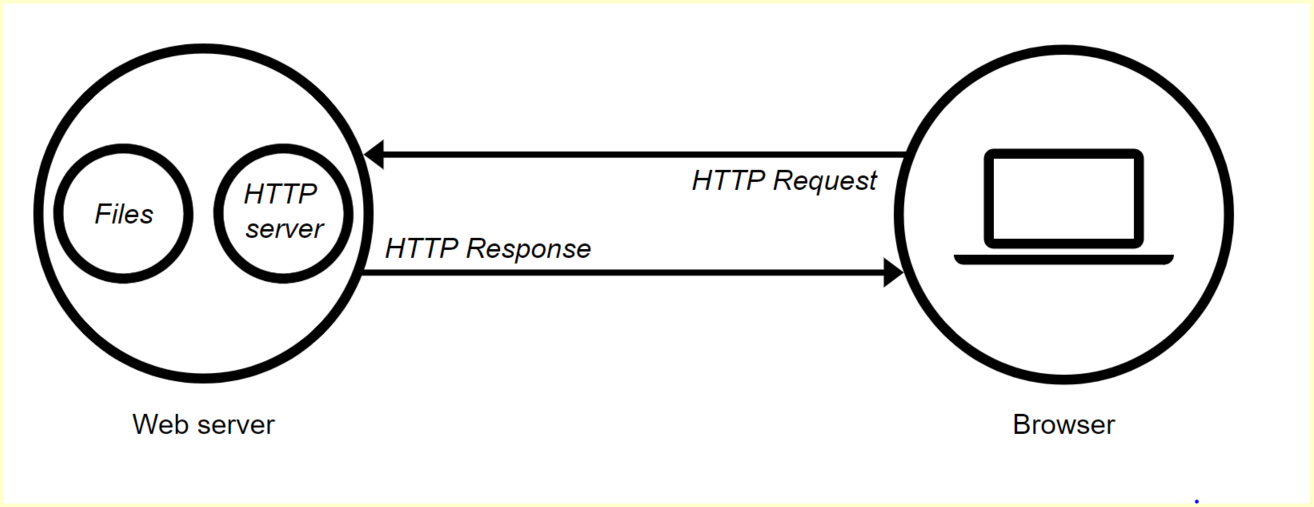
Web Programming Models

Fundamental Theory and Implementation: Overview


Files
HTTP server
HTTP Request
HTTP Response
Web server
Browser

The Web

- It is an information space where documents and other web resources are identified by Uniform Resource Locators (URLs, such as https://www.example.com/)
- It may be interlinked by hypertext, and are accessible via the Internet.
- The resources of the WWW may be accessed by users via a software application called a web browser.

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

Principles of good web design

1. **Visitor-centric, clear purpose**
2. **Progressive disclosure** It is an interaction design technique often used in human computer interaction)

 to help maintain the focus of a user's attention by reducing clutter, confusion, and cognitive workload.

1. **Displays quickly**
2. **Browser compatible**
3. **Intuitive navigation**
4. **Spelling, grammar, writing**
5. **Secure (eCommerce)**
6. **Attractive design, easy to read**
7. **Cultural bias? (Regional? Domestic? International?)**
8. **No technical problems (broken links, buggy scripts)**
9. **Maintainable (separate content from style)**
10. **Search Engine Accessible**

Web Design Principles

- Universal
- Decentralized
- Modular
- Extensible
- Scalable
- Accessible
- Forward/backwards compatibility

Web Components

- Clients and Servers
- Internet Service Providers
- Web Site Hosting Services
- Domains Names, URL’s and Ips
- Registrars

Clients & Servers

Clients (Browser)

- Internet Explorer
- Firefox
- Mozilla
- Netscape
- Opera
- Amaya
- AOL
- MSN

Servers

- Apache
- Microsoft
- Netscape
- zeus
- AOLserver
- AV
- JavaWebServer
- Oracle

Web Components

- Clients and Servers
- **Internet Service Providers**
- Web Site Hosting Services
- Domains Names, URL’s and Ips
- Registrars

Internet Service Providers

Connect Clients to the Internet

- Phone Company
- AOL
- Earthlink
- Verizone
- NetZero

- Basic internet connection
- Dialup/DSL/Cable/Sat
- Email

Web Components

- Clients and Servers
- Internet Service Providers
- **Web Site Hosting Services**
- Domains Names, URL's and Ips
- Registrars

Web Hosting Services

Connects Web Sites to the Internet

- Computer (server) farm
- Web server software
- Firewall hardware and software
- IT services
  - (Backup, troubleshooting, hardware repair)
- Disk space
- Bandwidth / connection to internet
- Routers and switchers
- Email server / storage

Web Components

- Clients and Servers
- Internet Service Providers
- Web Site Hosting Services
- Domains Names, URL’s and Ips
- Registrars

Domain's URL's and Internet protocol (IPs)

- Domain name: The specific address of a computer on the Internet
  - microsoft.com
- Uniform Resource Locator (URL):
  - http://www.microsoft.com/faqs.html
- Internet protocol (IP) address
  - 192.168.1.1

Domain Registrar

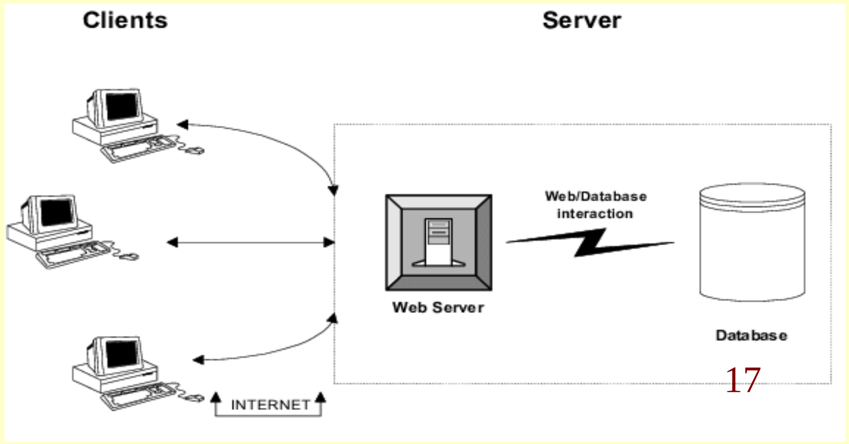
- A company that provides domain name registration services for a fee.
- Maintain database which maps domain names to IP’s
- Propagate new domain name/IP address information across the internet

Creating a Web Site

1. Choose a domain name
2. Register with a Registrar
3. Choose a hosting service
4. Tell Registrar the IP address
5. Create web content
6. Store (publish) onto hosting server (FTP)
7. Submit new site to search engines

