

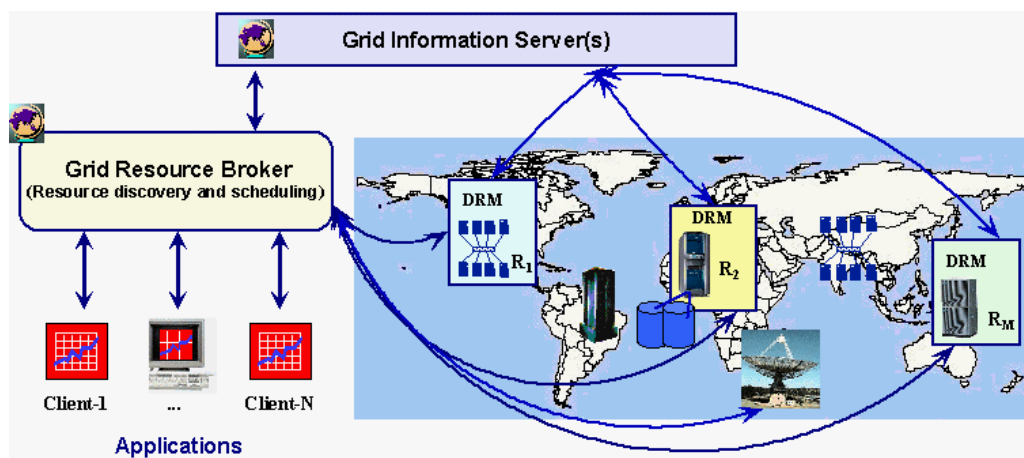
Mục lục

I.	Giới thiệu	2
II.	Kiến trúc định hướng dịch vụ	3
1.	Kiến trúc.....	3
2.	Dịch vụ (SERVICE)	3
3.	Kiến trúc Định hướng Dịch vụ	5
III.	Web Service	8
1.	Cấu trúc Web Service	8
2.	XML – EXTENSIBLE MARKUP LANGUAGE.....	9
3.	SOAP - SIMPLE OBJECT ACCESS PROTOCOL.....	10
4.	WSDL - WEB SERVICES DESCRIPTION LANGUAGE.....	13
IV.	Grid Service.....	14
V.	Open Grid Service Architecture (OGSA)	14
1.	Khái niệm	14
2.	OPEN GRID SERVICES INFRASTRUCTURE	15
3.	DATA AND INFORMATION SERVICES	19
4.	RESOURCE AND SERVICE MANAGEMENT	19
VI.	WSRF - Web Service Resource Framework	19
1.	WEB SERVICE RESOURCE FRAMEWORK.....	19
2.	Hiện thực WSRF :	22
VII.	Kết luận.....	23

I. Giới thiệu

Sự phát triển ngày càng phổ biến của Internet, cùng với năng lực tính toán của máy tính ngày càng mạnh và mạng tốc độ cao cũng như các thiết bị có chi phí ngày càng thấp đang thay đổi cách tính toán và sử dụng các máy tính. Các tài nguyên được phân bố theo các vị trí địa lý khác nhau, cần phải được liên kết với nhau để phục vụ các bài toán tính toán lớn. Chính vì vậy cần áp dụng tính toán lưới để giải quyết việc kết hợp các tài nguyên đó lại.

Trong môi trường này, nhiều tài nguyên tính toán như các siêu máy tính, các cụm máy tính, thiết bị trực quan, hệ thống lưu trữ và cơ sở dữ liệu, đặc biệt các thiết bị khoa học như kính thiên văn kết nối ở mức luận lý với nhau và thể hiện ra bên ngoài đến người sử dụng như là một tài nguyên tích hợp đơn (*single integrated resource*) (xem hình 1).



Hình 1. Góc nhìn tổng quát của hệ thống GRID

Về cơ bản, người sử dụng tương tác với trung tâm môi giới tài nguyên (*resource broker*). Như vậy sẽ không thấy được những phức tạp của tính toán lưới. Khi broker biết được tài nguyên mà người sử dụng có thể truy cập thông qua một hoặc nhiều hệ thống quản lý thông tin lưới (*grid information server*), broker sẽ đàm phán với các tài nguyên hoặc các nhà cung cấp của chúng bằng cách sử dụng các dịch vụ trung gian. Khi đã đạt được thỏa thuận với các tài nguyên, sẽ thực hiện việc lập lịch (*scheduling*) bằng cách ánh xạ các công việc đến các tài nguyên, sắp xếp ứng dụng và dữ liệu để xử lý và cuối cùng trả về tập hợp kết quả. Trong quá trình đó,

cần theo dõi tiến trình thực hiện ứng dụng để việc quản lý các thay đổi trong cơ sở hạ tầng lưới (grid infrastructure) và lỗi tài nguyên.

Có một số dự án trên thế giới, đang phát triển các thành phần, các dịch vụ và các ứng dụng trên hệ thống tính toán lưới: Globus, Legion, NetSolve, Ninf, Apple, Nimrod/G, và JaWS.

Trong môi trường tính toán lưới, việc truy cập vào các tài nguyên phân bố theo cơ chế truy cập ngang hàng. Vì vậy cần phải áp dụng các chính sách bảo mật cũng như xác thực quyền truy cập của người dùng.

Với việc Internet ngày càng phát triển mạnh, nhu cầu sử dụng các tài nguyên sẵn có ngày càng nhiều, và các tài nguyên đó tương tác với nhau để phục vụ các yêu cầu của người sử dụng. Tuy nhiên, trong môi trường phân bố, các tài nguyên đó lại được quản lý bởi các tổ chức khác nhau, có chính sách khác nhau vì vậy sẽ phát sinh các vấn đề rất phức tạp trong việc kết hợp các tài nguyên này với nhau.

→ Vì vậy cần có mô hình kiến trúc quản lý tài nguyên.

II. Kiến trúc định hướng dịch vụ

1. Kiến trúc

Một mô tả hình thức của một hệ thống, xác định mục đích, chức năng, thuộc tính bên ngoài có thể nhìn thấy, và các giao diện của nó. Nó cũng bao gồm các mô tả của các thành phần bên trong của hệ thống và các mối quan hệ của chúng, cùng với các nguyên tắc điều chỉnh thiết kế, hoạt động của nó, và sự tiến hóa.

2. Dịch vụ (service)

Về mặt định nghĩa, dịch vụ 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 interface đã được định nghĩa trước đó. Thông thường việc giao tiếp này được thực hiện trên các giao thức đã được chuẩn hóa và sử dụng rộng rãi.

Một ví dụ đơn giản của dịch vụ chính là hoạt động của một nhà hàng. Khi khách hàng vào nhà hàng và gọi thức ăn, khách hàng đang tiến hành gọi yêu cầu cho dịch vụ “phục vụ khách hàng” của nhà hàng. Nhân viên nhà hàng nhận được yêu cầu của khách, nếu món ăn khách hàng yêu cầu nhà hàng không có hoặc đã hết, nhân viên nhà hàng sẽ từ chối hoặc đề nghị khách hàng gọi món khác. Nếu nhà hàng có thể đáp ứng được yêu cầu của khách, món ăn sẽ được chế biến và mang ra cho khách hàng thưởng thức sau một khoảng thời gian chờ. Ở đây, yêu cầu chính là món

ăn mà khách hàng muốn thưởng thức, còn kết quả trả về của dịch vụ phục vụ nhà hàng chính là từ chối (nếu nhà hàng không đáp ứng được yêu cầu của khách) hay là món ăn mà khách hàng cần.

