

Ultimate Extensible Distributed System

Jim Fawcett

CSE686 - Internet Programming

Summer 2009

Your Assignment

- Your supervisor just handed you a spec for implementation of:
 - Distributed system with universal connectability using sockets
 - Can process an open-ended variety of documents
 - Expandable by 5 orders of magnitude in ten years
 - Can add new tools easily
 - Supports 50 million users a day without gridlock.
- You say ***NO WAY!***
- Well, maybe.

Introduction to Internet and Web

- This presentation addresses two questions:
 - Is that possible?
 - Well yes - look over there - the web!
 - How was it accomplished?
 - Processing structure and protocols
 - Programming tools
 - Web servers and browsers that host:
 - Script languages, e.g., Javascript, VBScript, Perl, Ruby, ...
 - Programming languages:
Visual Basic, Java, C++, C#, ...
 - And, of course, some very smart people

Table of Contents

- Introduction to the Internet and Web
- Internet Design Principles
- Internet and Web History
- Web Technologies
- Pinging Various URLs
- Web Processing Models
- Programming The Web
- Extending The Web
- People in the Web

Goals of the Internet:

- Build distributed system to share documents.
- Support expansion by 5 orders of magnitude in ten years - 200 hosts to 500 million hosts.
- Manage communication between hundreds of millions of machines every day without collapsing from congestion.
- Provide for arbitrary extensions:
 - From static text documents to graphics, dynamic content, streaming video, programmable interfaces, voice, ...

Internet Design Principles

- Goal is connectivity
- Achieved with Internet Protocol (IP)
 - Stateless so survives failures - no need to backup
- Made scalable with end-to-end intelligence
 - Transport Control Protocol (TCP)
 - Sender does not send until receipt is acknowledged
 - Amount sent is based on receiver's current available buffer size - so receiver won't be flooded.
 - Be strict when sending and tolerant when receiving
- Protocol Specific Packet Headers
- Internet Design
- Robustness and the Internet

Original Goals of the Web

- Universal readership
 - When content is available it should be accessible from any type of computer, anywhere.
- Interconnecting all things
 - Hypertext links everywhere.
 - Simple authoring

Web Design Principles

- Universal
- Decentralized
- Modular
- Extensible
- Scalable
- Accessible
- Forward/backwards compatibility
- Architecture of World Wide Web

Basic Concepts

- Client/Server Model
- Universal Addressing
 - TCP/IP, DNS
- Search Engines
- Universal Protocols
 - HTTP, URLs, HTML, FTP
- Format Negotiation through HTTP
- Hypertext → Hypermedia via HTML
 - XHTML → HTML5
 - Support for text, images, sound, and scripting

Internet and Web History

[Table of Contents](#)

Internet History

- 1961 - First paper on packet-switching theory, Kleinrock, MIT
- 1969 - ARPANet goes on line
 - Four hosts, each connected to at least two others
- 1974 - TCP/IP, Berkley Sockets invented
- 1983 - TCP/IP becomes only official protocol
- 1983 - Name server developed at University of Wisconsin.
- 1984 - Work begins on NSFNET
- 1990 - ARPANET shutdown and dismantled
- 1990 - ANSNET takes over NSFNET
 - Non-profit organization - MERIT, MCI, IBM
 - Starts commercialization of the internet
- 1995 - NSFNET backbone retired
- 1998 - DNS transferred from Dept of Commerce to ICANN
- 2000 - Web size estimates surpass 1 billion indexable pages

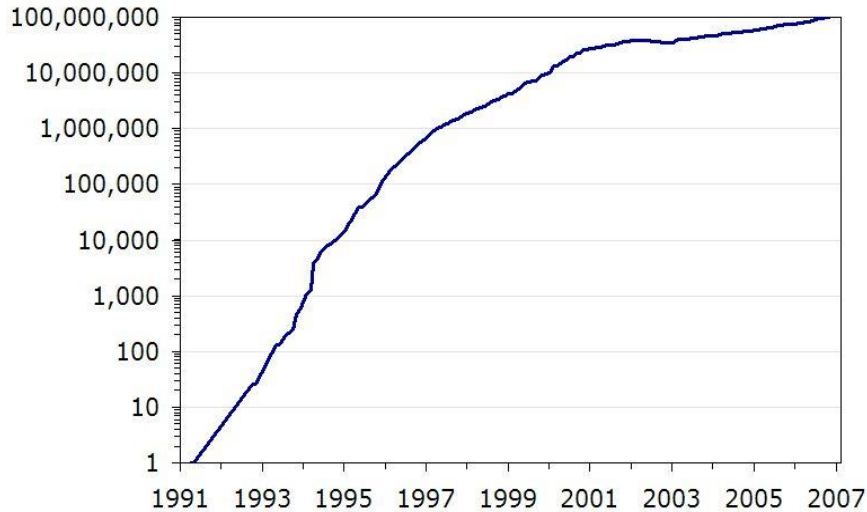
Web History

- 1990 - World Wide Web project
 - Tim Berners-Lee starts project at CERN
 - Demonstrates browser/editor accessing hypertext files
 - HTTP 0.9 defined, supports only hypertext, linked to port 80
- 1991 - first web server outside Europe
 - CERN releases WWW, installed at SLAC
- 1992 - HTTP 1.0, supports images, scripts as well
- 1993 - Growth phase
- 1994 - CERN and MIT agree to set up WWW Consortium
- 1999 - HTTP 1.1, supports open ended extensions

Web Growth Phase - 1993

- InterNIC created to provide registration services
- WWW (port 80 HTTP) traffic is 1% of NSFNET traffic
- 200 Known HTTP servers
- Article on WWW in New York Times
- Mosaic first release

The following chart plots the number of sites from 1991 to 2006. Note the use of a logarithmic scale, which is the only way to represent the Web's fast rate of change in its early years.

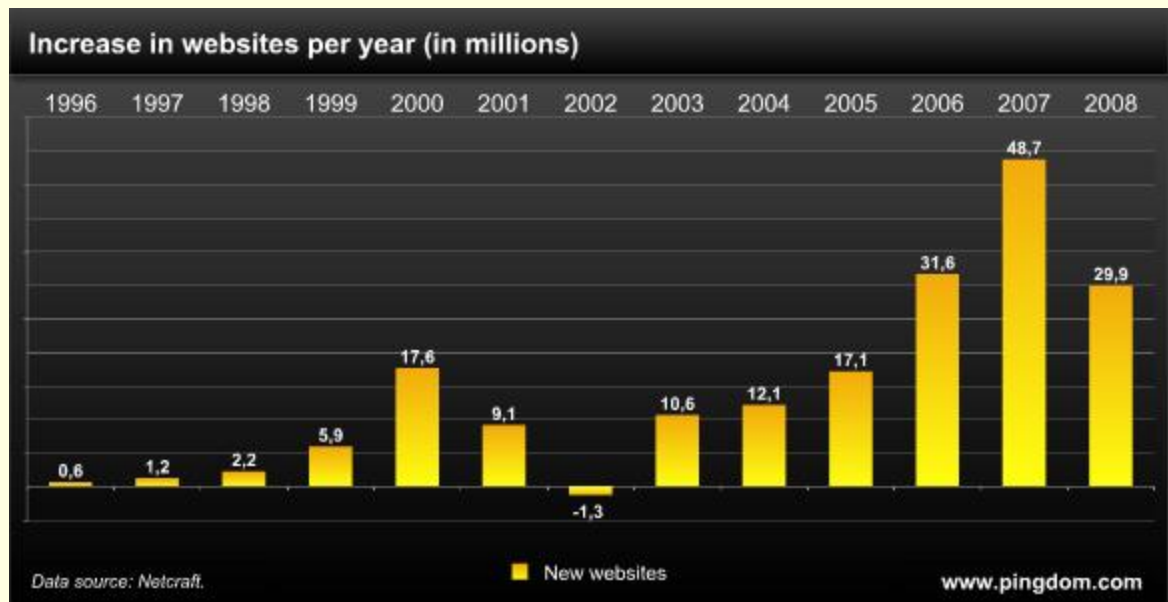


The **number of Internet websites** each year since the Web's founding.

Web Growth

<http://www.useit.com/alertbox/web-growth.html>

<http://www.techcrunch.com/2009/05/08/is-the-growth-of-the-web-slowing-down-or-just-taking-a-breather/>



[Table of Contents](#)

Web Technologies

[Table of Contents](#)

Tools: Servers on the Internet

- HTTP - HyperText Transport Protocol
 - JSP and ASP add dynamic content
 - Web Services add RPC program interface
- FTP - File Transport Protocol
- Gopher - Text and Menus
- NNTP - Network News Transfer Protocol
- DNS - Distributed Name Service
- telnet - log into a remote computer
- New tools - if they use TCP/IP just add them

