NSCOM01

TCP-based Network Application Protocols

3rd Term– AY2022 – 2023

Instructor: Dr. Marnel Peradilla

SPIRAL REVIEW: TRANSPORT SERVICES

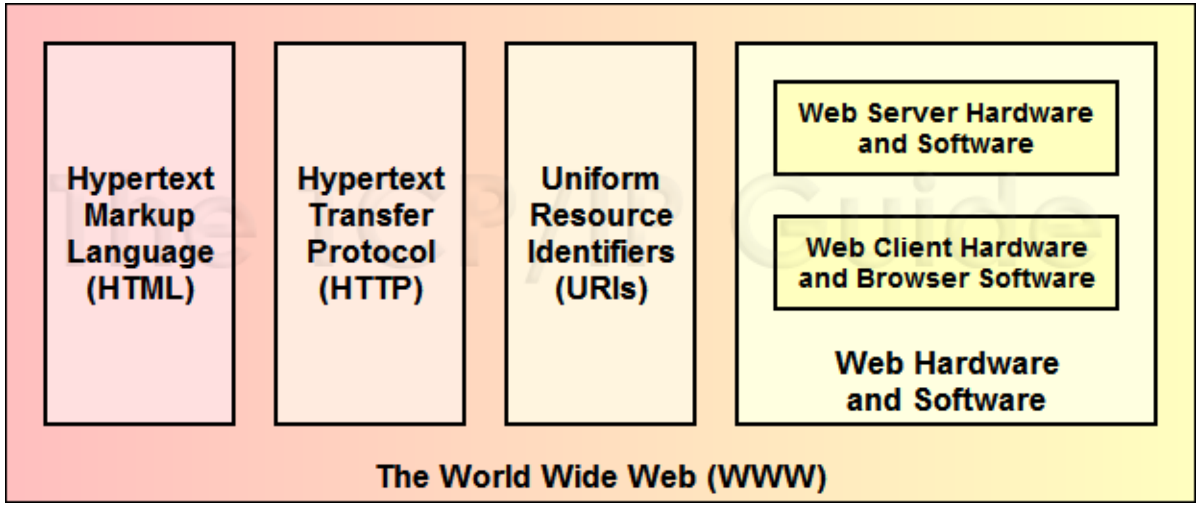
- ❑ **The Transport Control Protocol (TCP) is a connection-oriented transport protocol used in TCP/IP networks**
- ❑ **Provides reliable communication between pairs of processes (TCP users) across a variety of reliable and unreliable networks**
  - Features:

1. Stream-oriented – Data is sent in segments but handled as streams
2. Connection Oriented – Includes mechanisms to establish, track state and terminate a connection between 2 hosts
3. Guaranteed delivery – packets are acknowledged by receiving hosts
4. Flow control - Data transmission adapts to network conditions and host capability
5. Ordered delivery – Segments may arrive out of-order but are reassembled in the correct sequence

