HyperText Transfer Protocol (HTTP)

AUBURN UNIVERSITY

CPSC 4970 Applied Cyber Security

What is HTTP?

AUBURN

- **A Protocol**
  - Standard procedure for defining and regulating communication.
  - FTP, TCP, UDP, HTTP, SMTP

- World Wide Web **communication standard**
  - HTTP is the lingua franca of the WWW

- Application-layer protocol for transferring data
  - Between server and client
  - Data can be plaintext, hypertext, image, video, etc.

HTTP Relevance To Security

AUBURN

- Pervasiveness of HTTP today in the interconnectedness of mobile devices, physical devices (IoT), clients, servers make it the most common attack surface.
- Many vulnerabilities exist in the communication and execution of HTTP requests on browser and server side
- Having a good understanding of HTTP protocol basics enables effective use of security scanning tools and how vulnerabilities work.
- Vulnerability scanning and mitigations are related how HTTP communications are structured.
  - HTTP Headers
  - HTTP Encryption
- HTTP is a relatively straightforward protocol to understand.

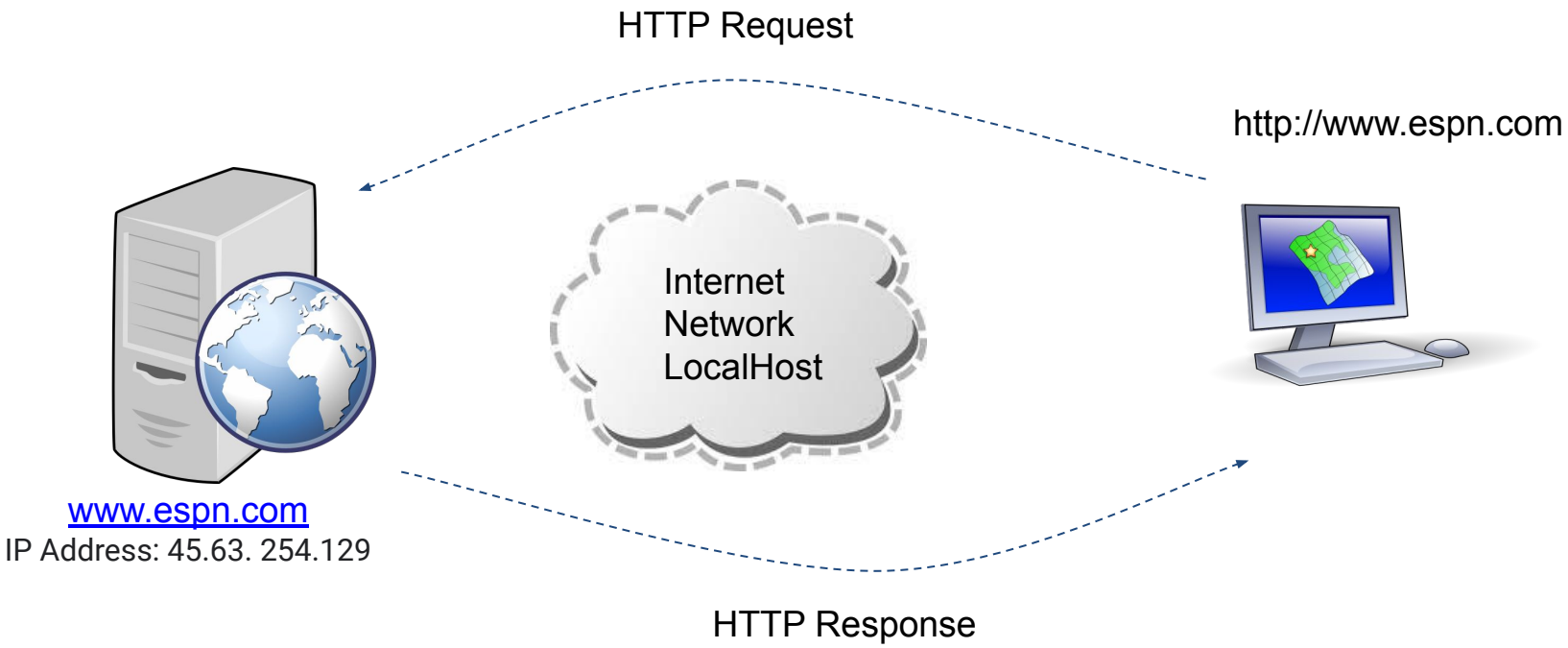
History of the World Wide Web

AUBURN

- 1989 – "Information Management: A Proposal"
  - Tim Berners-Lee, CERN March 1989
  - "Mesh" —> "World Wide Web"
- 1991 - First World Wide Web site
- 1993 - Mosaic - first widely adopted browser
  - Lead by Marc Andreessen and James Clark
- 1994 – Netscope founded by Andressen and Clark
- 1994 – World Wide Web Consortium (W3C) founded by Lee
  - Develops standards for the Web.

How HTTP Works

- HTTP communicates between two applications
  - An client program - typically a browser, sending connection requests
  - A server program - typically a web server, receiving and responding to requests

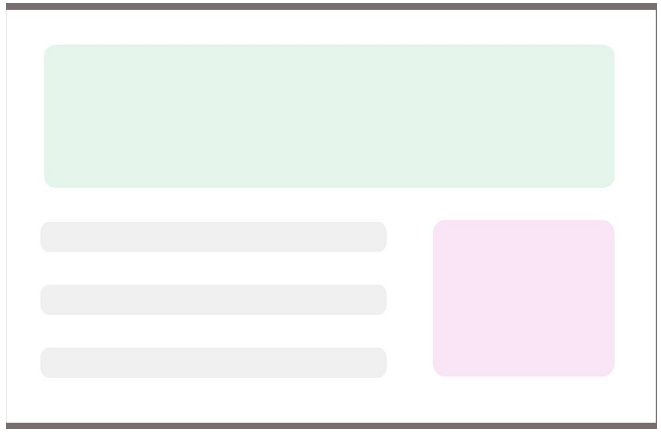


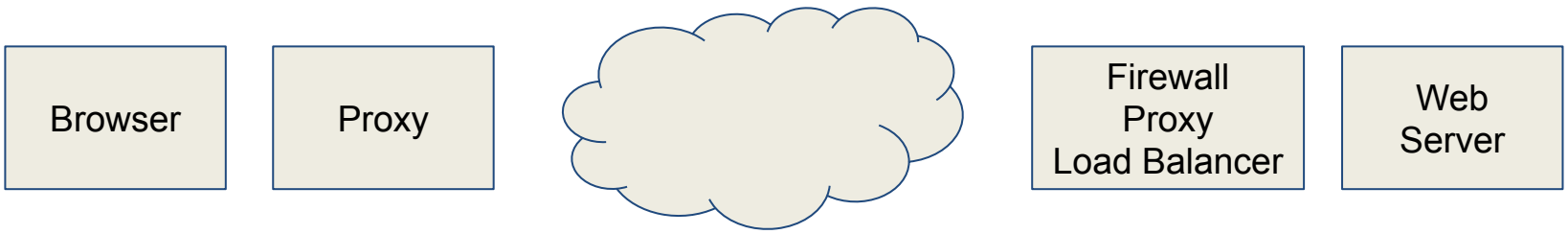
How the Magic work.

