

CÁC CÔNG NGHỆ LẬP TRÌNH HIỆN ĐẠI



Khoa Công nghệ thông tin
Bộ môn Công nghệ phần mềm

RESTful Web Service

Nguyễn Huy Khánh
nhkhanh@fit.hcmus.edu.vn

Mục tiêu



- Hiểu được cách thức hoạt động của WWW trên HTTP
- Hiểu được nguồn gốc và ý nghĩa của REST web service
- Ứng dụng tạo REST web service bằng Visual Studio



Giao thức HTTP

Kiến trúc REST

RESTful Web service

ASP.NET Web API



Giao thức HTTP

Kiến trúc REST

RESTful Web service

ASP.NET Web API

Định nghĩa



Sir Tim Berners-Lee
Nhà phát minh WWW

● WWW

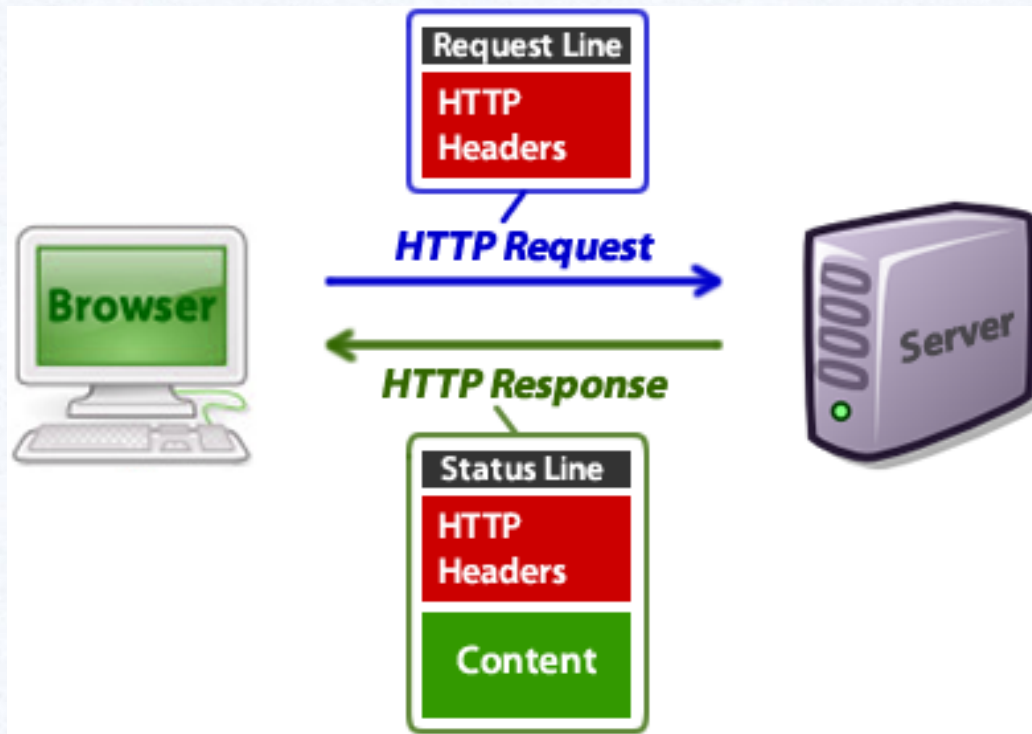
- World wide web
- Là hệ thống các tài liệu liên kết với nhau thông qua internet

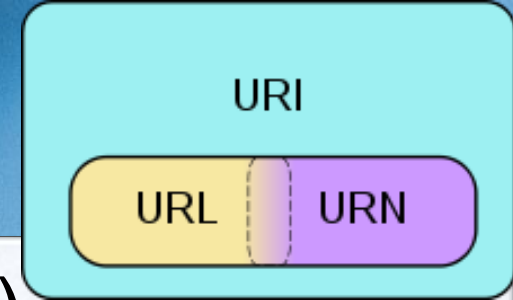
● HTTP

- Hypertext Transfer Protocol
- Là giao thức mạng cho phép các hệ thống thông tin phân phối, và cộng tác với nhau. HTTP là nền tảng giao tiếp dữ liệu cho WWW
(1990)

HTTP

- HTTP hoạt động trên cơ chế giao thức request – response trong mô hình điện toán client – server





- URI (Uniform Resource Identifier)
 - Là một chuỗi để xác định một tài nguyên trên internet
- URL (Uniform Resource Locator)
 - Là một URI cho biết sự tồn tại của một tài nguyên và cách thức để nhận tài nguyên đó
- URN (Uniform Resource Name)
 - Là một URI nhằm xác định tài nguyên bằng tên và độc lập với vị trí lưu trữ

HTTP Status code

- 2xx Success
 - 200 – OK
 - 201 - Created
- 3xx Redirection
 - 304 – Not Modified
- 4xx Client Error
 - 403 Forbidden
 - 404 – Not Found
- 5xx Server Error
 - 503 – Service Unavailable
 - 504 – Gateway Timeout

<http://www.w3.org/Protocols/rfc2616/rfc2616-sec10.html>

Ví dụ về HTTP - Request

GET <http://www.fit.hcmus.edu.vn/vn/> HTTP/1.1

Accept: application/x-ms-application, image/jpeg, application/xaml+xml, image/gif, image/pjpeg, application/x-ms-xbap, application/x-shockwave-flash, application/vnd.ms-excel, application/vnd.ms-powerpoint, application/msword, */*

Accept-Language: vi-VN

User-Agent: Mozilla/4.0 (compatible; MSIE 8.0; Windows NT 6.1; WOW64; Trident/4.0; GTB6.5; Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; SV1) ; SLCC2; .NET CLR 2.0.50727; .NET CLR 3.5.30729; .NET CLR 3.0.30729; Media Center PC 6.0; .NET4.0C; .NET4.0E; OfficeLiveConnector.1.5; OfficeLivePatch.1.3; InfoPath.3; AskTbGGSV5/5.8.0.12217) chrome/6.0.472.63

Accept-Encoding: gzip, deflate

Connection: Keep-Alive

Host: www.fit.hcmus.edu.vn

Cookie:

.ASPXANONYMOUS=MG6LCiuSywEkAAAANTA2ZWNiYTAAtYThiNy00MDA1LTkyNjUtYTIIYzAxNTA3MTU10

Ví dụ về HTTP - Response

HTTP/1.0 200 OK

Cache-Control: private

Content-Type: text/html; charset=utf-8

Server: Microsoft-IIS/7.0

X-AspNet-Version: 2.0.50727

Set-Cookie: DotNetNukeAnonymous=7db3c001-c407-4adb-a60f-053b5dc76dc2;
expires=Thu, 30-Sep-2010 03:12:33 GMT; path=/; HttpOnly

Set-Cookie: language=vi-VN; path=/; HttpOnly

X-Powered-By: ASP.NET

Date: Thu, 30 Sep 2010 02:52:33 GMT

Content-Length: 24404

X-Cache: MISS from vweb.hcmuns.edu.vn

Via: 1.0 vweb.hcmuns.edu.vn:80 (squid/2.6.STABLE16)

Connection: keep-alive

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">

...

