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 ARTIFICIAL INTELLIGENCE

AI IN WEB TECHNOLOGY
III YEAR - VI SEM

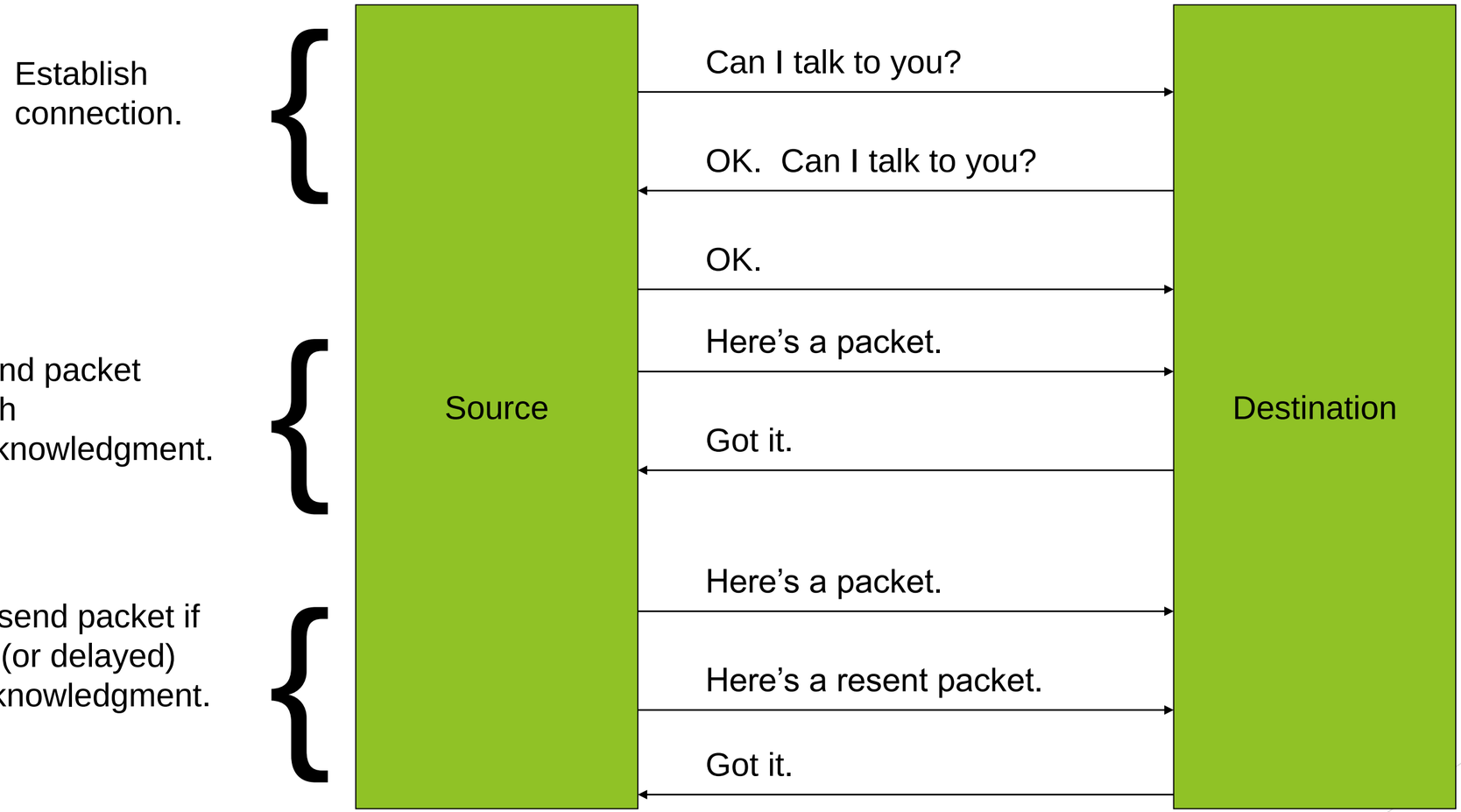
UNIT 1 – INTRODUCTION TO WEB TECHNOLOGY AND DESIGN

INTRODUCTION TO WEB TECHNOLOGY AND DESIGN

Transmission Control Protocol (TCP)

- Limitations of IP:
  - No guarantee of packet delivery (packets can be dropped)
  - Communication is one-way (source to destination)
- TCP adds concept of a connection on top of IP
  - Provides guarantee that packets delivered
  - Provide two-way (full duplex) communication