Một hệ thống được thiết kế theo kiểu hướng dịch vụ (service oriented) 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 dịch vụ trong hệ thống giao tiếp với nhau thông qua việc gửi nhận các thông điệp (message).

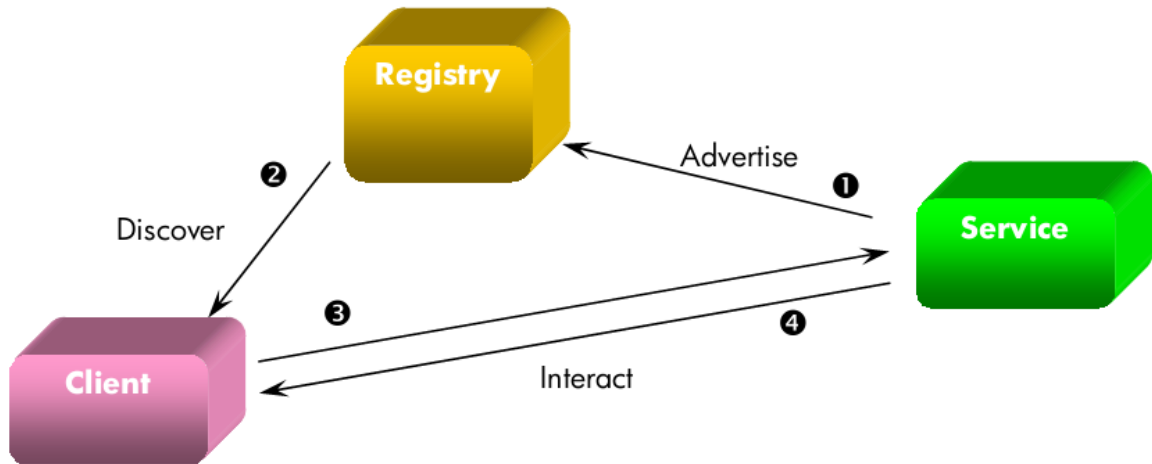
a. Các đặc điểm chính của dịch vụ

- Các dịch vụ có thể là những cá thể độc lập hữu ích, hoặc chúng được tích hợp kết hợp với nhau để tạo ra những dịch vụ ở mức cao hơn. Ưu điểm chính của đặc điểm này là khả năng tái sử dụng các chức năng đã có sẵn.
- Các dịch vụ giao tiếp với khách hàng của họ bằng cách trao đổi thông điệp: chúng được định nghĩa bởi các thông điệp mà họ có thể chấp nhận và các phản hồi mà họ có thể cung cấp.
- Các dịch vụ có thể tham gia vào quy trình làm việc, trong đó thứ tự tin nhắn được gửi và nhận ảnh hưởng đến kết quả của các hoạt động thực hiện bởi một dịch vụ. Khái niệm này được định nghĩa là "dịch vụ biên đạo".
- Các dịch vụ có thể hoàn toàn khép kín, hoặc chúng có thể phụ thuộc vào sự sẵn có của các dịch vụ khác, hoặc vào sự tồn tại của một nguồn tài nguyên như một cơ sở dữ liệu. Trong trường hợp đơn giản, một dịch vụ có thể thực hiện một phép tính như tính căn bậc ba của một số cho trước mà không cần phải tham khảo bất kỳ tài nguyên bên ngoài, hoặc nó có thể đã được cài sẵn tất cả các dữ liệu mà nó cần trong suốt thời gian sống của nó. Ngược lại, một dịch vụ thực hiện chuyển đổi tiền tệ cần truy xuất theo thời gian thực đến các thông tin tỷ giá hối đoái để mang lại giá trị chính xác.
- Các dịch vụ quảng cáo các chi tiết chẳng hạn như những khả năng, giao diện, chính sách, và các giao thức liên lạc mà chúng hỗ trợ. Các chi tiết thực hiện như ngôn ngữ lập trình và nền tảng lưu trữ không liên quan đến khách hàng, và không được tiết lộ.

b. Sự tương tác giữa các dịch vụ

Hình 1 minh họa một chu kỳ dịch vụ tương tác đơn giản, bắt đầu với một dịch vụ quảng cáo riêng của mình thông qua một dịch vụ đăng ký có tiếng (1). Một khách hàng tiềm năng, có thể là hoặc không là một dịch vụ, truy vấn đến cơ quan đăng ký (2) để tìm kiếm một dịch vụ đáp ứng được

nhu cầu của nó. Đăng ký trả về một danh sách (có thể rỗng) của các dịch vụ phù hợp, và khách hàng lựa chọn và gửi một thông điệp yêu cầu ngược trở lại, bằng cách sử dụng bất kỳ giao thức đã được xác định chung (3). Trong ví dụ này, dịch vụ sẽ phản hồi (4), hoặc với các kết quả của yêu cầu hoặc với một thông báo lỗi.



Hình 2. Sự tương tác của các dịch vụ trong môi trường định hướng dịch vụ

Sự minh họa này cho thấy trường hợp đơn giản, nhưng trong một khung cảnh thực thế giới, chẳng hạn như một ứng dụng thương mại, quá trình có thể được đáng kể phức tạp hơn. Ví dụ, một dịch vụ nhất định có thể chỉ hỗ trợ các giao thức HTTPS, được giới hạn người dùng có thẩm quyền, yêu cầu xác thực Kerberos, cung cấp các mức độ khác nhau về hiệu suất cho người sử dụng khác nhau, hoặc yêu cầu thanh toán để sử dụng. Dịch vụ có thể cung cấp chi tiết như vậy trong nhiều cách khác nhau, và khách hàng có thể sử dụng thông tin này để tạo vùng chọn của nó. Một số thuộc tính, chẳng hạn như điều khoản thanh toán và mức độ của dịch vụ bảo lãnh, có thể cần phải được thành lập bởi một quá trình đàm phán trước khi khách hàng có thể sử dụng các dịch vụ đã chọn. Và, trong khi minh họa này cho thấy một đơn giản, đồng bộ, mô hình trao đổi thông điệp hai chiều, một loạt các mẫu có thể cho ví dụ, một sự tương tác có thể là một chiều, hoặc các phản ứng có thể không phải đến từ các dịch vụ mà khách hàng gửi yêu cầu, nhưng từ một số dịch vụ khác đã hoàn tất giao dịch.

3. Kiến trúc định hướng dịch vụ

Kiến trúc định hướng dịch vụ (SOA) cung cấp cơ chế cho phép các hệ thống hoạt động trên các nền tảng khác nhau có thể giao tiếp với nhau.

Một hệ thống được xây dựng theo SOA bao gồm các dịch vụ thỏa mãn các đặc điểm chính của dịch vụ đã nêu ở trên. Mỗi dịch vụ trong hệ thống có thể được sửa đổi một cách độc lập với các dịch vụ khác nhằm mục đích đáp ứng một yêu cầu mới từ thực tế.

Trong công nghệ phần mềm, SOA là một tập hợp các nguyên tắc và phương pháp thiết kế và phát triển phần mềm trong các hình thức dịch vụ tương thích.

