

Ultimate Extensible Distributed System

Jim Fawcett

CSE681 - Software Modeling & Analysis

Fall 2010

Nota Bene

- These notes I used in CSE686 - Internet Programming, a course I no longer teach.
- They are out of date, but much of the material is still relevant.

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](#)

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:

- 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, ...

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

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

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
 - 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

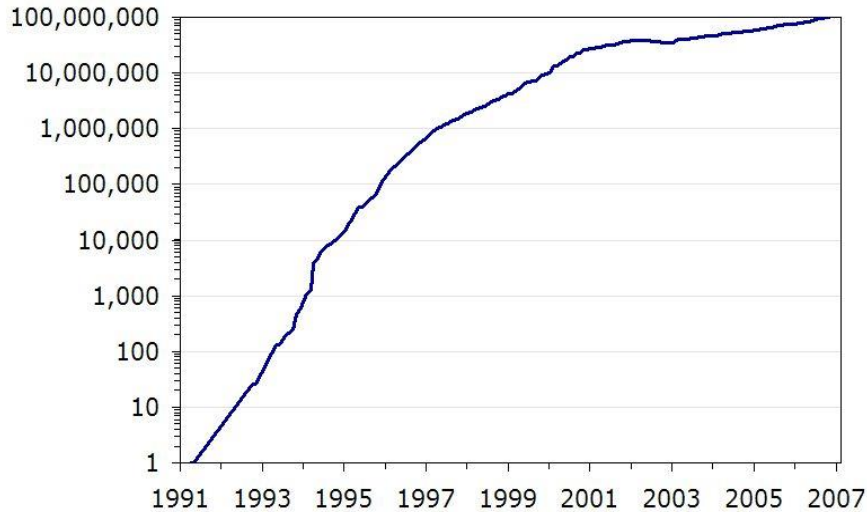
Web History

- 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.

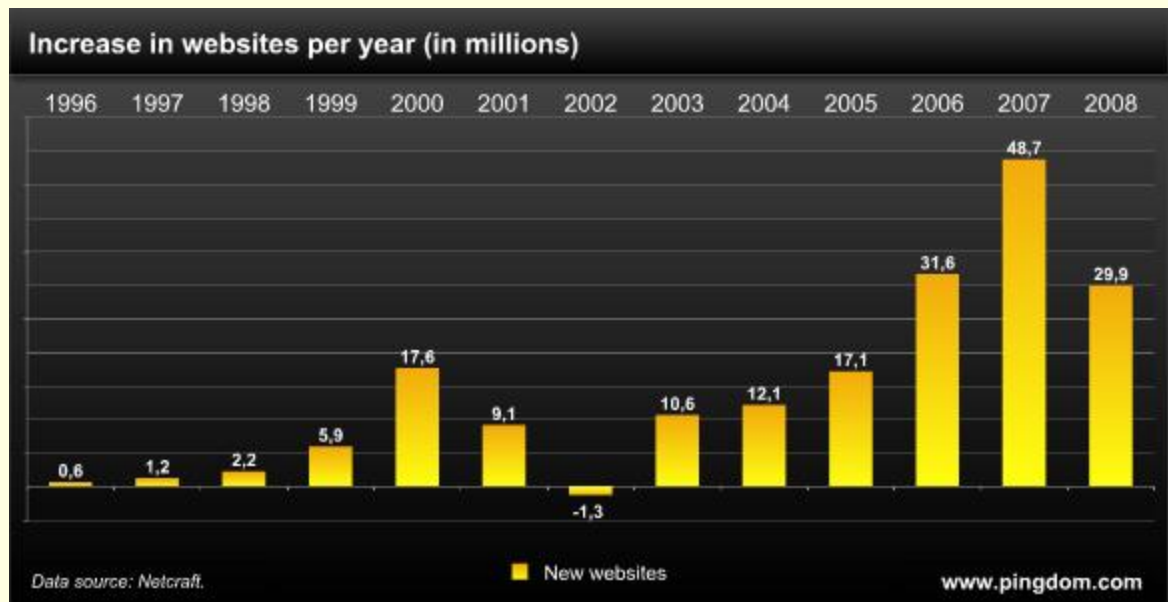


The **number of Internet websites** each year since the Web's founding.

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](#)

