sở dữ liệu. Để khắc phục các vấn đề này,FAO đưa ra một tiếp cận dựa trên các công nghệ Web Service, XML. Mục tiêu chính: Tạo ra môi trường nơi mà các hệ thống thông tin mới dựa trên web có thể được phát triển nhanh chóng và dễ dàng,sử dụng bất cứ nền tảng công nghệ nào. Triển khai các máy tạo và phát triển báo cáo động của một kho văn bản XML để quản lý siêu dữ liệu và các phương án về ngôn ngữ theo một cách chung.

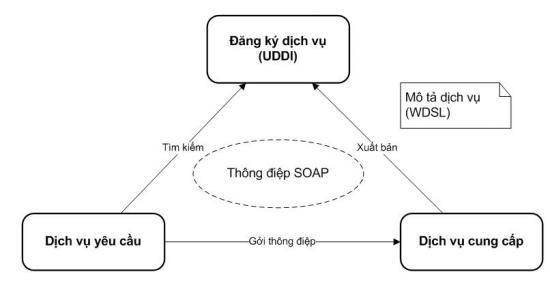
## 

## 1.3.2 Sử dụng Web Service trong công nghệ di động

Hãng T-Mobile đặt niềm tin vào Web Service như một nền tảng cho việc quảng bá dữ liệu tới các khách hàng và nhân viên sử dụng di động. Hãng có phần mềm trung gian với khoảng 50-60 Web Services được tích hợp bên trong cho phép tích hợp các dịch vụ của T-Mobile với nhau như nhận dạng,cá nhân hóa và thanh toán hóa đơn,với các dịch vụ quảng bá nội dung thông tin cho máy di động cho người sử dụng được 250 đối tác cung cấp. Không cần phải quan tâm tới việc các hệ thống là Microsoft, Linux…, miễn là đầu ra ở dạng XML và sử dụng SOAP là được[1].

**1.4. Mô hình Web Service, ưu và nhược điểm**

## 1.4.1 Mô hình Web Service



*Hình 3.2: Mô hình Web Service* Nhà cung cấp đăng ký Web Service với UDDI.

Người sử dụng tìm kiếm dịch vụ trên UDDI qua một URL thích hợp. UDDI trả lại một bản mô tả WSDL cho nhà cung cấp.

Người sử dụng triệu gọi dịch vụ bằng một cuộc gọi SOAP tới nhà cung cấp Nhà cung cấp trả lại kết quả của cuộc gọi SOAP cho người sử dụng

## 1.4.2 Ưu điểm

Cho phép chương trình được viết bằng các ngôn ngữ khác nhau trên các nền tảng khác nhau giao tiếp được với nhau dựa trên một nền tảng tiêu chuẩn

Đơn giản (chỉ dùng URL)

Làm việc với các giao thức chuẩn Web như XML, HTTP và TCP/IP.

Sự an toàn của máy chủ cơ sở dữ liệu luôn được bảo mật một cách chắc chắn. Web Service làm giảm giá thành cho việc tích hợp các hệ thống khác nhau.

## 1.4.3 Nhược điểm

Phụ thuộc vào tốc độ đường truyền Internet.

Web Service thiếu cơ chế khôi phục đủ tin cậy để đảm bảo giao dịch được khôi phục lại trạng thái ban đầu trong trường hợp xảy ra sự cố

Số lượng các ứng dụng cộng tác cùng hoạt động sẽ ảnh hưởng tới hiệu suất tối ưu của Web Service.

Tải trọng: ứng dụng Web Service là các ứng dụng sử dụng rất nhiều thông điệp. Khả năng bùng nổ số lượng giao dịch trao đổi sẽ làm hệ thống máy chủ ứng dụng và kiến trúc hạ tầng hệ thống thông tin của doanh nghiệp trở nên ngưng trệ.

Vì Web Service đòi hỏi kết nối thông qua khá nhiều máy chủ trung gian cho nên băng thông/tốc độ của hạ tầng mạng và các yêu tố liên quan tới hệ thống rõ ràng có vai trò quan trọng góp phần cải thiện hiệu năng của toàn bộ các ứng dụng WS

**1.5 Các thành phần chính của Web Service**

|  |
| --- |
| Sổ đăng ký  UDDI |
| Mô tả dịch vụ |
| WSDL XML |
| Giao thức truyền thông |
| SOAP |
| Giao thức giao vận  HTTP |

*Hình 3.3: Các thành phần chính của Web Service*

XML được sử dụng để định dạng dữ liệu, SOAP được sử dụng trao đổi dữ liệu, WSDL được sử dụng để mô tả dịch vụ hiện có và UDDI được sử dụng để liệt kê các Web Service hiện có.

## 1.5.1 Giao thức giao vận HTTP

### 1.5.1.1 Giao thức HTTP

Tầng giao vận liên quan tới cơ chế sử dụng để chuyển yêu cầu dịch vụ và thông tin phản hồi từ phía nhà cung cấp dịch vụ tới người sử dụng dịch vụ. Có rất nhiều tiêu chuẩn sử dụng xung quanh WS, nhưng phổ biến nhất vẫn là giao thức HTTP.

Giao thức HTTP thường được sử dụng đối với yêu cầu dịch vụ và đáp ứng.

### 1.5.1.2 Ưu điểm

HTTP là nền tảng hạ tầng phổ biến và sẵn sàng nhất.

Giao thức HTTP hoàn toàn mở và khai triển trên rất nhiều loại hệ thống

Hầu hết mọi tổ chức đều chấp nhận cho phép trao đổi thông tin dựa trên giao thức HTTP vượt qua tường lửa bảo vệ.

### 1.5.1.3 Nhược điểm

HTTP là một giao thức đơn giản và không có tính trạng thái, không được thiết kế đặc biệt cho mục đích vận chuyển dữ liệu của các ứng dụng.

Giao thức không hỗ trợ lưu trữ trạng thái

Không phải là một giao thức đáng tin cậy phù hợp với nhu cầu truyền dữ liệu.