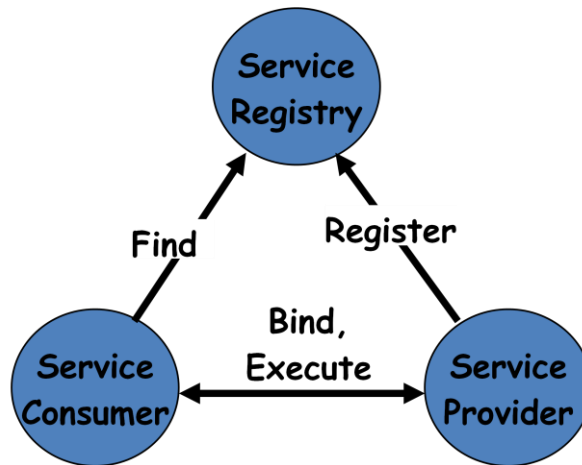
Theo OASIS (Organization for the Advancement of Structured Information Standards): SOA là một mô hình tổ chức và sử dụng khả năng phân phối có thể là dưới sự kiểm soát của các lĩnh vực quyền sở hữu khác nhau. Nó cung cấp thống nhất một phương tiện để cung cấp, khám phá, tương tác với và sử dụng khả năng để sản xuất hiệu ứng mong muốn phù hợp với điều kiện tiên quyết thể đo lường được và mong đợi.

Theo Thomas ERL – một trong những người có những đóng góp lớn trong lĩnh vực SOA, điện toán đám mây: SOA đại diện cho một kiến trúc mở, nhanh nhẹn, mở rộng, liên, composable bao gồm của tự trị, có khả năng QoS, nhà cung cấp dịch vụ đa dạng, tương thích, phát hiện, và có khả năng tái sử dụng, thực hiện như các dịch vụ Web.

a. Các nhân tố của SOA

Hình 3 mô tả các nhân tố tham gia trong một hệ thống xây dựng theo SOA.

- Service Provider: Cung cấp các dịch vụ không trạng thái phục vụ cho một nhu cầu nào đó mà người sử dụng (service consumer) không cần quan tâm đến vị trí thực sự mà dịch vụ họ cần sử dụng đang hoạt động.
- Service Consumer: người dùng sử dụng dịch vụ được cung cấp bởi Service Provider
- Service Registry: Nơi lưu trữ thông tin về các dịch vụ của các Service Provider khác nhau, Service Consumer dựa trên những thông tin này để tìm kiếm và lựa chọn Service Provider phù hợp.



Hình 3. Mô hình hoạt động của SOA

Service Provider sẽ đăng kí thông tin về dịch vụ mà mình có thể cung cấp (các chức năng có thể cung cấp, khả năng của hệ thống (tài nguyên, hiệu suất), giá cả dịch vụ, ...) vào Service Registry. Service Consumer khi có nhu cầu về một dịch vụ nào đó sẽ tìm kiếm thông tin trên Service Registry. Ngoài chức năng hỗ trợ tìm kiếm, Service Registry còn có thể xếp hạng các Service Provider dựa trên các tiêu chí về chất lượng dịch vụ, bầu chọn từ các khách hàng đã sử dụng dịch vụ, ... Những thông tin này sẽ hỗ trợ thêm cho quá trình tìm kiếm của Service Consumer. Khi đã xác định được Service Provider mong muốn, Service Consumer thiết lập kênh giao tiếp trực tiếp với Service Provider nhằm sử dụng tài nguyên hoặc tiến hành thương lượng thêm (về mặt giá cả, tài nguyên sử dụng, ...)

b. Ưu điểm của SOA

Sử dụng mô hình SOA trong việc thiết kế hệ thống mang lại lợi ích về mặt kinh tế cũng như kỹ thuật.

- Lợi ích kinh tế
 - Doanh nghiệp có điều kiện tập trung thời gian để tìm kiếm các giải pháp cho các bài toán liên quan đến kinh tế.
 - Thúc đẩy sự phát triển của hệ thống hiện có cũng như cung cấp khả năng mở rộng hệ thống trong tương lai.
- Lợi ích kỹ thuật
 - Hệ thống xây dựng theo mô hình SOA đảm bảo các service trong hệ thống có tính độc lập cao (độ kết dính thấp) (autonomous và loose coupling).
 - Ở góc nhìn người sử dụng, vị trí các service có tính trong suốt (transparency), việc di dời các service đến một máy

tính khác không ảnh hưởng khả năng phục vụ yêu cầu khách hàng.

- Hoạt động của các service có tính động, hành vi của các service tùy thời điểm, tùy yêu cầu cần xử lý mà có sự khác nhau (late binding).

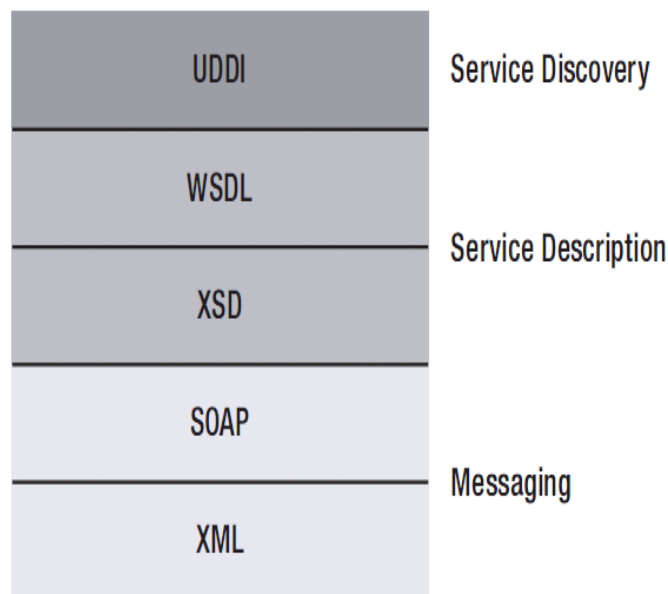
III. Web Service

Theo Wiki: nó là một phương pháp giao tiếp giữa hai thiết bị điện tử trên web.

Theo W3C: nó là một hệ thống phần mềm được thiết kế để hỗ trợ tương tác giữa máy này với máy khác thông qua mạng.

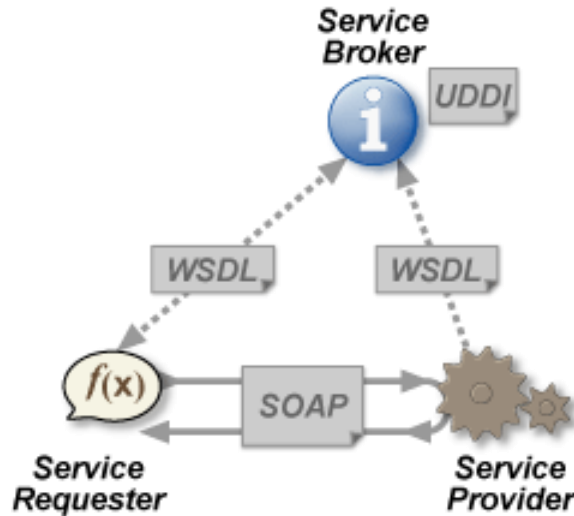
Webservice là một hiện thực của service sử dụng web protocol để giao tiếp. Web service cung cấp hệ thống phần mềm hỗ trợ việc giao tiếp giữa các máy tính trên mạng. Một tiêu chuẩn của việc tích hợp các ứng dụng dựa trên web bằng cách sử dụng XML, SOAP, WSDL và UDDI tiêu chuẩn mở thông qua một nền tảng giao thức Internet.

1. Cấu trúc web service



Hình 4. Cấu trúc web service

Hình 4 mô tả các tầng hình thành nên Web service. Hình 5 mô tả các thành phần tương ứng với các tầng này.