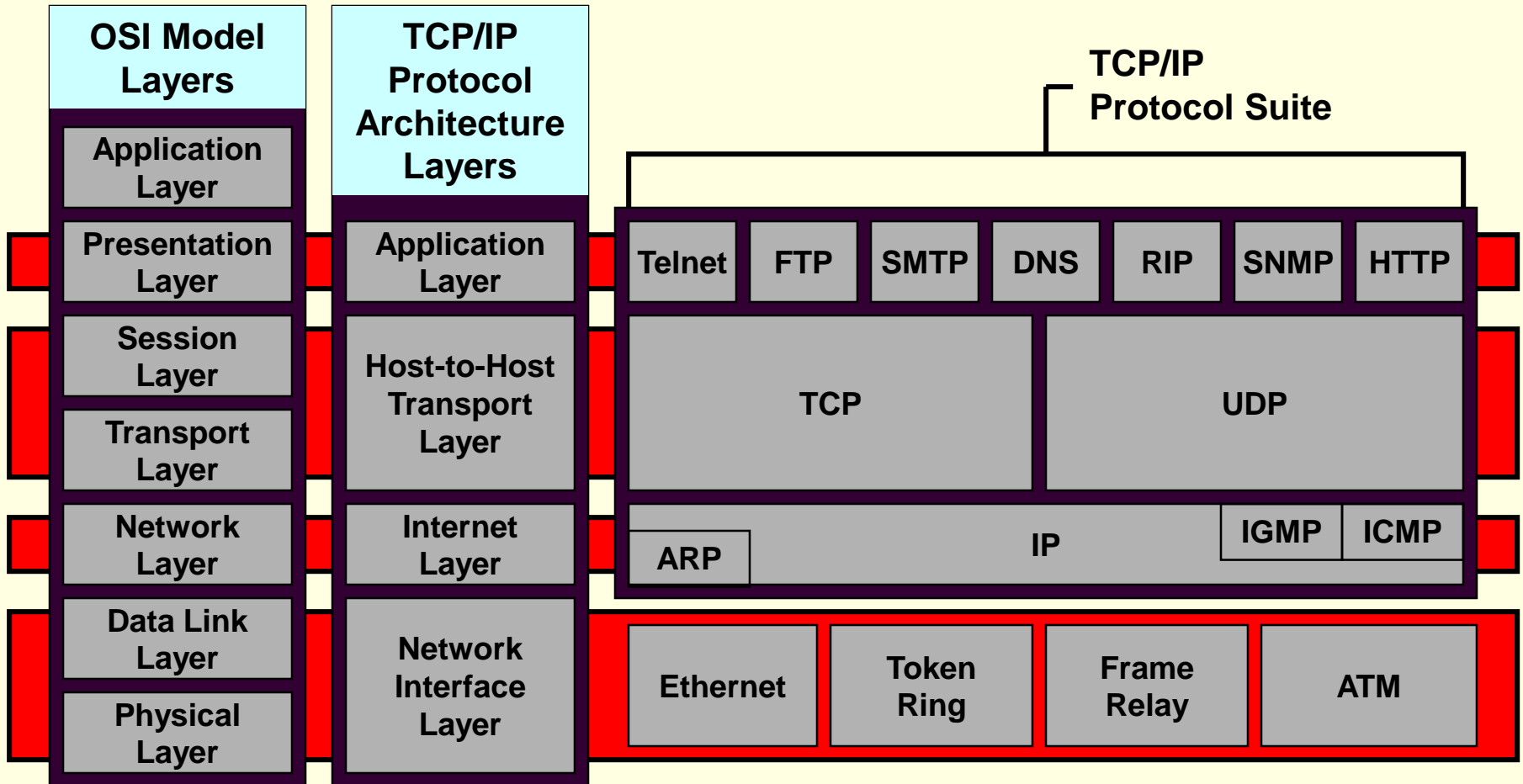
Web Technologies

[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 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,XIPý¼jà 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

