



SNS COLLEGE OF TECHNOLOGY

Coimbatore-35
An Autonomous Institution

Accredited by NBA – AICTE and Accredited by NAAC – UGC with 'A+' Grade
Approved by AICTE, New Delhi & Affiliated to Anna University, Chennai

DEPARTMENT OF AIML

AI - WEB TECHNOLOGY

III YEAR - VI SEM

UNIT 1 – Web Site Basics And HTML

TOPIC 5 - Web Clients Web Servers





Web Clients



- ▶ Many possible web clients:
 - ▶ Text-only “browser” (lynx)
 - ▶ Mobile phones
 - ▶ **Robots** (software-only clients, e.g., search engine “crawlers”)
 - ▶ etc.
- ▶ We will focus on traditional web browsers



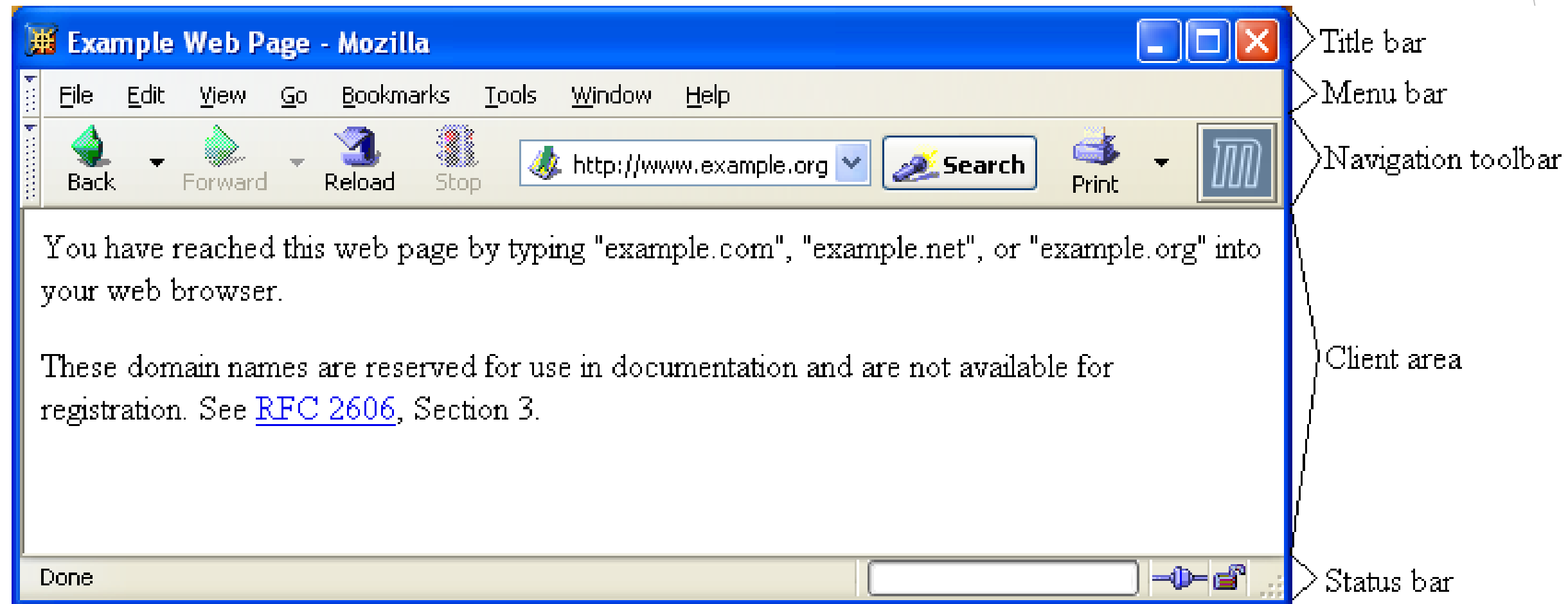
Web Browsers

- ▶ First graphical browser running on general-purpose platforms: Mosaic (1993)