Basic Concepts

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

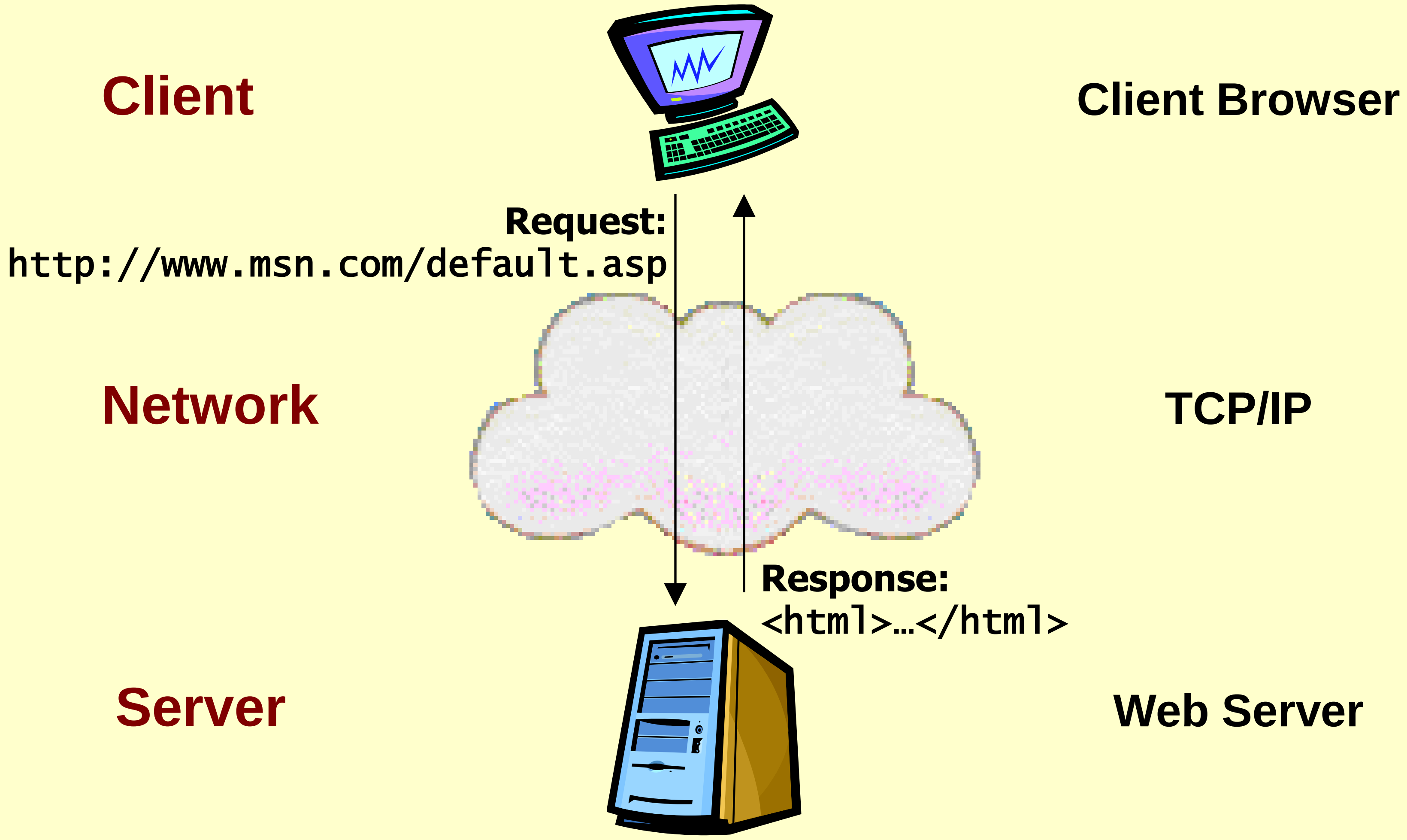


Servers on the Internet

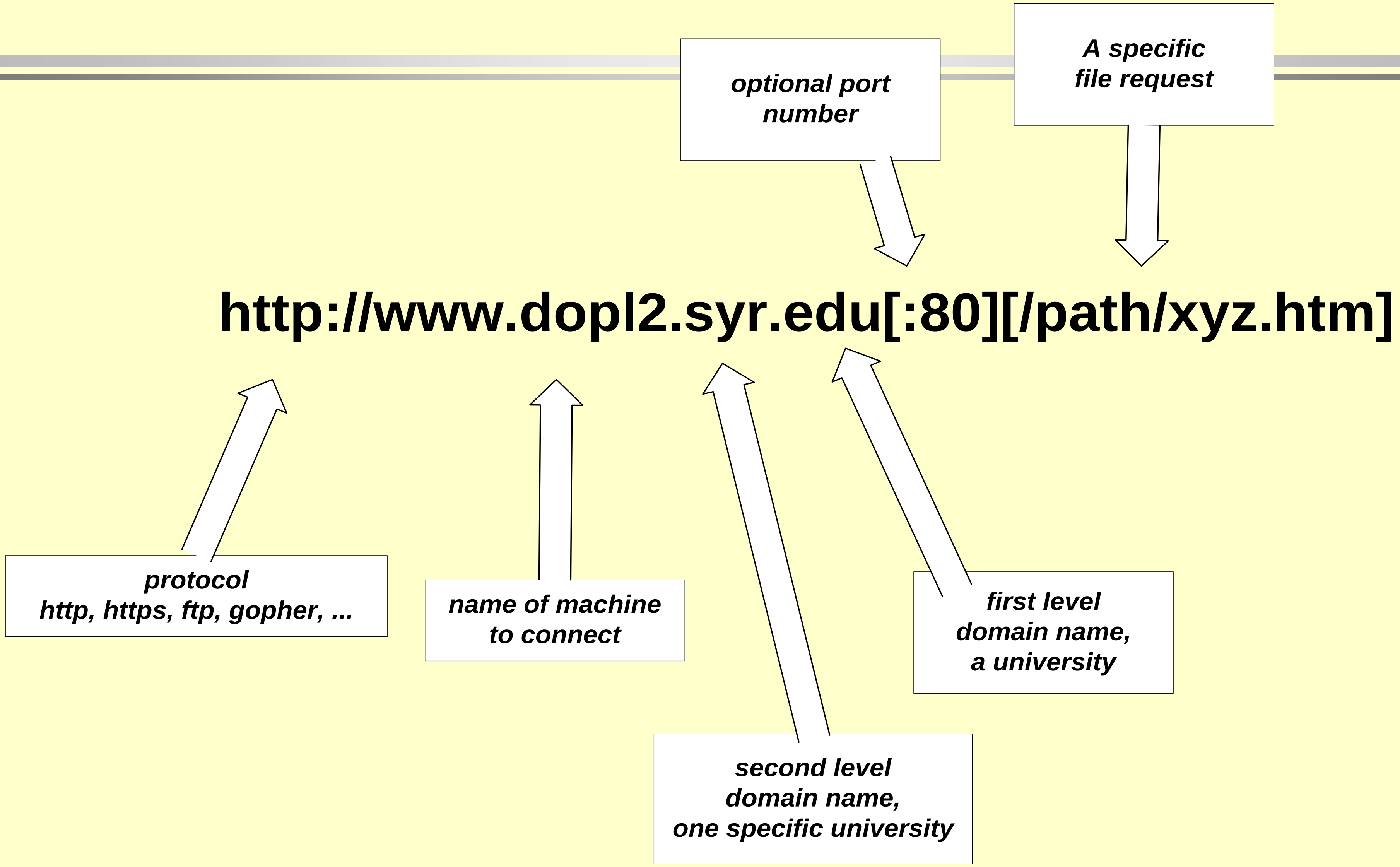
• HTTP	- HyperText Transport Protocol
• FTP	- File Transport Protocol
• NNTP	- Network News Transfer Protocol
• DNS	- Distributed Name Service
• telnet	- log into a remote computer

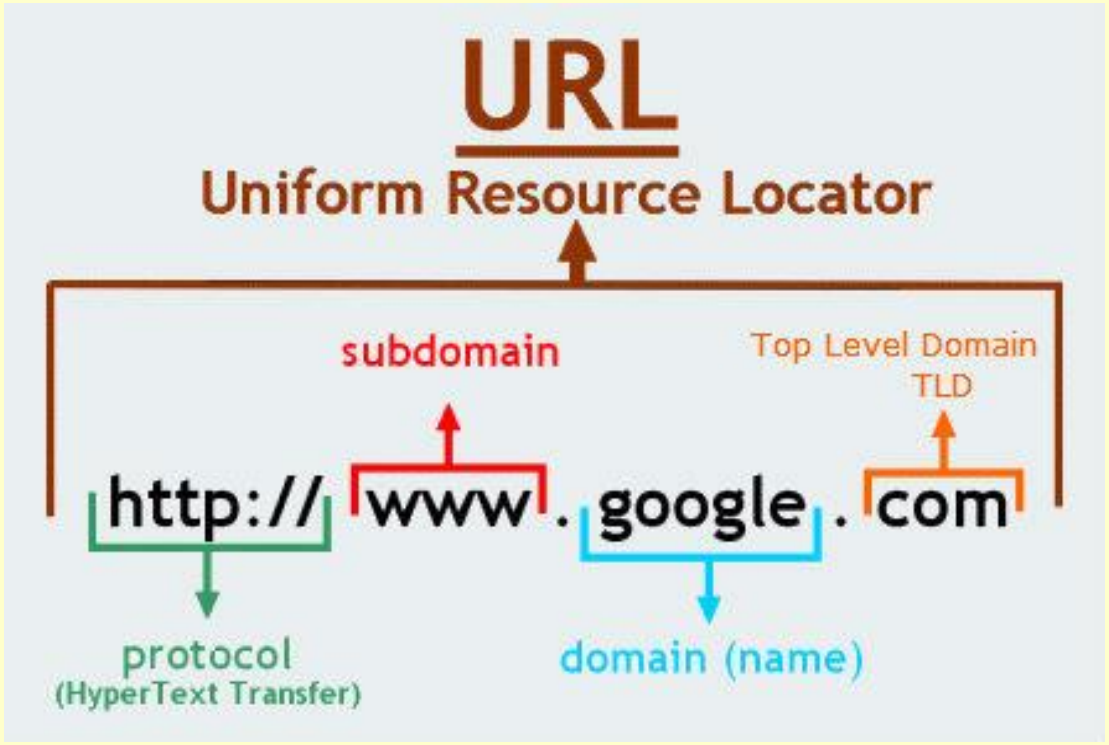
Internet Technologies

WWW Architecture



Address Resolution


optional port number
A specific file request
http://www.dopl2.syr.edu[:80][/path/xyz.htm]
protocol http, https, ftp, gopher, ...
name of machine to connect
second level domain name, one specific university
first level domain name, a university


URL
Uniform Resource Locator
subdomain
Top Level Domain
TLD
http://
www
.
google
.
com
protocol
(HyperText Transfer)
domain (name)

Domain Names Type

Top Level Domains:

1. .com
2. .net
3. .edu
4. .org
5. .mil
6. .gov

Country Code top-level domains:

.in

.uk…. etc

Networks

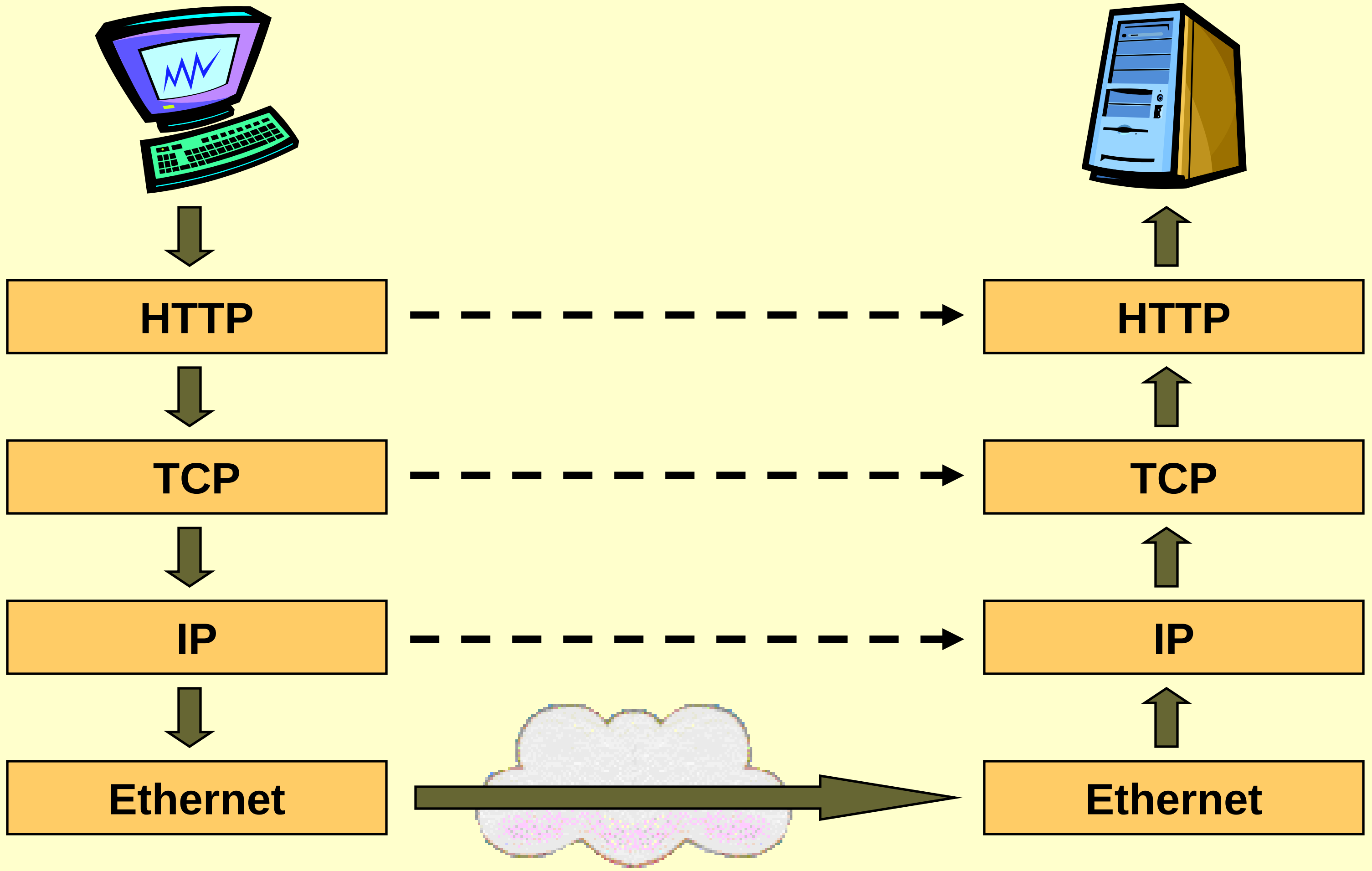
Networks

- Network = an interconnected collection of independent computers

- Why have networks?
  - Resource sharing
  - Reliability
  - Cost savings
  - Communication