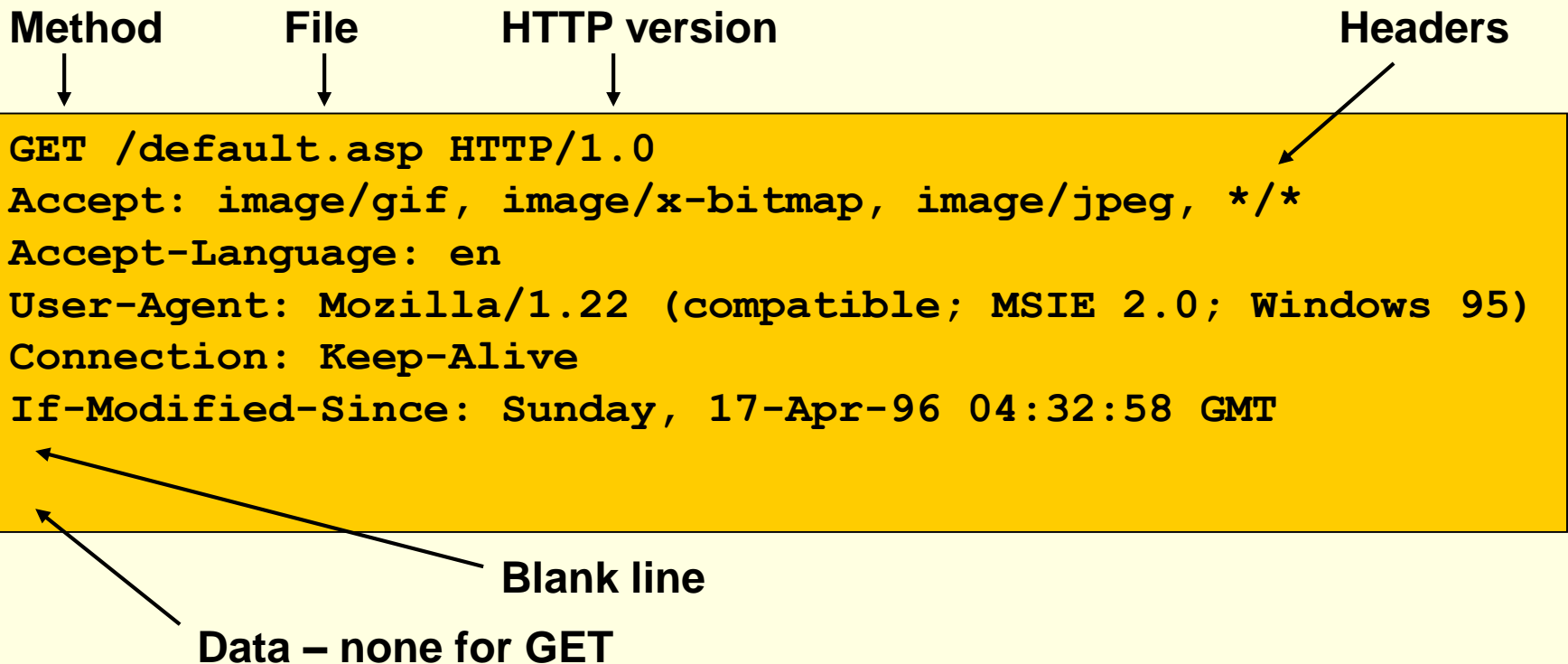
Typical HTTP Transaction

- 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

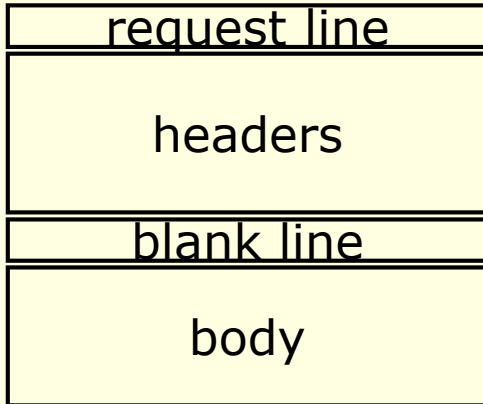


Multipurpose Internet Mail Extensions (MIME)

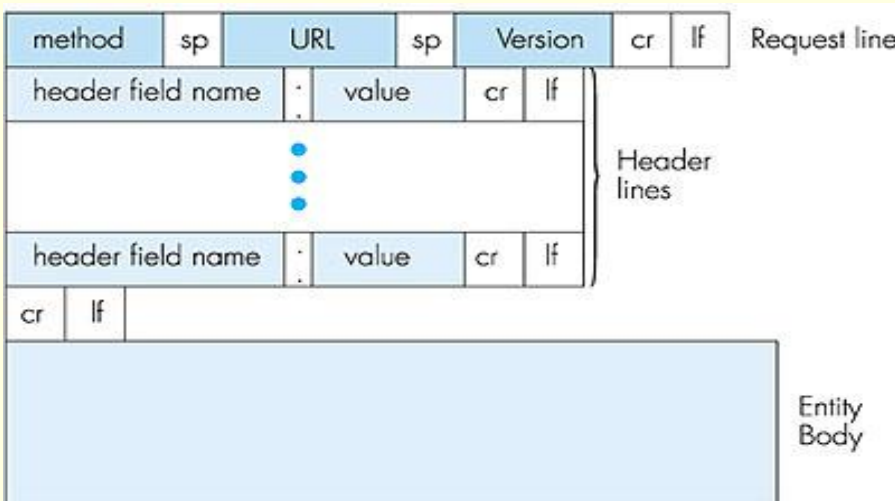
[skip to HTTP Response](#)

- 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

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

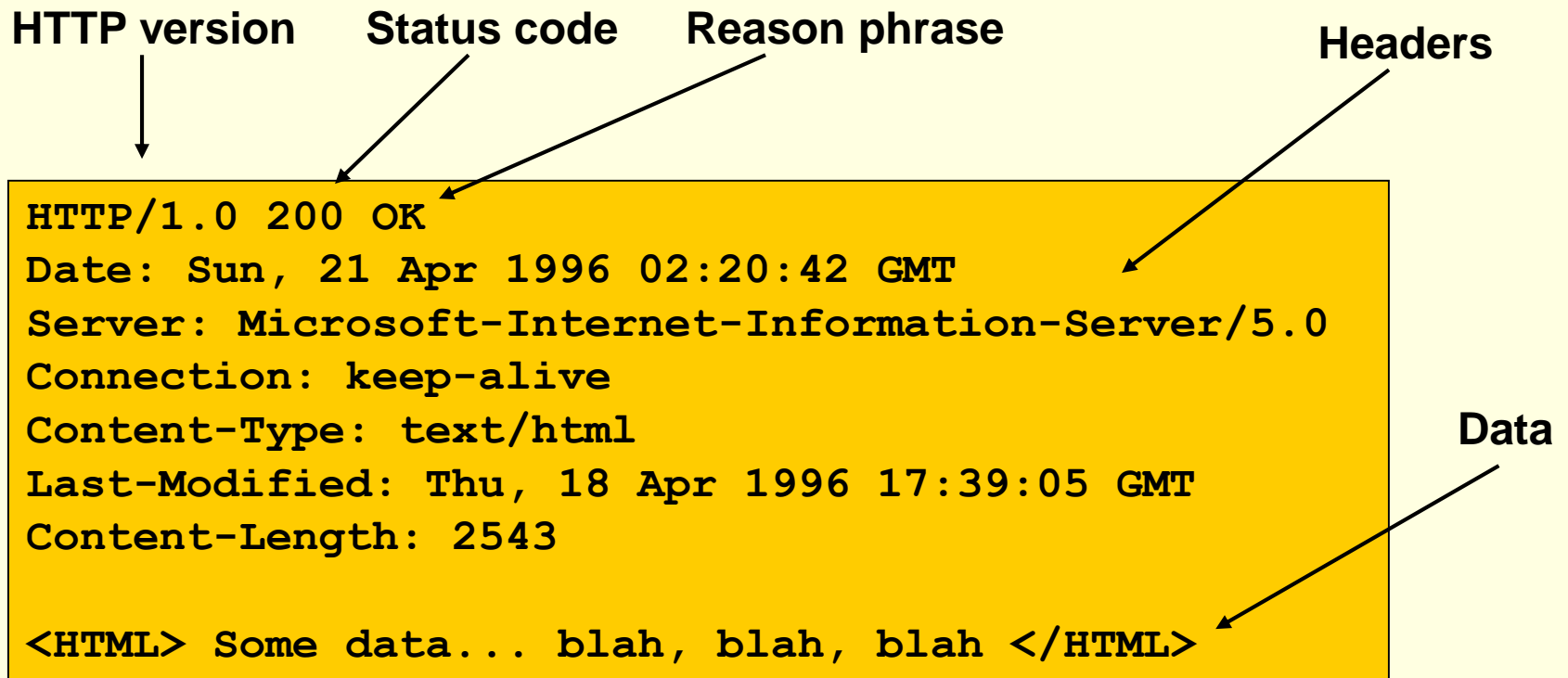
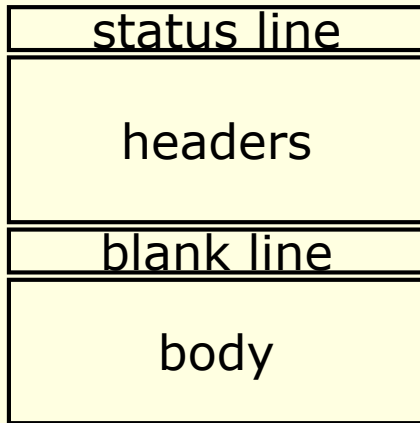