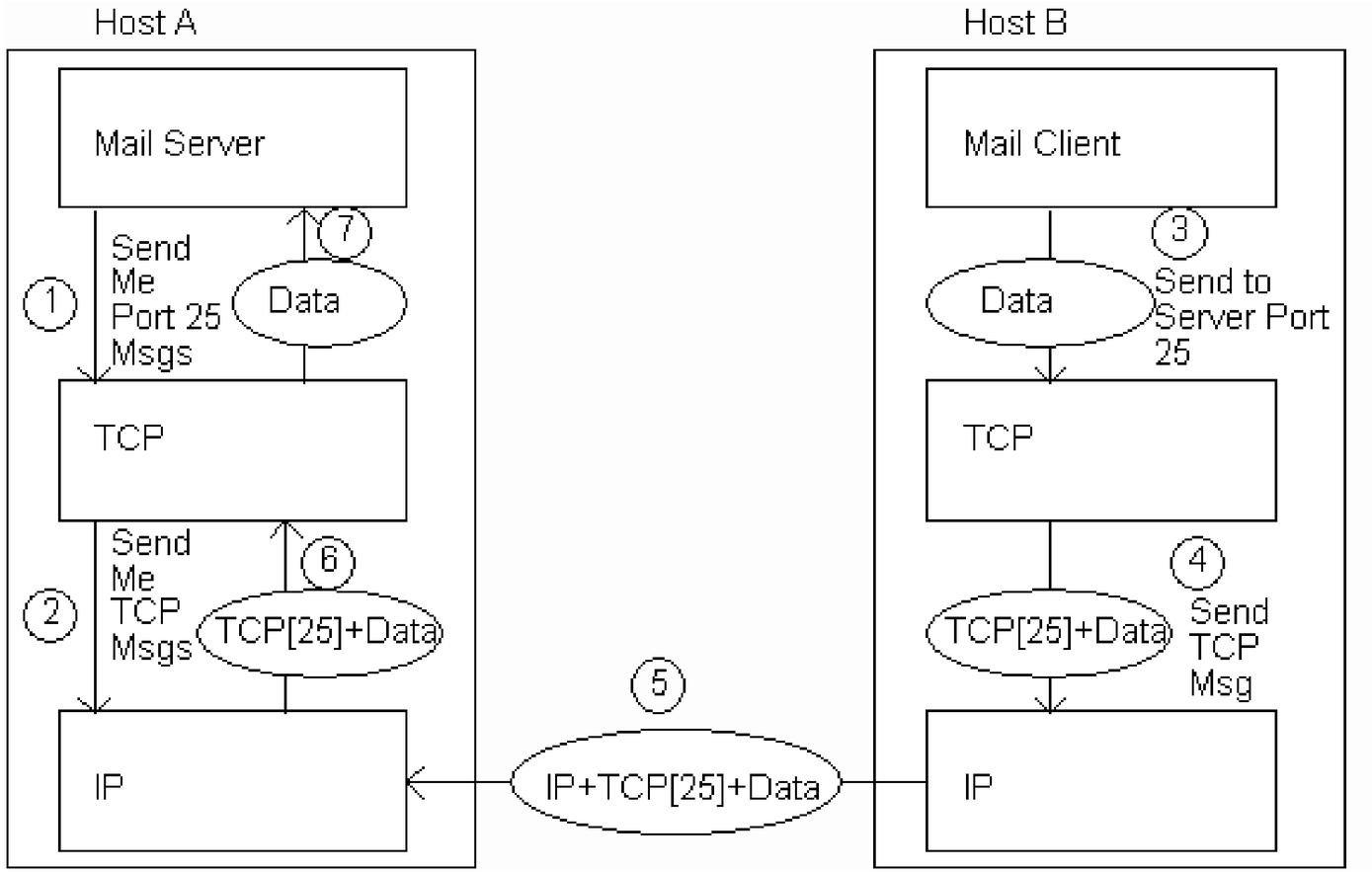
TCP


Establish connection.
Send packet with acknowledgment.
Resend packet if no (or delayed) acknowledgment.
Source
Destination
Can I talk to you?
OK. Can I talk to you?
OK.
Here's a packet.
Got it.
Here's a packet.
Here's a resent packet.
Got it.

TCP

- TCP also adds concept of a port
  - TCP header contains port number representing an application program on the destination computer
  - Some port numbers have standard meanings
    - Example: port 25 is normally used for email transmitted using the Simple Mail Transfer Protocol (SMTP)
  - Other port numbers are available first-come-first served to any application

TCP


Host A
Mail Server
Send Me Port 25 Msgs
1
7
Data
TCP
Send Me TCP Msgs
2
6
TCP[25]+Data
IP
5
IP+TCP[25]+Data
Host B
Mail Client
3
Data
Send to Server Port 25
TCP
4
TCP[25]+Data
Send TCP Msg
IP

User Datagram Protocol (UDP)

- Like TCP in that:
  - Builds on IP
  - Provides port concept
- Unlike TCP in that:
  - No connection concept
  - No transmission guarantee
- Advantage of UDP vs. TCP:
  - Lightweight, so faster for one-time messages

Domain Name Service (DNS)

- DNS is the “phone book” for the Internet
  - Map between host names and IP addresses
  - DNS often uses UDP for communication
- Host names
  - Labels separated by dots, e.g., www.example.org
  - Final label is *top-level domain*
    - Generic: .com, .org, etc.
    - Country-code: .us, .il, etc.

DNS

- Domains are divided into second-level domains, which can be further divided into subdomains, etc.
  - E.g., in www.example.com, example is a second-level domain
- A host name plus domain name information is called the fully qualified domain name of the computer
  - Above, www is the host name, www.example.com is the FQDN

DNS

- `nslookup` program provides command-line access to DNS (on most systems)
- looking up a host name given an IP address is known as a reverse lookup
  - Recall that single host may have multiple IP addresses.
  - Address returned is the canonical IP address specified in the DNS system.

DNS

- `ipconfig` (on windows) can be used to find the IP address (addresses) of your machine
- *`ipconfig /displaydns`* displays the contents of the DNS Resolver Cache (*`ipconfig /flushdns`* to flush it)

Analogy to Telephone Network

- IP ~ the telephone network
- TCP ~ calling someone who answers, having a conversation, and hanging up
- UDP ~ calling someone and leaving a message
- DNS ~ directory assistance

Higher-level Protocols

- Many protocols build on TCP
  - Telephone analogy: TCP specifies how we initiate and terminate the phone call, but some other protocol specifies how we carry on the actual conversation
- Some examples:
  - SMTP (email)
  - FTP (file transfer)
  - HTTP (transfer of Web documents)

World Wide Web

- Originally, one of several systems for organizing Internet-based information
  - Competitors: WAIS, Gopher, ARCHIE
- Distinctive feature of Web: support for hypertext (text containing links)
  - Communication via Hypertext Transfer Protocol (HTTP)
  - Document representation using Hypertext Markup Language (HTML)

World Wide Web

- The Web is the collection of machines (Web servers) on the Internet that provide information, particularly HTML documents, via HTTP.
- Machines that access information on the Web are known as Web clients.
- A Web browser is software used by an end user to access the Web.

Hypertext Transfer Protocol (HTTP)

- HTTP is based on the request-response communication model:
  - Client sends a request
  - Server sends a response
- HTTP is a stateless protocol:
  - The protocol does not require the server to remember anything about the client between requests.

HTTP

- Normally implemented over a TCP connection (80 is standard port number for HTTP)
- Typical browser-server interaction:
  - User enters Web address in browser
  - Browser uses DNS to locate IP address
  - Browser opens TCP connection to server
  - Browser sends HTTP request over connection
  - Server sends HTTP response to browser over connection
  - Browser displays body of response in the client area of the browser window

HTTP

- The information transmitted using HTTP is often entirely text
- Can use the Internet's Telnet protocol to simulate browser request and view server response

HTTP

Connect

```
$ telnet www.example.org 80
Trying 192.0.34.166...
Connected to www.example.com (192.0.34.166).
Escape character is '^]'.
GET / HTTP/1.1
Host: www.example.org
```

Send Request

Receive Response

```
HTTP/1.1 200 OK
Date: Thu, 09 Oct 2020 20:30:49 GMT
…
```

HTTP Request

- Structure of the request:
  - start line
  - header field(s)
  - blank line
  - optional body

HTTP Request

- Structure of the request:
  - **start line**
  - header field(s)
  - blank line
  - optional body

HTTP Request

- Start line
  - Example: `GET / HTTP/1.1`
- Three space-separated parts:
  - HTTP request method
  - Request-URI (Uniform Resource Identifier)
  - HTTP version

HTTP Request

- Start line
  - Example: `GET / HTTP/1.1`
- Three space-separated parts:
  - HTTP request method
  - **Request-URI**
  - HTTP version

HTTP Request

- Uniform Resource Identifier (URI)
  - Syntax: *scheme : scheme-depend-part*
    - Ex: In http://www.example.com/
      the scheme is http
  - Request-URI is the portion of the requested URI that follows the host name (which is supplied by the required Host header field)
    - Ex: / is Request-URI portion of http://www.example.com/

URI

- URI's are of two types:
  - Uniform Resource Name (URN)
    - Can be used to identify resources with unique names, such as books (which have unique ISBN's)
    - Scheme is `urn`
  - Uniform Resource Locator (URL)
    - Specifies location at which a resource can be found
    - In addition to `http`, some other URL schemes are `https`, `ftp`, `mailto`, and `file`

HTTP Request

- Start line
  - Example: `GET / HTTP/1.1`
- Three space-separated parts:
  - **HTTP request method**
  - Request-URI
  - HTTP version

HTTP Request

- Common request methods:
  - GET
    - Used if link is clicked or address typed in browser
    - No body in request with GET method
  - POST
    - Used when submit button is clicked on a form
    - Form information contained in body of request
  - HEAD
    - Requests that only header fields (no body) be returned in the response

