Web Application Fundamentals

Tessema Mengistu (Ph.D.)

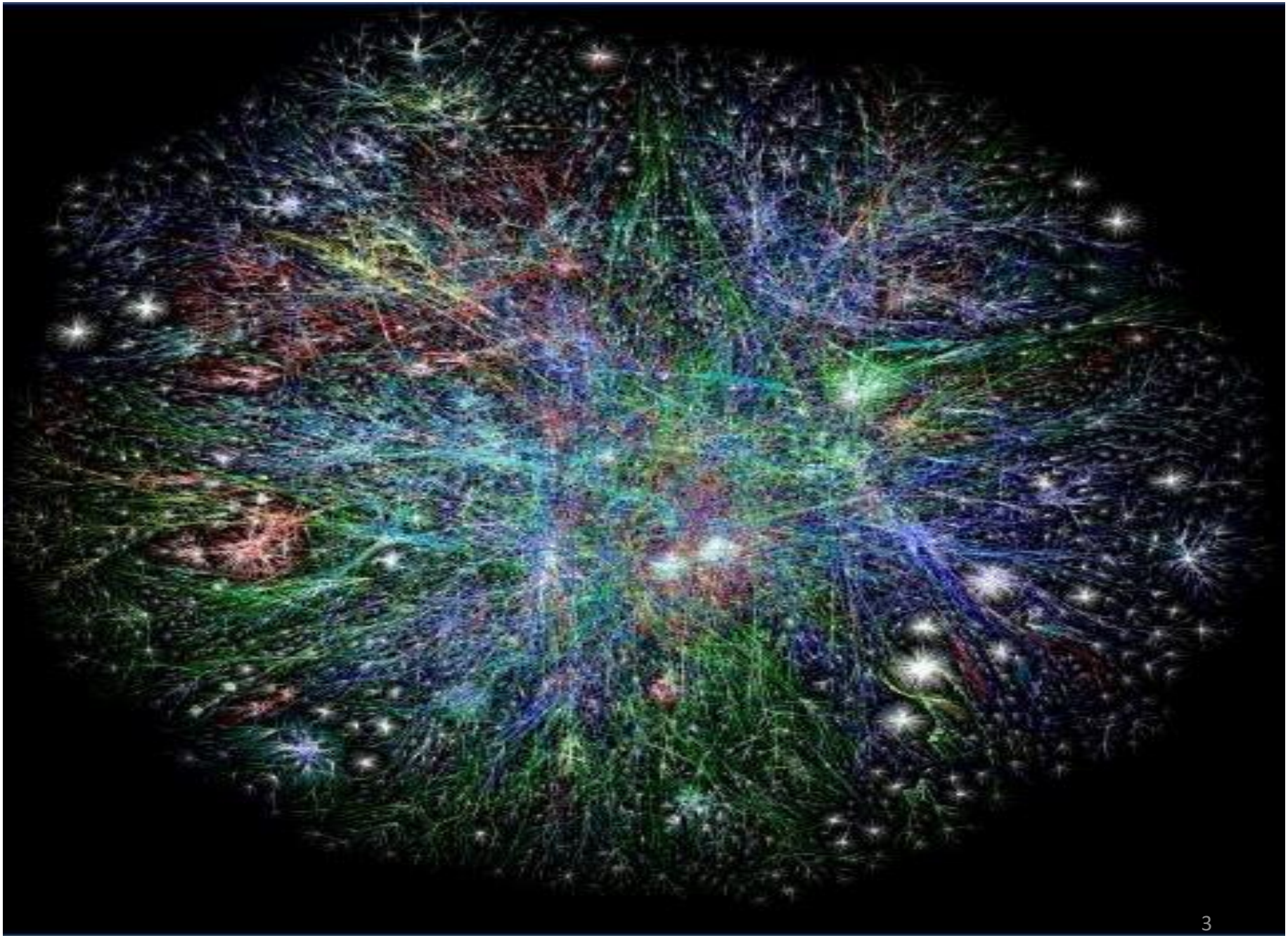
Department of Computer Science

Virginia Tech

mengistu@vt.edu