Table of Contents

Response Message



```

HTTP/1.1 200 OK
Date: Tue, 08 Oct 2002 00:31:35 GMT
Server: Apache/1.3.27 tomcat/1.0
Last-Modified: 7Oct2002 23:40:01 GMT
ETag: "20f-6c4b-3da21b51"
Accept-Ranges: bytes
Content-Length: 27723
Keep-Alive: timeout=5, max=300
Connection: Keep-Alive
Content-Type: text/html
    
```

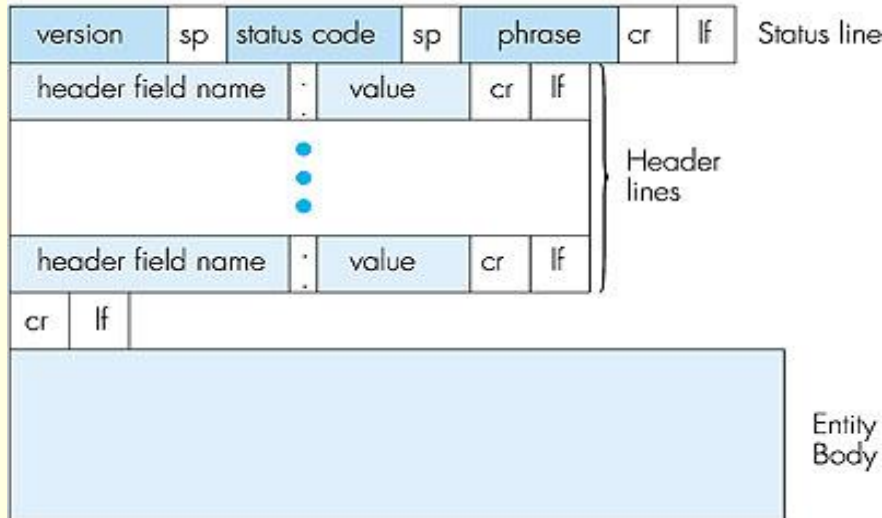


Table of Contents

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

Programming The Web

[Table of Contents](#)

