

# AN INTRODUCTION TO NETCENTRIC DEVELOPMENT

**Protocols, Domains, Routing, and Tools**

COS216  
AVINASH SINGH  
DEPARTMENT OF COMPUTER SCIENCE  
UNIVERSITY OF PRETORIA

# VIDEO

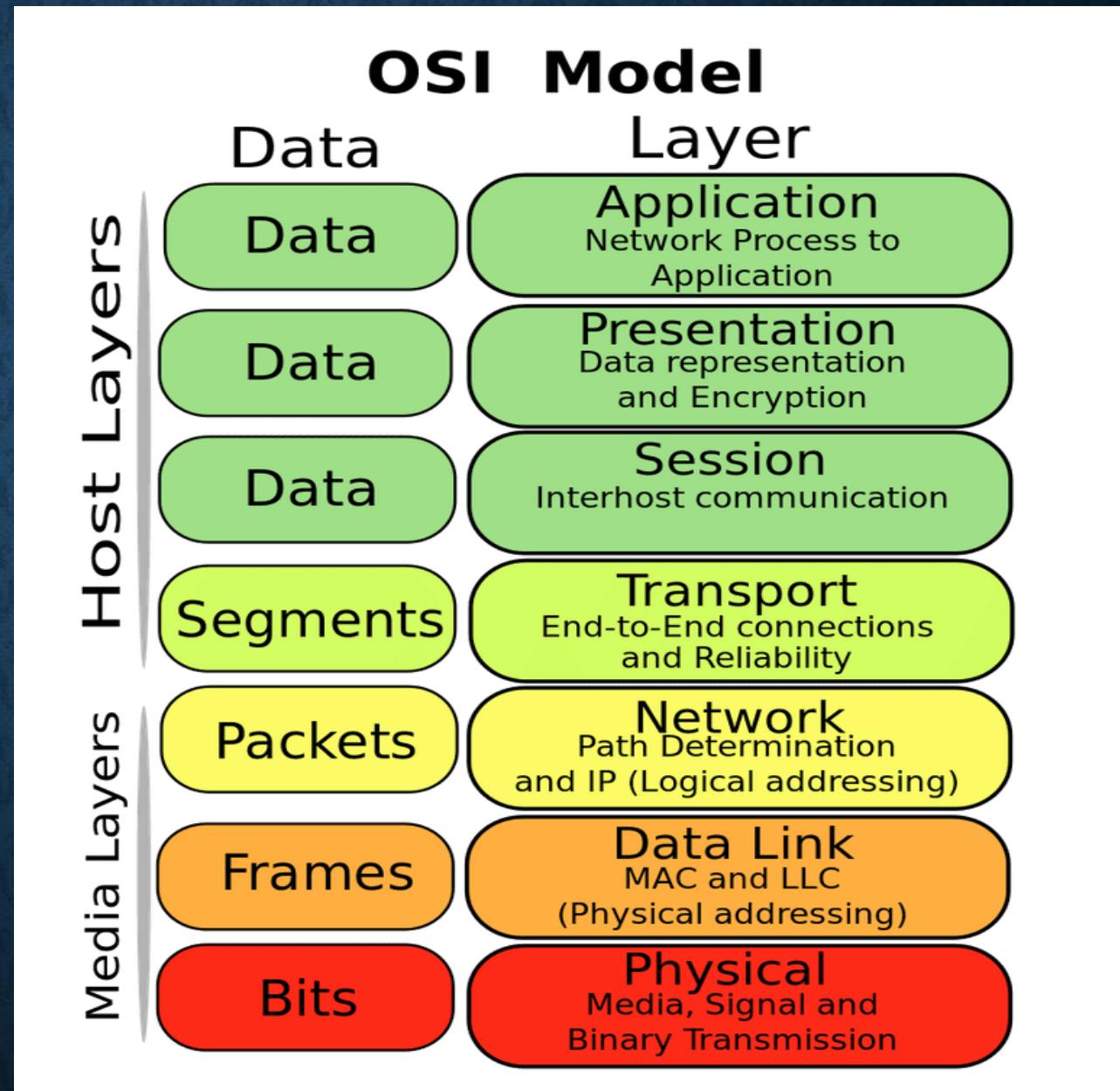
<https://www.youtube.com/watch?v=PG9oKZdFb7w>



# OVERVIEW OF PROTOCOLS

- A set of rules/procedures regulating the communication between computers
- A protocol specifies:
  - How and when communication starts
  - Which part of the communication comes at which time
  - How and when the communication ends
- Protocols are specified and maintained by:
  - Open consortiums
  - Industry organizations
  - Governments
- Different protocols exist on different layers

# OSI MODEL



# TRANSPORT PROTOCOLS

- Transmission Control Protocol (TCP)
  - Provides reliable delivery of a stream of octets
  - Between programs running on computers connected to a network
  - Provides ordering and error-checking of messages
  - WWW, email, file transfer, video streaming
- User Datagram Protocol (UDP)
  - Transport protocol like TCP
  - Provides connectionless datagram service
  - Emphasizes latency over reliability
  - Does not guarantee delivery, ordering, or protection
  - Torrents, DNS, VoIP, VPN, online games, video streaming