Hình 5. Sơ đồ hoạt động của Web service

Webservice có thể coi là một mô hình SOA trong thực tế, sơ đồ hoạt động cũng gồm 3 thành phần chính:

- **Service Provider:** Dùng Web Services Description Language (WSDL) để mô tả dịch vụ mà mình có thể cung cấp cho Service Broker (tương tự với Service Registry trong SOA).
- **Service Broker:** Lưu trữ thông tin về các service được cung cấp bởi các Service Provider. Cung cấp chức năng tìm kiếm hỗ trợ Service Requester (Service Consumer trong SOA) trong việc xác định Service Provider phù hợp. Thành phần chính của Service Broker là Universal Discovery, Description, and Integration (UDDI) repositories.
- **Service Requester:** Dùng WSDL để đặc tả nhu cầu sử dụng (loại service, thời gian sử dụng, resource cần thiết, mức giá ...) và gọi cho Service Broker. Bằng việc sử dụng UDDI và chức năng tìm kiếm của Service Broker, Service Requester có thể tìm thấy Service Provider thích hợp. Ngay sau đó, giữa Service Requester và Service Provider thiết lập kênh giao tiếp sử dụng SOAP để thương lượng giá cả và các yếu tố khác trong việc sử dụng service.

2. XML – eXtensible Markup Language

Nó là một ngôn ngữ đánh dấu để định dạng và trao đổi dữ liệu có cấu trúc

Nó cho phép các nhà thiết kế để tạo ra các thẻ tùy biến của họ, cho phép định nghĩa, truyền tải, xác nhận, và giải thích dữ liệu giữa các ứng dụng và giữa các tổ chức.

```
<note>
  <to>Tove</to>
  <from>Jani</from>
  <heading>Reminder</heading>
  <body>Don't forget me this weekend!</body>
</note>
```

Hình 6. Ví dụ về một đoạn mã XML

3. SOAP - Simple Object Access Protocol

SOAP là một protocol giao tiếp dùng trong Web service được xây dựng dựa trên XML. SOAP được sử dụng để đặc tả và trao đổi thông tin về các cấu trúc dữ liệu cũng như các kiểu dữ liệu giữa các thành phần trong hệ thống.

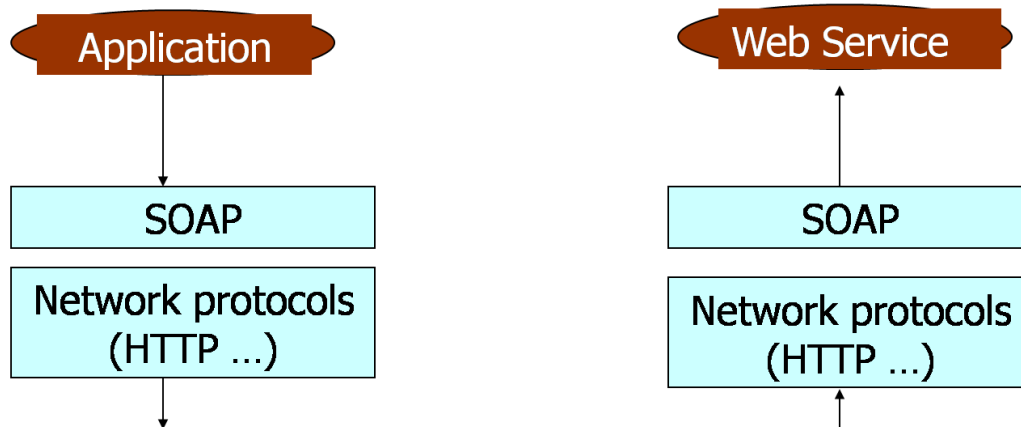
Sử dụng SOAP, ứng dụng có thể yêu cầu thực thi các phương thức trên máy tính ở xa mà không cần quan tâm đến chi tiết về nền tảng cũng như các phần mềm trên máy tính đó.

a. Các đặc điểm của SOAP

- Khả năng mở rộng (Extensible): Cung cấp khả năng mở rộng phục vụ cho nhu cầu đặc thù của ứng dụng và nhà cung cấp. Các chức năng về bảo mật, tăng độ tin cậy có thể đưa vào phần mở rộng của SOAP. Các nhà cung cấp dịch vụ khác nhau, tùy vào đặc điểm hệ thống của mình có thể định nghĩa thêm các chức năng mở rộng nhằm tăng thêm lợi thế cạnh tranh cũng như cung cấp thêm tiện ích cho người sử dụng.
- Có thể hoạt động trên các network protocol đã được chuẩn hóa (HTTP, SMTP, FTP, TCP, ...)
- Độc lập với nền tảng sử dụng cũng như ngôn ngữ lập trình hay mô hình lập trình được sử dụng.

b. SOAP Message Structures

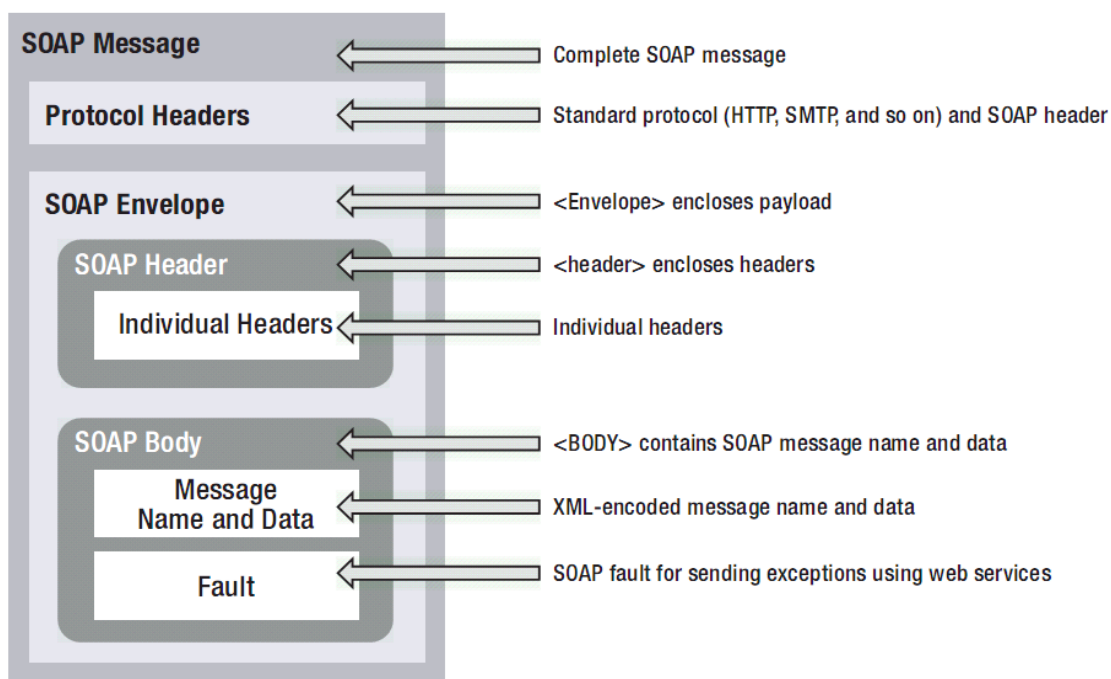
Hình 7 mô tả giao tiếp của một ứng dụng với một web service được thực hiện qua thông điệp SOAP sử dụng network protocol HTTP. Ứng dụng sẽ đặc tả yêu cầu trong SOAP message và thông qua network protocol gửi đến cho web service. Web service sẽ nhận và phân tích yêu cầu sau đó trả về kết quả thích hợp.



Hình 7. Truyền thông điệp sử dụng SOAP

Hình 8 mô tả cấu trúc một thông điệp SOAP. Một thông điệp SOAP bao gồm các thành phần sau:

- **Protocol Header:** Cho biết thông tin về các chuẩn giao thức được sử dụng.
- **SOAP Envelop:** Thông tin chính của message bao gồm:
 - SOAP Header: Chứa các SOAP header.
 - SOAP body: Thông tin về name và data được đặc tả dưới dạng XML. Ngoài ra còn có trường lỗi được dùng để gọi các web service exception.