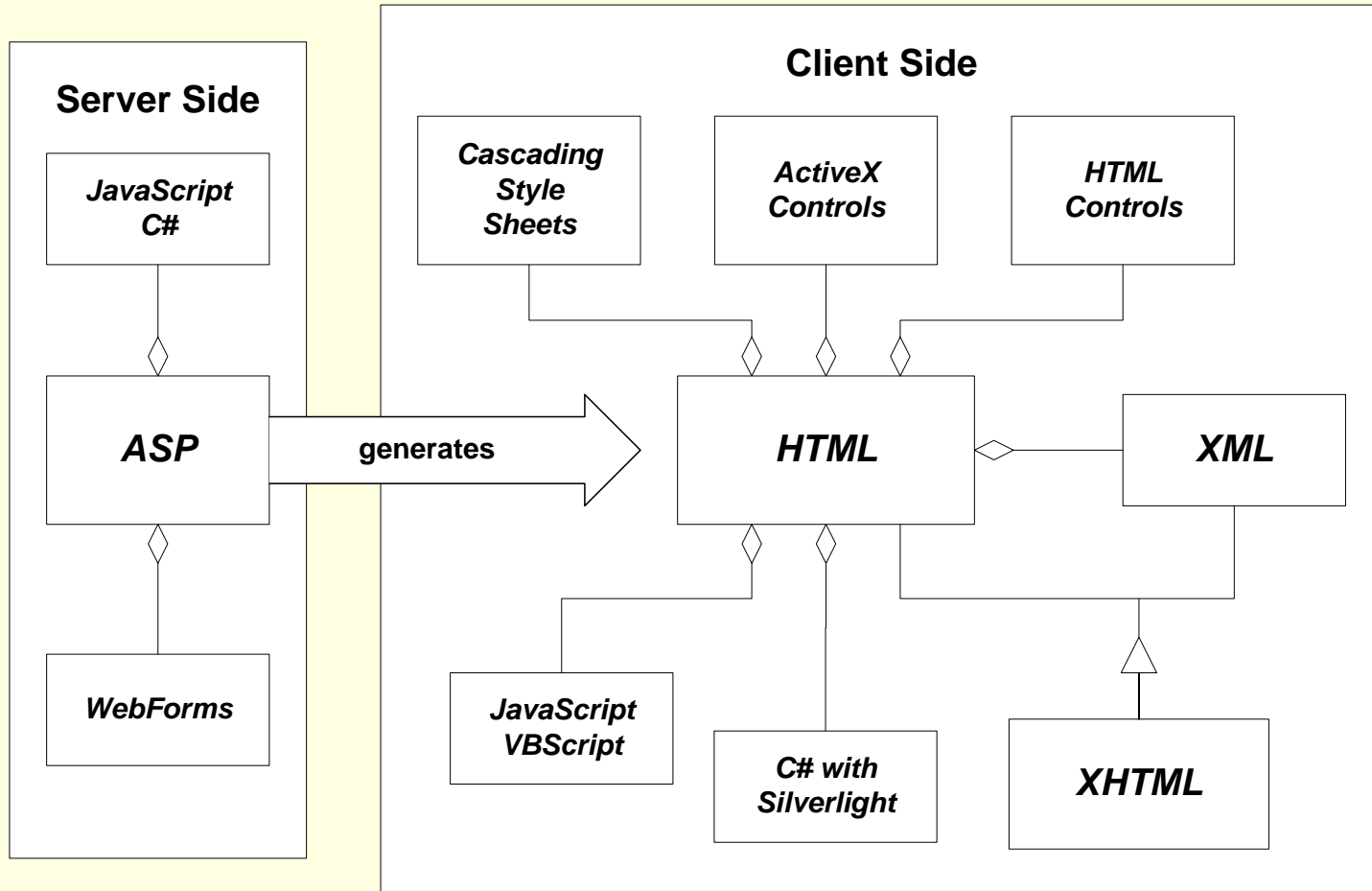
Web Programming Model

- 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
 - Can modify html document using scripts sent from server and interpreted by client.
 - Silverlight uses C# in embedded CLR in Browser plug-in
 - .Net controls, Java applets - need permissions
- Server-Side Programming
 - ASP script, C# code-behind
 - Server components
 - Session, Application, ADO, FileSystem, ...
 - Web controls used on ASPX pages
 - Web services

Web Programming - Language Model



[Table of Contents](#)

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, Web Services, etc.
 - Dynamically builds a custom HTML response for a client

