Networks, Internet and Protocols

Network Fundamentals, Network Services

SoftUni Team

Technical Trainers

Software University

SoftUni

Software University

https://about.softuni.bg

Have a Question?

Software University

sli.do

#qa-fund

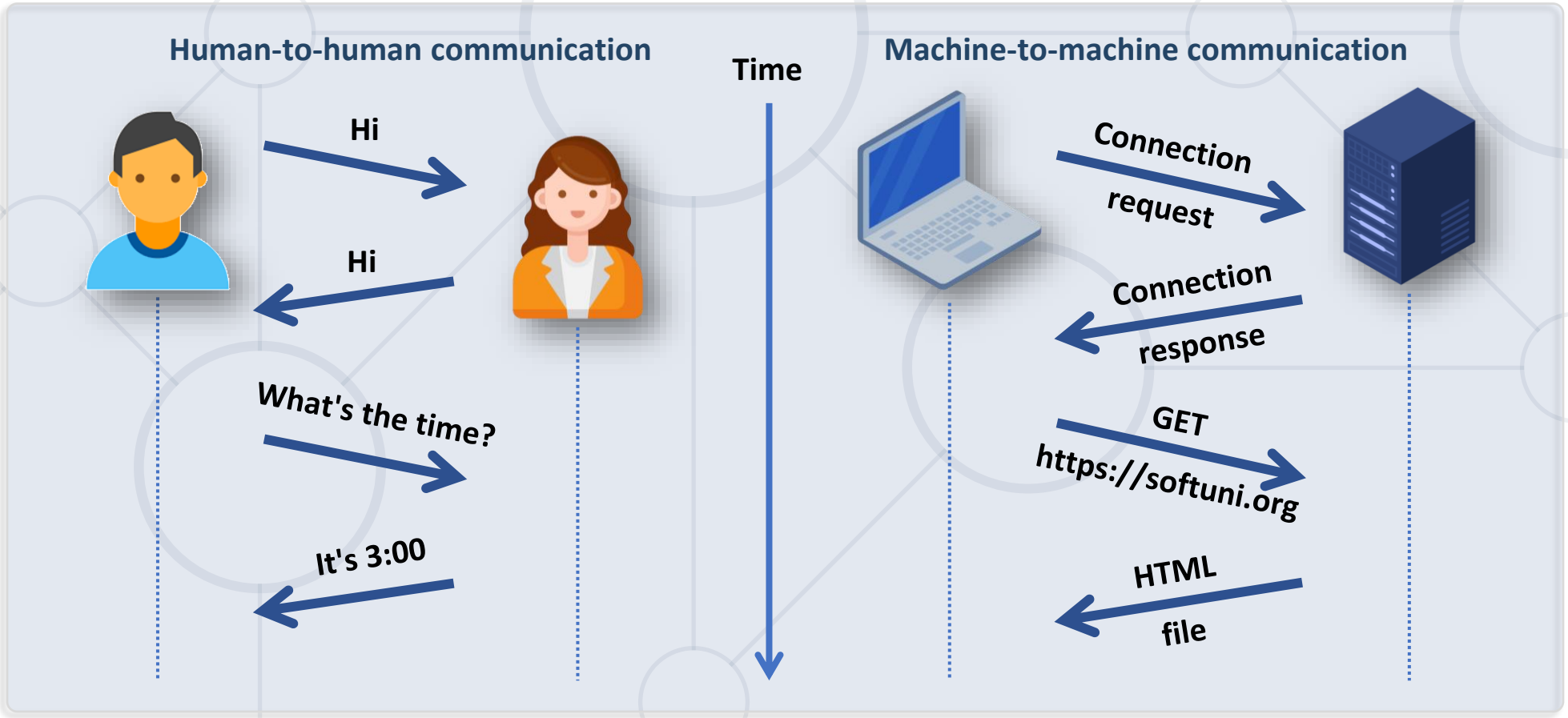
Table of Contents

Network Fundamentals

OSI Model, MAC Address, IP Address, TCP and Ports

What is a Network Protocol?

- **Network protocol** - a **set of rules** that determine how data is transmitted between different devices on the same network



Network Protocols: Essentials

- **Network protocols** enable **standardized communication** between devices / programs
  - Typically, one party sends a **request** (command / question / other) and receives a **response** from the other party
- **Network protocols** govern aspects of **data transmission**, **addressing**, **routing**, **flow-control**, and **error handling**
- Most protocols are described in **public documents**
  - Example: https://www.rfc-editor.org/rfc/rfc5321

Network Layering Models

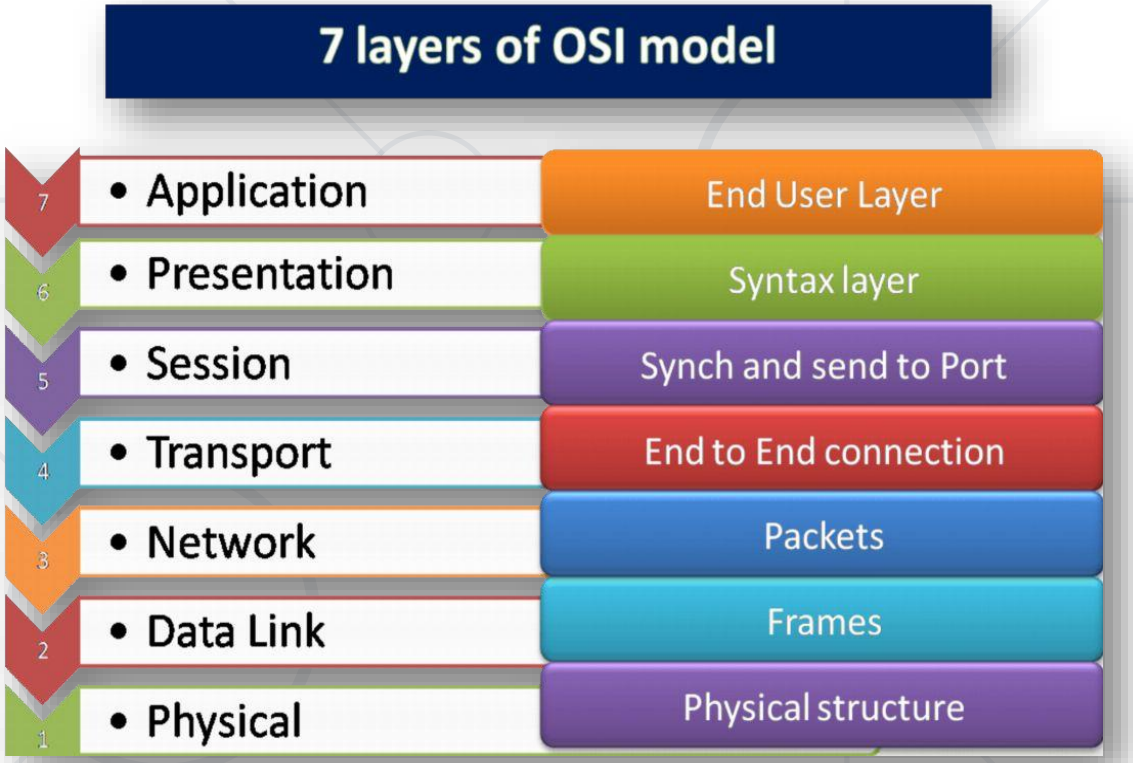
- **Layers** organize networking into a **structured framework**
  - Facilitate the understanding, design, and management of **complex networks**
  - **Simplifies** network communication and troubleshooting
  - Encourages protocol **interoperability** and **modularity**
- Examples:
  - **OSI** model (7 layers)
  - **TCP/IP** model (4 layers)

The OSI Model

Understanding the 7 Network Communication Layers

OSI Model Overview

- The **OSI Model** (**O**pen **S**ystems **I**nterconnection Model)
- Developed by the International Organization for Standardization (**ISO**) in the **1970s**
- **Framework** for understanding and designing network protocols and communication

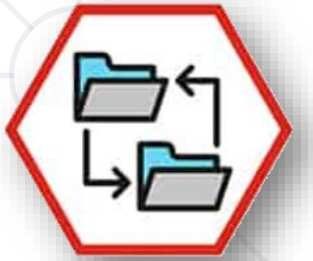


- **7 layers** – each layer stacks on the previous and **adds functionality** to the data transmitted

OSI Model Layers

- **Physical Layer (Layer 1)** – cables and radio
  - **Converts digital data** into electrical impulses, radio signals, or optical signals for transmission
  - **Devices**: hubs, repeaters, antennas
  - **Protocols**: **Ethernet**, **Wi-Fi**, **Bluetooth**, **USB**, **RS-232**

- **Data Link Layer (Layer 2)** – MAC address, frames
  - **Manages data transmission**, error detection / correction
  - **MAC address**: unique identifier for network interfaces
  - **Devices**: switches, bridges, network interface cards (NICs)
  - **Protocols**: **Ethernet**, Point-to-Point Protocol (**PPP**)



OSI Model Layers

- **Network Layer (Layer 3)** – hosts and IP address, packets
  - **Packet routing:** host → router → router → … → end host, Shortest Path First (SPF), Distance Vector (DV), Link State (LS)
  - **Devices**: routers, layer 3 switches
  - **Protocols**: Internet Protocol (**IP**), **IPv6**, Internet Control Message Protocol (**ICMP**), **IPsec** (IP security), **ARP**

- **Transport Layer (Layer 4)** – ports
  - **Error checking**, flow control, congestion control, multiplexing
  - **TCP** – session-based bi-directional, reliable communication
  - **UDP** – fast, best-effort single packet delivery (connectionless)
  - **QUIC** – modern session-based protocol, multiplexed, low-latency

OSI Model Layers

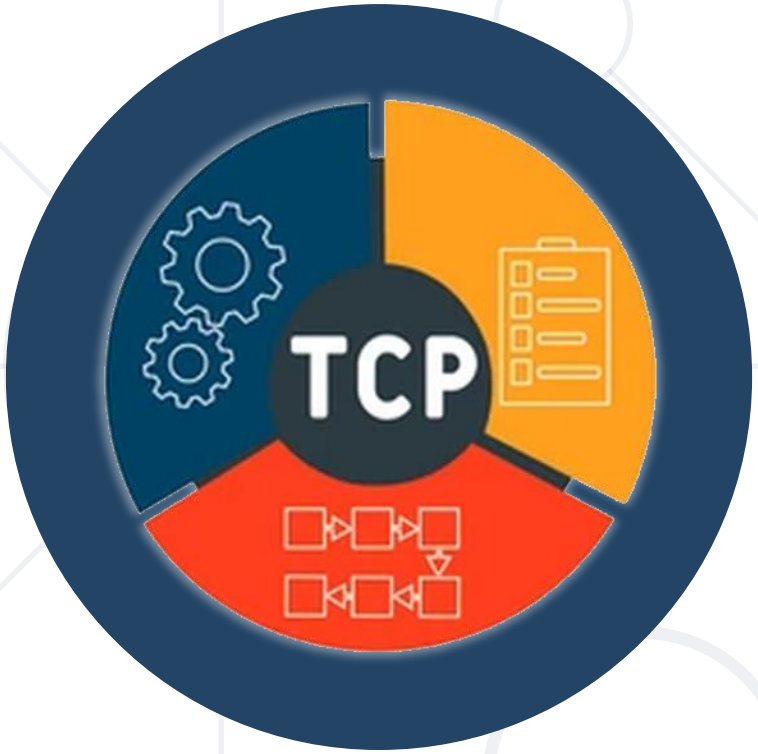
- **Session Layer (Layer 5)** – sessions
  - **Functions**: dialog control, token management, synchronization
  - **Protocols**: Secure Sockets Layer (**SSL**), Transport Layer Security (**TLS**), Remote Procedure Call (**RPC**), Session Initiation Protocol (**SIP**), Network File System (**NFS**)