HTTP Request

- Structure of the request:
  - start line
  - **header field(s)**
  - blank line
  - optional body

HTTP Request

- Header field structure:
  - *field name* : *field value*
- Syntax
  - Field name is not case sensitive
  - Field value may continue on multiple lines by starting continuation lines with white space
  - Field values may contain MIME types, quality values, and wildcard characters (*'s)

Multipurpose Internet Mail Extensions (MIME)

- Convention for specifying content type of a message
  - In HTTP, typically used to specify content type of the body of the response
- MIME content type syntax:
  - *top-level type / subtype*
- Examples: text/html, image/jpeg

HTTP Quality Values and Wildcards

- Example header field with quality values:

```
accept:
   text/xml,text/html;q=0.9,
   text/plain;q=0.8, image/jpeg,
   image/gif;q=0.2,*/*;q=0.1
```

- Quality value applies to all preceding items
- Higher the value, higher the preference
- Note use of wildcards to specify quality 0.1 for any MIME type not specified earlier

HTTP Request

- Common header fields:
  - Host: host name from URL (required)
  - User-Agent: type of browser sending request
  - Accept: MIME types of acceptable documents
  - Connection: value `close` tells server to close connection after single request/response
  - Content-Type: MIME type of (POST) body, normally application/x-www-form-urlencoded
  - Content-Length: bytes in body
  - Referer: URL of document containing link that supplied URI for this HTTP request

HTTP Response

- Structure of the response:
  - status line
  - header field(s)
  - blank line
  - optional body

HTTP Response

- Structure of the response:
  - **status line**
  - header field(s)
  - blank line
  - optional body

HTTP Response

- Status line
  - Example: HTTP/1.1 200 OK
- Three space-separated parts:
  - HTTP version
  - status code
  - reason phrase (intended for human use)

HTTP Response

- Status code
  - Three-digit number
  - First digit is class of the status code:
    - 1=Informational
    - 2=Success
    - 3=Redirection (alternate URL is supplied)
    - 4=Client Error
    - 5=Server Error
  - Other two digits provide additional information
  - See http://www.w3.org/Protocols/rfc2616/rfc2616-sec10.html

HTTP Response

- Structure of the response:
  - status line
  - **header field(s)**
  - blank line
  - optional body

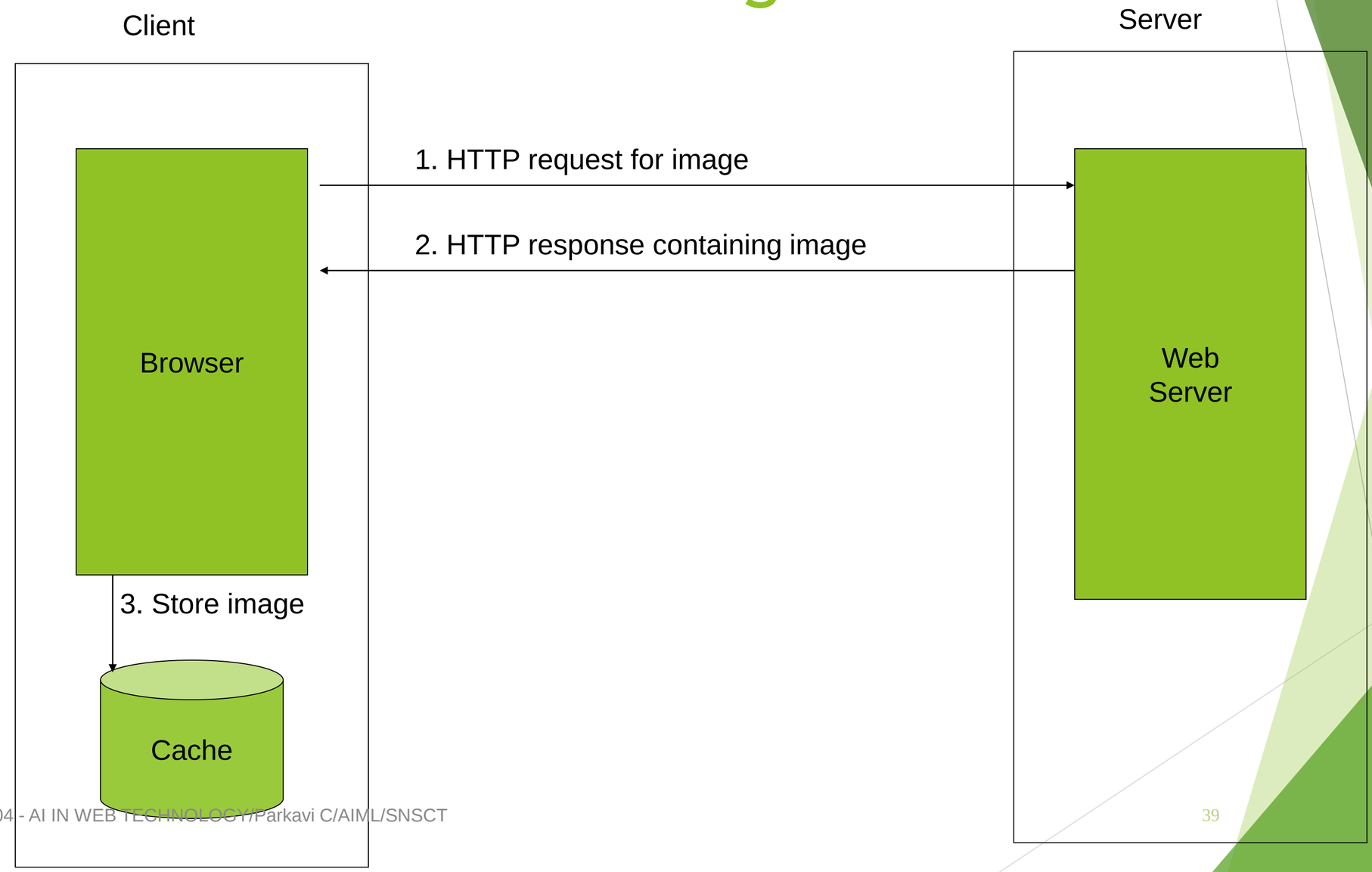
HTTP Response

- Common header fields:
  - Connection, Content-Type, Content-Length
  - Date: date and time at which response was generated (required)
  - Location: alternate URI if status is redirection
  - Last-Modified: date and time the requested resource was last modified on the server
  - Expires: date and time after which the client's copy of the resource will be out-of-date
  - ETag: a unique identifier for this version of the requested resource (changes if resource changes)

