Week 6 – Web application programming

Web Basics

Web Server Related Technologies

Hypertext Transfer Protocol (HTTP)

Simple Object Access Protocol (SOAP)

Cascading Style Sheets (CSS)

Javascript (JSP)

Asynchronous JavaScript and XML (AJAX)

HyperText Markup Language (HTML)

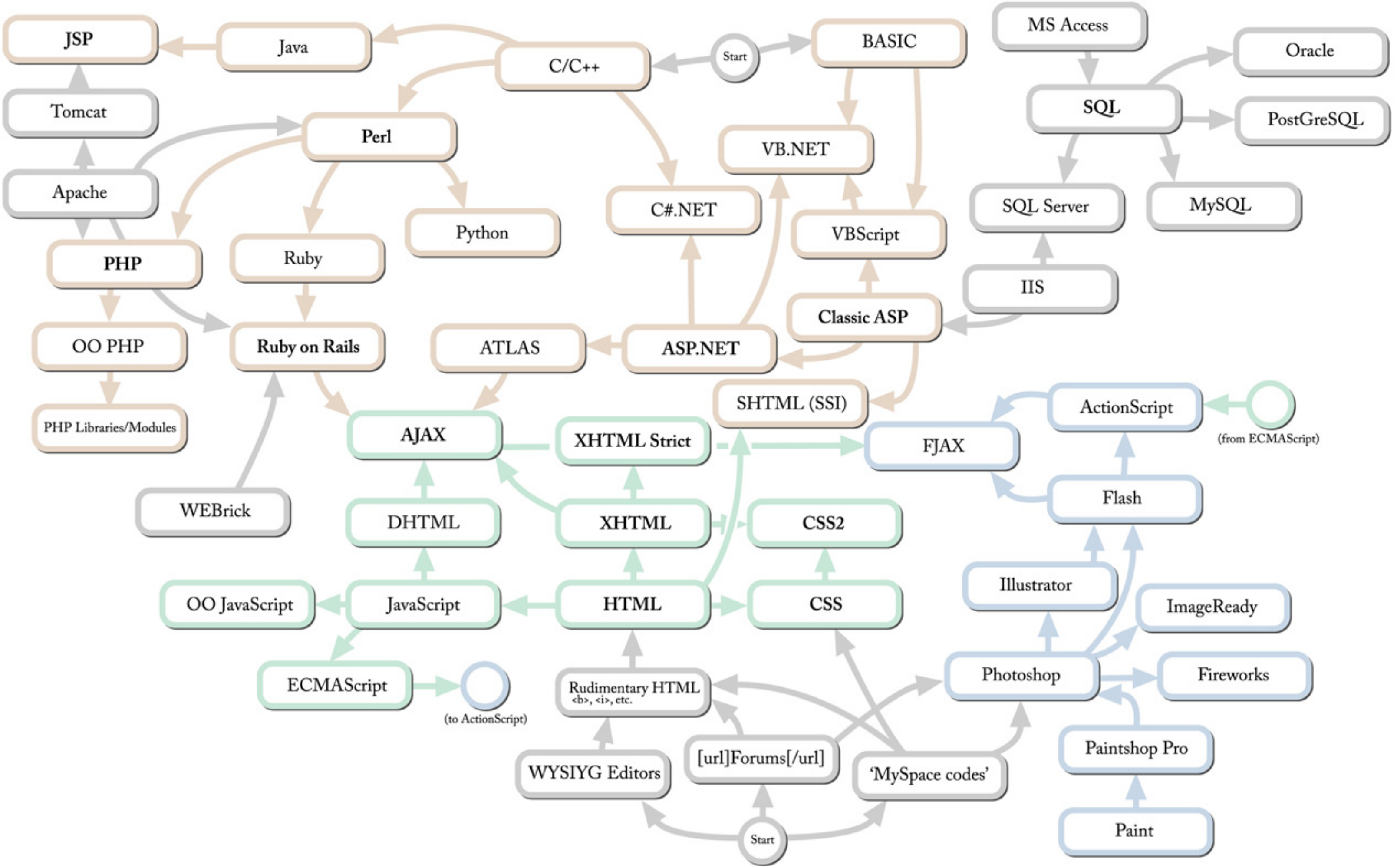
Dynamic HTML (DHTML)

Extensible Markup Language (XML)

Security Assertion Markup Language (SAML)

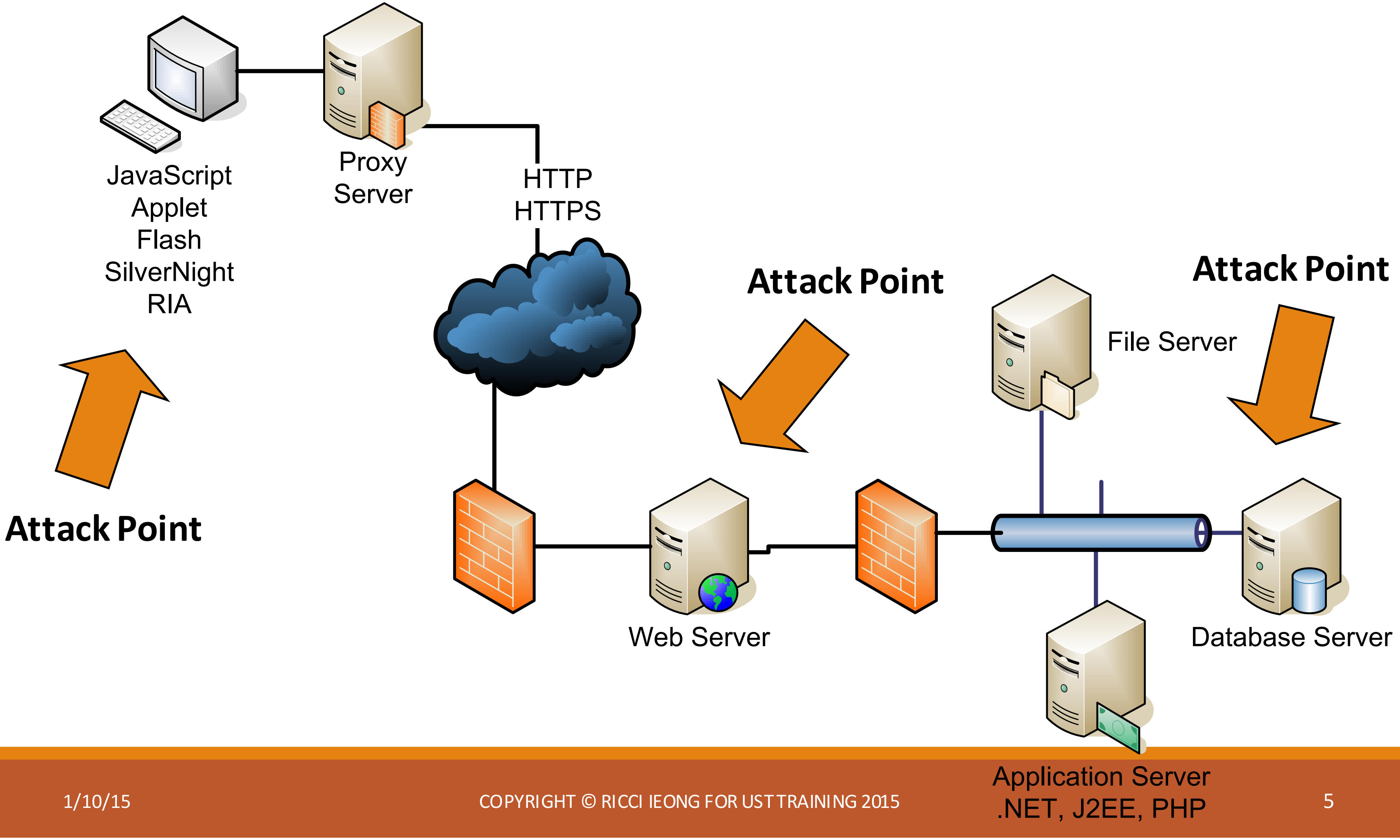
Service Provisioning Markup Language (SPML)

Web Server Related Technologies



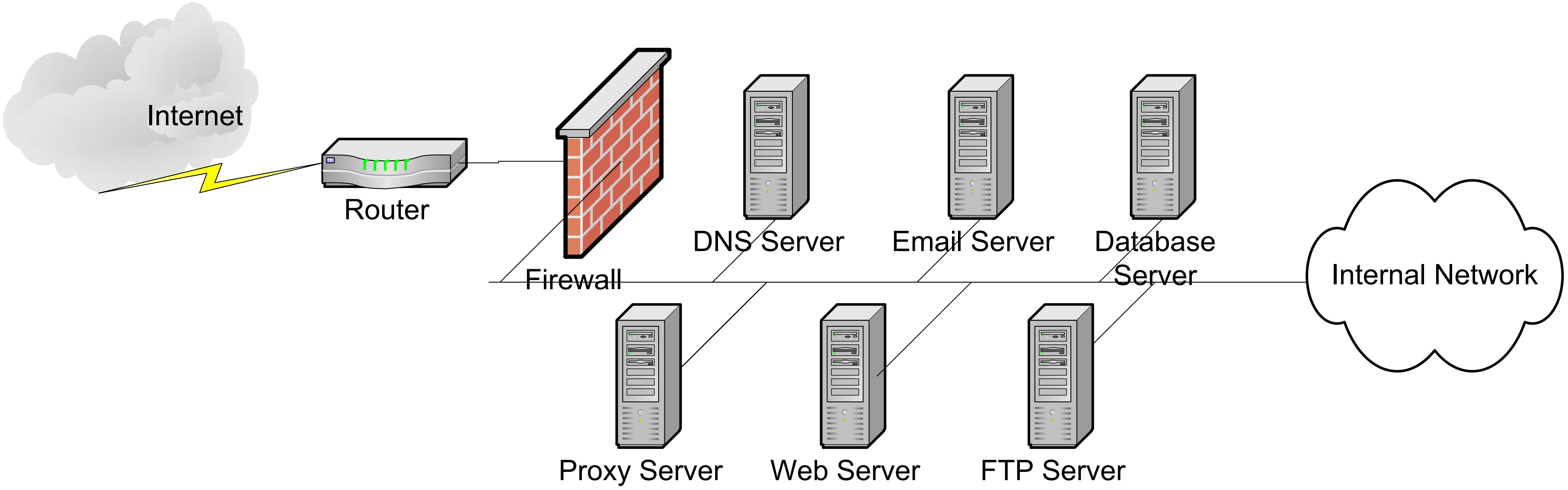
http://modernl.com/article/web-tech-family-tree

Typical Web Architecture



Web Architecture

Web Architecture



Components in a Web application

Components include

- Operating Systems
- Web Server

Other Supporting servers

- Database Server
- DNS Server
- SMTP Server
- FTP Server

What Does a Web Server Do?