## 1.5.2 Giao thức truyền thông SOAP

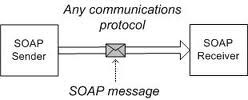
### 1.5.2.1 Khái niệm

SOAP là gì:

* + - * + SOAP là giao thức truyền thông giữa các ứng dụng.
        + SOAP được thiết kế để liên lạc qua Internet và làm việc qua tường lửa.
        + SOAP độc lập nền tảng, độc lập ngôn ngữ.
        + SOAP dựa trên XML, đơn giản và dễ mở rộng. SOAP có đặc trưng:
        + SOAP được thiết kế đơn giản và dễ mở rộng
        + Tất cả các message SOAP đều được mã hóa sử dụng XML.
        + SOAP sử dụng giao thức truyền dữ liệu riêng.
        + SOAP không bị ràng buộc bởi ngôn ngữ lập trình hoặc công nghệ nào
        + SOAP không quan tâm đến công nghệ gì được sử dụng để thực hiện miễn là người dùng sử dụng các message theo định dạng XML.

=> Tóm lại: SOAP là giao thức mà định nghĩa cái cách để chuyển một XML message từ A đến B dựa trên giao thức chuẩn web HTTP (hoạt động trên cổng

1. qua giao thức Internet TCP/IP.



*Hình 3.4: Thông điệp SOAP*

Tại sao phải có SOAP:

* + Phát triển các ứng dụng cho phép các chuơng trình trao đổi qua Internet.
  + Các ứng dụng liên lạc với nhau bằng cách sử dụng các cuộc gọi thủ tục ở xa giữa các đối tuợng như DCOM,CORBA
  + SOAP cung cấp cách để liên lạc giữa các ứng dụng chạy trên các hệ điều hành khác nhau,với các công nghệ khác nhau và ngôn ngữ khác nhau.

### 1.5.2.2 Định dạng thông điệp

Một thông điệp SOAP là một văn bản XML được mô tả bởi một thành phần Envelop,chứa một thành phần Body bắt buộc và một thành phần Header không bắt buộc. Thành phần Body có thể chứa một số Body Entries. Thành phần không bắt buộc Fault chỉ có trong thông điệp khi có báo cáo về một quá trình xử lý ngoại lệ.

Phần tử Body mô tả về phuơng thức duới dạng XML và chỉ chứa các tham số hay các truờng duới dạng các thẻ.

Với Document người phát triển phải xử lý gần như là toàn bộ, họ phải đưa ra một loạt các tham số dưới dạng các thẻ XML

### 

### 1.5.2.3 Mã hóa thông điệp

Dữ liệu được mã hoá và gói vào trong phần tử Body của một thông điệp và được gửi đến Host. Host giải mã dữ liệu được định dạng XML về dạng đối tuợng ban đầu.

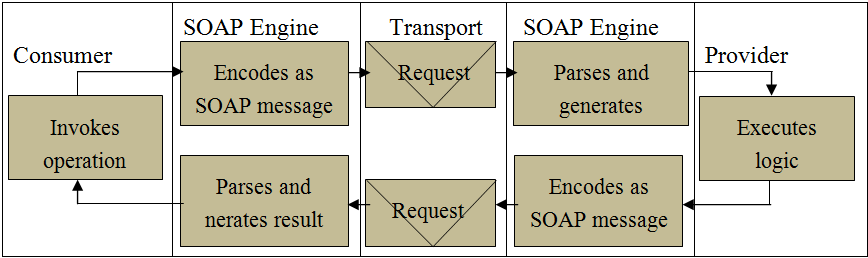
SOAP Remote Procedure Call (RPC encoding): Là kiểu mã hóa đơn giản nhất cho người phát triển. Bạn gọi tới một đối tượng từ xa, kèm theo là các tham số cần thiết. Các tham số được chuyển lần lượt dưới dạng XML và truyền đến đích sử dụng giao thức giao vận như HTTP hay SMTP. Sau khi nhận được, dữ liệu được chuyển trở lại thành dạng đối tượng và kết quả được trả về cho phương thức gọi. SOAP RPC xử lý tất cả công việc mã hóa và giải mã,thậm chí đối với các kiểu dữ liệu phức tạp.

SOAP Remote Procedure Call Literal encoding (SOAP RPC-literal): Sử dụng một dạng thức mã hóa do người sử dụng chỉ định để mã và giải mã dữ liệu dạng XML.

SOAP document-style encoding: Toàn bộ XML được gửi đến máy chủ và người lập trình xác định giao thức giao vận,phân tích dữ liệu dạng XML ở thông điệp yêu cầu và đáp ứng để tìm dữ liệu cần thiết.

### 1.5.2.4 Quá trình xử lý thông điệp

Một thông điệp SOAP giúp cho khách hàng và nơi cung cấp Web Service hoàn thành những tác vụ mà không lo lắng đến sự phức tạp của việc xử lý thông điệp SOAP.

Một processor của khách hàng chuyển các lời yêu cầu phương thức vào trong một thông điệp SOAP.Thông điệp này được truyền qua tầng giao vận (HTTP và SMTP) tới processor của nơi cung cấp,tại đây thông điệp sẽ được phân tích thành lời yêu cầu phương thức.Sau đó nơi cung cấp sẽ thực hiện những bước logic cần thiết và trả lại kết quả cho processor của nó,processor này sẽ phân tích thông tin trong thông điệp hồi đáp.Thông điệp này được truyền qua tầng giao vận tới khách hàng yêu cầu. Processor của nó phân tích thông điệp hồi đáp thành kết quả dưới dạng một đối tượng.

*Hình 3.5: Quá trình xử lý thông điệp SOA*

## 1.5.3 Ngôn ngữ đánh dấu,mở rộng XML

### 1.5.3.1 Khái niệm XML

XML là nền tảng của Web Service và được dùng để trao đổi dữ liệu. XML là một chuẩn nổi tiếng cho việc tổ chức,lưu trữ và trao đổi dữ liệu.

XML được hỗ trợ bởi hầu hết các ngôn ngữ lập trình hiện đại (DotNet, Java…)

XML được sử dụng rộng rãi trong việc trao đổi dữ liệu trên môi trường Internet. XML dùng các thẻ để tổ chức và lưu trữ dữ liệu

### 

### 1.5.3.2 Đặc điểm của XML.

XML là tự do và mở rộng được. Trong XML các thẻ không được định nghĩa trước mà do người dùng tự phát minh ra thẻ.