Outline

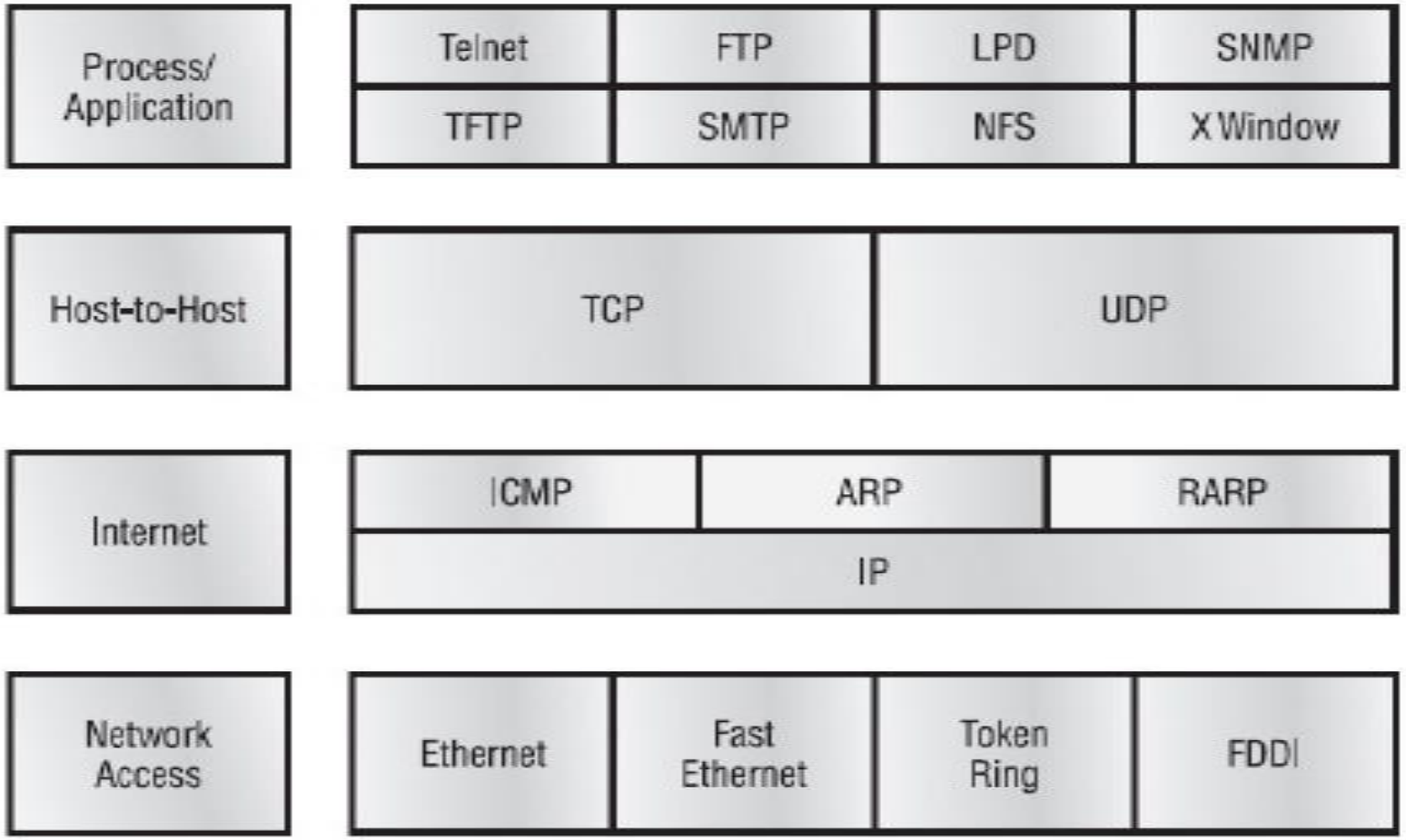
- Overview of the Internet
- The World Wide Web
- Web Application Architecture
- Web Programming Tools