// Nội dung trang web

HTTP Request method

- HTTP định nghĩa chín phương thức chính:

HEAD	GET	POST
PUT	DELETE	TRACE
OPTIONS	CONNECT	PATCH

Safe và Idempotent

	Safe	Idempotent
GET	✓	✓
POST		
PUT		✓
DELETE		✓

HTTP Method – GET, HEAD, DELETE

● GET

- Dùng để tải một biểu diễn của tài nguyên
- Được sử dụng nhiều nhất
- Biểu diễn của tài nguyên có thể bao gồm HTML, JPG, XML, ...

● HEAD

- Tương tự như GET thay vì tải toàn bộ thể hiện thì chỉ tải HTTP Header
- Dùng để kiểm tra thay đổi khi muốn tải lại tài nguyên có kích thước lớn

● DELETE

- Dùng để xóa tài nguyên

HTTP Method - PUT

- Dùng để lưu một biểu diễn vào một tài nguyên
- Đăng tải một tập tin vào một vị trí xác định trên website
- Nếu PUT thực hiện hai lần thì lần sau sẽ đè lên lần trước
- Ví dụ: Đăng hình đại diện cho profile cá nhân

HTTP Method - POST

- Tạo một tài nguyên tương tự như PUT, nhưng server sẽ quyết định cách lưu trữ thay vì client như PUT
- Ví dụ: Tạo một bài viết trên blog
- Đăng tải một tập tin vào một vị trí xác định trên website