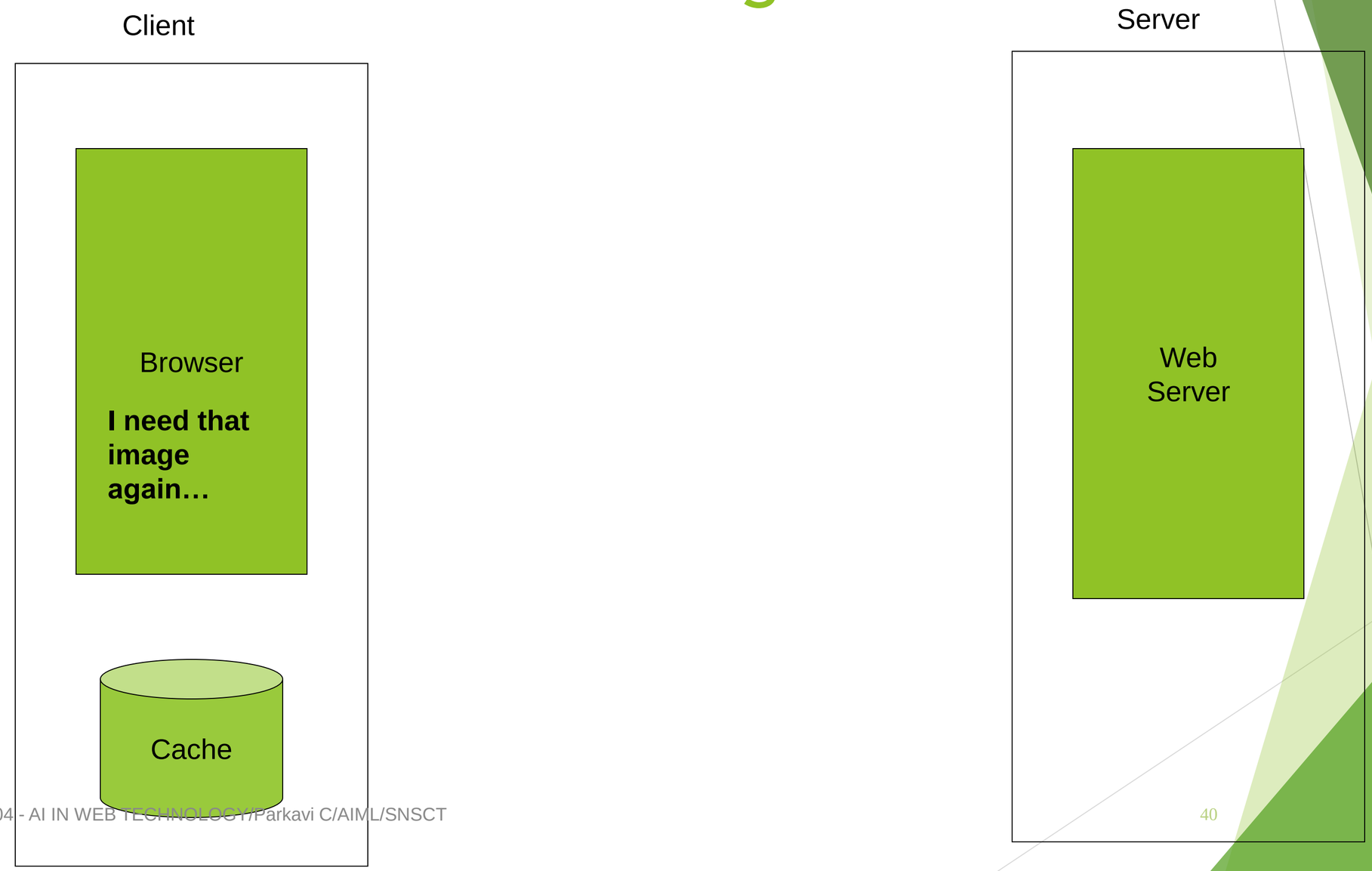
Client Caching

- A cache is a local copy of information obtained from some other source
- Most web browsers use cache to store requested resources so that subsequent requests to the same resource will not necessarily require an HTTP request/response
  - Ex: icon appearing multiple times in a Web page