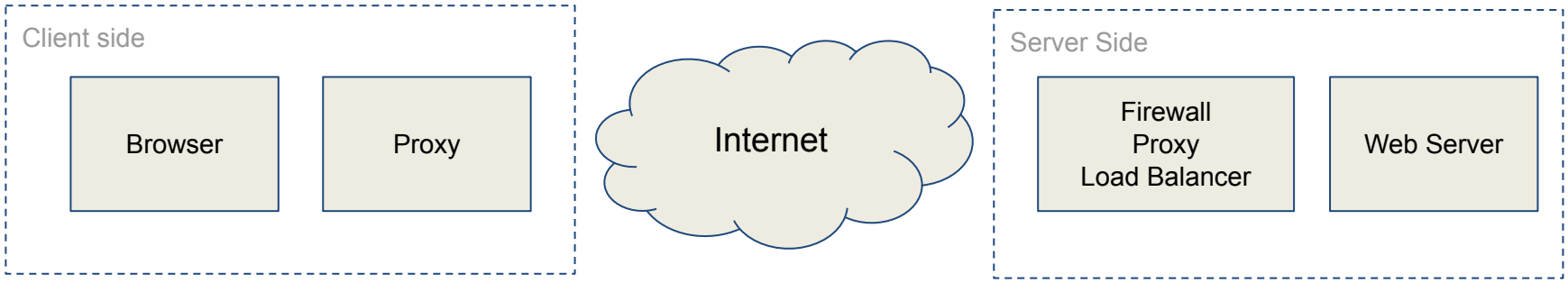
AUBURN

We click on a Hyperlink

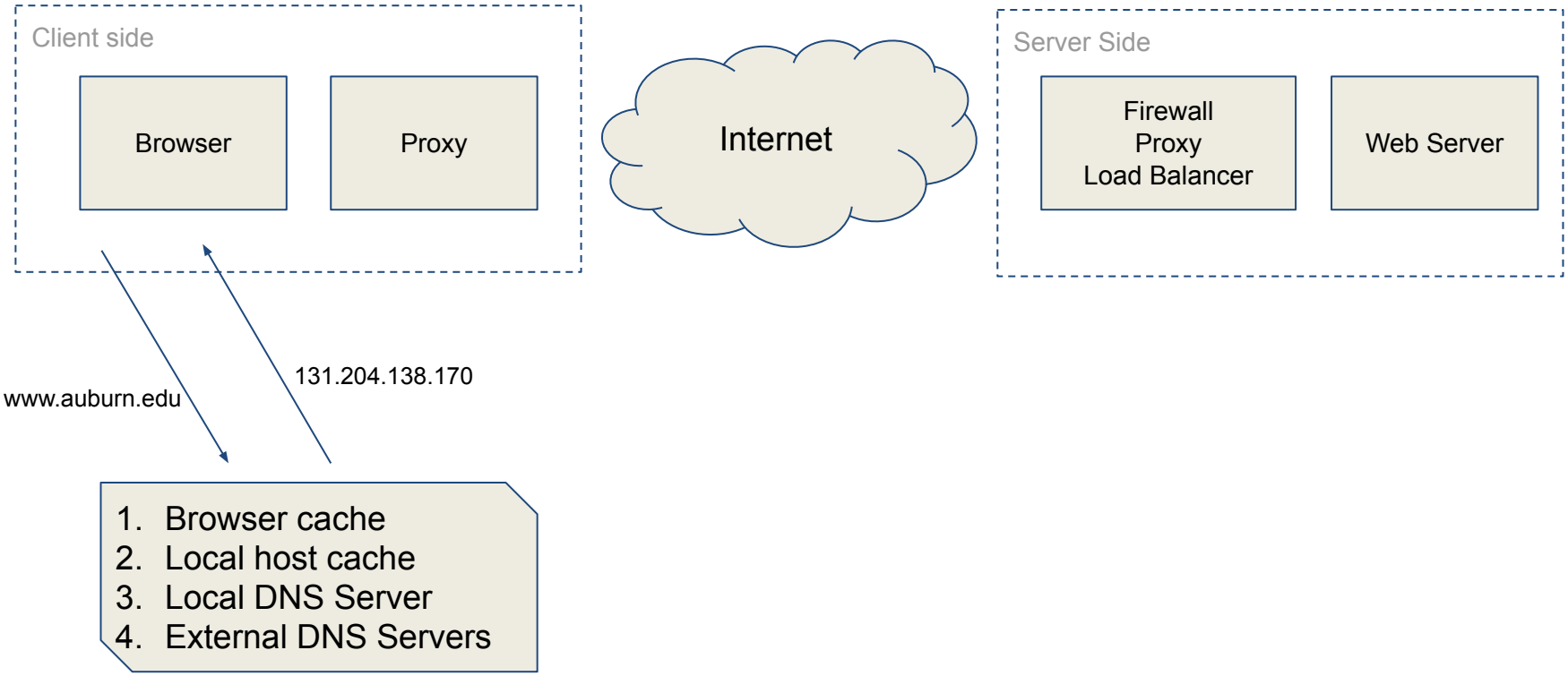
And a browser page is rendered



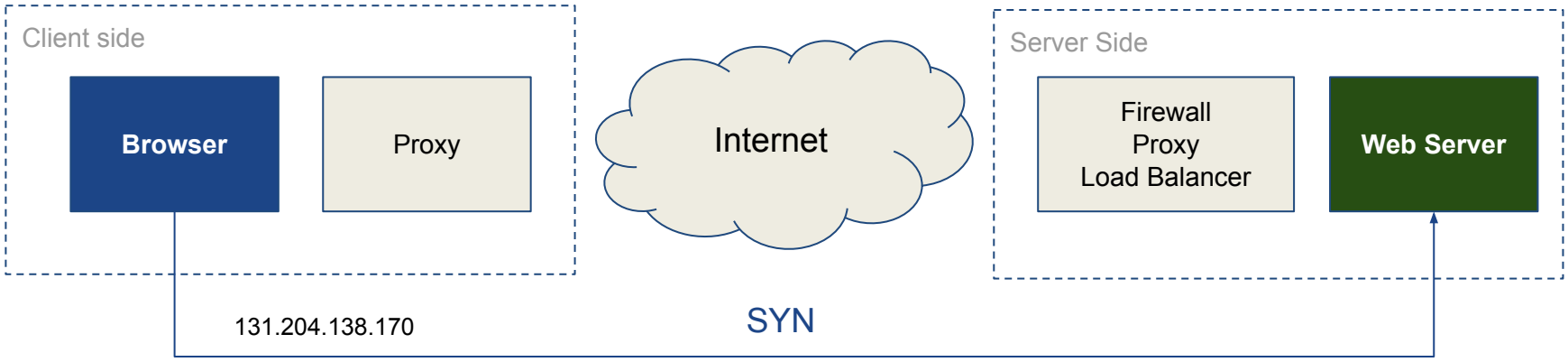

Browser
Proxy
Firewall
Proxy
Load Balancer
Web
Server


Client side
Browser
Proxy
Internet
Server Side
Firewall
Proxy
Load Balancer
Web Server