Network Protocol Stack

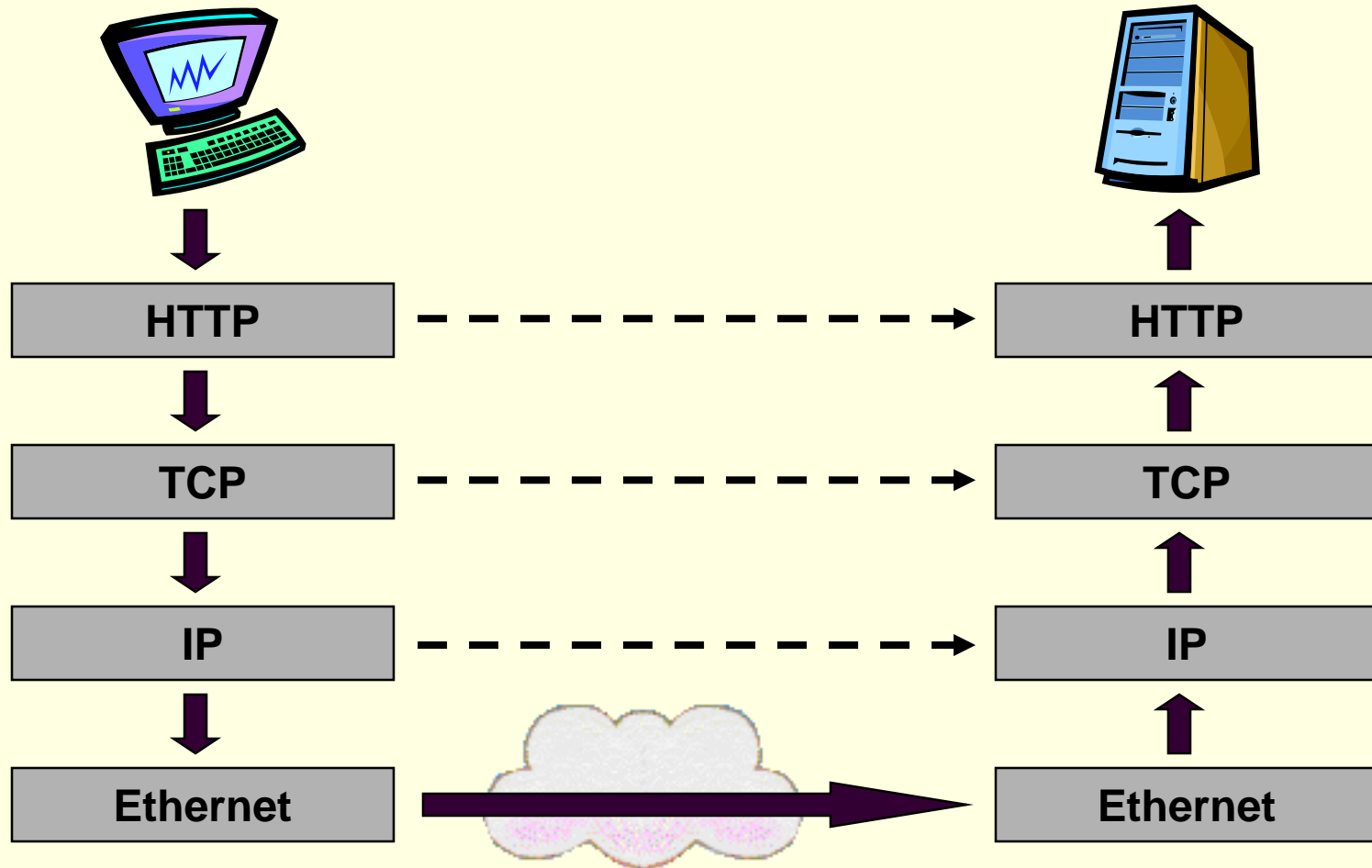
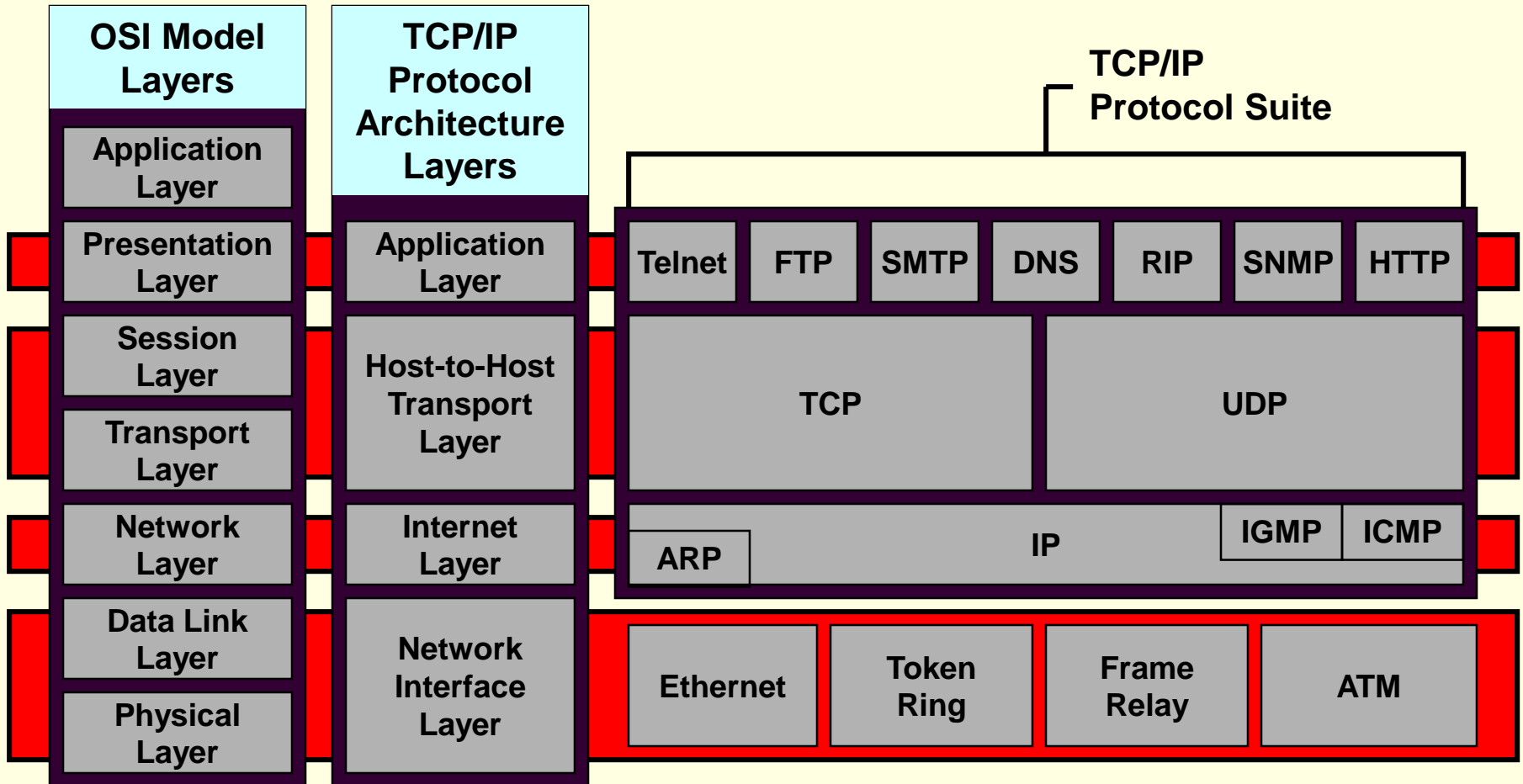


Table of Contents

Network Protocols



[Table of Contents](#)

Networks - Transport Layer

- Provides efficient, reliable and cost-effective service
- Uses Sockets programming model
- Ports identify application
 - Well-known ports identify standard services (e.g. HTTP uses port 80, SMTP uses port 25)
- Transmission Control Protocol (TCP)
 - Provides reliable, connection-oriented byte stream
- UDP
 - Connectionless, efficient, unreliable

Communication Between Networks

- Internet Protocol (IP)
 - Routable, connectionless datagram delivery
 - Specifies source and destination
 - Does not guarantee reliable delivery
 - Large message may be broken into many datagrams, not guaranteed to arrive in the order sent
- Transport Control Protocol (TCP)
 - Reliable stream transport service
 - Datagrams are delivered to the receiving application in the order sent
 - Error control is provided to improve reliability

Pinging Various URLs

```
C:\ CMD.EXE
Pinging bismark [192.168.0.103] with 32 bytes of data:
Reply from 192.168.0.103: bytes=32 time=1ms TTL=128
Reply from 192.168.0.103: bytes=32 time=2ms TTL=128
Reply from 192.168.0.103: bytes=32 time=19ms TTL=128
Reply from 192.168.0.103: bytes=32 time=6ms TTL=128

Ping statistics for 192.168.0.103:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 1ms, Maximum = 19ms, Average = 7ms

C:\temp
>ping www.ecs.syr.edu

Pinging ecswww.syr.edu [128.230.208.33] with 32 bytes of data:
Reply from 128.230.208.33: bytes=32 time=22ms TTL=113
Reply from 128.230.208.33: bytes=32 time=23ms TTL=113
Reply from 128.230.208.33: bytes=32 time=24ms TTL=113
Reply from 128.230.208.33: bytes=32 time=23ms TTL=113

Ping statistics for 128.230.208.33:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 22ms, Maximum = 24ms, Average = 23ms

C:\temp
>ping www.moscow-guide.ru

Pinging moscow-guide.ru [81.176.69.152] with 32 bytes of data:
Reply from 81.176.69.152: bytes=32 time=156ms TTL=42
Reply from 81.176.69.152: bytes=32 time=156ms TTL=42
Reply from 81.176.69.152: bytes=32 time=178ms TTL=42
Reply from 81.176.69.152: bytes=32 time=155ms TTL=42

Ping statistics for 81.176.69.152:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 155ms, Maximum = 178ms, Average = 161ms
```