Hình 8. Cấu trúc thông điệp SOAP

Sau đây là ví dụ về thông điệp SOAP, trong đó thông điệp yêu cầu được sử dụng để yêu cầu web service thực hiện phương thức 'int doubleAnInteger (int numberToDouble)' nhằm nhân đôi một số nguyên. Thông điệp yêu cầu sẽ truyền đổi số 123(số nguyên cần nhân đôi) trong phần nội dung của SOAP. Thông điệp trả về mang thông tin kết quả trả về trong phần nội dung. Kết quả của phép nhân đôi số nguyên 123 (được truyền bởi message request) là 246.

- Request

```
<?xml version="1.0" encoding="UTF-8" standalone="no" ?>
<SOAP-ENV:Envelope
  SOAP-
  ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
  xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/"
  xmlns:xsi="http://www.w3.org/1999/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/1999/XMLSchema">
  <SOAP-ENV:Body>
    <ns1:doubleAnInteger
      xmlns:ns1="urn:MySoapServices">
      <param1 xsi:type="xsd:int">123</param1>
    </ns1:doubleAnInteger>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

- Response

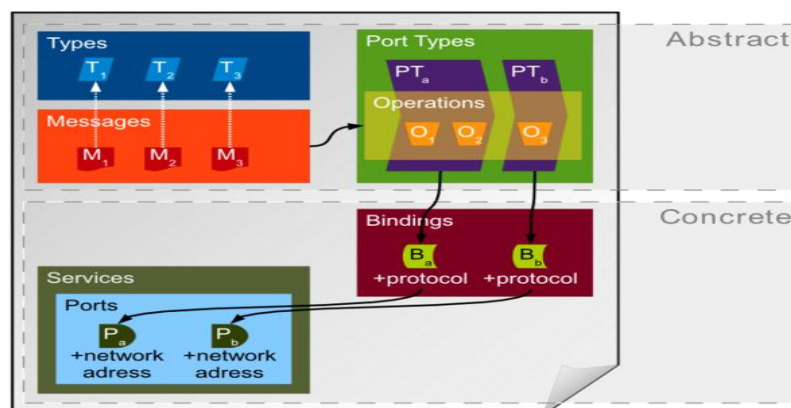
```
<?xml version="1.0" encoding="UTF-8" ?>
<SOAP-ENV:Envelope
  xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsi="http://www.w3.org/1999/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/1999/XMLSchema">
  <SOAP-ENV:Body>
    <ns1:doubleAnIntegerResponse
      xmlns:ns1="urn:MySoapServices"
      SOAP-ENV:
      encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
      <return xsi:type="xsd:int">246</return>
    </ns1:doubleAnIntegerResponse>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

4. WSDL - Web Services Description Language

Việc đặc tả chức năng, interface giao tiếp của một web service được thực hiện dựa vào Web Service Description Language (WSDL). Việc đặc tả này bao gồm protocol được sử dụng, cấu trúc và định dạng dữ liệu dùng để giao tiếp...

Hình 9 mô tả các thành phần cơ bản của một file WSDL dùng để đặc tả một web service.

- **Services:** Chứa các method có thể được sử dụng thông qua các web protocol.
- **Ports:** Địa chỉ dùng để kết nối đến web service. Thông thường, ports được mô tả bằng một HTTP URL.
- **Port Types:** 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.
- **Operations:** Mỗi operation có thể được xem như một method hay một lời gọi hàm trong các ngôn ngữ lập trình cổ điển.
- **Binding:** chỉ định port type, các operation, SOAP binding style (RPC/Document), SOAP protocol được dùng.
- **Message:** Mỗi message tương ứng với một hành động và chứa các thông tin cần thiết để thực thi hành động đó. Mỗi message có một định danh duy nhất và một hay nhiều thành phần logic. Các logical part được phân biệt với nhau qua name và có thể lưu trữ các tham số cần cho hành động.



Hình 9. Tổng quan WSDL

- **Element:** Được định nghĩa trong Types. Mỗi element có một name duy nhất và kiểu dữ liệu. Element được dùng để đặc tả dữ liệu dùng trong message. Element có thể đặc tả các dữ liệu đơn giản (string, integer) hay phức tạp hơn như array, struct, ...
- **XSD file:** Các element thường được định nghĩa trong các XML Schema Definition (XSD) file. XSD file có thể ở trong cùng file WSDL hoặc ở file riêng biệt.

IV. Grid Service

Về mặt định nghĩa, Grid Service là Web Service được thay đổi cho phù hợp với môi trường grid.

Các đặc điểm của Grid service :

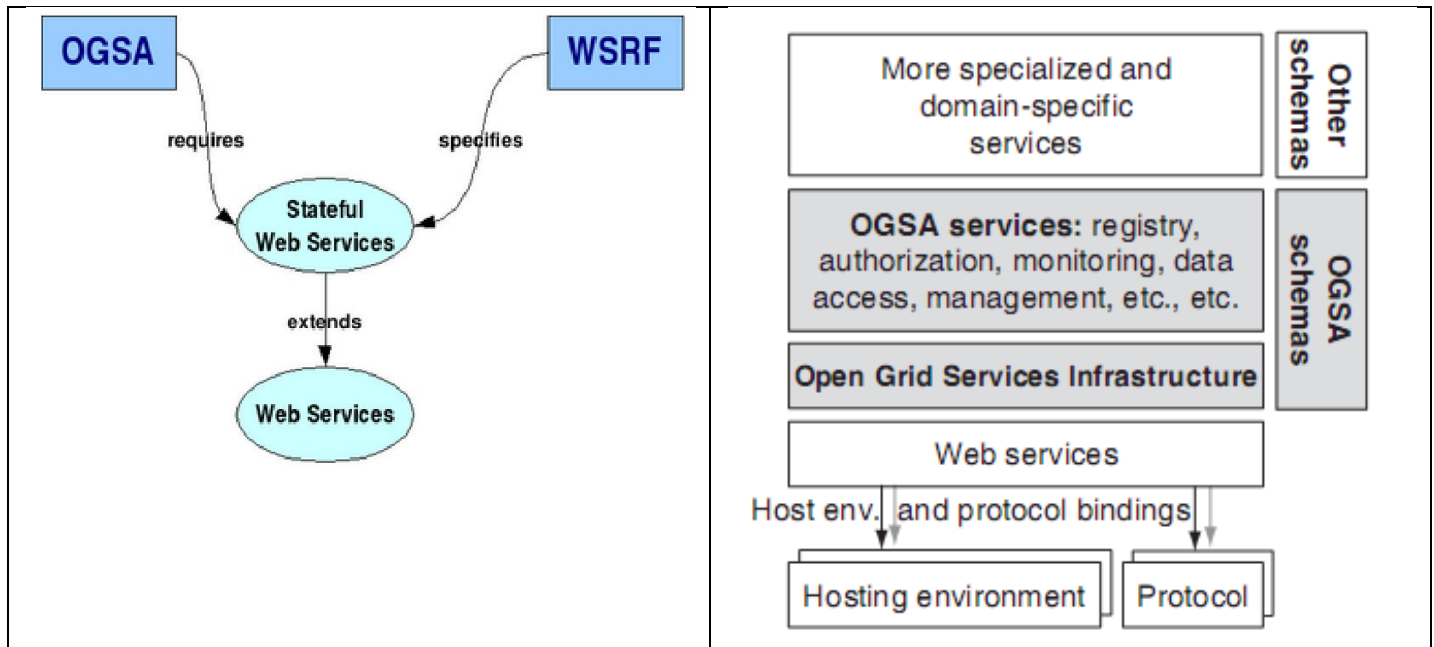
- **Transient in Nature (Lifetime, Migration):** đối với môi trường grid, khi có một yêu cầu được gửi đến, một service sẽ được tạo ra để đáp ứng yêu cầu đó. Khi service đáp ứng xong yêu cầu, service sẽ bị hủy. Đối với web service, khi nhận được một yêu cầu mới, service sẽ tạo ra một thread để xử lý yêu cầu đó.
- **Stateful Service:** Grid service là stateful, grid service cần lưu trữ thông tin về mỗi session giữa consumer và provider. So với Web Service, Grid Service sử dụng Service Data nhằm lưu trữ thông tin về trạng thái hiện tại của service (state information) (kết quả cuối cùng hay kết quả trung gian của các operation, thông tin về runtime của hệ thống, ...) hoặc thông tin về chính service (service metadata) (system data, các interface được hỗ trợ, giá sử dụng service, ...)
- Grid service interface được đặc tả thông qua GWSDL (Grid WSDL).
- Cung cấp cơ chế quản lý thời gian sống (lifecycle management).
- Grid Service cung cấp cơ chế định nghĩa các portType mở rộng. Mọi grid service đều phải thừa kế từ portType chuẩn GridService. Ngoài ra, mỗi grid service có thể thừa kế (extend) các portType của hệ thống cung cấp sẵn nhằm tăng thêm tính năng của service.