HTTP

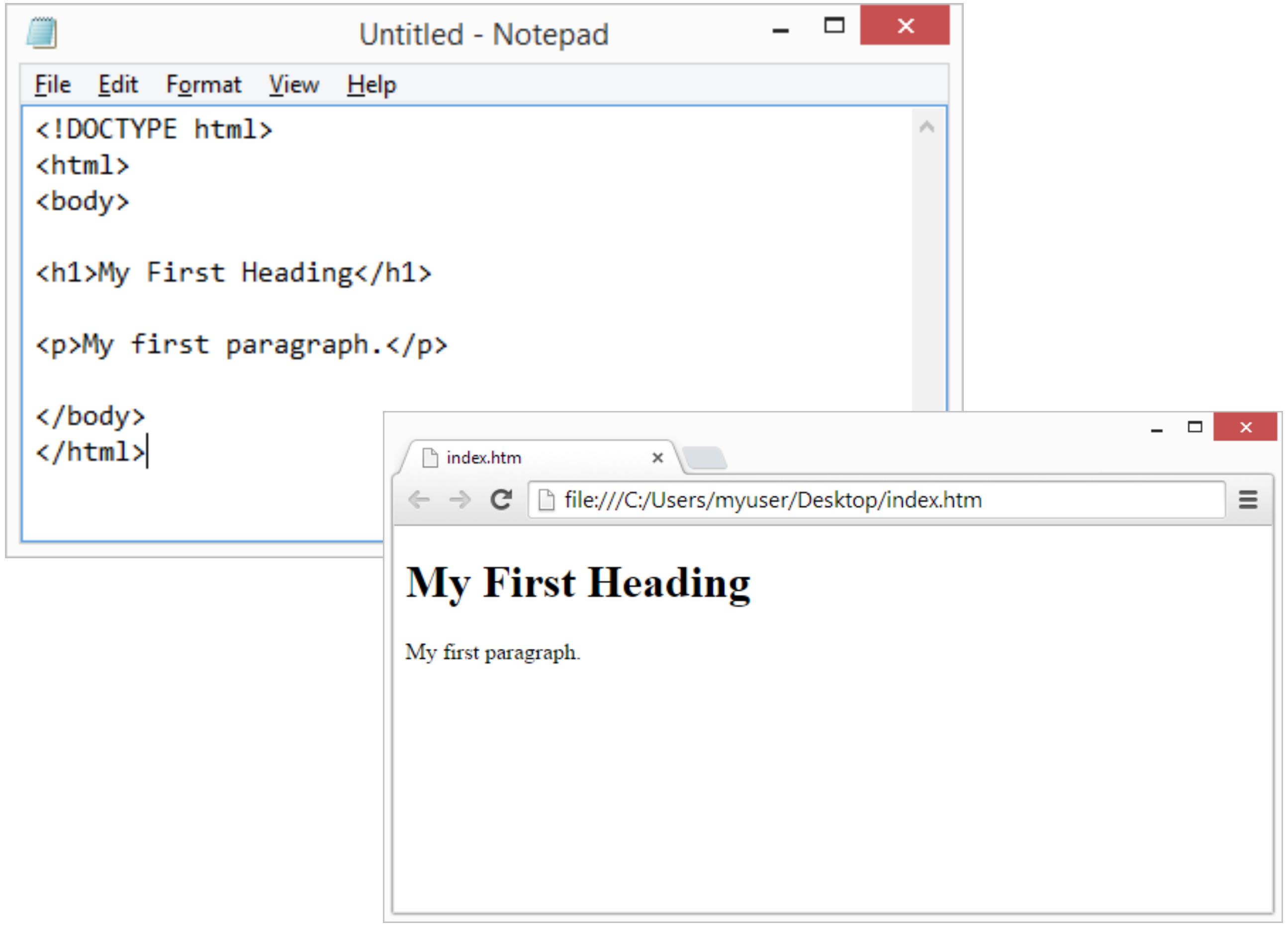
WWW

- ❑ **The concept of the World Wide Web was proposed by Tim Berners-Lee as a “web” of electronically-linked documents**
- ❑ **The World Wide Web’s infrastructure is made up of 3 main components**
  - Hypertext is a document format that supports the linking of related documents together, be it text or multimedia
  - Uniform Resource Indicators (URI) are a method of identifying and referencing resources on the WWW
  - Hypertext Transfer Protocol (HTTP) is the application protocol that makes it possible to transfer hypertext documents and other files between client and server


Hypertext Markup Language (HTML)
Hypertext Transfer Protocol (HTTP)
Uniform Resource Identifiers (URIs)
Web Server Hardware and Software
Web Client Hardware and Browser Software
Web Hardware and Software
The World Wide Web (WWW)

HYPERTEXT AND HTML

- **Hypertext Markup Language (HTML) the standard language used by the WWW to define hypertext documents**
  - Contains instructions to specify how a document is linked to another for implementation of hypertext functions
  - Also includes extra information interspersed with document content to instruct a client on how to correctly format and display the document to the user



```
<!DOCTYPE html>
<html>
<body>

<h1>My First Heading</h1>

<p>My first paragraph.</p>

</body>
</html>
```

file:///C:/Users/myuser/Desktop/index.htm

My First Heading

My first paragraph.