- **Presentation Layer (Layer 6)** – data formats
  - **Functions**: data representation, encryption, decryption, compression, decompression
  - **Standards**: **ASCII**, **UTF-8**, **JPEG**, **MPEG**

OSI Model Layers

Software University

- **Application Layer (Layer 7)** – applications
  - **Networking for applications**, e. g. Web browsers use DNS, HTTP and HTTPS to open a Web site
  - Layer 7 **protocols**
    - Hypertext Transfer Protocol (**HTTP**) and **HTTPS** (secure HTTP over SSL)
    - File Transfer Protocol (**FTP**) – transfer files
    - Simple Mail Transfer Protocol (**SMTP**) and **IMAP** (mailbox access)
    - Domain Name System (**DNS**) – host to IP address
    - Telnet and Secure Shell (**SSH**) – session to a remote host

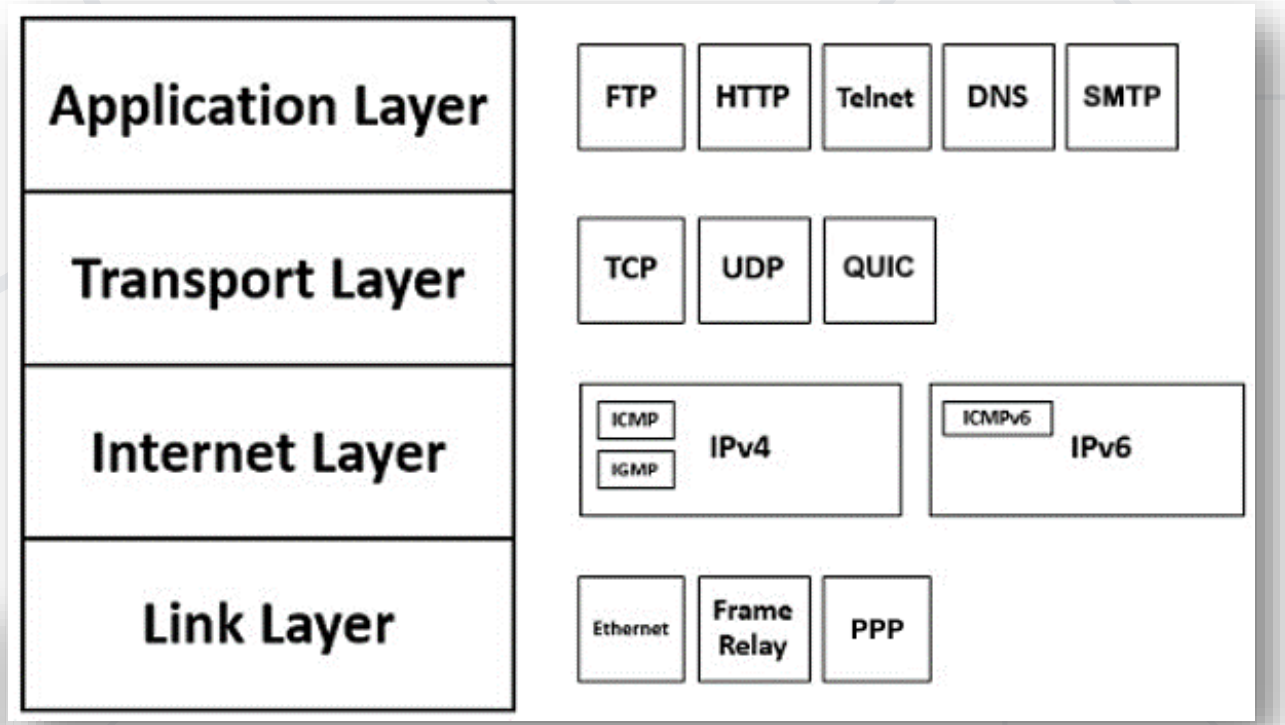


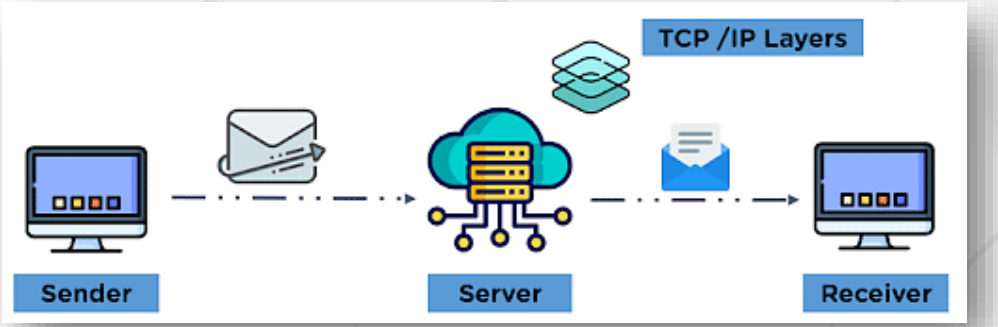
TCP/IP Model

The 4 Layers in the TCP/IP Protocol Suite

TCP/IP Protocol Suite

- **TCP/IP Protocol Suite** (TCP/IP Model) == **Transmission Control Protocol** / **Internet Protocol**
  - **Simplified version of OSI**, with only 4 layers
  - Easier for developers, QAs and IT professionals





TCP/IP Layers

- **Link Layer**
  - Combines the **functionalities of OSI Physical and Data Link layers**
  - Transmission and reception of data packets over a physical medium
  - Management of data link connections
- **Internet Layer**
  - **Corresponds to the OSI Network Layer**
  - Handling the logical addressing and routing of data packets

TCP/IP Layers

- **Transport Layer**
  - Closely **resembles the OSI Transport Layer**
- **Application Layer**
  - **Merges** the functionalities of **OSI Session, Presentation, and Application layers**

MAC, IP, Netmask, Gateway

Physical (MAC) Address, Network (IP) Address, Subnet Mask, Network Address and Gateway

Media Access Control (MAC) Address

- **MAC address** is a unique **hardware identifier** assigned to network interface cards (NICs)
  - **Format**: 48-bit (6 hex numbers), e. g. **9c-93-4e-3f-14-f7**
  - **Ethernet**, **WiFi** and **Bluetooth** devices have **MAC address**
- **Generally hardcoded** by the manufacturer
  - Decode a MAC address: https://dnschecker.org/mac-lookup.php
- Can be manually **changed** (depends)

Result for: 9C-93-4E-3F-14-F7

Address Prefix	9C:93:4E
Vendor / Company	Xerox Corporation
Start Address	9C934E000000
End Address	9C934EFFFFFF
Company Address	Mail Stop 0214 - 7e Webster Ny 14580 Us

Internet Protocol (IP) Address + Netmask

- **IP address** == 32-bit identifier (e. g. **192.168.0.61**) assigned to devices in a network for **addressing** and **routing** purposes
- **Netmask** (e. g. **255.255.255.0**) is a 32-bit number, used to masks out the **network part** of an IP address (IP bitwise AND mask == network address)
- **Network address + mask** (e. g. **192.168.0.0/24**) identifies the network
- **Gateway** (e. g. **192.168.0.1**) is the **router IP** used to access Internet
- **IPv6 address** == 128-bit address for the modern Internet (e. g. **2606:4700:0000:0000:0000:0000:6810:85e5**)
  - Not massively used, needs additional router configuration

Internet Protocol (IP) Address + Netmask

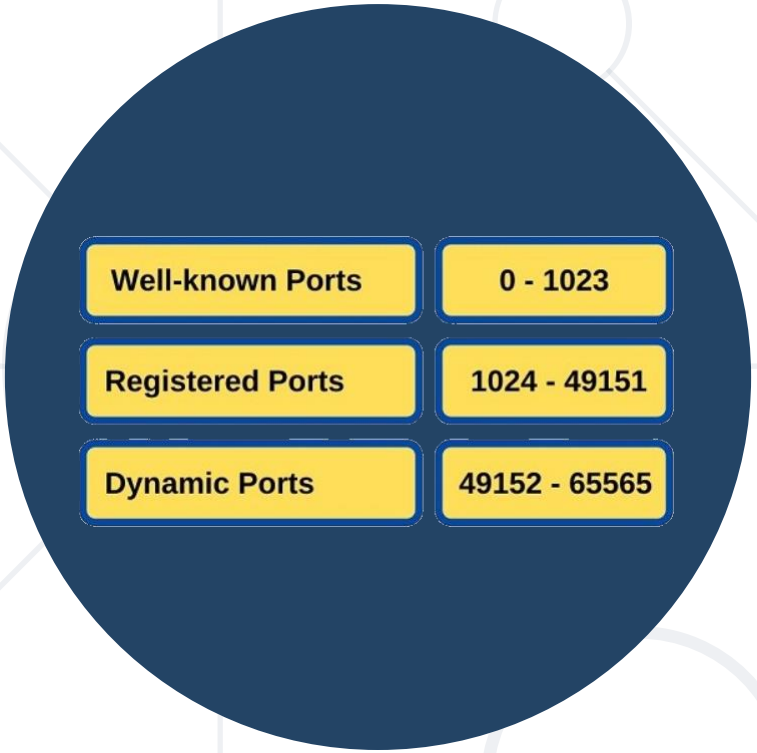
- **IP + netmask + gateway + DNS** are assigned:
  - **Statically** (manually by hand)
  - **Dynamically** (by the router using the **DHCP protocol**)

Wi-Fi properties

IP assignment:	Manual
IPv4 address:	192.168.0.144
IPv4 mask:	255.255.255.0
IPv4 gateway:	192.168.0.1

Wi-Fi properties

IP assignment:	Automatic (DHCP)
DNS server assignment:	Automatic (DHCP)


Well-known Ports
0 - 1023
Registered Ports
1024 - 49151
Dynamic Ports
49152 - 65565

Ports

Ports Overview

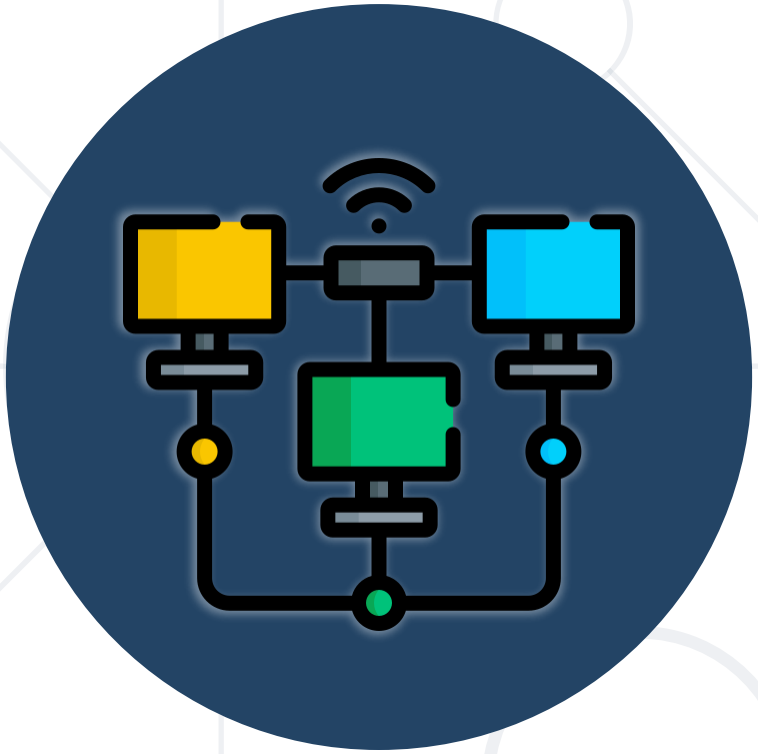
- **Numerical identifiers** used to distinguish specific processes or services running on a device within a network
- Facilitate end-to-end communication between applications on different devices
- **Types of Ports**
  - **TCP ports -** Used for connection-oriented communication, ensuring reliability and data integrity
  - **UDP ports -** Used for connectionless communication, providing faster data transmission with minimal overhead