DNS Lookup of “www.auburn.edu”

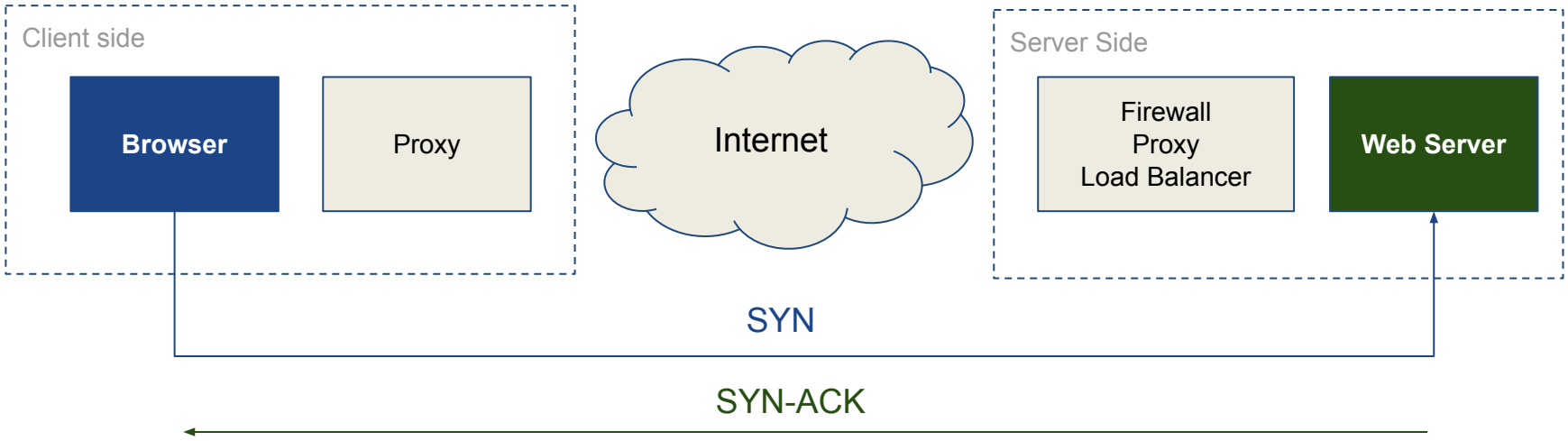


TCP Socket Connection to IP Address



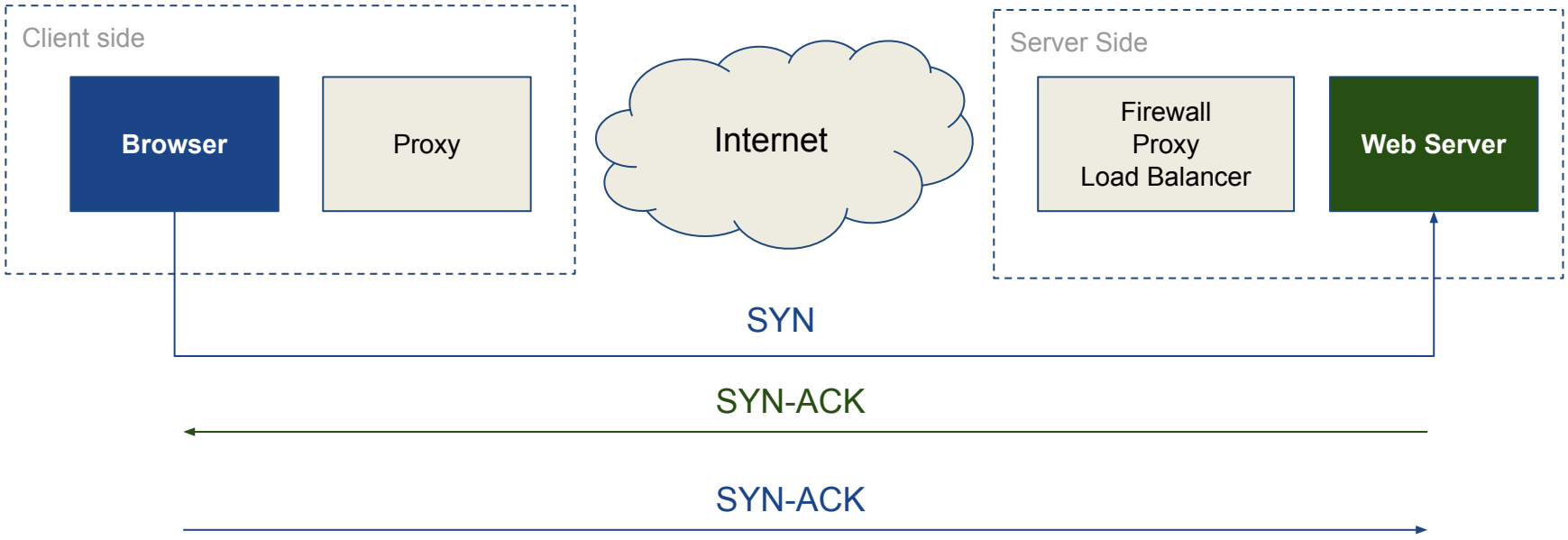
TCP Connection: “Hello There”

