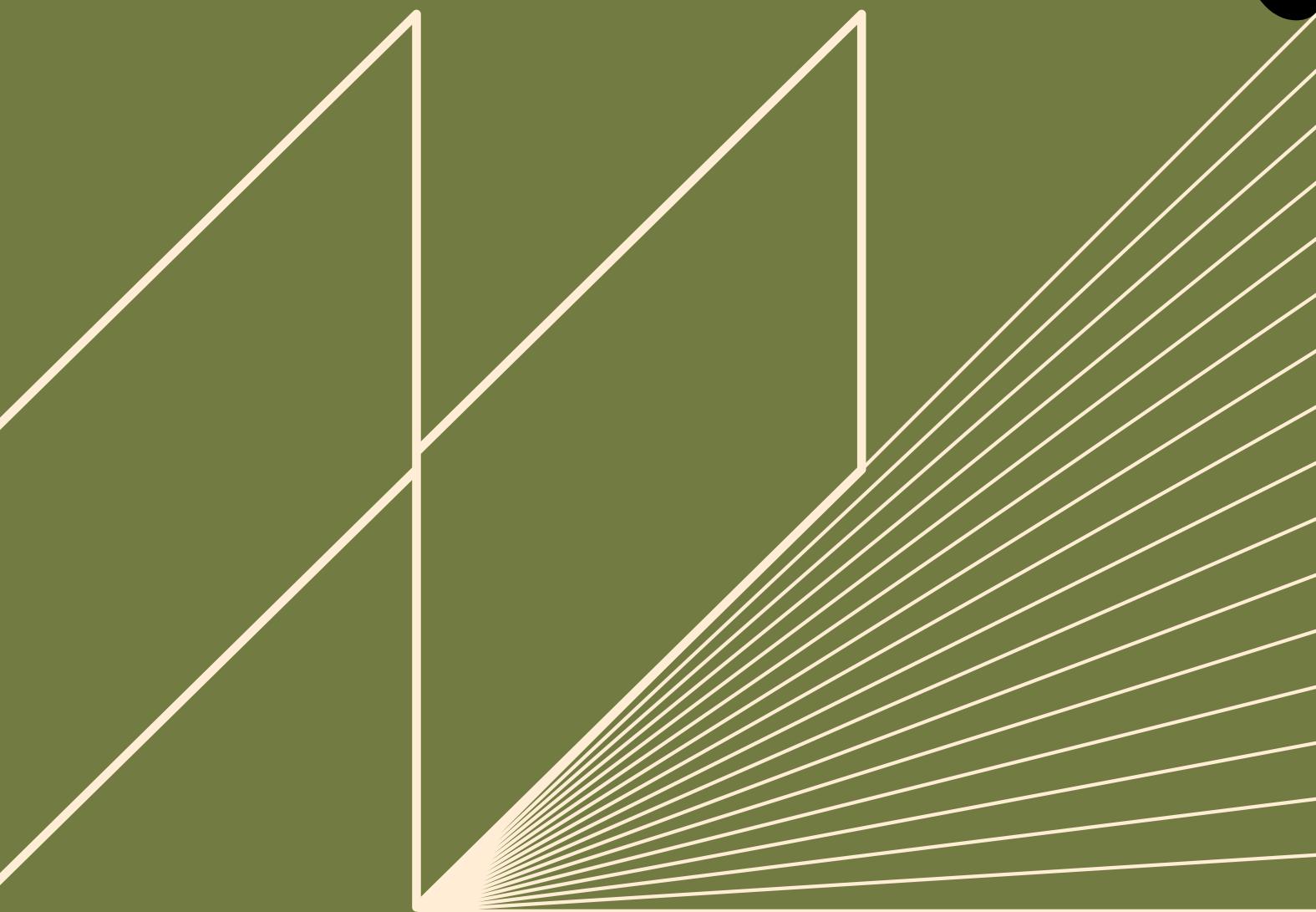


HTTP / HTTPS

The Protocols Powering the Web

Date : Jun 26, 2025



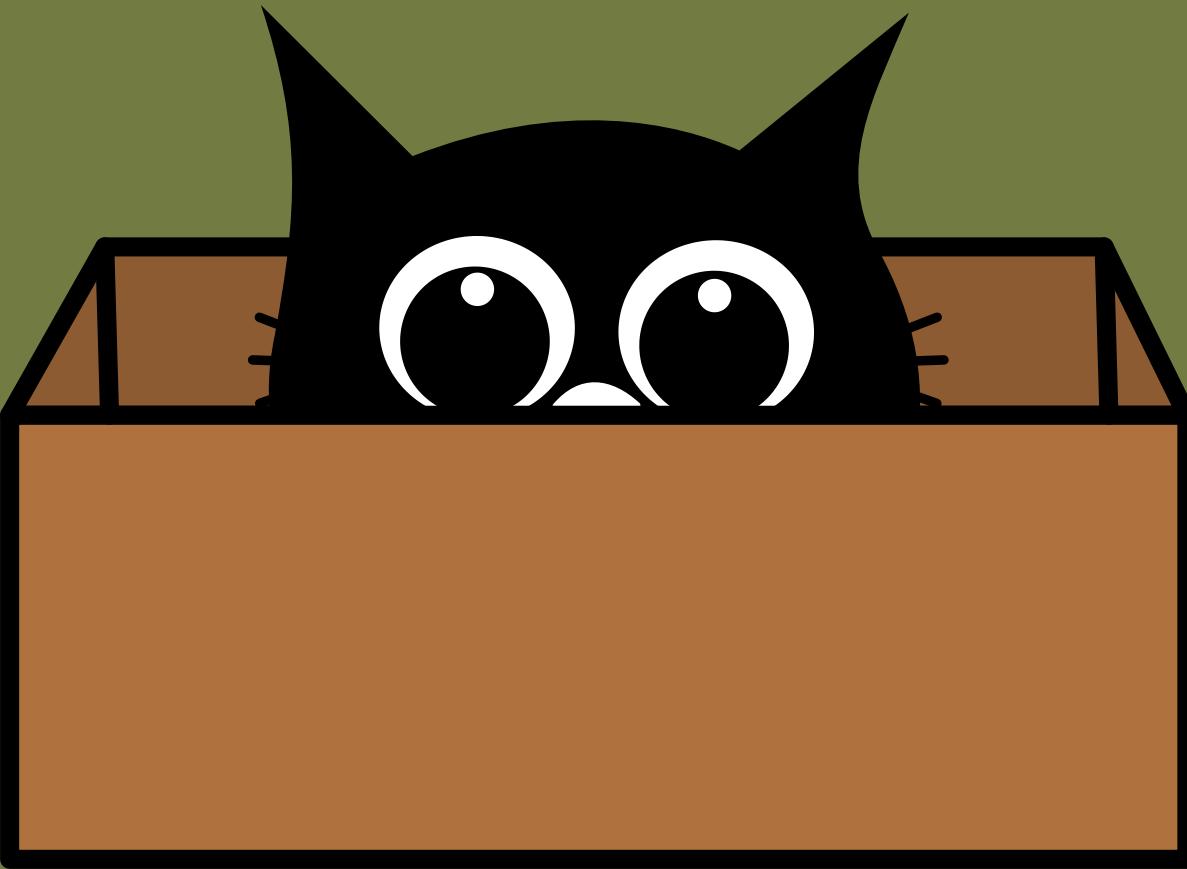