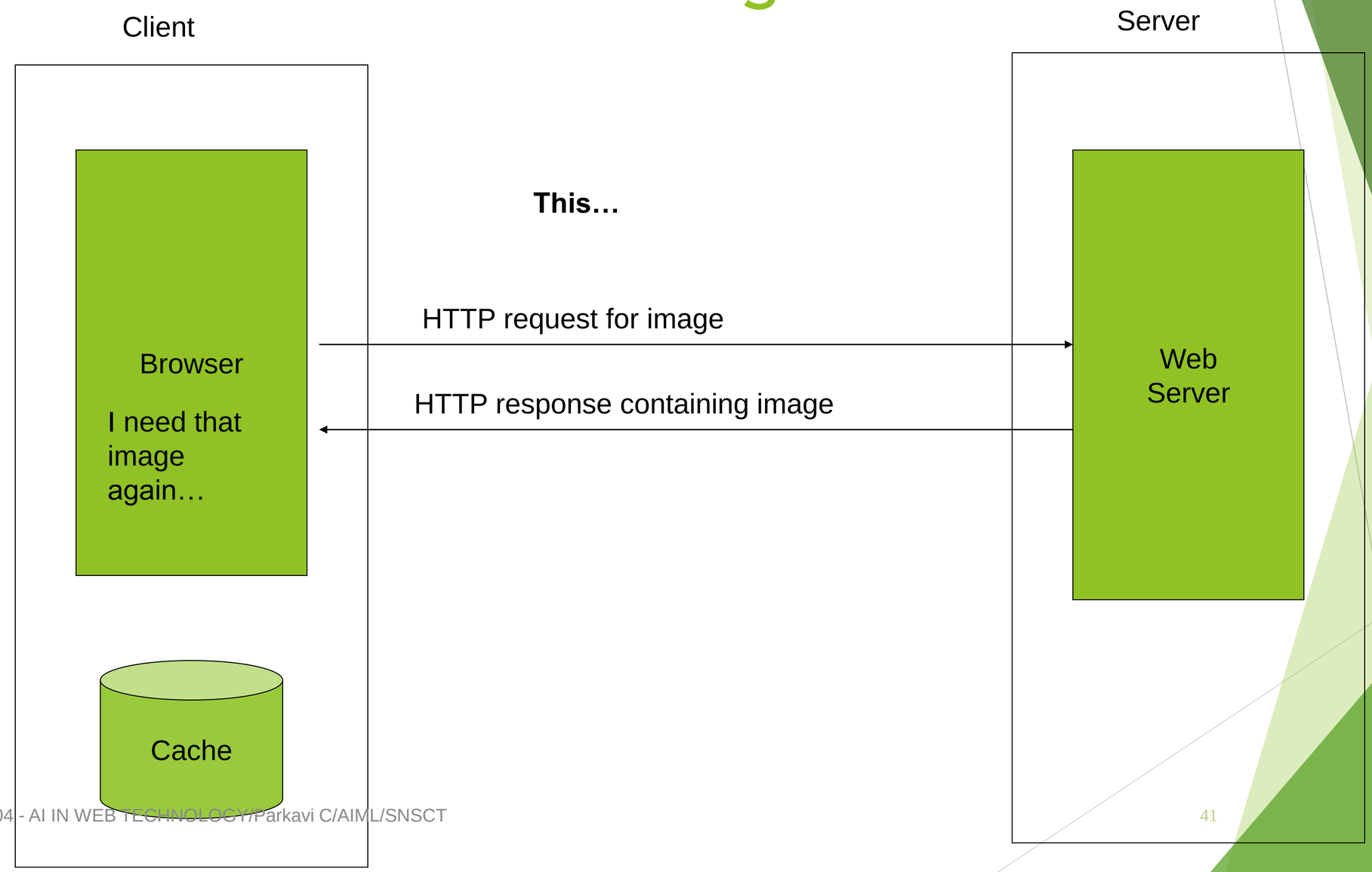
Client Caching



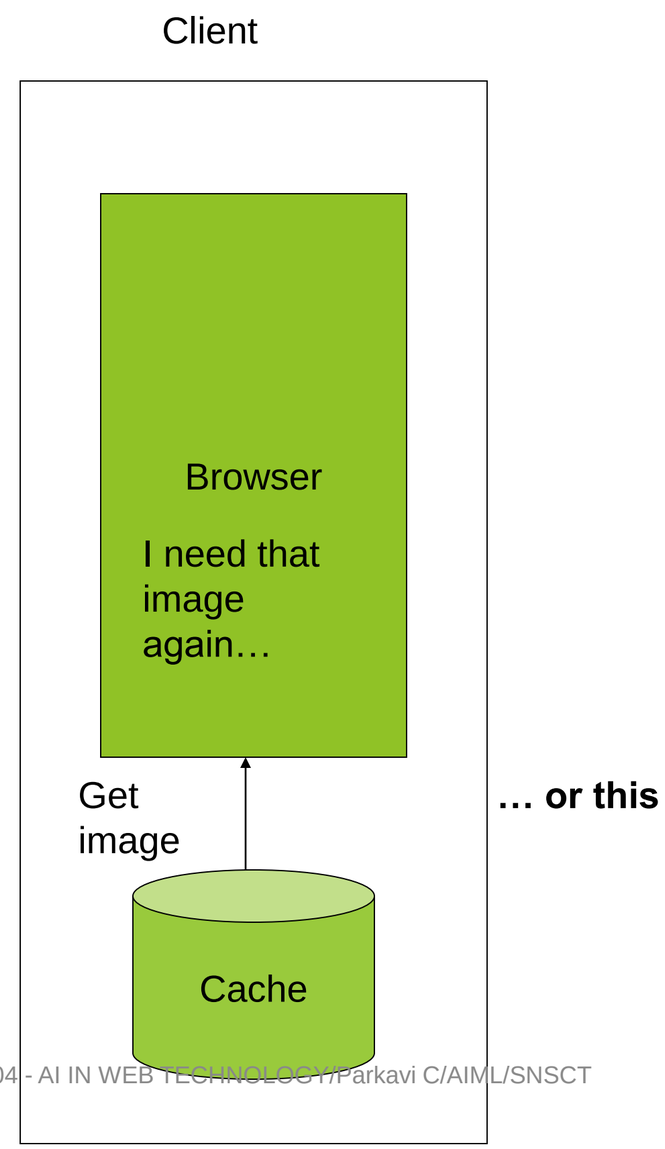
Client Caching



Client Caching



Client Caching



Client Caching

- Cache advantages
  - (Much) faster than HTTP request/response
  - Less network traffic
  - Less load on server
- Cache disadvantage
  - Cached copy of resource may be invalid (inconsistent with remote version)

Client Caching

- Validating cached resource:
  - Send HTTP HEAD request and check Last-Modified or ETag header in response
  - Compare current date/time with Expires header sent in response containing resource
  - If no Expires header was sent, use heuristic algorithm to estimate value for Expires
    - Ex: Expires = 0.01 * (Date – Last-Modified) + Date

Character Sets

- Every document is represented by a string of integer values (code points)
- The mapping from code points to characters is defined by a character set
- Some header fields have character set values:
  - Accept-Charset: request header listing character sets that the client can recognize
    - Ex: accept-charset: ISO-8859-1,utf-8;q=0.7,*;q=0.5
  - Content-Type: can include character set used to represent the body of the HTTP message
    - Ex: Content-Type: text/html; charset=UTF-8

Character Sets

- Technically, many “character sets” are actually character encodings
  - An encoding represents code points using variable-length byte strings
  - Most common examples are Unicode-based encodings UTF-8 and UTF-16
- IANA maintains complete list of Internet-recognized character sets/encodings

Character Sets

- Typical US PC produces ASCII documents
- US-ASCII character set can be used for such documents, but is not recommended
- UTF-8 and ISO-8859-1 are supersets of US-ASCII and provide international compatibility
  - UTF-8 can represent all ASCII characters using a single byte each and arbitrary Unicode characters using up to 4 bytes each
  - ISO-8859-1 is 1-byte code that has many characters common in Western European languages, such as é

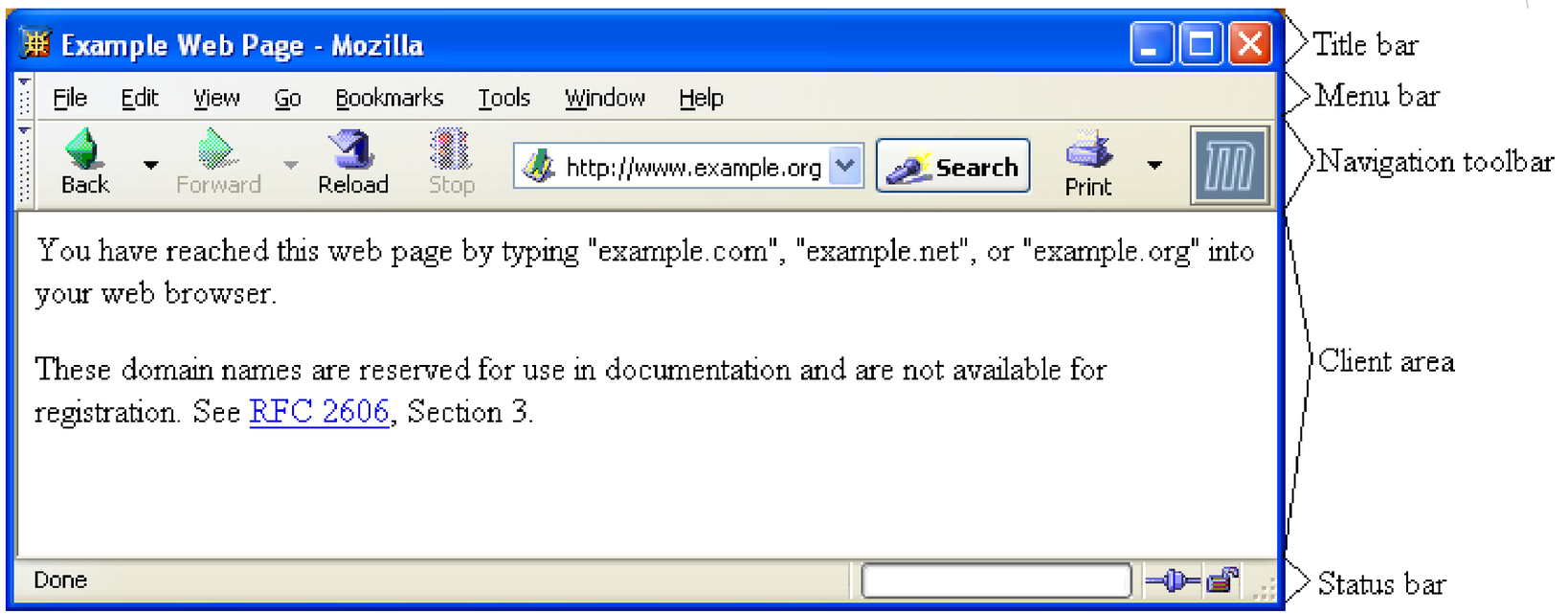
Web Clients

- 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)