TCP Socket Connection to IP Address



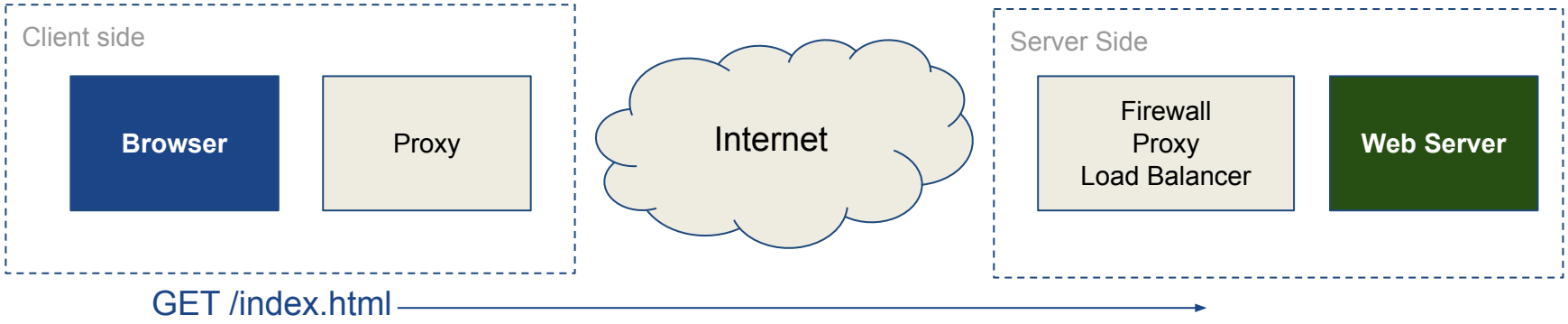
TCP Connection: “Hello Back”

TCP Socket Connection to IP Address



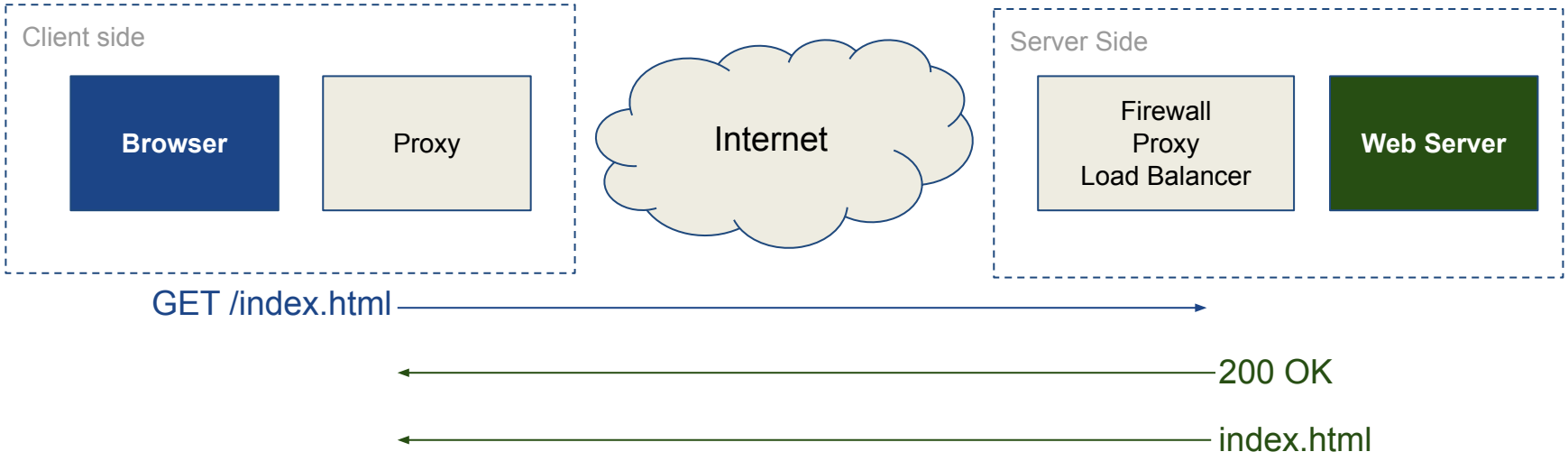
TCP Connection: “Let’s Chat” - Connection Established!

HTTP Protocol Conversation



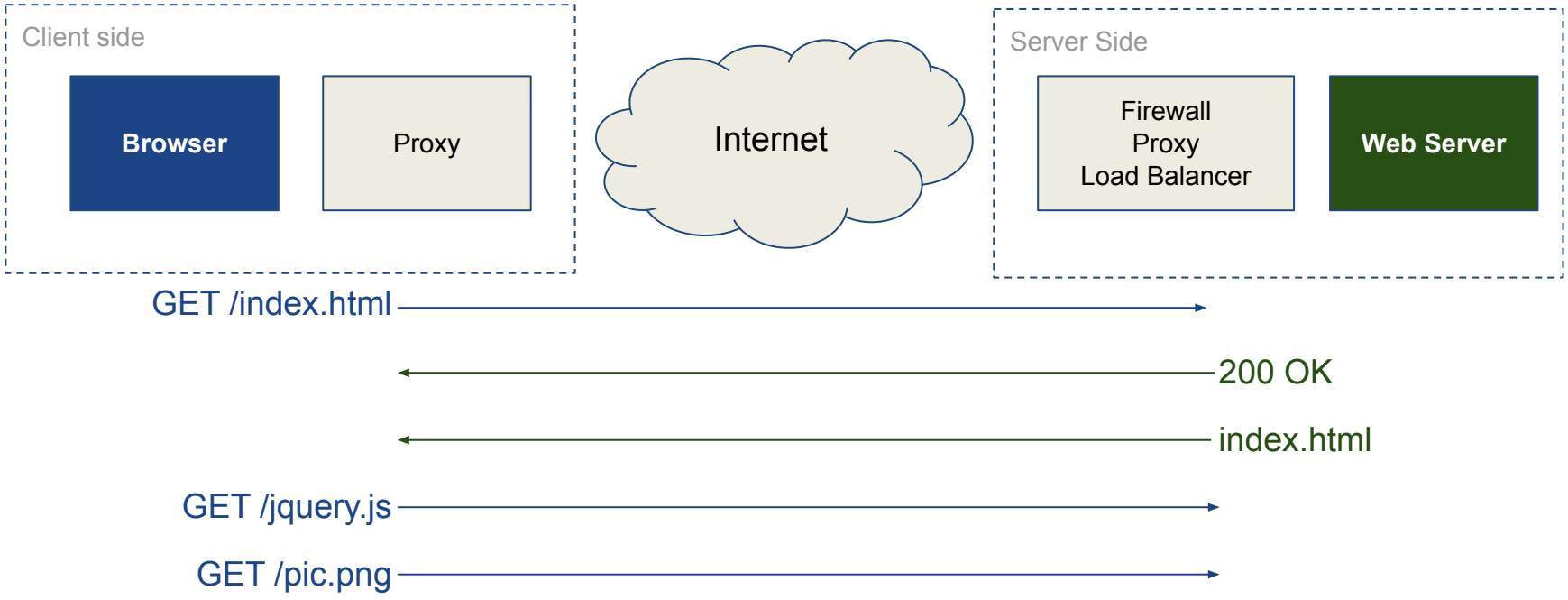
HTTP: Have this file?