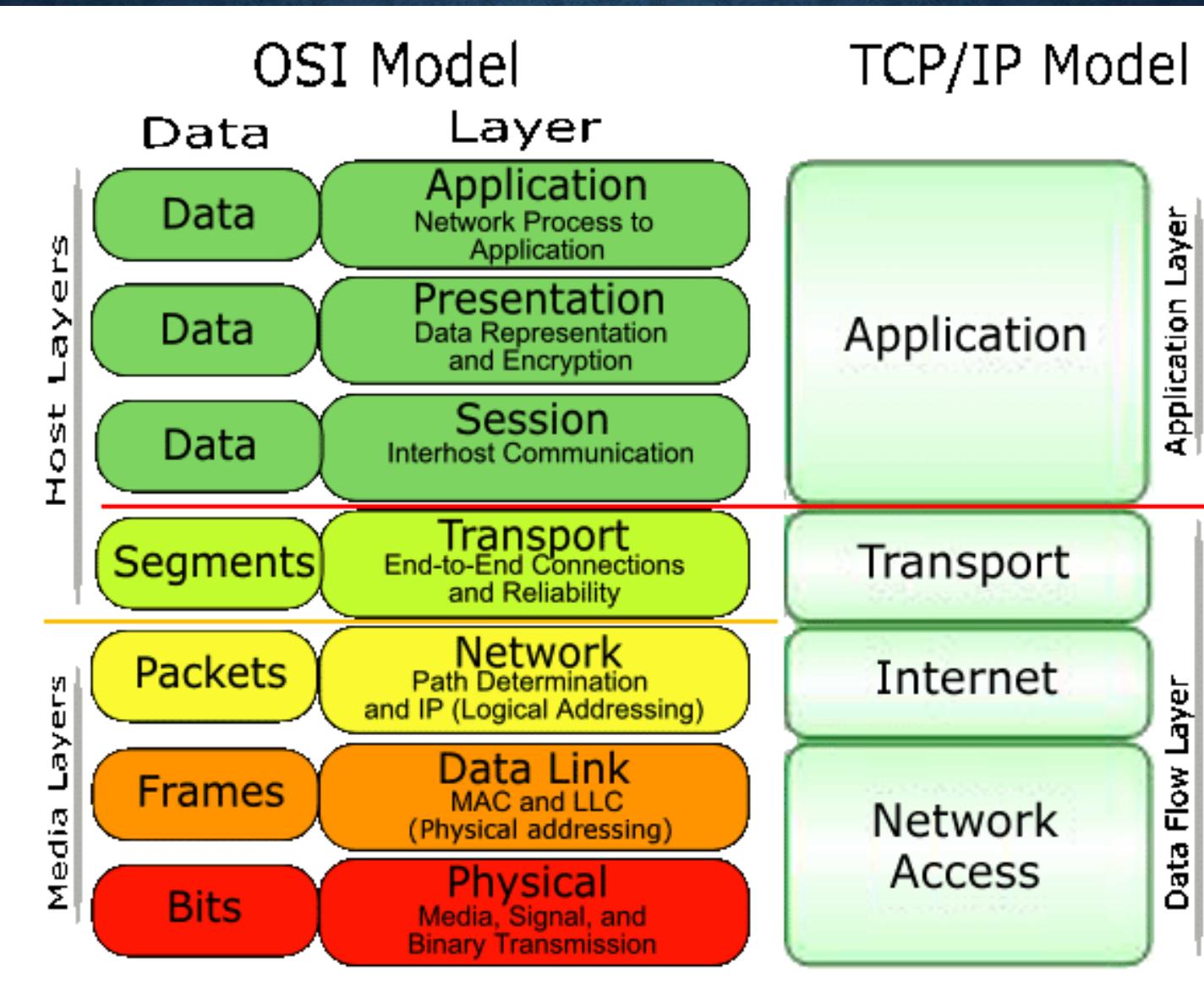
# INTERNET PROTOCOL

- Internet Protocol (IP)
- Responsible for relaying datagrams on a network
- Routing packets based on the IP address and packet headers
- IP uses addresses for routing

# TCP/IP PROTOCOLS

- Often referred to as a single unit: TCP/IP
- Responsible for most transmission on the internet
- TCP (or UDP) is responsible for creating, ordering, and checking packets
- IP is responsible for getting those packets to their destination

# OSI VS TCP/IP MODEL



# APPLICATION LAYER PROTOCOLS

- HyperText Transfer Protocol (HTTP)
  - Stateless protocol for transferring hypermedia, main protocol in browsers
- File Transfer Protocol (FTP)
  - Protocol for transferring files
- Email Protocols:
  - Simple Mail Transfer Protocol (SMTP)
  - Post Office Protocol version 3 (POP3)
  - Internet Message Access Protocol (IMAP)
- Ripple (XRP): payment protocol for sending money over the internet
- BitTorrent: protocol for peer-to-peer file sharing
- And several other protocols ...

# IP ADDRESSES

- Each computer connected to a network/internet has a unique IP address
  - Part of the Internet Protocol
  - Corporations vs local networks
  - Static vs dynamic IP addresses
- IP version 4 (IPv4):
  - Format: aaa.bbb.ccc.ddd
  - Each field (eg: aaa) has a value in [0,255] and is represented by one byte
  - Eg: 192.168.1.100

# IP ADDRESSES

- Each computer connected to a network/internet has a unique IP address
  - Corporations vs local networks
  - Static vs dynamic IP addresses
- IP version 4 (IPv4):
  - Format: aaa.bbb.cccddd
  - Each field (eg: aaa) has a value in [0,255] and is represented by 1 byte (8 bits)
  - Eg: 192.168.1.100
  - 4 fields x 8 bits = 32 bits per address
  - $2^{32} = 4,294,967,296$  = 4 billion distinct addresses
  - Not enough to serve 7 billion people

# IP ADDRESSES

- IP version 6 (IPv6):
  - Uses 128-bit addresses
  - Format: aaaa:bbbb:cccc:dddd:eeee:ffff:gggg:hhhh
  - Each field is in hexadecimal
  - Eg: 2001:db8:85a3:0:8a2e:370:7334
  - $2^{128} = 340,282,366,920,938,463,463,374,607,431,768,211,456 = 340 \text{ undecillion distinct addresses}$

# IP ADDRESSES

- IP version 6 (IPv6):
  - Uses 128-bit addresses
  - Format: aaaa:bbbb:cccc:dddd:eeee:ffff:gggg:hhh
  - Each field is in hexadecimal
  - Leading 0s can be omitted
  - Eg: 2001:db8:85a3:0:0:8a2e:370:7334
  - $2^{128} = 340,282,366,920,938,463,463,374,607,431,768,211,456 = 340 \text{ undecillion distinct addresses}$

# LOCALHOST

- Your local machine has a special IP address
- IPv4: 127.0.0.1
- IPv6: ::1
- Hostname: localhost



# OVERVIEW OF DOMAINS

- Computers can easily remember “random” numbers (IP addresses)
- Humans not so much – they can remember words much better
- The words are called domains
- Eg: google.com or cs.up.ac.za
- Domain Name System (DNS):
  - Converts domain names (words) into IP addresses (numbers)
  - Hence, converts from human notation into machine notation
  - These names are called Fully Qualified Domain Names (FQDN)
  - The server doing the conversion is the Domain Name Server (DNS) – yes, DNS is reused

# DOMAIN NAME SERVER

- The DNS operates as follows:
  - DNS contains a large table with domain names and IP addresses
  - If a domain name comes in as a request, it searches through the table to find the IP address. This process is called “resolving”
  - The IP address is returned to the caller
  - DNS uses UDP
- DNS is created and maintained by:
  - Internet Service Providers (ISPs) – most home routers' default settings
  - Large corporations (eg: Google and Microsoft)
  - Governments (eg: China and North Korea)
  - Root DNS