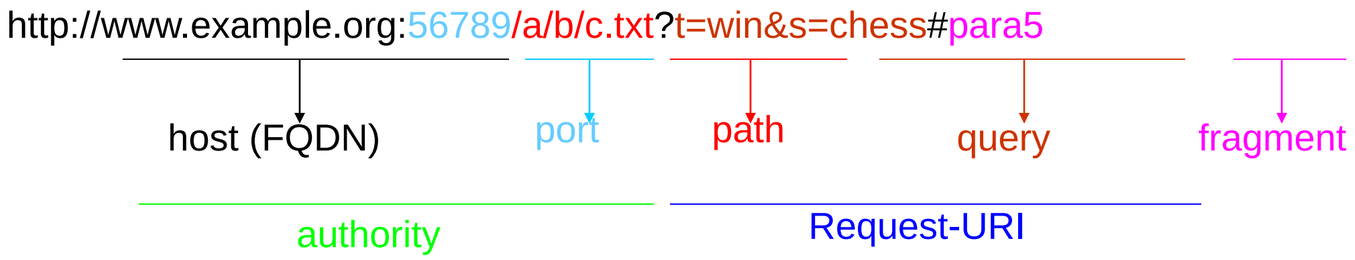
Web Browsers



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

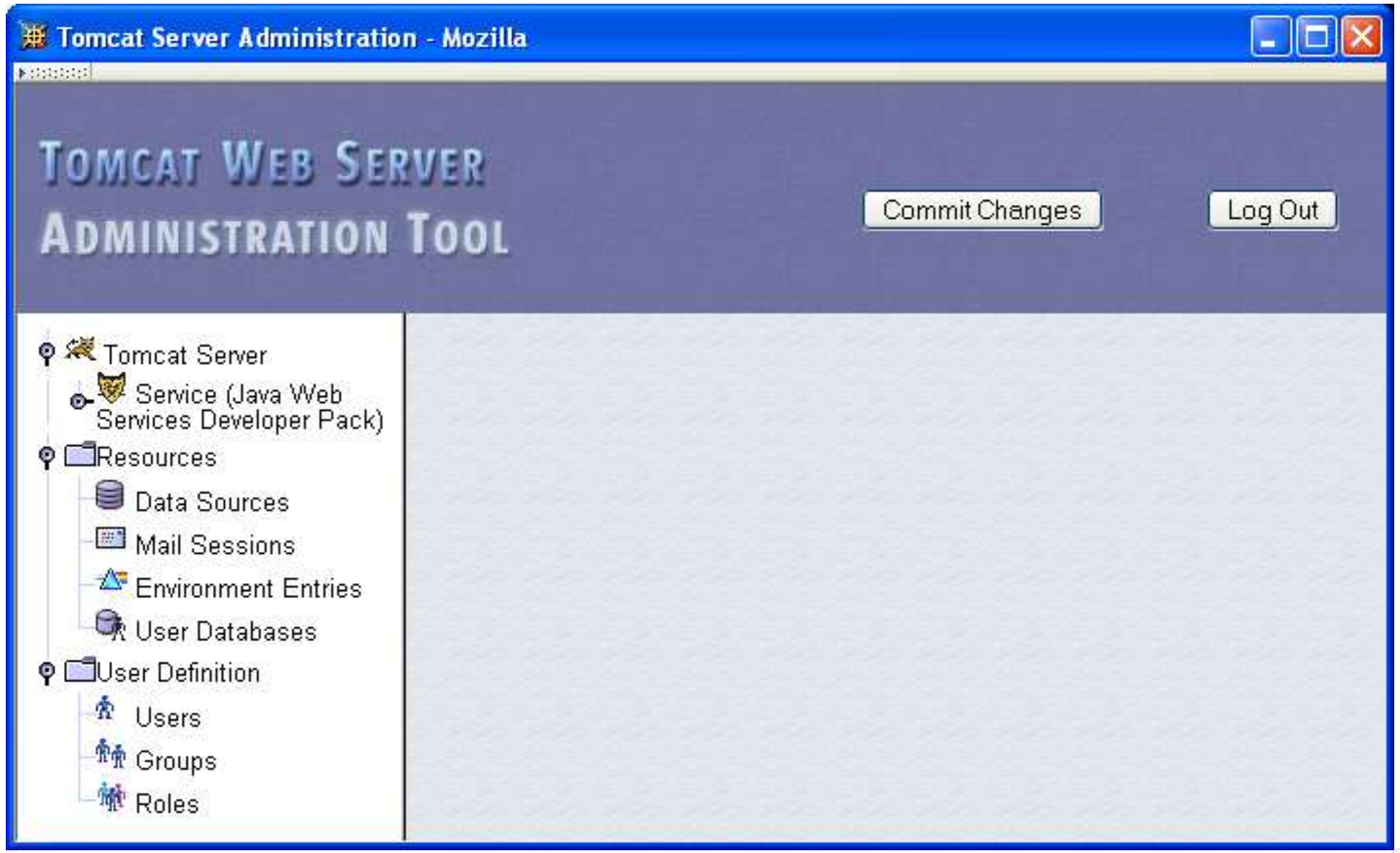
Web Servers

- 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



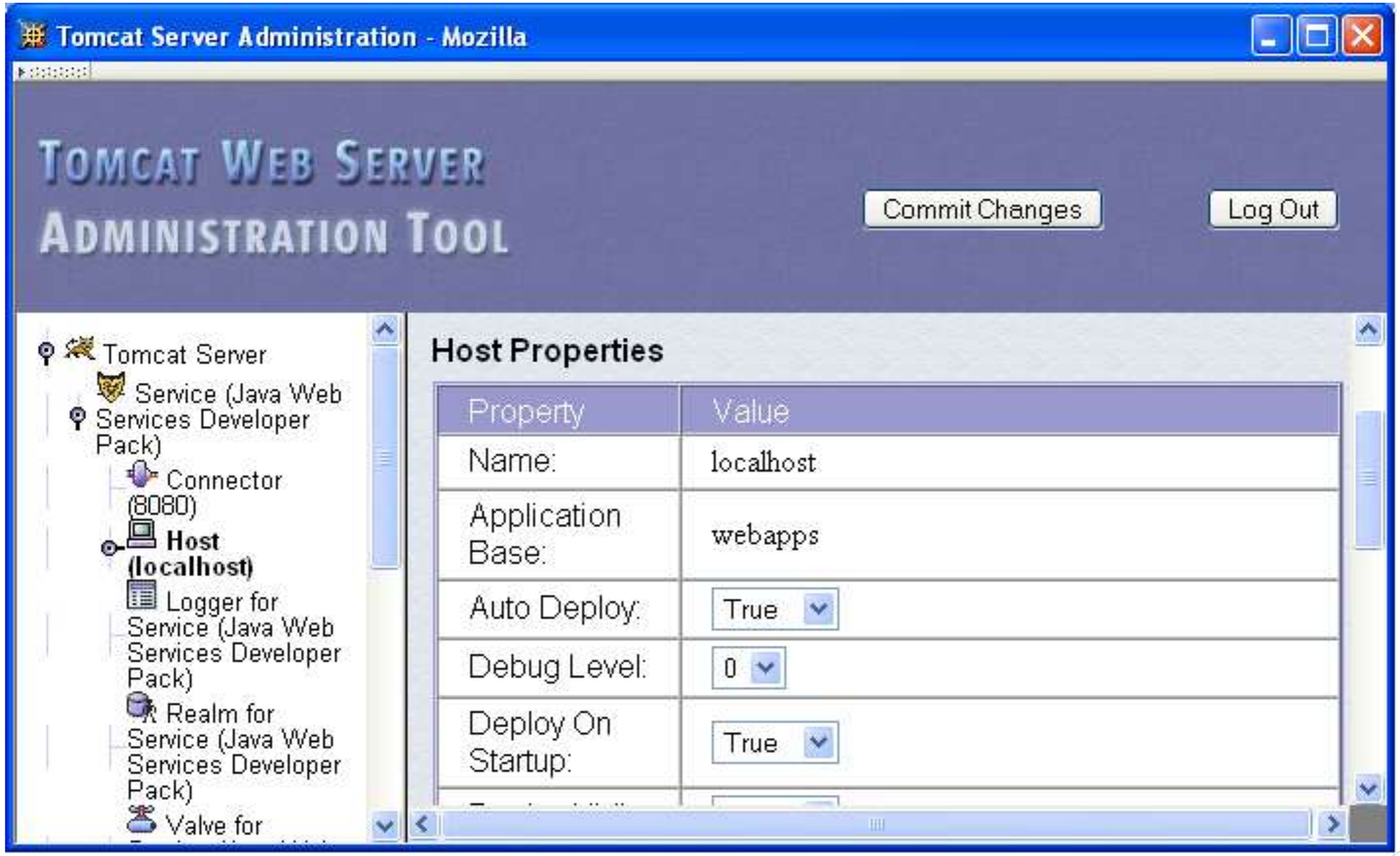
Tomcat Web Server

Tomcat Web Server