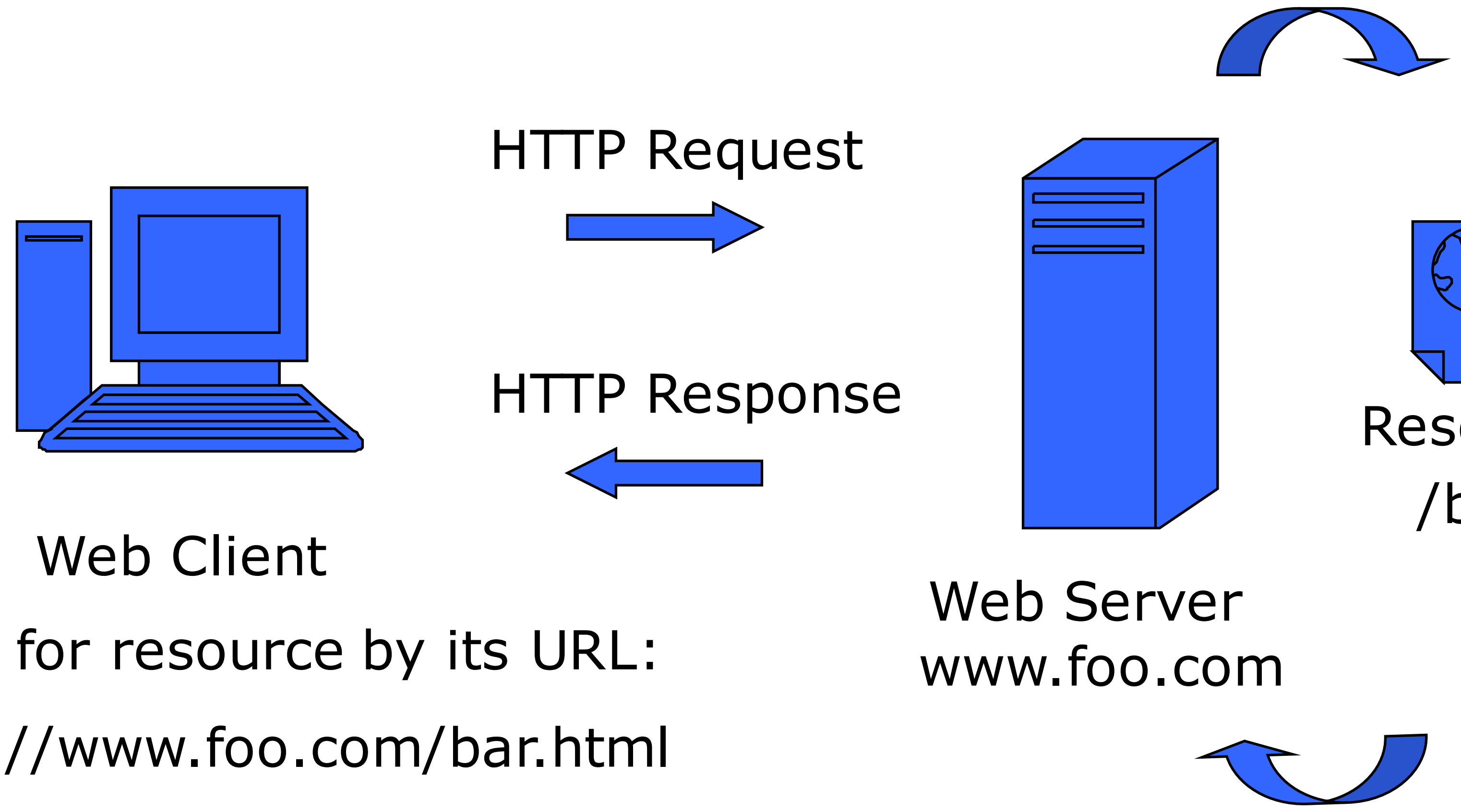
Communicate with client browser

Display HTML into formatted pages

Display pictures and programs into animated pictures/programs

Accept input from client browser

Basic Web Server response cycle



HTTP Request

HTTP Response

Web Client

Web Server
www.foo.com

Resource
/bar

Asks for resource by its URL:

http://www.foo.com/bar.html

What Does a Proxy Server Do?

Between client and server

Receives the client request

Decides if request will go on to the server

May have cache & may respond from cache

Acts as the client with respect to the server

Uses one of it’s own IP addresses to get page from server

Simple Proxy

Harvest/Squid

Provide web content for a fixed user base

Standalone operation

May be transparent

Commodity product/technology

Easy to get 90% correct

Transparent Proxy

No client configuration

Violates end-to-end paradigm

- Client thinks it is talking directly to server
- Server thinks it is talking to cache

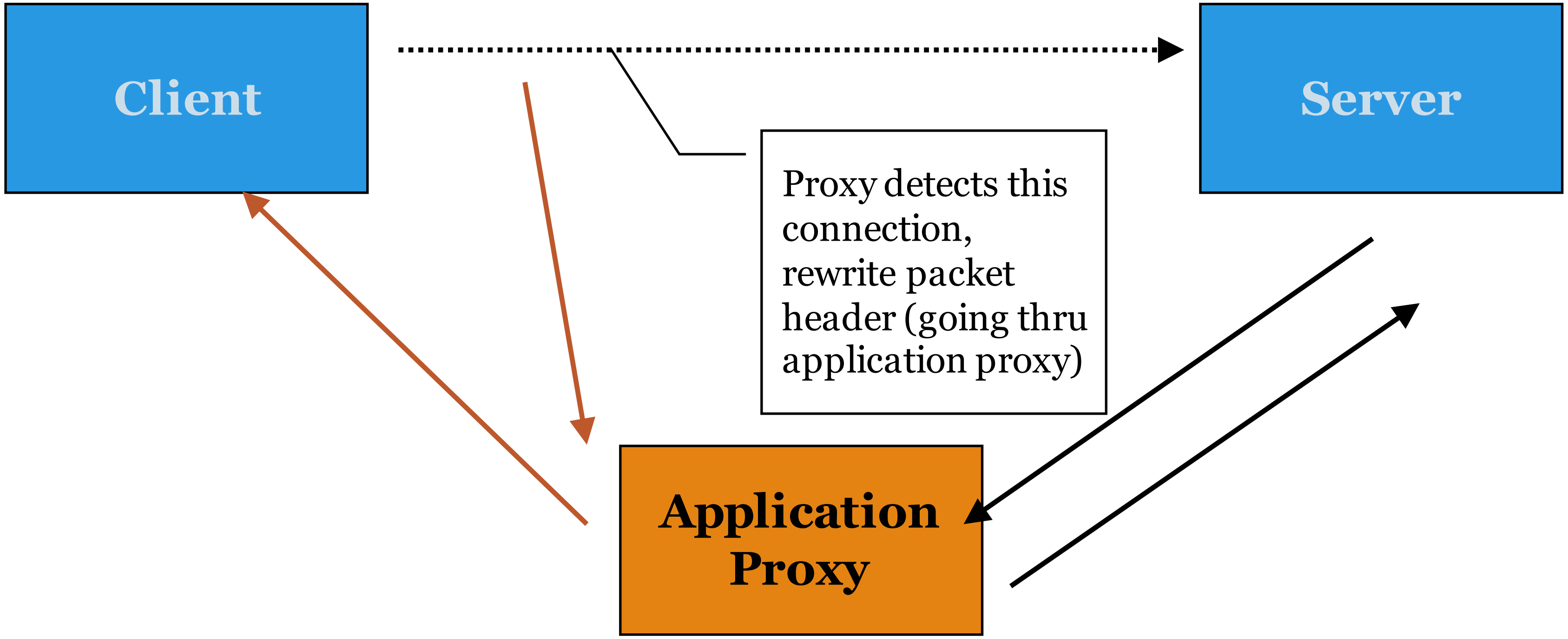
Implemented as

- Pass-through unit
- L4 switch

Firewalls – Transparent Proxy

Basically, application proxy with network address translation (NAT)

- don’t require client configuration



Proxy detects this connection, rewrite packet header (going thru application proxy)

Reverse Proxy (Surrogate)

Designed to offload duties from one or more specific servers

Data size is limited to size of static content on the server

Challenge is fast, disk-less operation

Cache consistency is easy

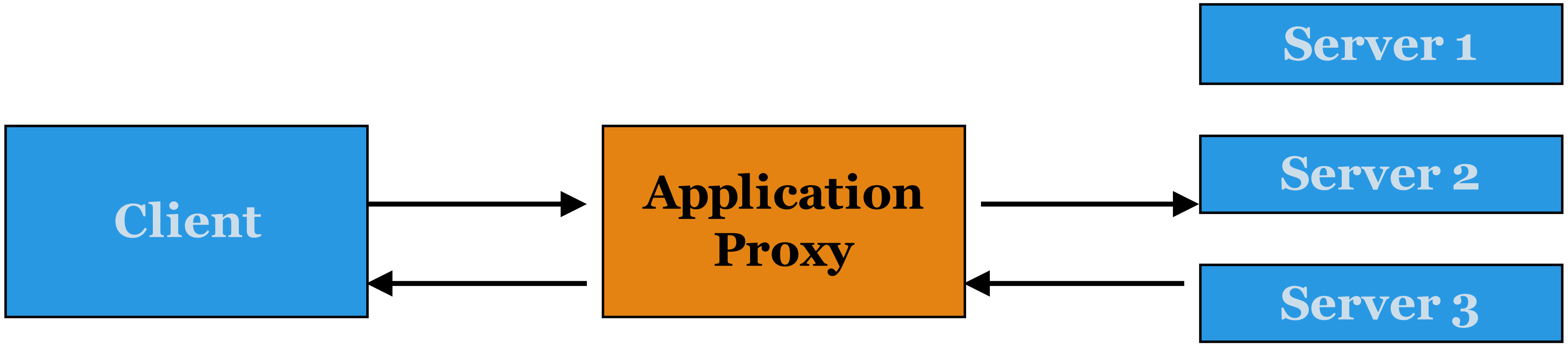
Single point of failure

Reverse Proxy

Acts as a broker between two entities, validating and processing a transaction in such a way that the actual parties to the transaction do not directly communicate with one another

- How is it different from application proxy?