Port Numbers

- Used to **identify** a network service
- Network services registry in **`/etc/services`**
- Some of them are:
  - **22** – SSH, **53** – DNS, **80** – HTTP, **110** – POP3, **123** – NTP , **143** – IMAP

```
tcpmux          1/tcp
echo            7/tcp
echo            7/udp
discard         9/tcp           sink null
discard         9/udp           sink null
systat          11/tcp          users
```

Ports	Port Numbers
Well-known (or system) ports	**0 – 1023**
Registered (or user) ports	**1024 – 49151**
Dynamic (and / or private) ports	**49152 – 65535**

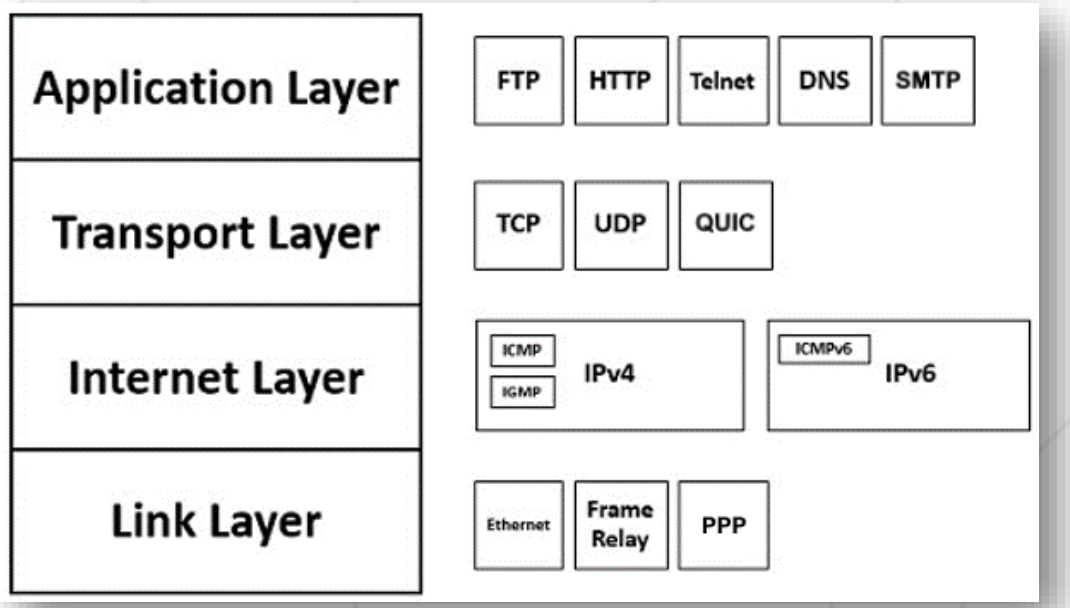


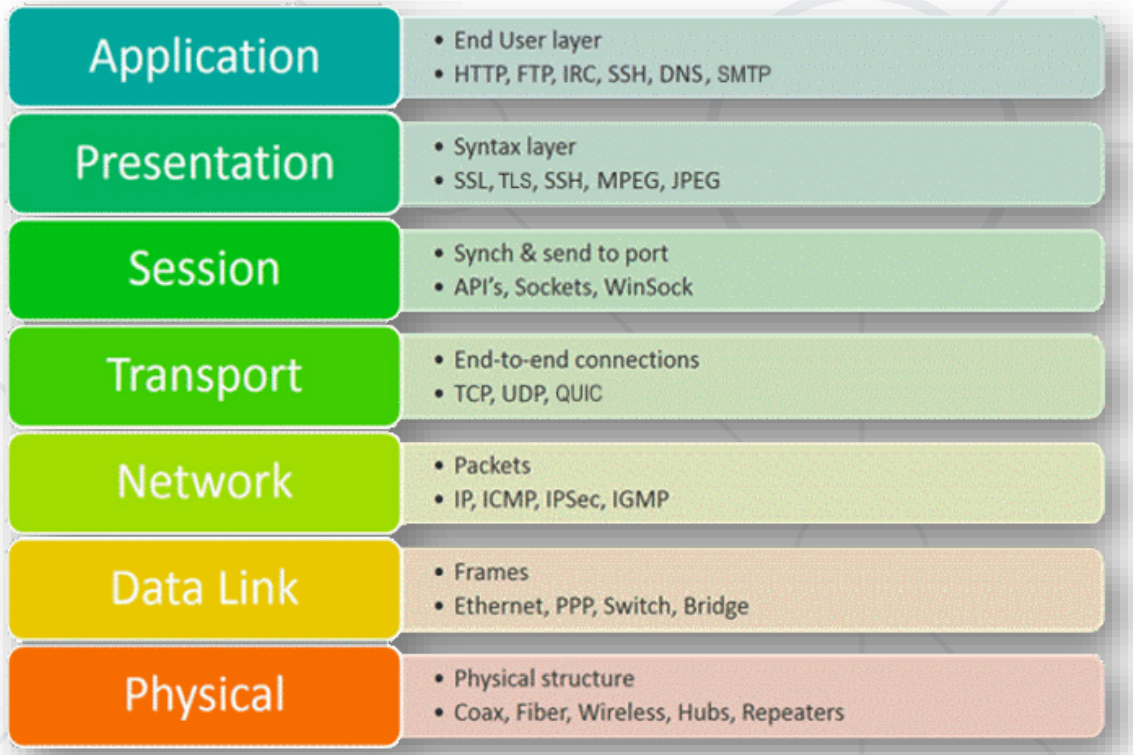
Networking: Summary

OSI Model, TCP/IP, Network Protocols

Networking and Internet Protocols Summary

- Communication in Internet uses **networking protocols**
  - The **OSI model** defines 7 layers of networking protocols
  - The **TCP/IP** model – 4 layers





Key Network Protocols

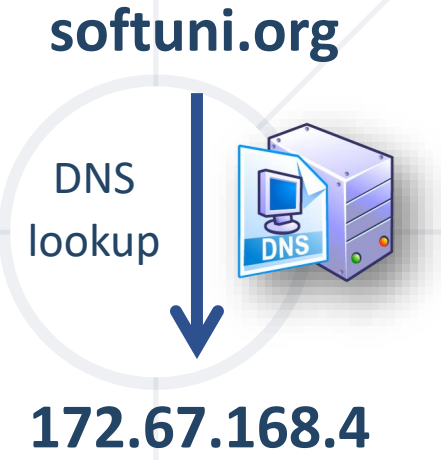
- **IP**: host-to-host communication in local networks and Internet
  - Uses **IP** address + **netmask** + **gateway** + **DNS**
- **TCP**: implements reliable transport of **data streams**; uses **ports** to distinguish connections
- **UDP**: transports **single packets**, connectionless, faster, has no error checking; uses **ports** to distinguish connections
- **QUIC**: modern fast transport for **multi-streams**, based on UDP + TLS; uses **ports** to distinguish connections
- **ICMP**: diagnostics protocol, used by **ping** and **traceroute**
- **DNS**: maps **hosts to IP** addresses (e. g. softuni.org → 172.67.168.4)
- **HTTP**: **request-response** text-based protocol for the **Web**

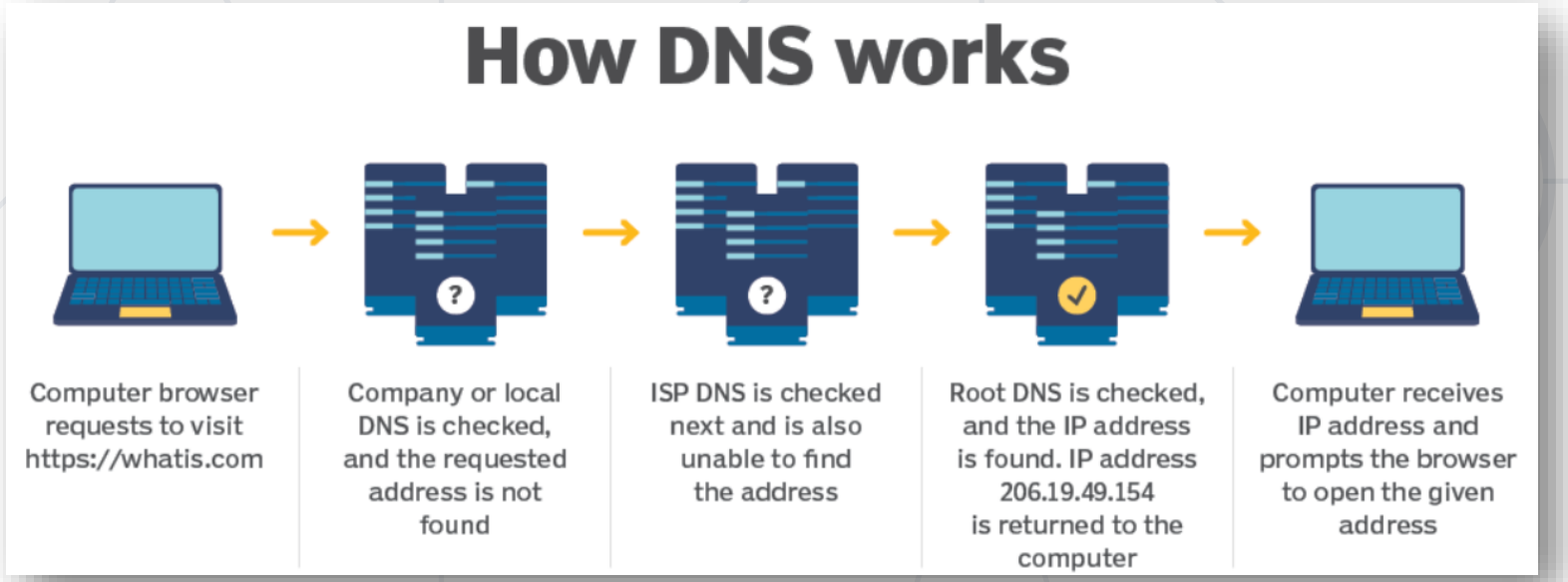
Web Fundamentals

WWW, Domains, DNS, URL

Domain Name System (DNS)