Tomcat Web Server

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

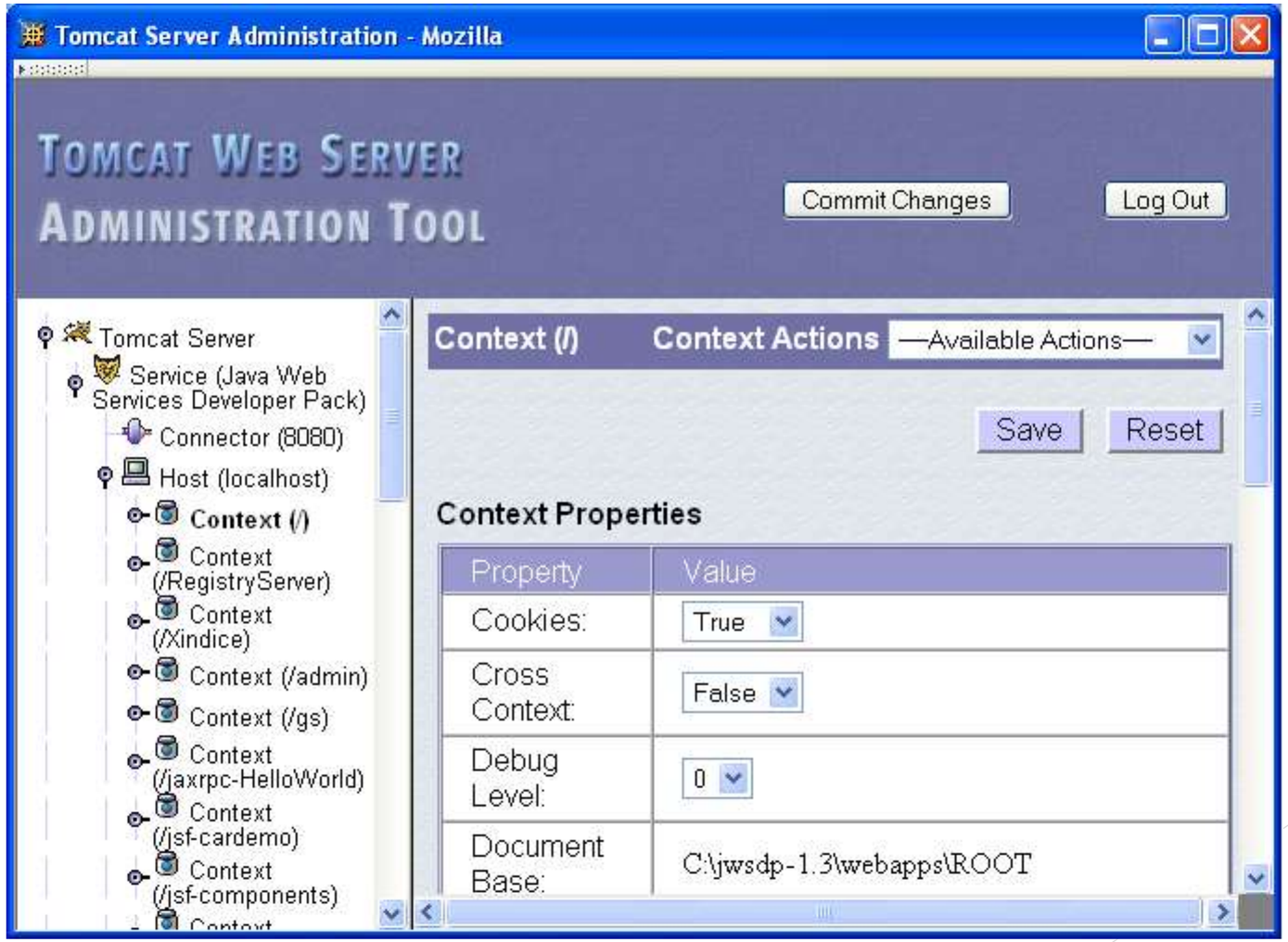
Tomcat Web Server



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 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 JWSDP 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`