Traditional HTML Serving Model

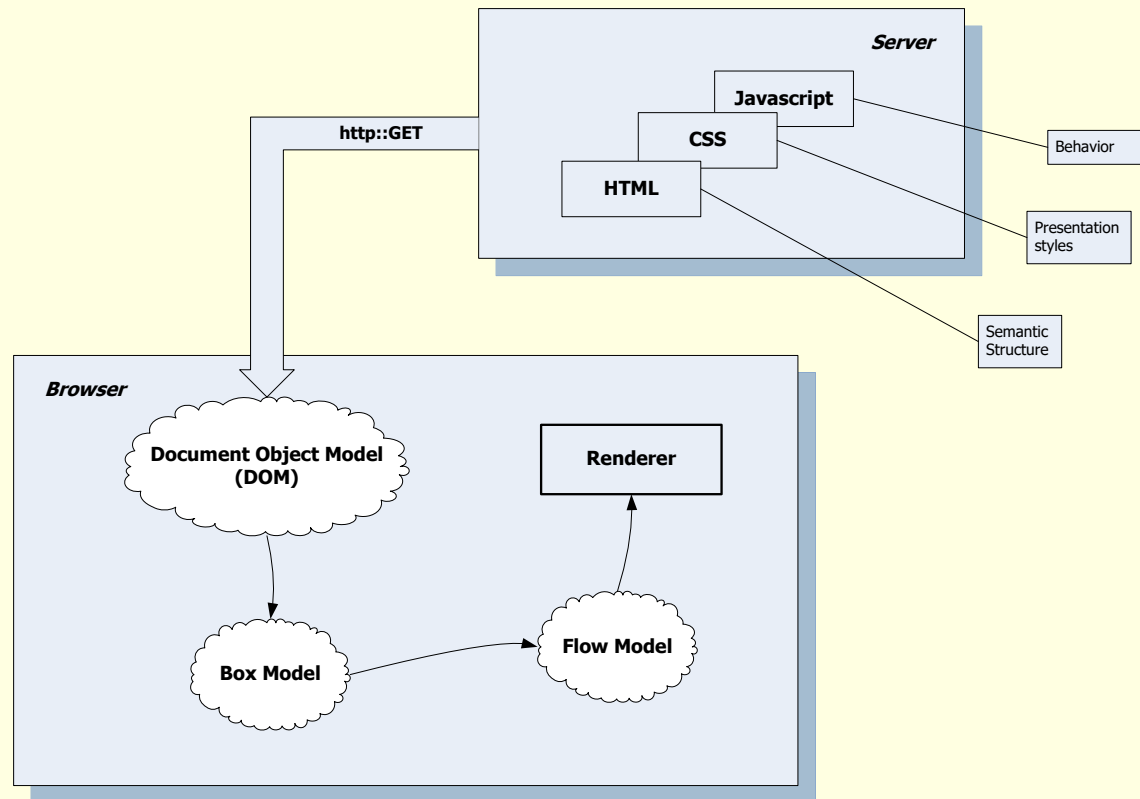
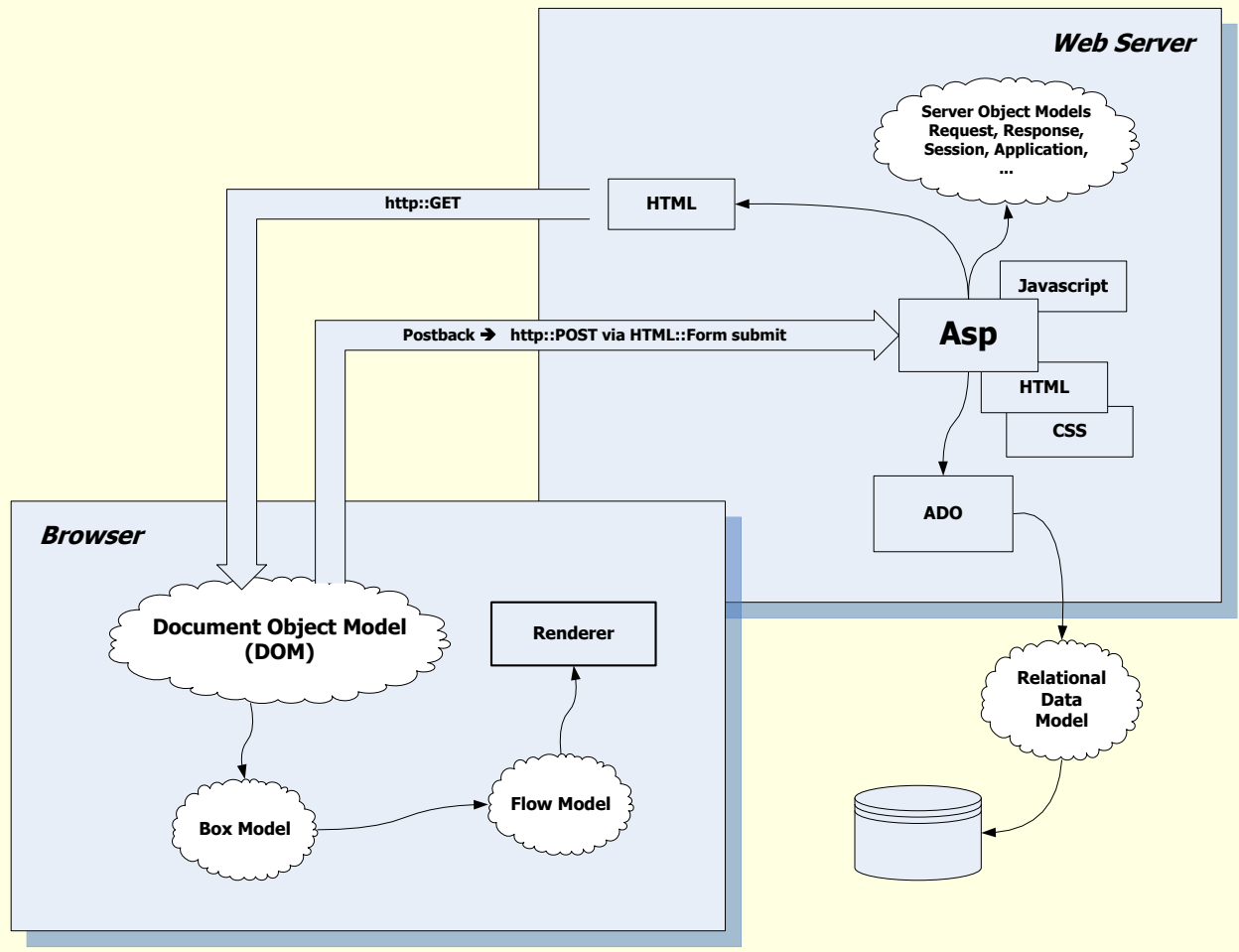