XML rất quan trọng đối với sự phát triển của web trong tương lại. XML sẽ là công cụ xử lý và truyền dữ liệu phổ biến nhất.

XML là công cụ dùng được trên mọi nền phần cứng, độc lập với phần cứng và phần mềm để truyền (trao đổi,chia sẻ) thông tin.

### 1.5.3.3 XML được sử dụng như thế nào

XML được thiết kế để lưu trữ và trao đổi dữ liệu nhưng không hiển thị dữ liệu.

XML có thể trao đổi dữ liệu giữa các hệ thống không tương thích.

### 1.5.3.4 Cấu trúc tài liệu XML

XML hợp khuôn dạng: khai báo XML và dữ liệu XML.

XML hợp lệ: Là tài liệu được kết hợp với định nghĩa kiểu tư liệu (Document Type Definition) và tuân theo tiêu chuẩn đó.

### 1.5.3.4 Quy tắc cú pháp ngôn ngữ XML

Các khai báo XML cần được đặt ở dòng đầu tiên của tài liệu.

Mọi phần tử XML đều phải có thẻ đóng: />

Tất cả các tài liệu XML phải có thẻ gốc trong đó thẻ đầu tiên là thẻ gốc.

Các thẻ XML phân biệt hoa\_thường và khoảng trắng được giữ lại.

Các giá trị thuộc tính phải luôn đặt trong ngoặc kép.

### 1.5.3.5 Ưu điểm của XML

Đơn giản, ổn định, linh hoạt và có tính mở rộng cao.

XML được chấp nhận rộng rãi. Rất nhiều công cụ và tiện ích sẵn có đáp ứng nhu cầu phân tích và chuyển đổi dữ liệu XML hoặc hiển thị chúng.

### 1.5.3.6 Nhược điểm của XML

Sự phức tạp

Việc chuẩn hóa Dung lượng lớn

## 1.5.4 Ngôn ngữ mô tả dịch vụ WSDL

### 1.5.4.1 Khái niệm

WSDL là ngôn ngữ dựa trên XML và mô tả cách thức truy cập Web Service

WSDL thường được sử dụng với SOAP và cấu trúc XML để cung cấp Web Service qua Internet. Một máy khách kết nối tới Web Service có thể đọc WSDL để xác định hàm nào hiện đang có trên máy chủ. Khách có thể sử dụng SOAP để gọi một trong nhiều hàm được liệt kê trong WSDL.

=> Tóm lại:

WSDL mô tả Web Service theo cú pháp tổng quát XML, bao gồm các thông tin: tên service, giao thức và kiểu mã hoá, tham số, kiểu dữ liệu …

WSDL chỉ định các đặc tính vận hành của Web Service. Ngôn ngữ mô tả những khái niệm trả lời cho các câu hỏi sau:

* + - * + Cái gì (Web Service làm gì) ?
        + Ở đâu (nơi chứa Web Service) ?
        + Như thế nào (Web Service có thể kích hoạt bằng cách nào) ?

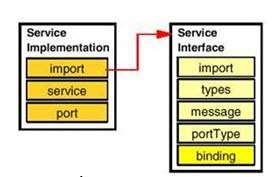
### 1.5.4.2 Cấu trúc WSDL

Một WSDL hợp lệ gồm có hai phần:

* + - * + Phần giao diện mô tả giao diện và giao thức kết nối.
        + Phần thi hành mô tả thông tin để truy xuất service.

Cả 2 thành phần này sẽ được lưu trong hai tập tin XML, bao gồm:

* + - * + Tập tin giao diện service (cho phần 1)
        + Tập tin thi hành service (cho phần 2)



*Hình 3.6: Service Interface và Service Implementation*

### 1.5.4.3 Tập tin giao diện – Service Interface

*<PortType>* : các phép toán được thực hiện bới Web Service.

*<Message>* : mô tả thông điệp được gửi giữa Máy khách và Máy chủ.

*<Types>* : kiểu dữ liệu được sử dụng bởi Web Service.

*<Binding>* : các giao thức truyền thông được sử dụng bởi Web Service.

* *Types:* Định nghĩa loại dữ liệu được sử dụng bởi các WebService.

**<wsdl:types>**

**<xsd:schema .../">**

**</wsdl:types>**

* *Message*

Mỗi message được thiết kế phục vụ cho việc truyền hoặc nhận thông tin. Mỗi message có thể bao gồm một hoặc nhiều phần. Mỗi phần có thể được so sánh với thông số của một chức năng gọi trong ngôn ngữ lập trình truyền thống

**<message name = “runtoken”> \***

**<part name = “nmtoken” element=”qname”? type=”qname”?/> \***

**</message>**

* *PortType*

Các phần tử *<PortType>* là thành phần quan trọng nhất của WSDL.

Nó định nghĩa một Web Service, các tác vụ mà service cung cấp và định dạng các thông điệp được sử dụng để khởi động các tác vụ này

**<wsdl: portType name=”nmtoken”> \***

**<wsdl: operation name=”nmtoken” .../> \***

**</wsdl: portType>**

* *Binding*

*<binding>* dùng để định dạng thông điệp và chi tiết giao thức của mỗi *portType*

và được gán 1 giao thức truyền tin để có thể truy xuất và tương tác với WSDL. Binding chứa 1 hoặc nhiều hoạt động (operation)

Một binding gồm 2 thuộc tính: tên(name) và loại(type).

* Name: định nghĩa tên của binding.
* Type: chỉ ra cổng cho binding.

### 1.5.4.4 Tập tin thi hành – Service Implementation

* *Port*

Một *<port>* định nghĩa một thiết bị đầu cuối bằng cách chỉ định một địa chỉ duy nhất cho binding. Mỗi port có 2 thuộc tính:name và binding.

*Name*: cung cấp một cái tên duy nhất trong tất cả các port

*Binding*: chỉ các *binding* đang được sử dụng luật liên kết, xác định bởi WSDL

* *Service*

Các phần tử *<service>* định nghĩa các cổng hỗ trợ bởi Web Service.