Ping in network
– few millisecond

Ping in Syracuse
– few tens of millisecond

Ping to Moscow
– few hundreds of millisecond

Tracing HTTP Message with Tracert

```
C:\>tracert www.moscow-guide.ru

Tracing route to moscow-guide.ru [81.176.69.152]
over a maximum of 30 hops:

  0  1 ms    1 ms    1 ms    192.168.0.1
  1  7 ms    7 ms    7 ms    10.101.208.1
  2  8 ms   10 ms    7 ms    fas3-2.syrcnybsh-rtr01.nyroc.rr.com [24.92.227.138]
  3  7 ms    9 ms    7 ms    srp2-0.syrcnyspp-rtr04.nyroc.rr.com [24.92.227.217]
  4  8 ms    7 ms    7 ms    srp10-0.syrcnyspp-rtr01.nyroc.rr.com [24.92.224.137]
  5  7 ms    7 ms    8 ms    srp8-0.syrcnyspp-rtr02.nyroc.rr.com [24.92.224.138]
  6  11 ms   11 ms   11 ms    son0-1-1.albnywav-rtr03.nyroc.rr.com [24.92.224.170]
  7  13 ms   12 ms   11 ms    pop1-alb-P7-0.atdn.net [66.185.133.229]
  8  14 ms   12 ms   11 ms    bb1-alb-P0-1.atdn.net [66.185.148.100]
  9  18 ms   15 ms   19 ms    bb2-nye-P3-0.atdn.net [66.185.152.71]
 10  16 ms   29 ms   16 ms    pop1-nye-P1-0.atdn.net [66.185.151.51]
 11  16 ms   15 ms   15 ms    0.so-2-0-0.BR1.NYC4.ALTER.NET [204.255.173.33]
 12  17 ms   15 ms   15 ms    0.so-6-0-0.XL1.NYC4.ALTER.NET [152.63.21.78]
 13  16 ms   18 ms   15 ms    0.so-4-0-0.TL1.NYC9.ALTER.NET [152.63.0.173]
 14  *      18 ms   16 ms    0.so-7-0-0.IL1.NYC9.ALTER.NET [152.63.9.245]
 15  15 ms   40 ms   15 ms    0.so-1-0-0.IR1.NYC12.ALTER.NET [152.63.23.62]
 16  95 ms   94 ms   95 ms    so-0-0-0.TR2.LND9.ALTER.NET [146.188.15.26]
 17  96 ms   97 ms   94 ms    so-6-0-0.XR1.LND9.ALTER.NET [146.188.15.42]
 18  94 ms   94 ms   94 ms    POS3-0.cr1.lnd10.gbb.uk.uu.net [158.43.150.97]
 19  99 ms   98 ms   99 ms    pos3-0.cr1.lnd8.gbb.uk.uu.net [158.43.253.142]
 20 104 ms   98 ms   99 ms    ge0-0.gw1.lnd8.gbb.uk.uu.net [158.43.188.25]
 21 149 ms  149 ms  150 ms    rtcomm-gw.customer.ALTER.NET [146.188.66.50]
 22 156 ms  156 ms  156 ms    msk-dsr7-ge1-0-0-0.rt-comm.ru [217.106.7.200]
 23 156 ms  159 ms  155 ms    81.176.69.152

Trace complete.
```

HTTP Messages

as seen by packet sniffer

TCP 113 192.168.0.102 207.46.144.188 2834 80 [2004.05.19 - 12:15:20.718]

```
E qSó@ €...šÀ` fİ.¼
P,X {ÈEPDpÑ¼ GET /ms.htm HTTP/1.1
Connection: Keep-Alive
Host: www.microsoft.com
```

method

Request Message

TCP 1102 207.46.144.188 192.168.0.102 80 2834 [2004.05.19 - 12:15:20.843]

```
E Nç-@ nEİ.¼À` f P
{ÈE,XIPý¶jà HTTP/1.1 200 OK
Cache-Control: max-age=60
Content-Length: 669
Content-Type: text/html
Last-Modified: Thu, 11 Jul 2002 17:05:42 GMT
Accept-Ranges: bytes
ETag: "be61bb30fd28c21:27b"
Server: Microsoft-IIS/6.0
P3P: CP="ALL IND DSP COR ADM CONo CUR CUSo IVAo IVDo PSA PSD TAI TELo OUR SAMo CNT COM INT NAV ONL PHY PRE PUR UNI"
X-Powered-By: ASP.NET
Date: Wed, 19 May 2004 16:15:16 GMT
```

headers

Response Message

```
<!--TOOLBAR_START-->
<!--TOOLBAR_EXEMPT-->
<!--TOOLBAR_END-->
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN"
"http://www.w3.org/TR/REC-html40/loose.dtd">
<HTML>
<HEAD>
<META HTTP-EQUIV="Refresh" CONTENT="0; URL=/">
<TITLE>Microsoft Corporation -- Where Do You Want to Go Today?</TITLE>
</HEAD>
<BODY BGCOLOR="#FFFFFF" TEXT="#000000">
<FONT FACE="Verdana, Arial, Helvetica" SIZE=2>
If your browser can't handle redirect, please click <a href=/">here</a>
</FONT>
</BODY>
</HTML>
```

message body

Typical HTTP Transaction

- Client browser finds a machine address from an internet Domain Name Server (DNS).
- Client and Server open TCP/IP socket connection.
- Server waits for a request.
- Browser sends a verb and an object:
 - GET XYZ.HTM or POST form
 - If there is an error server can send back an HTML-based explanation.
- Server applies headers to a returned HTML file and delivers to browser.
- Client and Server close connection.
 - It is possible for the client to request the connection stay open with HTTP 1.1.

HTTP Methods

- GET request-URI HTTP/1.1
 - Retrieve entity specified in request-URI as body of response message
- POST request-URI HTTP/1.1
 - Sends data in message body to the entity specified in request-URI
- PUT request-URI HTTP/1.1
 - Sends entity in message body to become newly created entity specified by request-URI
- HEAD request-URI HTTP/1.1
 - Same as GET except the server does not send specified entity in response message
- DELETE request-URI HTTP/1.1
 - Request to delete entity specified in request-URI.
- TRACE request-URI HTTP/1.1
 - Request for each host node to report back

HTTP Request

Method

File

HTTP version

Headers

GET /default.asp HTTP/1.0

Accept: image/gif, image/x-bitmap, image/jpeg, */*

Accept-Language: en

User-Agent: Mozilla/1.22 (compatible; MSIE 2.0; Windows 95)

Connection: Keep-Alive

If-Modified-Since: Sunday, 17-Apr-96 04:32:58 GMT

Blank line

Data – none for GET

[Table of Contents](#)

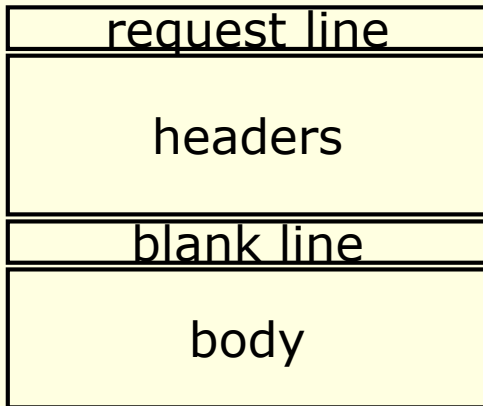
Multipurpose Internet Mail Extensions (MIME)

[skip to HTTP Response](#)

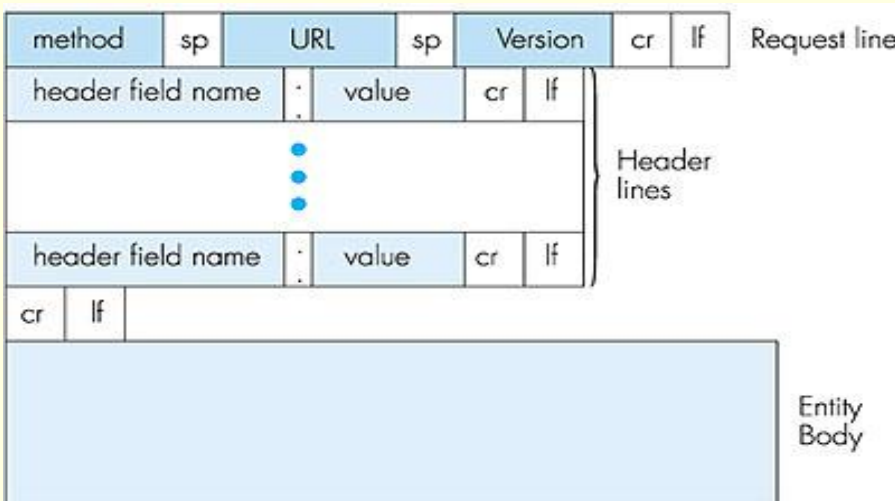
- Defines types of data/documents
 - text/plain
 - text/html
 - image/gif
 - image/jpeg
 - audio/x-pn-realaudio
 - audio/x-ms-wma
 - video/x-ms-asf
 - application/octet-stream

[Table of Contents](#)

Request Message



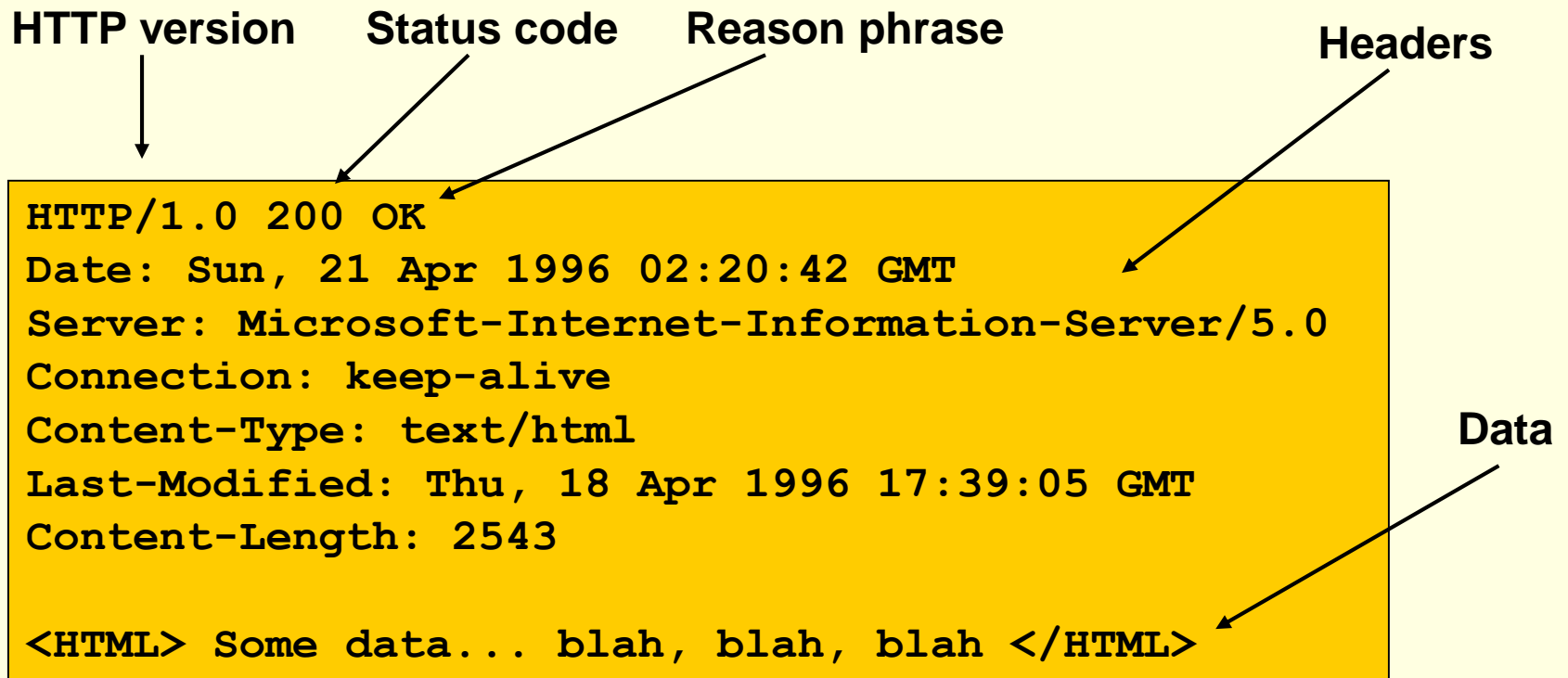
request methods:
**DELETE, GET, HEAD, POST,
 PUT, TRACE**



```
GET /pub/index.html HTTP/1.0
Date: Wed, 20 Mar 2002 10:00:02 GMT
Pragma: no-cache
From: amer@udel.edu
User-Agent: Mozilla/4.03
```