- Web technologies add:
  - New business models: e-commerce, advertising
  - Entertainment
  - Applications without a client-side install

Network Protocol Stack



Networks - Transport Layer

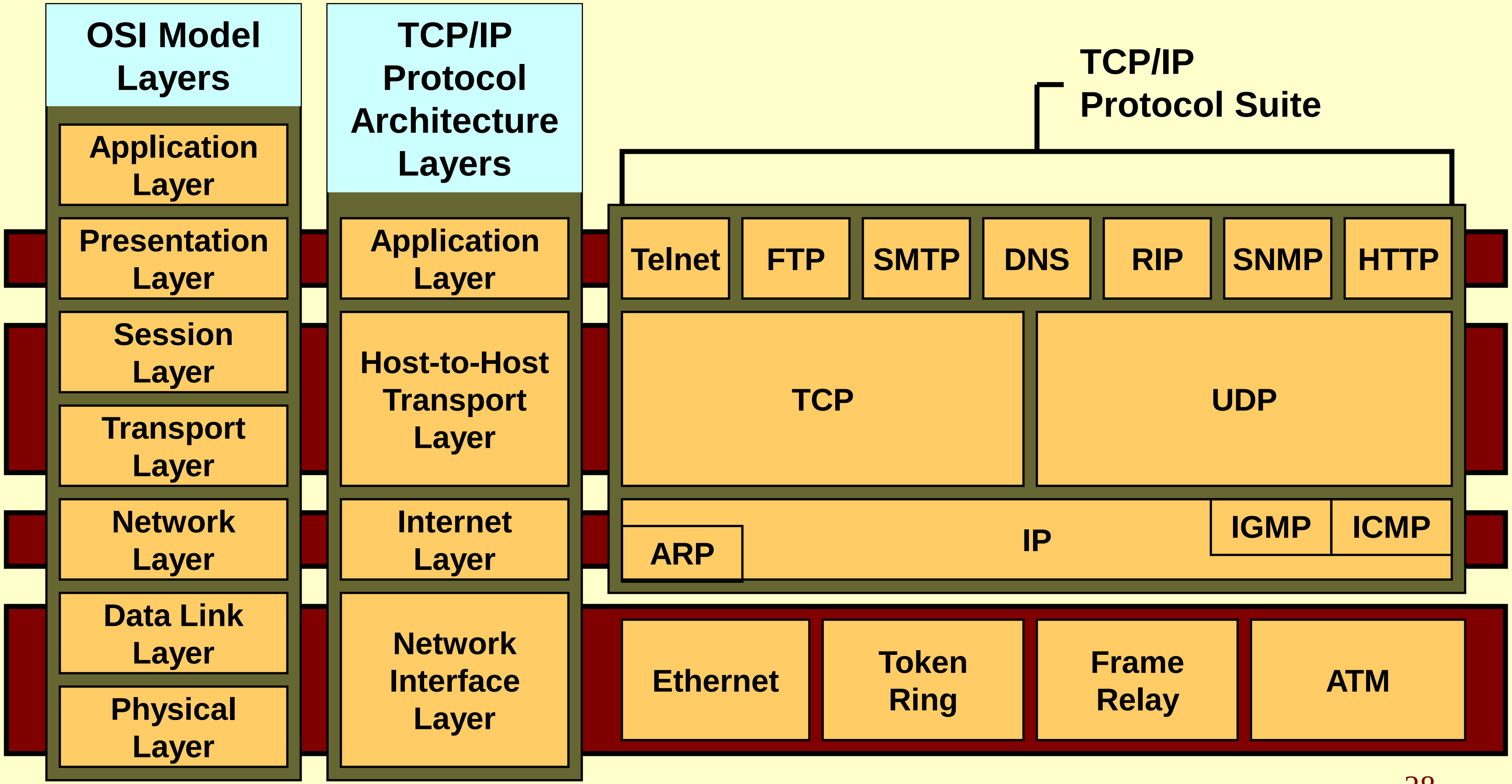
- Provides efficient, reliable and cost-effective service
- Uses the Sockets programming model
- Ports identify application
  - Well-known ports identify standard services (e.g. HTTP uses port 80, SMTP uses port 25)
- Transmission Control Protocol (TCP)
  - Provides reliable, connection-oriented byte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

Network Protocols



HTTP Protocol

HTTP Protocol

- Client/Server, Request/Response architecture
  - You request a Web page
    - e.g. `http://www.msn.com/default.asp`
    - HTTP request
  - The Web server responds with data in the form of a Web page
    - HTTP response
    - Web page is expressed as HTML
  - Pages are identified as a Uniform Resource Locator (URL)
    - Protocol: `http`
    - Web server: `www.msn.com`
    - Web page: `default.asp`

HTTP is Stateless

- HTTP is a stateless protocol
- Each HTTP request is independent of previous and subsequent requests
- Statelessness has a big impact on how scalable applications are designed

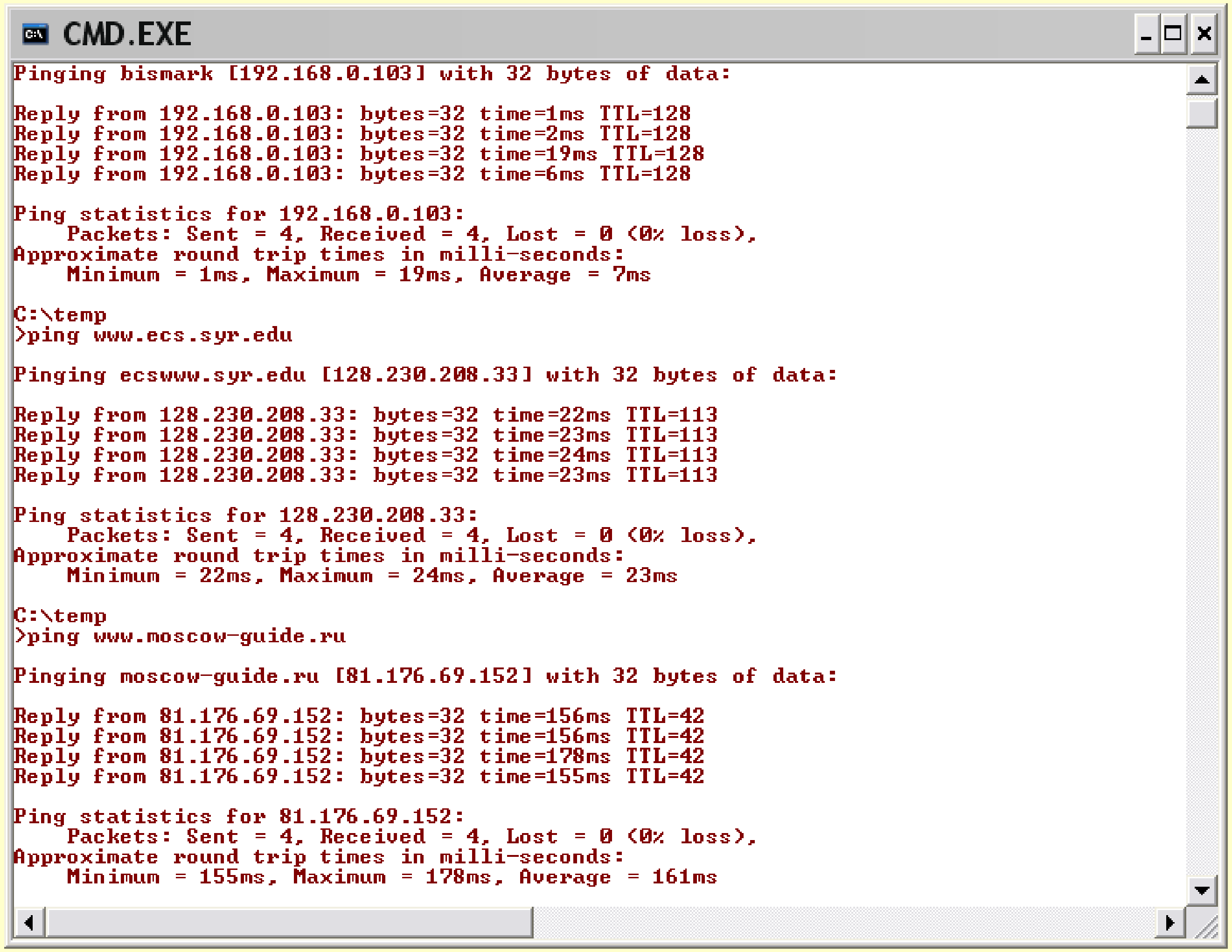
Cookies

- A mechanism to store a small amount of information (up to 4KB) on the client
- A cookie is associated with a specific web site
- Cookie is sent in HTTP header
- Cookie is sent with each HTTP request
- Can last for only one session (until browser is closed) or can persist across sessions
- Can expire some time in the future

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 – requires design effort to do that.

Pinging Various URLs



```
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
```

Multipurpose Internet Mail Extensions (MIME)

- 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

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

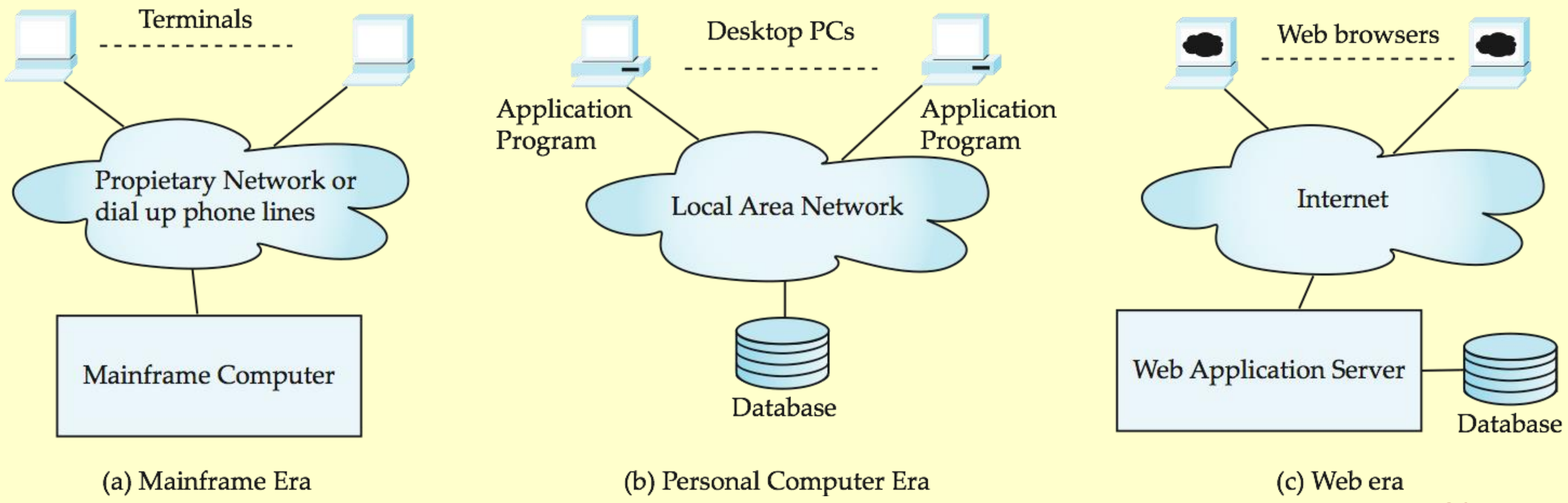
Application Programs and User Interfaces

Application Programs and User Interfaces

- Most database users do *not* use a query language like SQL
- An application program acts as the intermediary between users and the database
  - Applications split into
    - front-end
    - middle layer
    - backend
- Front-end: user interface
  - Forms
  - Graphical user interfaces (GUI)
  - Many interfaces are Web-based

Application Architecture Evolution

- Three distinct era's of application architecture
  - mainframe (1960's and 70's)
  - personal computer era (1980's)
  - Web era (1990's onwards)



Web Interface

- Web browsers have become the de-facto standard user interface to databases
  - Enable large numbers of users to access databases from anywhere
  - Avoid the need for downloading/installing specialized code, while providing a good graphical user interface
    - Javascript, Flash and other scripting languages run in browser, but are downloaded transparently
  - Examples: banks, airline and rental car reservations, university course registration and grading, an so on.