HTTP Protocol Conversation



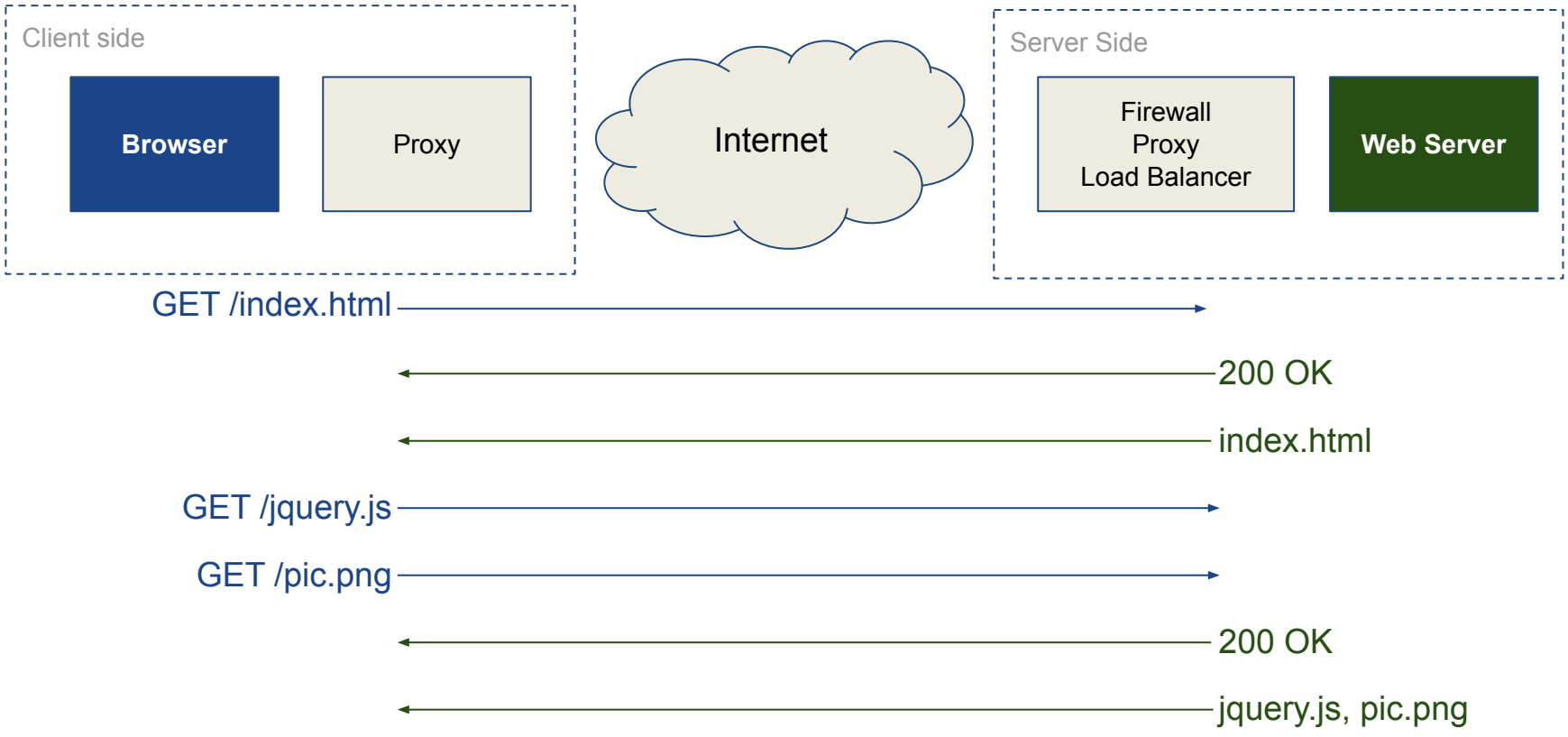
HTTP: Yes, here you go…

HTTP Protocol Conversation



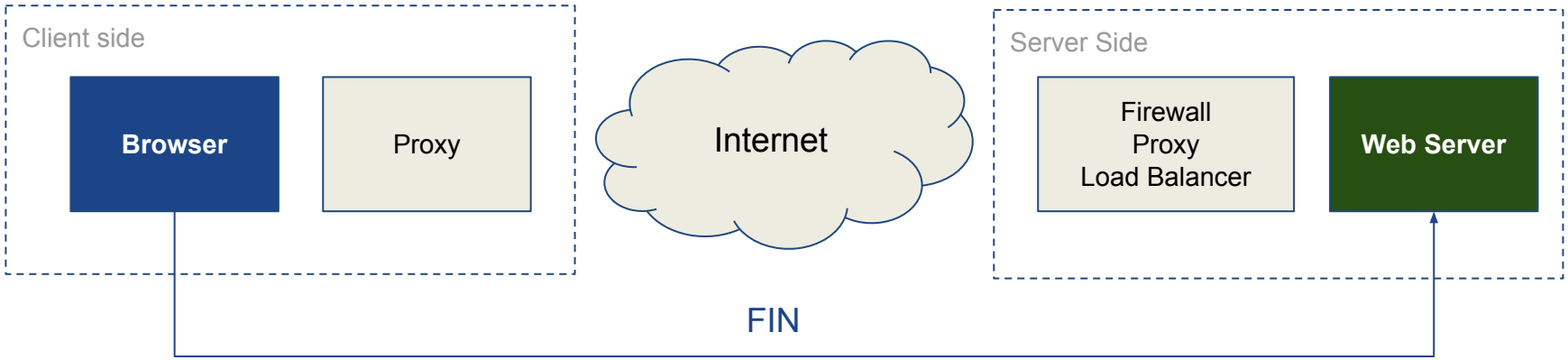
HTTP: Now I need these…

HTTP Protocol Conversation



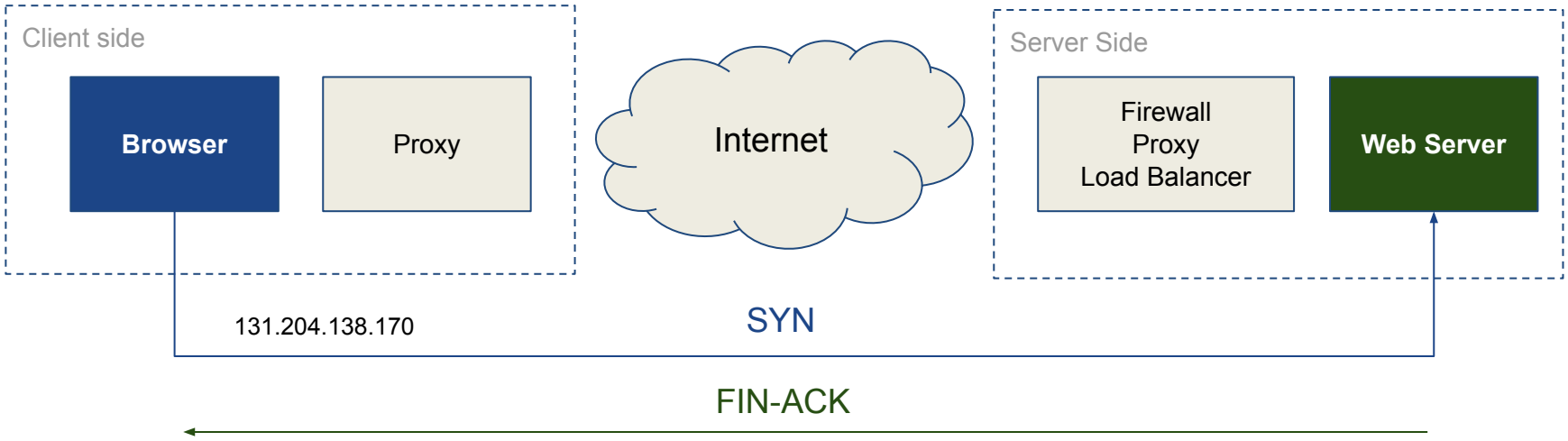
HTTP: No problem, here you go…

Close TCP Connectcion



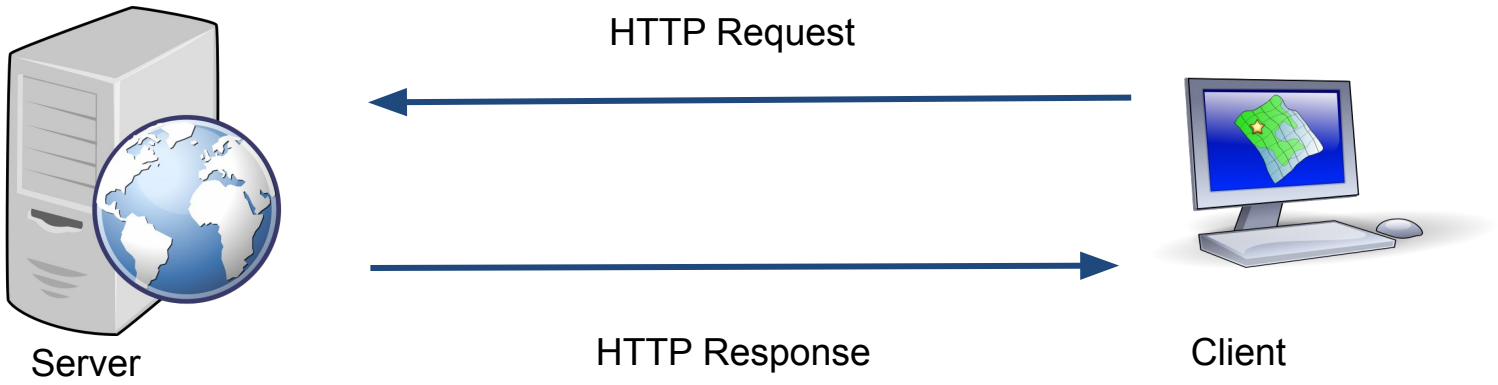
TCP Connection: Goodbye

TCP Socket Connection to IP Address



TCP Connection: See ya!

Simply Put

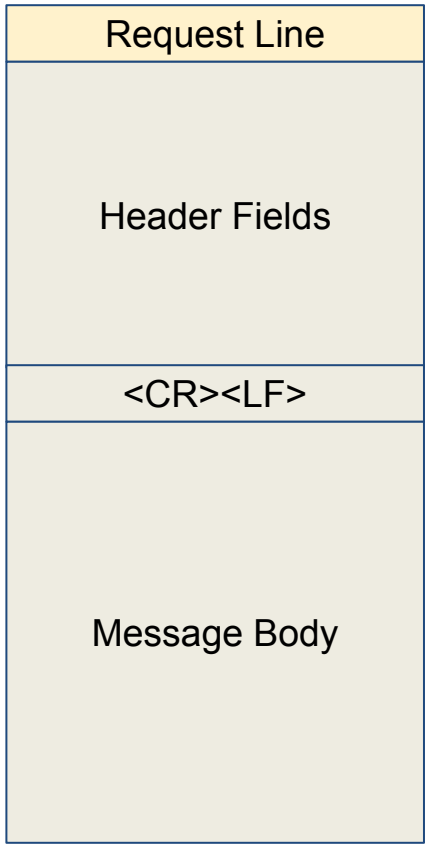


HTTP Request Structure

AUBURN

Request Line	GET /index.html HTTP/1.1