# ROOT DNS

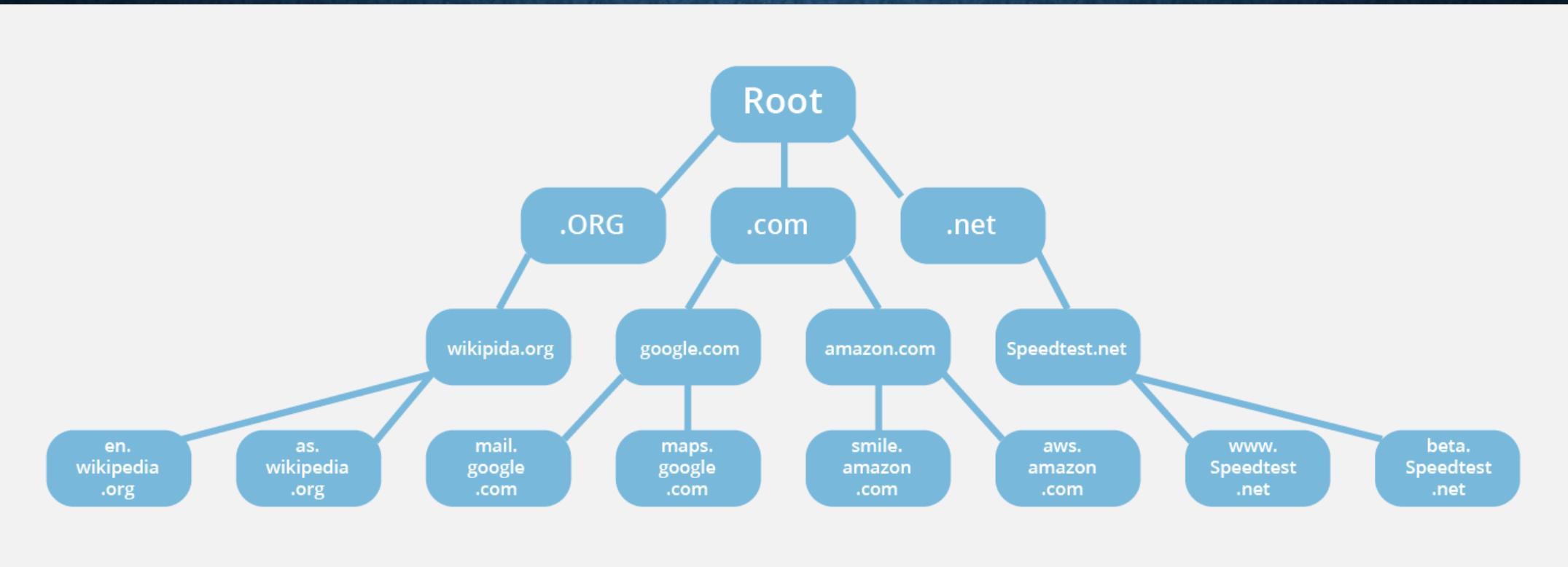
- Not all DNS servers keep track of all domains
- DNS entries are updated on regular intervals
- DNS works in a hierarchy
  - Root DNS is the top hierarchy, keeping track of top-level domains (TLDs)
  - Lower level DNS may keep track of certain domains, depending on the requirements
  - DNS can resolve addresses on local machines, eg: localhost
- Currently 13 root DNS servers (IPs)
  - With anycast addressing there are 1600+ DNS root servers (Feb 2023)

# ROOT DNS

## List of Root Servers

HOSTNAME	IP ADDRESSES	OPERATOR
a.root-servers.net	198.41.0.4, 2001:503:ba3e::2:30	Verisign, Inc.
b.root-servers.net	199.9.14.201, 2001:500:200::b	University of Southern California, Information Sciences Institute
c.root-servers.net	192.33.4.12, 2001:500:2::c	Cogent Communications
d.root-servers.net	199.7.91.13, 2001:500:2d::d	University of Maryland
e.root-servers.net	192.203.230.10, 2001:500:a8::e	NASA (Ames Research Center)
f.root-servers.net	192.5.5.241, 2001:500:2f::f	Internet Systems Consortium, Inc.
g.root-servers.net	192.112.36.4, 2001:500:12::d0d	US Department of Defense (NIC)
h.root-servers.net	198.97.190.53, 2001:500:1::53	US Army (Research Lab)
i.root-servers.net	192.36.148.17, 2001:7fe::53	Netnod
j.root-servers.net	192.58.128.30, 2001:503:c27::2:30	Verisign, Inc.
k.root-servers.net	193.0.14.129, 2001:7fd::1	RIPE NCC
l.root-servers.net	199.7.83.42, 2001:500:9f::42	ICANN
m.root-servers.net	202.12.27.33, 2001:dc3::35	WIDE Project

# ROOT DNS



# ROOT DNS MAP



<https://root-servers.org/>

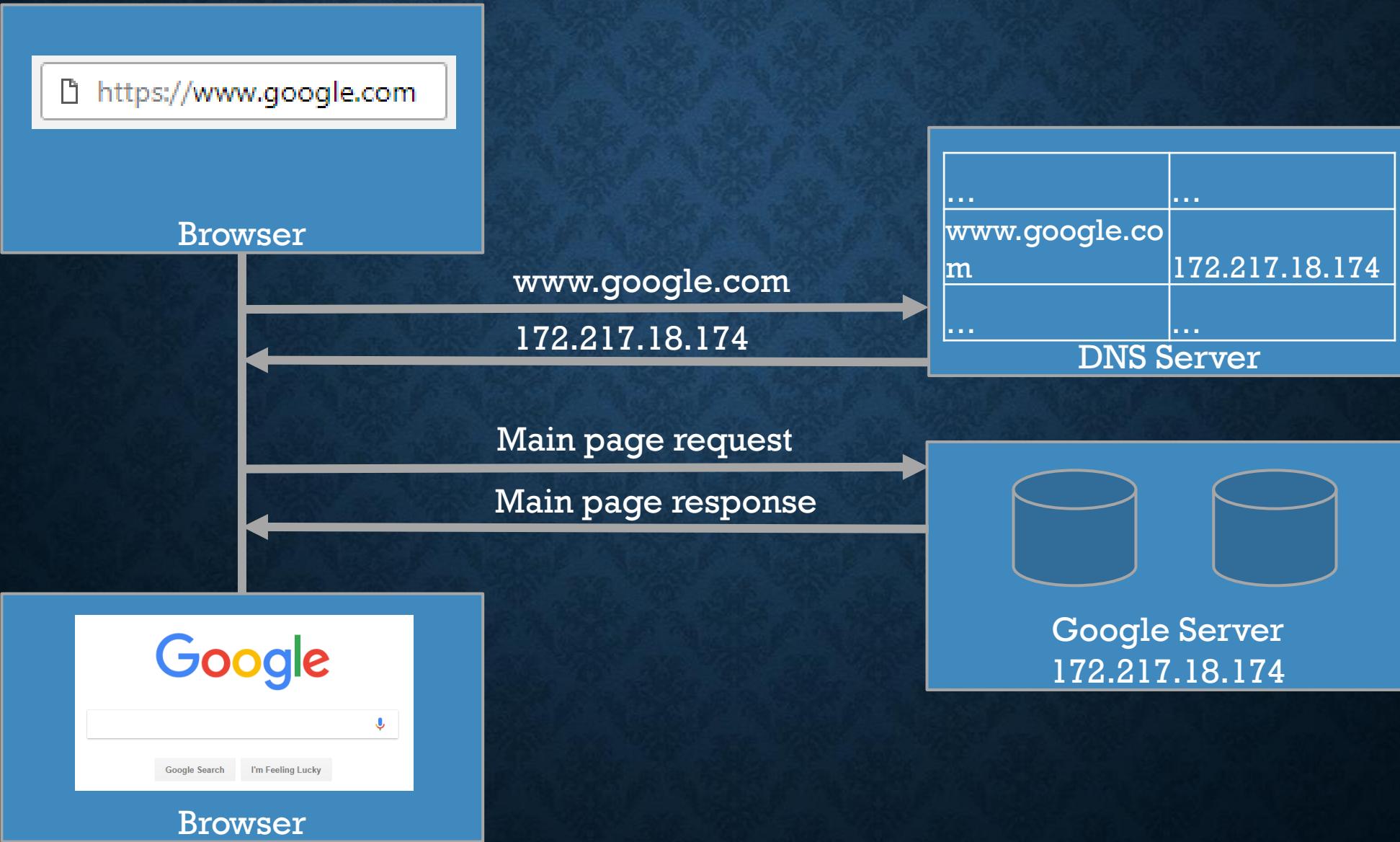
# DNS OVER HTTPS (DOH)