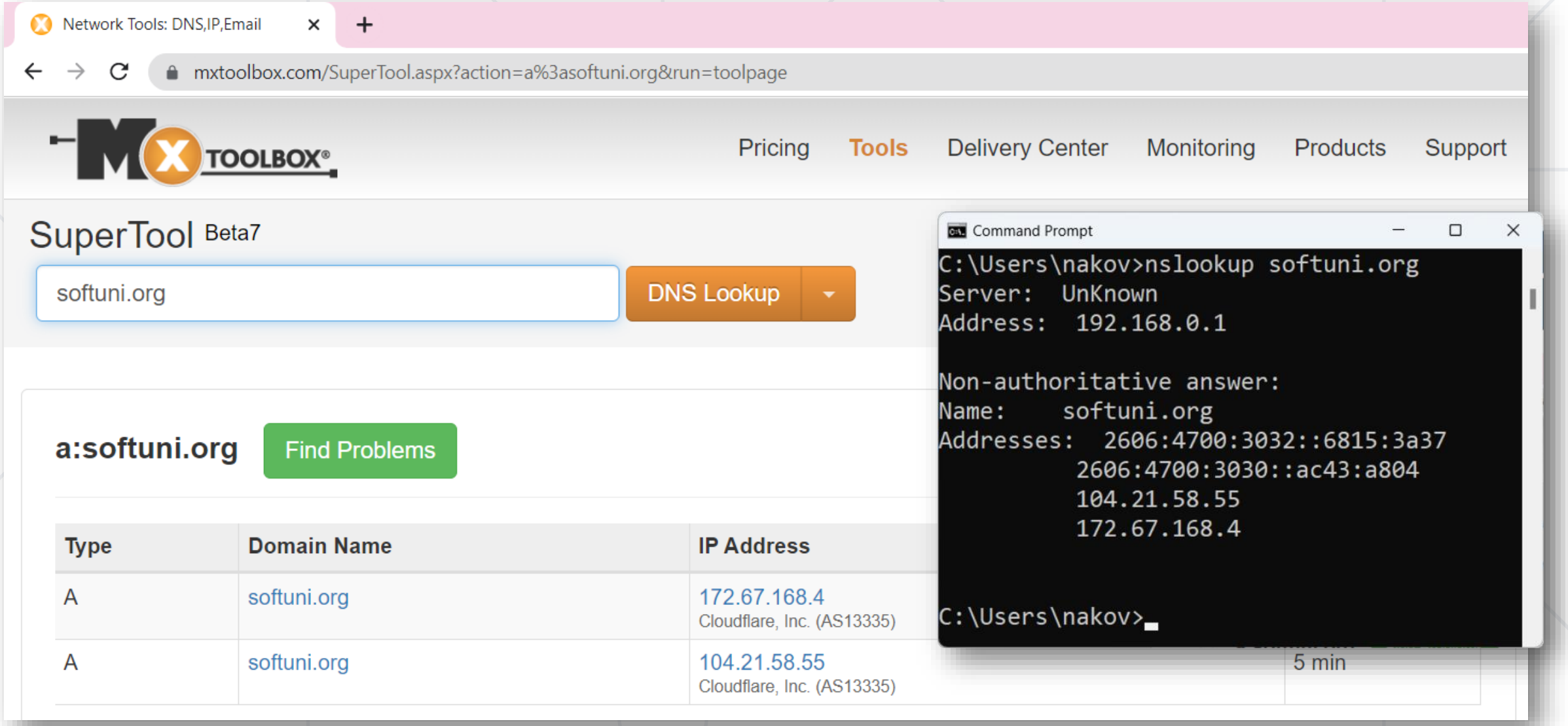
- A hierarchical, distributed system (part of Internet) that **translates domain names into IP addresses**
- Facilitates the resolution of human-readable **domain names** to machine-readable **IP addresses**





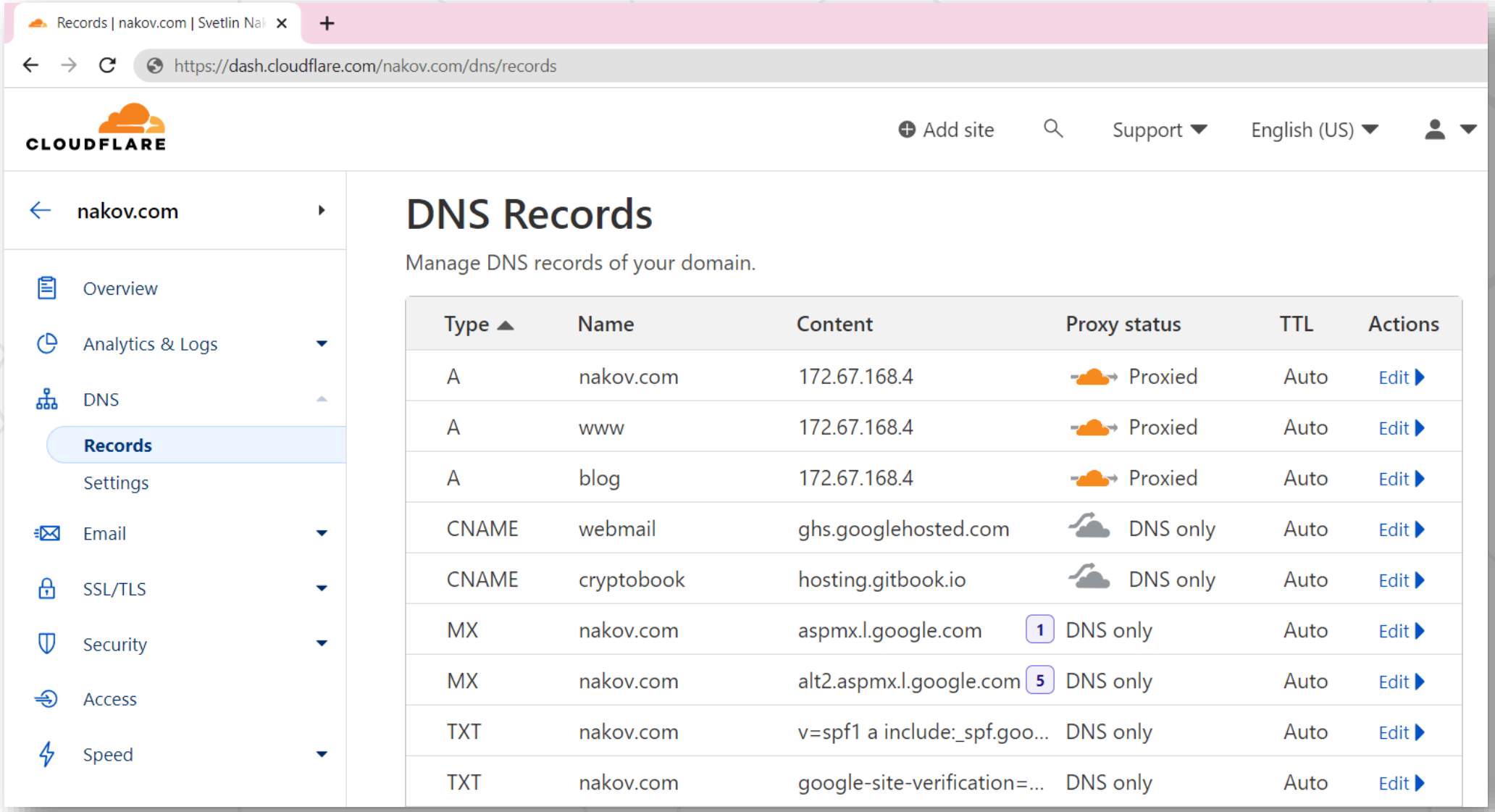
DNS Lookup – Example

Software University



DNS Configuration – Example

Software University



Domain Names

- **Domain name** == a **unique**, human-readable name for **Internet host** / machine / web site
  - Examples: softuni.org, www.cloudflare.com, students.softuni.bg
  - **Simplify navigation** to websites, easier to remember and share
- Domain structure
  - **Top-level domains (TLDs)** – .com, .net, .org, .info, .us, .uk, .de, .uk
  - **Second-level domains (SLDs)** – website's name, softuni.org
  - **Subdomains** – inner hosts, e. g. blog.nakov.com

What is a URL?

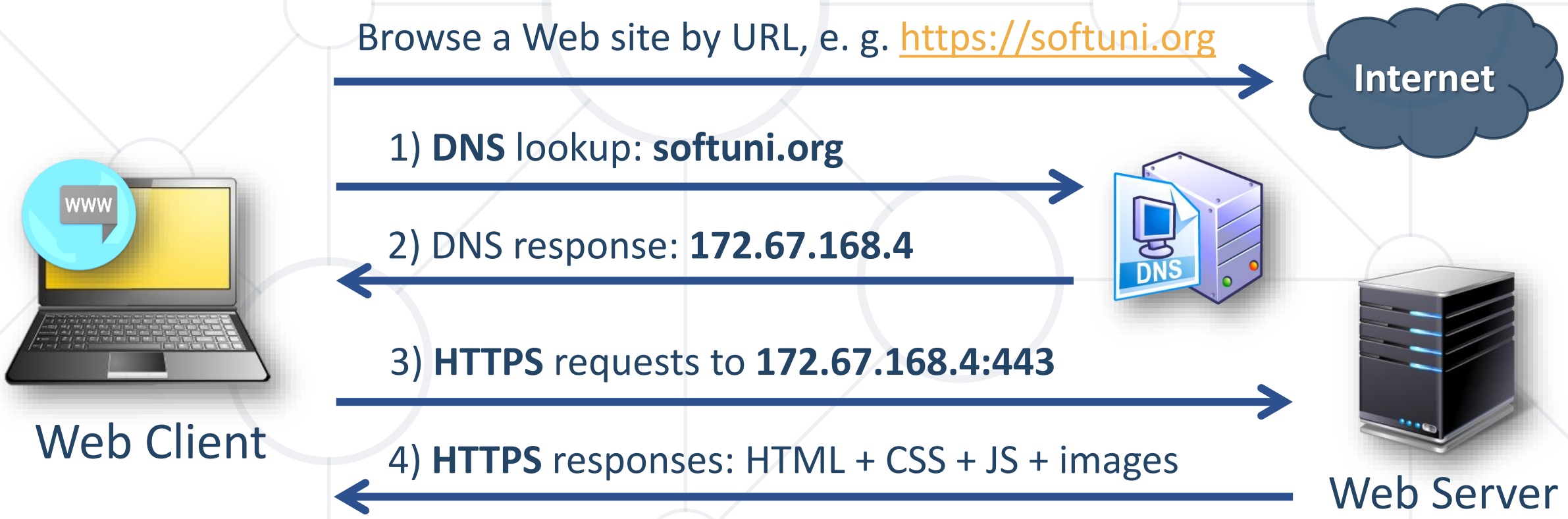
- A **URL** (Uniform Resource Locator) is a **unique address** pointing to a **website**, a **web page**, or a **document** on the Internet
  - Example: https://java-book.softuni.org/home?lang=en
- **Structure-wise**, a URL consists of multiple **elements**
  - Communication protocol, e. g. **https://**
  - Subdomain, e. g. **java-book**
  - Domain name, e. g. **softuni.org**
  - Path to the resource, e. g. **/home**
  - Parameters, e. g. **?lang=en**

Uniform Resource Locator (URL) Example

- **Network protocol** (`https`, `http`, `ftp`, …) – HTTPS in most cases
- **Domain**, **host**, or **IP** address (`softuni.org`, `mail.yahoo.com`, `127.0.0.1`, `[::1]`, `[2606:4700::6810:85e5]`, `webmail`)
- **Port** (the default HTTPS port is `443`) – integer [0…65535]
- **Path** (`/forum`, `/path/index.php`) – a script / page on the Web server
- **Query string** (`?id=27&lang=en`) – parameters in format **key=value**
- **Fragment** (`#slides`) – navigate to certain section in the page