Header Fields	Host: localhost Accept: text/html X-Content-Type-Options: nosniff JSESSIONID: jd74jfhnmdnb37hdnndh
<CR><LF>	<CR><LF>
Message Body	<!DOCTYPE html> <html> <body> My Spring Boot Web App </body> </html>

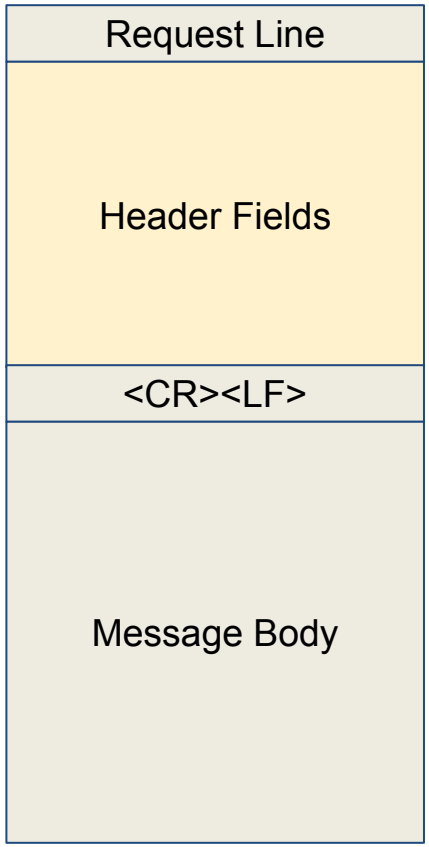
HTTP Request - Status Line



Format: <Method> <Request-URI> SP <HTTP-Version> CRLF

- Method
  - GET - retrieve information
  - POST - send information
  - PUT - create or update a resource
  - DELETE - delete a resource
- Request URI
  - Path to resource (file)
- HTTP-Version
  - Protocol Version to Use