### 1.5.4.5 Ưu điểm của WSDL

Như một yêu cầu cơ bản đối với ứng dụng của bất cứ Web Service,WSDL là yêu cầu bắt buộc đáp ứng nhu cầu công bố giao tiếp và thỏa thuận cho các dịch vụ khác kích hoạt.

### 1.5.4.6 Nhược điểm của WSDL

Tài liệu không cung cấp một số thông tin người sử dụng có nhu cầu như :

* + - * + Ai cung cấp dịch vụ ?
        + Loại hình kinh doanh cung cấp dịch vụ ?
        + Các dịch vụ khác cùng do nhà cung cấp dịch vụ này cung cấp ?
        + Dịch vụ này sẽ cung cấp với chất lượng dịch vụ như thế nào ?
        + Đây là dịch vụ miễn phí hay có thu phí ?

## 1.5.5 Tích hợp mô tả trình bày tổng hợp UDDI

### 1.5.5.1 Khái niệm

UDDI là một chuẩn công nghiệp cho việc công bố và tìm kiếm thông tin về Web Service. Nó định nghĩa một khung thông tin cho phép bạn mô tả và phân loại tổ chức của bạn, dịch vụ của nó và những chi tiết kỹ thuật về giao diện của Web Service mà bạn trình bày.

UDDI chỉ định cách thức lưu trữ và nhận thông tin về các dịch vụ và đặt biệt là nhà cung cấp dịch vụ cùng với các giao tiếp kỹ thuật.

UDDI dựa vào những chuẩn đã có như là ngôn ngữ đánh dấu mở rộng (XML) và giao thức truy cập đối tượng đơn giản (SOAP). Tất cả các cài đặt của UDDI đều hỗ trợ các đặc tả UDDI.

=> Tóm lại: Để có thể sử dụng các dịch vụ, trước tiên khách phải tìm dịch vụ, ghi nhận thông tin về cách sử dụng dịch vụ và biết được đối tượng cung cấp dịch vụ. UDDI định nghĩa thành phần cho biết trước các thông tin này để cho phép máy khác truy tìm và nhận lại những thông tin yêu cầu sử dụng Web Service.

### 1.5.5.2 Đặc điểm của UDDI

UDDI là phần chứa các thông tin của web service, xây dựng trên nền tảng .NET UDDI được miêu tả bởi ngôn ngữ WSDL và giao tiếp thông qua SOAP

Nhiệm vụ UDDI: tìm đúng dịch vụ và định nghĩa cách kíck hoạt dịch vụ

### 1.5.5.3 Nội dung của thƣ mục UDDI

Một nội dung thư mục UDDI là một tệp XML mô tả một nghiệp vụ và các dịch vụ nó chào. Nội dung trong UDDI có ba phần:

* + - * + White pages: chứa thông tin liên hệ và các định dạng của Web Service. Những thông tin này cho phép đối tượng khác xác định được dịch vụ.
        + Yellow pages: chứa thông tin mô tả Web Service. Những thông tin này cho phép các đối tượng thấy được từng loại của Web Service.
        + Green pages: chứa thông tin kỹ thuật mô tả các hành vi và các chức năng
        + Loại dịch vụ – tModel: chứa các thông tin về loại dịch vụ được sử dụng.

### 1.5.5.4 Cấu trúc sổ đăng ký UDDI

UDDI cung cấp 4 cấu trúc dữ liệu mô tả dịch vụ mà nó đưa ra: BusinessEntity, BusinessService, BusinessTemplate và tModels.

* + - * + BusinessEntity: mô tả nhà cung cấp dịch vụ
        + BusinessService: chứa các thông tin chung về dịch vụ
        + BindingTemplate: chứa thông tin kỹ thuật cách thức truy cập vào dịch vụ
        + tModels (Technical Model- mô hình kỹ thuật): chứa các thông tin về loại Web Service sử dụng. Được sử dụng để lấy thông tin chi tiết về giao diện của Web Service và làm cho chúng có thể sử dụng lại giữa các dịch vụ tương thích.

### 1.5.5.5 Các kiểu sổ đăng ký UDDI

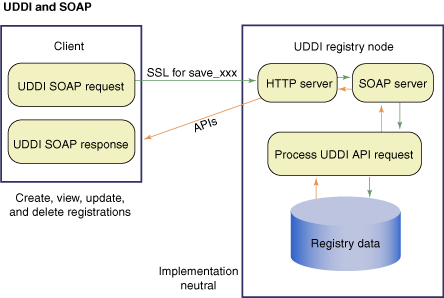
Mô hình đám mây các nút UDDI: một cơ chế khai triển công khai của tiêu chuẩn UDDI đó là Sổ đăng ký kinh doanh UDDI hoặc UBR. UBR bao gồm vài nút UDDI. Những nút này do các công ty như IBM, Microsoft hay SAP và NTT quản lý. Khi một nhà cung cấp dịch vụ muốn công bố dịch vụ của họ, họ sẽ tới một trong các địa chỉ UBR như: [http://uddi.ibm.com](http://uddi.ibm.com/). Và sau đó đăng ký rồi công bố dịch vụ của họ. Dữ liệu tiếp tục nhân bản tới các nút khác trong cùng hệ thống UBR.

Nhóm hoặc các sổ đăng ký cộng tác: những triền khai này tập trung vào một số lượng cụ thể các đối tác đã từng biết

Sổ đăng ký riêng tư: hầu hết các công ty đều hướng tới việc bắt đầu các dự án Web Service thông qua một sổ đăng ký UDDI riêng biệt

### 1.5.5.6 UDDI làm việc như thế nào

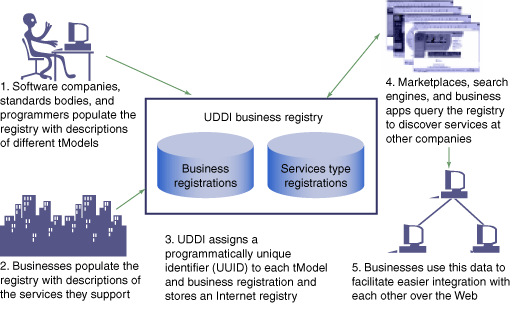
Bản ghi của UDDI chứa mô tả của doanh nghiệp,có thể truy cập bằng máy tính và các dịch vụ mà chúng hỗ trợ. UDDI cung cấp một lược đồ và mô hình lập trình với định nghĩa luật giao tiếp với bản ghi. Tất cả các hàm API trong đặc tả UDDI được định nghĩa trong XML, gói gọn trong một phong bì SOAP, và gửi qua HTTP.