HTTP Response

skip to Programming the Web



Status Codes

200	OK		
201	Created		
202	Accepted		
204	No Content		
301	Moved Permanently		
302	Moved Temporarily		
304	Not Modified		
400	Bad Request		
401	Unauthorized		
403	Forbidden		
404	Not Found		
500	Internal Server Error		
501	Not Implemented		
502	Bad Gateway		
503	Service Unavailable		

Classes:	
1xx: Informational	- not used, reserved for future
2xx: Success	- action was successfully received, understood, and accepted
3xx: Redirection	- further action needed to complete request
4xx: Client Error	- request contains bad syntax or cannot be fulfilled
5xx: Server Error	- server failed to fulfill an apparently valid request

Web Processing Models

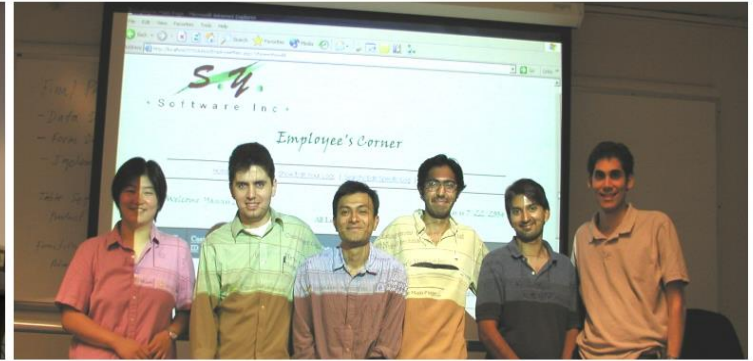
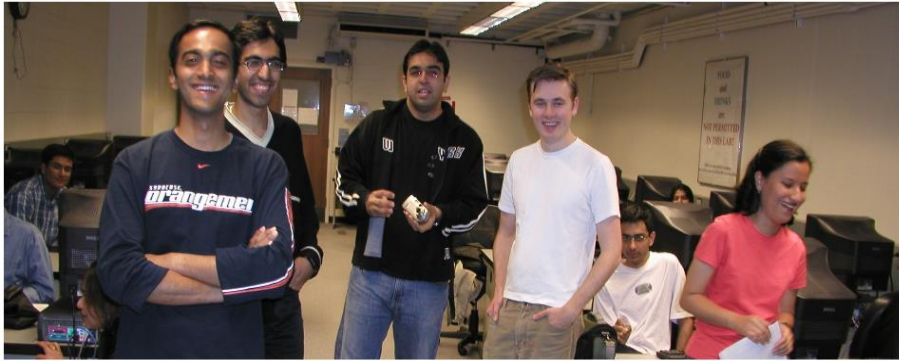
- HyperText Markup Language (HTML)
 - Web of linked documents
 - Unlimited scope of information content
- HyperText Transfer Protocol (HTTP)
 - Universal access
 - HTTP is a "request-response" protocol specifying that a client will open a connection to server then send request using a very specific format. Server will then respond and close connection.
- Graphical Browser Client
 - Sophisticated rendering makes authoring simpler
- HTML File Server
 - Using HTTP, Interprets request, provides appropriate response, usually a file in HTML format

CSE686 - Internet Programming

Program Models for the Web

Revised: 09/12/2013 16:18:57

[Home](#) [Courses ▾](#) [Core Tech ▾](#) [Directories ▾](#) [Research ▾](#) [Notices](#) [EECS ▾](#) [Resources ▾](#) [Projects ▾](#) [Blog ▾](#)



Each Class starts with a Lecture and Ends with an in-class Assignment

Resources:

[Table of Contents](#)


```
http://www.lcs3.syr.edu/faculty/fawcett/handouts/Webpages/CSE686.htm - Original Source
File Edit Format
1 <!DOCTYPE html>
2 <html>
3 <head>
4 <!--
5 - CSE681.htm - Software Modeling and Analysis course page
6 - ver 1.3 - 18 Aug 2011
7 - Jim Fawcett, Syracuse University
8 -->
9 <meta http-equiv="content-type" content="text/html;charset=UTF-8" />
10 <meta name="description" content="Software Engineering course notes. Code Samples. Software Links" />
11 <meta name="keywords" content="Lecture, Notes, Code, Syracuse,University" />
12 <meta name="Author" content="Jim Fawcett" />
13 <meta name="Author" content="James Fawcett" />
14 <meta http-equiv="Content-Type" content="text/html;charset=utf-8" />
15 <title>CSE686-IP</title>
16 <script type="text/javascript" src="js/jquery-1.6.2.min.js"></script>
17 <script type="text/javascript" src="js/TopMenu.js"></script>
18 <script type="text/javascript" src="js/Fallback.js"></script>
19 <link rel="stylesheet" href="css/TopLevel.css?v=1.0" />
20 <link rel="stylesheet" href="css/CourseSupplements.css?v=1.0" />
21 <link rel="stylesheet" href="css/Fallback.css?v=1.0" />
22 </head>
23 <body>
24 <header>
25 <div class="container">
26 <div id="topleft">
27 Course Notes
28 </div>
29 <div id="topright">
30 Computer Engineering
31 </div>
32 </div>
33 <hgroup id="pagetitle">
34 <h1 id="title">CSE686 - Internet Programming</h1>
35 <h3 id="subtitle">Program Models for the Web</h3>
36 <div id="pagedate" class="center">
37 <script type="text/javascript">
38 document.write("Revised: " + document.lastModified)
39 </script>
40 </div>
41 </hgroup>
42 </header>
43
44 <!-- Site navigation menus are generated with Javascript, using a Fallback link to a Navigation page -->
45 <nav>
46 <div id="nav">
47 <div id="remove">
```

Table of Contents

Extension - Cascading Style Sheets

- Help to separate content from presentation
- Defines styles using C-structure like notation:
 - `body { font-family: tahoma; font-size: medium; }`
 - may apply to specific tags, as above
 - `.notice { color: red; font-size: large; }`
 - defines a class called notice
 - `...`
 - by default can be applied to any tag

Extension - JavaScript

- Help separate content from behavior
- Support dynamic presentation using JavaScript code sent to the browser
 - `<script type="text/javascript" src="js/TopMenu.js"></script>`

```

1  /*****
2  * CSE686-LectNav.js - Lecture page navigation code for SMA
3  * ver 1.1 - 15 Aug 2011
4  * Jim Fawcett, Syracuse University
5  *****/
6
7  $(document).ready(function() {
8      $("nav").html(
9          "<hr class='menubegin' /\>\
10         <div id='menu'\>\
11             <ul>\
12                 <li aria-haspopup='false'\><a href='.././Webpages/FawcettHome.htm'\>Home</a>\
13                 </li>\
14                 <li aria-haspopup='false'\><a href='.././webpages/cse686.htm'\>Course</a></li>\
15                 <li aria-haspopup='false'\><a href='.././webpages/cse686.htm#syllabus'\>Syllabus</a></li>\
16                 <li aria-haspopup='true'\><a href='#'\>Lectures &#9662</a>\
17                     <ul class='active'\>\
18                         <li aria-haspopup='false'\><a href='cse686codeL1.htm'\>L1: Intro</a></li>\
19                         <li aria-haspopup='false'\><a href='cse686codeL2.htm'\>L2: Classic</a></li>\
20                         <li aria-haspopup='false'\><a href='cse686codeL3.htm'\>L3: Scripts</a></li>\
21                         <li aria-haspopup='false'\><a href='cse686codeL4.htm'\>L4: Asp.Net</a></li>\
22                         <li aria-haspopup='false'\><a href='cse686codeL5.htm'\>L5: MVC1</a></li>\
23                         <li aria-haspopup='false'\><a href='cse686codeL6.htm'\>L6: MVC2</a></li>\
24                         <li aria-haspopup='false'\><a href='cse686codeL7.htm'\>L7: XML</a></li>\
25                         <li aria-haspopup='false'\><a href='cse686codeL8.htm'\>L8: RelData</a></li>\
26                         <li aria-haspopup='false'\><a href='cse686codeL9.htm'\>L9: WebSrv</a></li>\
27                         <li aria-haspopup='false'\><a href='cse686codeL10.htm'\>L10: WPF</a></li>\
28                         <li aria-haspopup='false'\><a href='cse686codeL11.htm'\>L11: Clnts</a></li>\
29                         <li aria-haspopup='false'\><a href='cse686codeL12.htm'\>L12: State</a></li>\
30                         <li aria-haspopup='false'\><a href='cse686codeL13.htm'\>L13: ADO</a></li>\
31                         <li aria-haspopup='false'\><a href='cse686codeL14.htm'\>L14: Fn1Prj</a></li>\
32                         <li aria-haspopup='false'\><a href='cse686codeL15.htm'\>L15: Fn1Prj</a></li>\
33                         <li aria-haspopup='false'\><a href='#'\></a>&nbsp;⋮</li>\
34                     </ul>\
35                 </li>\
36                 <li aria-haspopup='true'\><a href='#'\>Labs &#9662</a>\
37                     <ul class='active'\>\
38                         <li aria-haspopup='false'\><a href='../assignments/assignment1.htm'\>L1: Home</a></li>\
39                         <li aria-haspopup='false'\><a href='../assignments/assignment2.htm'\>L2: Nav</a></li>\
40                         <li aria-haspopup='false'\><a href='../assignments/assignment3.htm'\>L3: Asp.Net</a></li>\
41                         <li aria-haspopup='false'\><a href='../assignments/assignment4.htm'\>L4: Design</a></li>\

