# HOW INTERNET WORKS

## From URL to Web Page

By Md. Tanjim Mahmud Tuhin
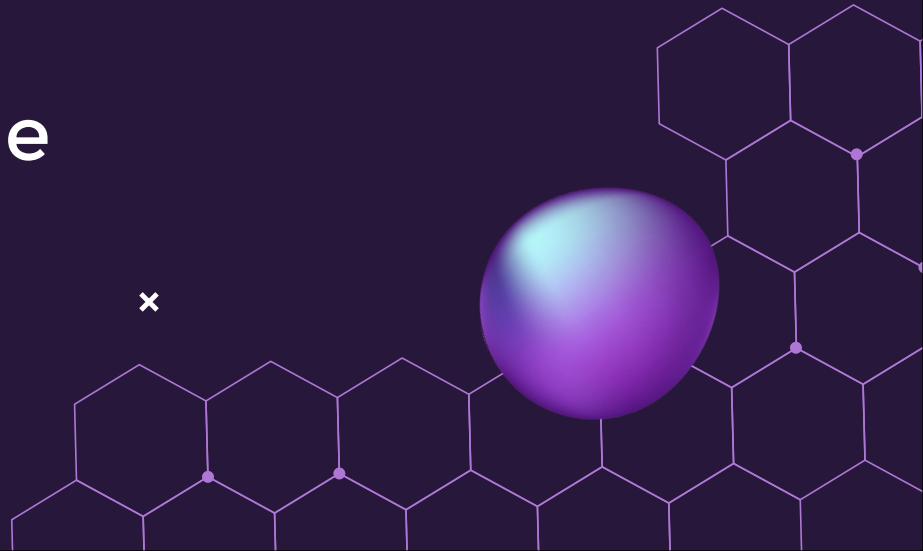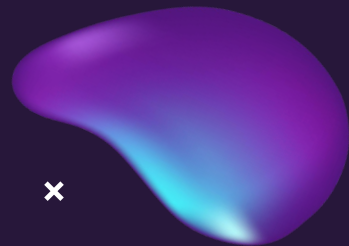251-56-012

# TABLE OF CONTENTS

# TABLE OF CONTENTS

# TABLE OF CONTENTS

# Ever wondered...

What actually happens when you type a URL like www.google.com and press Enter?

# Why it matters:

For our existence actually.

# In this presentation

How your browser finds the website you're looking for.
What DNS, IP, and HTTP mean in this process.
How data travels across the internet.
How your browser turns code into a web page.
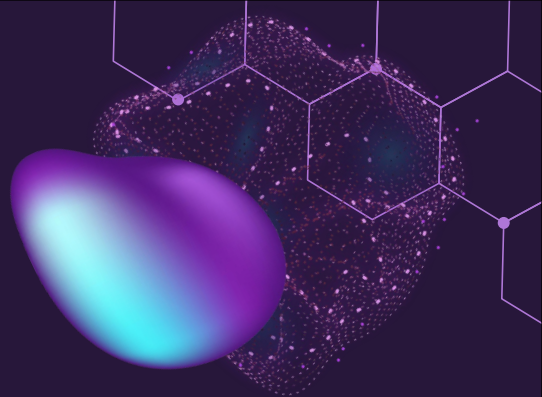
# The Journey Begins – From URL to Web Page



Type URL or Click Link

URL goes to DNS Server
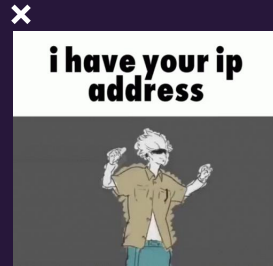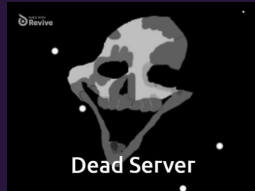
From server get IP address

Browser Sends HTTP Request
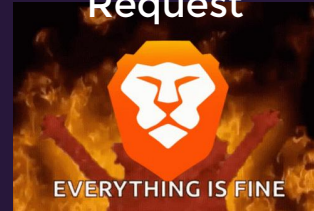
Server Responds with Data

Browser Renders the Page

# What is Internet Really

The internet is a global network of interconnected computers and devices that communicate with each other using standardized protocols (like TCP/IP).