Nội dung



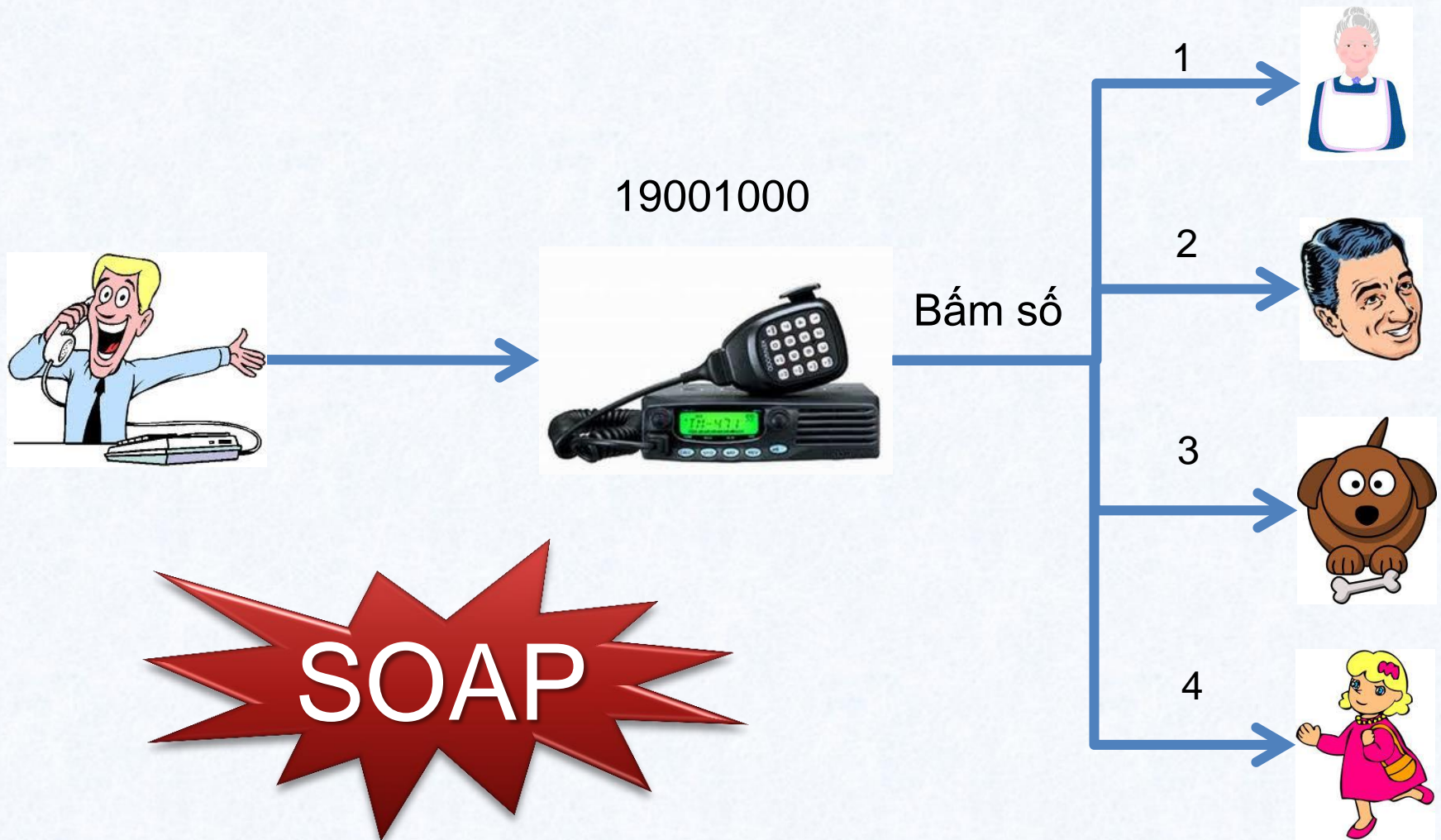
Giao thức HTTP

Kiến trúc REST

RESTful Web service

ASP.NET Web API

Ví dụ

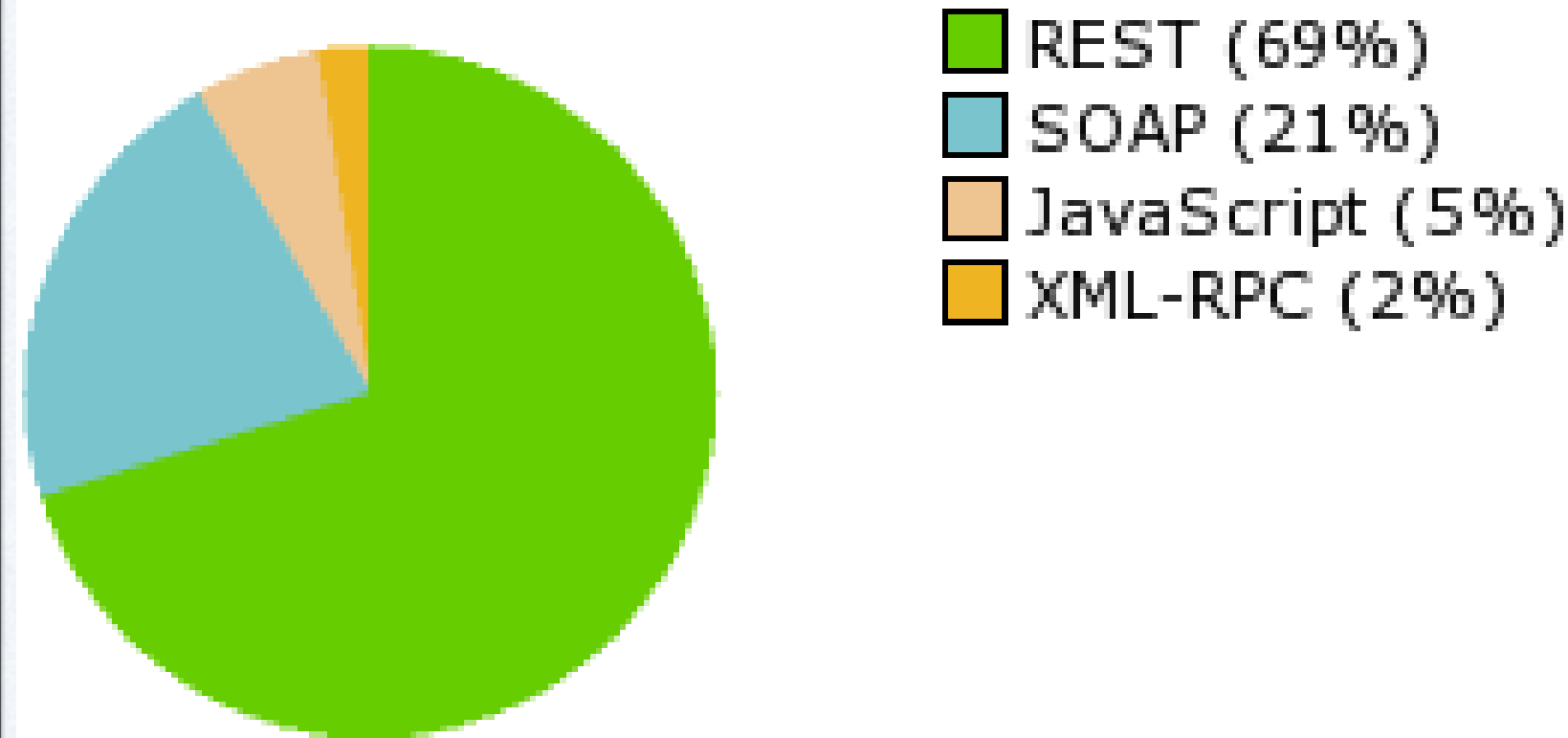


Ví dụ



Tỉ lệ các giao thức của Web API

Protocol Usage by APIs



ProgrammableWeb.com 10/05/12

Định nghĩa REST



Roy Fielding
Một trong các tác giả của giao thức HTTP

- REST (Representational State Transfer) là một kiểu kiến trúc phần mềm cho các hệ thống phân tán siêu truyền thông như là WWW

(2000)