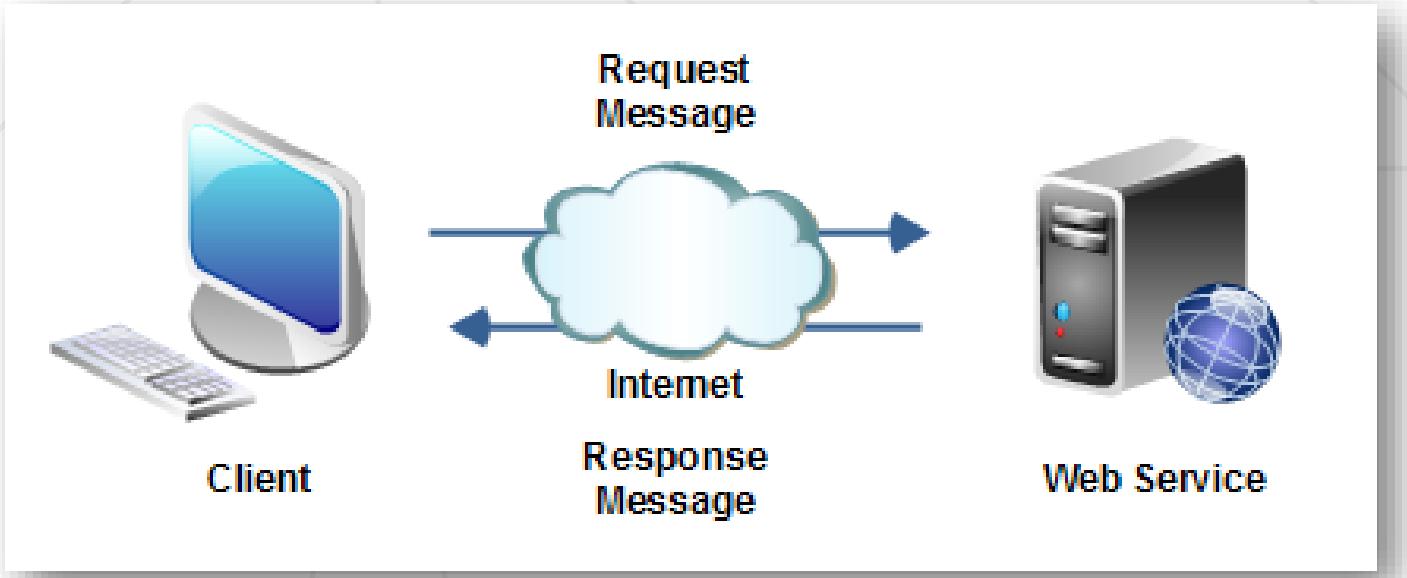
WWW (World Wide Web)

- A **global**, **interconnected system** of documents, images, and other resources, accessed through the Internet using **Web browsers**



What is a Web Service?

- **Web services** implement **communication** between software **systems** or **components** of over the **network**
  - Using standard **protocols**, such as HTTP, JSON and XML
  - Exchanging **messages**, holding data and operations



Web Service Call – Examples

Software University

api.zippopotam.us/us/90222


```
{
    "post code": "90222",
    "country": "United States",
    "country abbreviation": "US",
    "places": [
        {
            "place name": "Compton",
            "longitude": "-118.2357",
            "state": "California",
            "state abbreviation": "CA",
            "latitude": "33.9099"
        }
    ]
}
```


api.github.com/users

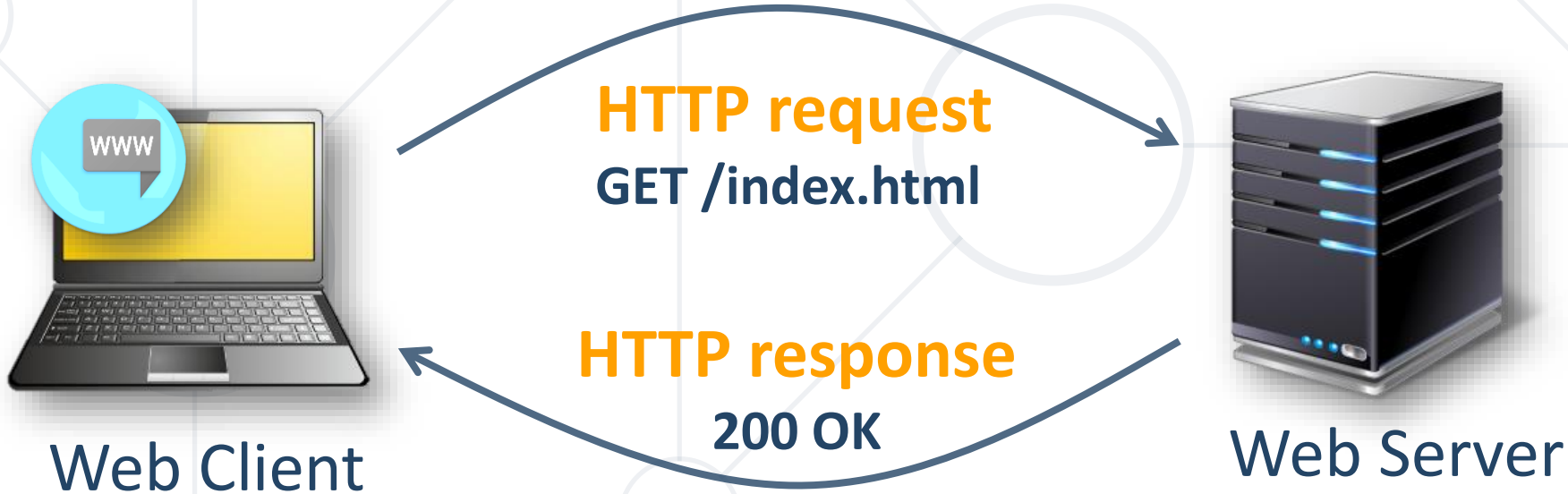

```
[
  {
    "login": "mojombo",
    "id": 1,
    "node_id": "MDQ6VXNlcjE=",
    "avatar_url": "https://avatars.githubusercontent.com/u/1?v=4",
    "gravatar_id": "",
    "url": "https://api.github.com/users/mojombo",
    "html_url": "https://github.com/mojombo",
    "followers_url": "https://api.github.com/users/mojombo/followers",
    "following_url": "https://api.github.com/users/mojombo/following{/other_user}",
    "gists_url": "https://api.github.com/users/mojombo/gists{/gist_id}",
    "starred_url": "https://api.github.com/users/mojombo/starred{/owner}{/repo}",
    "subscriptions_url": "https://api.github.com/users/mojombo/subscriptions",
    "organizations_url": "https://api.github.com/users/mojombo/orgs",
    "repos_url": "https://api.github.com/users/mojombo/repos",
    "events_url": "https://api.github.com/users/mojombo/events{/privacy}",
    "received_events_url": "https://api.github.com/users/mojombo/received_events",
    "type": "User",
    "site_admin": false
  },
  { … }, // 18 items
  { … }, // 18 items
  { … }, // 18 items
  { … }, // 18 items
```

HTTP Protocol – Basics

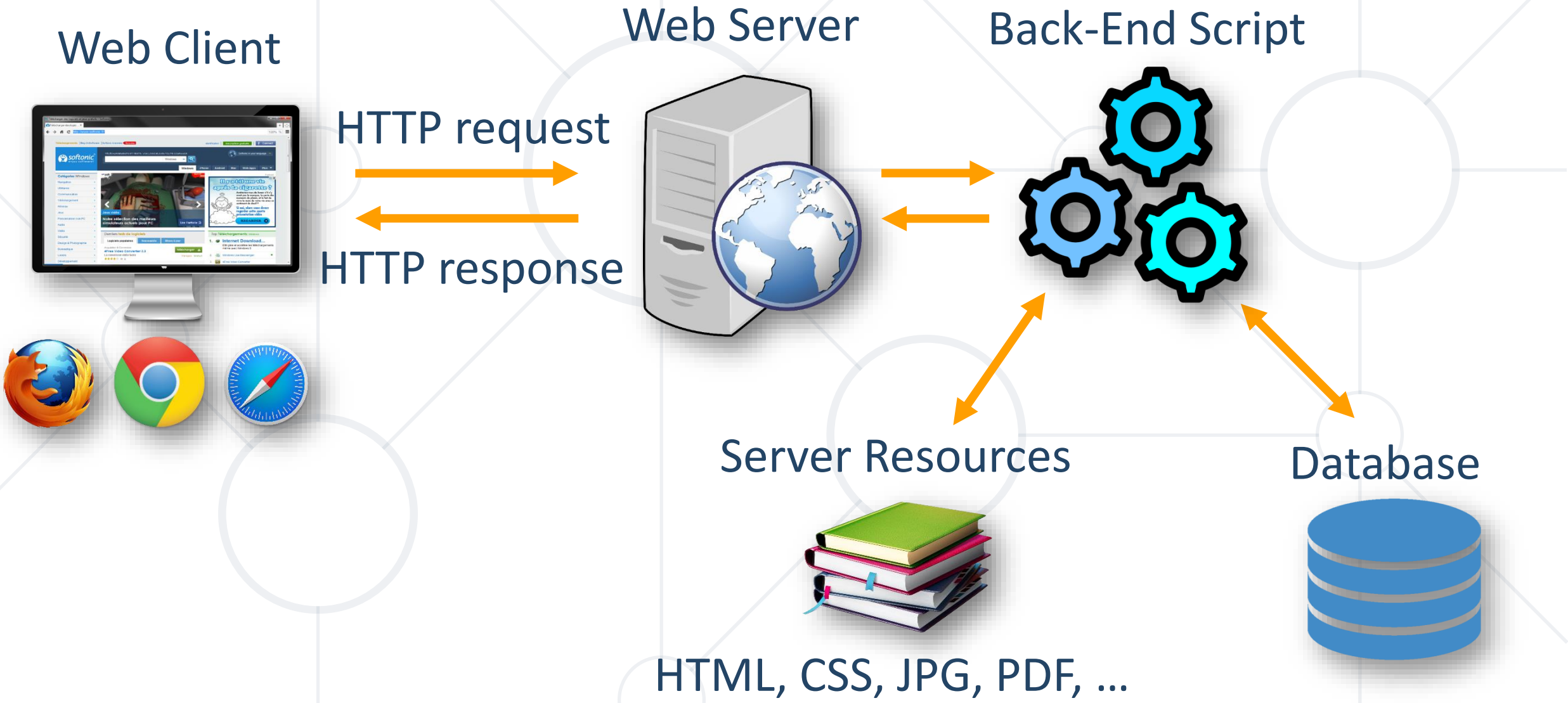
Request-Response Text-Based Protocol for the Web