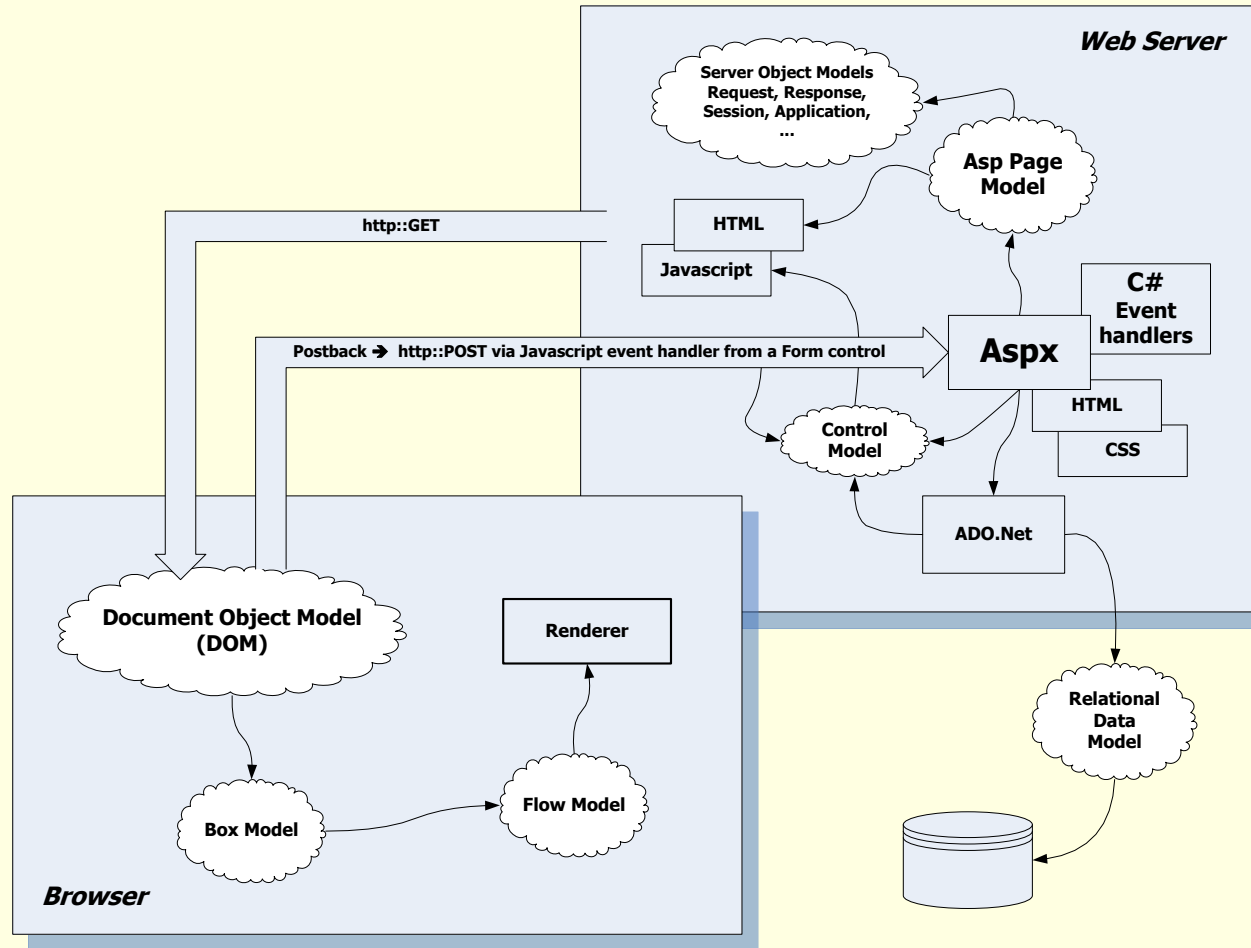
Table of Contents

ASP Dynamic Serving Model

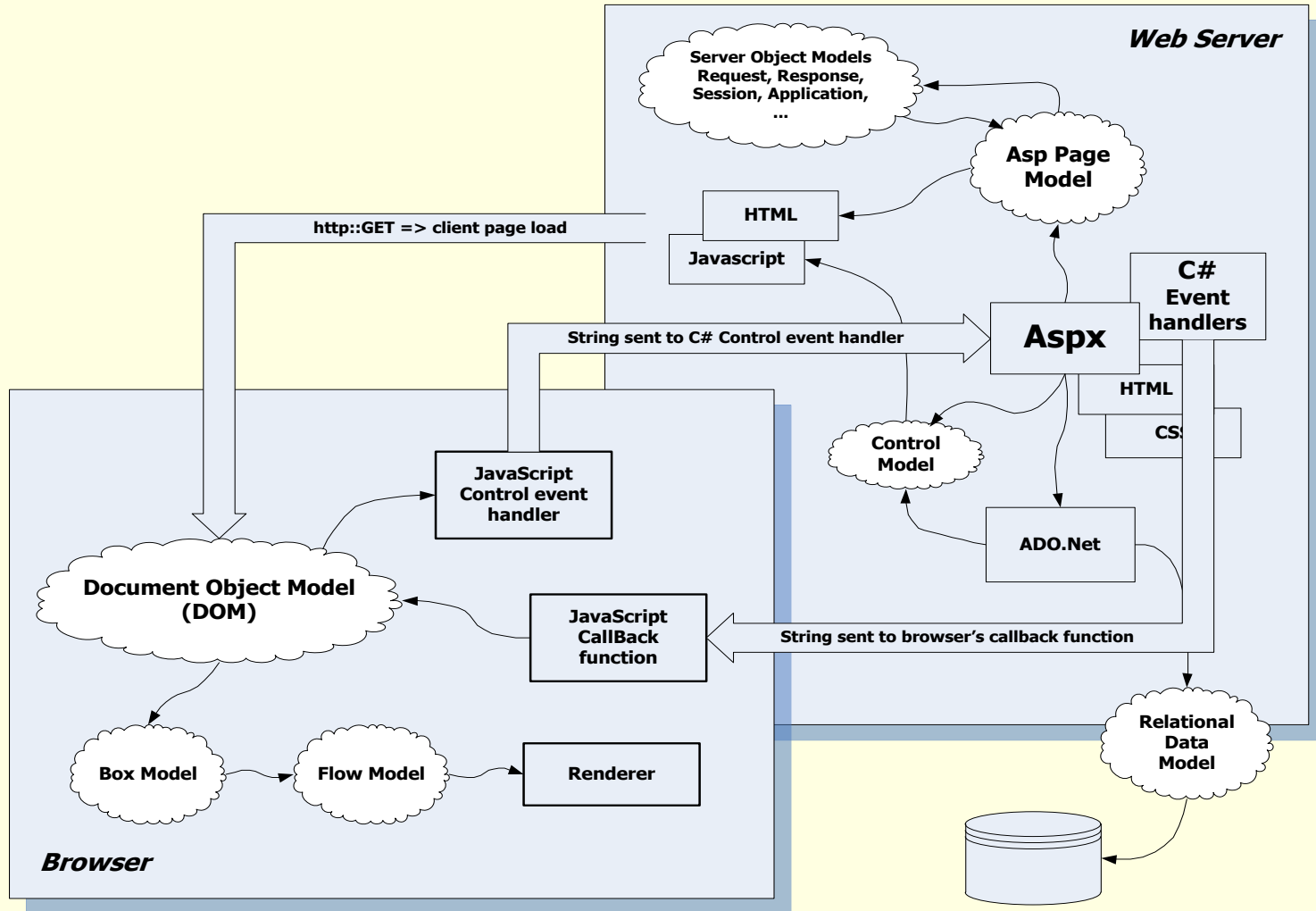


[Table of Contents](#)

ASP.NET Serving Model



Asp.Net Ajax Serving Model



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



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}
a:visited {color: darkblue}
a:active {color: darkorange}
body { color: darkred; background: yellow; }
body { margin-left: 10%; margin-right: 10%; }