*Hình 3.7: Luồng thông báo UDDI giữa Máy khách và Registry*

Hình vẽ miêu tả sự truyền tải thông báo UDDI, từ yêu cầu SOAP của máy khách thông qua giao thức HTTP đến một nút bản ghi đăng ký và quay lại. Máy chủ SOAP của hệ thống đăng ký tiếp nhận các thông điệp, xử lý nó, và trả lại một kết quả SOAP đến máy khách. Theo chính sách của bản ghi, các yêu cầu từ máy khách mà bắt buộc phải chỉnh sửa dữ liệu phải là các giao dịch đảm bảo an ninh và được xác thực.

Vậy UDDI làm việc như thế nào?



*Hình 3.8: Cách thức làm việc của UDDI*

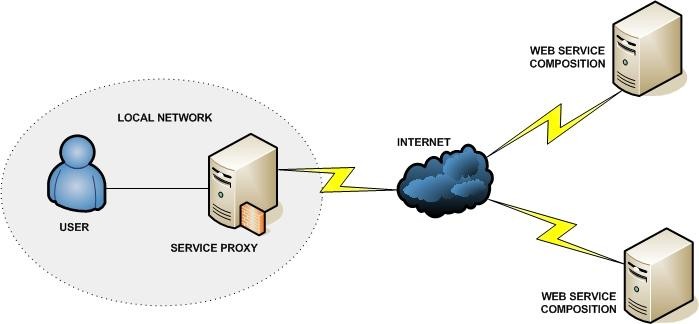
Một bản ghi UDDI được xây dựng trên dữ liệu cung cấp bởi khách hàng của nó. Có vài bước để tạo ra dữ liệu hữu dụng trong UDDI. Như trong bước 1, công bố thông tin hữu ích đến bản ghi bắt đầu khi các công ty phần mềm và các cá nhân định nghĩa các đặc tả liên quan đến công nghiệp hay kinh doanh, mà họ đăng ký với UDDI. Những thứ này được biết như là các mô hình kỹ thuật, hoặc thông dụng hơn là tModels.

Trong bước 2, các công ty cũng đăng ký bản mô tả kinh doanh các dịch vụ của họ. Một bản ghi UDDI sẽ theo dõi tất cả các điểm này bằng cách gán cho mỗi điểm một định danh duy nhất, được biết đến như là một khóa định danh phổ biến duy nhất (Unique Universal Identifier - UUID) như trong bước 3. Một khóa UUID được đảm bảo là duy nhất và không bao giờ thay đổi trong một bản ghi UDDI. Những khóa này trông giống như một chuỗi số thập lục phân ngẫu nhiên có định dạng. Chúng có thể được sử dụng để tham chiếu đến một điểm mà chúng được gán vào. Các khóa UUID tạo trong một bản ghi chỉ có nghĩa nội trong bản ghi đó.

Các khách hàng khác, như là e-Marketplaces, máy tìm kiếm, và ứng dụng thương mại trong bước 4, sử dụng một bản ghi UDDI để khám phá các dịch vụ quan tâm. Và ngược lại, các doanh nghiệp khác có thể yêu cầu các dịch vụ này, cho phép sự tích hợp đơn giản và thay đổi theo thời gian như minh họa trong bước 5.

**1.6 Tìm hiểu về Service Proxy**

Service Proxy về bản chất cũng là một Web Service được triển khai ở phía Máy khách. Service Proxy chứa các đoạn mã để chỉ rõ sự kết hợp giữa các giao diện Web Service, Service Proxy thường nằm phía bên trong một hệ thống mạng máy tính phức tạp. Mô hình tổng quan của một hệ thống với Service Proxy được thể hiện thông qua hình dưới đây[5]



*Hình 3.9: Minh hoạ mô hình Web Service với Service Prox*

Service Proxy sẽ thực thi phương thức giống như phương thức được triển khai trên các remote Web Service, tuy nhiên Service Proxy không thực hiện bất kì một thao tác tính toán nào cả, nó chỉ có nhiệm vụ nhận các yêu cầu từ phía khách rồi chuyển tiếp các thông điệp yêu cầu đến các remote Web Service, tại remote Web Service sẽ thực thi các thao tác tính toán trên các dữ liệu được chuyển đến đó và trả lại kết quả cho Service Proxy. Service Proxy nhận kết quả trả về và chuyển tiếp cho máy khách.

Một Service Proxy sẽ thực thi lần lượt ba thao tác yêu cầu dưới đây để thực hiện một lời gọi phương thức tới một remote Web Service:

* + - Truyền đối số
    - Xây dựng lời gọi Web Service
    - Đọc kết quả trả về từ Remote Web Service

Chúng ta thường sử dụng Service Proxy trong trường hợp số lượng mã tích hợp Web Service thường lớn, và tồn tại việc trùng lặp các lời gọi tới cùng một dịch vụ trong các vị trị khác nhau của chương trình.

Và khi sử dụng Service Proxy chúng ta hoàn toàn có thể:

* + - Nhóm dịch vụ bằng kĩ thuật đóng gói, lựa chọn các thứ bậc của dịch vụ.
    - Chia lớp con từ lớp trừu tượng do đó cung cấp thêm các dịch vụ khác.
    - Mỗi một lớp của Service Proxy trình bày Web Service.

Thông thường thì chúng ta không phải tự viết ra Service Proxy. Service Proxy có thể dễ dàng tự được sinh ra từ file WSDL.