Benefits/drawbacks is similar to application proxy



Main Proxy Functions

Caching

Firewall

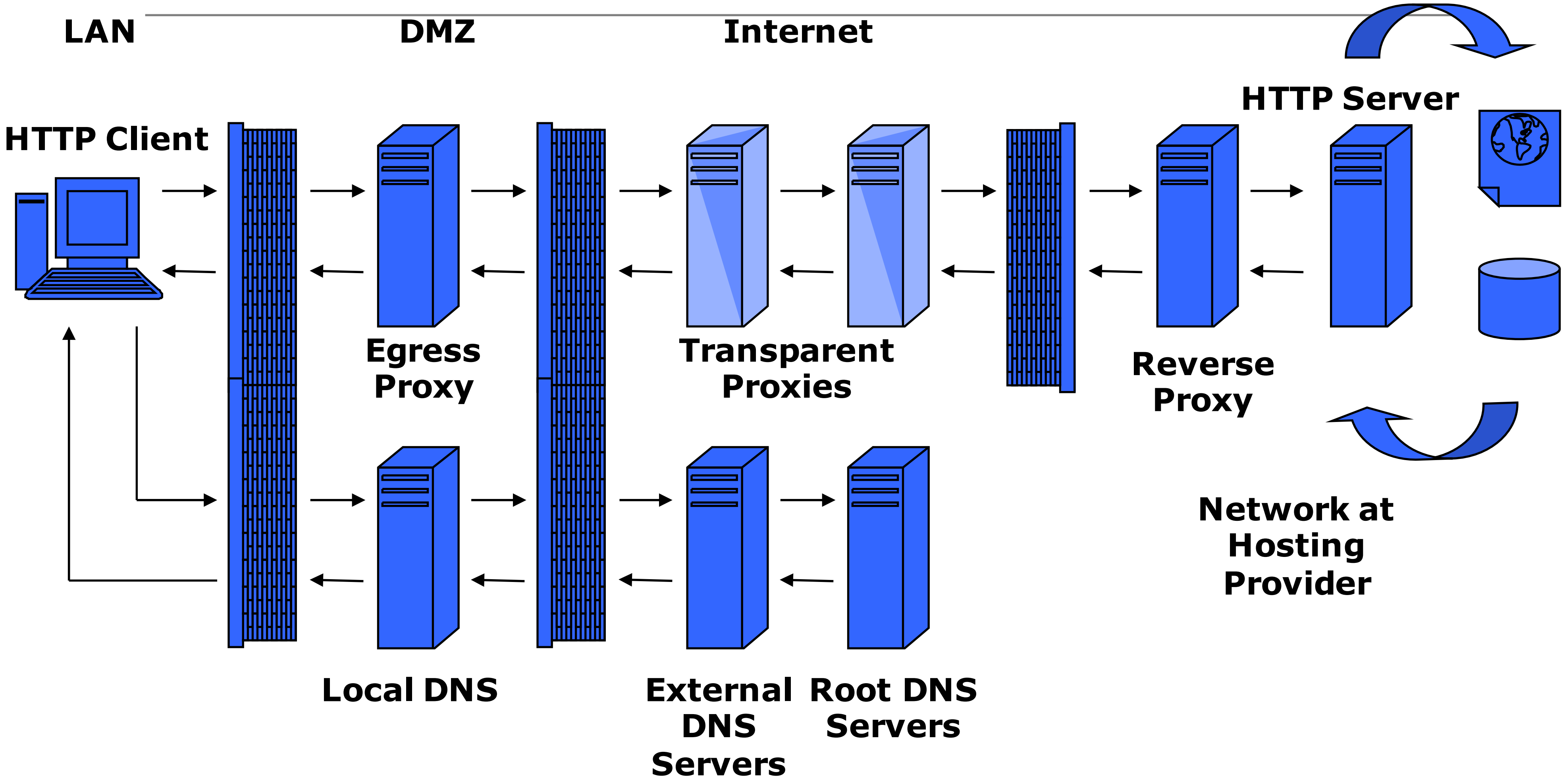
Access control

Anonymizer

Filtering

Logging

An HTTP Request/Response Chain



Determine the path of the proxy access

Use Trace to determine the path

Trace /index.html HTTP/1.0

Web protocol and Web application

Uniform Resource Identifier vs Uniform Resource Locator

URI

- Syntax
  <scheme name> : <hierarchical part> [? <query>] [# <fragment>]

URL syntax

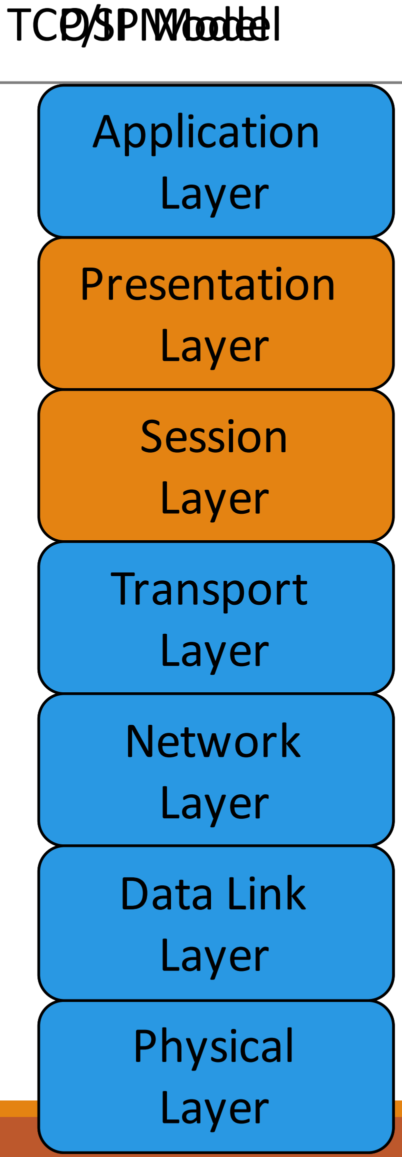
- A subset of URI
- Syntax
  scheme://username:password@domain:port/path?query_string#anchor

TCP/IP Basics

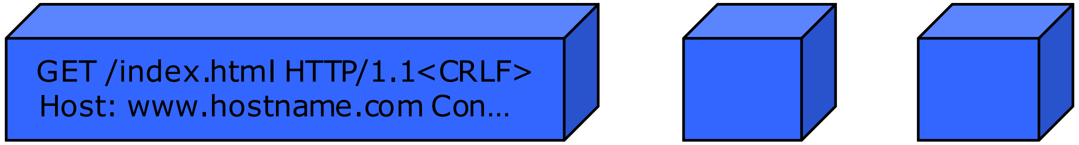
History of TCP/IP

5 layers

- Application layer
- Transport layer
- Network layer
- Data Link layer
- Physical layer

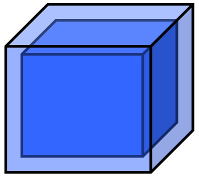


How an HTTP Message is delivered over TCP/IP connection



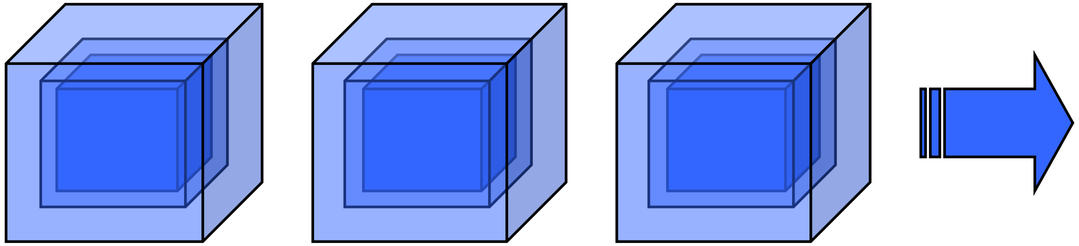
```
GET /index.html HTTP/1.1<CRLF>
Host: www.hostname.com Con…
```

HTTP Message's data stream is chopped up into chunks small enough to fit in a TCP segment



The chunks ride inside TCP segments used to reassemble them correctly on the other end of the connection

The segments are shipped to the right destination inside IP datagrams



An Introduction to HTTP

Hyper Text Transfer Protocol

One of the *application layer protocols* that make up the Internet

- HTTP over TCP/IP
- Like SMTP, POP, IMAP, NNTP, FTP, etc.

The underlying language of the Web

Three versions have been used, two are in common use and have been specified

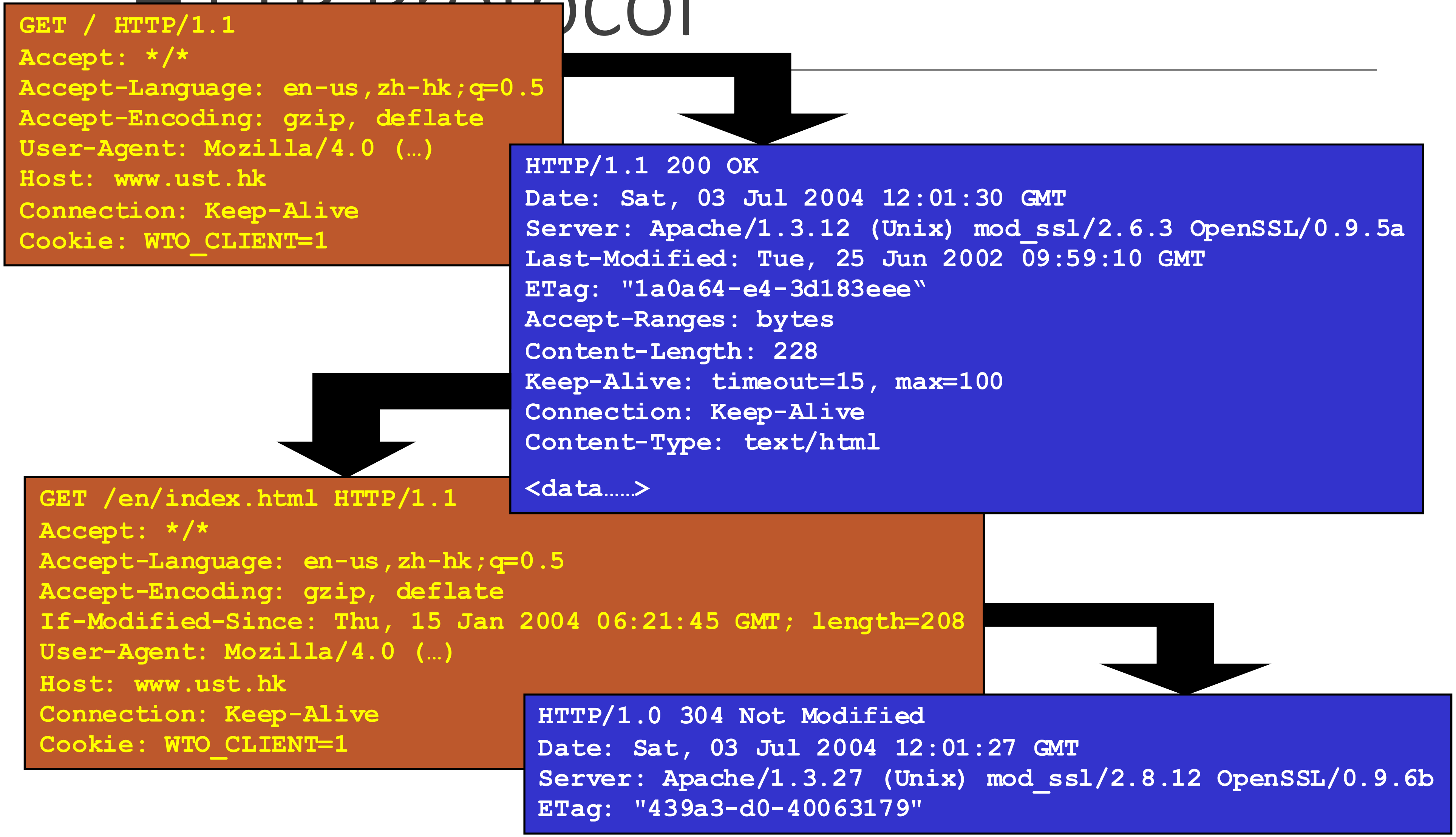
- RFC 1945 HTTP 1.0 (1996)
- RFC 2616 HTTP 1.1 (1999)