```

Table of Contents

Programming The Web

[Table of Contents](#)

Web Programming Model

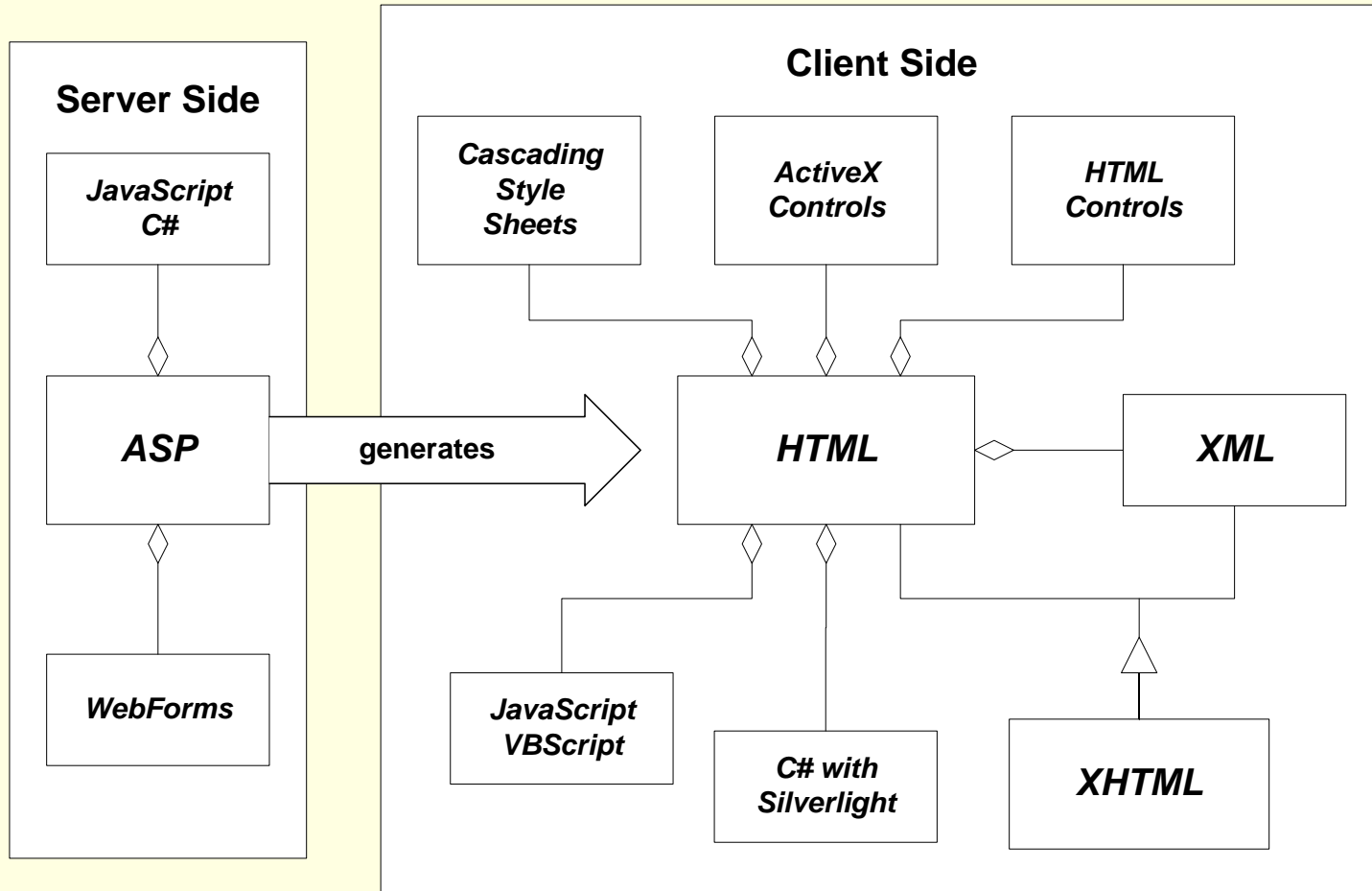
- Packaged functionality
 - Web server supports default and user supplied controls
- Dynamic content display
 - ASP, JSP generates HTML using server data
 - Browser interprets client side scripts
- Machine-to-Machine
 - Web services provide RPC interface

Programming the Web

- Client-Side Programming
 - JavaScript
 - Dynamic HTML
 - Can modify html document using scripts sent from server and interpreted by client.
 - .Net controls - need permissions
- Server-Side Programming
 - ASP script
 - Server components
 - C# code-behind
 - ADO
 - Web controls used on ASPX pages
 - Web services

[Table of Contents](#)

Web Programming - Language Model



Programming the Web

Server-Side Code

- What is server-side code?
 - Software that runs on the server, not the client
 - Receives input from
 - URL parameters
 - HTML form data
 - Cookies
 - HTTP headers
 - Can access server-side databases, e-mail servers, files, mainframes, etc.
 - Dynamically builds a custom HTML response for a client