The World Wide Web

- The Web is a distributed information system based on hypertext.
- Most Web documents are hypertext documents formatted via the HyperText Markup Language (HTML)
- HTML documents contain
  - text along with font specifications, and other formatting instructions
  - hypertext links to other documents, which can be associated with regions of the text.
  - forms, enabling users to enter data which can then be sent back to the Web server

Uniform Resources Locators

- In the Web, functionality of pointers is provided by Uniform Resource Locators (URLs).
- URL example:

  http://www.acm.org/sigmod

  - The first part indicates how the document is to be accessed
    - "http" indicates that the document is to be accessed using the Hyper Text Transfer Protocol.
  - The second part gives the unique name of a machine on the Internet.
  - The rest of the URL identifies the document within the machine.
- The local identification can be:
    - The path name of a file on the machine, or
    - An identifier (path name) of a program, plus arguments to be passed to the program

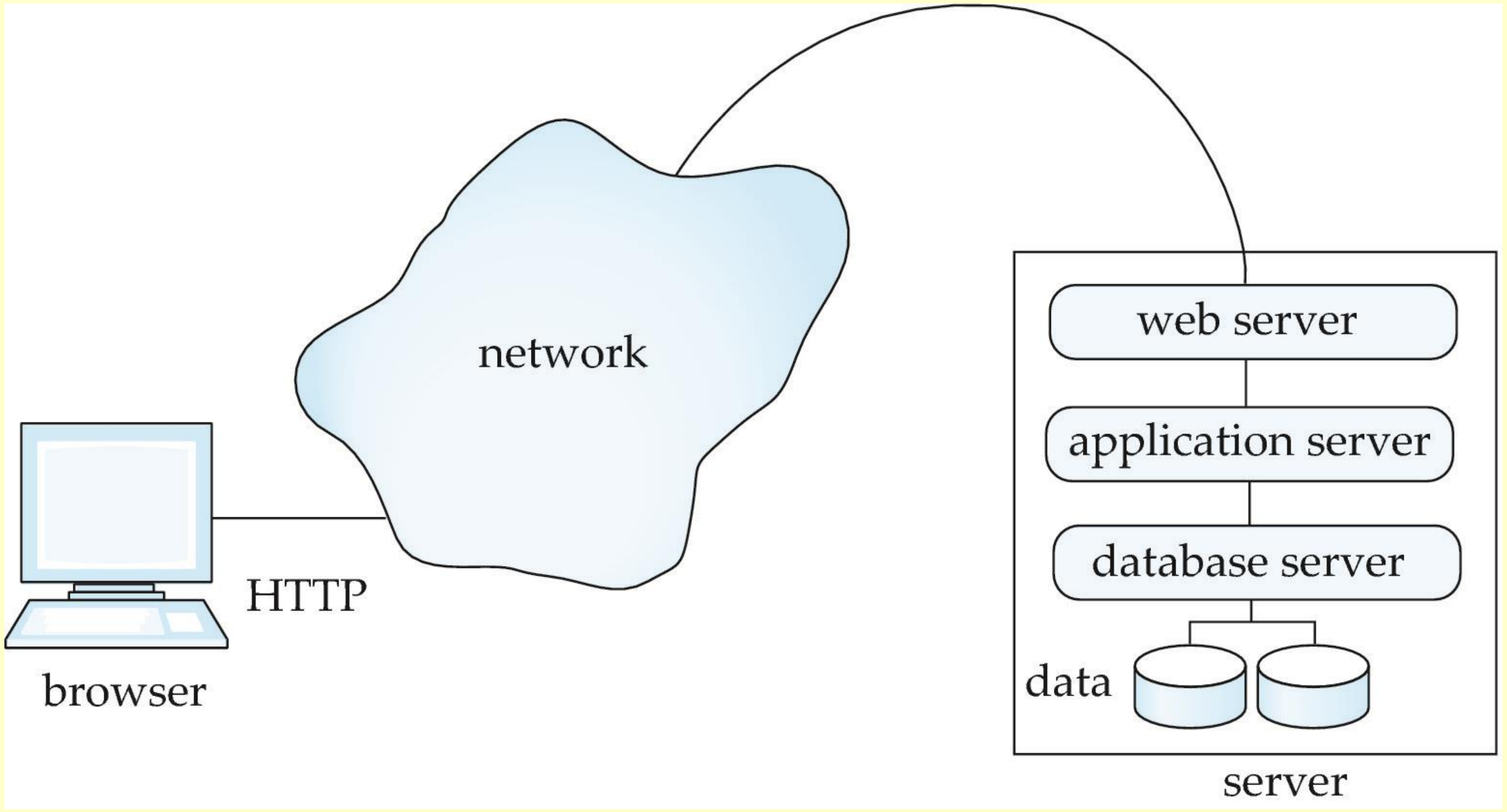
HTML and HTTP

- HTML provides formatting, hypertext link, and image display features
  - including tables, stylesheets (to alter default formatting), etc.
- **HTML also provides input features**
    - Select from a set of options
      - Pop-up menus, radio buttons, check lists
    - Enter values
      - Text boxes
  - Filled in input sent back to the server, to be acted upon by an executable at the server
- HyperText Transfer Protocol (HTTP) used for communication with the Web server

Web Servers

- A Web server can easily serve as a front end to a variety of information services.
- The document name in a URL may identify an executable program, that, when run, generates a HTML document.
  - When an HTTP server receives a request for such a document, it executes the program, and sends back the HTML document that is generated.
  - The Web client can pass extra arguments with the name of the document.
- To install a new service on the Web, one simply needs to create and install an executable that provides that service.
  - The Web browser provides a graphical user interface to the information service.
- Common Gateway Interface (CGI): a standard interface between web and application server

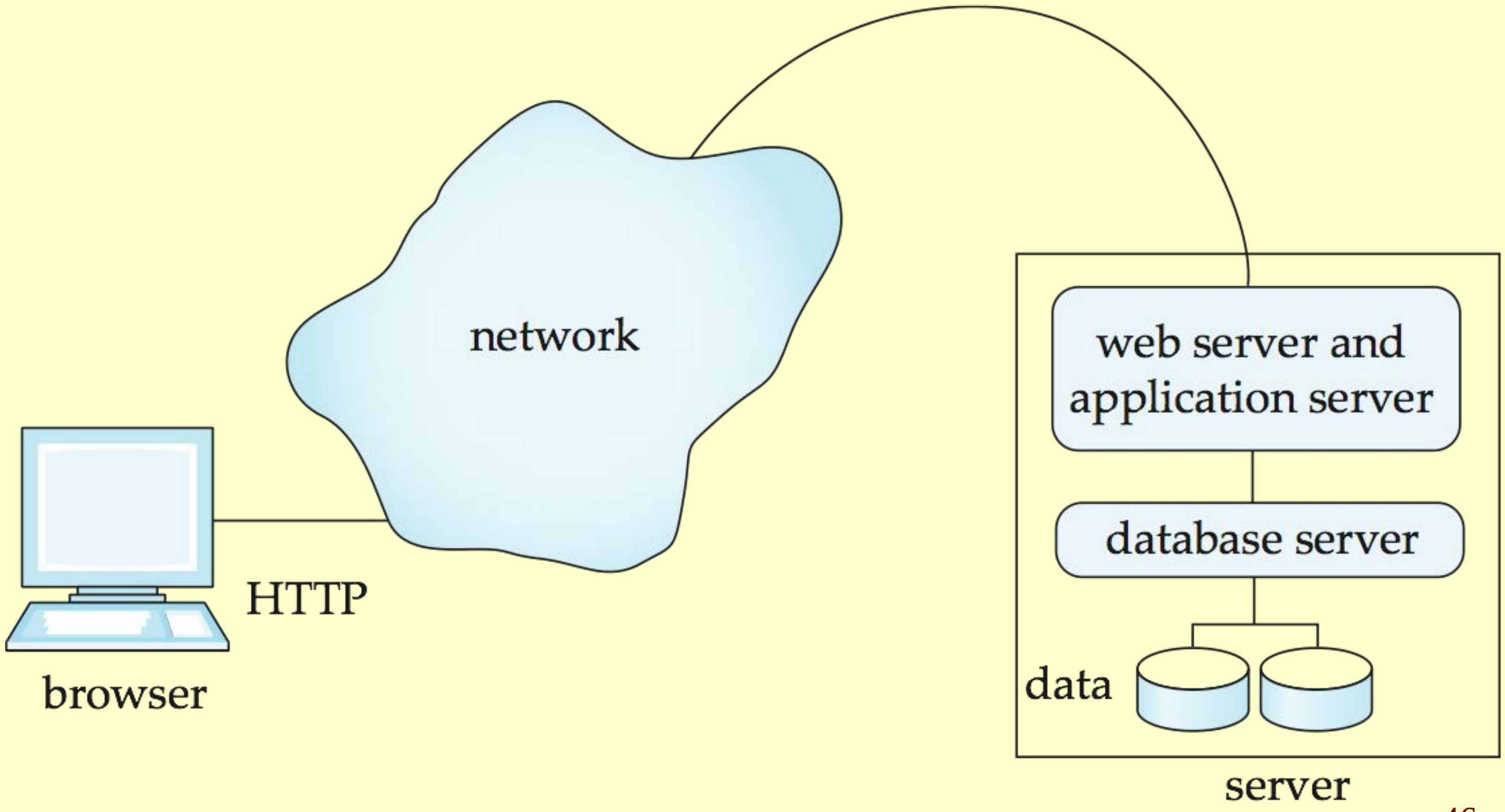
Three-Layer Web Architecture (Web based application)