Group Members

Mithu Sharkar -2021831035

Md Mahabub Rana Saikat -2021831029

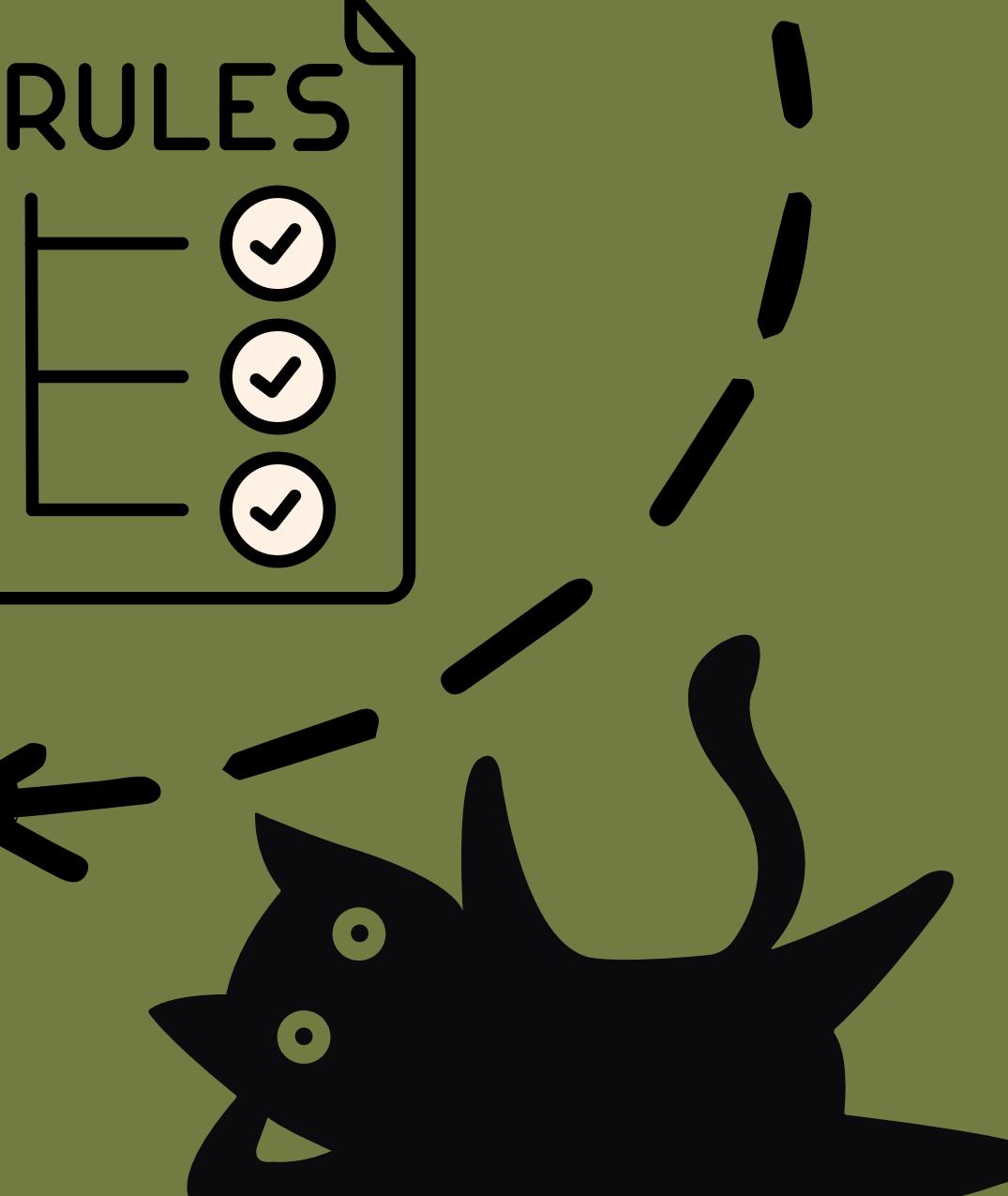