V. Open Grid Service Architecture (OGSA)

1. Khái niệm

Open Grid Services Architecture (OGSA) được phát triển bởi [The Global Grid Forum](#) với mục đích đưa ra kiến trúc chung, chuẩn và mở cho các ứng dụng tính toán lưới. OGSA hướng đến việc chuẩn hóa các

service trong ứng dụng lưới bằng cách đưa các interface chuẩn cho các service này.



Hình 10. Mô hình OGSA

Các thành phần chính của OGSA

- Open Grid Services Infrastructure
- OGSA services
- OGSA schemas

OGSA được xây dựng dựa trên Web services. OGSA có thể được triển khai trên nhiều môi trường khác nhau, giao tiếp với nhau thông qua các giao thức (protocol).

Web services là công cụ được sử dụng khá phổ biến tuy nhiên bản thân Web services vẫn còn một số những hạn chế như:

- Khởi tạo webservice
- Thời gian sống của webservice
- Quản lý lỗi

OGSA giải quyết các vấn đề trên và được hiện thực ở lớp cơ sở hạ tầng (OGSI). Web services tuân theo các chuẩn OGSI được gọi là Grid Service.

OGSI định nghĩa các chuẩn của giao tiếp và các hành vi phục vụ cho việc mô tả và tìm kiếm dịch vụ, khởi tạo instance, quản lý thời gian sống, nhóm service...

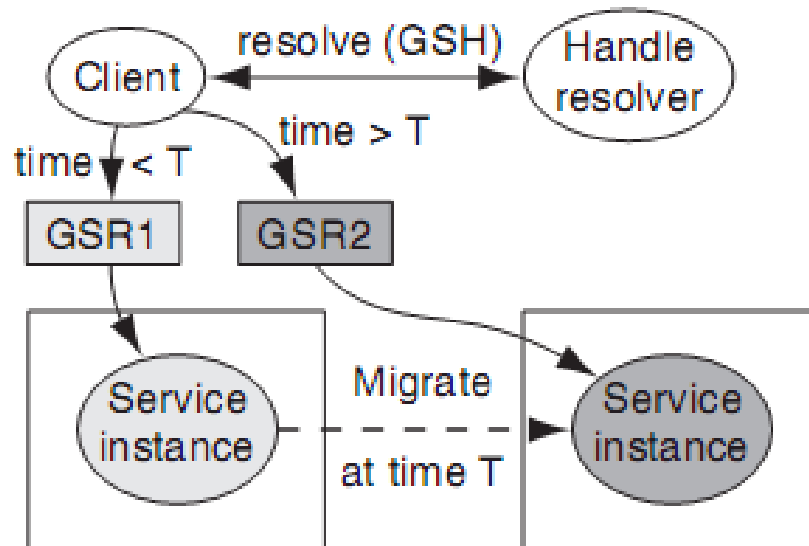
2. Open Grid Services Infrastructure

a. Các đặc tính của OGSI

- Grid service descriptions and instances
 - Trong mô hình Web services chuẩn, service được tạo ra cũng như bị hủy nằm ngoài phạm vi của bản thân Web services. Tuy nhiên trong thực tế, ứng dụng thường xuyên khởi tạo service mới và khi các tác vụ đã hoàn tất thì Web services bị hủy đi.
 - OGSi cung cấp một cơ chế phân biệt giữa phần mô tả và phần instance của service. Phần mô tả định nghĩa các interface và hành vi của service. Các service instance sẽ hiện thực các hành vi này.
- Service state, metadata, and introspection
 - WSDL chuẩn không có khái niệm “trạng thái” của service. Vì thế OGSi đưa ra khái niệm service data để biểu diễn siêu dữ liệu và các trạng thái. Các thông tin này được xem như một phần mô tả của service và có thể được truy xuất từ service instance.
 - Ứng dụng client có thể yêu cầu service instance trả về các thông tin của chính service đó. Ví dụ: các interface, đặc tả thông tin trạng thái hiện tại của service, thời điểm kết thúc của service...
 - OGSi cung cấp hai cơ chế truy xuất trạng thái ứng dụng:
 - Pull mode: OGSi cung cấp tác vụ *findServiceData* kết hợp với service interface cho phép client có thể query thông tin trạng thái ứng dụng.
 - Push mode: client đăng ký các với service thông tin trạng thái mình mong muốn thông qua các interface *NotificationSource*, *NotificationSubscription*, và *NotificationSink*. Khi trạng thái service thay đổi, service sẽ thực hiện hành vi callback thông báo cho client biết.
- Naming và name resolution
 - Vì Grid service được khởi tạo động và có trạng thái nên cần phải có cách thức để phân biệt giữa các instance của grid service. Do đó ta cần phương thức đặt tên cho các instance này.
 - OGSi đưa ra cơ chế đặt tên hai mức. Mỗi grid instance sẽ được gán một Grid Service Handle(GSH) và được quản lý

bởi HandleResolver. GSH là toàn cục và duy nhất tồn tại trong suốt thời gian sống của instance.

- Client thể giao tiếp với service instance cũng như các service instance với nhau thông qua Grid Service Reference(GSR).
- Không giống với GSH, GSR được tạo ra, thay đổi, hủy trong suốt thời gian sống của service instance.



Hình 11. Cơ chế phân giải tên

- Service life cycle
 - Một trong những điểm khác biệt chính giữa grid service và Web service là Grid service có thể transient(tạo và hủy). OGSi cung cấp cơ chế cho phép các instance khởi tạo, hủy cũng như quản lý thời gian sống của các instance này.
 - Service instance có thể quy định thời gian sống bằng các phương thức sau:
 - Khởi tạo ban đầu: khi có yêu cầu khởi tạo mới một grid service, client gửi kèm khoảng thời gian sống cụ thể của service instance.
 - Kết thúc tường minh(Explicit termination): client chủ động gửi yêu cầu hủy grid service instance.
 - Yêu cầu điều chỉnh thời gian sống(Requesting a lifetime modification): client có thể yêu cầu điều

chỉnh thời gian sống của service instance bằng cách gửi thông điệp “keepalive” đến server.