**1.7 Bảo mật Web Service:**

## 1.7.1 Tổng quan về an toàn Web Service

Từ những giai đoạn đầu tiên của Internet, các doanh nghiệp luôn đòi hỏi rất khắt khe về vấn đề bảo mật trong thương mại điện tử. Những hạn chế của tường lửa như việc giám sát các gói tin được truyền tải dựa trên giao thức HTTP là chưa có; điều này có thể khiến cho máy chủ có nguy cơ bị những cuộc tấn công không hề biết được biết trước. Đã có rất nhiều các thuật toán đưa ra cơ chế và những chuẩn về bảo mật như sự mã hoá khoá thông tin, chữ ký số …; nhưng hầu hết chỉ tập trung vào việc đưa ra các định dạng bảo về dữ liệu trong quá trình trao đổi, không quan tâm đến việc xác định các nghi thức mà các bên cần thực hiện khi tương tác với nhau.

Ngoài ra, những chuẩn chung về việc chỉ ra nghi thức giao tiếp giữa Web Service là chưa có, đã khiến cho các sản phẩm hỗ trợ bảo mật của Web Service không thể tích hợp với nhau, mặc dù các sản phầm này đều được thiết kế dựa trên chuẩn về bảo mật cho web service.

Một chuẩn an toàn chung cho các hệ thống giao dịch trên mạng thường phải tập trung vào những điều sau[7] :

* + - Identification: định danh được những ai truy cập tài nguyên hệ thống.
    - Authentication: chứng thực truy cập tài nguyên của người muốn sử dụng.
    - Authorization: cho phép giao dịch khi đã xác nhận định danh người truy cập.
    - 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 hợp tính hợp pháp hóa của thông tin đến từ một phía thứ ba ngoài hai phía là người gửi và người nhận.

## 1.7.2 Khái niệm:

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 Security đảm bảo cho tính an toàn, sự toàn vẹn thông điệp và tính tin cậy của thông điệp

## 1.7.3 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

Để cấu hình máy chủ an toàn cần có một dấu hiệu an toàn hợp lệ cũng như phải chỉ rõ một Callback handler để đọc dấu hiệu an toàn trong SOAP máy khách và xác nhận nó.

## 1.7.4 Các bước tạo sự an toàn thông tin trọng một ứng dụng.

## *phía máy khách.*

Chỉ rõ những thành phần của thông điệp mà phải có chữ ký hoặc một dấu hiệu chứng thực nào đó (nằm ở phần thông điệp).

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

### 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 điệp đến không có một chữ ký hợp lệ thì sẽ thất bại.

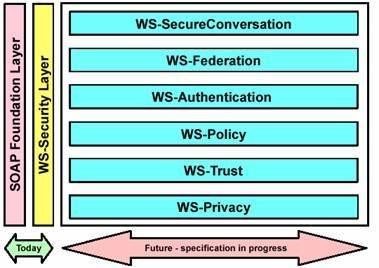
### Chỉ rõ một khóa để duyệt chữ ký của thông điệp đến xem có hợp lệ hay không.

### Chỉ rõ giải thuật mà khóa sử dụng để đảm bảo toàn vẹn của thông điệp gửi đến.

### Thông điệp phản hồi phải được ký và cung cấp thông tin chữ ký khi phản hồi.

## 1.7.5 Những thành phần mở rộng của Web Service Security

Do Web Service Security chỉ là một lớp trong nhiều lớp của giải pháp an toàn đầy đủ, nên cần một mô hình an toàn chung lớn hơn để có thể bao phủ tất cả các khía cạnh an toàn khác nhau như đăng ký (login) và không từ chối (non-repudiation).



*Hình 4.1 Mô hình bảo mật Web Service*

Trong mô hình này các thành phần quan trọng bao gồm:

WS-SecureConversation Describes: Quản lý và xác nhận các thông điệp trao đổi giữa các phần bao gồm sự trao đổi ngữ cảnh, thiết lập, dẫn xuất ra các phiên.

WS-Authentication Describes: Quản lý những dữ liệu, chính sách cần chứng thực.

WS-Policy Describes: Quản lý những ràng buộc của những chính sách an toàn ở các điểm trung gian và đầu cuối.

WS-Trust Describes: Cho phép Web Service an toàn trao đổi, tương tác với nhau.

## 1.7.6 Một số các kỹ thuật Web Service Security.

### eXtensible Access Control Markup Language(XACML).

### Security Assertion Markup Language (SAML).

### XML Key Management Specification (XKMS).

### Web Services Policy Framework (WS-Policy).

### eXentisble Rights Markup Language (XrML).

### Secure Socket Layer (SSL).

# CHƯƠNG 5: TRIỂN KHAI ỨNG DỤNG VÀ ĐÁNH GIÁ KẾT QUẢ

Từ các kiến thức về SOA và Web Service cũng như các kỹ thuật bảo mật Web Service ở các chương trên, chương năm sẽ đề xuất giải pháp để thực hiện bài toán đã đặt ra, triển khai và xây dựng hệ thống.

## Mô tả hệ thống cần xây dựng

Hình 5.1: Hệ thống truyền dữ liệu cần xây dựng

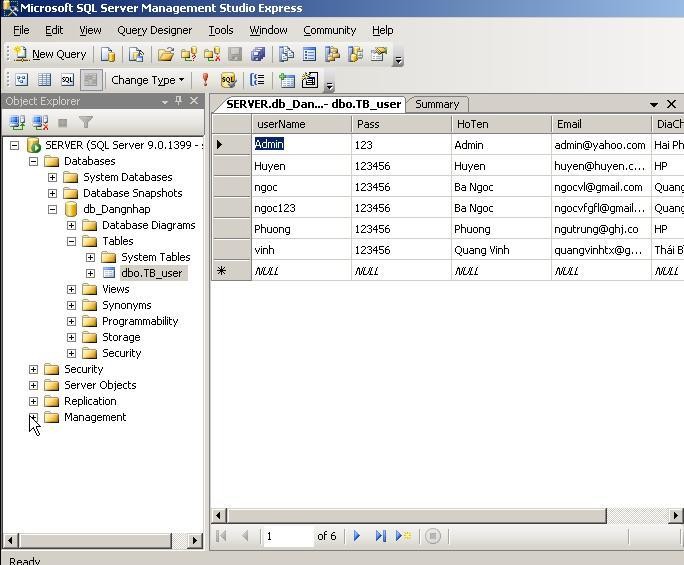
Hệ thống nghiên cứu và xây dựng bao gồm hai máy tính (máy Database Máy chủ và máy Web Máy chủ).