HTTP Basics

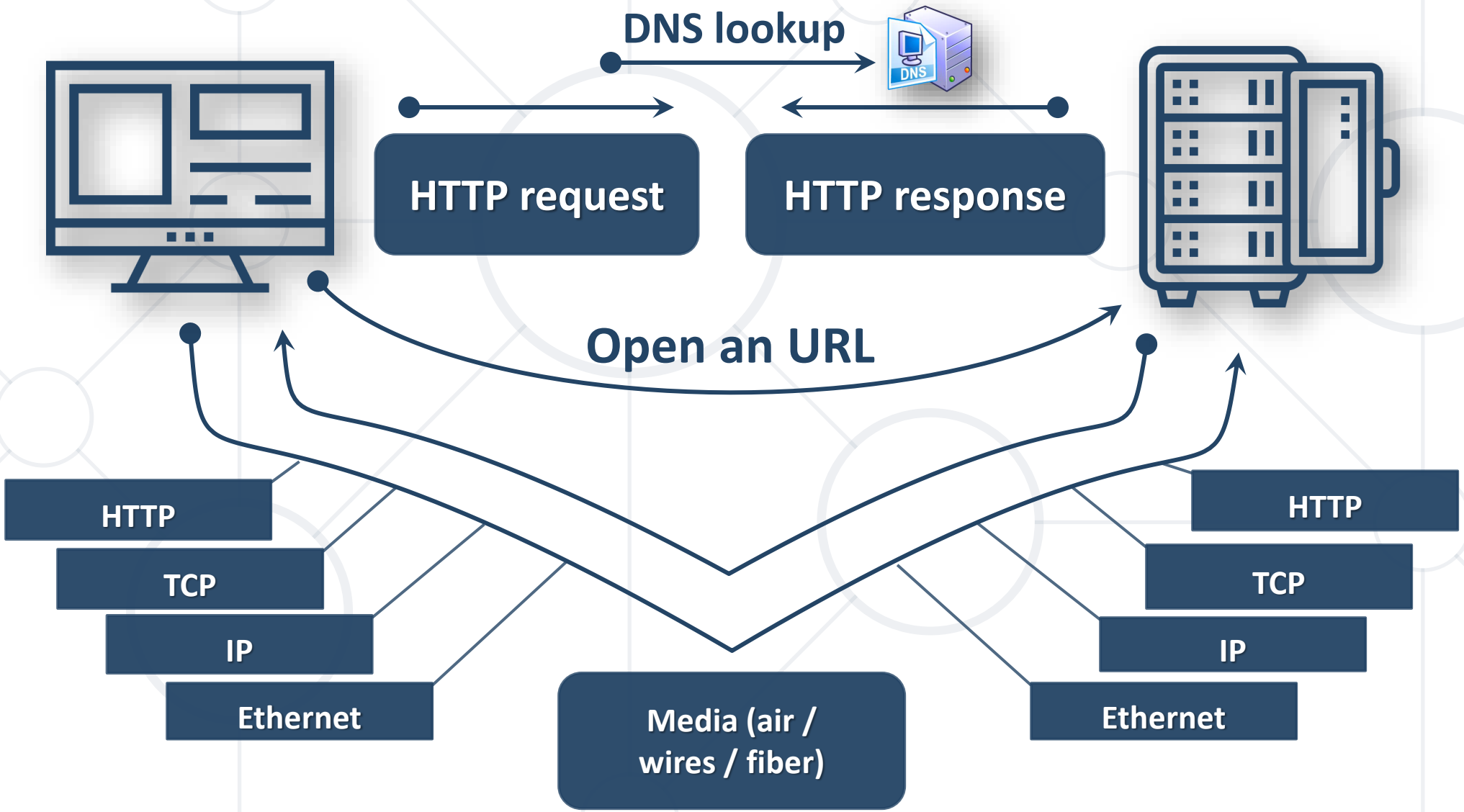
- **HTTP** (**H**yper**T**ext **T**ransfer **P**rotocol)
  - **Text-based** client-server protocol for the Internet
  - For transferring **Web resources** (HTML files, images, styles, etc.)
  - **Request-response** based



Web Server: How It Works?



Network Layers and HTTP



HTTP Request Methods

- **HTTP request methods** specify the desired **action** to be performed on the requested resource (identified by URL)

Method	Description
GET 	Retrieve a resource
POST 	Create / store a resource
PUT 	Update (replace) a resource
DELETE 	Delete (remove) a resource
PATCH 	Update resource partially (modify)
HEAD 	Retrieve the resource's headers

CRUD == the four main functions of persistent storage

Other Methods
CONNECT
OPTIONS
TRACE

HTTP Response Status Codes

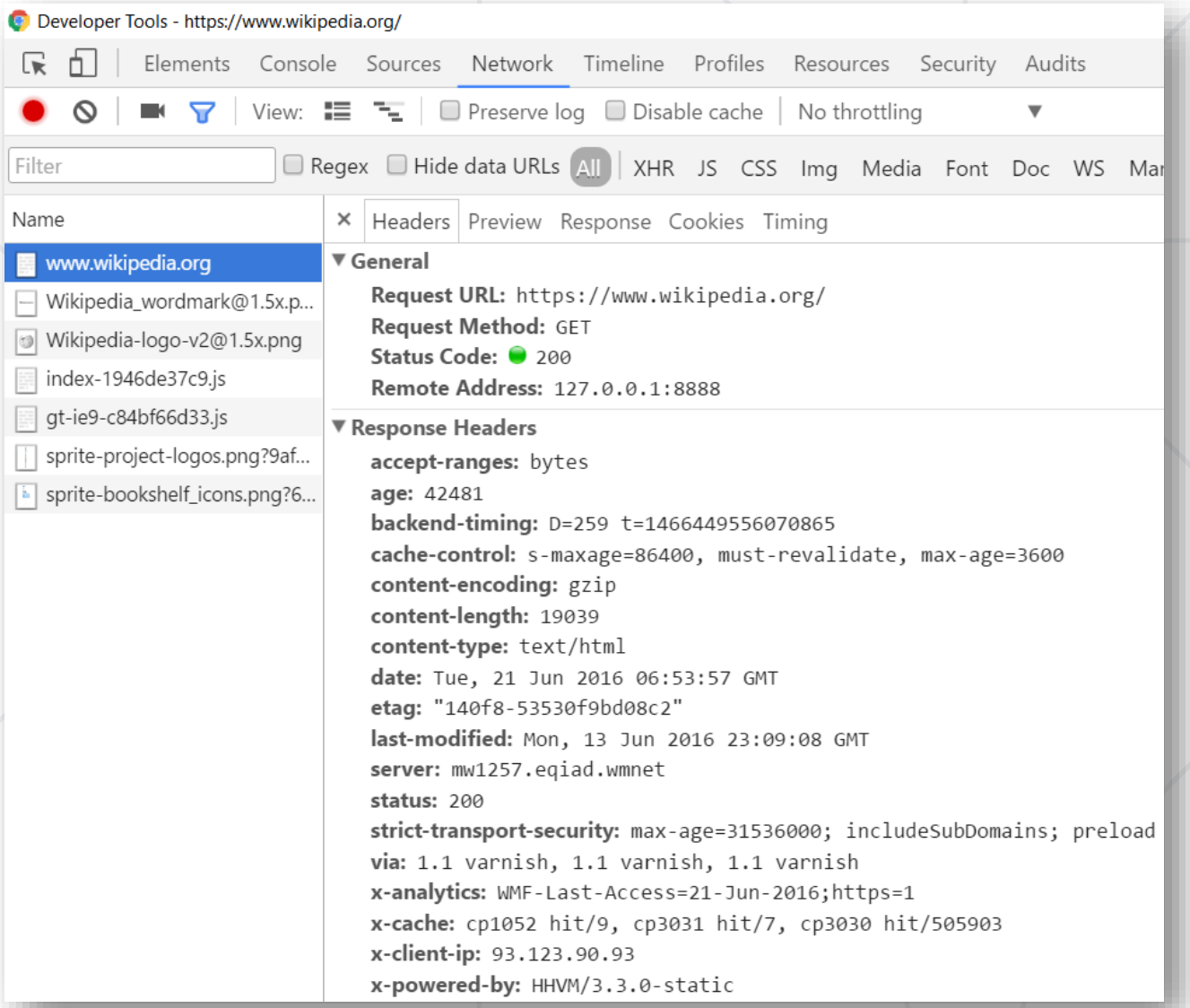
Software University

Status Code	Action	Description	
200	OK	Successfully retrieved resource	Success
201	Created	A new resource was created	
204	No Content	Request has nothing to return	
301 / 302	Moved	Moved to another location (redirect)	Redirect
400	Bad Request	Invalid request / syntax error	Error
401 / 403	Unauthorized	Authentication failed / access denied	
404	Not Found	Invalid resource requested	
409	Conflict	Conflict detected, e.g. duplicated email	
500 / 503	Server Error	Internal server error / service unavailable	

HTTP Dev Tools

In-Browser Tools for Developers and QAs

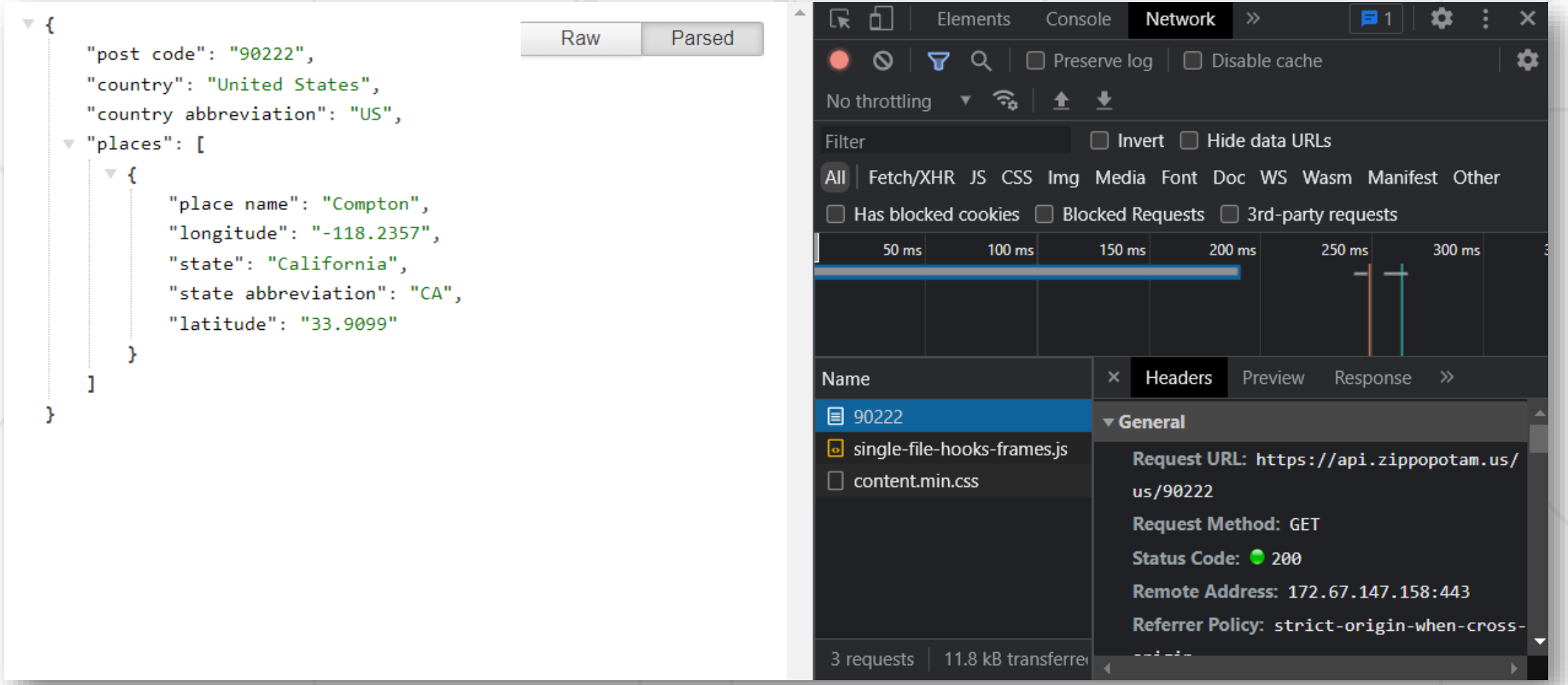
HTTP Developer Tools: Network Inspector



- Chrome Developer Tools
  - Press **[F12]** in Chrome
  - Open the [Network] tab
  - Inspect the HTTP traffic