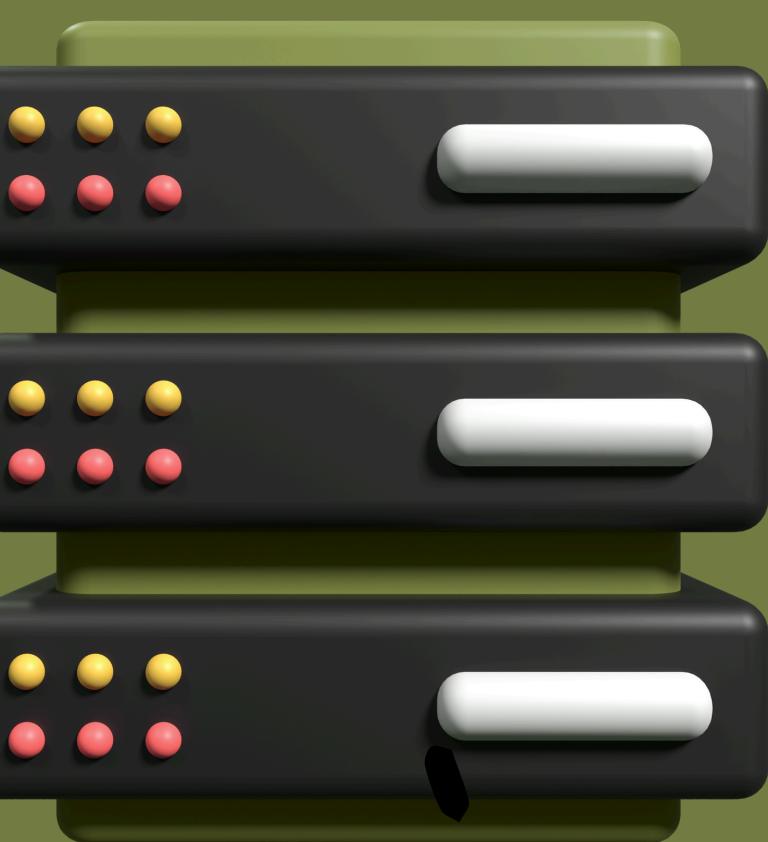
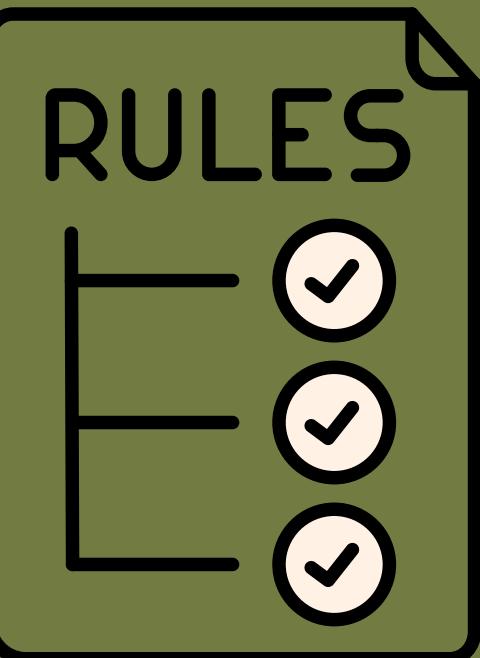
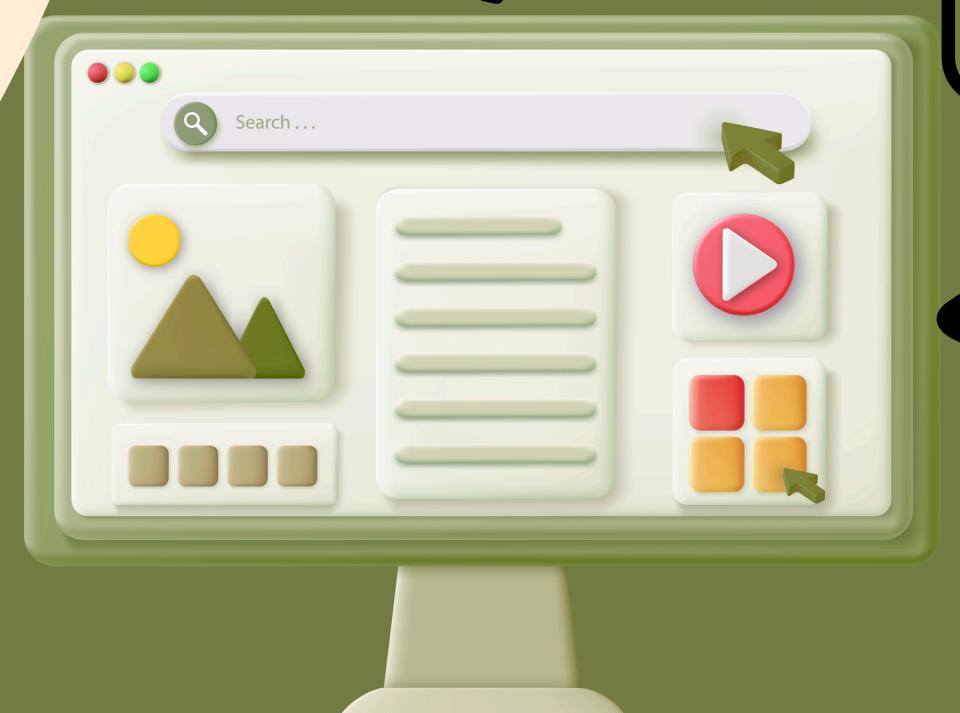
Md Altaf Hossain -2021831002