Hyper Text Transfer Protocol

- Defined in RFC2616
- Run on TCP
- Commonly used version:
  - HTTP/1.0
  - HTTP/1.1, added
    - Persistent connections
    - OPTIONS method
    - Host header
    - Cache-Control header
    - etc

HTTP Protocol

HTTP Protocol



```
GET / HTTP/1.1
Accept: */*
Accept-Language: en-us,zh-hk;q=0.5
Accept-Encoding: gzip, deflate
User-Agent: Mozilla/4.0 (…)
Host: www.ust.hk
Connection: Keep-Alive
Cookie: WTO_CLIENT=1
```

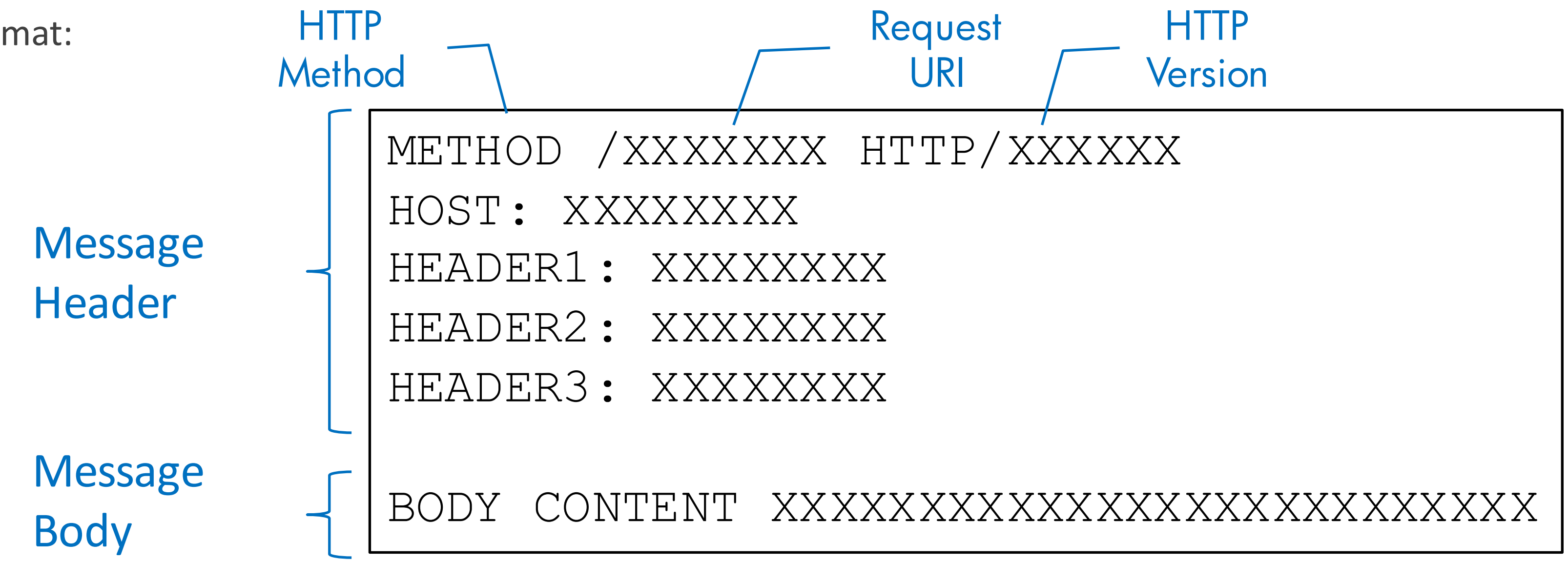
```
HTTP/1.1 200 OK
Date: Sat, 03 Jul 2004 12:01:30 GMT
Server: Apache/1.3.12 (Unix) mod_ssl/2.6.3 OpenSSL/0.9.5a
Last-Modified: Tue, 25 Jun 2002 09:59:10 GMT
ETag: "1a0a64-e4-3d183eee"
Accept-Ranges: bytes
Content-Length: 228
Keep-Alive: timeout=15, max=100
Connection: Keep-Alive
Content-Type: text/html

<data……>
```

```
GET /en/index.html HTTP/1.1
Accept: */*
Accept-Language: en-us,zh-hk;q=0.5
Accept-Encoding: gzip, deflate
If-Modified-Since: Thu, 15 Jan 2004 06:21:45 GMT; length=208
User-Agent: Mozilla/4.0 (…)
Host: www.ust.hk
Connection: Keep-Alive
Cookie: WTO_CLIENT=1
```

```
HTTP/1.0 304 Not Modified
Date: Sat, 03 Jul 2004 12:01:27 GMT
Server: Apache/1.3.27 (Unix) mod_ssl/2.8.12 OpenSSL/0.9.6b
ETag: "439a3-d0-40063179"
```

HTTP Request

Format:



HTTP Method

Request URI

HTTP Version

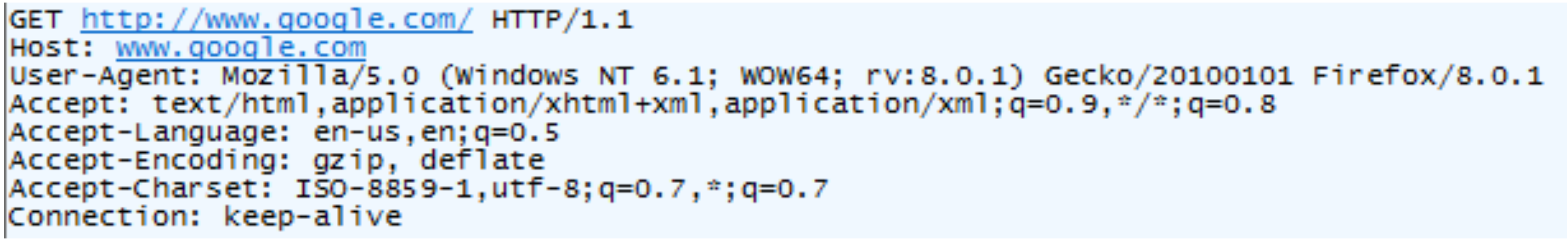
Message Header

Message Body

```
METHOD /XXXXXXX HTTP/XXXXXX
HOST: XXXXXXXX
HEADER1: XXXXXXXX
HEADER2: XXXXXXXX
HEADER3: XXXXXXXX

BODY CONTENT XXXXXXXXXXXXXXXXXXXXXXXXXX
```

- Each line end with the characters 0x0D, 0x0A (\r\n)
- Two blank lines indicates end of message header section
- Body content is optional, length defined by `Content-Length` Header

HTTP Request Example



```
GET http://www.google.com/ HTTP/1.1
Host: www.google.com
User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:8.0.1) Gecko/20100101 Firefox/8.0.1
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-us,en;q=0.5
Accept-Encoding: gzip, deflate
Accept-Charset: ISO-8859-1,utf-8;q=0.7,*;q=0.7
Connection: keep-alive
```

Common HTTP Methods

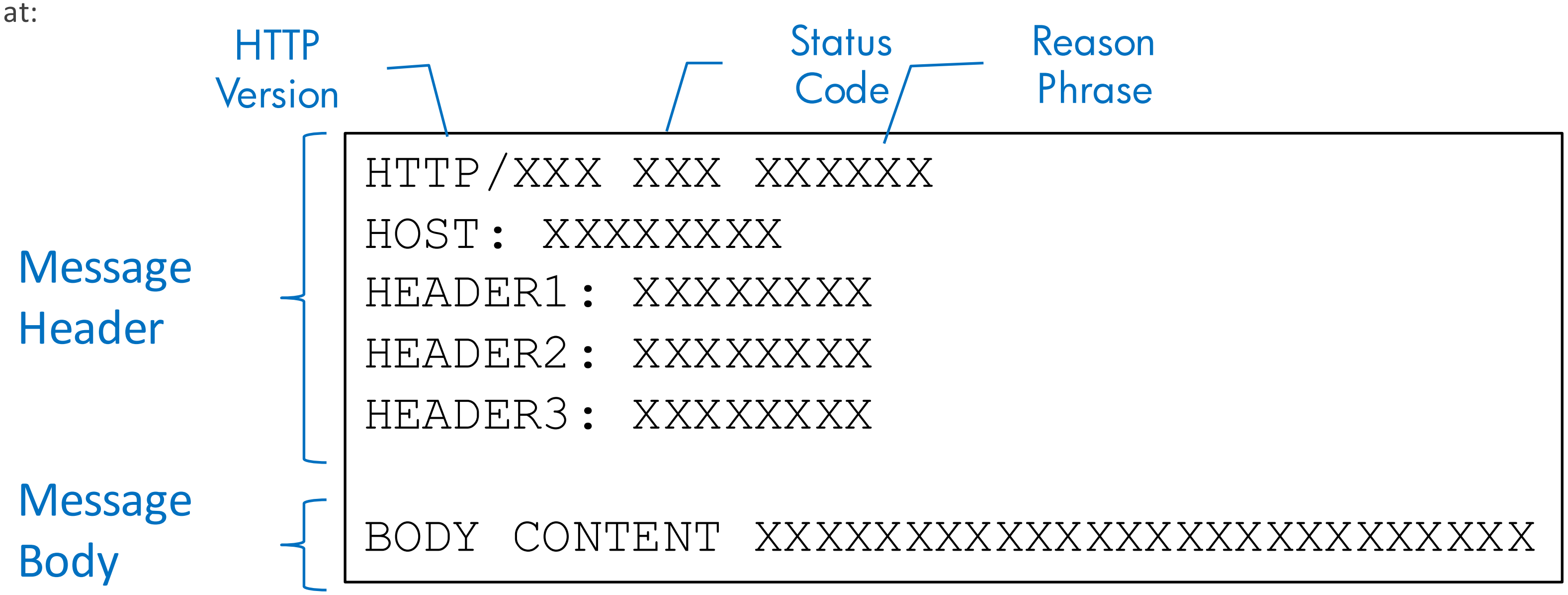
Method	Description
GET	Retrieve information of entity identified by the request URI (Header + content, if any)
POST	Submitting data to the entity identified by the request URI
HEAD	Retrieve only the message header of entity identified by the request URL
PUT	Store (i.e. upload) the content to the entity identified by the request URI
DELETE	Remove the entity identified by the request URI
TRACE/TRACK	Ask the server to invoke an application loopback of the request (i.e. echo the request) as message body
CONNECT	Request the server to establish tunnel to the requested URI (usually used in proxy)
OPTIONS	Return the list of supported (not necessarily allowed) HTTP methods