Traditional HTML Serving Model

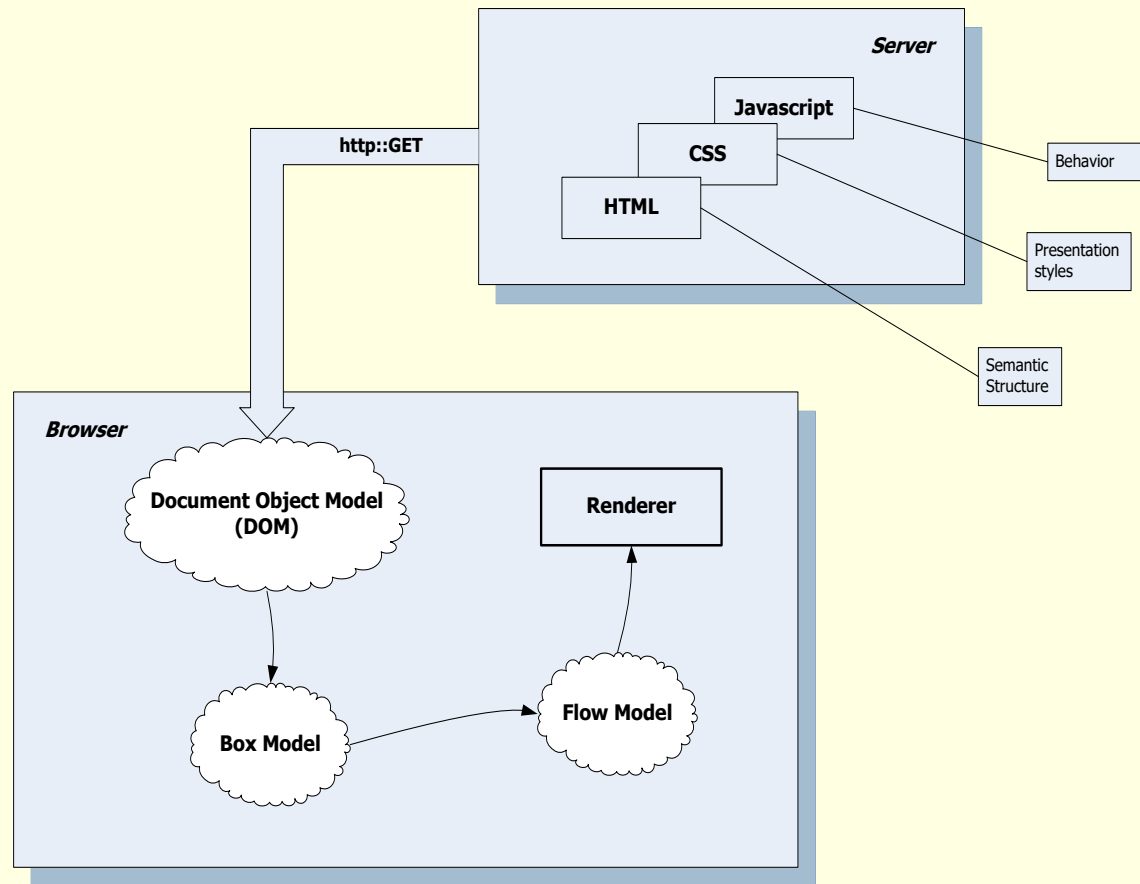


Table of Contents

ASP Dynamic Serving Model

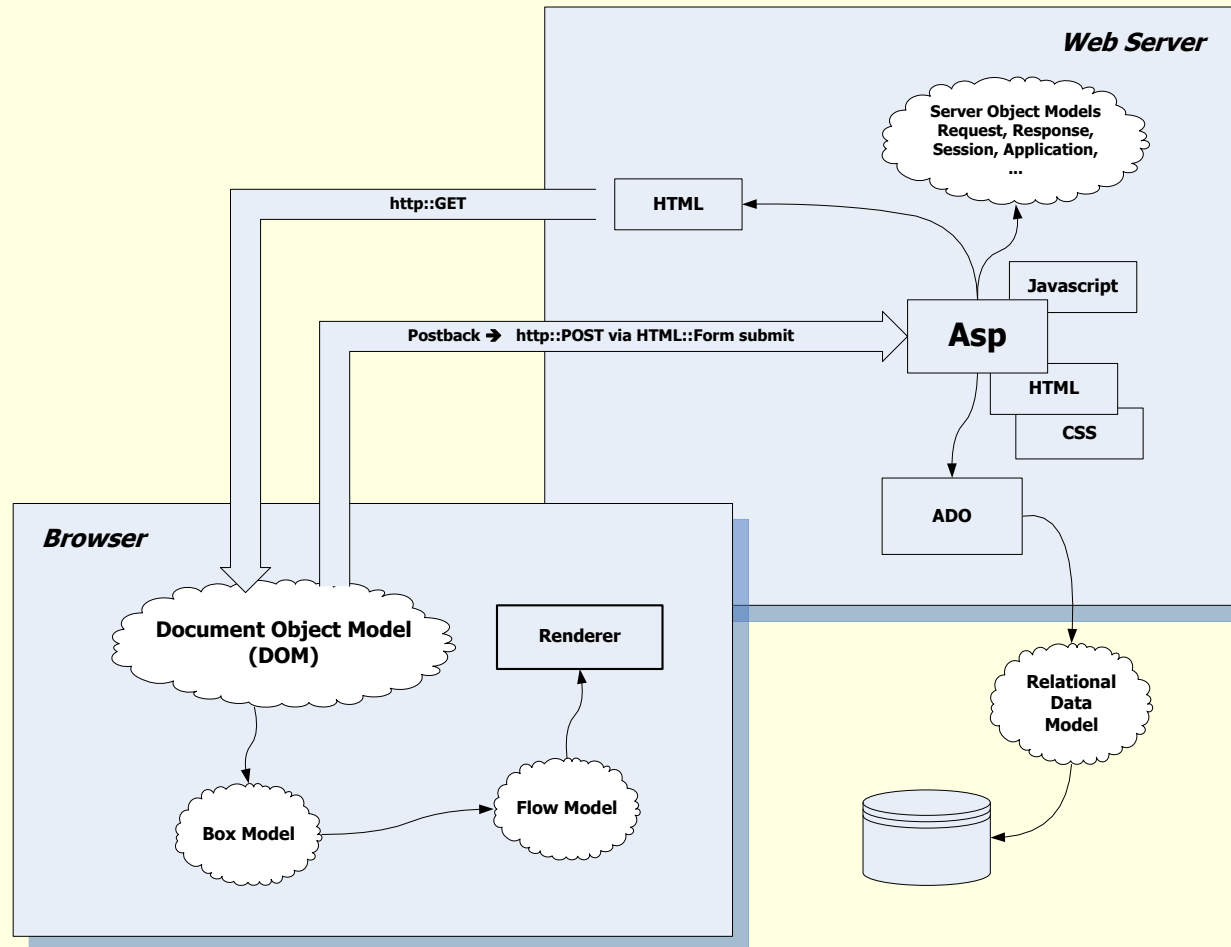


Table of Contents

ASP.NET Serving Model

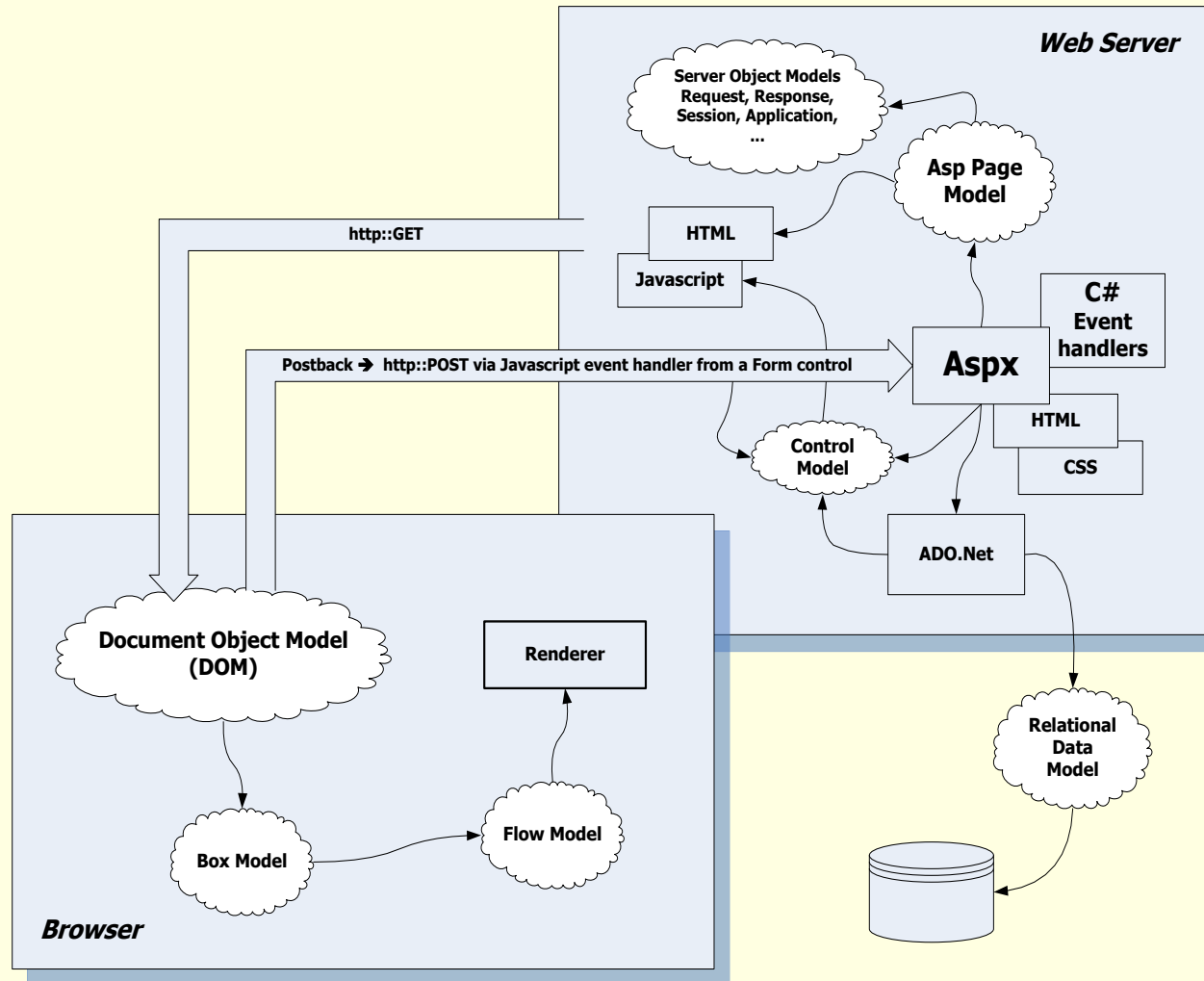


Table of Contents

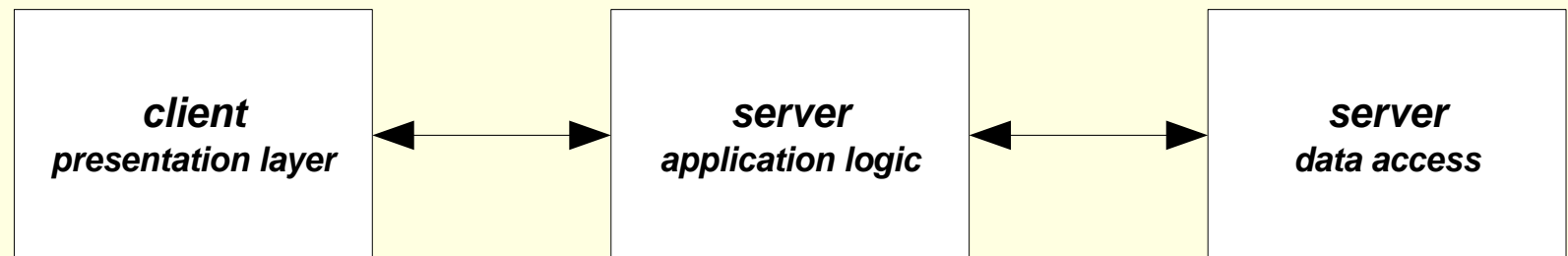
Programming the Web

Server-Side Code

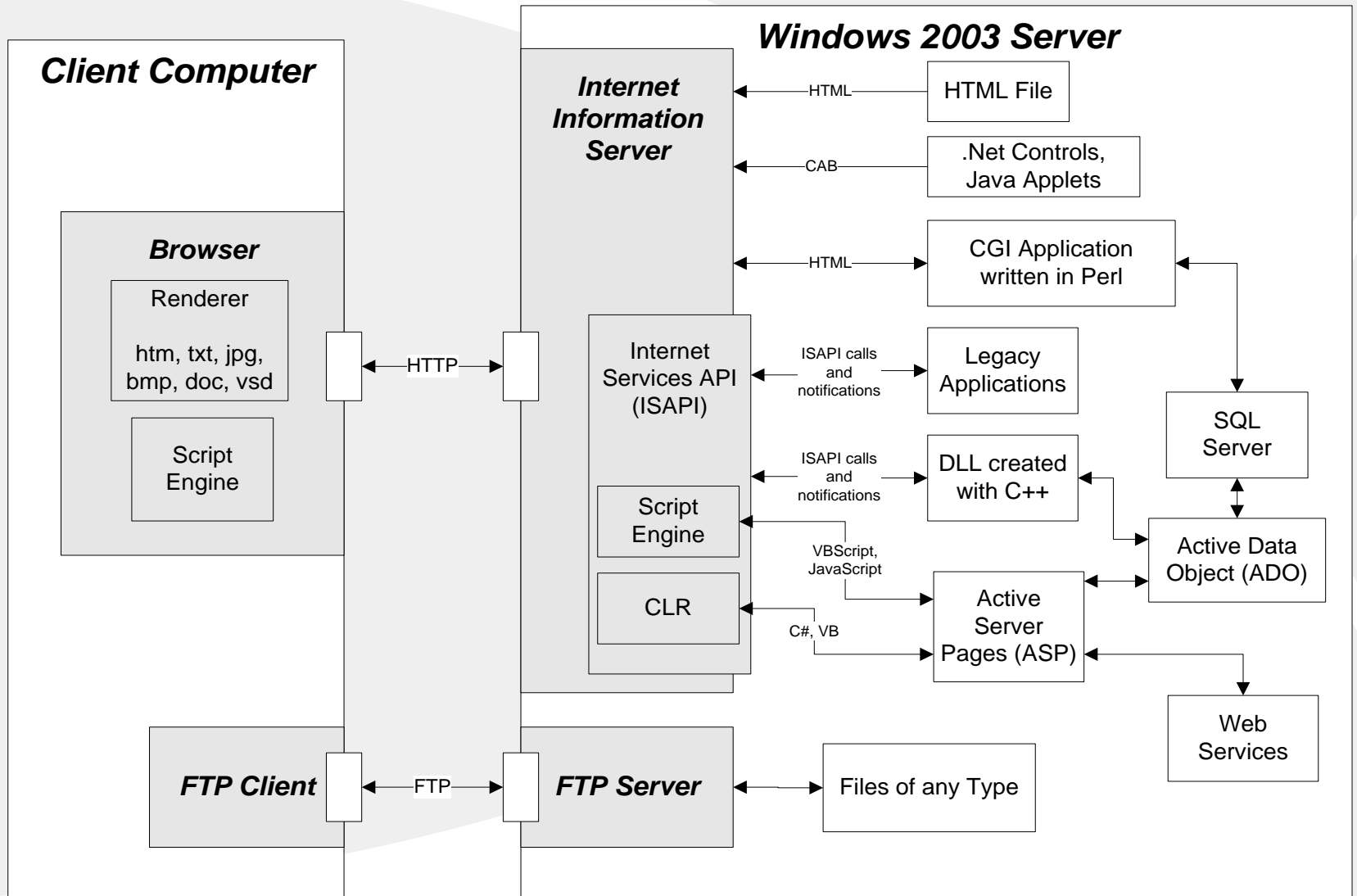
- Why server-side code?
 - Accessibility
 - You can reach the Internet from any browser, any device, any time, anywhere
 - Manageability
 - Does not require distribution of application code
 - Easy to change code
 - Security
 - Source code is not exposed
 - Once user is authenticated, can only allow certain actions
 - Scalability
 - Web-based 3-tier architecture can scale out