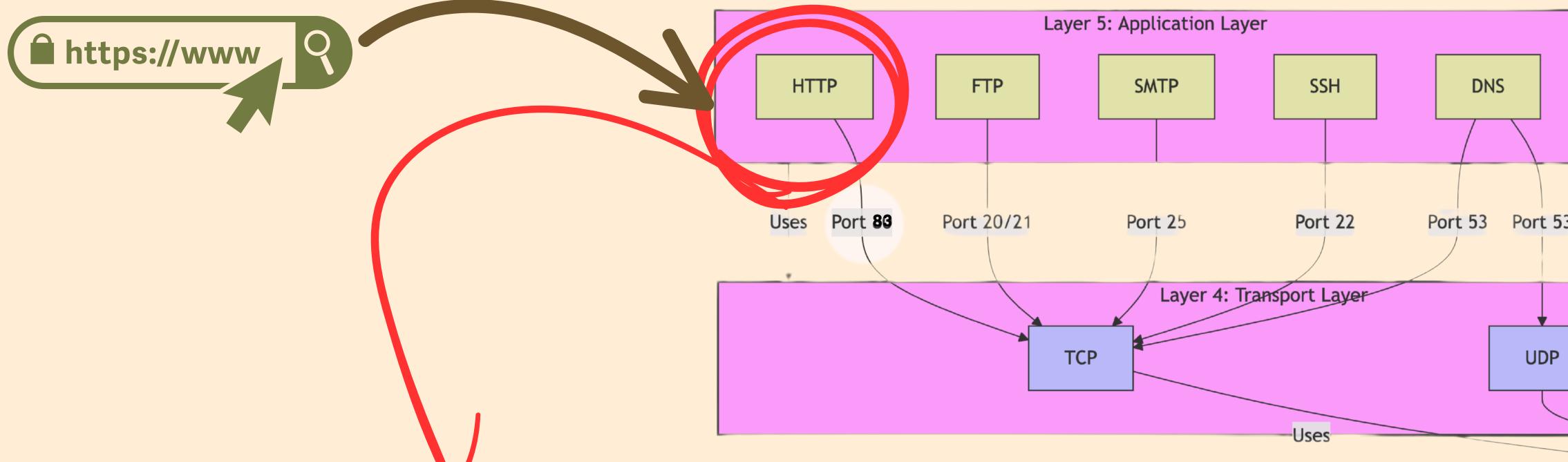
Md Sobuj Mahmud -2021831044



What is Protocol?

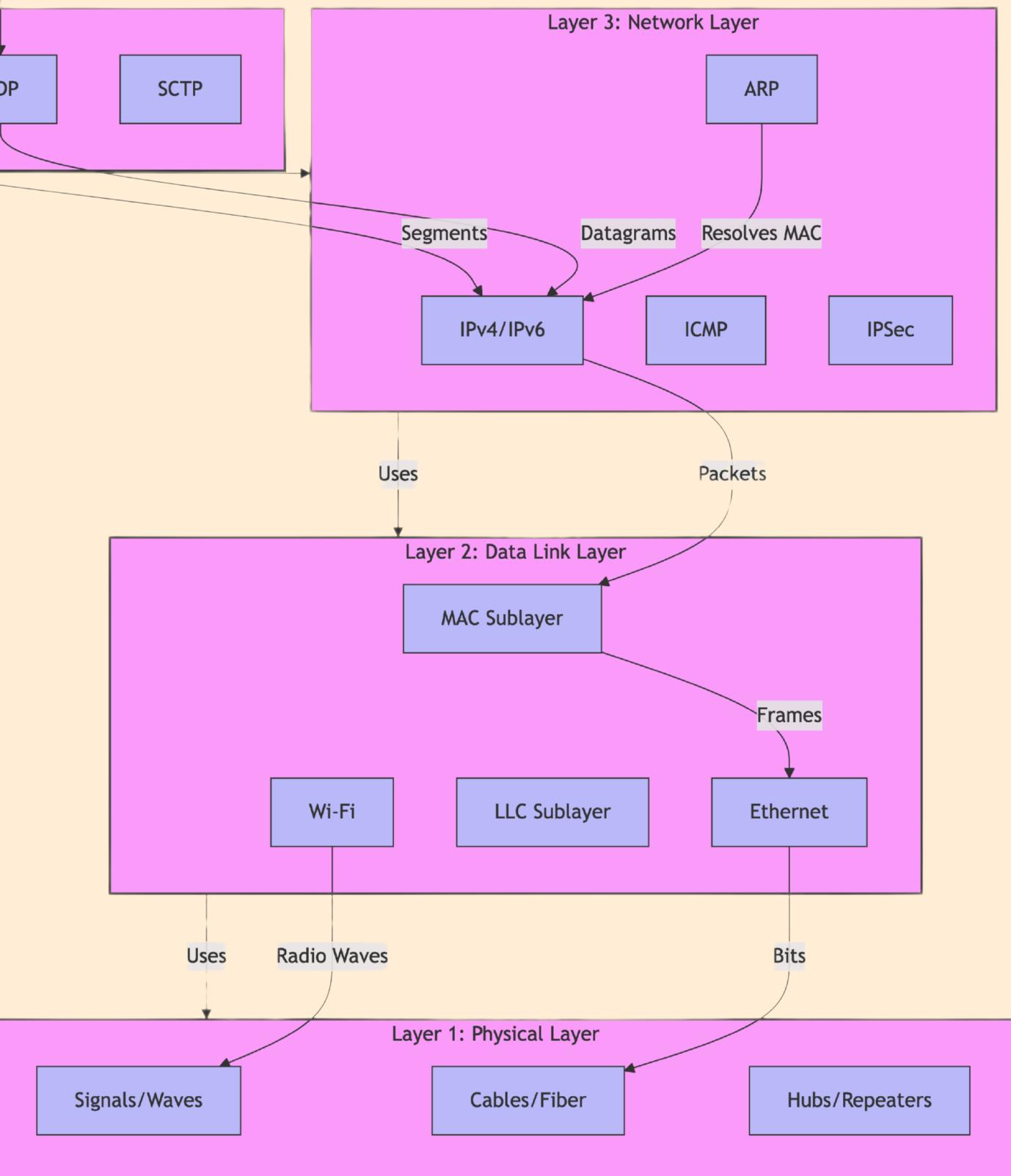
"Rules for
Communication
That Help Devices
Talk to Each Other
Smoothly"





What is HTTP/ HTTPS ?

"HTTP (HyperText Transfer Protocol) & HTTPS (HTTP Secure) – Application Layer Protocols for Sending and Receiving Data Between Web Browsers and Servers!"