HTTP Requests and DevTools – Example

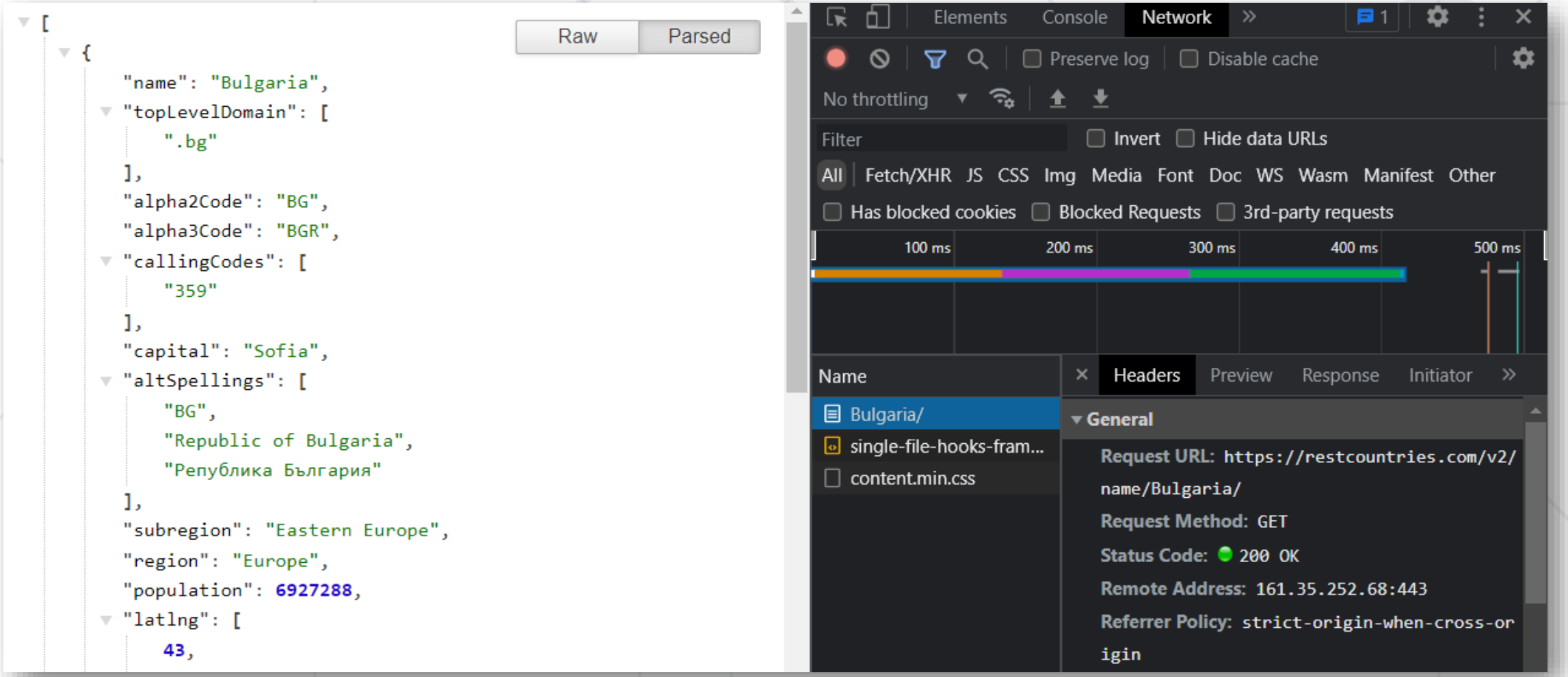
- https://api.zippopotam.us/us/90222



HTTP Requests and DevTools – Example

Software University

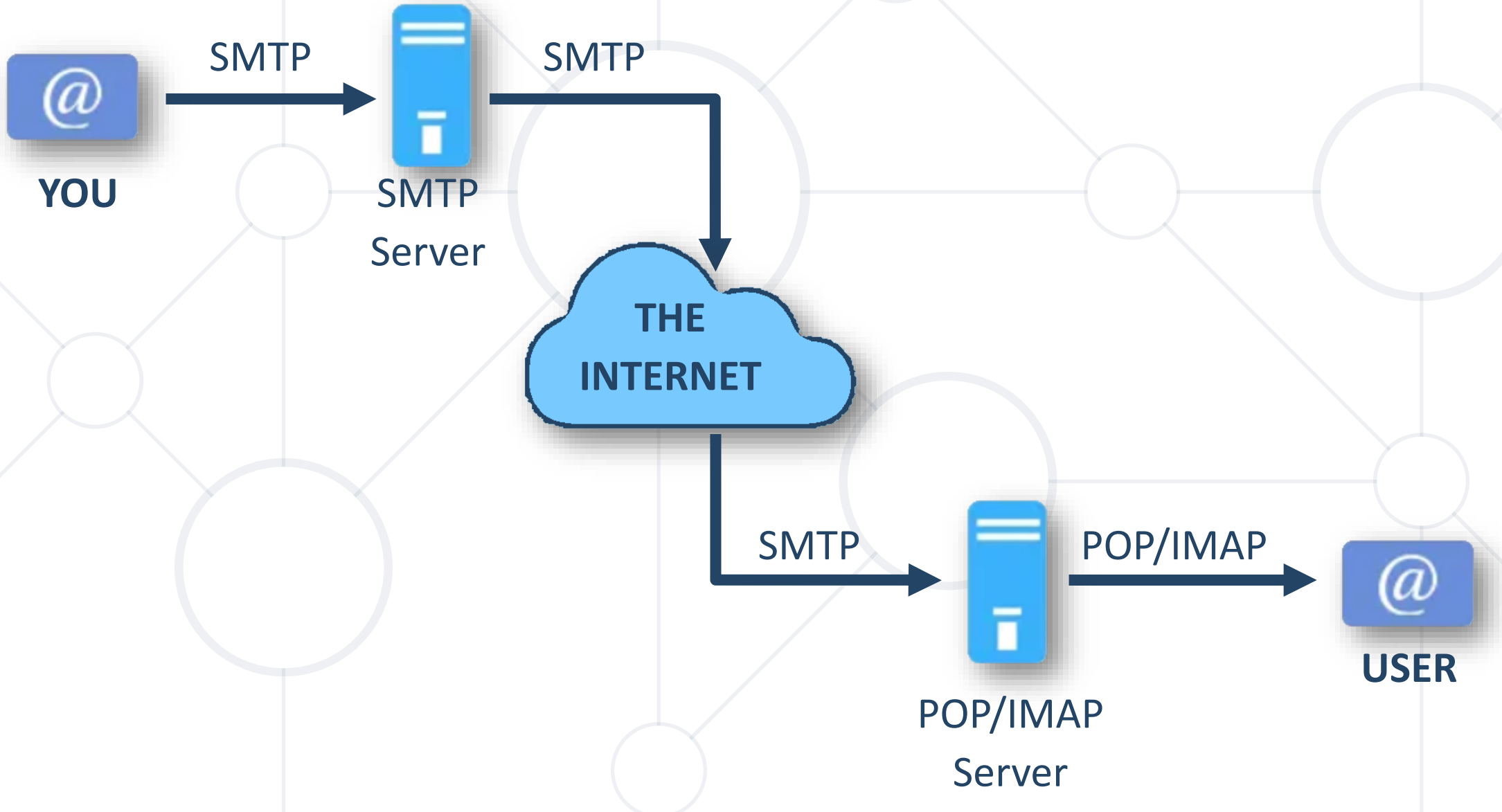
- https://restcountries.com/v2/name/Bulgaria



Email

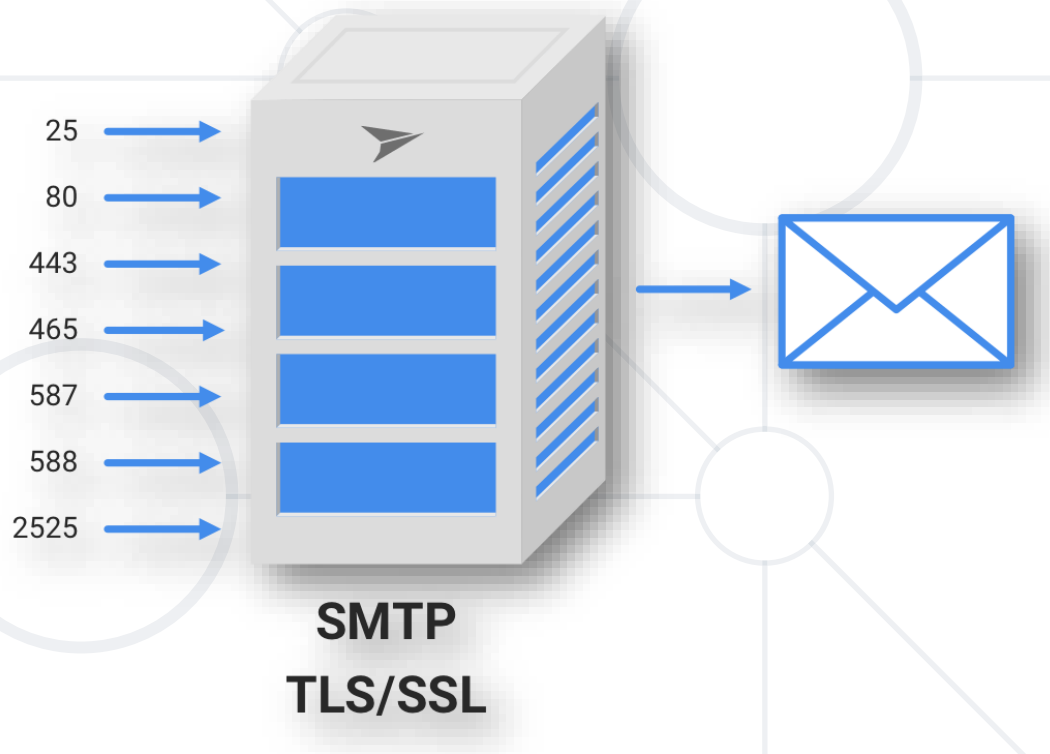
Email Protocols: SMTP and IMAP

How Does Email Work?