Máy DatabaseMáy chủ là nơi sẽ lưu trữ cơ sở dữ liệu của hệ thống và thực hiện tạo một Web Service có nhiệm vụ hiển thị cơ sở dữ liệu trên trình duyệt. Máy DatabaseMáy chủ được cài đặt hệ điều hành Windows Máy chủ 2003 Enterprise Edition để giúp cho việc thiết lập các cơ chế IIS cũng như FTP dễ dàng hơn.Web Service luôn luôn sẵn sàng nhận lệnh và khi có một lời gọi tới nó Web Service sẽ được thực thi và phục vụ cho lời gọi đó. Nhiệm vụ của Web Service dùng để thực hiện câu lệnh kết nối cơ sở dữ liệu và được public trên hệ thống mạng.

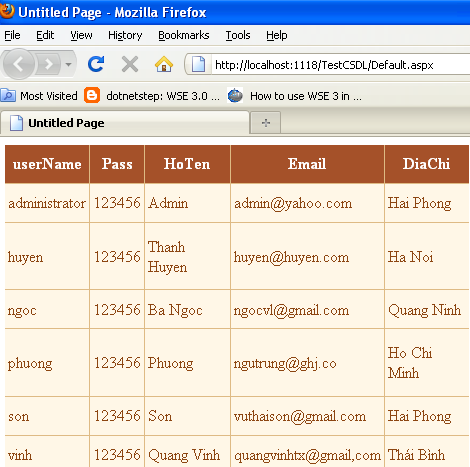
Máy WebMáy chủ là nơi sẽ thực hiện một lời gọi tới Web Service bằng một trình duyệt bất kỳ như FireFox, IE , Opera … Lúc này khi cần hiển thị cơ sở dữ liệu, WebMáy chủ sẽ thực hiện một lời gọi và nhờ có Web Service đã được public trên hệ thống mà các cơ sở dữ liệu sẽ được hiển thị mà không cần phải truy cập trực tiếp vào máy DatabaseMáy chủ.

## Triển khai hệ thống

Để xây dựng một hệ thống hoàn chỉnh thực hiện chức năng hiển thị dự liệu cũng như việc bảo mật đòi hỏi phải có một thời gian dài. Trong thời gian qua tôi đã tiến hành nghiên cứu và sử dụng mô hình tương tác cơ sở dữ liệu giữa hai máy Database và máy Web, bước đầu được những kết quả và xây dựng xong chức năng hiển thị cơ sở dữ liệu mà hệ thống đặt ra và chạy thử.

Ngôn ngữ mà tôi sử dụng để xây dựng hệ thống là ASP.NET của bộ Visual Studio 2008 và Microsoft SQL Máy chủ 2005. Với những tính năng nổi bật của ngôn ngữ này và nhận thấy phù hợp với việc xây dựng và triển khai hệ thống cần xây dựng. Ngoài ra tôi sử dụng các Application có sẵn trong hệ điều hành Windows XP như Internet Information Service (IIS), Microsoft .NET Framework 3.5

Hình 5.2: Cơ sở dữ liệu User trên máy DatabaseMáy chủ

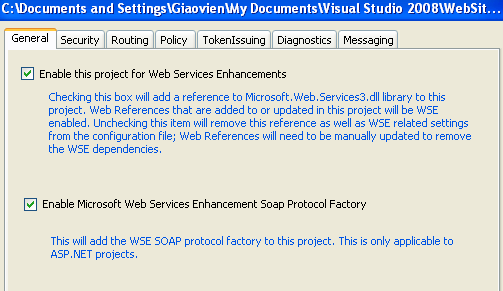
Máy WebMáy chủ thực thi việc gọi đến Web Service đã được public trên hệ thống để hiển thị cơ sở dữ liệu trên máy DatabaseMáy chủ bằng việc sử dụng một trình duyệt

Hình 5.3: WebMáy chủ gọi tới Web Service để hiển thị dữ liệu

## Tích hợp các thẻ bảo mật cho chương trình với công cụ WSE

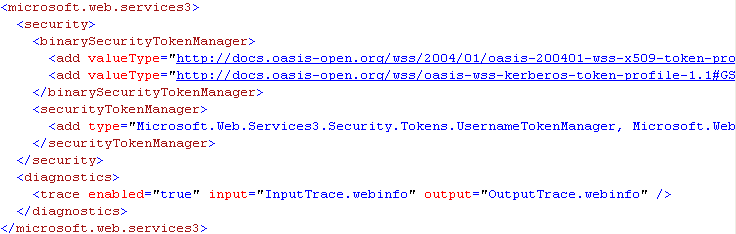
|  |  |
| --- | --- |
| Thực thi WSE với công cụ WSE phải chắc chắn rằng máy tính của chúng ta đã được cài đặt Visual Studio 2008 và bộ thư viện WSE 3.0. Sau đó thực hiện các bước để cấu hình và triển khai bộ thư viện WSE 3.0 lên chương trình. | Hình 5.4: Cấu hình WSE 3.0 |

Sau đó tôi sẽ cấu hình WSE 3.0 để tích hợp các thẻ bảo mật cho chương trình.



Hình 5.5: Triển khai WSE 3.0 cho chương trình hệ thống

Hai chức năng này cho phép chương trình sẽ được đặt trong môi trường bảo mật của WSE 3.0 với bộ thư viện Microsoft.Web.Service3.dll. Tại thẻ Security tôi thêm hai thẻ chính của WSE 3.0 là X509v3 Token Manager và Kerberos Token Manager. Ngoài ra tôi triển khai một Token trong Security Token Managers là Username Token Manager. Sau khi các thẻ trong thư viện WSE 3.0, chương trình đã được bảo mật. Những thông tin trong thông điệp Request mà bên WebMáy chủ gửi đến cho DatabaseMáy chủ để hiển thị dữ liệu cũng được mã hóa và bảo đảm. Và ngay cả trên đường truyền dữ liệu từ WebMáy chủ và DatabaseMáy chủ cũng đã được hỗ trợ bảo mật bởi Firewall được thiết lập mặc định trên các hệ điều hành.