Two-Layer Web Architecture (client-server Application)

- Multiple levels of indirection have overheads

  Alternative: two-layer architecture



Programming the Web

Programming the Web

- Client-Side Programming
  - JavaScript
  - Dynamic HTML
  - .Net controls

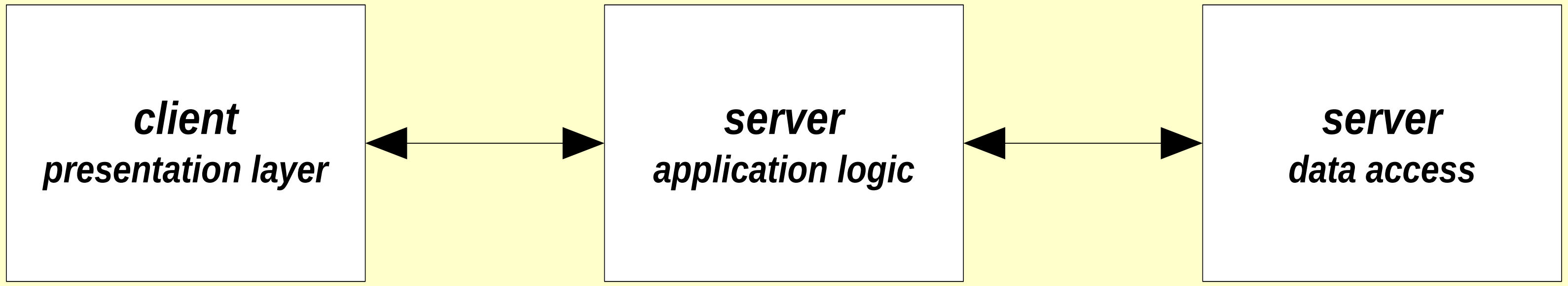
- **Server-Side Programming**
  - ASP script
  - Server components
  - C# code-behind
  - ADO
  - Web controls used on ASPX pages
  - Web services

Web Processing Models

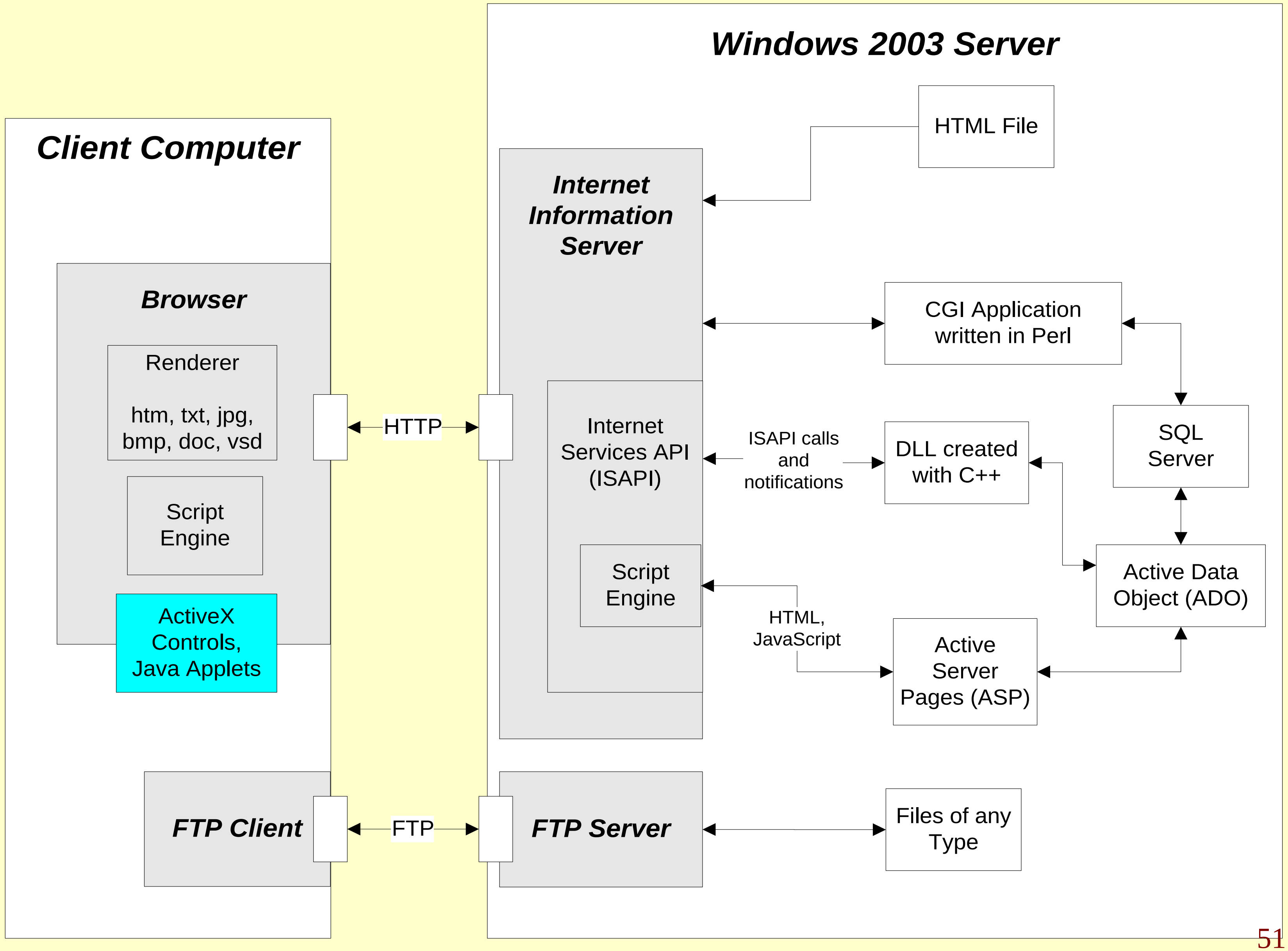
- ***HyperText Transfer Protocol (HTTP)***
  - Universal access
  - HTTP is a "request-response" protocol specifying that a client will open a connection to server then send request using a very specific format. Server will respond and then close connection.
- ***HyperText Markup Language (HTML)***
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

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

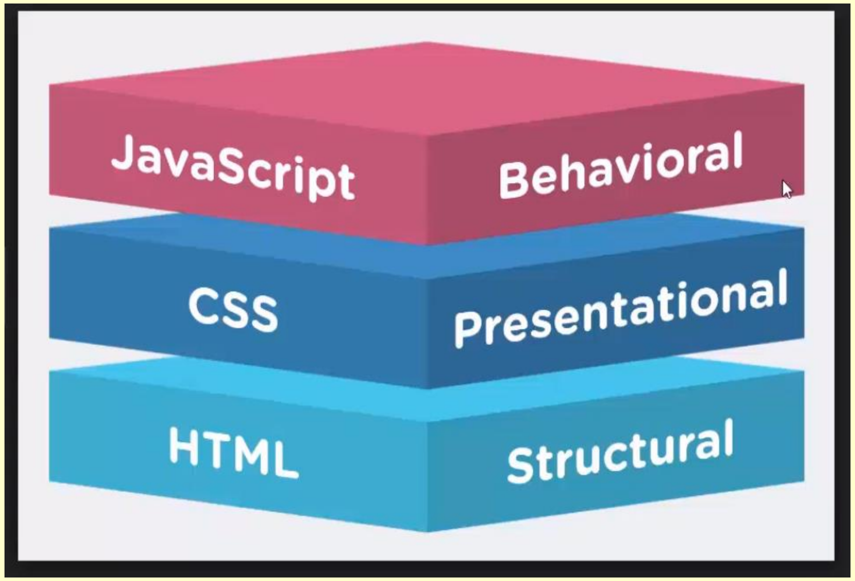


Client/Server - Current Web Model


Windows 2003 Server
Client Computer
Browser
Renderer
htm, txt, jpg, bmp, doc, vsd
Script Engine
ActiveX Controls, Java Applets
HTTP
FTP Client
FTP
Internet Information Server
Internet Services API (ISAPI)
Script Engine
FTP Server
HTML File
CGI Application written in Perl
ISAPI calls and notifications
DLL created with C++
SQL Server
Active Data Object (ADO)
HTML, JavaScript
Active Server Pages (ASP)
Files of any Type

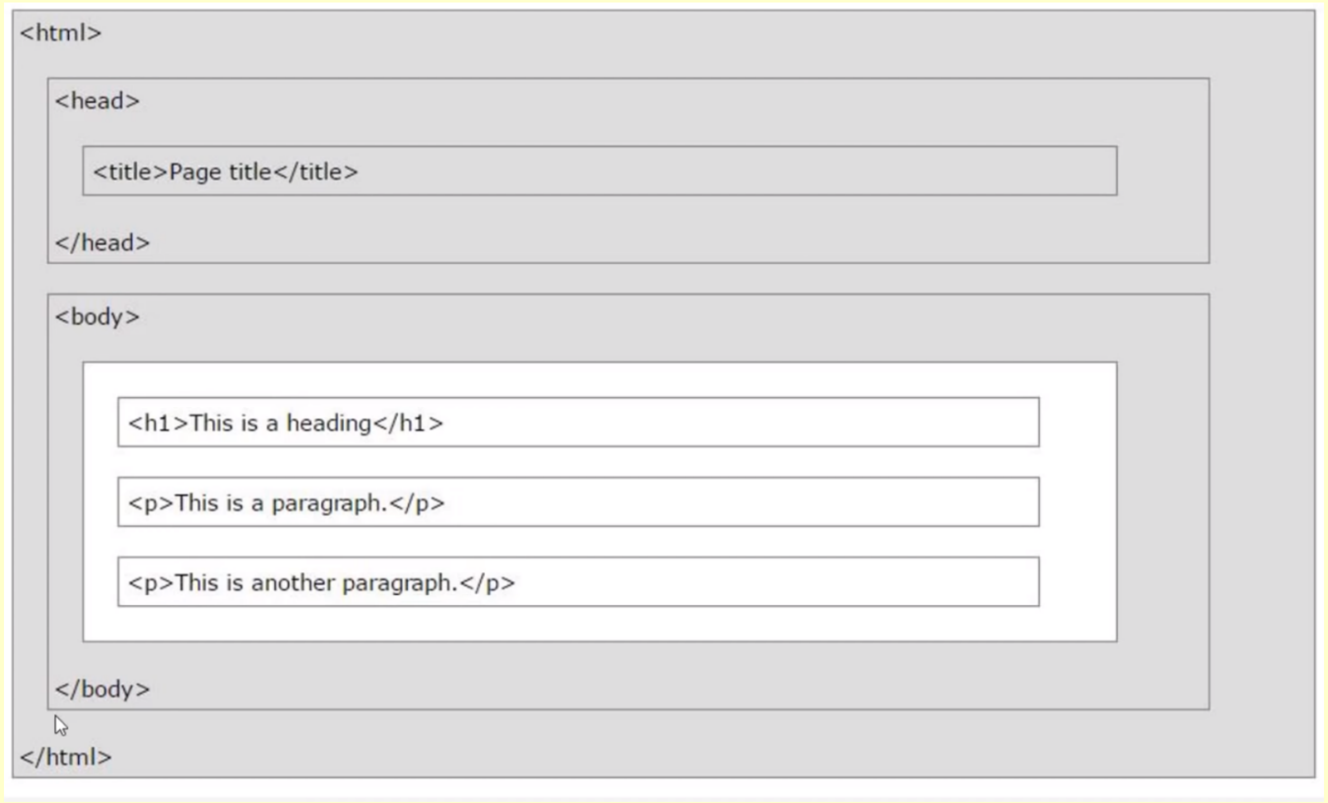
Programming the Web
Client-Side Code

- **What is client-side code?**
  - Software that is downloaded from Web server to browser and then executes on the browser client
- **Why client-side code?**
  - Better scalability: less work done on server
  - Better performance/user experience
  - Create UI constructs not inherent in HTML
    - Drop-down and pull-out menus
    - Tabbed dialogs
  - Cool effects, e.g. animation
  - Data validation