Ý nghĩa của REST

Representational State Transfer

Vào trang <http://courses.fit.hcmus.edu.vn/ccnlthd.html>



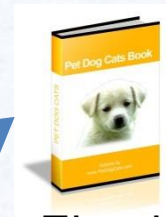
Người dùng



Trình duyệt



CCNLTHD



Ebook



Demo



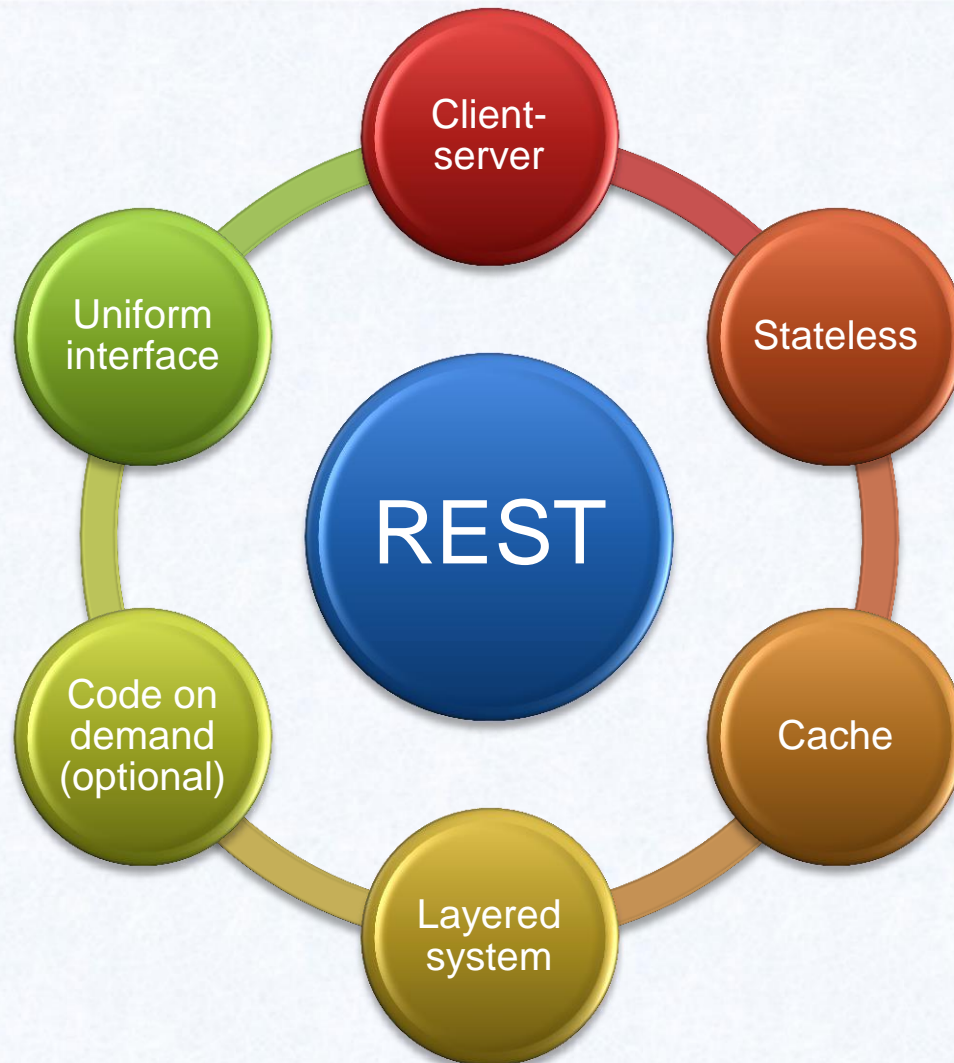
Diễn đàn

Bấm vào diễn đàn

Ý nghĩa của REST

- Web là một tập hợp các tài nguyên
- Ví dụ trang web môn học có nhiều môn học trong đó có môn CNLTHD (tài nguyên)
- <http://courses.fit.hcmus.edu.vn/ccnlthd.html>
- Khi người dùng vào liên kết trên sẽ nhận một biểu diễn (**representation**) của tài nguyên (ccnlthd.html)
- Biểu diễn này đặt trình duyệt vào một trạng thái (**state**)
- Biểu diễn này bao gồm nhiều liên kết đến các tài nguyên khác (ebook, slide, diễn đàn, ...)
- Khi người dùng kích vào một trong các liên kết này
- Người dùng nhận được biểu diễn tương ứng của tài nguyên
- Trình duyệt lại ở một trạng thái khác
- Như vậy trình duyệt đã chuyển đổi (**transfer**) trạng thái ứng với mỗi tài nguyên

Ràng buộc của kiến trúc REST



Client-Server

- Phân tách giao diện của client ra khỏi dữ liệu
- Cho phép mỗi thành phần phát triển độc lập
- Hỗ trợ đa nền tảng



Figure 5-2. Client-Server

Stateless

- Mỗi yêu cầu từ client phải có đủ thông tin cần thiết để server có thể hiểu được mà không cần phải lưu trữ thêm thông tin nào trước đó.

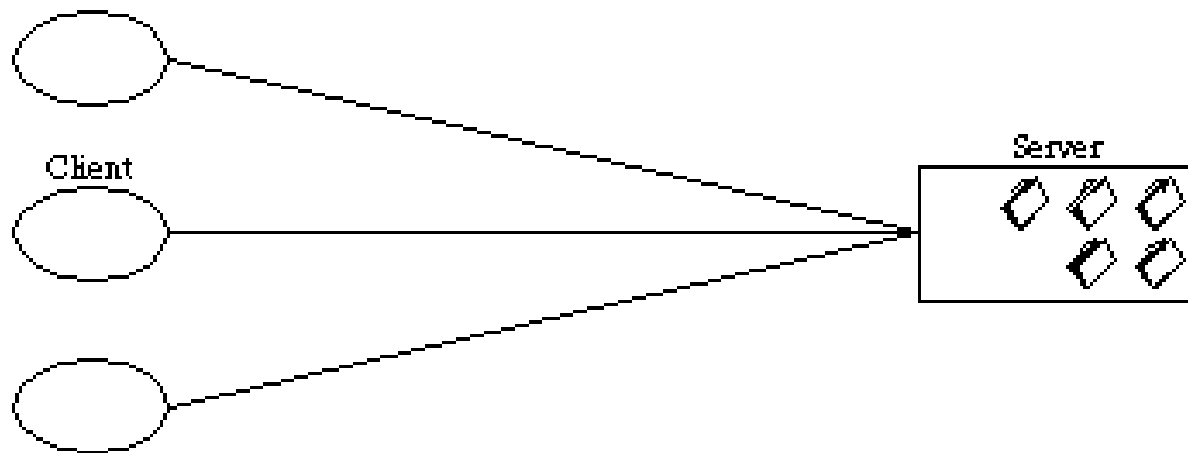


Figure 5-3. Client-Stateless-Server

Cache

- Dữ liệu phản hồi từ yêu cầu được gán nhãn rõ ràng hoặc ngầm định rằng có thể được dùng lại hoặc không được dùng lại

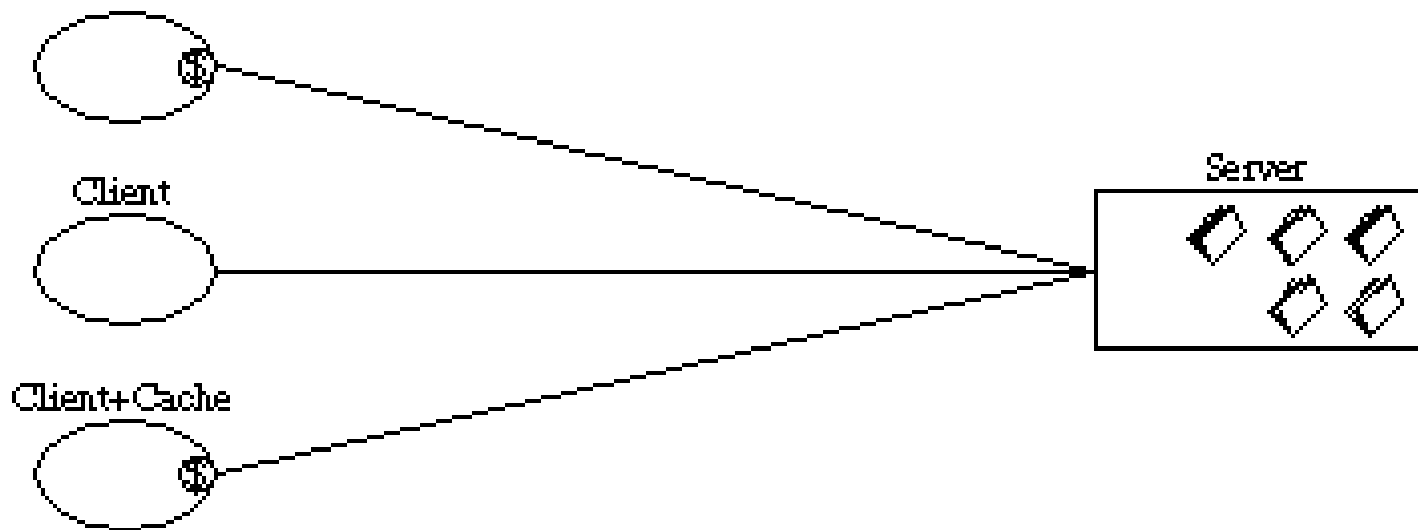


Figure S-4. Client-Cache-Stateless-Server

Uniform interface

- Tất cả tài nguyên được truy cập thông qua một interface thống nhất (HTTP GET, PUT, POST, DELETE, ...)