- Fault type : OGSi định nghĩa chuẩn các kiểu chuẩn cho tất cả các thông điệp lỗi được trả về bởi Grid service.
- Service groups: OGSi cung cấp cơ chế tổ chức service instance theo nhóm. OGSi định nghĩa các interface (WSDL portTypes). Mỗi loại portType đưa ra tập các tác vụ và OGSi quy định hành vi cụ thể cho các tác vụ này. Hình 12 liệt kê một số các portType

PortType	Mô tả
GridService	Cung cấp các hành vi chính bao gồm phần dữ liệu và quản lý thời gian sống của service.
HandleResolver	Ánh xạ từ GSH sang GSR
NotificationSource	Cho phép client đăng ký các thông điệp notification
NotificationSubscription	Quản lý thời gian sống và các thuộc tính của các subscription.
NotificationSink	Định nghĩa tác vụ cho việc phân phối các thông điệp notification đến service instance.
Factory	Tác vụ chuẩn tạo grid service instance
ServiceGroup	Quản lý nhóm các service instance
ServiceGroupRegistration	Cho phép thêm hoặc xóa Grid service ra khỏi ServiceGroup

Hình 12. Danh sách port type

b. Các dịch vụ lõi của OGSi

Core service cung cấp các dịch vụ về phân giải tên, tìm kiếm dịch vụ, bảo mật, chính sách, và truyền thông điệp. Các dịch vụ cơ bản:

- Name resolution and discovery
- Service domains

- Security
 - Policy
 - Messaging, queuing, and logging
 - Events
 - Metering and accounting
3. Data and Information Services
 - Data naming and access
 - Replication
 - Metadata and provenance
 4. Resource and Service Management
 - Provisioning and resource management
 - Service orchestration
 - Transactions
 - Administration and deployment
- VI. WSRF - Web Service Resource Framework
1. Web Service Resource Framework

OGSI cung cấp một cơ cấu cho phép phát triển, triển khai và quản lý thời gian sống (lifetime management) các dịch vụ lưới. Tuy nhiên, OGSI vẫn còn để lộ nhiều điểm yếu như:

- ✓ Giải quyết quá nhiều vấn đề trong cùng một bản đặc tả.
- ✓ GWSDL không theo chuẩn của Web service khiến các công cụ có sẵn trong Web service không sử dụng được.
- ✓ Sự nhập nhằng giữa Web service không trạng thái (stateless) và Web service có trạng thái (stateful).

WSRF trên cơ sở nhóm lại các tác vụ của OGSI đồng thời sử dụng các chuẩn mới hình thành của kiến trúc Web services như WS-Addressing, WS-Notification để giải quyết các nhược điểm của OGSI. WSRF chứa tất cả các chức năng của OGSI và cải tiến OGSI ở ba bước sau:

- ✓ Giới thiệu khái niệm WS-Resource
- ✓ Tách biệt giữa chức năng dịch vụ và việc kích hoạt dịch vụ
- ✓ Dịch vụ notification theo chuẩn WS-Notification của Web services.

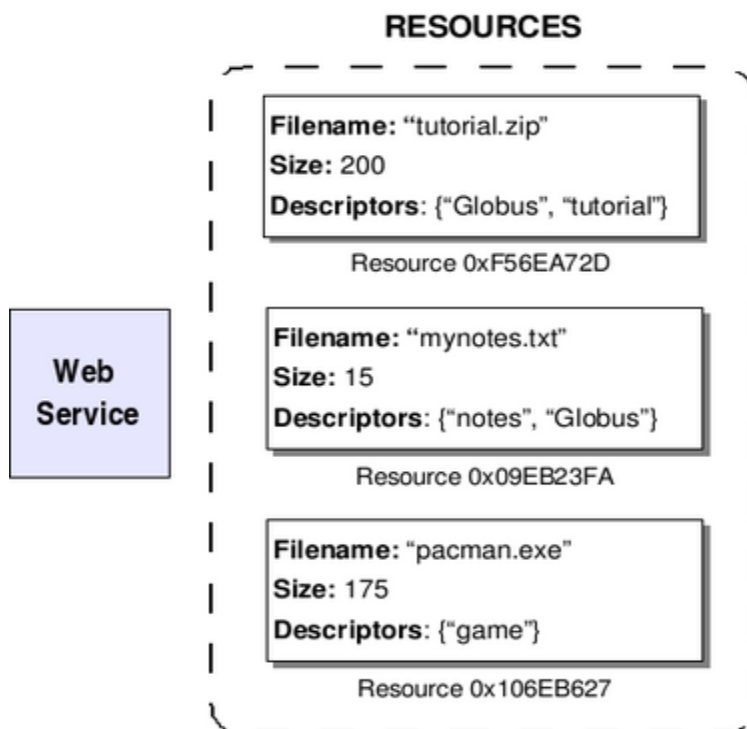
WSRF nhóm lại các chức năng của OGSI thành năm nhóm chính sau:

- ✓ WS-ResourceProperties: đặc tả các dữ liệu của tài nguyên, tương tự như Grid Data Elements của OGSI, và các cơ chế cho phép truy cập, xóa hoặc thay đổi các thông tin này.

- ✓ **WS-ResourceLifetime**: cơ chế kết thúc một WS-Resource; các đặc tả về thông tin trao đổi giữa người yêu cầu dịch vụ và WS-Resource nhằm kết thúc dịch vụ ngay lập tức hoặc là một thời điểm định sẵn.
- ✓ **WS-RenewableReferences**: dựa trên chuẩn WS-Addressing của Web service để cập nhật các thông tin mới về địa chỉ dịch vụ (endpoint references) khi các thông tin hiện tại trở thành không hợp lệ.
- ✓ **WS-ServiceGroup**: đặc tả giao tiếp cho phép một tài nguyên tham gia vào nhóm; đặc tả các tác vụ cho phép lấy thông tin tài nguyên của một thành viên của nhóm cũng như thông tin về các nhóm. Một nhóm là một tập các Web services và WS-Resources của một lĩnh vực cụ thể (domain specific).
- ✓ **WS-BaseFault**: định nghĩa các kiểu chuẩn của lỗi và cách sử dụng chúng trong WSDL.

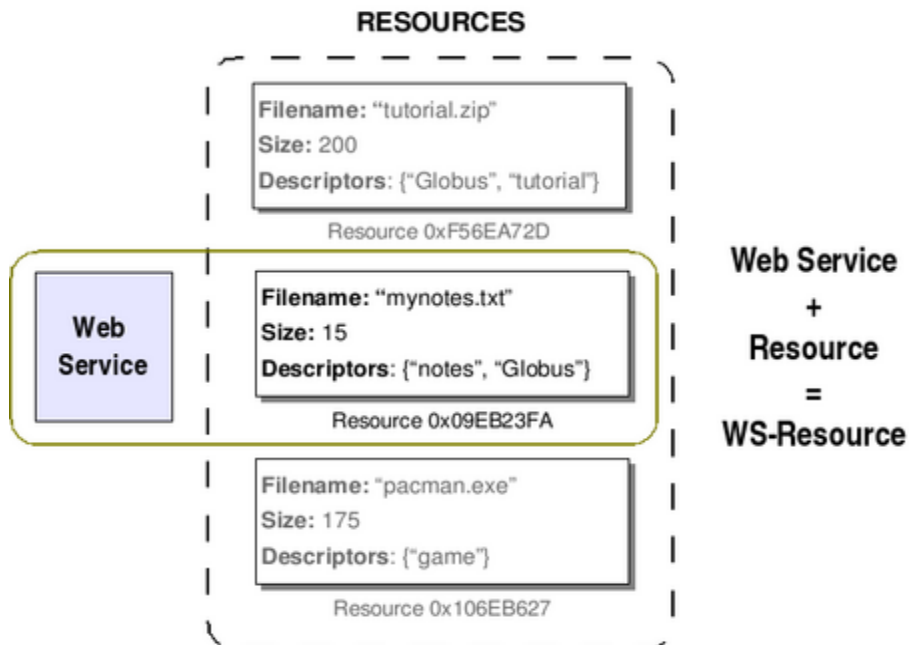
WSRF tách biệt giữa Web services (không trạng thái) và Resource (có trạng thái) và định ra một cơ chế cho việc kết hợp giữa hai thành phần này lại với nhau.