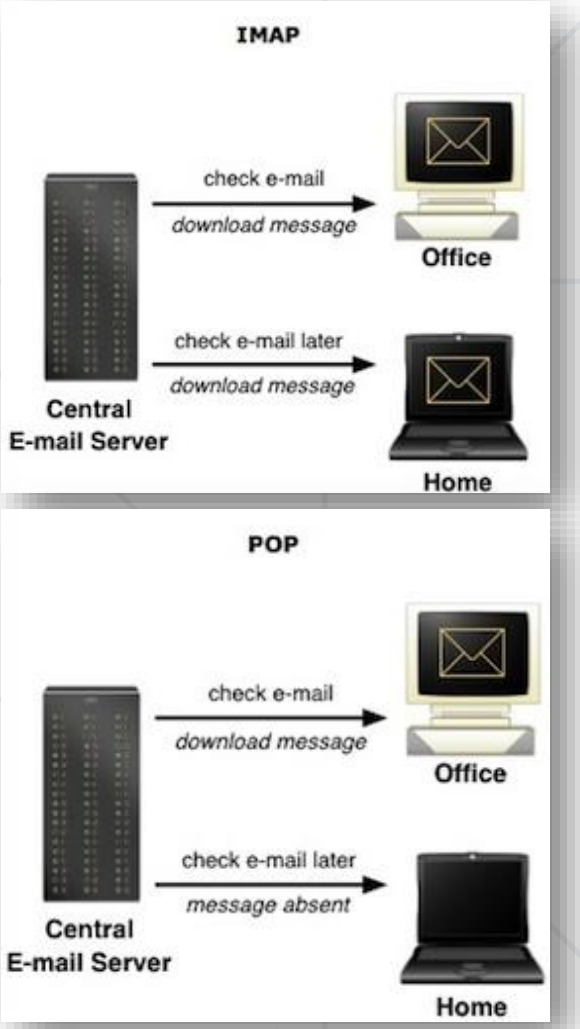
Sending Email: the SMTP Protocol

- **SMTP** (Simple Mail Transfer Protocol)
  - Send / receive email messages between mail servers
- Mail client apps (like Thunderbird) use SMTP to send emails
- **SMTPS** (secure SMTP) uses additional SSL for security
  - **SMTP** – port **25**, **SMTPS** – port **587**

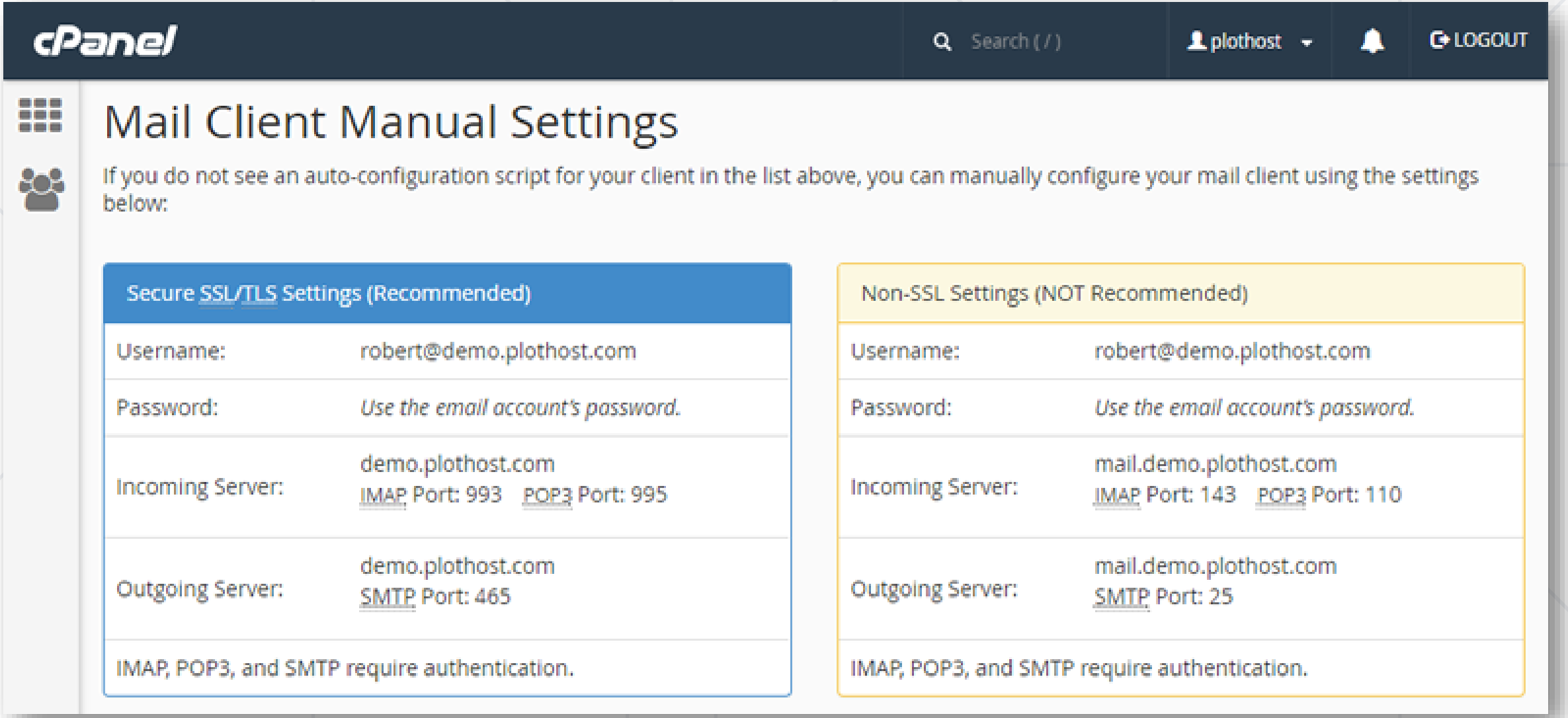


IMAP/POP

- **IMAP (Internet Message Access Protocol)**
  - Retrieve email messages from server mailbox
  - Allows management of email messages on the server from different devices (**sync and delete**)
  - More popular and flexible
- **POP (Post Office Protocol)**
  - Once downloaded to a client, the message is removed from the server (**download and delete**)
  - Difficult to access email messages from different devices or locations



Configuring an Email Client

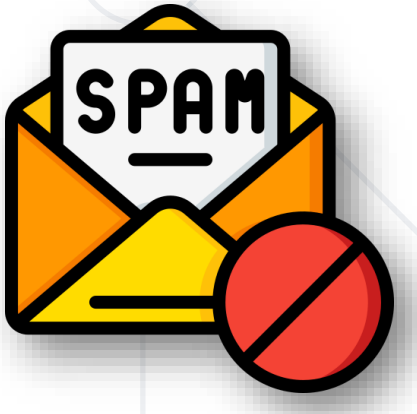


Email Forwarding

- **Redirect** incoming emails to another email address
  - E. g. peter@softuni.org → peter1997@gmail.com
- Useful for managing multiple email accounts
  - **Server-based forwarding** –the mail server automatically forward incoming messages to another email address
  - **Client-based forwarding** – setting up email forwarding using the email client settings
  - **Email filters** – setting up filters to forward messages that match specific criteria

Spam Filters

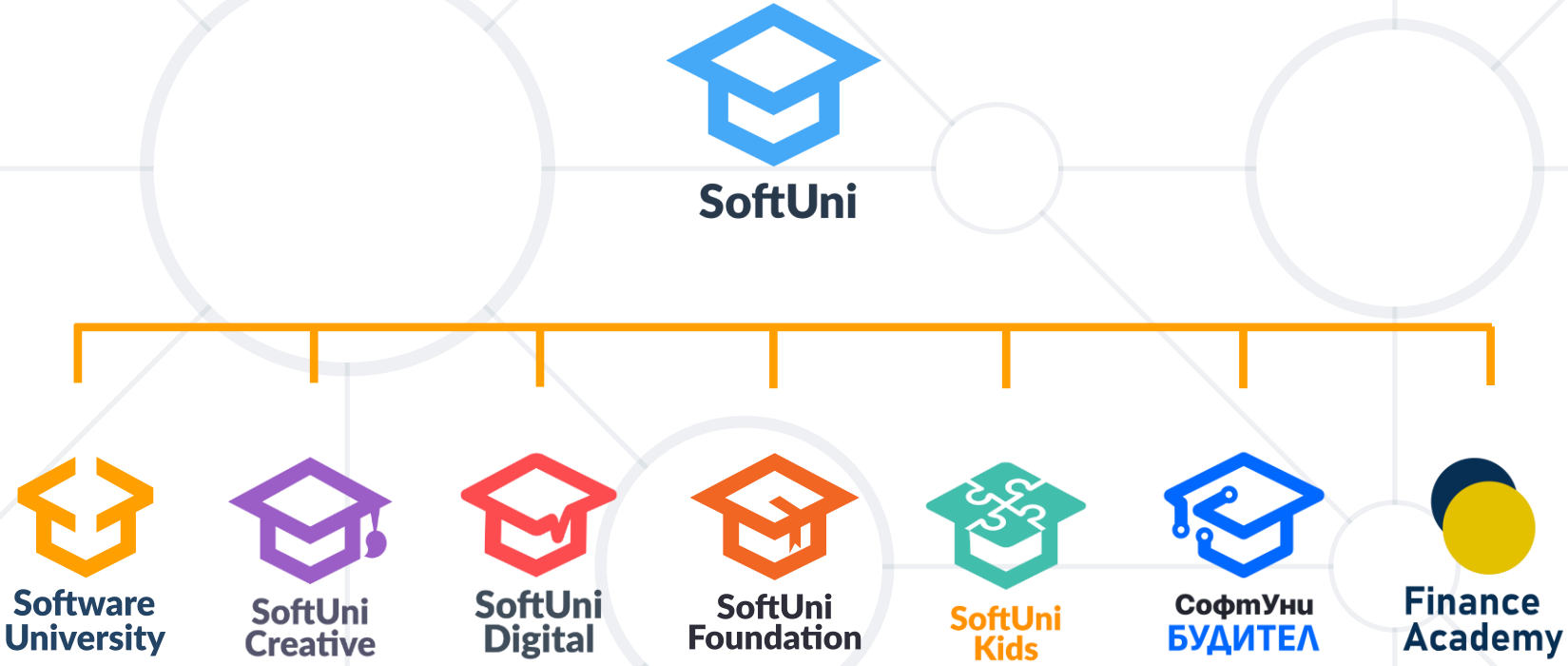
- **Detect** and **filter** out unwanted or harmful email messages
  - Typically, move spam emails to the "SPAM" folder
- **Rule-based** filtering and **Machine learning-based** filtering
- Some filtering is usually conducted automatically by an SMTP
- **Reject**, **redirect**, or **quarantine** based on the email content
- **Customizable** for individual needs and preferences
- Setting up rules to **block/allow** emails from specific senders or domains



Summary

- Networking protocols: rules for communication
- Network layer models: OSI Model, TCP/IP
- MAC address, IP address, netmask, ports
- Domains and DNS, WWW
- HTTP request (GET, POST) + HTTP response
- Browser Dev Tools: in-browser debugging
- How does an email work? SMTP / IMAP

Questions?



SoftUni Diamond Partners

Software University

Trainings @ Software University (SoftUni)

- Software University – High-Quality Education, Profession and Job for Software Developers
  - softuni.bg, about.softuni.bg
- Software University Foundation
  - softuni.foundation
- Software University @ Facebook
  - facebook.com/SoftwareUniversity

License

- This course (slides, examples, demos, exercises, homework, documents, videos and other assets) is **copyrighted content**
- Unauthorized copy, reproduction or use is illegal
- © SoftUni – https://about.softuni.bg/
- © Software University – https://softuni.bg