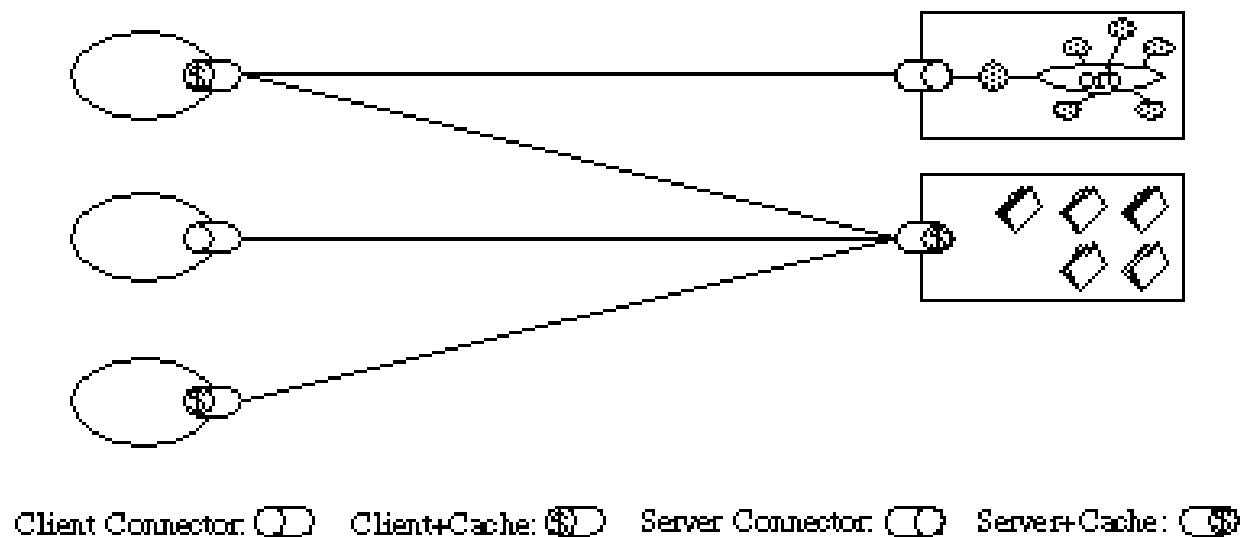


Figure S-6. Uniform-Client-Cache-Stateless-Server

Layered system

- Cho phép thêm các lớp (proxy server, cache server, gateway, ...) giữa các thành phần trên hệ thống mà các thành phần đó không biết được đang tương tác qua những lớp trung gian nào

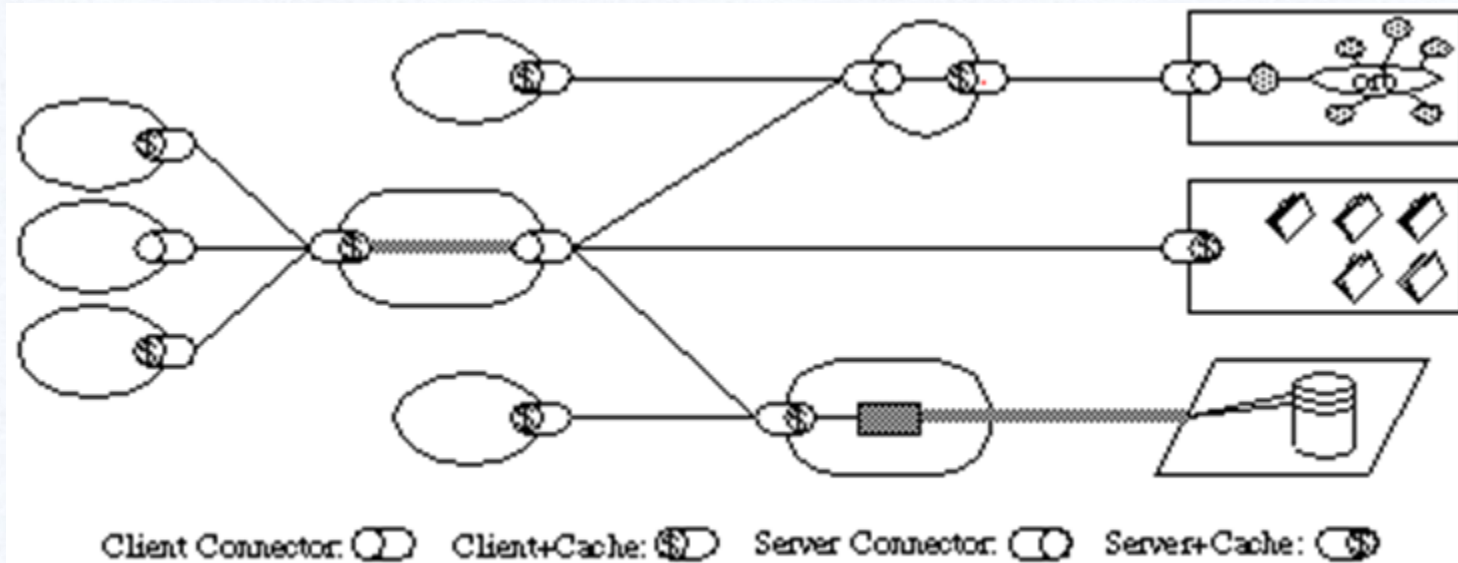
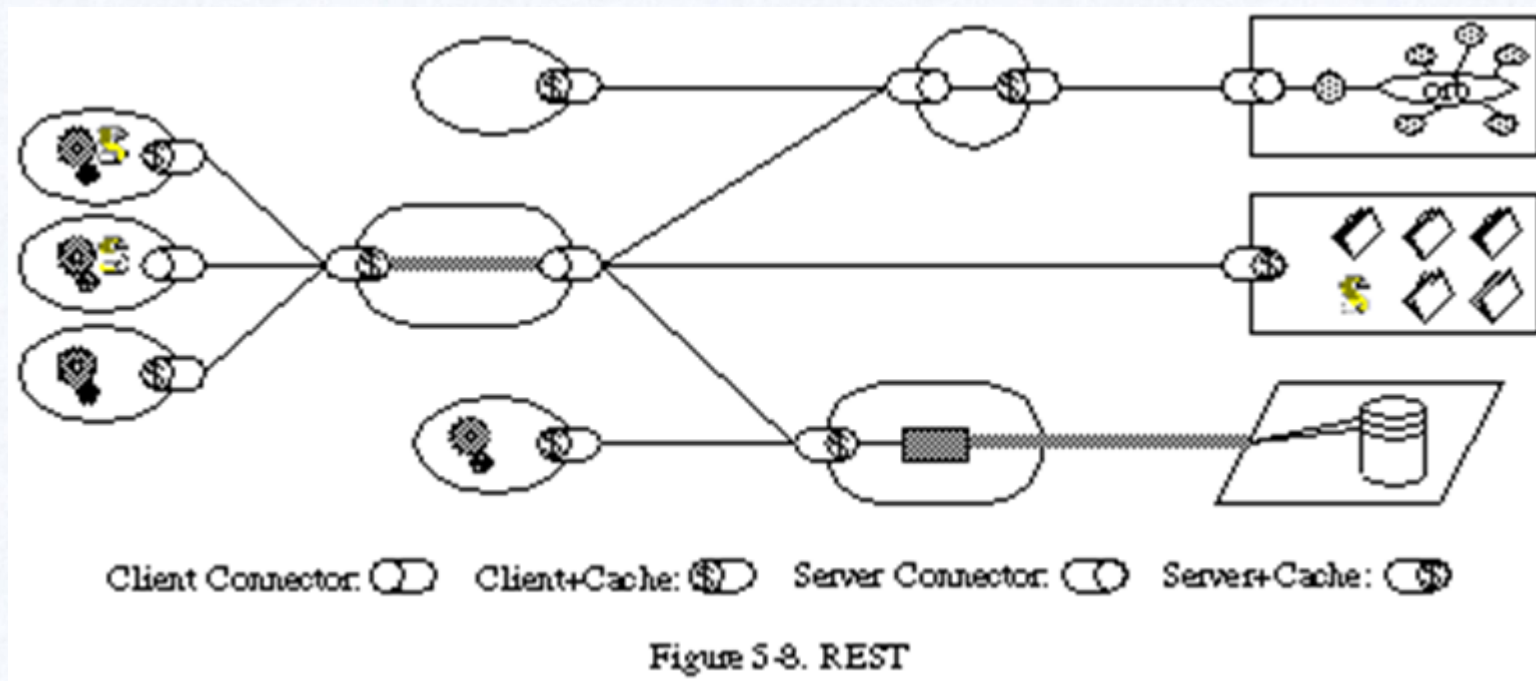