- Domain Name System (DNS) resolution via the HTTPS protocol
- Prevents eavesdropping (Government)
- Increases Privacy (Encryption) [does not **ensure** privacy]
- Prevents Man-in-the-Middle attacks
- Google and Mozilla have begun testing this, with Firefox already available to the end user

# EXAMPLE OF DNS TABLE

mail.xx.com	IN	A	102.34.23.6
host.xx.com	IN	MX	mail.xx.com
www.xx.com	IN	A	102.34.23.8
ftp.xx.com	IN	CNAME	www.xx.com

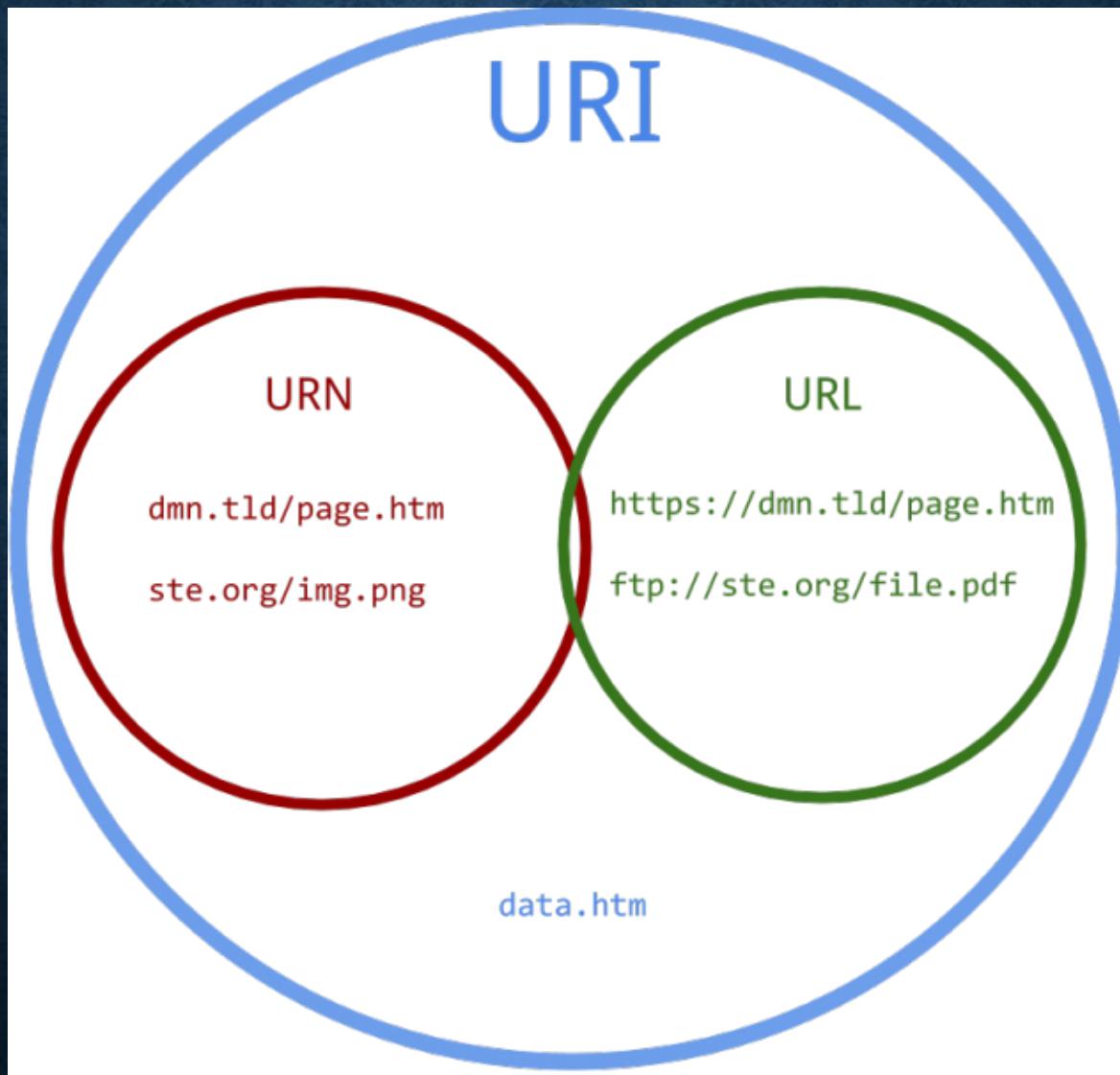
# OVERVIEW OF DNS



# UNIFORM RESOURCE IDENTIFIER

- Uniform Resource Identifier (URI)
- String of characters used to uniquely identify a resource on the internet
- Uniform Resource Locator (URL)
  - Web address
- Uniform Resource Name (URN)
  - Complement URLs
  - Mechanism for the identification of resources in particular namespaces

# UNIFORM RESOURCE IDENTIFIER



# UNIFORM RESOURCE LOCATOR

- URLs adhere to the URI syntax:

**scheme:[//[user[:password]@]host[:port]][/path][?query][#fragment]**

- Example:

**https://satoshi:Complex1\$Pass@gitHub.com:80/satoshi/bitcoin/readme.txt**

# UNIFORM RESOURCE LOCATOR

**scheme:[//[user[:password]@]host[:port]][/path][?query][#fragment]**

**[https://satoshi:Complex1\\$Pass@github.com:80/satoshi/bitcoin/readme.txt](https://satoshi:Complex1$Pass@github.com:80/satoshi/bitcoin/readme.txt)**

- The scheme consisting of a sequence of characters
- Typically the protocol:
  - http: Standard HTTP protocol
  - https: Secured HTTP protocol with SSL/TLS encryption
  - ftp: Standard FTP protocol
  - ftps: Secured FTP protocol with SSL/TLS encryption
  - file: Indicates a local file. Eg: file:///C:/Users/satoshi/Desktop/test.html

# UNIFORM RESOURCE LOCATOR

**scheme://[:user[:password]@]host[:port]][/path][?query][#fragment]**

**https://satoshi:Complex1\$Pass@github.com:80/satoshi/bitcoin/readme.txt**

- Scheme is followed by:
  - A colon
  - 2 or 3 forward slashes

# UNIFORM RESOURCE LOCATOR

**scheme://[user[:password]@]host[:port]][/path][?query][#fragment]**

**https://satoshi:Complex1\$Pass@github.com:80/satoshi/bitcoin/readme.txt**

- Some sites support the passing of usernames and password to access restricted content.
- Content can be accessed without have to manually login.
- The username and password is separated by a colon.
- The password is followed by a @ before the host.
- If no login details are provided, omit: user:password@