UNIFORM RESOURCE INDICATORS

- **Uniform Resource Identifiers (URIs) were developed as a standard format to refer to World Wide Web resources in order to easily and consistently identify them**
- **2 types:**
  - Uniform Resource Locators (URL) – used almost exclusively by the WWW and identify resources by specifying their location on a network
  - Uniform Resource Names (URN) – considered experimental and identify resources using location-independent names

URL

❑ URLs commonly follow the format

```
<scheme>://host>:<port>/<url-path>?<query>#<bookmark>
```

<scheme>	Protocol to be used for resource access or retrieval
<host>	Fully qualified domain name or IP address of server hosting the resource
<port>	Hosting service port number. If unspecified, then uses the standard port associated with the protocol specified by the scheme
<url-path>	Directory path pointing to the specific resource to be retrieved
<query>	Optional query or other information to be passed to the Web server
<bookmark>	Optional Identifier for a particular location within an HTML document

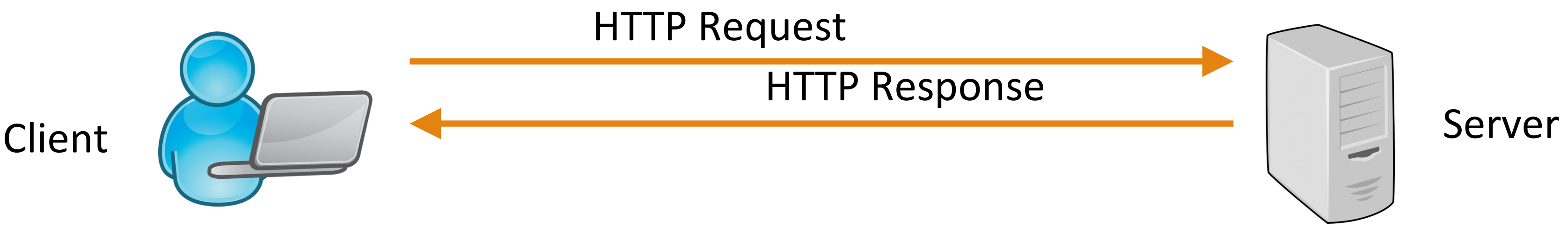
```
http://www.example.com:123/forum/questions/?tag=fish&order=newest#top
```

HYPERTEXT TRANSFER PROTOCOL

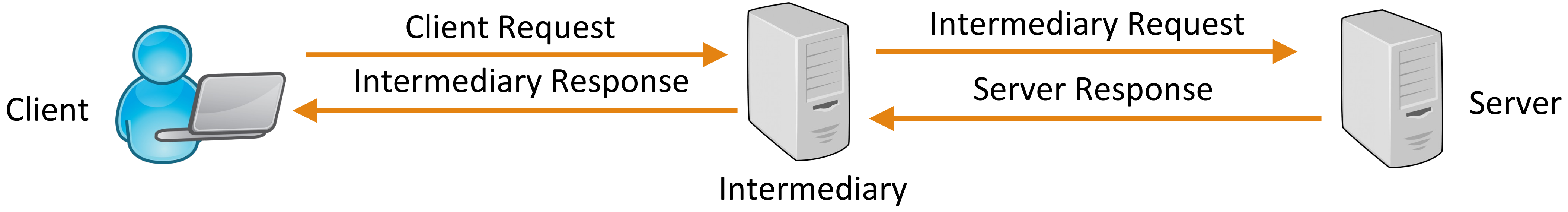
- **HTTP is a request-response protocol that is used primarily by WWW for resource transfer between web clients and servers**
- **It is stateless (does not maintain a state of a session) and asynchronous (an html document is loaded asynchronous by the browser, as soon as parts of it are available)**
- **Latest widely available version is HTTP 2.0 but a large number of sites still use HTTP 1.1**
- **By default uses TCP port 80, but port 8080 and 8008 are also commonly associated**
- **On its own, HTTP is a plaintext protocol. Websites that need data confidentiality and authentication use Secure HTTP (HTTPS) instead.**
  - Uses TCP port 443
  - Uses the same operations as HTTP, but integrates Transport Layer Security, an encryption protocol