Web Browsers





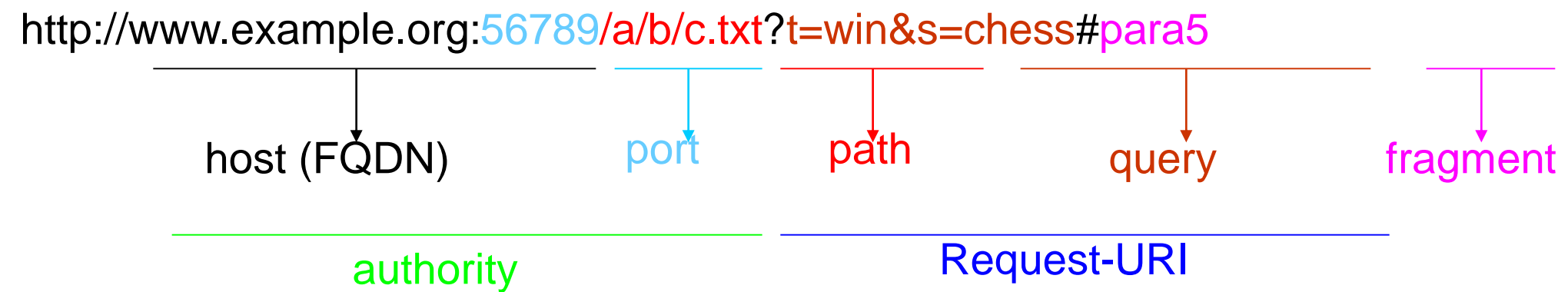
Web Browsers



- ▶ Primary tasks:
 - ▶ Convert web addresses (URL's) to HTTP requests
 - ▶ Communicate with web servers via HTTP
 - ▶ **Render** (appropriately display) documents returned by a server



HTTP URL's



- ▶ Browser uses authority to connect via TCP
- ▶ Request-URI included in start line (/ used for path if none supplied)
- ▶ Fragment identifier not sent to server (used to scroll browser client area)



Web Browsers



- ▶ Standard features
 - ▶ Save web page to disk
 - ▶ Find string in page
 - ▶ Fill forms automatically (passwords, CC numbers, ...)
 - ▶ Set preferences (language, character set, cache and HTTP parameters)
 - ▶ Modify display style (e.g., increase font sizes)
 - ▶ Display raw HTML and HTTP header info (e.g., Last-Modified)
 - ▶ Choose browser themes (skins)
 - ▶ View history of web addresses visited
 - ▶ Bookmark favorite pages for easy return



Web Browsers



- ▶ Additional functionality:
 - ▶ Execution of **scripts** (e.g., drop-down menus)
 - ▶ **Event** handling (e.g., mouse clicks)
 - ▶ GUI for **controls** (e.g., buttons)
 - ▶ **Secure communication** with servers
 - ▶ Display of non-HTML documents (e.g., PDF) via **plug-ins**



Web Servers

- ▶ Basic functionality:
 - ▶ Receive HTTP request via TCP
 - ▶ Map Host header to specific **virtual host** (one of many host names sharing an IP address)
 - ▶ Map Request-URI to specific resource associated with the virtual host
 - ▶ File: Return file in HTTP response
 - ▶ Program: Run program and return output in HTTP response
 - ▶ Map type of resource to appropriate MIME type and use to set Content-Type header in HTTP response
 - ▶ Log information about the request and response



Web Servers

- ▶ httpd: UIUC, primary Web server c. 1995
- ▶ Apache: “A patchy” version of httpd, now the most popular server (esp. on Linux platforms)
- ▶ IIS: Microsoft Internet Information Server
- ▶ Tomcat:
 - ▶ Java-based
 - ▶ Provides **container** (Catalina) for running Java **servlets** (HTML-generating programs) as back-end to Apache or IIS
 - ▶ Can run stand-alone using Coyote HTTP front-end



Web Servers



- ▶ Some Coyote communication parameters:
 - ▶ Allowed/blocked IP addresses
 - ▶ Max. simultaneous active TCP connections
 - ▶ Max. queued TCP connection requests
 - ▶ “Keep-alive” time for inactive TCP connections
- ▶ Modify parameters to **tune** server performance



Web Servers



- ▶ Some Catalina container parameters:
 - ▶ Virtual host names and associated ports
 - ▶ Logging preferences
 - ▶ Mapping from Request-URI's to server resources
 - ▶ Password protection of resources
 - ▶ Use of server-side caching



Tomcat Web Server



- ▶ HTML-based server administration
- ▶ Browse to <http://localhost:8080> and click on Server Administration link
 - ▶ localhost is a special host name that means “this machine”



Tomcat Web Server





Tomcat Web Server





Tomcat Web Server

The screenshot shows the Tomcat Server Administration tool interface. The title bar reads "Tomcat Server Administration - Mozilla". The main header area contains the text "TOMCAT WEB SERVER ADMINISTRATION TOOL" and two buttons: "Commit Changes" and "Log Out".