# UNIFORM RESOURCE LOCATOR

**scheme://[user[:password]@]host[:port]][/path][?query][#fragment]**

**https://satoshi:Complex1\$Pass@github.com:80/satoshi/bitcoin/readme.txt**

- The host as a FQDN
- Includes:
  - Top-level domain (TLD). Eg: com
  - Domain name. Eg: google
  - Subdomains. Eg: www
- Can also be:
  - IP address
  - Localhost. Eg: 127.0.0.1 or localhost

# UNIFORM RESOURCE LOCATOR

**scheme://[user[:password]@]host[:port]][/path][?query][#fragment]**

**https://satoshi:Complex1\$Pass@github.com:80/satoshi/bitcoin/readme.txt**

- The port of the server, separated by a colon
- Different applications and protocols run over different ports
- If the port is omitted, the browser uses the default port:
  - http: port 80
  - https: 443
  - ftp: 21

# UNIFORM RESOURCE LOCATOR

**scheme://[user[:password]@]host[:port]][/path][?query][#fragment]**

**https://satoshi:Complex1\$Pass@github.com:80/satoshi/bitcoin/readme.txt**

- The full path of the resource
- Can be a directory or file path
- Uses Unix-style paths, separated by forward slashes
- The path can be:
  - A real path with subdirectories and files
  - An emulated path that does not have real directories or files, but the path is interpreted by the server

# UNIFORM RESOURCE LOCATOR

**scheme://[:user[:password]@]host[:port]][/path][?query][#fragment]**

**<https://github.com/satoshi/bitcoin/script.php?type=data&json=1>**

- A query with parameters
- Values can be send to the server (script or web server)
- The query:
  - Starts with a question mark
  - Parameters are separated by an ampersand
  - Parameters take the format: name=value

# UNIFORM RESOURCE LOCATOR

**scheme://[:user[:password]@]host[:port]][/path][?query][#fragment]**

<https://github.com/satoshi/bitcoin/index.html#section1>

- A fragment on the current page
- Is interpreted on the client-side
- Typically used to jump to a specific heading on a website when opened, instead of showing the top-most part of the site

# DOMAIN NAMES

**www.google.com**

**cs.up.ac.za**

- Top-level domains (TLDs)
  - Always required
  - First look-up when resolving the host
  - Managed by the Internet Corporation for Assigned Names and Numbers (ICANN)

# DOMAIN NAMES

**www.google.com**

**cs.up.ac.za**

- Second-level domains (SLDs)
  - Optional, depending on TLD
  - Second look-up when resolving the host, after the TLD
  - Managed by local domain name registrar. Eg: \*.za is managed by ZA Central Registry
  - Can vary depending on the registrar:
    - co (commercial), ac (academic institutions), gov (government institutions), net (network providers), mil (Department of Defense), etc

# DOMAIN NAMES

**www.google.com**

**cs.up.ac.za**

- Domain name
  - Domain registered by company or individual
  - Many companies provide services for domain registrations
  - Although the domain belongs to you, it is still under control of the registrar