h1 { margin-left: -6%; }
h2 { margin-left: -3%; }
</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>

```

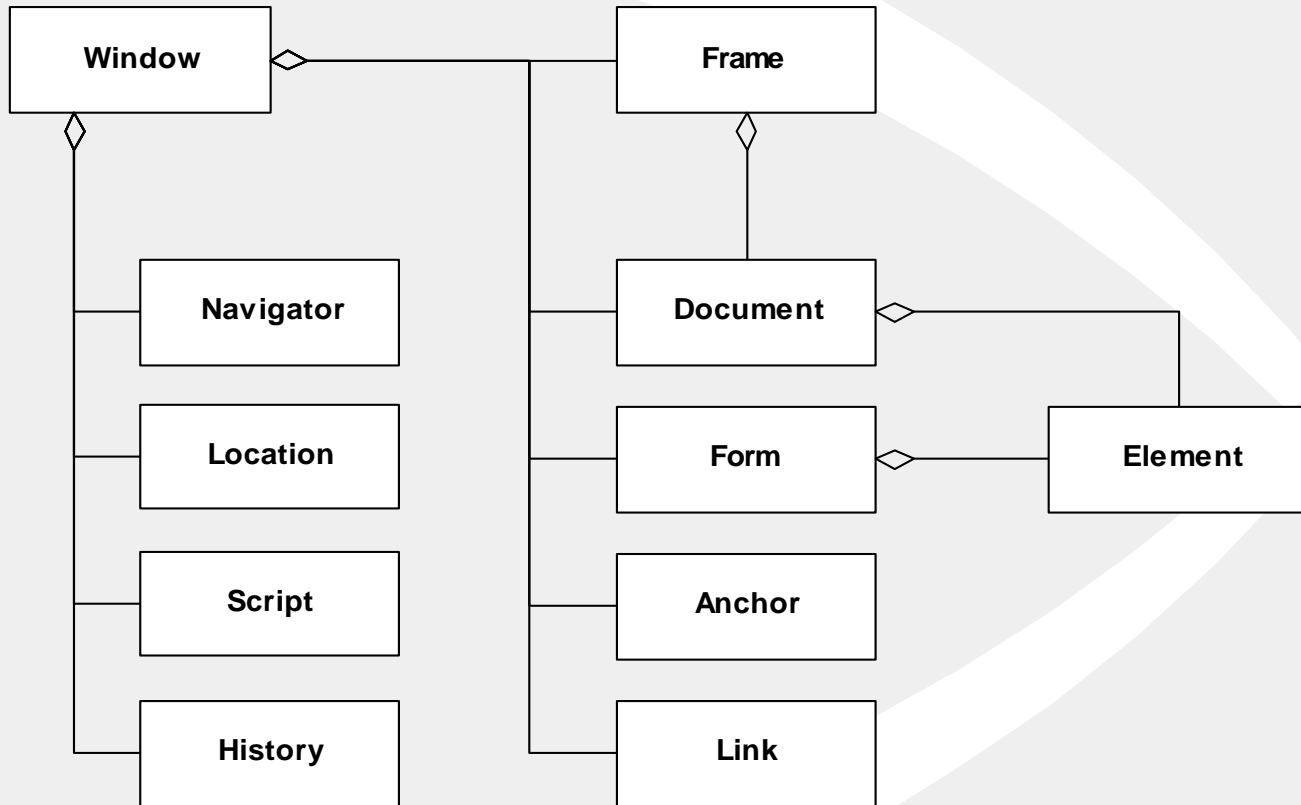
Done My Computer

Start C s. Tod... clas... SST .. c. D. M t. M C. t. t. C. R. d.

4:20 PM

Table of Contents

Browser Object Model



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**

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

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](#)