Common HTTP Request Headers

Request Header	Description
Host	Specify the host requested
Authorization	For HTTP access authentication
Cache-Control	Specify the caching option of the client
Content-Type	Type of message body (in Internet Media Type format, RFC1590)
Content-Length	Size of message body
Content-Range	Use with `multipart/byte-ranges` media type for requesting partial content
Connection	Whether to use persistent connection or not
Referer	Specify where the request origins from
Origin	A proposal for content security policy, similar to Referer header, but only include the scheme, host and port (i.e. no path data or query string)
User-Agent	Strings identifying the type of user agent used by HTTP client
Cookie	Browser cookies

HTTP Response

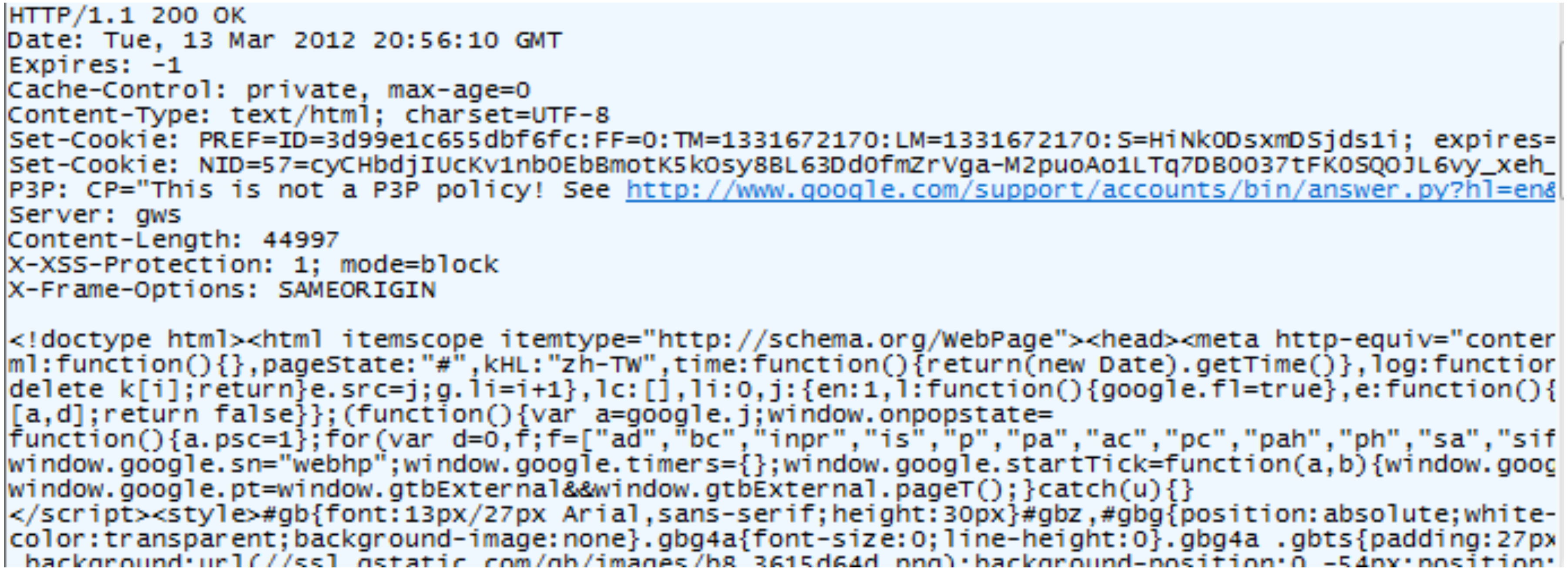
Format:



```
HTTP/XXX XXX XXXXXX
HOST: XXXXXXXX
HEADER1: XXXXXXXX
HEADER2: XXXXXXXX
HEADER3: XXXXXXXX

BODY CONTENT XXXXXXXXXXXXXXXXXXXXXXXXXX
```

- Each line end with the characters 0x0D, 0x0A (\r\n)
- Two blank lines indicates end of message header section
- Body content is optional, length defined by `Content-Length` Header

HTTP Response Example



```
HTTP/1.1 200 OK
Date: Tue, 13 Mar 2012 20:56:10 GMT
Expires: -1
Cache-Control: private, max-age=0
Content-Type: text/html; charset=UTF-8
Set-Cookie: PREF=ID=3d99e1c655dbf6fc:FF=0:TM=1331672170:LM=1331672170:S=HiNkODsxmDSjds1i; expires=
Set-Cookie: NID=57=cyCHbdjIUcKv1nbOEbBmotK5kOsy8BL63Dd0fmZrVga-M2puoAo1LTq7DB0037tFKOSQOJL6vy_xeh_
P3P: CP="This is not a P3P policy! See http://www.google.com/support/accounts/bin/answer.py?hl=en&
Server: gws
Content-Length: 44997
X-XSS-Protection: 1; mode=block
X-Frame-Options: SAMEORIGIN

<!doctype html><html itemscope itemtype="http://schema.org/WebPage"><head><meta http-equiv="conter
ml:function(){},pageState:"#",kHL:"zh-TW",time:function(){return(new Date).getTime()},log:functior
delete k[i];return}e.src=j;g.li=i+1},lc:[],li:0,j:{en:1,l:function(){google.fl=true},e:function(){
[a,d];return false}};(function(){var a=google.j;window.onpopstate=
function(){a.psc=1};for(var d=0,f;f=["ad","bc","inpr","is","p","pa","ac","pc","pah","ph","sa","sif
window.google.sn="webhp";window.google.timers={};window.google.startTick=function(a,b){window.goog
window.google.pt=window.gtbExternal&&window.gtbExternal.pageT();}catch(u){}
</script><style>#gb{font:13px/27px Arial,sans-serif;height:30px}#gbz,#gbg{position:absolute;white-
color:transparent;background-image:none}.gbg4a{font-size:0;line-height:0}.gbg4a .gbts{padding:27px
 background:url(//ssl.gstatic.com/gb/images/b8_3615d64d.png);background-position:0 -54px;position:
```

Class of HTTP Response

- Specified by the 1st digit of the response status code

Class	Descriptions
1xx	Informational – Request received, continuing process
2xx	Success – The action was successfully received, understood and accepted
3xx	Redirection – Further action must be taken in order to complete the request
4xx	Client Error – The request contains bad syntax or cannot be fulfilled
5xx	Server Error – The server failed to fulfill an apparently valid request

Common HTTP Status (1)

Status Code	Reason Phrase	Remarks
100	Continue	Tell the client that it may continue to send message body. Requiring the `Expect: 100-continue` header. (HTTP/1.1 only)
200	OK	
201	Created	Entity is recreated on the server
204	No Content	
206	Partial Content	For `multipart/byte-ranges` media type
301	Moved Permanently	
304	Not Modified	Tell the HTTP client that the requested entity is not updated since last time, therefore ok to return cached content to user
307	Temporary Redirect	

Common HTTP Status (2)

Status Code	Reason Phrase	Remarks
400	Bad Request	The request is not in valid format
401	Unauthorized	Tell the client that authorization is required
403	Forbidden	Authentication failed or the user doesn't allow to access the requested entity
404	Not Found	
405	Method Not Allowed	
500	Internal Server Error	Possible cause: server misconfiguration, server script error
501	Not Implemented	
503	Service Unavailable	May indicate that the server is unable to handle HTTP request.

Extension of HTTP Status

- The HTTP server may extend the standard set of HTTP status
- E.g. IIS 7.0 defines a set of sub-status code to tell more information about the reason:

Status Code	Reason Phrase
401.1	Login Failed
401.2	Login Failed due to Server Configuration
401.3	Unauthorized due to ACL on resource
403.1	Execute access forbidden
403.2	Read access forbidden
403.3	Write access forbidden
403.4	SSL required

- More info: http://support.microsoft.com/kb/943891

Common HTTP Response Headers

Request Header	Description
WWW-Authenticate	For HTTP access authentication
Content-Type	Type of message body (in Internet Media Type format, RFC1590)
Content-Length	Size of message body
Content-Range	For `multipart/byte-ranges` media type
Location	Specify the location of the requested entity (often seen in redirection class of responses)
Date	Server date and time
Server	Specify the server information
Last-Modified	Last modified time of the requested resources
Set-Cookie	Instruct the HTTP client to set a browser cookie

HTTP Access Authentication (1)

- Defined in RFC2617
- Specified by the `Authorization` request header:
- If the client requests an entity requiring authorization, the server responds with a 401 message with `WWW-Authenticate` header, which is a challenge string.
- If the client failed to authenticate, then the server responds with a 403 message.
- Two type of common authentication methods
  - Basic
  - Digest

HTTP Access Authentication (2)

- Basic Authentication Example
  - Server request authentication in a 401 message:
    - `WWW-Authenticate: Basic realm="Restricted area"`
  - Client send authentication information:
    - `Authorization: Basic dXNlcjpwYXNzd29yZA==`
    - The blue string is the Base 64 encoded version of the user id and password (user:password in the example)
    - Easy to decode

HTTP Access Authentication (3)

- Digest Authentication Example
  - Server request authentication in a 401 message:

```
– WWW-Authenticate: Digest realm="realm",
          qop="auth,auth-int",
          algorithm="MD5",
          nonce="dcd98b7102dd2f0e8b11d0f600bfb0c093",
          opaque="5ccc069c403ebaf9f0171e9517f40e41"
```

  - Client send authentication information:

```
– Authorization: Digest username="user",
          realm="realm",
          nonce="dcd98b7102dd2f0e8b11d0f600bfb0c093",
          uri="/dir/index.html",
          qop=auth, nc=00000001, nonce="0a4f113b",
          response="6629fae49393a05397450978507c4ef1",
          opaque="5ccc069c403ebaf9f0171e9517f40e41“
```

  - Involves the use of hashing algorithm (by default MD5) and nonce to produce digest and checksum

Web programming

HyperText Markup Languge (HTML)

The most popular language for web pages

Latest specifications is 4.01 by W3C (Dec. 1999)

5.0 is published by W3C

A structure document with different elements, tags, attributes, etc.