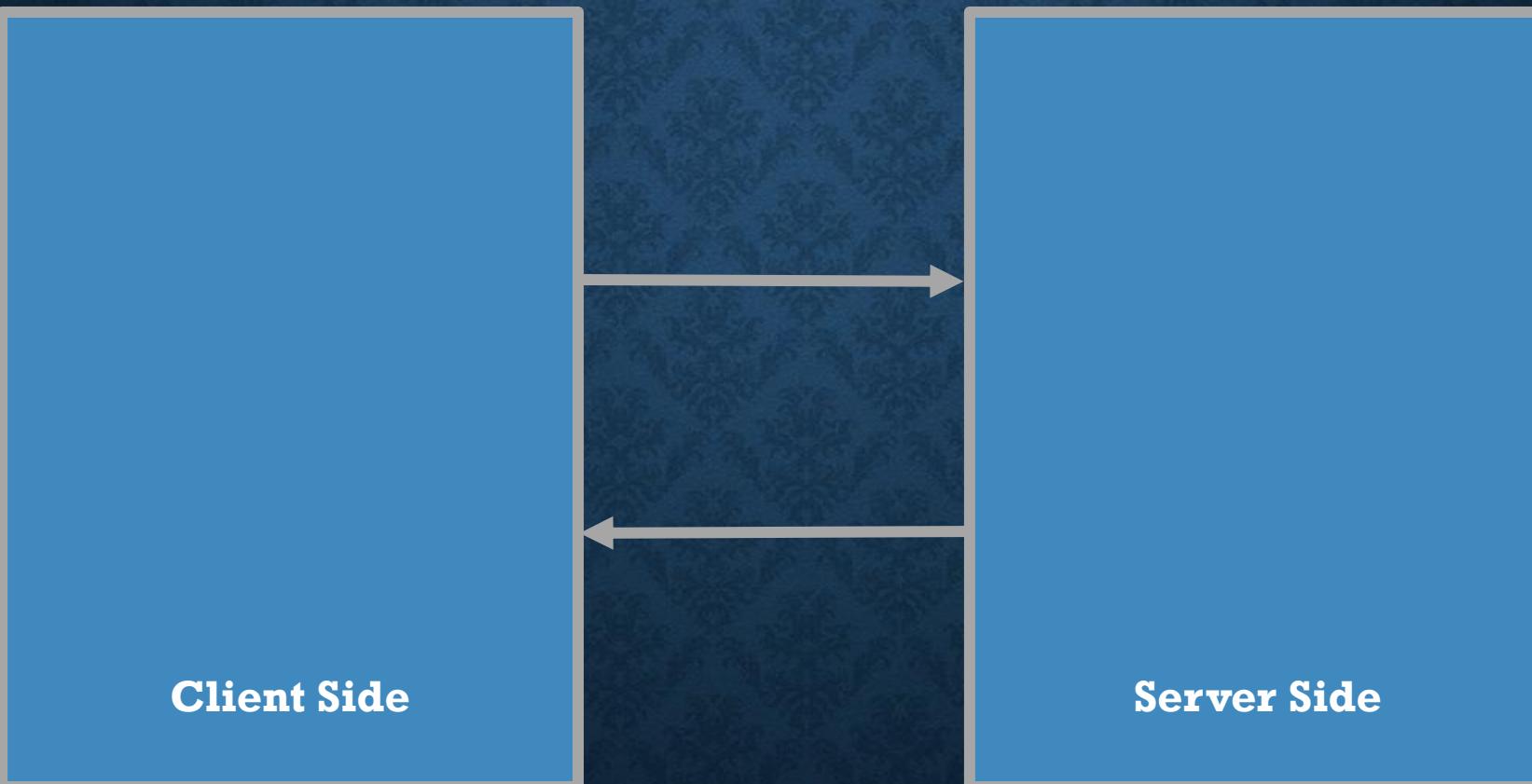
# DOMAIN NAMES

**www.google.com**

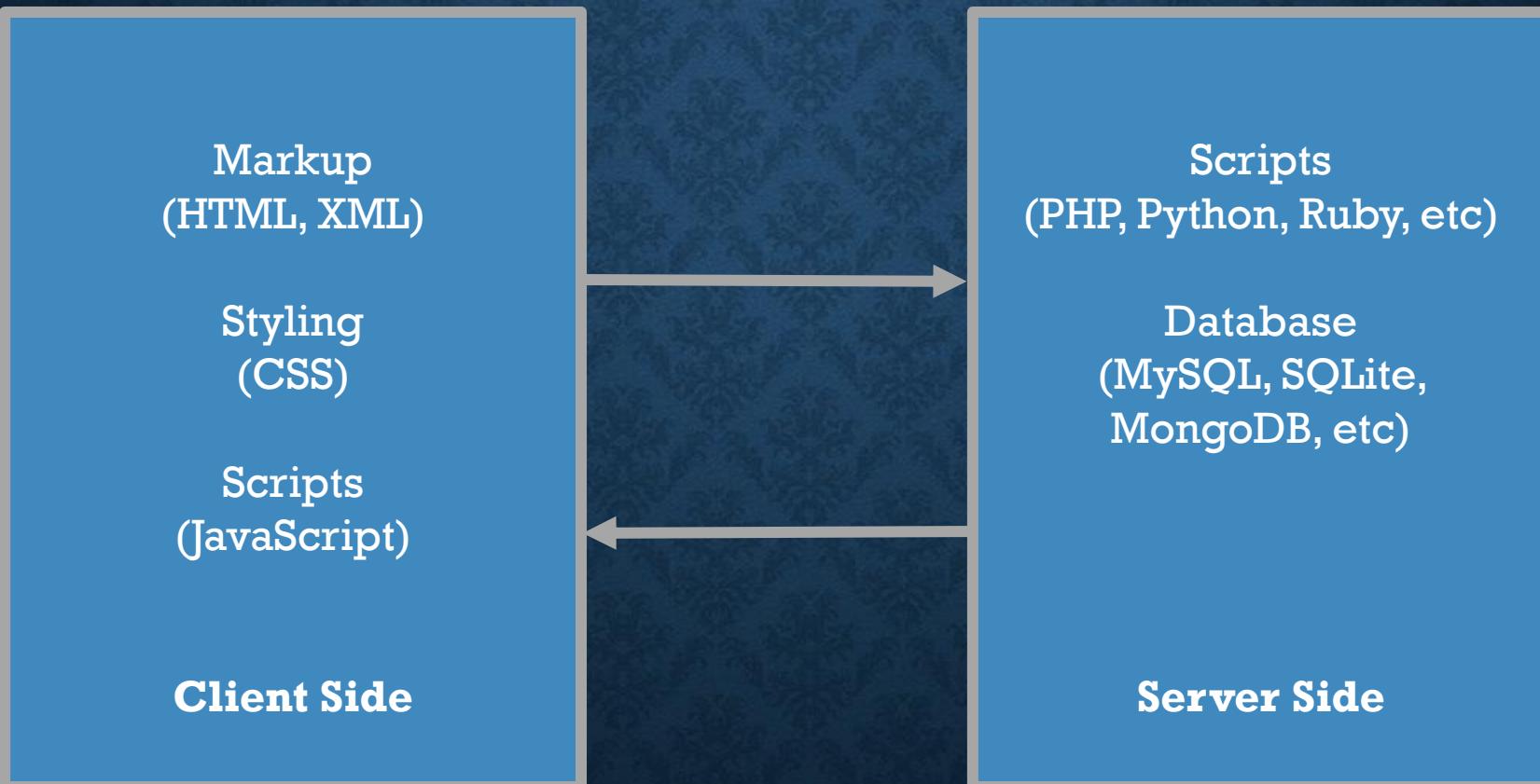
**cs.up.ac.za**

- Subdomain
  - Zero or more subdomains
  - Under the control of the company/individual who registered the domain name.
  - Does not require an additional registration, can be added/removed directly on the server and by adding DNS entries
  - Most common one: www (world wide web)
  - Most modern websites can be accessed without www