The internet's not a cloud, just wires and servers with hard drives storing web pages.
DSL, cable, fiber, or satellite - each connects you to that data, fast or slow.

# 01 Domain Name System (DNS)

## What is DNS? ✕

DNS is like the internet's phonebook – it converts human-friendly domain names (like facebook.com) into IP addresses (like 157.240.22.35) that computers can understand.

## Why It's Needed:

Humans remember names (e.g., youtube.com)
Computers use IP addresses to locate websites
DNS acts as the translator between them

# DNS in Action

## Step-by-Step Breakdown:

1. You type `www.google.com` and press Enter
2. Browser checks local DNS cache – is the IP already saved?
3. If not, it asks your system's DNS resolver(e.g., from your ISP)
4. Resolver queries:
    - Root DNS Server – where to look next?
    - .com TLD Server – gives Google's DNS info
    - Authoritative DNS Server – gives final IP address (e.g., `142.250.183.132`)
5. The IP is returned to your browser
6. Now, your browser can contact the Google web server
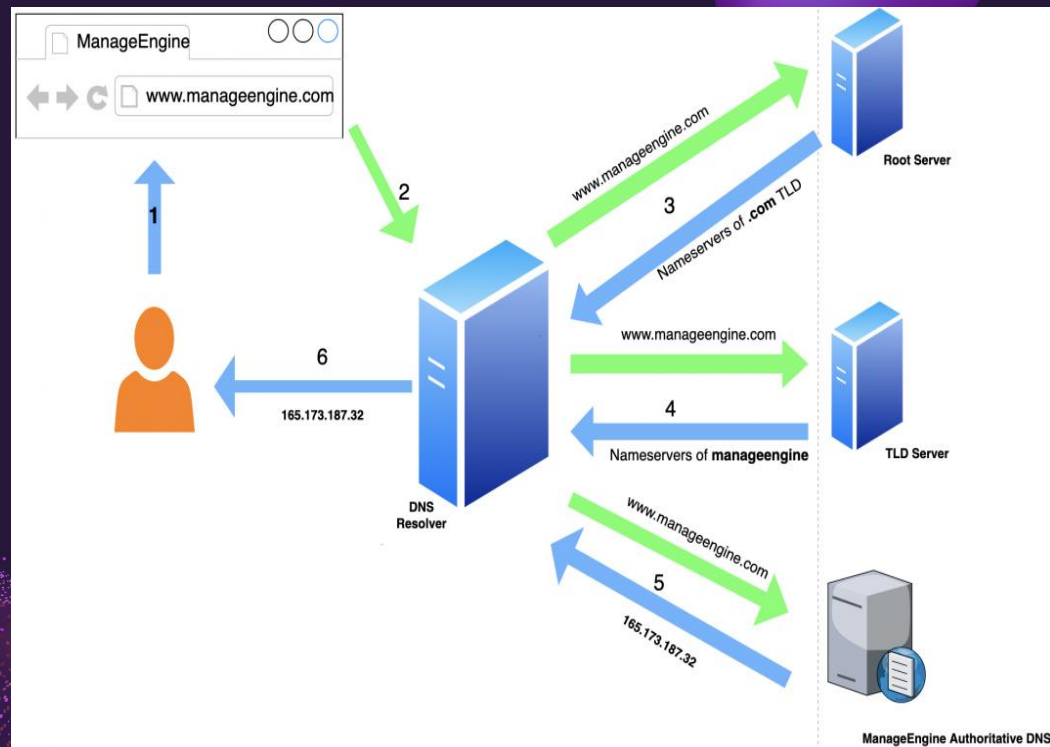
# 02 IP Address and Routing
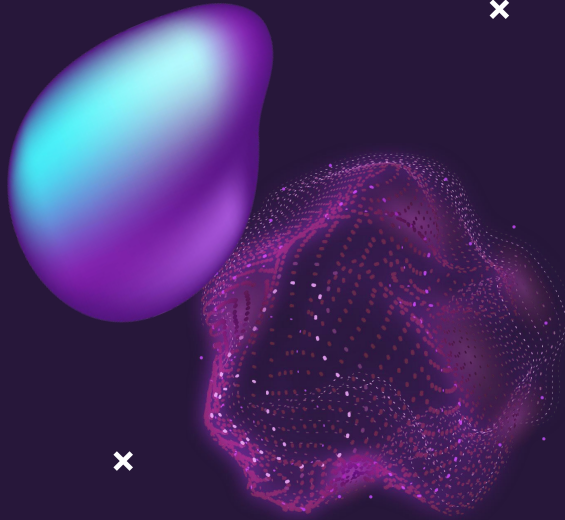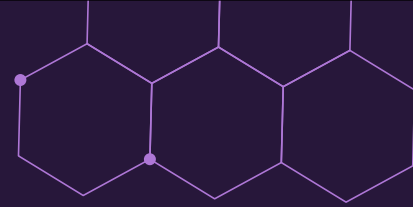
## What is an IP Address?