JavaScript
Behavioral
CSS
Presentational
HTML
Structural

HyperText Markup Language (HTML)

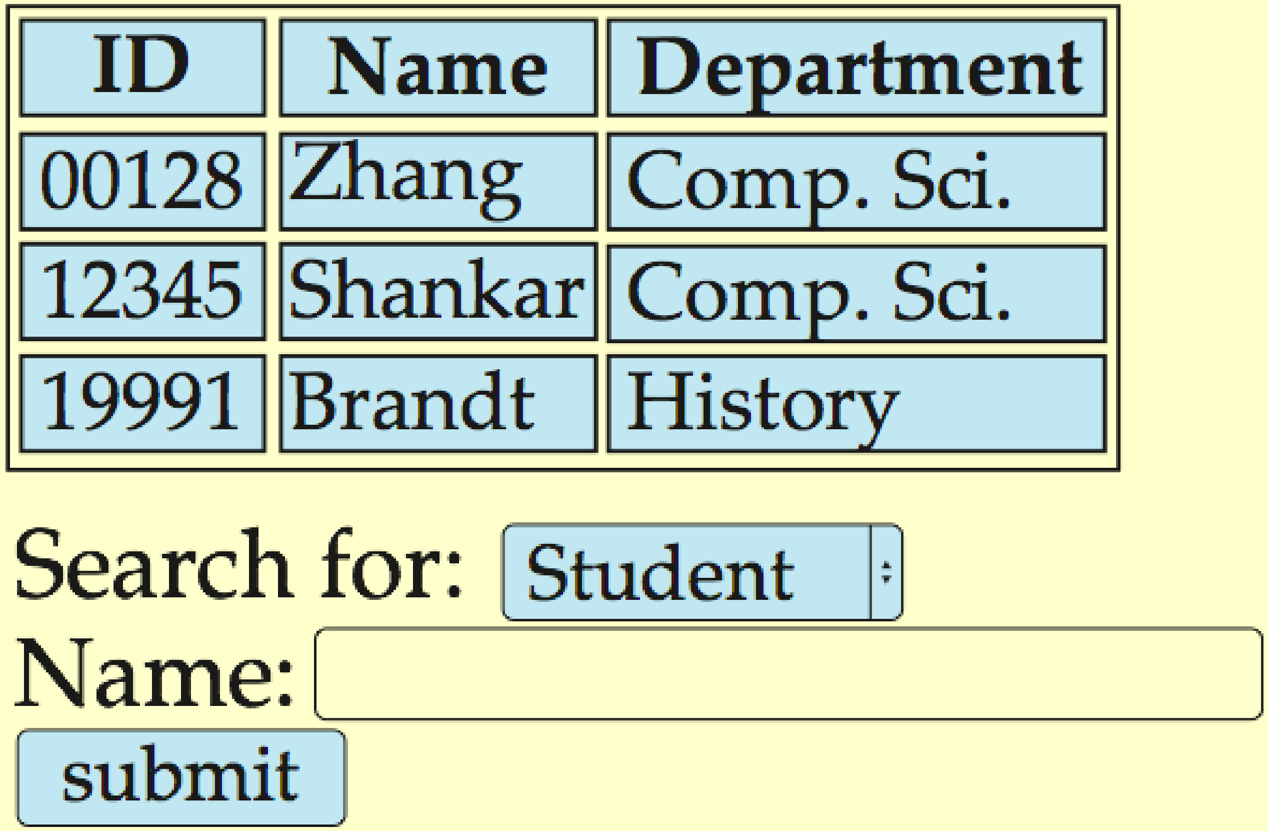
- The markup language used to represent Web pages for viewing by people
  - Designed to display data, not store/transfer data
- Rendered and viewed in a Web browser
- Can contain ***links*** to images, documents, and other pages
- Not extensible – uses only tags specified by the standard
- Derived from Standard Generalized Markup Language (SGML)
- HTML 3.2, 4.01, 5, XHTML 1.0


<html>
<head>
<title>Page title</title>
</head>
<body>
<h1>This is a heading</h1>
<p>This is a paragraph.</p>
<p>This is another paragraph.</p>
</body>
</html>

Sample HTML Source Text

```
<html>
<body>
  <table border>
    <tr> <th>ID</th> <th>Name</th> <th>Department</th> </tr>
    <tr> <td>00128</td> <td>Zhang</td> <td>Comp. Sci.</td> </tr>
    ….
  </table>
   <form action="PersonQuery" method=get>
    Search for:
       <select name="persontype">
           <option value="student" selected>Student </option>
           <option value="instructor"> Instructor </option>
       </select> <br>
    Name: <input type=text size=20 name="name">
    <input type=submit value="submit">
  </form>
</body> </html>
```

Display of Sample HTML Source



ID	Name	Department
00128	Zhang	Comp. Sci.
12345	Shankar	Comp. Sci.
19991	Brandt	History

Search for: Student

Name:

Programming the Web
Server-Side Code

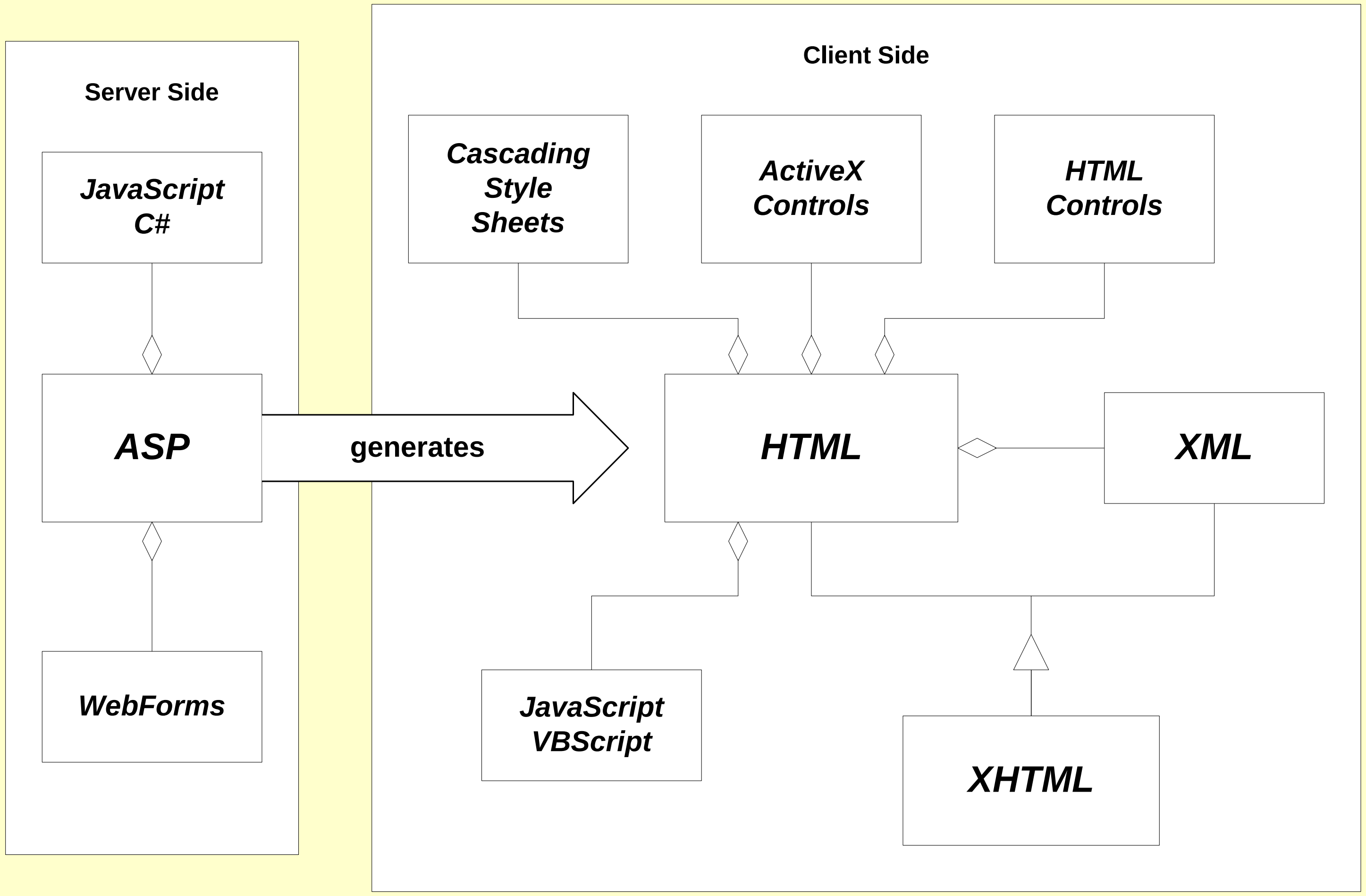
- **What is server-side code?**
  - Software that runs on the server, not the client
  - Receives input from
    - URL parameters
    - HTML form data
    - Cookies
    - HTTP headers
  - Can access server-side databases, e-mail servers, files, mainframes, etc.
  - Dynamically builds a custom HTML response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

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

Web Programming – Language Model



Programming Paradigms

Event-Based Programming

- When something of interest occurs, an event is raised and application-specific code is executed
- Events provide a way for you to hook in your own code into the operation of another system
- Event = callback
- User interfaces are all about events
  - `onClick`, `onMouseOver`, `onMouseMove`…|
- Events can also be based upon time or interactions with the network, operating system, other applications, etc.