Internet

- A network of networks, joining many government, university and private computers together and providing an infrastructure for different uses.
- Uses **TCP/IP** protocols.
- Based on **Packet Switching.**

TCP/IP

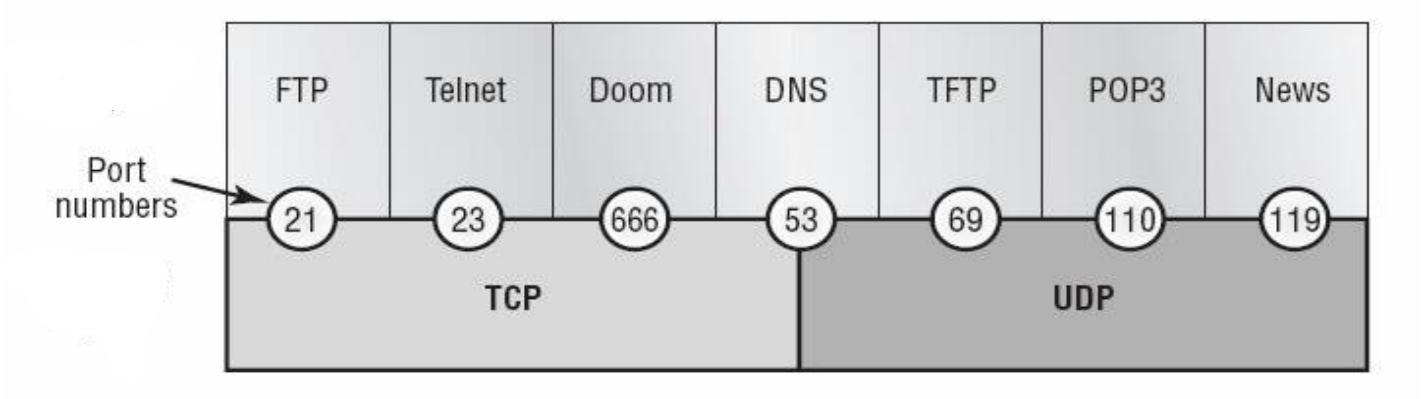

Process/ Application
Telnet
FTP
LPD
SNMP
TFTP
SMTP
NFS
X Window
Host-to-Host
TCP
UDP
Internet
ICMP
ARP
RARP
IP
Network Access
Ethernet
Fast Ethernet
Token Ring
FDDI

TCP/IP

- **Transmission Control Protocol (TCP)**
  - Connection based protocol
  - Uses three-way handshaking
- **User Datagram Protocol (UDP)**
  - Connectionless protocol
- TCP for reliability and UDP for faster transfers

TCP/IP

- TCP and UDP must use **port numbers** to communicate with the upper layers.



- TCP Ports
  - Telnet 23
  - SMTP 25
  - HTTP 80
  - FTP 21
  - HTTPS 443
  - SSH 22