An IP address is like the home address of a device on the internet. It tells the internet where to send data.

Example: 142.250.183.132 = Google's server IP
Two types:
IPv4 (e.g., 192.168.1.1)
IPv6 (e.g., 2001:0db8:85a3::8a2e:0370:7334) – newer, longer

## How Routing Works

Once the IP is known, the data (your request) travels across many routers and networks

Each router makes a decision: "What's the best next hop to reach this IP?"
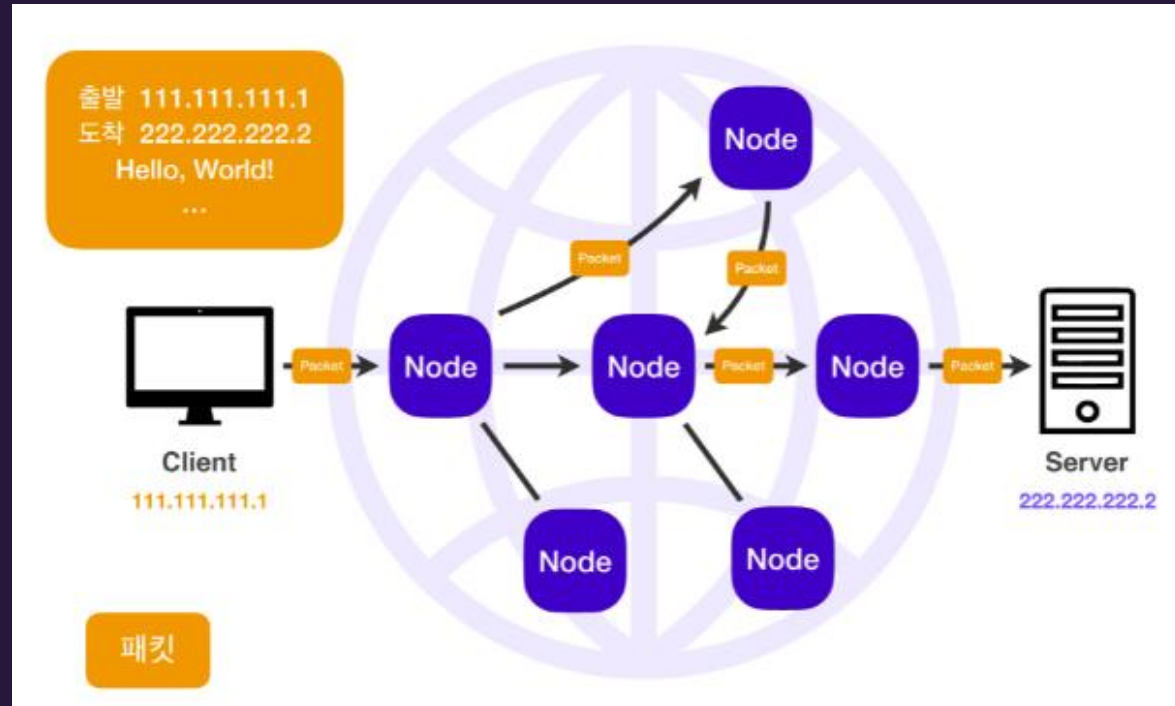
# IP Packet Journey

## What is a Packet?

A packet is a small chunk of data sent across the internet.Your browser's request is split into multiple packets and sent piece by piece.

## How It Works:

➡ Each packet contains:
1.Source & destination IP
2.Part of the message (e.g., a piece of a webpage)

➡ Routers inspect the IP and forward packets toward their destination

# 03 TCP/IP Protocol

## What is TCP/IP?

TCP/IP is the foundation of the internet.
It defines how data is packaged, sent, received, and understood between devices.