The left sidebar shows a tree view of the server configuration:

- Tomcat Server
 - Service (Java Web Services Developer Pack)
 - Connector (8080)
 - Host (localhost)
 - Logger for Service (Java Web Services Developer Pack)
 - Realm for Service (Java Web Services Developer Pack)
 - Valve for Service (Java Web Services Developer Pack)

Connector (8080) Connector Actions —Available Actions—

—Available Actions—

Save Reset

| General | |
|---------------------|------|
| Type: | HTTP |
| Scheme: | http |
| Debug Level: | 0 |
| Enable DNS Lookups: | True |



Tomcat Web Server



- ▶ Some Connector fields:
 - ▶ Port Number: port “owned” by this connector
 - ▶ Max Threads: max connections processed simultaneously
 - ▶ Connection Timeout: keep-alive time



Tomcat Web Server

The screenshot shows the Tomcat Server Administration tool interface. The title bar reads "Tomcat Server Administration - Mozilla". The main header area contains the text "TOMCAT WEB SERVER ADMINISTRATION TOOL" and two buttons: "Commit Changes" and "Log Out".

The left sidebar shows a tree view of the server configuration:

- Tomcat Server
 - Service (Java Web Services Developer Pack)
 - Connector (8080)
 - Host (localhost)**
 - Logger for Service (Java Web Services Developer Pack)
 - Realm for Service (Java Web Services Developer Pack)
 - Valve for ...

Host Properties

| Property | Value |
|--------------------|-------------------------------|
| Name: | localhost |
| Application Base: | webapps |
| Auto Deploy: | <input type="checkbox"/> True |
| Debug Level: | <input type="checkbox"/> 0 |
| Deploy On Startup: | <input type="checkbox"/> True |



Tomcat Web Server



- ▶ Each Host is a virtual host (can have multiple per Connector)
- ▶ Some fields:
 - ▶ Host: localhost or a fully qualified domain name
 - ▶ **Application Base**: directory (may be path relative to JWSDP installation directory) containing resources associated with this Host



Tomcat Web Server

Tomcat Server Administration - Mozilla

TOMCAT WEB SERVER ADMINISTRATION TOOL

Commit Changes Log Out

- Tomcat Server
 - Service (Java Web Services Developer Pack)
 - Connector (8080)
 - Host (localhost)
 - Context (/)**
 - Context (/RegistryServer)
 - Context (/Xindice)
 - Context (/admin)
 - Context (/gs)
 - Context (/jaxrpc-HelloWorld)
 - Context (/jsf-cardemo)
 - Context (/jsf-components)
 - Context (/jsf-portal)
 - Context (/jsf-portal2)
 - Context (/jsf-portal3)
 - Context (/jsf-portal4)
 - Context (/jsf-portal5)
 - Context (/jsf-portal6)
 - Context (/jsf-portal7)
 - Context (/jsf-portal8)
 - Context (/jsf-portal9)
 - Context (/jsf-portal10)
 - Context (/jsf-portal11)
 - Context (/jsf-portal12)
 - Context (/jsf-portal13)
 - Context (/jsf-portal14)
 - Context (/jsf-portal15)
 - Context (/jsf-portal16)
 - Context (/jsf-portal17)
 - Context (/jsf-portal18)
 - Context (/jsf-portal19)
 - Context (/jsf-portal20)

Context (/) Context Actions —Available Actions—

Save Reset

Context Properties

| Property | Value |
|----------------|---------------------------|
| Cookies: | True |
| Cross Context: | False |
| Debug Level: | 0 |
| Document Base: | C:\jwsdp-1.3\webapps\ROOT |



Tomcat Web Server

- ▶ **Context** provides mapping from Request-URI path to a **web application**
- ▶ **Document Base** field is directory (possibly relative to Application Base) that contains resources for this web application
- ▶ For this example, browsing to `http://localhost:8080/` returns resource from `c:\jwsdp-1.3\webapps\ROOT`
 - ▶ Returns `index.html` (standard **welcome file**)



Tomcat Web Server



- ▶ **Access log** records HTTP requests
- ▶ Parameters set using `AccessLogValve`
- ▶ Default location: `logs/access_log.*` under JWS DP installation directory
- ▶ Example “common” log format entry (one line):
`www.example.org - admin`
`[20/Jul/2005:08:03:22 -0500]`
`"GET /admin/frameset.jsp HTTP/1.1"`
`200 920`