HTTP

Stateless

Plaintex

Port:80

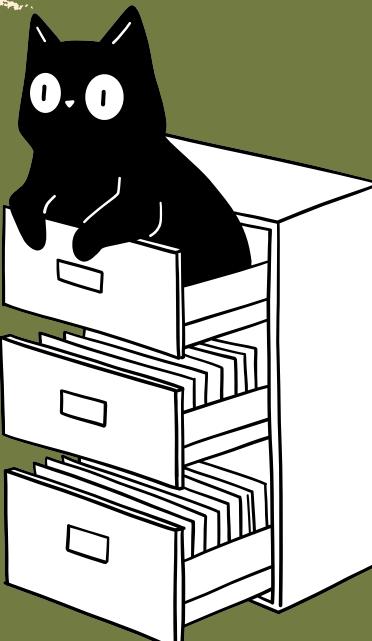
Fast but insecure

HTTPS

**Encryption
(SSL/TLS)**

Data Integrity

Authentication

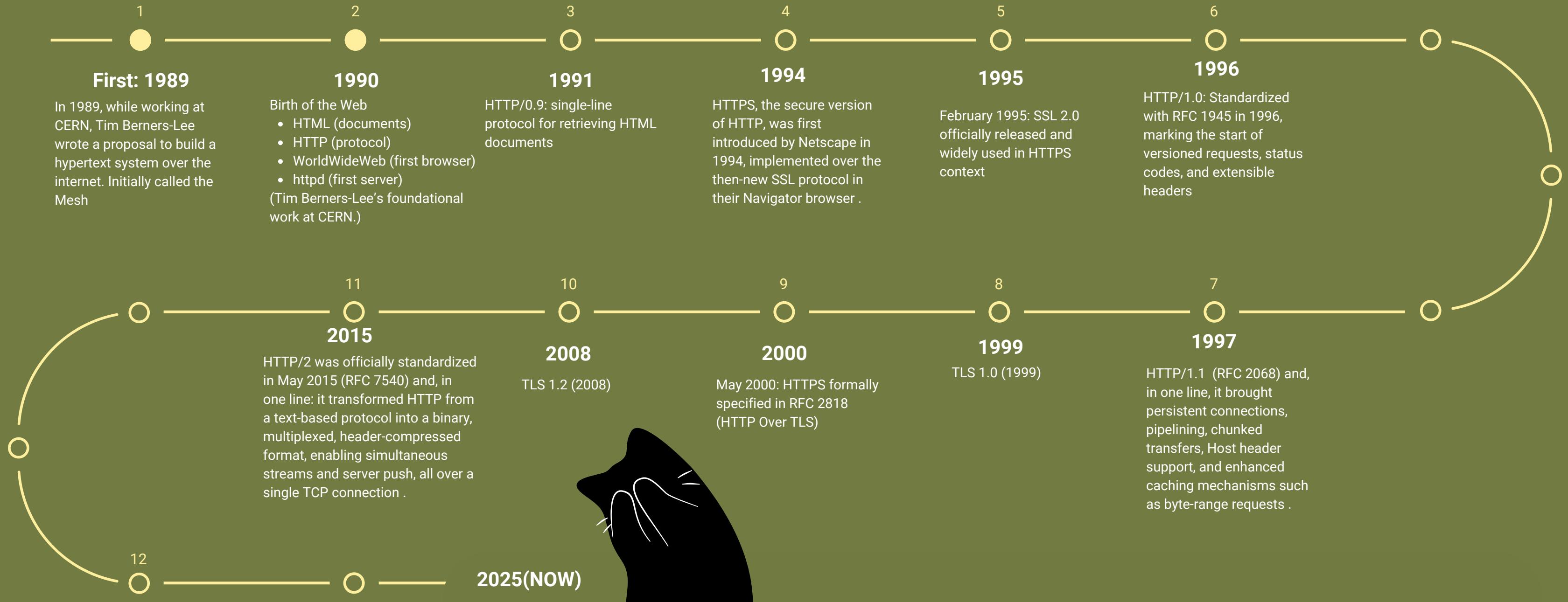


HTTP & HTTPS key Differences!



Feature	HTTP	HTTPS
Security	✗ No encryption	✓ Encrypted (SSL/TLS)
Port	80	443
Speed	Slightly faster	Minimal overhead
Verification	NO	YES(🔒)

HTTP/HTTPS EVOLUTION TIMELINE



Thank you, Tim Berners-Lee, for giving us HTTP and the Web—your vision has truly connected and empowered the world.