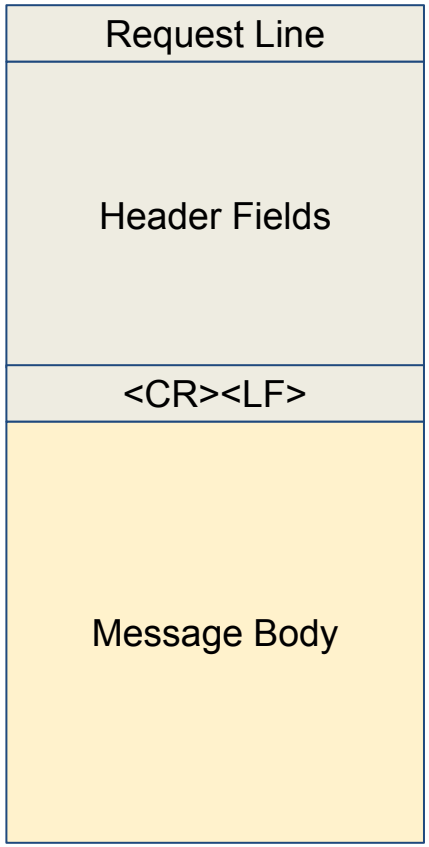
HTTP Request - Header Fields



Format: <Tag>: <value> CRLF

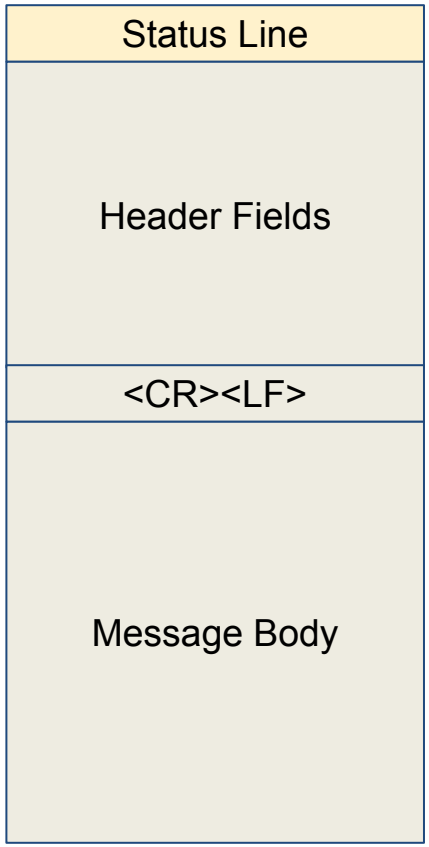
- Header Fields
  - Client and the server pass additional information with an HTTP request or response.
  - Case-insensitive name followed by a colon (:) then by its value
    - For example: “Accept: text/html”

HTTP Request - Message Body



- Contains contents to be sent to server
- POST/UPDATE methods will use to contain data to send to the server.

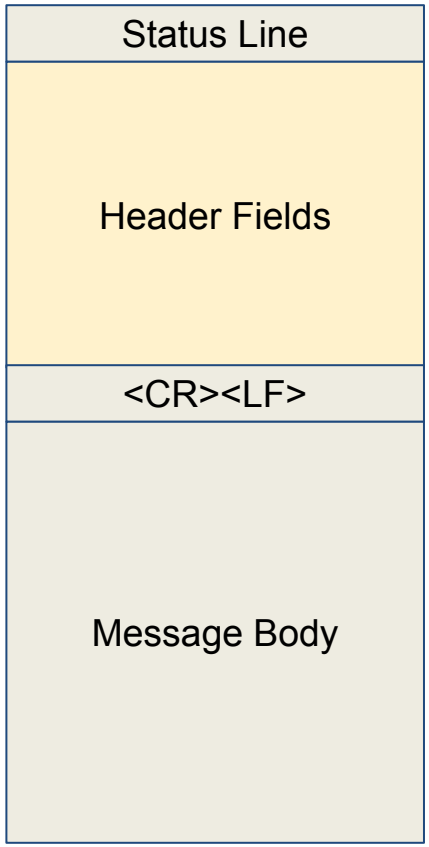
HTTP Response - Status line



Format: <HTTP-Version> <Status-Code> <Reason-Phrase> CRLF
Example: HTTP/1.1 200 OK
HTTP/1.1 404 Not Found
HTTP/1.1 500 Internal Server Error

- HTTP-Version
  - Protocol version - “HTTP/1.1”
- Status Code
  - Indication of server response type
  - First digit defines response class
    - 1xx - Informational -
    - 2xx - Success - It worked!!
    - 3xx - Redirection - browser redirect
    - 4xx - Client Error - invalid request
    - 5xx - Server Error - internal server problem
- Reason-Phrase
  - Human readable message

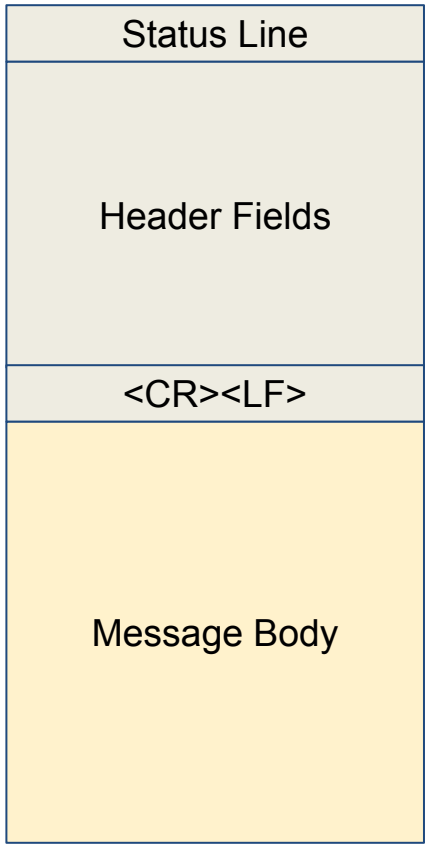
HTTP Response - Head Fields



Format: <Tag>: <value> CRLF

- Header Fields
  - Client and the server pass additional information with an HTTP request or response.
  - Case-insensitive name followed by a colon (:) then by its value
    - For example: “Accept: text/html”