Tomcat Web Server



- ▶ Other logs provided by default in JWSDP:
 - ▶ **Message log** messages sent to log service by web applications or Tomcat itself
 - ▶ `logs/jwsdp_log.*`: default message log
 - ▶ `logs/localhost_admin_log.*`: message log for web apps within /admin context
 - ▶ `System.out` and `System.err` output (exception traces often found here):
 - ▶ `logs/launcher.server.log`



Tomcat Web Server



- ▶ Access control:
 - ▶ Password protection (e.g., admin pages)
 - ▶ Users and **roles** defined in `conf/tomcat-users.xml`
 - ▶ Deny access to machines
 - ▶ Useful for denying access to certain users by denying access from the machines they use
 - ▶ List of denied machines maintained in RemoteHostValve (deny by host name) or RemoteAddressValve (deny by IP address)



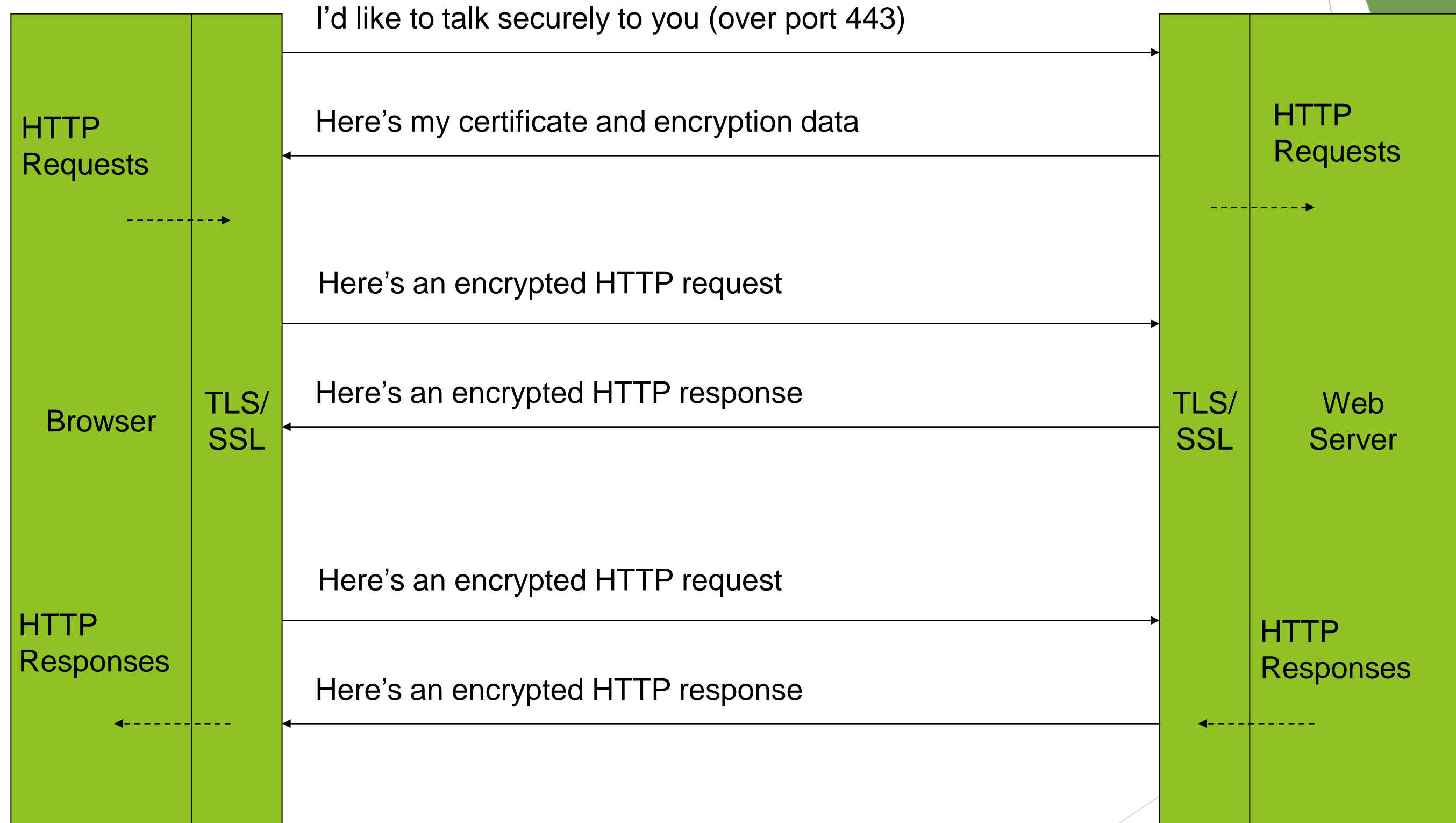
Secure Servers



- ▶ Since HTTP messages typically travel over a public network, private information (such as credit card numbers) should be **encrypted** to prevent **eavesdropping**
- ▶ **https** URL scheme tells browser to use encryption
- ▶ Common encryption standards:
 - ▶ Secure Socket Layer (SSL)
 - ▶ Transport Layer Security (**TLS**)