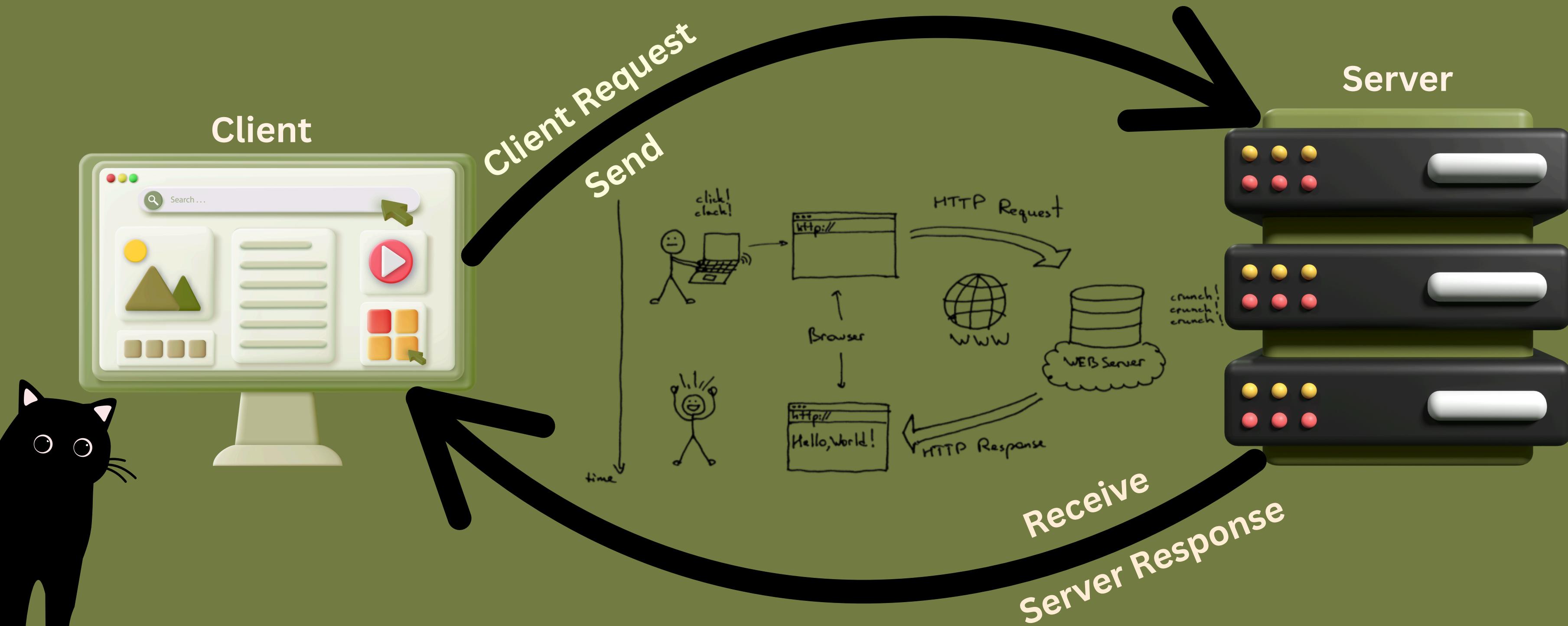


HTTP Version Comparison

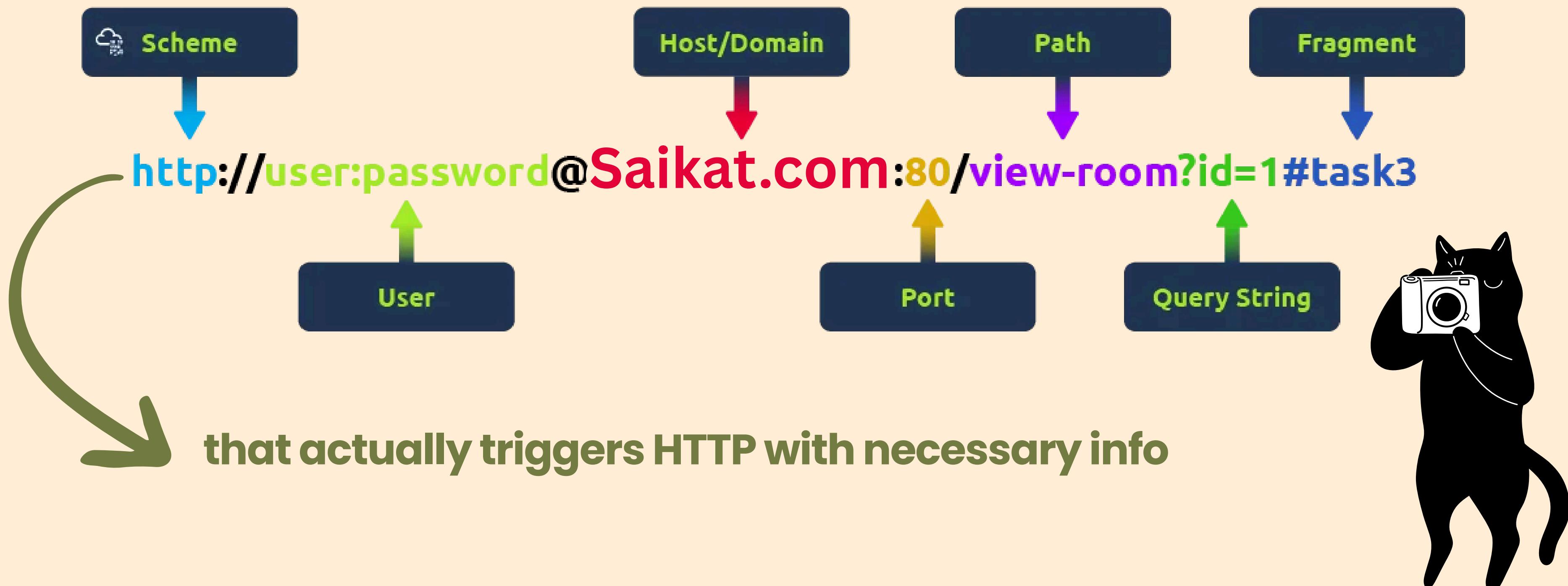
Feature	HTTP/1.1	HTTP/2	HTTP/3
Transport	TCP	TCP	QUIC (UDP)
Multiplexing	✗	✓	✓
Header Comp.	✗	✓	✓
Latency	High	Medium	Very Low
Packet Loss	Blocks all	Blocks all	Isolates streams
Encryption	Optional	De facto	Required

Does HTTP Only Send?

"No! HTTP Sends and Receives Data (Requests & Responses)."



URL Anatomy



Request

1. Method (GET, POST....)
2. Path (e.g:/index.html)
3. HTTP Version (e.g: HTTP/1.1)
4. Headers (metadata)
5. Body (optional)