## 4 Key Layers of TCP/IP Model:

### Application Layer

Where apps like browsers live

Protocols: HTTP, HTTPS, FTP, DNS

### Transport Layer

Ensures reliable data delivery
TCP = Reliable(sites)
UDP = Fast but no guarantee (streaming)

### Internet Layer

Handles IP addressing and routing

Protocol: IP

### Network Layer

Connects to physical network

(Ethernet, Wi-Fi)

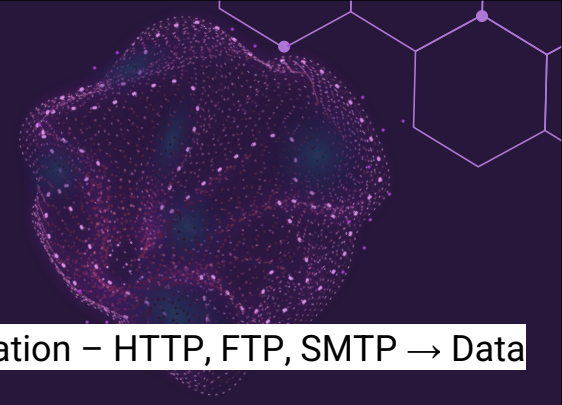# How TCP/IP works



TCP/IP Model

Applications

TCP

IP

Network

REALPARS

5. Application – HTTP, FTP, SMTP → Data

4. Transport – TCP/UDP → Segment

3. Network – IP, Routers → Packet

2. Data Link – Ethernet, Switches → Frame

1. Physical – Cables, NICs → Bits

# TCP 3 way handshake

## Before sending data, TCP does:
Client: SYN �
Server: ←    SYN-ACK
Client: ACK ➤

## Quick Analogy:

Think of TCP/IP like sending a letter:
TCP = careful post office (guarantees delivery),
IP = address on the envelope.



| Client | | Server |
|---|---|---|
| | SYN | |
| | SYN + ACK | |
| | ACK | |
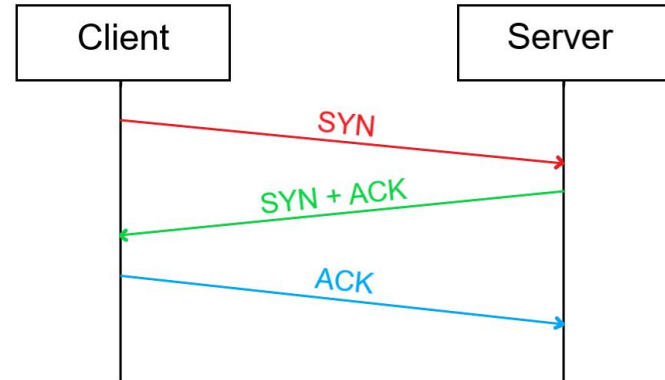
**3-Way TCP Handshake**

# 4

# HTTP Request

## 🌐What is HTTP?

HTTP (HyperText Transfer Protocol) is how your browser talks to web servers to request pages, images, and more.
You type a URL → Browser sends an HTTP GET request
The server receives it, processes it, and prepares a response
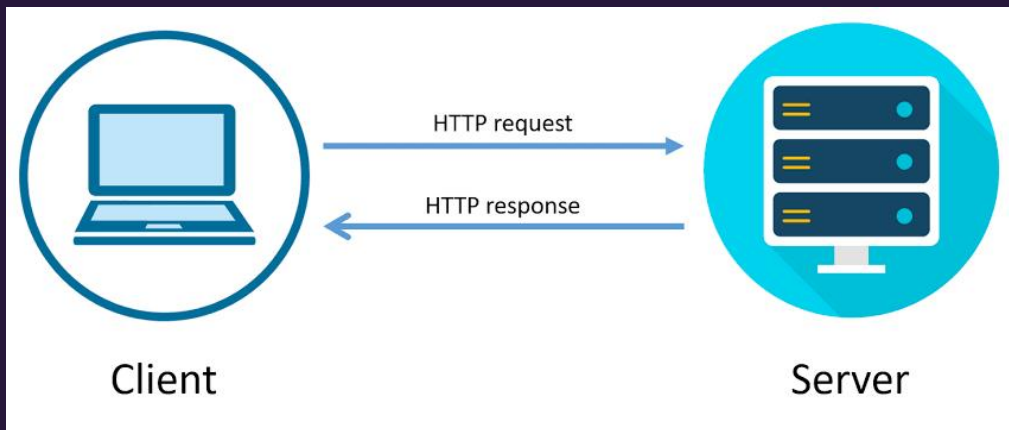
## 🧠Analogy:

Think of HTTP like a restaurant order:

You (browser) = Customer
HTTP GET = You place an order
Server = Kitchen
HTTP Response = Your food is served (HTML, CSS, images)



HTTP request

HTTP response

Client                          Server

# Server Response – Getting the Web Page Back

## 📧What Happens After the Request?

The server receives your HTTP request and responds by sending the web page data – usually HTML, CSS, JavaScript, and images.



## 📦Data Returned:

HTML – The structure of the webpage
CSS – Styling (colors, layout, fonts)
JavaScript – Interactive elements
Images, fonts, icons – All downloaded separately

# Browser Rendering – Turning Code into a Web Page

⚙ **What does the browser do with the data?**

It takes the HTML, CSS, and JavaScript, and turns it into the visual page you see.
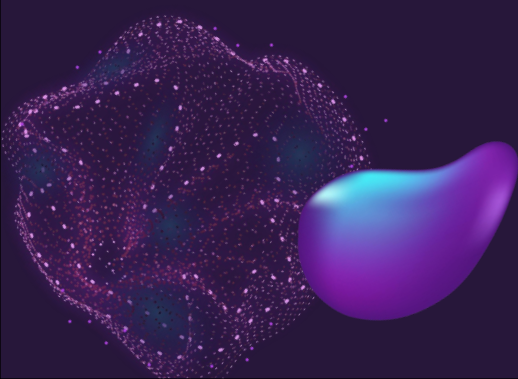
🔄**Rendering Steps (Simplified):**

Parse HTML – Browser builds the DOM (Document Object Model)

Parse CSS – Applies styles to the DOM

Run JavaScript – Adds interactivity or modifies content

Layout & Paint – Calculates positions and draws elements

Composite – Final page appears on your screen

# Caching and Performance – Making the Web Faster

×

## 🚀 What is Caching?

Caching means saving copies of web content so it loads faster the next time you visit.

❏ Your browser stores HTML, images, CSS, etc. locally
❏ If unchanged, it doesn't re-download every time

**Types of Caching:**
➢ Browser Cache – Stored on your device
➢ DNS Cache – Stores recently used IP addresses
➢ Server-Side Caching – Stores dynamic content on servers
➢ Content Delivery Networks (CDNs) – Store content on global servers for quicker access
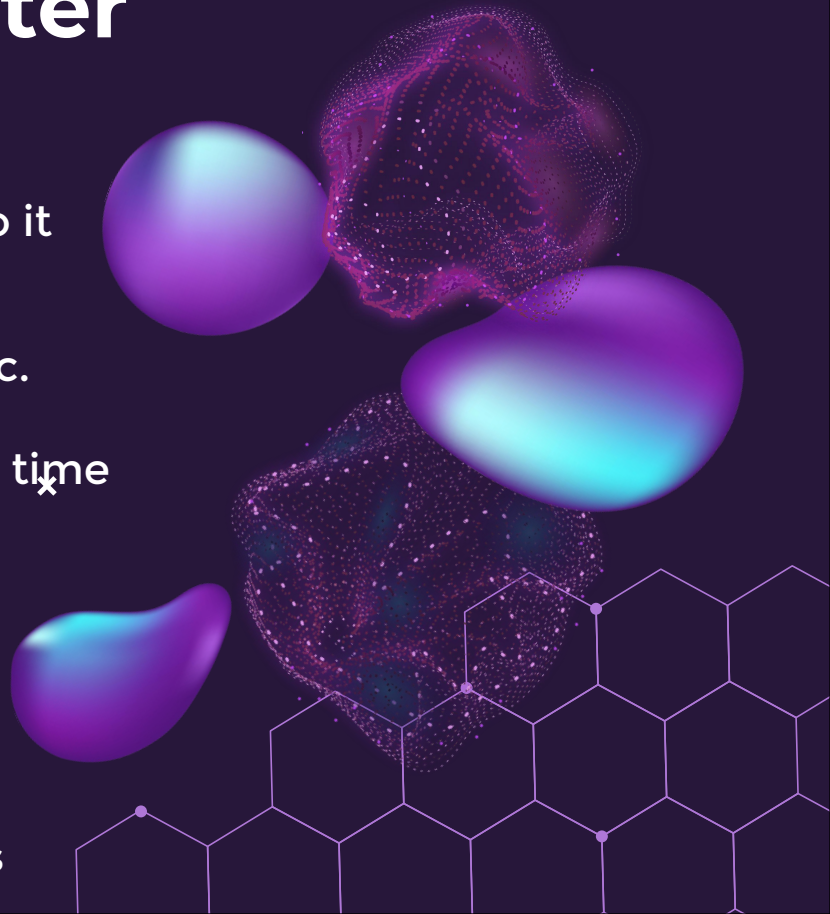
# Security in the Journey – Keeping Data Safe