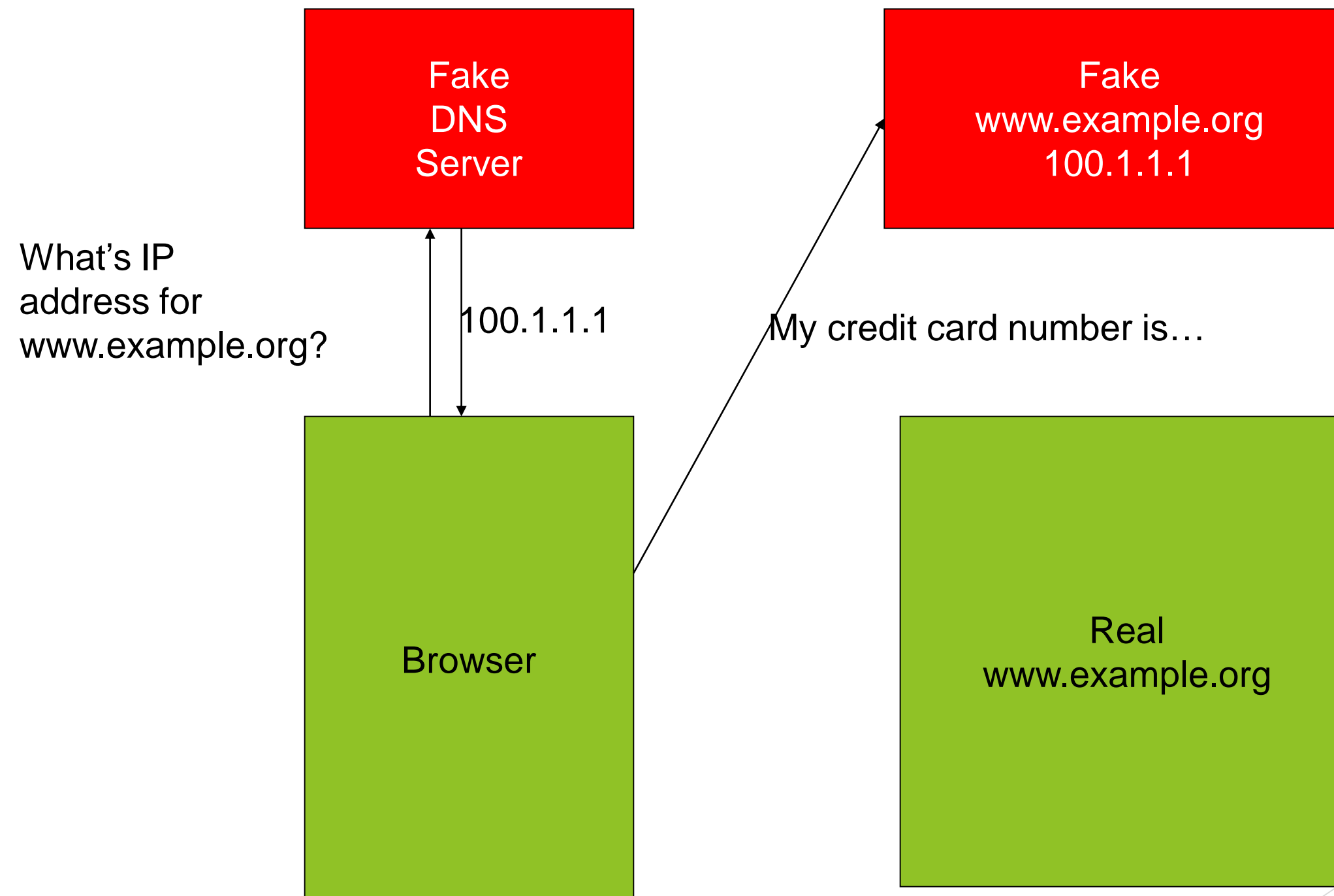
Secure Servers





Secure Servers

Man-in-the-Middle Attack





Secure Servers

Preventing Man-in-the-Middle