Difference browsers may generate page differently and may accept different elements

HTML

- Stands for “Hyper Text Markup Language”
- Defines how a webpage looks like
- Most commonly used version: HTML4
- Latest version: HTML5 (candidate recommendation), introduces many new features
  - <video>, <audio>, <canvas> tag
  - A large set of new API:
    - Web storage
    - Drag-and-drop
    - Cross-document messaging
    - etc
- Also some related specification published (not exactly in HTML5):
  - Geolocation API
  - File API
  - Indexed Database API
- XHTML is a XML serialization of HTML

Cascading Style Sheets (CSS)

Describe the look and free of a web page

- Analogy to templates of documents

Heavily used by web applications to have a consistent web layout to customers

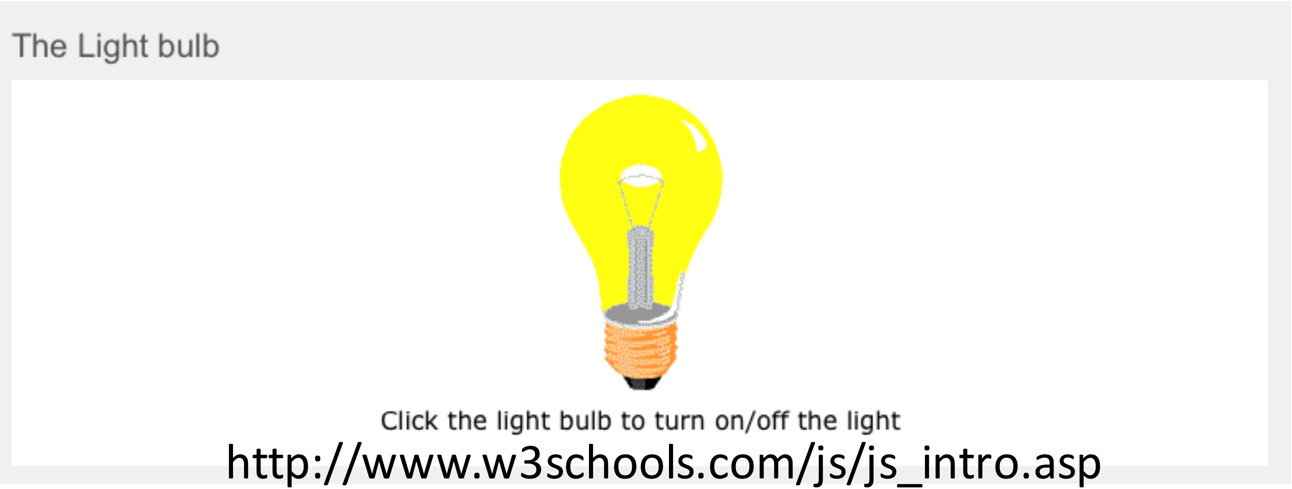
Nonetheless, not all web browsers support all CSS standards

JavaScript

JavaScript, was created to enable scripting access to all the elements of an HTML document.

It is most commonly used as part of web browsers, whose implementations allow client-side scripts to interact with the user, control the browser, communicate asynchronously, and alter the document content that is displayed

Ajax is the main process behind what is now known as Web 2.0, in which web pages have started to resemble standalone programs, because they don't have to be reloaded in their entirety.



The Light bulb

Click the light bulb to turn on/off the light

http://www.w3schools.com/js/js_intro.asp

Javascript

- Most common type of client scripts running in web browsers
- ECMA-based
- “Weak typed”
- Have some common frameworks / library
  - Jquery
  - Prototype, etc

JavaScript

Placed in HTML head section

- loaded before display web page

Placed in HTML body section

- loaded during (or after) display web page
- it is a good idea to place scripts at the bottom of the <body> element to improve page load because HTML display is not blocked by scripts loading

Placed in external files

- loaded as if it was located exactly where the <script> tag is located
- practical when the same code is used in many different web pages

JavaScript

Output to an HTML element

- document.getElementById("demo").innerHTML

Output to the HTML output

- document.write()

Output to an alert box

- window.alert()

Output to the browser console

- console.log()

JavaScript

Can change HTML content

- document.getElementById("demo").innerHTML = "hello";

Can change HTML attributes

- document.getElementById("demo").src = "hello.gif";

Can change HTML styles

- document.getElementById("demo").style.fontSize = "25px";

Can validate data

- if (document.getElementById("demo").value==3) {

Data Validation

The purpose of data validation is to ensure correct input to a computer application.

It is the process of ensuring that computer input is clean, correct, and useful.

Typical validation tasks are:

- has the user filled in all required fields?
- has the user entered a valid date?
- has the user entered text in a numeric field?

Validation can be defined by many different methods, and deployed in many different ways.

- Client side validation is performed by a web browser (e.g. JS), before input is sent to a web server.
- Server side validation is performed by a web server (e.g. PHP), after input has been sent to the server.

PHP

- PHP development began in 1994 when Rasmus Lerdorf wrote a series of Common Gateway Interface (CGI) binaries in C which he used to maintain his personal homepage
- PHP (recursive acronym for PHP: Hypertext Preprocessor) is a widely-used open source general-purpose scripting language that is especially suited for web development and can be embedded into HTML.
- PHP pages contain HTML with embedded code that does "something”

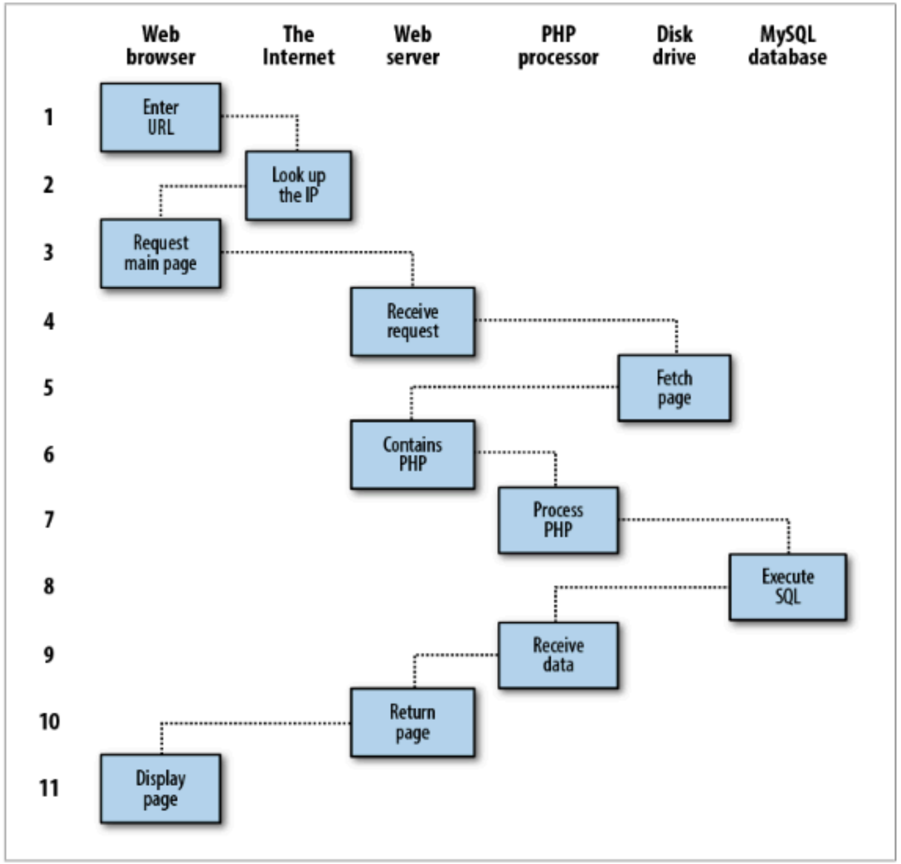


Figure 1-2. A dynamic client/server request/response sequence

http://php.net/manual/en/intro-whatis.php

PHP

Easy enough to be a beginner's first server side language

Deep enough to run the largest social network (Facebook)

Powerful enough to be at the core of the biggest blogging system on the web (WordPress)

Before you continue, you must familiar with HTML

Having a basic understanding of CSS and JS is a plus

PHP

Runs on various platforms (Windows, Linux, Unix, Mac OS X, etc.)

Compatible with almost all servers used today (Apache, IIS, etc.)

Supports a wide range of databases

Rree and open source

Runs efficiently on the server side

PHP

PHP tag

- <?php
-     echo "hello";
- ?>

Short echo tag

- <?="hello";?>

No trailing whitespace nor new lines after closing tag at the end of file

SQL query and integration

SQLite

MySQL

PostgreSQL

MS SQL

- Access
- SQL Server

Oracle

Sybase

DB2

SQL query and integration

DROP DATABASE dbname

CREATE DATABASE dbname

USE dbname

CREATE TABLE tblname (colname VARCHAR(10), …)

INSERT INTO (colname, …) VALUES (fieldval, …)

UPDATE tblname SET colname=fieldval, …

- WHERE … LIKE … AND … OR … IS NULL

DELETE FROM tblname

- WHERE … LIKE … AND … OR … IS NULL

DROP TABLE tblname

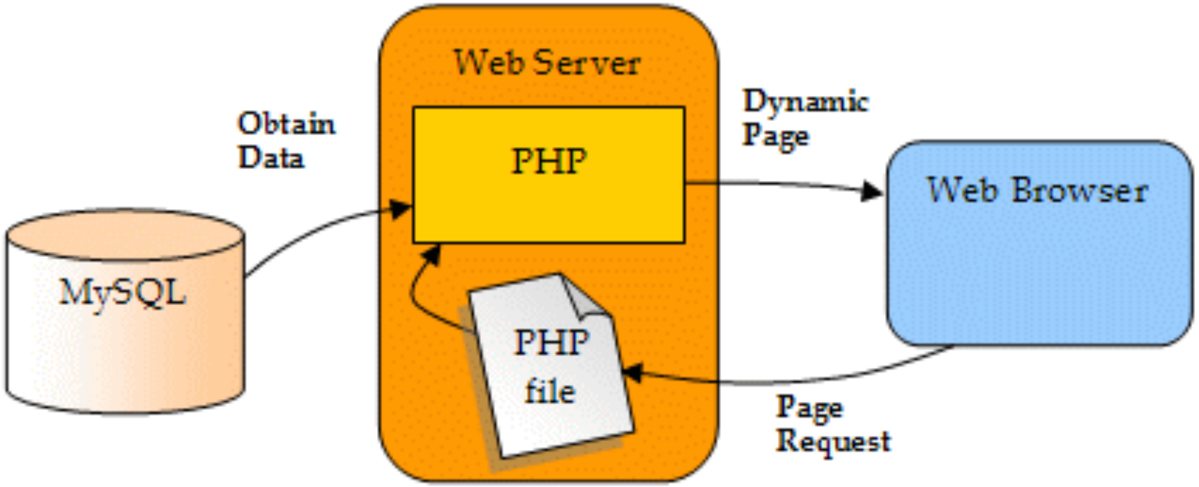
SQL query and integration

SELECT * FROM tblname

- WHERE … LIKE … AND … OR … IS NULL
- LIMIT
- ORDER BY colname
  - DESC
- GROUP BY colname
  - HAVING
- Join
  - inner join
  - outer join
  - UNION

SQL query and integration

- Web site generates the HTML to view the data.
- Database integration refers to combining a database with your web site so that web visitors can add, remove and update information in your database using a web browser. Most database-driven sites also allow users to search [query] the database too.
- Content management systems are now in use on millions of web sites, most often using Drupal, Joomla, Wordpress, or any of the blogging platforms.