Three Tier Architecture

- Client Tier
 - Presentation layer
 - Client UI, client-side scripts, client specific application logic
- Server Tier
 - Application logic, server-side scripts, form handling, data requests
- Data Tier
 - Data storage and access



Client/Server - Current Web Model



.Net Controls

- *The model of previous slide is very powerful!*
 - A browser that knows nothing about some sophisticated server-side processing can take advantage of that by downloading a .Net control that encapsulates all the intelligence necessary to work with the server.
 - Similarly, a browser can be given new processing capabilities, simply by loading a local web page that contains controls with the desired abilities.
 - Note that web page scripts do the same thing, only not quite so efficiently, and often with limitations on processing capabilities.

Displaying ActiveX Controls on a Web Page

Here is an example of an object tag and attributes for inserting a control on a Web page.

```
<OBJECT CLASSID="clsid:FC25B780-75BE-11CF-8B01-444553540000"  
  CODEBASE="/ie/download/activex/iechart.ocx" ID=chart1  
  WIDTH=400 HEIGHT=200  
  ALIGN=center HSPACE=0 VSPACE=0  
>  
  <PARAM NAME="BackColor" value="#ffffff">  
  <PARAM NAME="ForeColor" value="#0000ff">  
  <PARAM NAME="url" VALUE="/ie/controls/chart/mychart.txt">  
</OBJECT>
```

Run dirControl Object - Microsoft Internet Explorer

File Edit View Favorites Tools Help

← Back → Search Favorites History

Address C:\SU\cse791DO\CODE\dirControl\dirControl.html

Using dirControl Object in VBScript

This HTML page contains a VBScript segment that uses the dirControl object.
VBScript can communicate only with an IDispatch interface.

Directory Contents:

- .
- ..
- ATLclient.cpp
- client
- CurrDir.cpp
- CurrDir.h
- CurrDir.rgs
- Debug
- dirControl.aps
- dirControl.cpp
- dirControl.def
- dirControl.dsp
- dirControl.dsw
- dirControl.h
- dirControl.html
- dirControl.idl
- dirControl.ncb
- dirControl.opt
- dirControl.plg
- dirControl.rc
- dirControl.tlb
- dirControlps.def
- dirControlps.mk
- dirControl_i.c
- dirControl_p.c
- dlldata.c
- fileInfo
- FILEINFO.CPP

```

dirControl.html - Notepad
File Edit Format Help
<HTML>
<HEAD><TITLE>Run dirControl object</TITLE>
<style type="text/css">
  a:link {color: blue}           <!-- Cascading style sheet again -->
  a:visited {color: darkblue}   <!-- set link colors -->
  a:active {color: darkorange}
  body { color: darkred; background: yellow; } <!-- page colors -->
  body { margin-left: 10%; margin-right: 10%; } <!-- page margins -->
  h1 { margin-left: -6%; } <!-- pos of heading -->
  h2 { margin-left: -3%; } <!-- pos smaller one -->
</style>
</HEAD>
<BODY>
<center>
<H1>Using dircontrol object in vbscript</H1>
This HTML page contains a VBScript segment that uses the dirControl object.
<br>
VBScript can communicate only with an IDispatch interface.
</center>
<OBJECT ID="dirControl"
CLASSID="CLSID:B5F2D436-8933-4D7E-98D6-E93EA600B42D">
</OBJECT>
<pre>
<SCRIPT LANGUAGE="VBScript">
dirControl.SetCurrDir("c:\su\cse791do\code\dirControl")
Document.writeln ""
Document.write "directory Contents:"
Document.write dirControl.GetDirContents()
</SCRIPT>
</pre>
</BODY>
</HTML>

```

Done

Start

My Computer

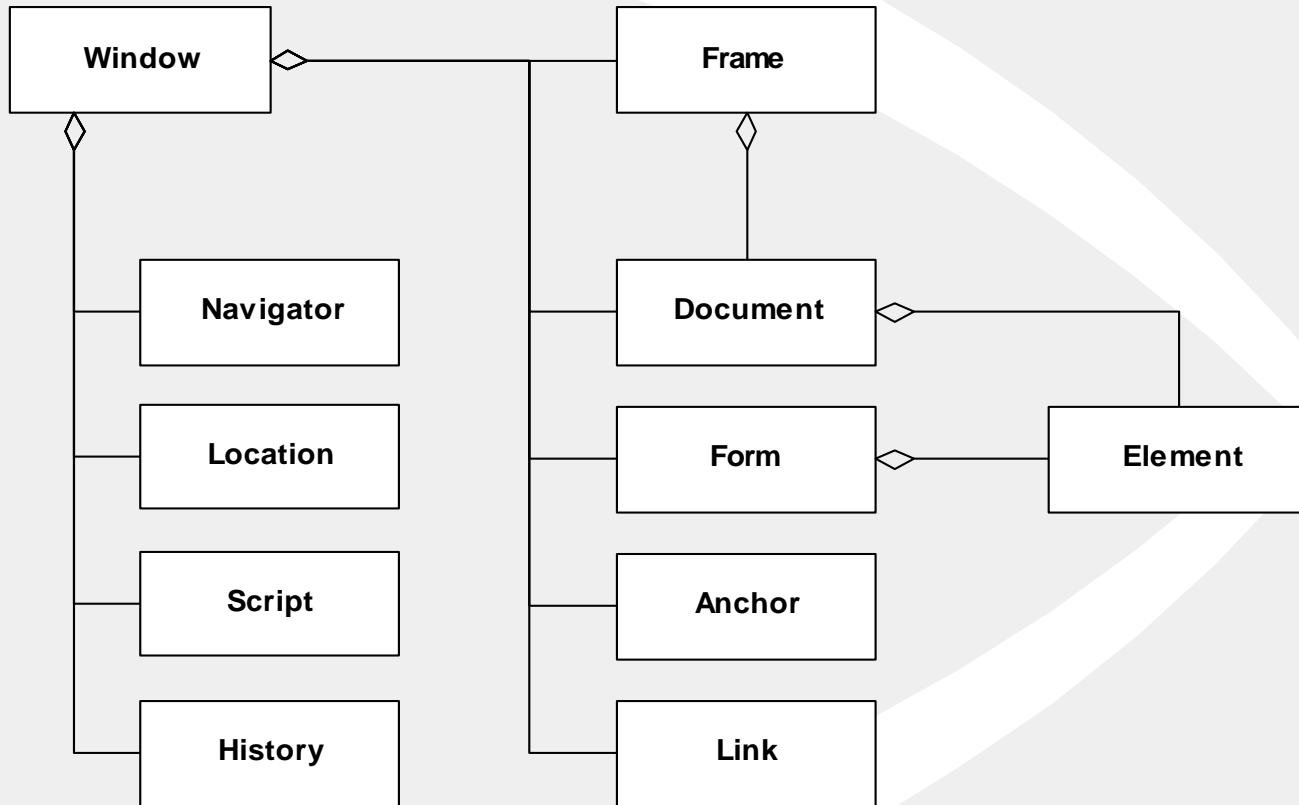
4:20 PM

Table of Contents

Browser Object Model

- Window
 - browser window
- Document
 - current HTML page
- Form
 - a form holds controls
 - often used to submit data to server
- Frame
 - frame in browser window
- Location
 - Location of current web page
 - URL, domain name, port, path, ...
- Navigator
 - Browser, itself
- History

Browser Object Model



Some Examples

- Basic HTML pages
 - Example #1

Server Object Model

- Application Object
 - Data sharing and locking across clients
- Request Object
 - Extracts client data and cookies from HTTP request
- Reponse Object
 - Send cookies or call Write method to place string in HTML output
- Server Object
 - Provides utility methods
- Session Object
 - If browser supports cookies, will maintain data between page loads, as long as session lasts.