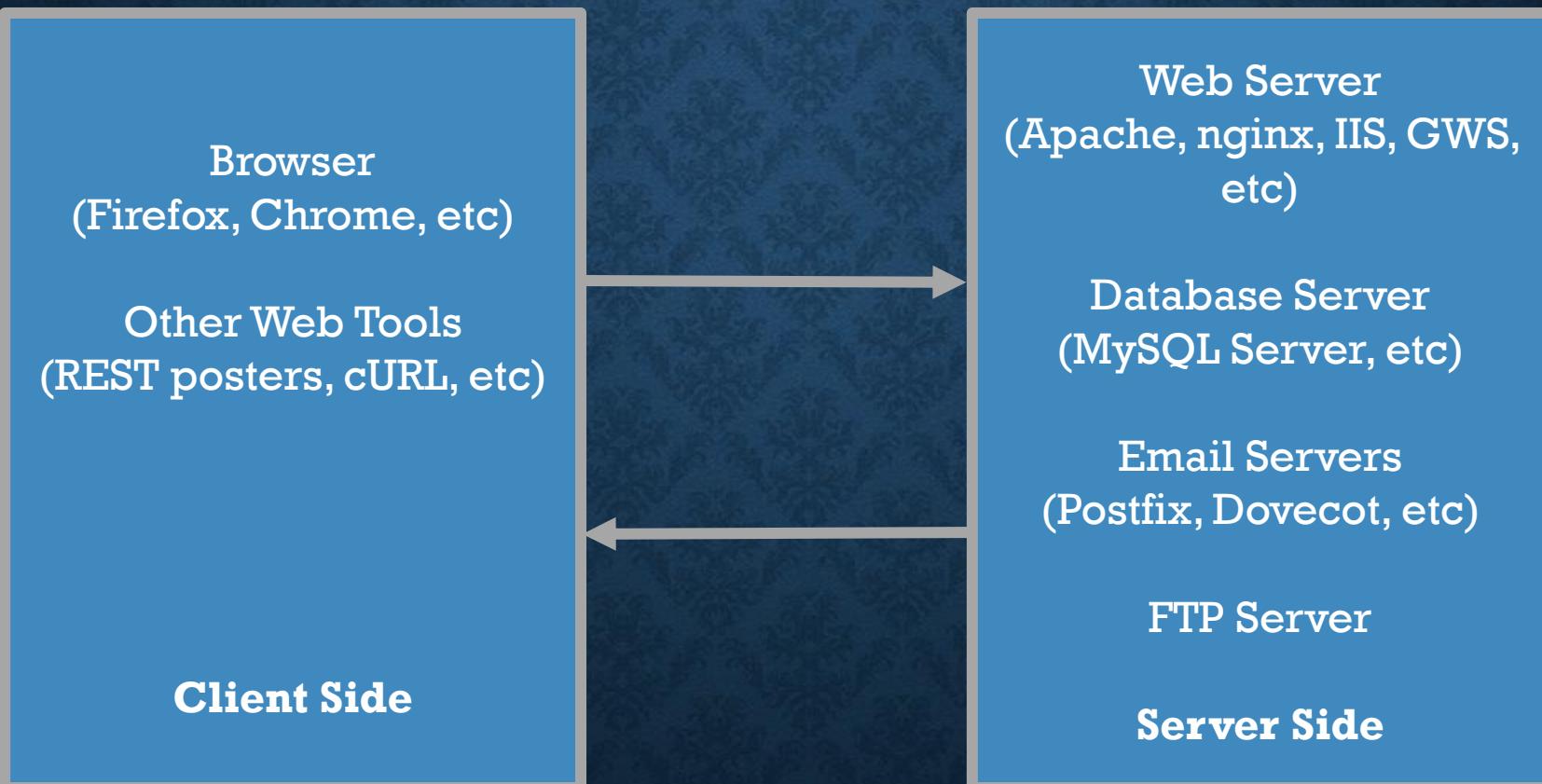
# CLIENT-SERVER ARCHITECTURE



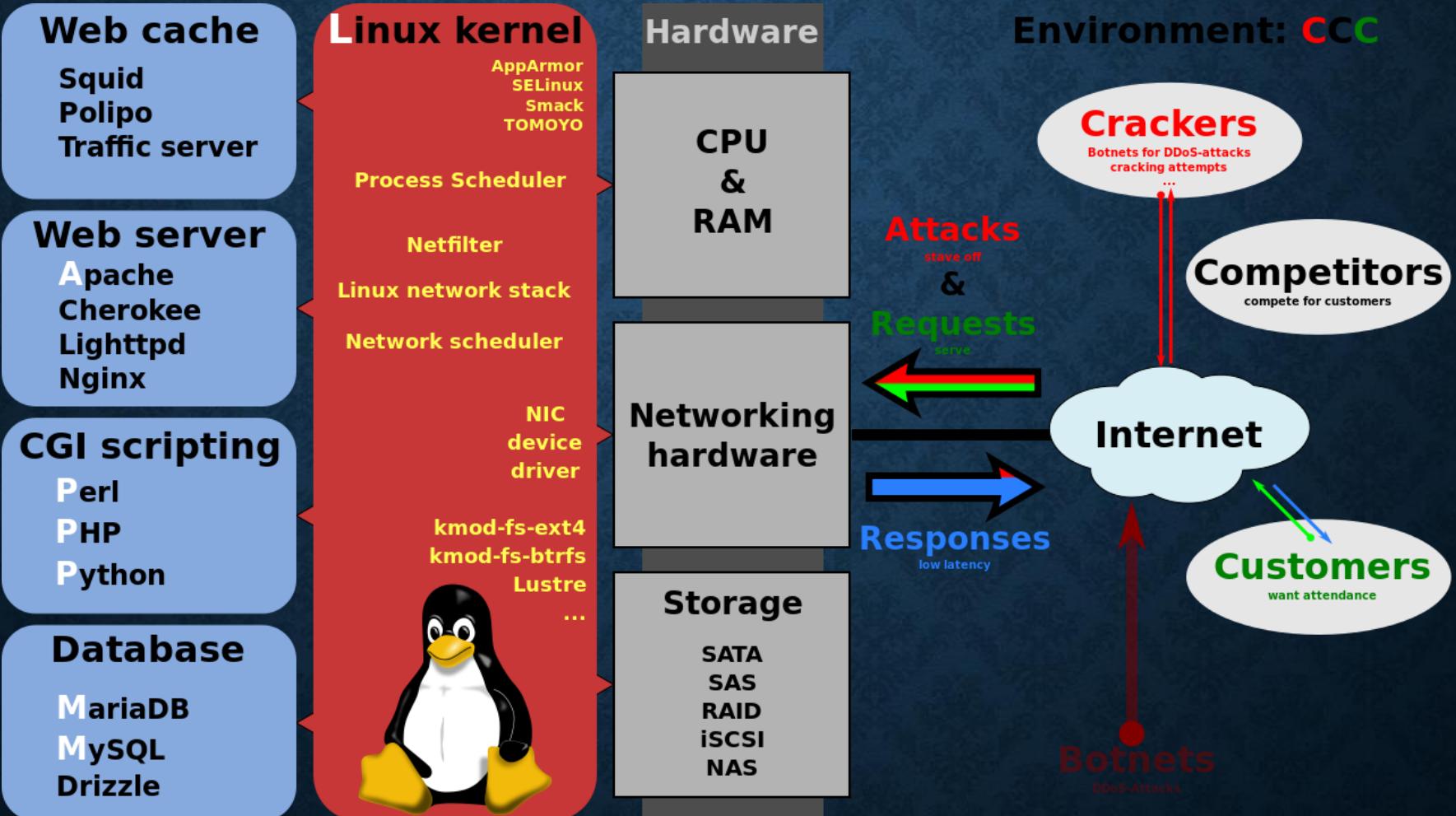
# CLIENT-SERVER LANGUAGES



# CLIENT-SERVER SOFTWARE



# CLIENT-SERVER LAMP (COS216)



# CLIENT-SERVER LAMP (COS216)

Linux-Apache-MySQL-PHP (LAMP)

Client Side:

- HTML
- CSS
- JavaScript

Server Side:

- Linux OS
- Apache
- PHP
- MySQL

Communication:

- JSON

# COS216 SERVER

**wheatley.cs.up.ac.za**

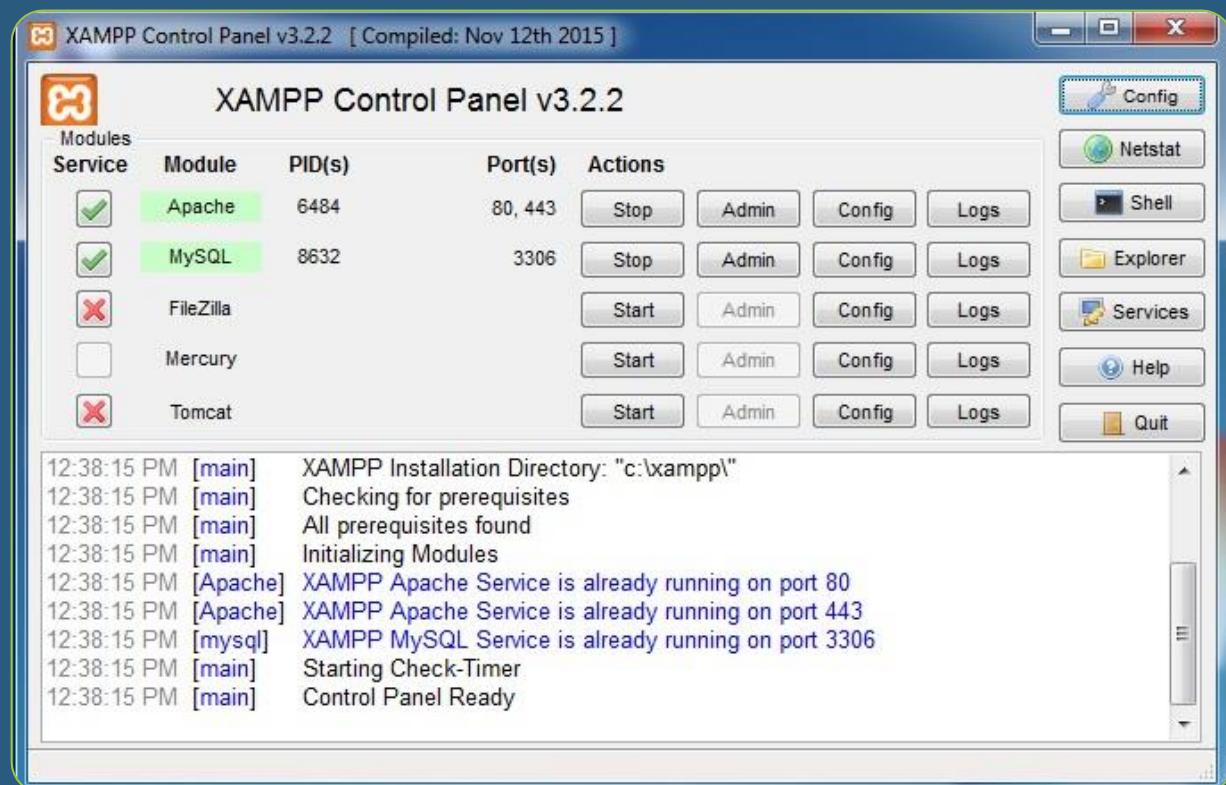
- Accessible from outside campus
- User your CS username and password
- Upload your files via FTP
- Access the website through your browser