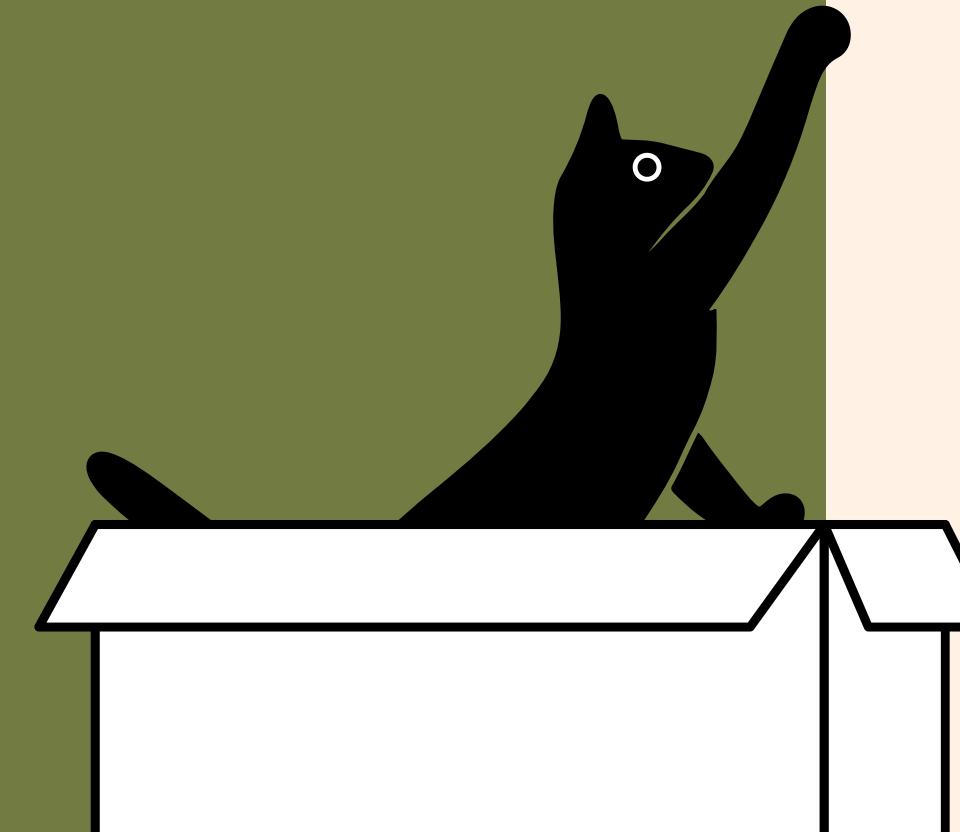
Response

1. Status Code (200 OK, 404)
2. Headers (info, type, cookies)
3. Body (HTML, JSON, images...)



Request

```
GET / HTTP/1.1
Host: tryhackme.com
User-Agent: Mozilla/5.0 Firefox/87.0
Referer: <https://tryhackme.com/>
[Blank Line]
```



Response

```
HTTP/1.1 200 OK
Server: nginx/1.15.8
Date: Fri, 09 Apr 2021 13:34:03 GMT
Content-Type: text/html
Content-Length: 98

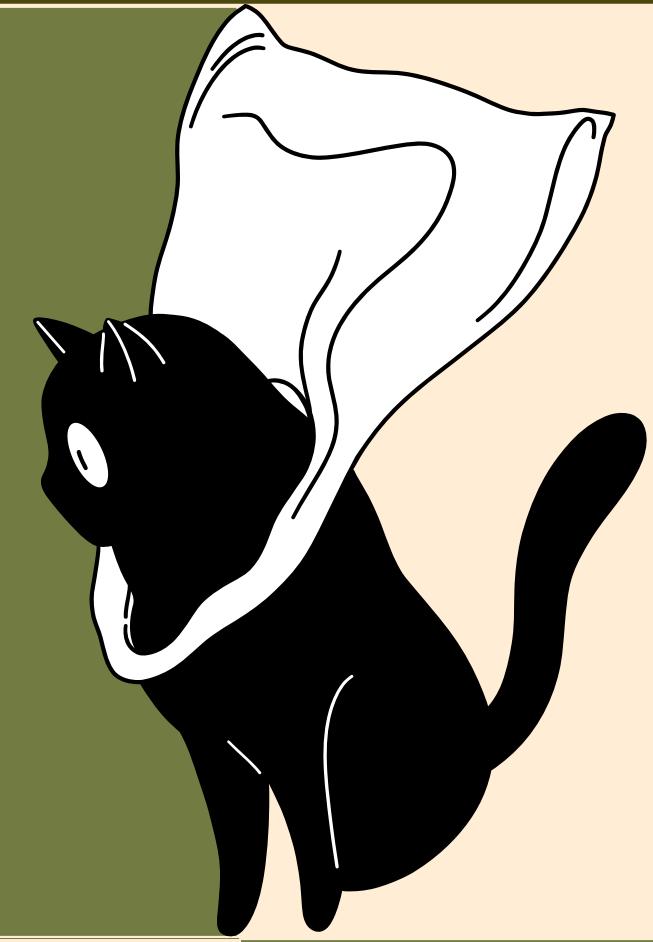
<html>
<head>
    <title>saikat</title>
</head>
<body>
    Welcome To saikat.com
</body>
</html>
```

HTTP Methods For Requests!



GET

POST



PUT

DELETE

GET

GET /products HTTP/1.1

Host: api.example.com

- **Purpose:** Retrieve data (read-only)
- **Idempotent:** Yes (safe to repeat)
- **Cacheable:** Yes
- **Body:** Never has one
- **Use Cases:**
 - Loading web pages
 - Fetching API data
 - Search queries