HTTP Response - Message Body

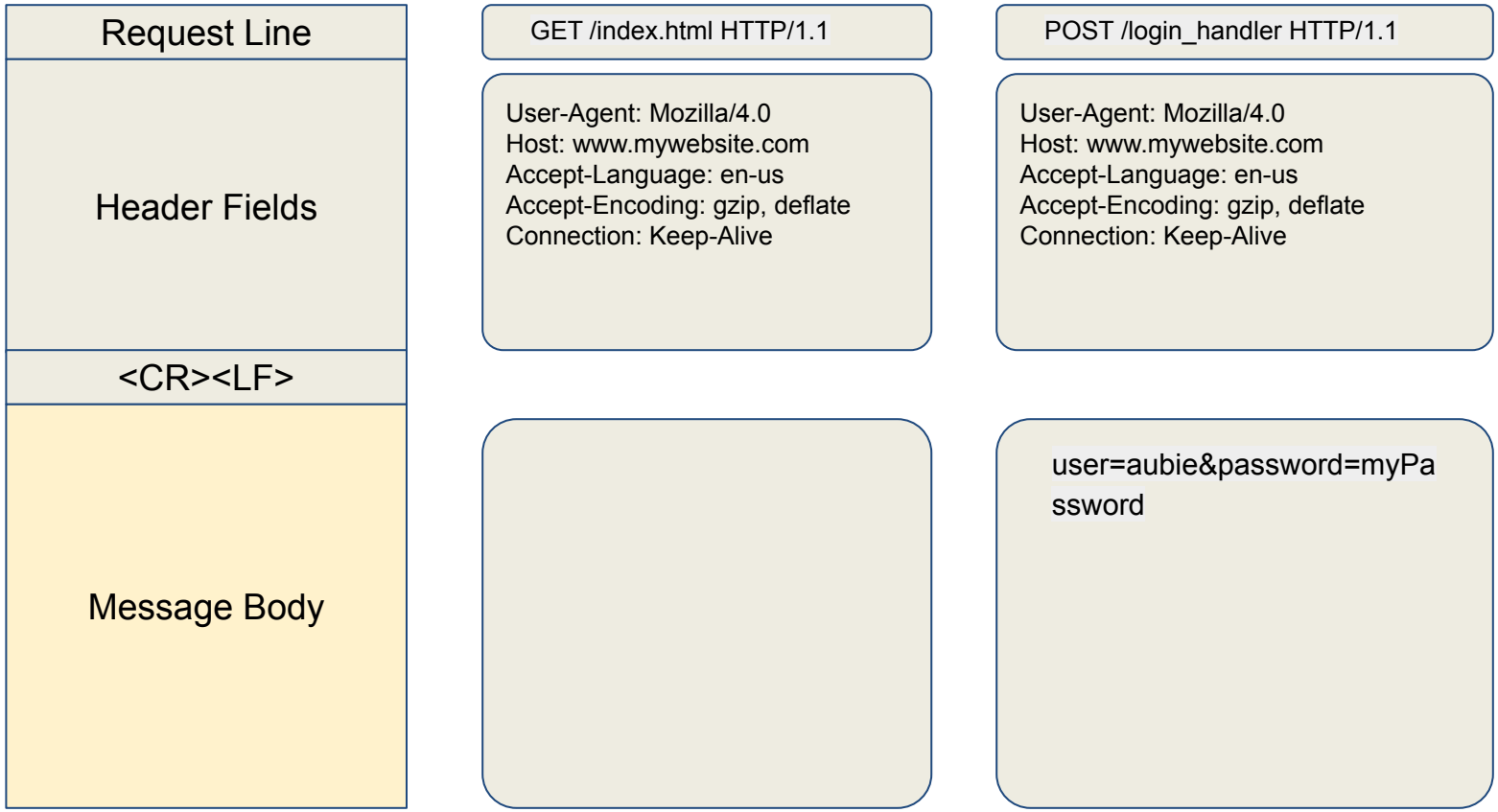


- Contains contents of the request resource
  - html files - returns exact contents

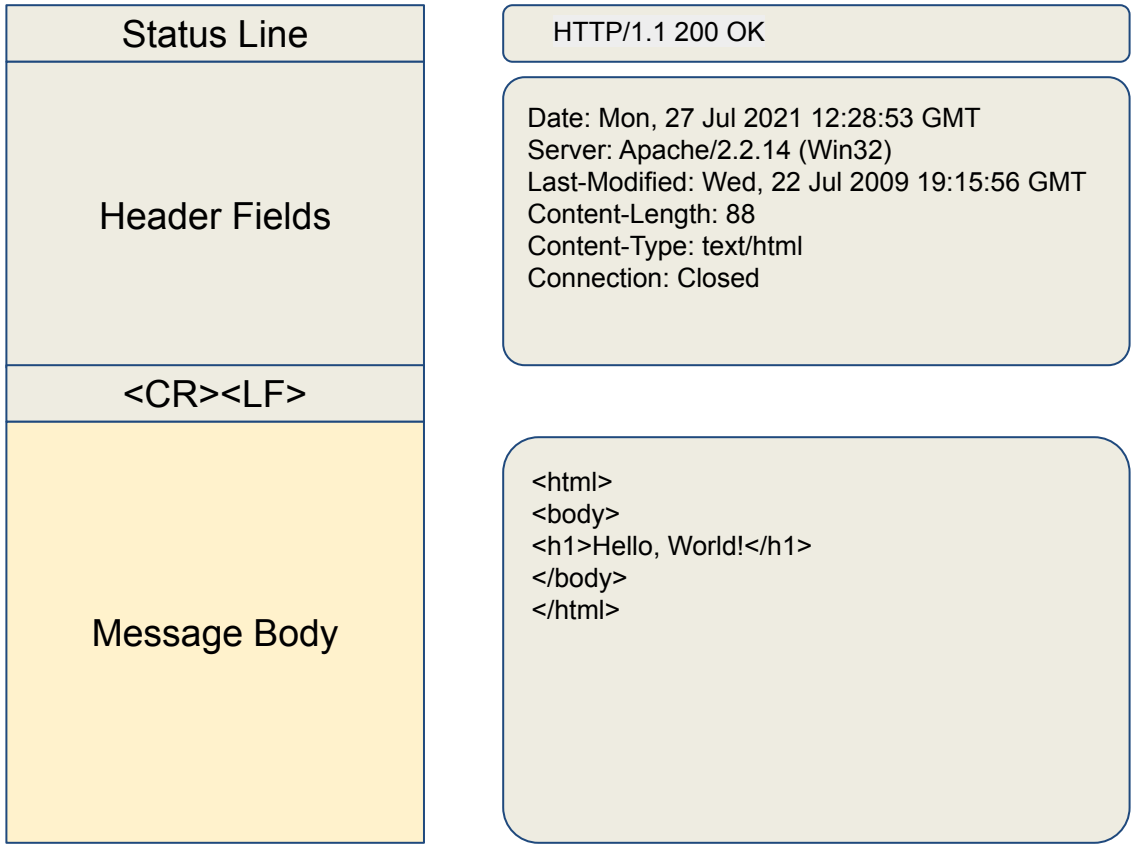
index.html

```
<html>
<body>
<h1>Hello, World!</h1>
</body>
</html>
```

HTTP Request - Example



HTTP Response - Example



HTTP is stateless

AUBURN

- Essentially means protocol does not save an information between requests
  - Up to the server and client to manage
- Enter “Cookies”
  - Data stored by client browser
  - Passed as values in the Header section on request and response
  - Maintains a “session” by storing information
  - Several key characteristics
    - name=value
    - expires=value
    - domain=value
    - path=value

AUBURN UNIVERSITY