HTTP OPERATIONAL MODEL

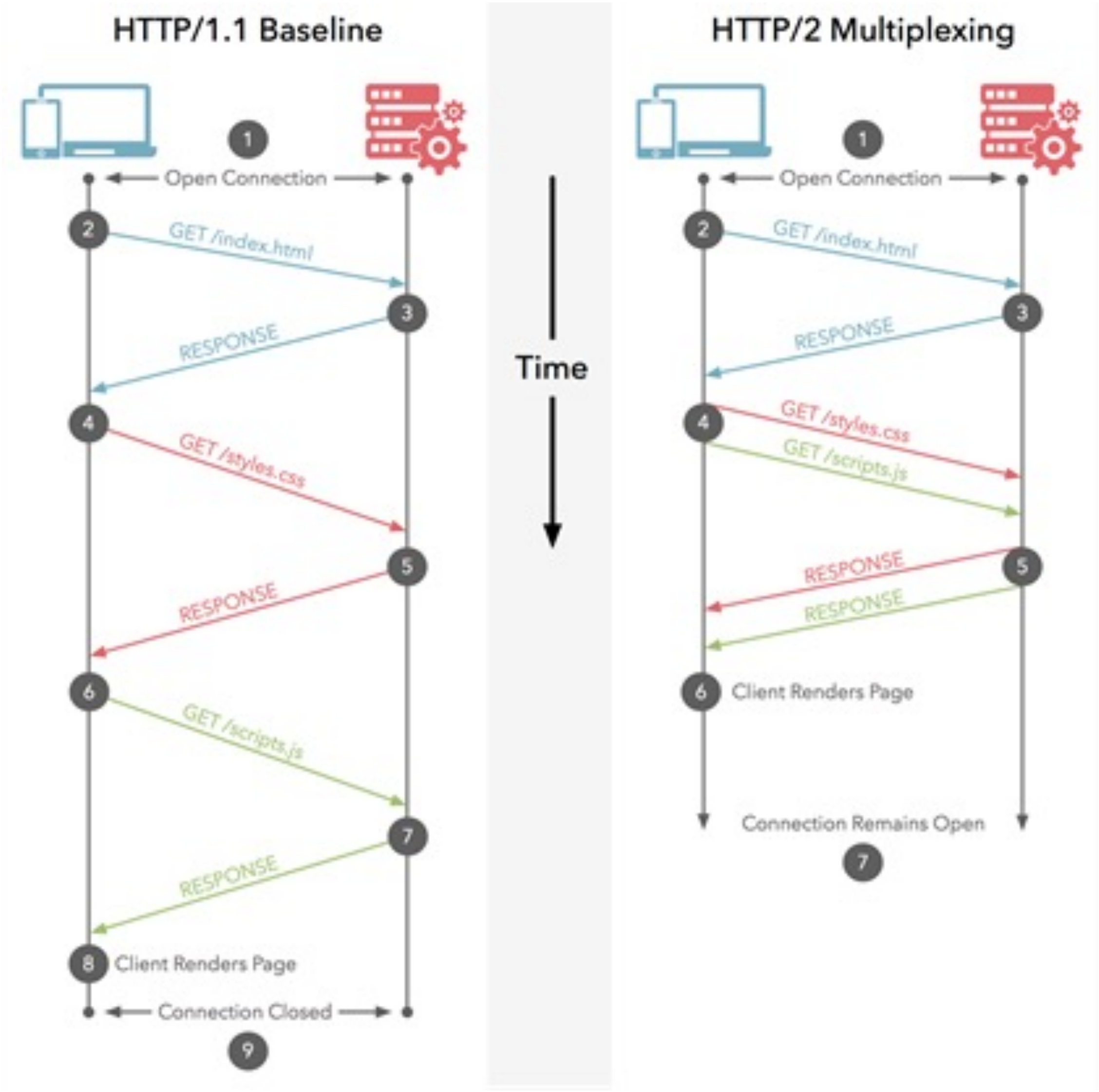
- ❑ **In its most basic model, HTTP involves a client which issues HTTP requests usually through a browser, and a web server that interprets the request, performs the relevant processing and returns an HTTP response.**