Hình 13 đưa ra một ví dụ về hai thành phần can bản nhất của WSRF: Web Service cung cấp một giao tiếp cho phép truy cập vào các tài nguyên (resources) có trạng thái (các file).



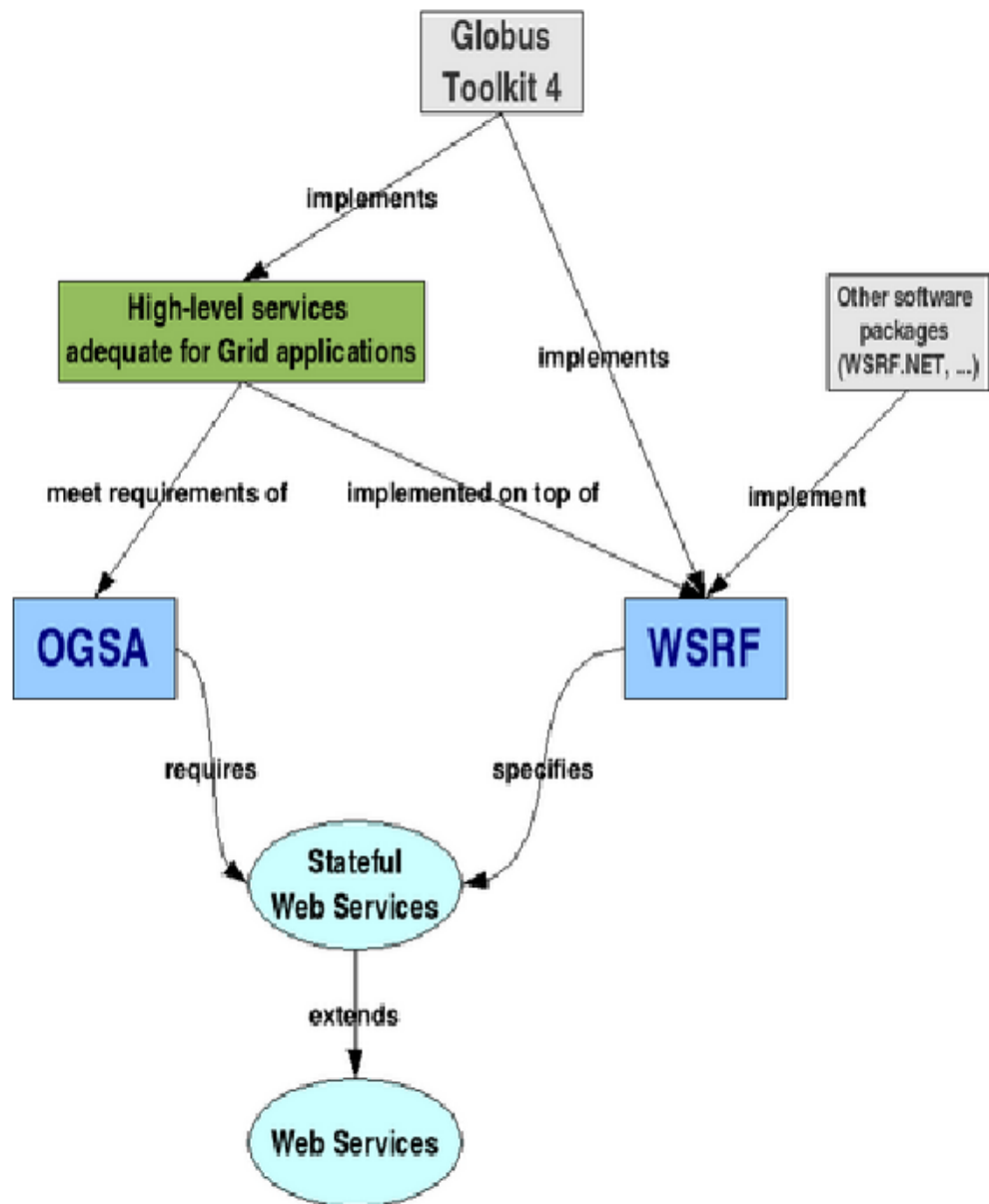
Hình 13 : Một Web service đại diện cho nhiều resources, mỗi resource là một file

WSRF sẽ quy định cách thức kết hợp để người dùng từ xa, thông qua giao tiếp Web service có thể truy cập vào tài nguyên yêu cầu. Việc kết hợp này được mô tả trong Hình.



Hình 14 : Kết hợp giữa Web service và Resources trong WSRF

2. Hiện thực WSRF :



Hình 15: Hiện thực WSRF

VII. Kết luận

Như vậy trong bài báo cáo này bọn em đã giới thiệu về SOA và cách Web Services được hiện thực như thế nào trong tính toán lưới.

Chúng ta thấy SOA ra đời để cung cấp cơ chế cho phép các hệ thống hoạt động trên các nền tảng khác nhau có thể giao tiếp với nhau. Qua đó các hệ thống khác nhau trong lưới có thể thấy nhau và làm việc được với nhau.

Ban đầu OGSA ra đời để đưa ra các chuẩn để cộng đồng tính toán lưới có thể áp dụng việc hiện thực các Grid Service. Nó ban đầu cũng lấy các tiêu chuẩn của Web Service bình thường vào. Nhưng sau đó thì lại phát triển thành đường riêng, vì vậy họ không còn lấy được những lợi ích từ các phiên bản nâng cấp sau này của Web Service.

Chính vì lí do đó WSRF ra đời để có thể vẫn dùng được điểm mạnh của Web Service khi họ phát triển, và vẫn có thể áp dụng được vào Grid Computing.

Globus đã hiện thực các ý tưởng của SOA, OGSA, WSRF trong Globus toolkit.

Globus toolkit là một bộ toolkit mã nguồn mở được sử dụng để xây dựng các hệ thống và các ứng dụng Grid. Globus toolkit hiện đang được phát triển bởi tổ chức Globus Alliance và nhiều tổ chức khắp nơi trên thế giới. Globus Alliance là một cộng đồng các cá nhân và tổ chức tham gia phát triển các công nghệ nền tảng cho Grid.

Mặc dù tính toán lưới đã không còn phổ biến nhưng những ý nghĩa của SOA hay OGSA, WSRF vẫn giúp cộng đồng công nghệ có thêm ý tưởng và lựa chọn cho các giải pháp của họ.

Tài liệu tham khảo

- An Overview of Service-oriented Architecture, Web Services and Grid Computing

- Web Services Implementation Methodology for SOA Application Elements of Service-Oriented Architecture - B. Ramamurthy

- wikipedia.org

- www.w3.org

- www.webservices.org

- IBM: <http://www.ibm.com/developerworks/library/ws->

esource/Globus Java WSRF Core 3.9.1

- <http://www.globus.org/toolkit/>