UDP Ports

SNMP 161
TFTP 69
DNS 53
POP3 110

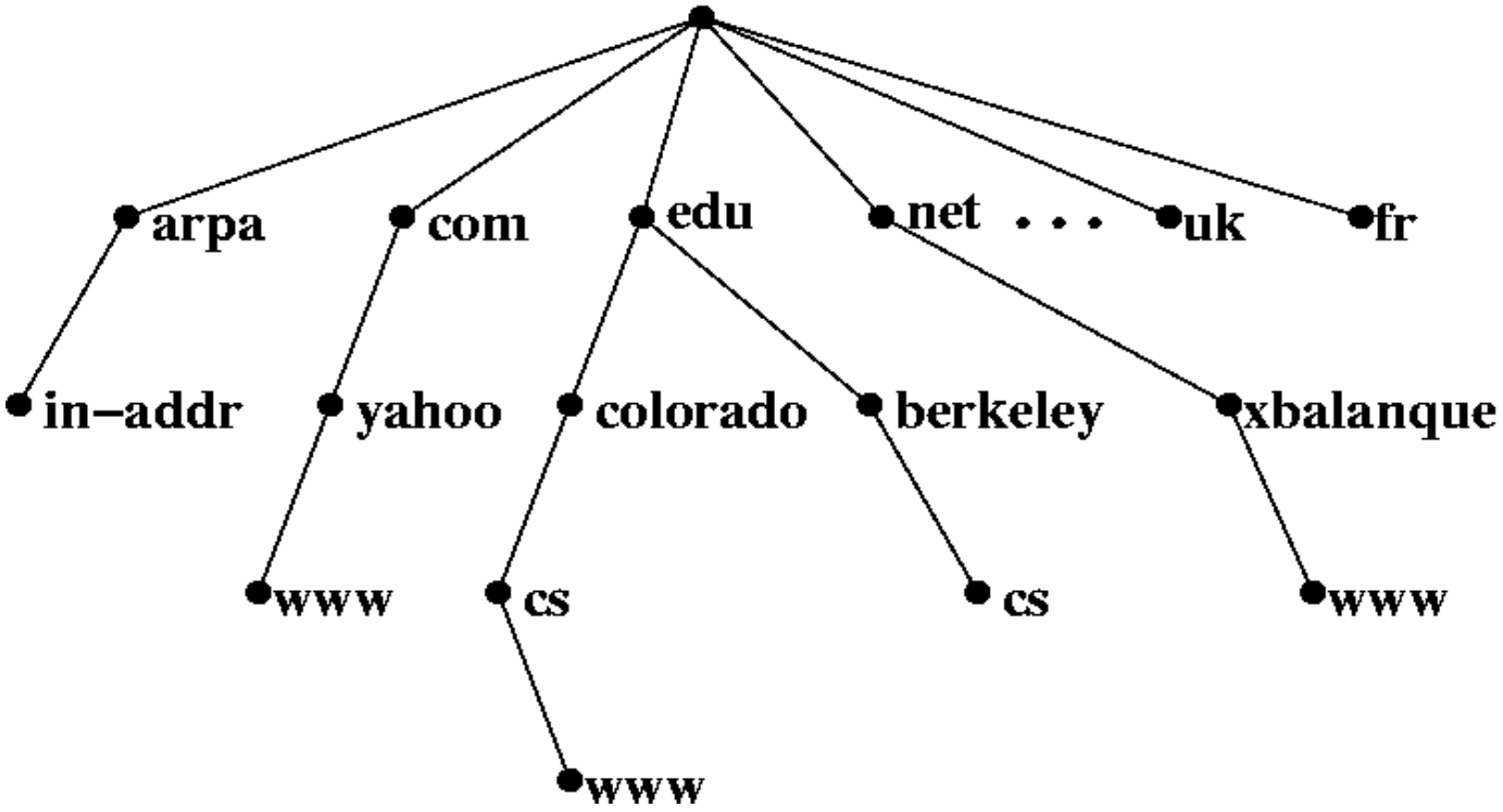
IP Addressing

- Internet Protocol (IP) Addresses
  - Every Computer on the Internet has two addresses
    - Hardware based address – MAC Address
    - Software address – IP Address
- IP Address
  - IPv4: 32-bit binary number
    - 192.168.10.0.
  - IPv6, has 128 bits (1998)
    - 42DE:7E55:63F2:21AA:CBD4:D773:CC21:554F
- Organizations are usually assigned groups of IPs for their computers