- ❑ **Intermediaries can act as servers from a client's perspective to do caching, translation, or filtering services (e.g. proxies or web caches)**



HTTP CONNECTION MANAGEMENT



❑ Older versions of HTTP used transitory connections

- A connection between client and server is created when a request is sent, and is immediately closed once the reply is received
- Slow and inefficient once webpages that contained a large amount of media became popular because each component required a separate connection to be used

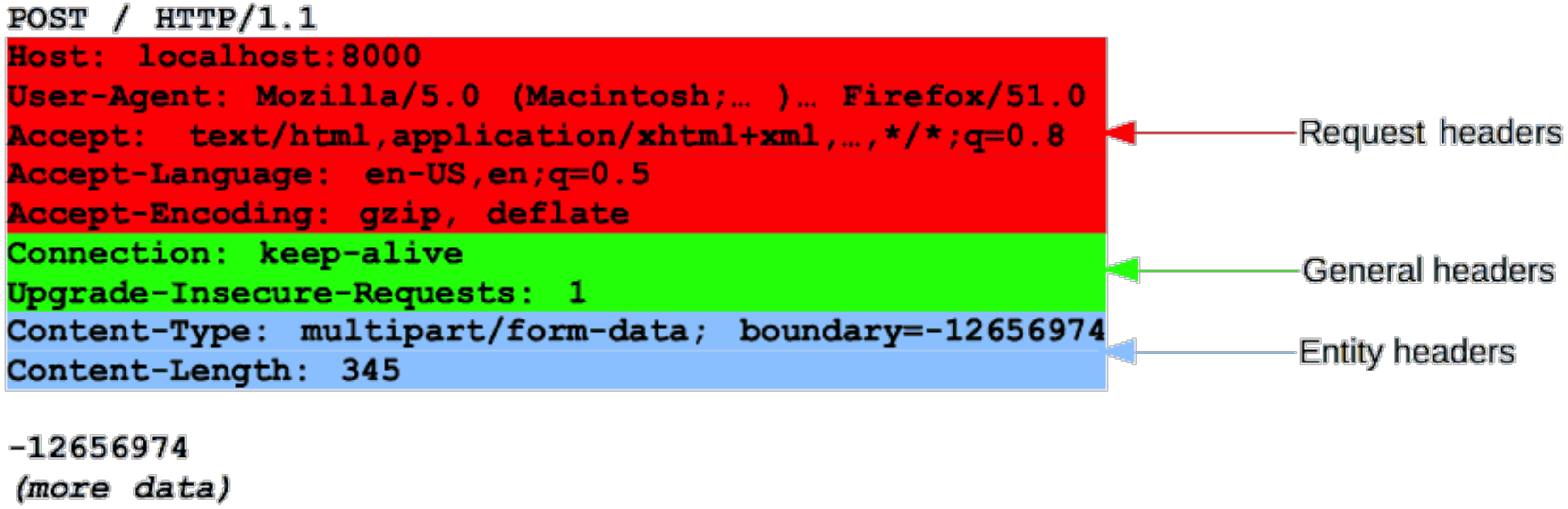
❑ Current versions support persistent connections

- Connection is kept alive after the response so that it may be reused for succeeding requests without the added delay of connection reestablishment
- To further speed up, multiplexing is supported in HTTP 2.0 → several requests may be sent in succession without waiting for each response to be completed first

HTTP REQUEST MESSAGE

- ❑ **HTTP Request messages are sent by the client to the server to specify an action that it wants the server to take**
- ❑ **Request message format:**

```
<request-line>
<request-headers>
<general-headers>
<entity-headers>
<empty-line>
[<message-body>]
[<message-trailers>]
```



```
POST / HTTP/1.1
Host: localhost:8000
User-Agent: Mozilla/5.0 (Macintosh;… )… Firefox/51.0
Accept: text/html,application/xhtml+xml,…,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Connection: keep-alive
Upgrade-Insecure-Requests: 1
Content-Type: multipart/form-data; boundary=-12656974
Content-Length: 345

-12656974
(more data)
```