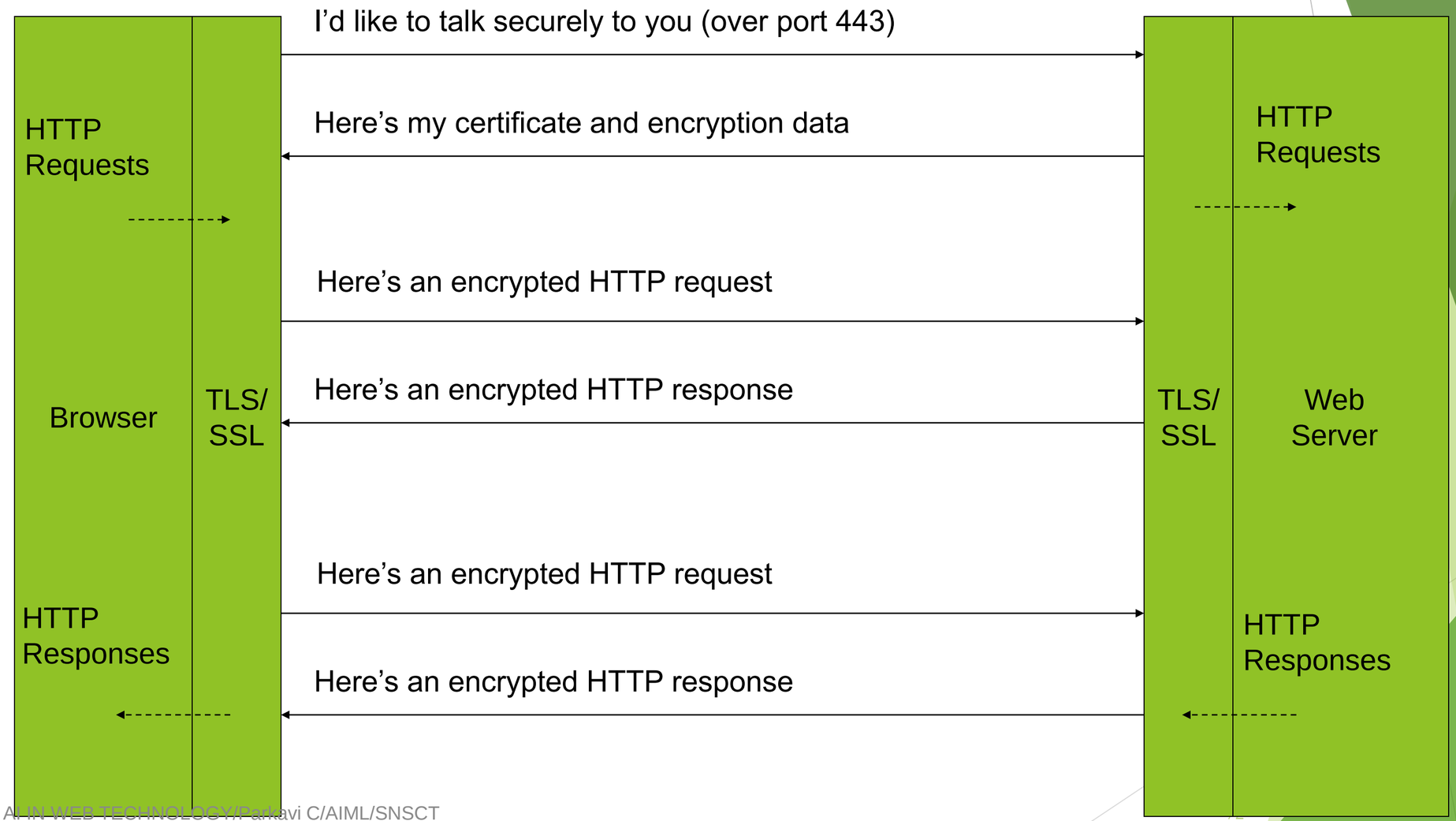
Tomcat Web Server

- 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)

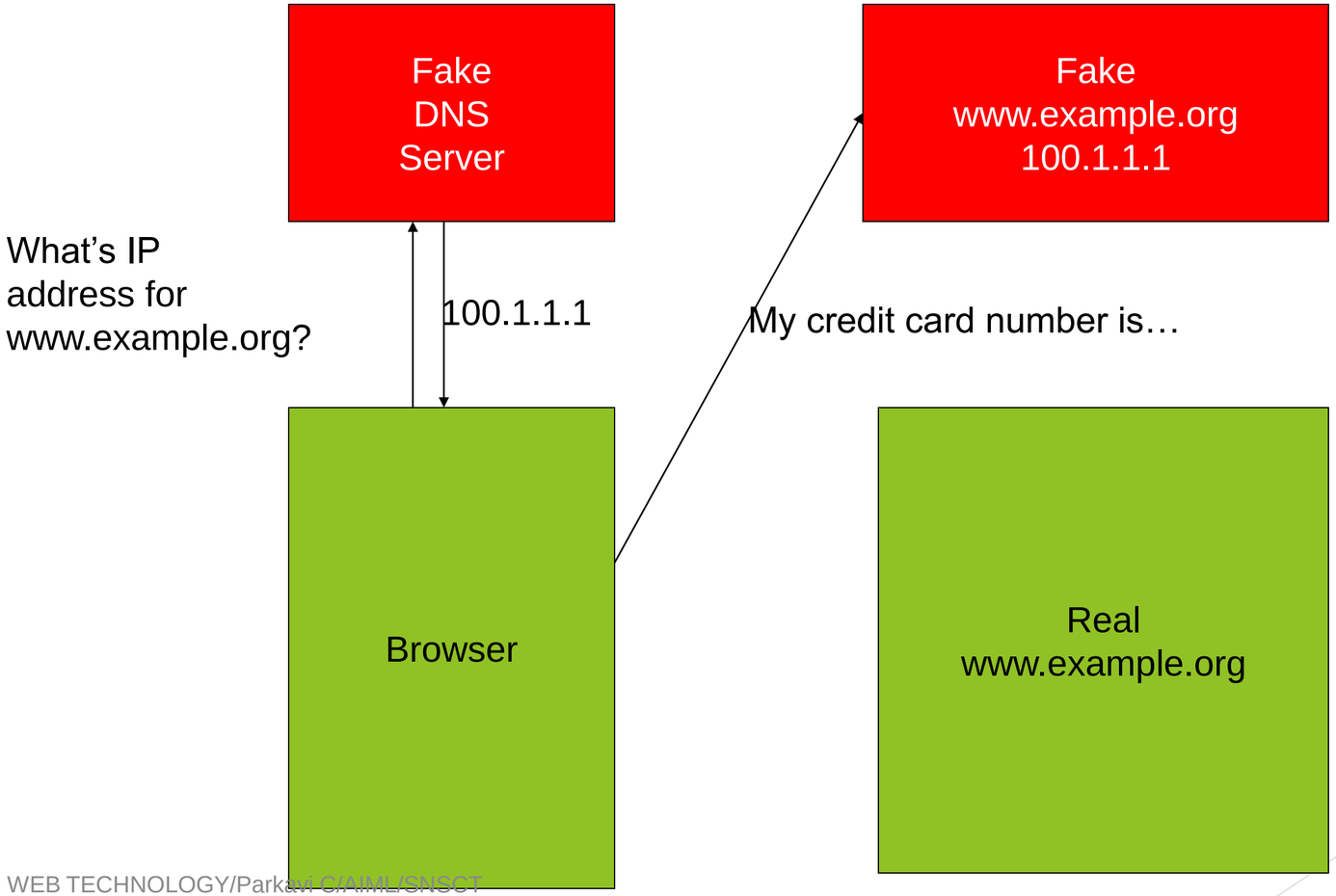
Secure Servers

- 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