http://www.blazedogwebdesign.com/database-integration/

SQL query and integration

PHP MySQL extension

- earlier versions of PHP
- deprecated in 2012
- no prepared statements support

PHP MySQLi extension

- PHP5 and later
- the "i" stands for improved
- offers a procedural API

PHP Data Objects (PDO)

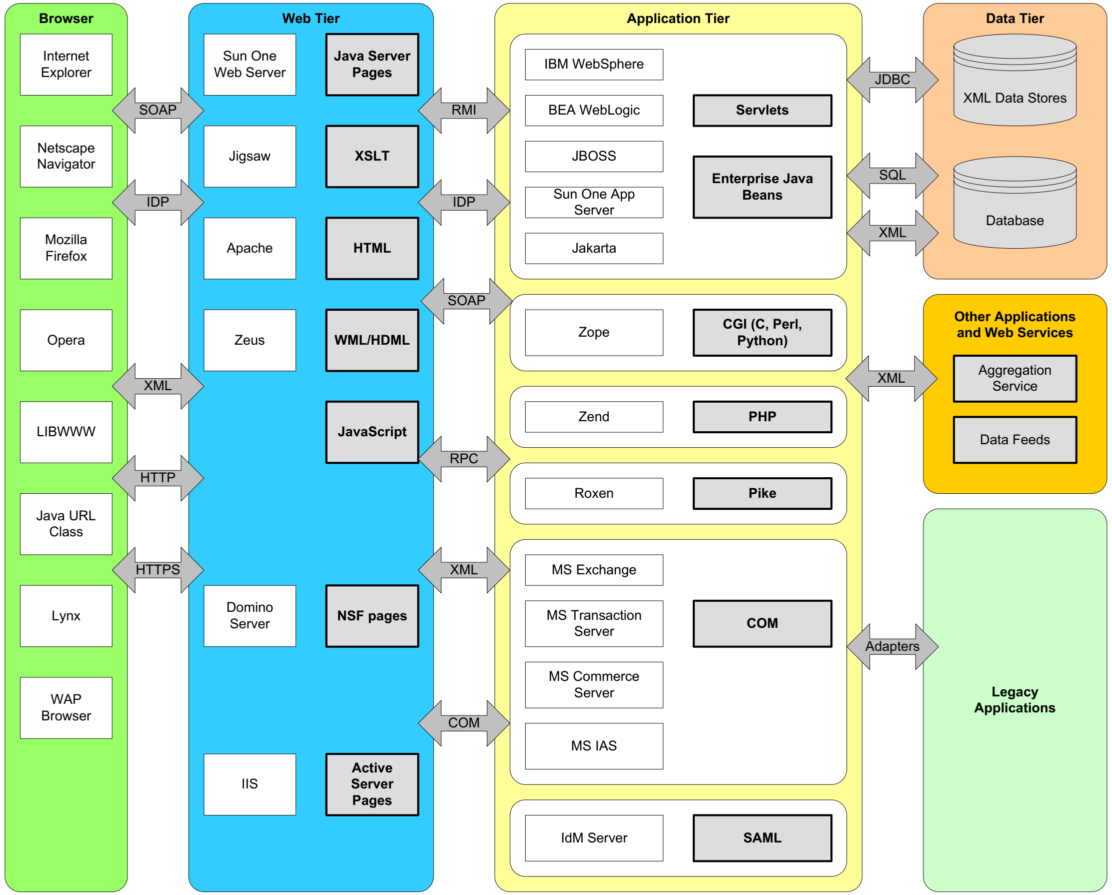
- work on 12 different database systems

Common Web Content

Browser plugin objects

- Common examples are:
  - Adobe Flash objects
  - Java applet
  - Silverlight objects
- Allowed more sophisticated interaction and logic than ordinary HTML + client script combination
- Able to interact with other content in a HTML page via client scripting

Web Applications Relationships



Other Common Web-related Technology & Concepts

Dynamic HTML (DHTML)

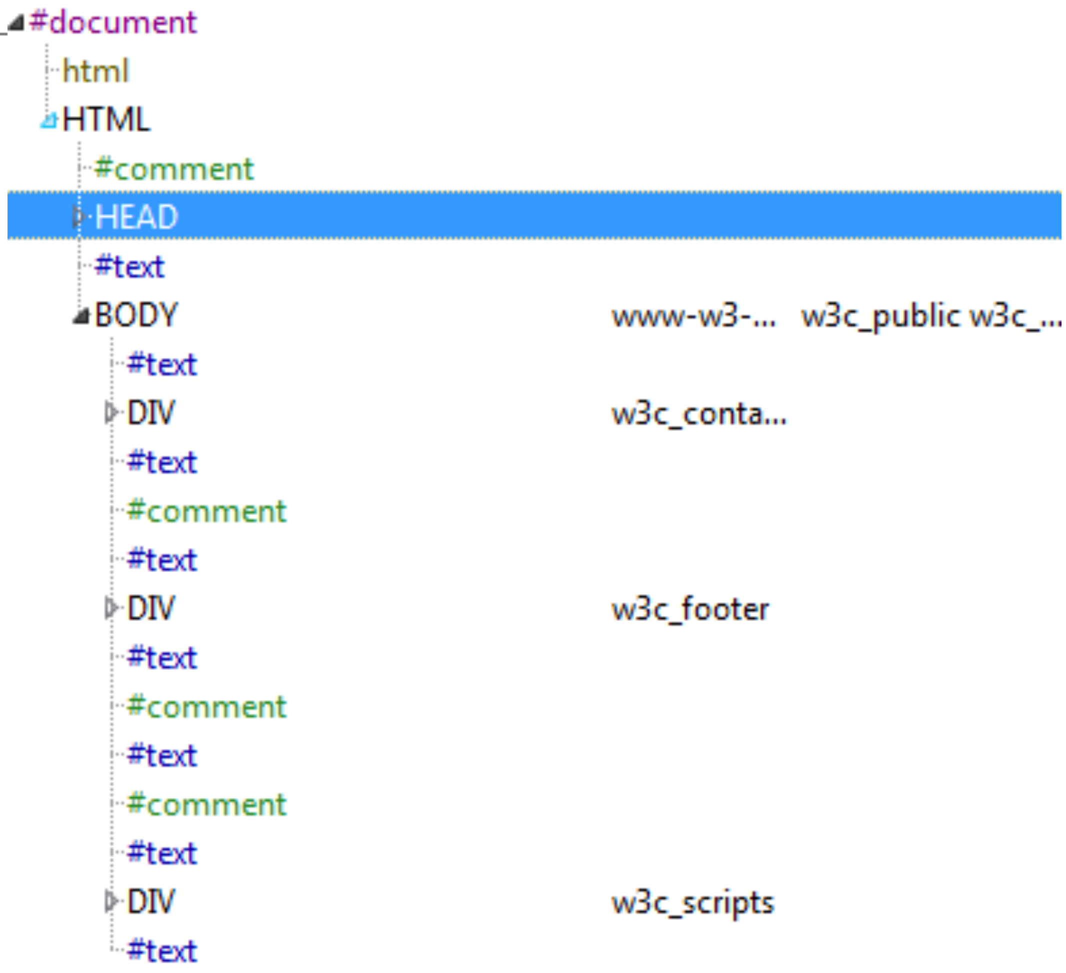
A general term to mean documents enable interactive web sites

A combination of HTML, CSS and scripts (or even

- I don’t think FLASH is an element of DHTML

HTML DOM

- DOM stands for “Document Object Model”
- Representing the HTML document in a tree structure
- How HTML pages actually exist in web client memory after parsing
- Can be manipulated by DOM scripting



```
#document
  html
  HTML
    #comment
    HEAD
    #text
    BODY                          www-w3-...   w3c_public w3c_...
      #text
      DIV                         w3c_conta...
      #text
      #comment
      #text
      DIV                         w3c_footer
      #text
      #comment
      #text
      #comment
      #text
      DIV                         w3c_scripts
      #text
```

XML

Contains markup & content, tag, element, attributes

Begins with an XML declaration:
<?xml version="1.0" encoding="UTF-8" ?>

Commonly used for exchanging data in the Internet

- Comply with the Document Type Definition (DTD) / XML Schema

Version 1.0 in Nov 1998

Version 1.1 in Feb 2004

XML Example

```
<?xml version="1.0" encoding="utf-8"?>

<rss version="2.0" xmlns:atom="http://www.w3.org/2005/Atom" xmlns:cf="http://www.microsoft.com/schemas/rss/core/2005">

<channel xmlns:cfi="http://www.microsoft.com/schemas/rss/core/2005/internal" cfi:lastdownloaderror="None">

<title cf:type="text">明報即時新聞RSS 總目錄</title>

<link>http://inews.mingpao.com/</link>

<description cf:type="text">明報即時新聞RSS</description>

<language>zh-TW</language>

<copyright cf:type="text">明報網站·版權所有·不得轉載 Copyright(C) Mingpao.com All rights reserved.</copyright>

<docs>http://inews.mingpao.com/rss/INews/ALL.xml</docs>

<lastBuildDate>Mon, 7 Jun 2010 01:45:10 GMT</lastBuildDate>

<atom:updated>2010-06-07T01:45:10Z</atom:updated>

<image><url>http://inews.mingpao.com/image/inews_top.gif</url><title>明報即時新聞 總目錄</title><link>http://inews.mingpao.com/</link>
```

Security Assertion Markup Language (SAML)

XML-based standard

- Exchange authentication and authorization data between an identity provider and a service provider

Latest specifications is 2.0 by OASIS (Mar. 2005)

Facilitate Federated SSO

Service Provisioning Markup Language (SPML)

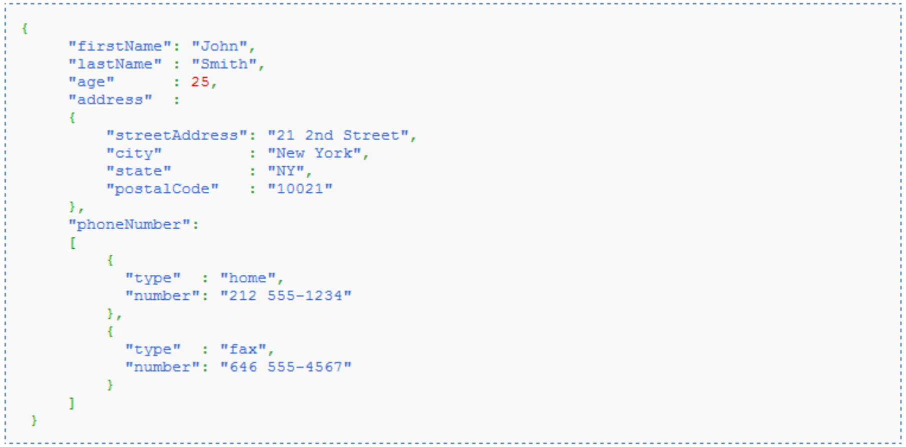
Used for exchanging user, resource and service provisioning information between cooperating organizations