developer.mozilla.org

HTTP REQUEST MESSAGE

- ❑ **The request line follows the syntax <METHOD> <request-uri> <HTTP-VERSION>**
- ❑ **<METHOD> indicates type of action that the client wants the server to take. Most common are:**
  - GET - requests to retrieve the specified resource
  - POST - requests that the server accept the entity enclosed in the request
  - Other commands (HEAD, PUT, DELETE, TRACE, OPTION, CONNECT) are available but not supported on browsers
- ❑ **<request-uri> indicates the resource to which the method is to be applied**
  - Only the URL path is placed here. It does not include the hostname or IP address of the server
- ❑ **<HTTP-VERSION> indicates to the server what HTTP version is used for the request**
- ❑ **Example:**
  - To request for the page https://en.wikipedia.org/wiki/Hypertext_Transfer_Protocol using HTTP 2.0

```
GET /wiki/Hypertext_Transfer_Protocol HTTP/2.0
```

HTTP REQUEST MESSAGE

- **A Request Message includes 3 headers**
- **Request Header**
  - Convey to the server more details about the nature of the client's request, and give the client more control over how the request is to be handled
  - Some common ones:

Accept	Tells the server what media types the client is willing to accept in the response
Accept-Language	Indicate what languages the client supports or expects the server to use in its response.
Accept-Encoding	Used to control whether or not the server may send content in compressed form
Host	Specifies the Internet host as using its domain name. Commonly used to determine which website is being accessed for servers that are hosting multiple sites
User-Agent	Provides information about the client software
Referer	Tells the server the URL of the resource from which the URL of the current request was obtained

HTTP REQUEST MESSAGE

❑ **A Request Message includes 3 headers**

❑ **General Header**

- contain information about the message which can be used to control how the message will be processed
- Some common ones:

Date	Indicates date and time when the request was issued
Connection	Specifies whether the client wants to use persistent connection (keep-alive) or close it after receiving the response (close).

❑ **Entity Header**

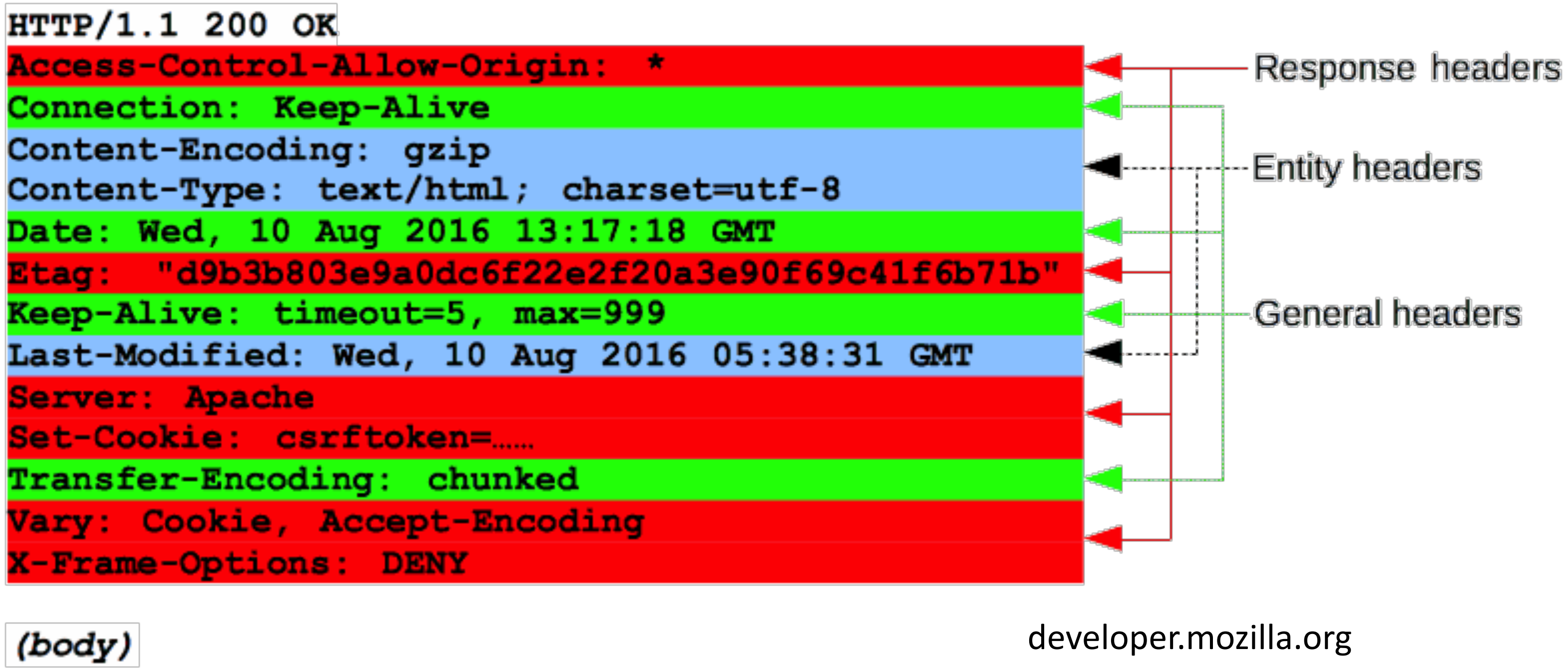
- Describes the entity contained in the body of the request if present
- Some common ones