Figure 5-7. Uniform-Layered-Client-Cache-Stateless-Server

Code-On-Demand

- Cho phép client mở rộng chức năng bằng cách tải mã nguồn theo dạng applet hoặc script



Nội dung



Giao thức HTTP

Kiến trúc REST

RESTful Web service

ASP.NET Web API

RESTful Web service

- Là một web service đơn giản sử dụng HTTP và tính chất của REST.
- Là một tập tài nguyên các thành phần được định nghĩa:
 - URI gốc cho web service
 - MIME type hỗ trợ bởi web service
 - Tập hành động hỗ trợ bởi web service sử dụng HTTP method (GET, POST, PUT, DELETE)

Các hành động của RESTful WS

HTTP Method	GET	PUT	POST	DELETE
CRUD	Read	Update	Create	Delete
<p>URI tập hợp http://hcmus.edu/CCNLTHD</p>	<p>Liệt kê URI và thông tin chi tiết của các thành phần</p>	<p>Thay thế toàn bộ tập hợp bằng tập hợp khác</p>	<p>Tạo một mẫu tin mới trong tập hợp. URL của mẫu tin được tự động tạo và thường được trả về cho người tạo</p>	<p>Xóa toàn bộ tập hợp</p>
<p>URI phần tử http://hcmus.edu/CNNLTHD/23</p>	<p>Nhận biểu diễn của một phần tử theo một dạng định dạng trên Internet</p>	<p>Cập nhật phần tử trong tập hợp. Nếu chưa có thì thêm vào</p>	<p>Phần tử đang xét phải có chứa một tập hợp và tiến hành thêm một phần tử vào tập hợp đó</p>	<p>Xóa phần tử đang xét</p>

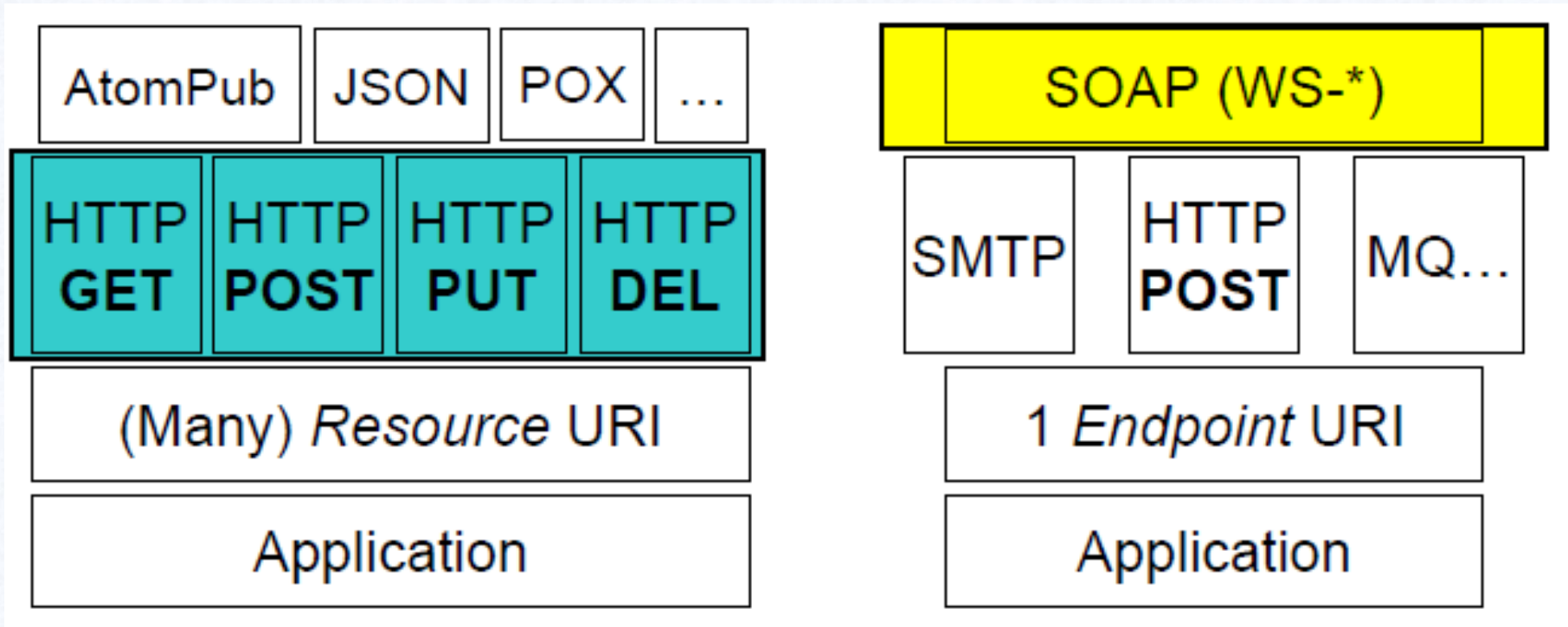
So sánh URL của SOAP và REST

Hành động	SOAP	REST
Xem học sinh	www.mg.com/service.asmx?op=GetStudent	GET www.mg.com/service/student/081233
Thêm học sinh	www.mg.com/service.asmx?op=InsertStudent	POST www.mg.com/service/student/class
Xóa học sinh	www.mg.com/service.asmx?op=DeleteStudent	DELETE www.mg.com/service/student/081233
Sửa học sinh	www.mg.com/service.asmx?op=UpdateStudent	PUT www.mg.com/service/student/081233
Danh sách lớp	www.mg.com/service.asmx?op=GetClass	GET www.mg.com/service/student/class