 **Security Note:** GET requests expose parameters in the URL - never use for sensitive data!



POST

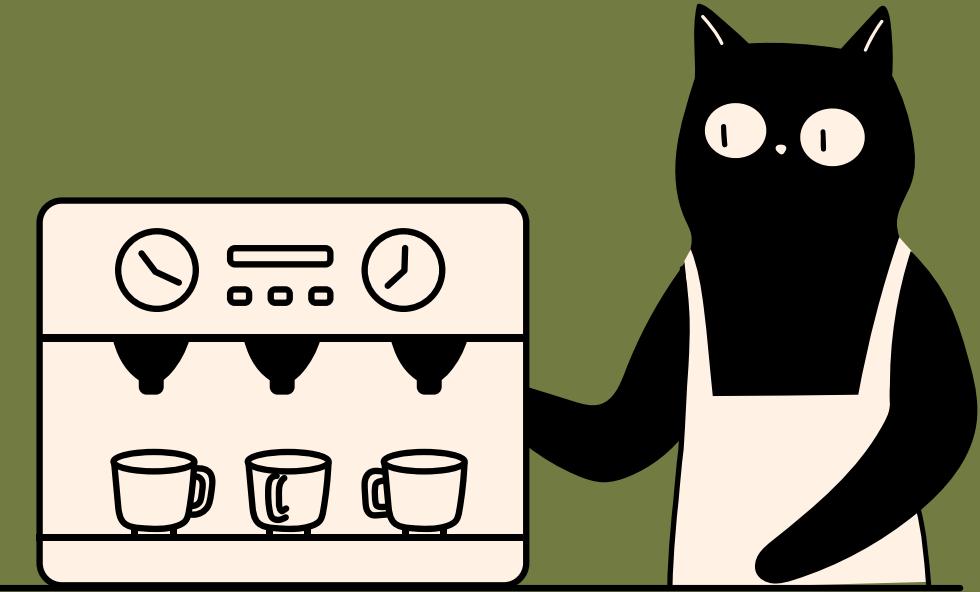
POST /users HTTP/1.1

Host: api.example.com

Content-Type: application/json

```
{"name": "John" "email": "john@example.com"}
```

- **Purpose:** Submit data to create new resources
- **Idempotent:** No (repeating may create duplicates)
- **Cacheable:** Only if explicitly allowed
- **Body:** Almost always has one
- **Use Cases:**
 - Form submissions
 - User registration
 - File uploads



PUT

```
PUT /users/123 HTTP/1.1
```

```
Host: api.example.com
```

```
Content-Type: application/json
```

```
{"name": "John Doe", "email": "johndoe@example.com"}
```

- **Purpose:** Update/replace entire resource
- **Idempotent:** Yes
- **Cacheable:** No
- **Body:** Contains complete new version
- **Use Cases:**
 - Updating user profiles
 - Replacing document versions



DELETE

`DELETE /users/123 HTTP/1.1`

`Host: api.example.com`

- **Purpose:** Remove resources
- **Idempotent:** Yes
- **Cacheable:** No
- **Body:** Usually none
- **Use Cases:**
 - Deleting accounts
 - Removing files
 - Canceling orders



HTTP Response Status Codes!



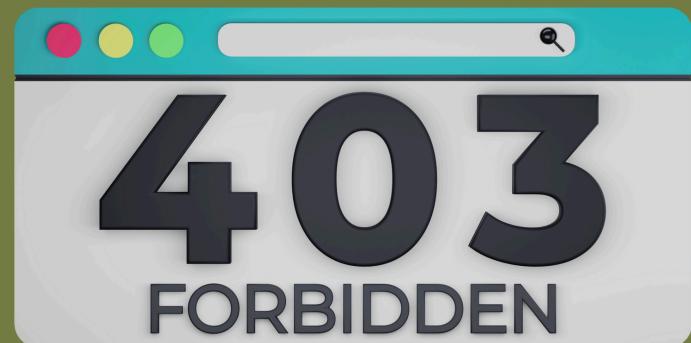
2XX - Success



3XX-Redirection



4XX- Client Error



5XX-Server Error



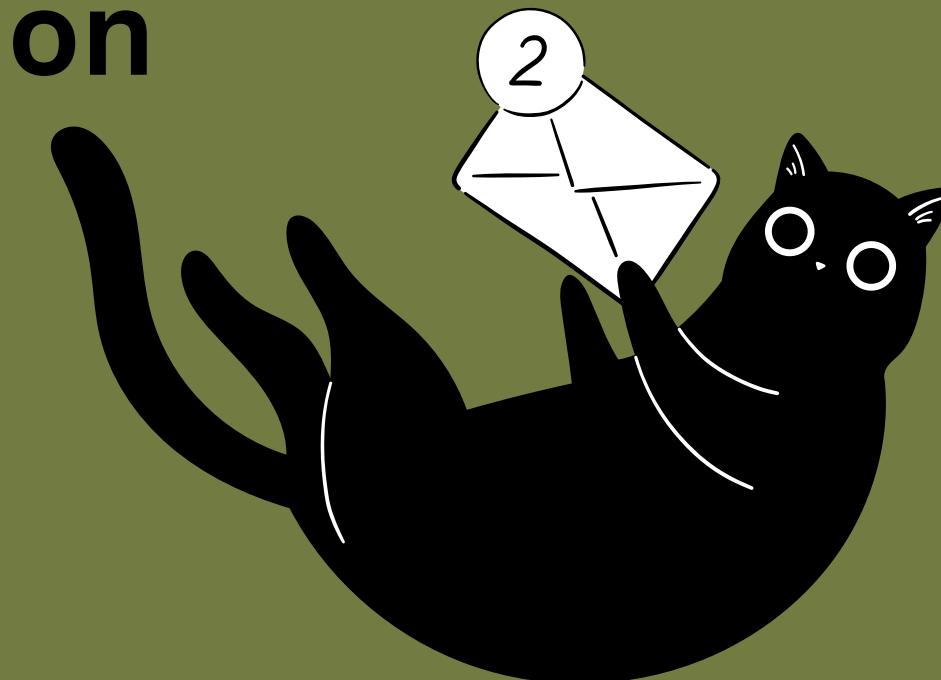
1XX - Informational (Rarely Used)

HTTP Headers



Request

1. Host
2. User-Agent
3. Content-Length
4. Accept-Encoding
5. Authorization
6. Accept



Response

1. Content-Type
2. Cache-Control
3. Set-Cookie
4. Content-Encoding
5. Strict-Transport-Security
6. Permissions-Policy
7. Referrer-Policy
8. ETag



HTTP Cookies: The Web's Memory!

Client

Server



GET /index.html HTTP/1.1

HTTP/1.1 200 OK

Set-Cookie: name=supsup; Expires=Wed, 21 Oct
2018 09:00:00 GMT; Domain=ssup2.com; Path=/

GET /home/address.html HTTP/1.1

Cookie: name=supsup

Does HTTP Works Only on Browser?

NO!

"Seriously? You
guys use my
name for
concatenation in
programming?
Huh!!!"



Q&A

“Let’s begin the Q&A. And by ‘Q,’ I mean please don’t. And by ‘A,’ I mean absolutely not.”

THANKYOU