Content-Length	Indicates the size of the body content
Content-Type	Specifies the MIME type (media type and subtype) of the content

HTTP RESPONSE MESSAGE

- ❑ **HTTP Response messages are sent by the server to the client to return the status and and any resources requested by the client**
- ❑ **Response message format:**

```
<status-line>
<general-headers>
<response-headers>
<entity-headers>
<empty-line>
[<message-body>]
[<message-trailers>]
```



```
HTTP/1.1 200 OK
Access-Control-Allow-Origin: *
Connection: Keep-Alive
Content-Encoding: gzip
Content-Type: text/html; charset=utf-8
Date: Wed, 10 Aug 2016 13:17:18 GMT
Etag: "d9b3b803e9a0dc6f22e2f20a3e90f69c41f6b71b"
Keep-Alive: timeout=5, max=999
Last-Modified: Wed, 10 Aug 2016 05:38:31 GMT
Server: Apache
Set-Cookie: csrftoken=......
Transfer-Encoding: chunked
Vary: Cookie, Accept-Encoding
X-Frame-Options: DENY

(body)
```

developer.mozilla.org

HTTP RESPONSE MESSAGE

- **The response line follows the format <HTTP-VERSION> <status-code> <reason-phrase>**
  - <status-code> and <reason-phase> provide a 3-digit code and a descriptive text for information about the result of the request

Status Code Format	Meaning	Description
1yy	Informational Message	Provides general information; does not indicate success or failure of a request.
2yy	Success	The method was received, understood and accepted by the server.
3yy	Redirection	The request did not fail outright, but additional action is needed before it can be successfully completed.
4yy	Client Error	Invalid request, bad syntax or unable to complete due to client's fault.
5yy	Server Error	Valid request but unable to complete it due to a server problem.

  - Successful request → HTTP/1.1 200 OK
  - Invalid resource issue → HTTP/1.1 404 Page Not Found

HTTP RESPONSE MESSAGE

- **A Response Message also includes 3 headers: Response, General and Entity headers. General and Entity headers serve the same purpose as those found in Requests**
- **Response Header**
  - Additional data that expands upon the summary result information in the status line
  - Some common ones:

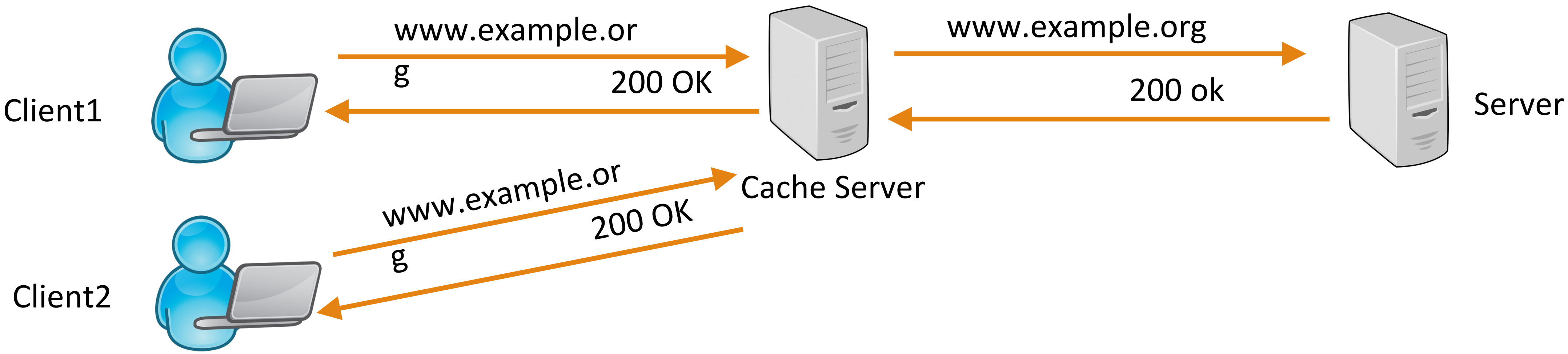
Age	Tells the client the approximate age of the resource,
Etag	Specifies the entity tag for the entity included in the response which can be used by the client in future requests to uniquely identify an entity
Server	Identifies the type and version of the server software generating the response

- **General Header**
  - Some common ones included in Responses:

Cache Control	Specifies directives that manage how caching is performed
Pragma	Often also used to control caching

HTTP CONSIDERATIONS

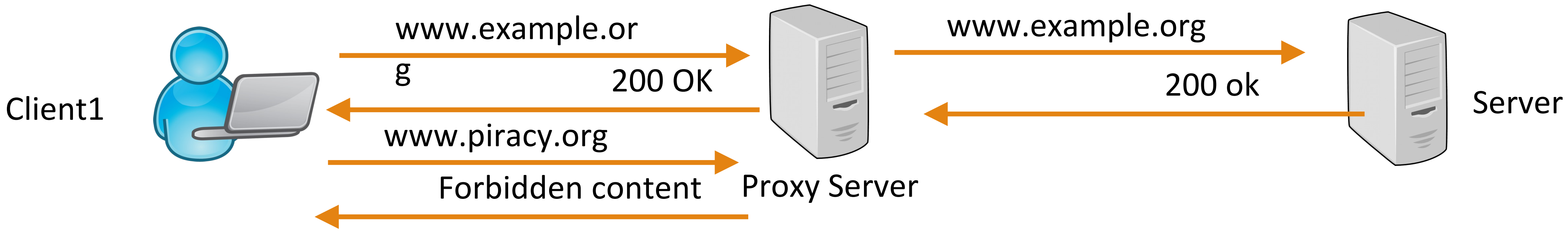
- **Caching is essential to the operation of HTTP/HTTPS to improve efficiency**
  - People tend to request the same resources repeatedly, especially those that are used in several different web pages (e.g. images, logos)
  - Dedicated cache servers or client caches are used to keep copies of resources closer to the clients to reduce the transfer time and bandwidth consumption



HTTP CONSIDERATIONS

❑ HTTP is often used to deliver harmful content to clients

- Proxy servers can offer a layer of security by examining both outgoing requests and incoming responses to filter out objectionable requests and harmful replies



REFERENCES

The TCP/IP Guide – Hypertext Transfer Protocol

http://www.tcpipguide.com/free/t_TCPIPWorldWideWebWWWTheWebandtheHypertextTransferP.htm

Mozilla Development Network - HTTP Messages :

https://developer.mozilla.org/en-US/docs/Web/HTTP/Messages

MESSAGE FROM DPO

"The information and data contained in the online learning modules, such as the content, audio/visual materials or artwork are considered the intellectual property of the author and shall be treated in accordance with the IP Policies of DLSU. They are considered confidential information and intended only for the person/s or entities to which they are addressed. They are not allowed to be disclosed, distributed, lifted, or in any way reproduced without the written consent of the author/owner of the intellectual property."