So sánh REST và SOAP

REST	SOAP
Là một kiến trúc phần mềm	Là một giao thức
Sử dụng các HTTP method làm hành động (GET, POST, PUT, DELETE)	Tự định nghĩa các hành động và chủ yếu sử dụng HTTP POST
Sử dụng nhiều URI endpoint	Sử dụng một URI endpoint
Dữ liệu dựa trên tài liệu con người dễ đọc được như XML, JSON, hình ảnh	Strong typing và bọc trong XML
Bảo mật dựa trên HTTP	Tự xây dựng các cách thức bảo mật

So sánh REST và SOAP



Nội dung



Giao thức HTTP

Kiến trúc REST

RESTful Web service

ASP.NET Web API

Cài đặt ASP.NET MVC

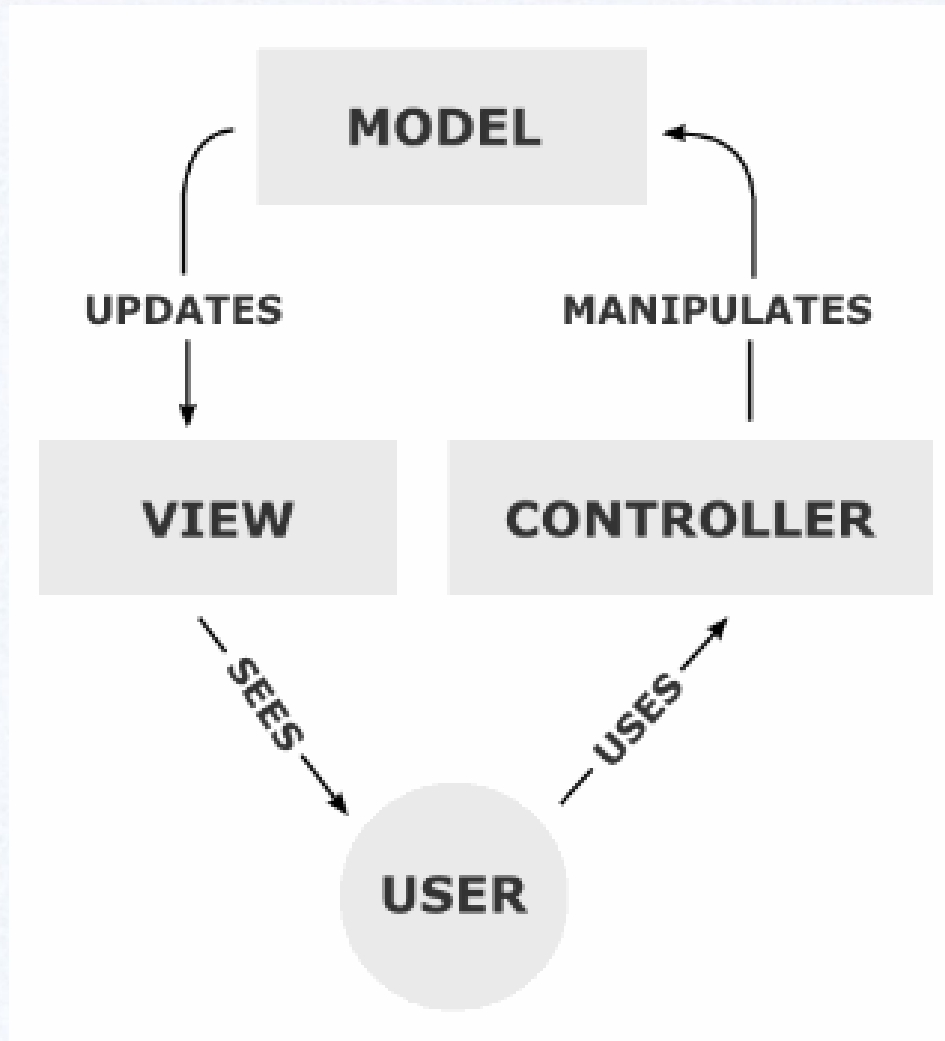
- Microsoft Visual Studio 2012
 - ASP.NET MVC 4



- Microsoft Visual Studio 2010
 - Web Platform Installer
 - ASP.NET MVC 4 (Visual Studio 10)