Server Components

skip to Security Issues

- Ad Rotator - rotates advertisements
- Browser Capabilities - determines type
- Database Access
 - Active Data Objects (ADO) provide common interface to a variety of data sources
- Content Linking
 - Creates list of web pages
- File Access Component
 - Provides access to server files from scripts

Server Object Model

**Application
Object**

**Request
Object**

**Response
Object**

**Server
Object**

**Session
Object**

**File Access
Component**

**Ad Rotator
Component**

**Browser Capabilities
Component**

**Content Linker
Component**

**Active Data Object
Component**

Server Side Programming with ASP

- An Active Server Page (ASP) consists of HTML and script.
 - HTML is sent to the client "as-is"
 - Script is executed on a server to dynamically generate more HTML to send to the client.
 - Since it is generated dynamically, ASP can tailor the HTML to the context in which it executes, e.g., based on time, data from client, current server state, etc.

```
test0.asp - Notepad
File Edit Format Help
<!-------
test.asp - demonstrate server side scripting
----->

<%@ Language=JScript %>
<html>
  <head>
    <title>Looping Test</Title>
  </head>

  <body bgcolor="burlywood" Text="darkslateblue"
    TopMargin="10" LeftMargin="10">
    <!-- display header -->
    <font size="4">
      <b>Looping with ASP</b>
    </font>
    <br />
    <hr size="1" color="#000000">

    <!-- looping with a for loop -->
    <%
      var i;
      for(i=1; i<8; i++) { %>
        <font size=<% =i %>>hello CSE 691/891<br />
      <% } %>
    <%
      var i;
      for(i=1; i<8; i++) { %>
        <font size=<% =8-i %>>hello CSE 891/691<br />
      <% } %>
    </hr>
  </body>
</html>
```

The screenshot shows a Microsoft Internet Explorer browser window titled "Looping Test - Microsoft Internet Explorer". The address bar shows "http://bismark/test0.asp". The main content area displays the output of an ASP page, which is a list of "hello" messages with varying font sizes and colors. The messages are: "hello CSE 691/891" (8 times), "hello CSE 891/691" (8 times), and "hello CSE 891/691" (8 times). The font sizes increase from 4 to 8, and the colors change from darkslateblue to burlywood.

Overlaid on the browser is a Notepad window titled "test0.asp - Notepad". The code in the Notepad window is as follows:

```
<!--
test.asp - demonstrate server side scripting
-->

<%@ Language=JScript %>
<html>
<head>
<title>Looping Test</Title>
</head>

<body bgcolor="burlywood" Text="darkslateblue"
TopMargin="10" LeftMargin="10">
<!-- display header -->
<font size="4">
<b>Looping with ASP</b>
</font>
<br />
<hr size="1" color="#000000">

<!-- looping with a for loop -->
<%
var i;
for(i=1; i<8; i++) { %>
<font size=<% =i %>>hello CSE 691/891<br />
<% } %>
<%
var i;
for(i=1; i<8; i++) { %>
<font size=<% =8-i %>>hello CSE 891/691<br />
<% } %>
</hr>
</body>
</html>
```

[Table of Contents](#)

Looping Test - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Back Forward Stop Home Search Favorites History

Address http://bismark/test0.asp

Looping with ASP

hello CSE 691/891
hello CSE 691/891
hello CSE 691/891
hello CSE 691/891
hello CSE 691/891
hello CSE 691/891
hello CSE 891/691
hello CSE 891/691
hello CSE 891/691
hello CSE 891/691
hello CSE 891/691
hello CSE 891/691
hello CSE 891/691

test0[1] - Notepad

File Edit Format Help

```
test.asp - demonstrate server side scripting

<html>
<head>
  <title>Looping Test</title>
</head>
<body bgcolor="burlywood" Text="darkslateblue"
  TopMargin="10" LeftMargin="10">
  <!-- display header -->
  <font size="4">
    <b>Looping with ASP</b>
  </font>
  <br />
  <hr size="1" color="#000000">

  <!-- looping with a for loop -->
    <font size=1>hello CSE 691/891<br />
    <font size=2>hello CSE 691/891<br />
    <font size=3>hello CSE 691/891<br />
    <font size=4>hello CSE 691/891<br />
    <font size=5>hello CSE 691/891<br />
    <font size=6>hello CSE 691/891<br />
    <font size=7>hello CSE 691/891<br />
    <font size=7>hello CSE 891/691<br />
    <font size=6>hello CSE 891/691<br />
    <font size=5>hello CSE 891/691<br />
    <font size=4>hello CSE 891/691<br />
    <font size=3>hello CSE 891/691<br />
    <font size=2>hello CSE 891/691<br />
    <font size=1>hello CSE 891/691<br />

  </hr>
</body>
</html>
```

Done

Start | Internet Explorer | Outlook | ... | 4:09 PM

Table of Contents

Server Side Programming with Asp.Net

- An Asp.Net application consists of:
 - Design Time:
 - A form with web controls
 - C# code behind event handlers
 - Run-Time:
 - Form is translated into an HTML form
 - Web controls become HTML elements with Javascript event handlers that postback to the server.
 - Asp.Net model makes control data available as properties of a Page class, transported from browser to server in a "hidden view-state".

ISAPI - Server Side Extensions

- Server Extensions work like CGI scripts to provide server-side processing, but they are DLLs, which reside in the memory space of the HTTP server.
- This is an enormous performance advantage over CGI extensions which need to spawn a new process each time they are run.
- The extension DLL exports `HttpExtensionProc()`, which is called by IIS when the user request asks for the extension processing.
- Active Server Page (ASP) scripts and (Asp.Net) C# code are easier ways to accomplish the same thing. One would expect the ASP script or C# to be faster than CGI but slower than an ISAPI extension.

Using Controls and Applets

- We've already seen how to include an ActiveX control on a web page.
- Now let's see how to do that for a Java Applet:
 - Java Applet - Sprites

Including Java Applet

```
<applet code=sprites.class width=700 height=125>
<!-- registration code to disable the floating -->
<!-- sign within the program panel and 5-min connection -->
<param name=regcode value=99999999>
<!-- frames per second -->
<param name=fps value=15>
<!-- background color -->
<param name=bgcolor value="185 235 255">
<!-- border width (0 = no border) -->
<param name=border value=3>
<!-- link address when mouse is clicked -->
<param name=url value="http://www.thejmaker.com/">
<!-- number of menu items -->
<param name=total value=5>
<!-- control of individual sprites -->
<!--   where | seperates sprite filename, start x and y, speed x and y, wrap-around option -->
<param name=menu0 value="sprite0.gif|0 30|1 0|1">
<param name=menu1 value="sprite1.gif|150 35|-1 0|0">
<param name=menu2 value="sprite2.gif|150 45|-3 0|0">
<param name=menu3 value="sprite3.gif|0 40|2 0|1">
<param name=menu4 value="sprite4.gif|100 3|-5 0|0">
</applet>
```


Security Issues

■ Threats

- Data integrity
 - code that deletes or modifies data
- Privacy
 - code that copies confidential data and makes it available to others
- Denial of service
 - code that consumes all of CPU time or disk memory.
- Elevation of privilege
 - Code that attempts to gain administrative access

Syracuse University - Microsoft Internet Explorer

File Edit View Favorites Tools Help


Back Forward Stop Home Search Favorites History Print Folders

Address https://www.syr.edu:443/

Favorites

Add... Organize...

- Folder Dell
- Folder Links
- Folder Media
- Building ASP Pages
- DevelopMentor
- Enterprise Zone Book of t...
- Google
- Jim Fawcett, Ph.D.
- Jim's site
- Liquid Music Network
- MSDN Online
- MSN
- Radio Station Guide
- RealPlayer Home Page
- Sigma Xi, The Scientific ...
- Syracuse Online Central ...
- The Weather Channel - E...
- TIME Magazine Contents...
- WDVL The Basics of HT...
- Web Events
- Welcome to Liquid Audio
- WWW.ECS.SYR.EDU - ...
- Radio



contact
search
home

visitors
prospective students
alumni
parents & friends

inside su
academics
students
faculty & staff
chancellor

of interest
community involvement
research & technology
news
giving to su
athletics
web cams & videos

su info

Syracuse University
Syracuse, NY 13244
(315) 443-1870
© 2000 Syracuse University

s y r a c u s e u n i v e r s i t y


w e l c o m e to

Syracuse


UNIVERSITY

Welcome to Syracuse University


Here you will find the richness and diversity of programs and people that are moving us forward toward our vision as the nation's leading student-centered research university. I invite you to explore our site and return often. I am convinced you will be impressed with Syracuse University today and with its plans for an even better future.



Kenneth A. Shaw
Chancellor and President



Chancellor's Annual Address



Protections

- Least privilege rule:
 - Use the technology with the fewest capabilities that gets the job done.
- Digital signing
 - Who are you?
- Security zones
 - Trusted and untrusted sites
- Secure sockets layer (SSL)
- Transport layer security (TLS)
- Encryption

Extending The Web

Current Extensions

- Describe data with XML
- Extend HTML into XHTML
- Separate style from content with CSS
 - Cascading style sheets
 - Can be included from a file to give uniform style of pages and documents
- Document Object Model - DOM
 - Defines a scripting interface

The Extensible Web

- Some recent W3C Technologies
 - www.w3.org/2003/Talks/0521-BudapestW3CTrack-IH/6.html
 - www.w3.org/2003/Talks/0521-BudapestW3CTrack-IH/23.html

Areas of Exploration

- XML - Universal Data Services
- TVWeb - merger of features
- MathML - Mathematical Markup Language
- RDF - Resource Description Framework
- Accessibility - for the handicapped
- SMIL - Synchronized Multimedia Integration Language
- Internationalization
- Speech

People in the Web

- Web Development
 - Web server, HTTP
 - Tim Berners-Lee, Robert Cailiau
 - Mosaic web browser
 - Marc Andreessen
- Internet
 - TCP/IP protocol
 - Vinton Cerf, Robert Kahn
 - Internet flow control
 - Larry Roberts

References

- [World Wide Web Consortium](#)
 - Excellent Tutorial Papers, standards
- Source of several slides used here
 - [Mark Sapposnek](#)
- [webdev.htm](#)
 - Tutorials
 - Web developer's links
 - Web designer's links
 - Tech details links
- XHTML Black Book, Steven Holzner, Coriolis, 2000
 - Aging but comprehensive treatment of HTML, XHTML, JavaScript
- [Web Developers Virtual Library](#)
 - More tutorials

End of Presentation

[Table of Contents](#)