Event-Based Programming on Client

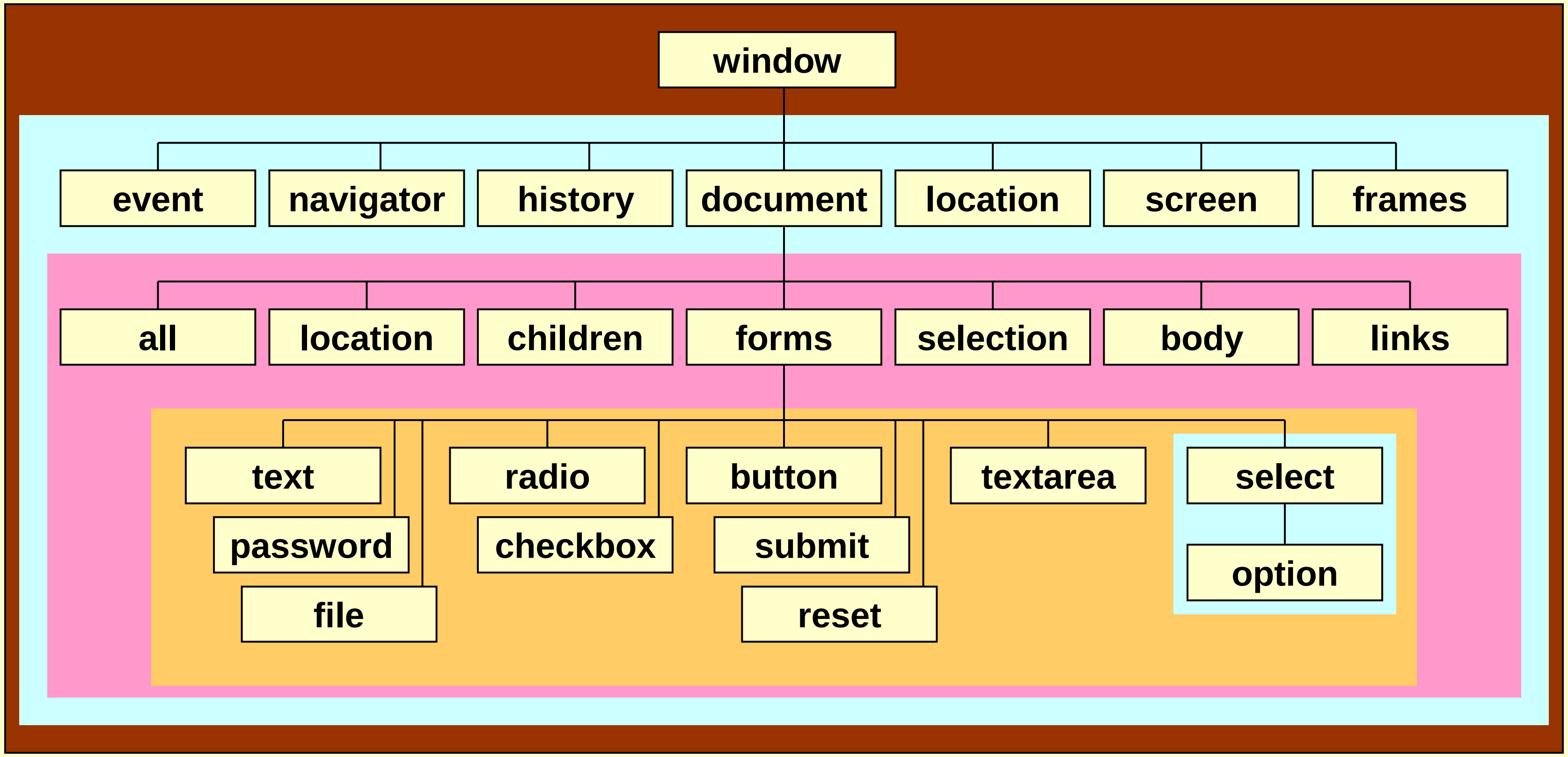
Dynamic HTML (DHTML)

- Script is embedded within, or attached to, an HTML pages
- Usually written in JavaScript (ECMAScript, JScript) for portability
  - Internet Explorer also supports VBScript and other scripting languages
- Each HTML element becomes an object that has associated events (e.g. `onClick`)
- Script provides code to respond to browser events

Programming the Web

DHTML

- DHTML Document Object Model (DOM)



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 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.

Programming the Web

Active Server Pages (ASP)

- Technology to easily create server-side applications
- ASP pages are written in a scripting language, usually VBScript or Jscript
- An ASP page contains a sequence of static HTML interspersed with server-side code
- ASP script commonly accesses and updates data in a database

Event-Based Programming on Server
ASP.Net

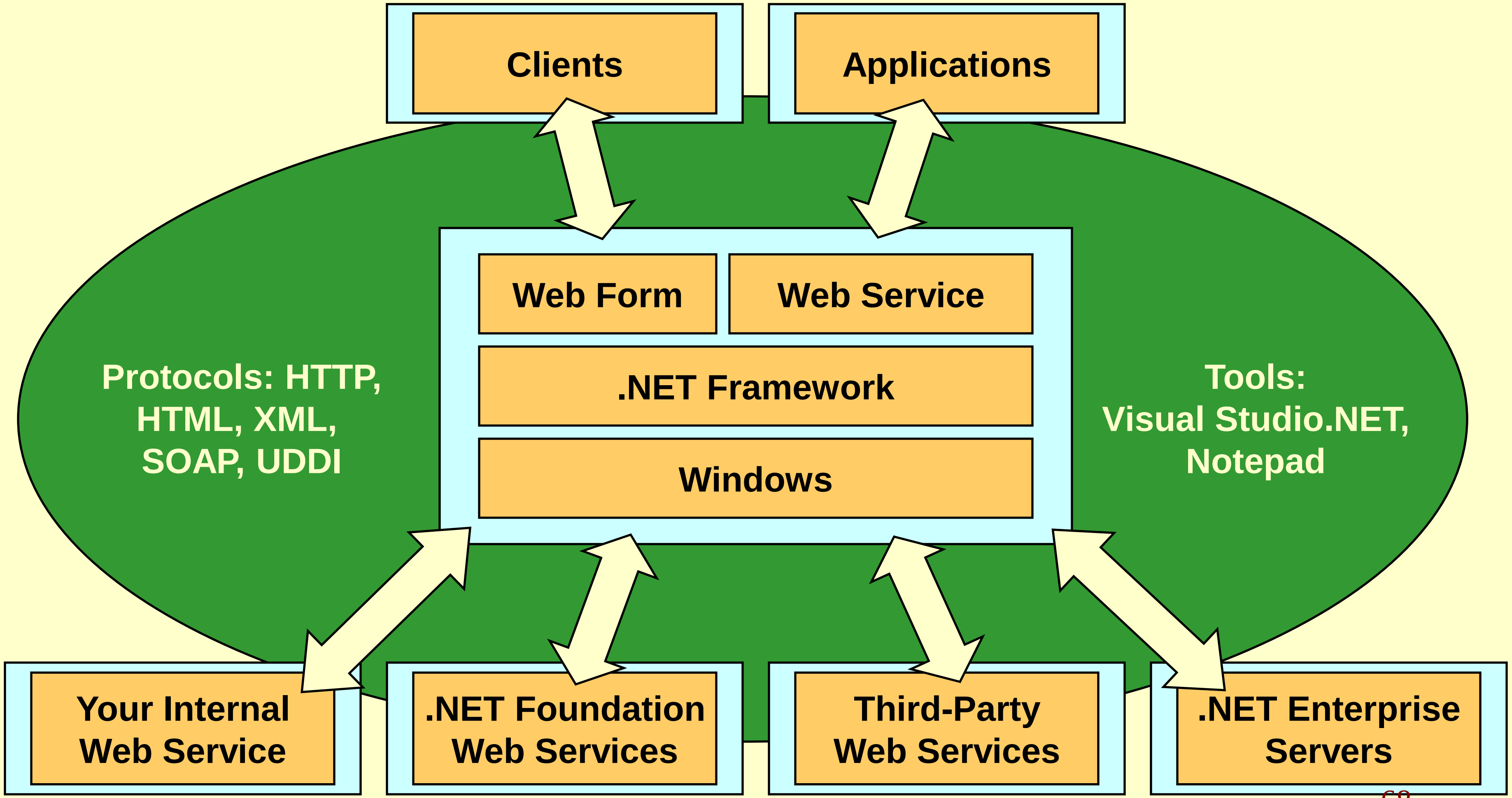
- Pages are constructed from HTML, Web Controls, and C# event handlers.

- The ASP.Net Page processing renders Web Controls on a page into HTML constructs with attached Javascript event handlers.
  - The Javascript handlers post messages back to the server describing the event, which is then handled by C# code on the server.

- The result of the handled event is usually another page sent back to the browser client.

Introduction to .NET

The .NET Platform


Clients
Applications
Web Form
Web Service
.NET Framework
Windows
Protocols: HTTP, HTML, XML, SOAP, UDDI
Tools: Visual Studio.NET, Notepad
Your Internal Web Service
.NET Foundation Web Services
Third-Party Web Services
.NET Enterprise Servers

Common Language Runtime
Assemblies

- Assembly
  - Logical unit of deployment
  - Contains Manifest, Metadata, MSIL and resources
- Manifest
  - Metadata about the components in an assembly (version, types, dependencies, etc.)
- Type Metadata
  - Completely describes all types defined in an assembly: properties, methods, arguments, return values, attributes, base classes, …

Common Language Runtime
Services

- Code management
- Conversion of MSIL to native code
- Loading and execution of managed code
- Creation and management of metadata
- Verification of type safety
- Insertion and execution of security checks
- Memory management and isolation
- Handling exceptions across languages
- Interoperation between .NET Framework objects and COM objects and Win32 DLLs
- Automation of object layout for late binding
- Developer services (profiling, debugging, etc.)