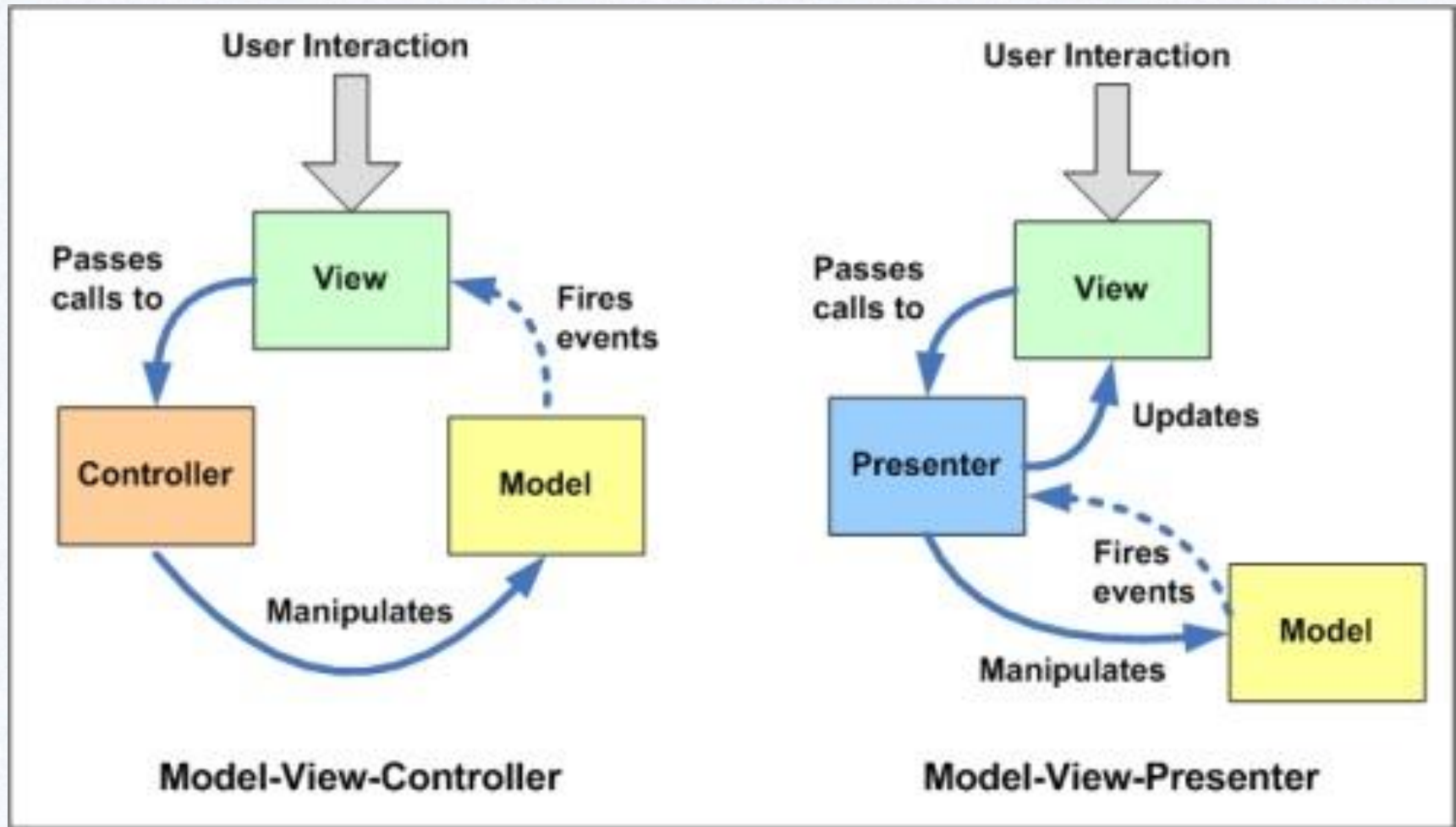


Kiến trúc MVC



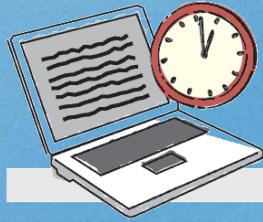
<http://en.wikipedia.org/wiki/Model%E2%80%93view%E2%80%93controller>

MVP với MVC



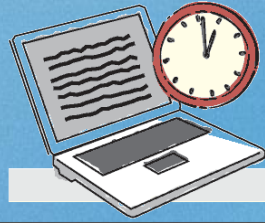
<http://joel.inpointform.net/software-development/mvvm-vs-mvp-vs-mvc-the-differences-explained/>

Bài tập cá nhân



- Xây dựng ứng dụng quản lý sinh viên cho phép lưu trữ thông tin sinh viên, bao gồm:
 - Họ tên
 - Điểm tích lũy
 - Hình đại diện
- Ứng dụng bao gồm các chức năng sau:
 - Liệt kê danh sách sinh viên
 - Thêm sinh viên
 - Xóa sinh viên
 - Sửa sinh viên
- Kiến trúc ứng dụng bao gồm hai thành phần:
 - Kết nối CSDL và cung cấp Restful Web service
 - Giao diện gọi các Web service ở thành phần trên để thực hiện các chức năng của người dùng.

Thang điểm

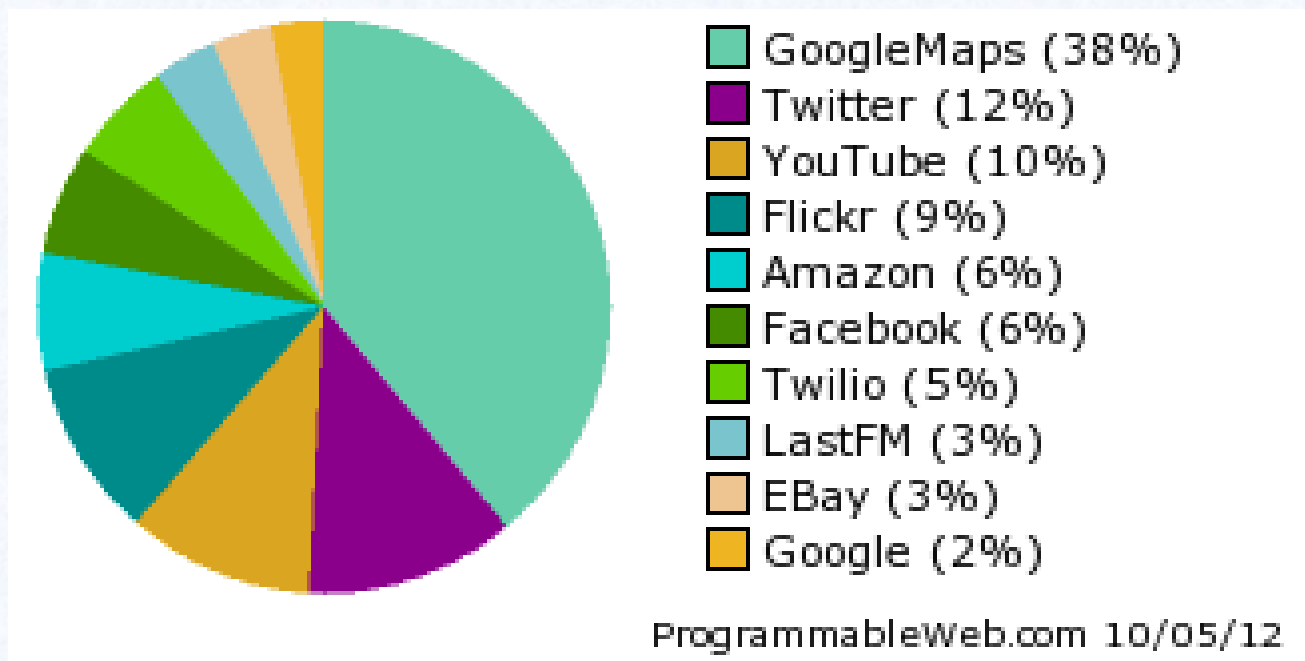


- Chức năng liệt kê: 1.5đ
- Chức năng thêm: 1.5đ
- Chức năng xóa: 1.5đ
- Chức năng sửa: 1.5đ
- Hiển thị hình đại diện: 2đ
- Đăng tải hình đại diện: 2đ

Đồ án tìm hiểu



- Chọn một Web API thông dụng
 - Tìm hiểu các chức năng mà Web API hỗ trợ
 - Viết các ứng dụng nhỏ minh họa



<http://www.programmableweb.com/apis/directory/1?sort=mashups>

Tham khảo thêm



- <http://www.programmableweb.com/>
- <http://www.asp.net/web-api>
- <http://www.west-wind.com/weblog/posts/2012/Aug/21/An-Introduction-to-ASPNET-Web-API>