DNS – Domain Name System

- Domain names
  - Example: www.vt.edu
- Form:
  - host-name.domain-names
    - First domain is the smallest; and the last is the largest
    - Last domain specifies the type of organization
- ***Fully qualified domain name*** - the host name plus all the domain names
- DNS servers - convert fully qualified domain names to IP address

DNS – Domain Name System



The World Wide Web(WWW)

Origins

- Tim Berners-Lee at CERN proposed the Web in 1989

- Purpose: to allow scientists to have access to many databases of scientific work through their own computers

- Document form: hypertext - texts with embedded links to documents

- Hypermedia – more than just text – images, sound, etc. as well

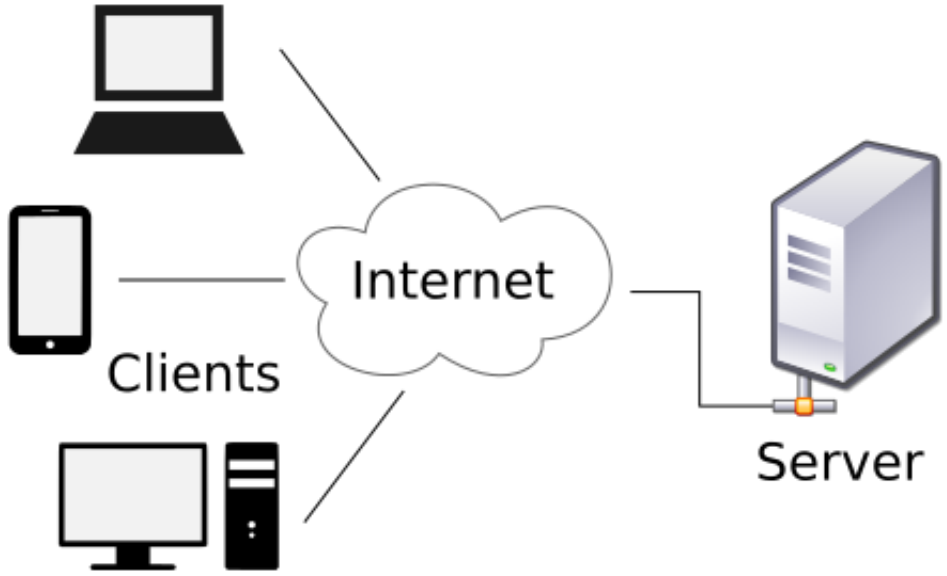
The World Wide Web(WWW)

Web vs. Internet

- The Web mainly uses one of the protocols, http, that runs on the Internet -- there are many other protocols used as well (telnet, mailto, ftp etc.)

The World Wide Web(WWW)

- Provide responses to browser requests, with either *existing* documents or *dynamically* built documents
- Browser-server connection is now maintained through more than one request-response cycle



The World Wide Web(WWW)

- A web **browser** (commonly referred to as a browser) is a software application for retrieving, presenting and traversing information resources on the Web
- Browsers are clients - always initiate, servers react (although sometimes servers require responses)
  - Most requests are for existing documents, using HyperText Transfer Protocol (HTTP)
  - But some requests are for program execution, with the output being returned as a document and sent back to client
- Chrome, Mozilla, Firefox, etc.

The World Wide Web(WWW)

- Web servers run as background processes in the operating system
- Monitor a communication port on the host (port 80 for the case of http), accepting HTTP messages when they appear

The World Wide Web(WWW)

- Apache (open source, fast, reliable)
  - Started as the NCSA server, named *httpd*
  - Maintained by editing its configuration file
  - Open source and works on different platforms (Linux, Windows , etc.).
- IIS (Internet Information Server) from Microsoft
  - Maintained through a program with a GUI interface
  - Works only for Windows environment
- Nginx - Open source
  - Runs in different platforms (Linux, Windows, Solaris, etc.)