Hình 5.6: Tích hợp thẻ Security vào trong WebService

## Đánh giá kết quả chạy thử nghiệm chương trình

Qua thời gian chạy thử nghiệm cho thấy, chương trình thực hiện được chức năng cơ bản là hiển thị dữ liệu và đảm bảo được một số vấn đề an toàn cần thiết khi trao đổi dữ liệu.

Kết quả: Việc sử dụng thẻ Username Token Manager đã giúp cho các bên tham gia giao tiếp xác thực lẫn nhau và tránh bị giả mạo. Dữ liệu trên đường truyền được mã hóa bởi hai cơ chế bảo mật X509v3 Token và Kerberos Token luôn được đảm bảo về tính bảo mật cao.

# CHƢƠNG 6: KẾT LUẬN

## Tổng kết

Web Service đã và đang được triển khai và áp dụng trong nhiều lĩnh vực đời sống như ngân hàng, chứng khoán, trao đổi dữ liệu … và ngày càng trở lên phổ biến. Cùng với sự phát triển của nó là những đòi hỏi về tính an toàn, khả năng bảo mật. Bằng việc sử dụng các kỹ thuật đảm bảo an ninh Web Service sẽ giúp cho người sử dụng Web Service trở nên an tâm hơn.

Việc chọn cơ chế an toàn cho Web Service phải đòi hỏi sao cho người dùng không cảm thấy quá phức tạp hay gò bó mà phải tạo nên sự trong suốt với người dùng. Do đó, nên chọn các cơ chế an toàn mà Web Service phụ thuộc vào loại dịch vụ đó và những tính năng mà dịch vụ này cung cấp. 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 an ninh Web Service 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.

## Kết quả đạt đƣợc của đồ án tốt nghiệp

Sau thời gian nghiên cứu tài liệu, tìm hiểu các chương trình mã nguồn mở, tôi đã hoàn thành xong đồ án tốt nghiệp với bài toán ban đầu đặt ra là “*Bảo mật Web Service*”. Với việc lựa chọn chương trình trao đổi dữ liệu giữa hai máy tính trong mạng và đảm bảo anh ninh cho việc truyền dữ liệu. Đồ án đã đạt được một số kết quả sau:

Phân tích bài toán và tính cấp thiết của việc đảm bảo an toàn cho các trang Web Service. Đưa ra hướng phát triển cho bài toán.

Nghiên cứu về kiến trúc hướng dịch vụ SOA, Web Service và các thành phần. Mối quan hệ ứng dụng kiến trúc SOA vào xây dựng Web Service và tích hợp chúng theo chuẩn.

Tìm hiểu thực trạng bảo mật Web Service hiện nay, các công nghệ đảm bảo an ninh Web Service như công nghệ bảo mật SSL và bộ thư viện WSE.

Triển khai ứng dụng truyền dữ liệu giữa máy tính trong một mạng cục bộ và tích hợp thẻ bảo mật trong bộ thư viện WSE để bảo mật thông tin cho các bên tham gia.

## Những hạn chế

Để xây dựng được một hệ thống hoàn chỉnh có thêm nhiều chức năng và đảm bảo tuyệt đối những yêu cầu đặt ra, phải cần rất nhiều thời gian. Trong thời gian nghiên cứu và triển khai đồ án, tôi cũng đã cố gắng đạt được những kết quả nhất định, tuy nhiên vẫn còn nhiều hạn chế:

Chương trình khá đơn giản chỉ với chức năng hiển thị dữ liệu, cũng như việc thiết kế dữ liệu chưa thực sự tốt.

Không được đưa ra áp dụng thực tế nên sẽ có khả năng nhiều lỗi mà người nghiên cứu không thể phát hiện ra

Về bảo mật, chưa tìm hiểu hết được các loại thẻ bảo mật Web Service, việc sử dụng bộ thư viện Web Service Enhancement vẫn chỉ dừng lại ở việc tích hợp vào chương trình mức cơ bản nhất, vẫn chưa đưa ra thực tế và sử dụng các phương pháp tấn công để kiểm tra độ bảo mật trên mức cơ bản của chương trình.

Nếu có điều kiện, trong tương lai tôi sẽ cố gắng tìm hiểu thêm về những mặt hạn chế của đồ án này và cố gắng khắc phục để tạo ra một chương trình hoàn chỉnh và có thể áp dụng vào thực tế.

# TÀI LIỆU THAM KHẢO

## Tài liệu Tiếng Việt

[1]Đại học Khoa học tự nhiên – Đại học Quốc gia Hồ Chí Minh (2005),

*Nghiên cứu kiến trúc hướng dịch vụ (Service Oriented Architecture) và ứng dụng.*

## Tài liệu Tiếng Anh

[2]Doug Tidwell – James Snell – Pavel Kulchenko, Publíher: O’Reily(2001),

*Programming Web Services with SOAP.*

[3]Elisha Bertino – Lozenzo D.Martino – Federica Paci – Anna C.Squicciarini,

*Seurity for Web Services and Service Oriented Architecture.*

[4]Freier, A.O, Karlton, Kocher, Scout(1996), *The SSL Protocol Version 3.0*

online: <http://wp.netscape.com/eng/ssl3/draft302.txt>

[5]Hogg,Jane(2006),*Microsoft, Web Service Security, Scenarios, Patterns, and Implementation Guidance for Web Service Enhancements (WSE 3.0)*

[6]Judith Hurwitz, Publisher: For Dummies, *Service Oriented Architecture*. [7]Jeaning Hall Gailey, *Understanding Web Services Specifications and the WSE* [8]OASIS Standard Specification, *Web Service Security Kerberos Token Profile.*

[9]*XML Key Management Specification (XKMS) (W3C Note, 30 Marc 2001),* online: <http://www.w3.org/TR/xkms>