Facilitate integration and interoperation of service provisioning requests

Latest specifications is 2.0 by OASIS (Apr. 2006)

JSON

- Stands for “JavaScript Object Notation”
- For representing simple data structures and objects
- Lightweight and easy to parse
- Described in RFC4627
- Commonly used for data retrieval and exchange



```
{
     "firstName": "John",
     "lastName" : "Smith",
     "age"      : 25,
     "address"  :
     {
         "streetAddress": "21 2nd Street",
         "city"         : "New York",
         "state"        : "NY",
         "postalCode"   : "10021"
     },
     "phoneNumber":
     [
         {
           "type"  : "home",
           "number": "212 555-1234"
         },
         {
           "type"  : "fax",
           "number": "646 555-4567"
         }
     ]
 }
```

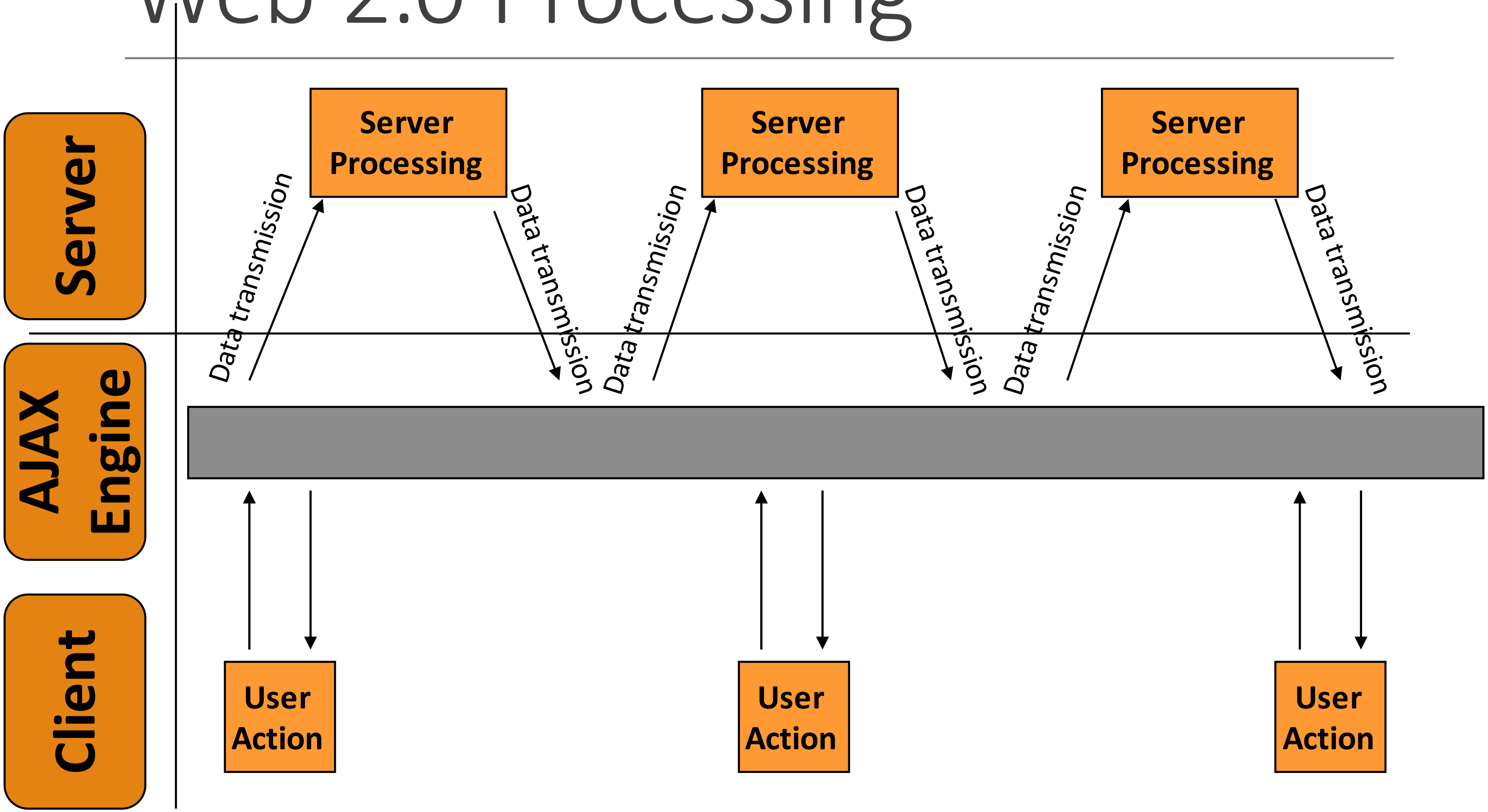
Asynchronous JavaScript and XML (AJAX)

Scripts on the client side to create more interactive web applications

- E.g. Gmail, etc.

Data can be exchanged **asynchronously** (in the background) between servers and client without the need to change existing page

Web 2.0 Processing

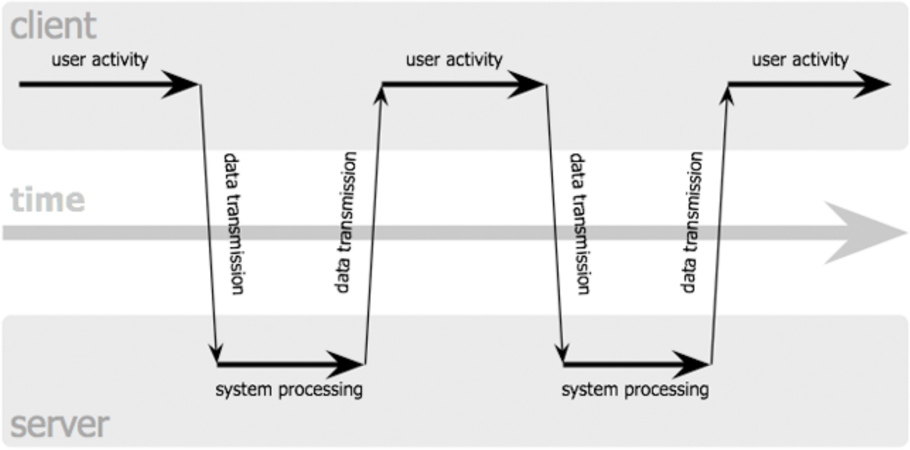


AJAX (1)

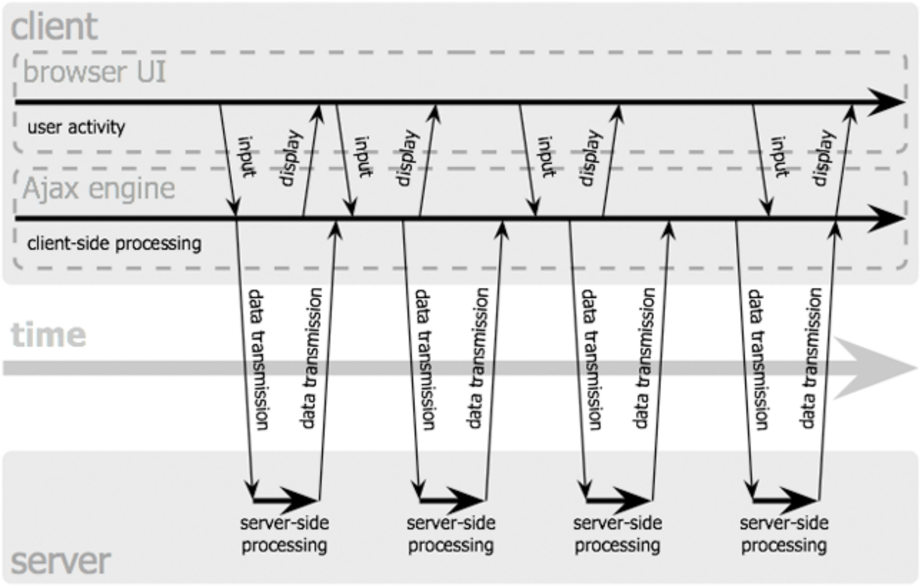
- Stands for “Asynchronous JavaScript and XML”
- A model for retrieving data from web server asynchronously
- Usually `XmlHttpRequest`
- Updates the page content dynamically based on the retrieved content
  - By modifying the DOM structure via DOM scripting
- More responsive UI
- Not necessary to use Javascript and XML (JSON is used in many cases)

AJAX (2)

classic web application model (synchronous)



Ajax web application model (asynchronous)



Jesse James Garrett / adaptivepath.com

(Source: http://www.adaptivepath.com/ideas/ajax-new-approach-web-applications)

SOAP

Application Layer

By World Wide Web Consortium (W3C)

Exchange structured information in the implementation of Web Services

Using XML

RESTful

Representational State Transfer (REST) is a software architecture style for building scalable web services. REST gives a coordinated set of constraints to the design of components in a distributed hypermedia system that can lead to a higher performing and more maintainable architecture

In RESTful architectures, the method information goes into the HTTP method (GET, POST, PUT, DELETE). Given the first line of an HTTP request to a resource-oriented RESTful web service (“GET /reports/open-bugs HTTP/1.1”)

Unlike SOAP-based web services, there is no "official" standard for RESTful web APIs. It relies on HTTP protocol for operations.

HTTP Method	Data operate	Description
POST	Create	Create a resource without id.
GET	Read	Get a resource.
PUT	Update	Update a resource or create a resource with id if not existed.
DELETE	Delete	Delete a resource

Features of RESTful

These resources can be pictures, video files, Web pages, business information, or anything that can be represented in a computer-based system.

The purpose of a service is to provide a window to its clients so that they can access these resources. In general, RESTful services should have following properties and features,

- Representations
- Messages
- URIs
- Uniform interface
- Stateless
- Links between resources
- Caching

REST is a way of developing lightweight Web services that are easy to implement, maintain, and discover. HTTP provides an excellent interface to implement RESTful services with features like a uniform interface and caching.

Reference Books

Related content	Book	Chapter
W6: Web Security	The InfoSec Handbook (2014)	Chapter 6: Application and Web Security
W6: Web Authentication	Computer Security Principles and Practice (2012)	Chapter 23: Internet Authentication Applications
W6: Web Security Trust Levels for B2C security services architecture, Continuum of Security Options	Computer Security Handbook (2014)	Chapter 30: E-Commerce and Web Server Safeguards

References

HTML

- http://www.w3.org

HTML & CSS & JavaScript

- http://www.w3schools.com

PHP

- http://www.php.net/manual/en/

MySQL

- http://dev.mysql.com/doc/refman/5.0/en/

REST

- http://www.restapitutorial.com
- http://www.ics.uci.edu/~fielding/pubs/dissertation/rest_arch_style.htm