## Why is Security Important?

Data travels across the internet, and security ensures that sensitive information doesn't get intercepted or tampered with.

## Security Protocols in the Process:

**HTTPS (SSL/TLS)**
Encrypts data between your browser and the server
Makes sure the server is authentic
You see the padlock icon in the address bar for secure sites
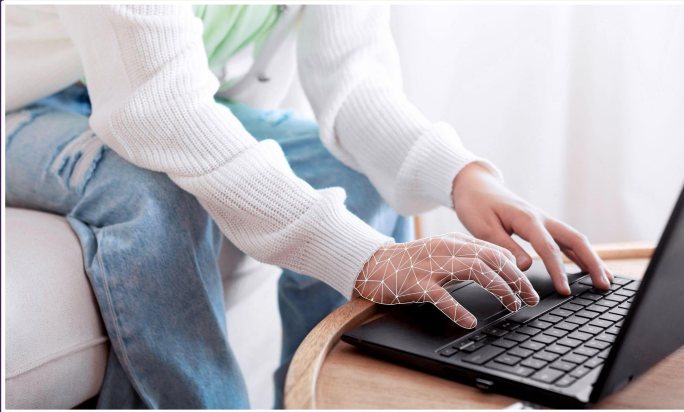
**DNSSEC**
Protects DNS from being spoofed or hacked
Ensures the IP address returned by DNS is the right one