# COS216 LOCALHOST

- If you want to work locally, you can setup your own server.
- We recommend using XAMPP:
  - <https://www.apachefriends.org>
  - Comes with Apache, PHP, MariaDB (open-source MySQL), FileZilla, and phpMyAdmin
  - Cross-platform: Linux, Windows, Mac
  - **Make sure to test your code on Wheatley before the pracs**

# XAMPP

- Easily manage your webserver



# XAMPP



Launch XAMPP



Start the Apache and MySQL servers (FTP server is not necessary)



Create a new directory under <xampp install path>/htdocs/COS216



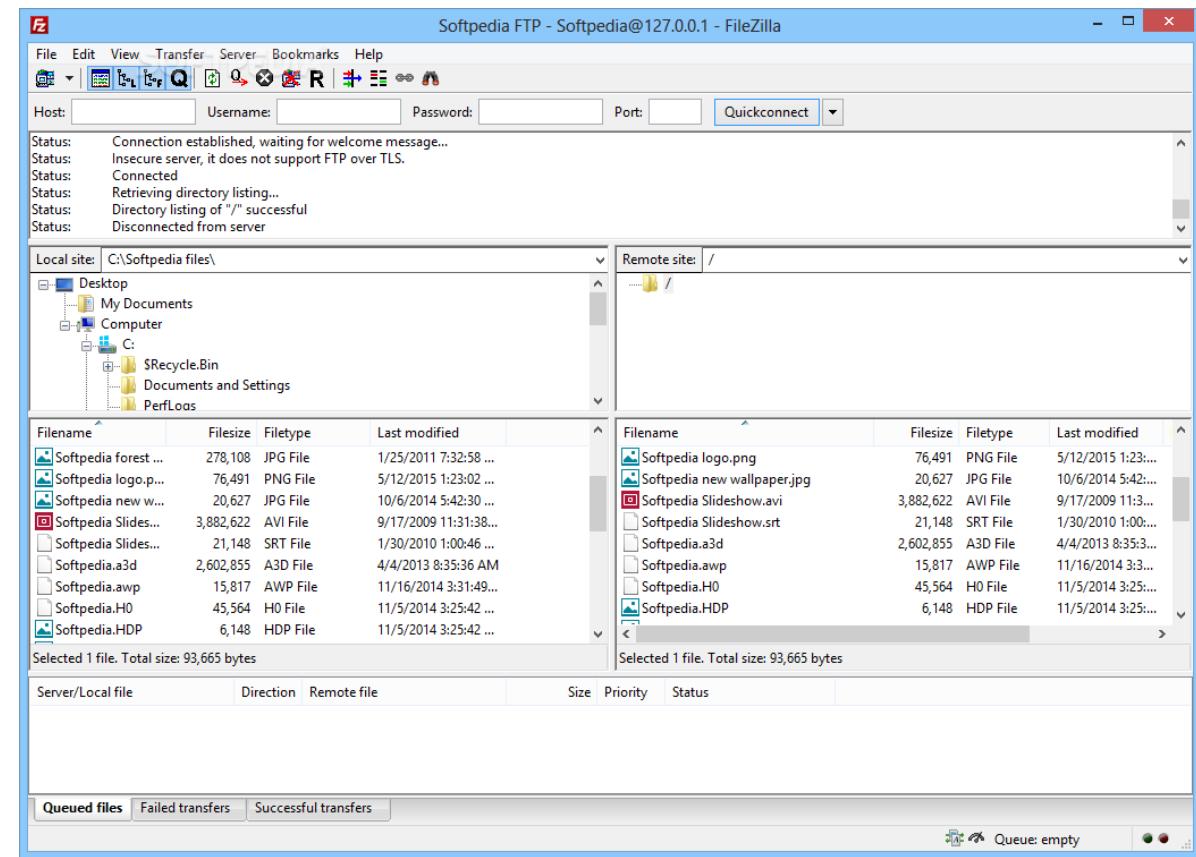
Create a new file test.txt under the COS216 directory



Access the file in your browser:  
<http://localhost/COS216/test.txt>

# **FTP - FILEZILLA**

Easily upload your files  
via FTP



# FTP

- Or use the terminal **ftp** command to upload
  1. `ftp wheatley.cs.up.ac.za`
  2. <enter your CS username>
  3. <enter your CS password>
  4. Change directory, upload files, delete files, rename or move files
  5. More info on the commands: <https://www.cs.colostate.edu/helpdocs/ftp.html>

# MYSQL - PHPMYADMIN

- Easily manage your databases  
via a web-based interface

The screenshot shows the phpMyAdmin interface for managing MySQL database privileges. The left sidebar lists various MySQL system tables under the 'mysql' database. The main content area displays a table titled 'Database privileges' with 22 rows, each representing a privilege name and its characteristics. The columns include: # (row number), Nombre (privilege name), Tipo (type), Cotejamiento (comparison), Atributos (attributes), Nulo (nullability), Predeterminado (default value), Extra (extra options), and Acción (action buttons). The 'Nombre' column lists privileges such as Host, Db, User, Select\_priv, Insert\_priv, Update\_priv, Delete\_priv, Create\_priv, Drop\_priv, Grant\_priv, References\_priv, Index\_priv, Alter\_priv, Create\_tmp\_table\_priv, Lock\_tables\_priv, Create\_view\_priv, Show\_view\_priv, Create\_routine\_priv, Alter\_routine\_priv, Execute\_priv, Event\_priv, and Trigger\_priv. The 'Tipo' column generally shows 'enum('N','Y') utf8\_general\_ci' or 'char(80)' or 'char(64)'. The 'Cotejamiento' column mostly shows 'utf8\_bin'. The 'Atributos' column often includes 'No' or 'N'. The 'Nulo' column includes 'No' or 'N'. The 'Predeterminado' column includes 'No'. The 'Extra' column includes 'Cambiar', 'Eliminar', and 'Más'. The 'Acción' column contains icons for these actions.

# MYSQL

- Or use the terminal **mysql** command to access your database
  - More details to follow in SQL Lectures