Common Language Runtime
Security

- Evidence-based security (authentication)
- Based on user identity and code identity
- Configurable policies
- Imperative and declarative interfaces

Windows Forms

- Framework for building rich clients
- Built upon .NET Framework, languages
- Rapid Application Development (RAD)
- Visual inheritance
- Anchoring and docking
- Rich set of controls
- Extensible controls
- Data-aware
- Easily hooked into Web Services
- ActiveX support
- Licensing support
- Printing support
- Advanced graphics

Web Forms

- Built with ASP.NET
  - Logical evolution of ASP
  - Similar development model: edit the page and go
- Requires less code
- New programming model
  - Event-driven/server-side controls
  - Rich controls (e.g. data grid, validation)
  - Data binding
  - Controls generate browser-specific code
  - Simplified handling of page state

Web Forms

- Allows separation of UI and business logic
- Uses .NET languages
  - Not just scripting
- Easy to use components
- XCOPY/FTP deployment
- Simple configuration (XML-based)

ADO.NET

- Similar to ADO, but better factored
- Language-neutral data access
- Supports two styles of data access
  - Disconnected
  - Forward-only, read-only access
- Supports data binding
- DataSet: a collection of tables
- Can view and process data relationally (tables) or hierarchically (XML)

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

Areas of Exploration

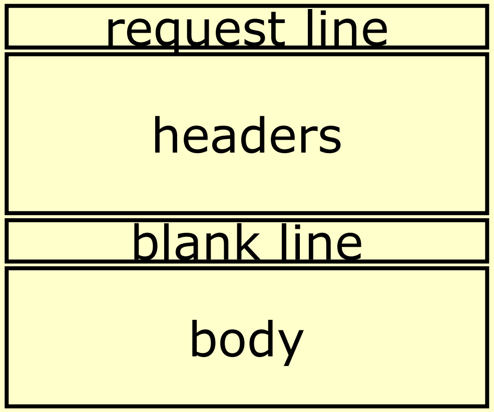
• XML	- Universal Data Services
• TVWeb	- merger of features
• MathML	- Mathematical Markup Language
• RDF	- Resouce Description Framework
• Accessibility	- for the handicapped
• SMIL	- Synchronized Multimedia Integration Language
• Internationalization	
• Speech	

References

- Introduction to the Web and .Net, Mark Sapossnek, Computer Science, Boston Univ.
  - slides available on www.gotdotnet.com
- World Wide Web Consortium
  - Excellent Tutorial Papers, standards
- XHTML Black Book, Steven Holzner, Coriolis, 2000
  - Very comprehensive treatment of HTML, XHTML, JavaScript
- Inside Dynamic HTML, Scott Issacs, Microsoft Press, 1997
- C# .Net Web Developer's Guide, Turtschi et. al., Syngress, 2002
  - Class text
- Web Developers Virtual Library
  - Excellent set of tutorials
- Class Web Links
  - Web links.htm

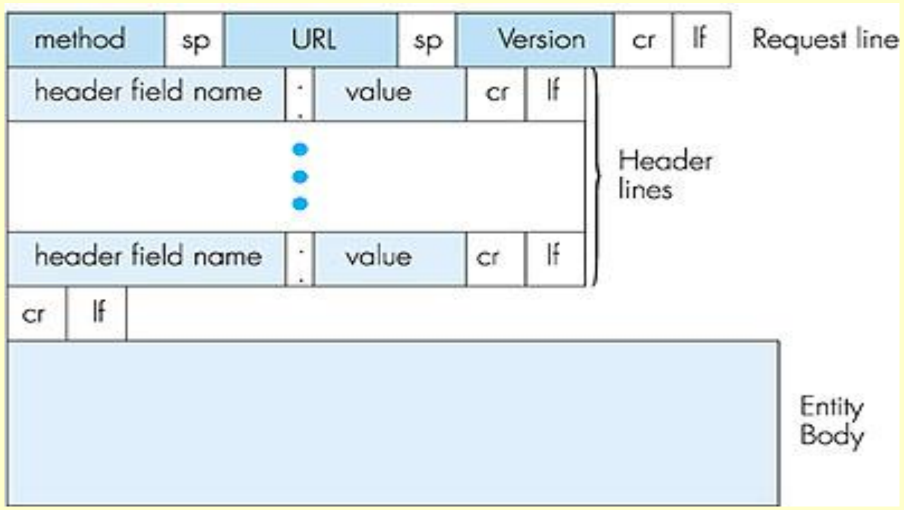
Appendix A
HTTP Message Headers

Request Message



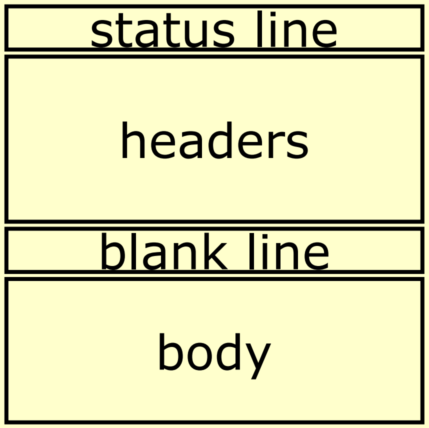
request methods:

DELETE, GET, HEAD, POST, PUT, TRACE

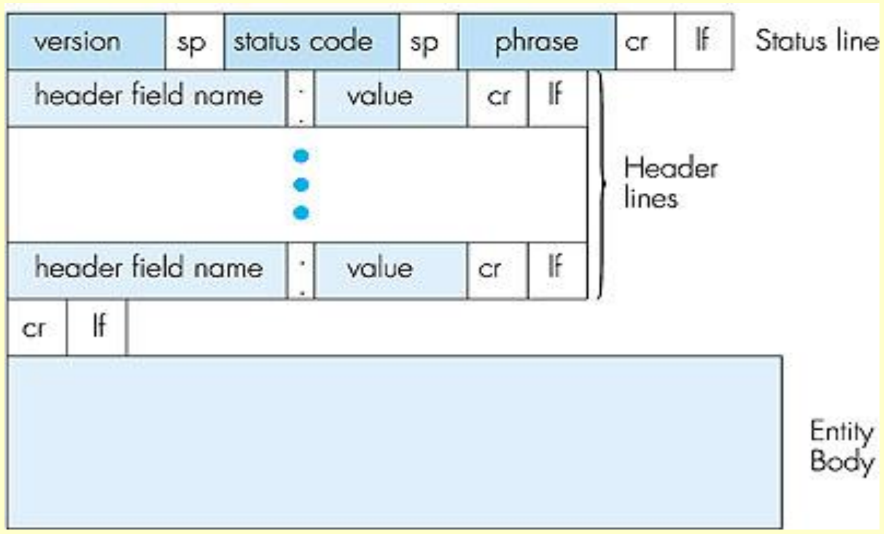


```
GET /pub/index.html HTTP/1.0
Date: Wed, 20 Mar 2002 10:00:02 GMT
Pragma: no-cache
From: amer@udel.edu
User-Agent: Mozilla/4.03
```

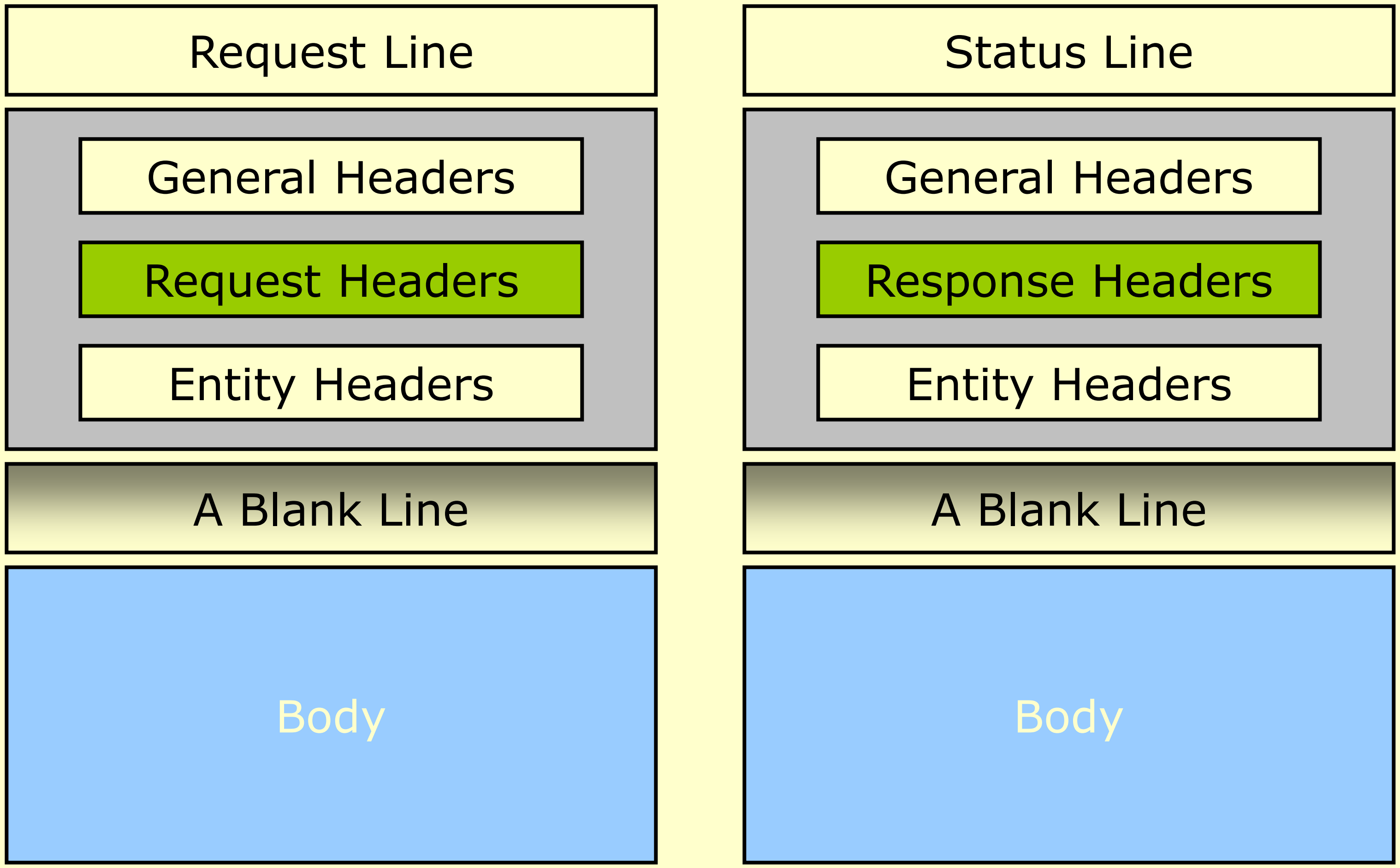
Response Message



```
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
```



Headers


Request Line
General Headers
Request Headers
Entity Headers
A Blank Line
Body
Status Line
General Headers
Response Headers
Entity Headers
A Blank Line
Body

Headers

General Headers

HTTP/1.0 & HTTP/1.1	New in HTTP/1.1
Date	Cache Control
Pragma	Connection
	Trailer
	Transfer-Encoding
	Upgrade
	Via
	Warning

Request Headers

HTTP/1.0 & HTTP/1.1	New in HTTP/1.1
Authorization	Accept
From	Accept-Charset
If-Modified-Since	Accept-Encoding
Referer	Accept Language
User-Agent	Expect
	Host
	If-Match
	If-None-Match
	If-Range
	If-Unmodified-Since
	Max-Forwards
	Proxy-Authorization
	Range
	TE

🟧 Headers present in HTTP/1.0 & HTTP/1.1

🟦 New Headers added in HTTP/1.1

Headers

Response Headers

🟧
- Location
- Server
- WWW-Authenticate

🟦
- Accept-Ranges
- Age
- ETag
- Proxy-Authenticate
- Retry-After
- Vary

Entity Headers

🟧
- Allow
- Content-Encoding
- Content-Length
- Content-Type
- Expires
- Last-Modified
- extension-header

🟦
- Content-Language
- Content-Location
- Content-MD5
- Content-Range

🟧 Headers present in HTTP/1.0 & HTTP/1.1

🟦 New Headers added in HTTP/1.1

End of Presentation