# How SSL/TLS Works:

Browser & server agree on encryption keys

All data exchanged is encrypted

Only the server and browser can decrypt it

# Common Threats:

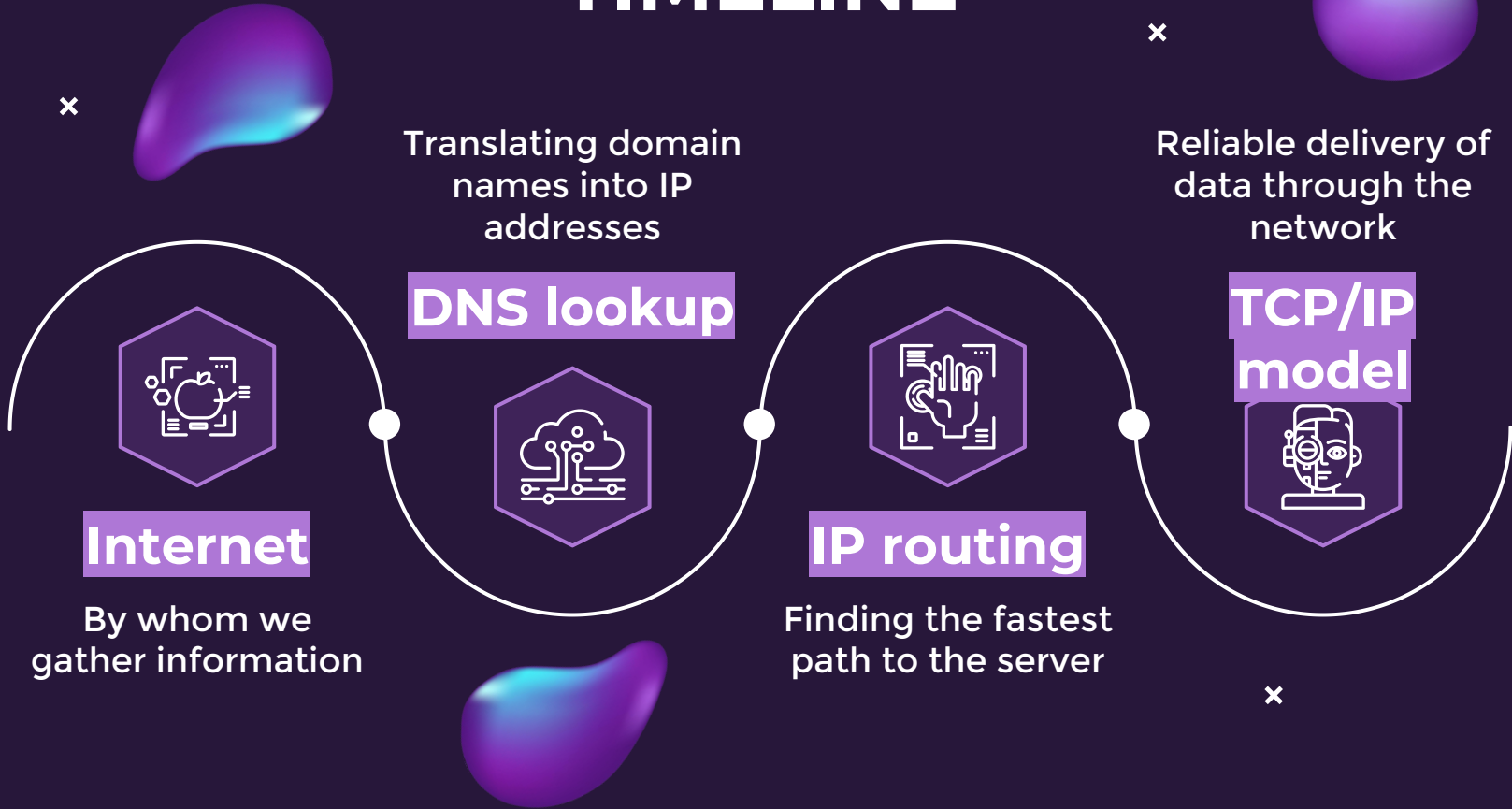**Man-in-the-Middle Attacks(MITM)**

Interception of data

**Phishing**

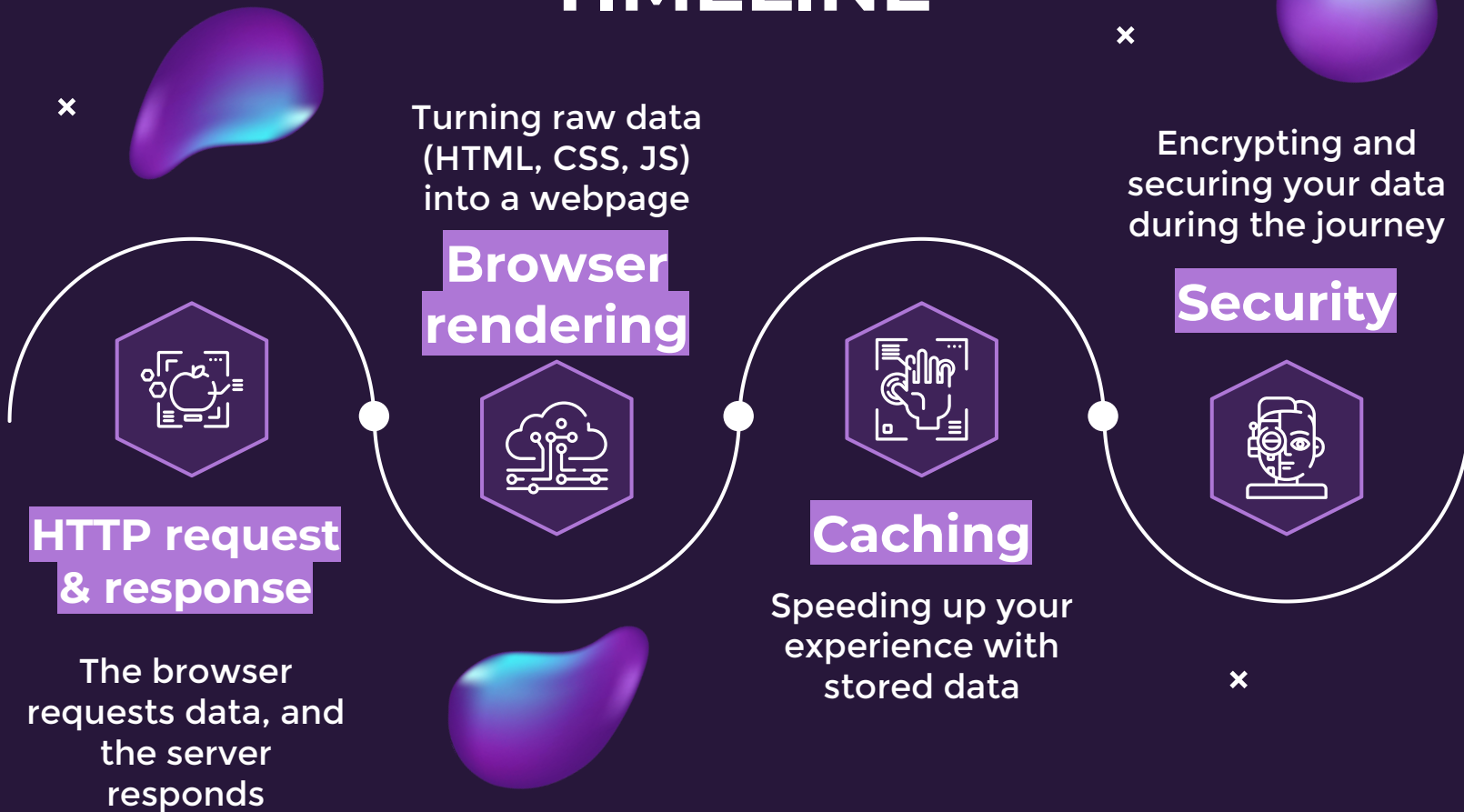Fake websites tricking you into revealing info

**DNS Spoofing**
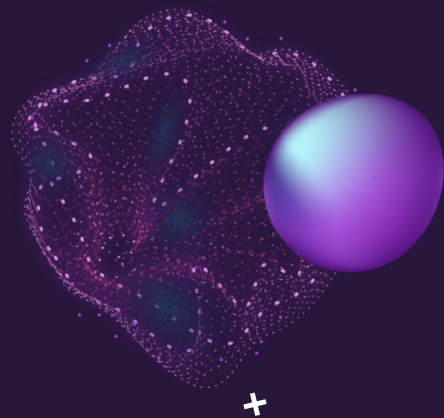
Redirecting traffic to malicious sites

# TIMELINE

**Internet**

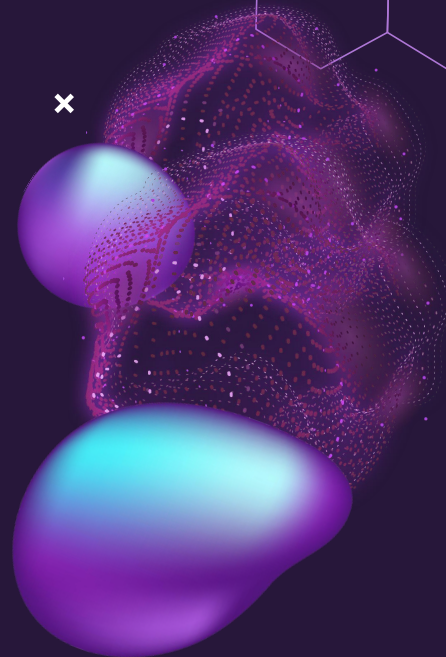By whom we gather information

Translating domain names into IP addresses

**DNS lookup**

**IP routing**

Finding the fastest path to the server

Reliable delivery of data through the network

**TCP/IP model**

# TIMELINE

Turning raw data
(HTML, CSS, JS)
into a webpage

**Browser
rendering**

Encrypting and
securing your data
during the journey

**Security**

**HTTP request
& response**

**Caching**

The browser
requests data, and
the server
responds

Speeding up your
experience with
stored data

# ANY QUESTION!!!

THANK YOU ETERNALLY