URL (Universal Resource Locator)

- Identify resources on the Internet
- General form:
  - *scheme:object-address*
    - The “scheme” is a communication protocol, such as *telnet* or *ftp*
  - For the *http* protocol, the object-address is:
    - fully-qualified-domain-name/doc-path
  - For the *file* protocol, only the doc-path is needed
  - Host name may include a port number that identifies a specific process
    - localhost:8080

HTTP: HyperText Transfer Protocol

- The main protocol used by ALL Web communications
- Has two phases:
  - Request
  - Response
- Each http communication consists two parts
  - The Header - information about communication
  - The Body – Data of the communication

HTTP: HyperText Transfer Protocol

Request Phase

- Form:

  HTTP method domain part of URL HTTP ver.

  Header fields

  blank line

  Message body

- An example of the first line of a request:

  GET /cs.vt.edu/degrees.html HTTP/1.1

HTTP: HyperText Transfer Protocol

Most commonly used HTTP methods:

- **GET** - Fetch a document
- **POST** - Execute the document, using the data in body
- **HEAD** - Fetch just the header of the document
- **PUT** - Store a new document on the server
- **DELETE** - Remove a document from the server

HTTP: HyperText Transfer Protocol

- Four categories of header fields:

  - general (for general info such as date),
  - request (used only in request message),
  - response (only for response),
  - entity (for both request and response)

- Common request fields:

  - Accept: text/plain
  - Accept: text/*
  - If-Modified_since: date

HTTP: HyperText Transfer Protocol

- Response Phase
  - Form:
    Status line
    Response header fields
    blank line
    Response body

  - Status line format:
    HTTP version status code explanation

  - Example:
    HTTP/1.1 200 OK
    (Current version is 1.1)

HTTP: HyperText Transfer Protocol

- Status code is a three-digit number; first digit specifies the

  general status

  1 => Informational

  2 => Success

  3 => Redirection

  4 => Client error

  - (e.g., 400 URL Error and 404 Not Found)

  5 => Server error

- The header field, content-type, is always required!

HTTP: HyperText Transfer Protocol

An example of a complete response header:

```
HTTP/1.1  200  OK
Date: Tues, 18 May 2004 16:45:13 GMT
Server: Apache (Red-Hat/Linux)
Last-modified: Tues, 18 May 2004 16:38:38 GMT
Etag: "841fb-4b-3d1a0179"
Accept-ranges: bytes
Content-length: 364
Connection: close
Content-type: text/html, charset=ISO-8859-1
[_____________________________________________]
```

- Both request and response headers must be followed by a *blank line*

Web Application Architecture

- A web application consists of two basic components:
  - **client-side** - runs in a browser
  - **server-side** - runs at the backend

Web Application Architecture

- Web applications consist of multiple layers (tiers) responsible for specific functions:
  - 2-tier architecture – Client/Server
  - 3-tier architecture – Model/View/Controller
  - Multi-tiered architecture
- A typical web app has three layers:
  - Presentation - handles user interactions on the client application
  - Application – handles the business logic
  - Data - manages data storage and management

Web Application Architecture

- Web Application (Software) Architecture
  - Client Side:
    - Single-Page Applications (SPA)
    - Multi-Page Applications (MPA)
    - Progressive Web Applications (PWA)
  - Server Side:
    - Microservices
    - Monolith
    - Serverless

Web Application Architecture

- Client: **Single-page Applications (SPA)**
  - All functionality takes place on a single page
  - When there is a change, the application doesn't reload the entire page, but only the part that is changed.
  - Advantages
    - Fast performance - fast, lightweight, and instantly responsive
    - Flexible UX
  - Disadvantage
    - First-load speed
    - Testing
- Example, Google Map, Twitter, etc.

Web Application Architecture

- Client: **Multi-Page Applications (MPA)**
  - Multiple workflows and functions
  - A few dynamic web pages each performing a distinctive function and reloading when a user sends a new request
  - Advantage
    - Rich functionality
  - Disadvantage
    - Slower speed

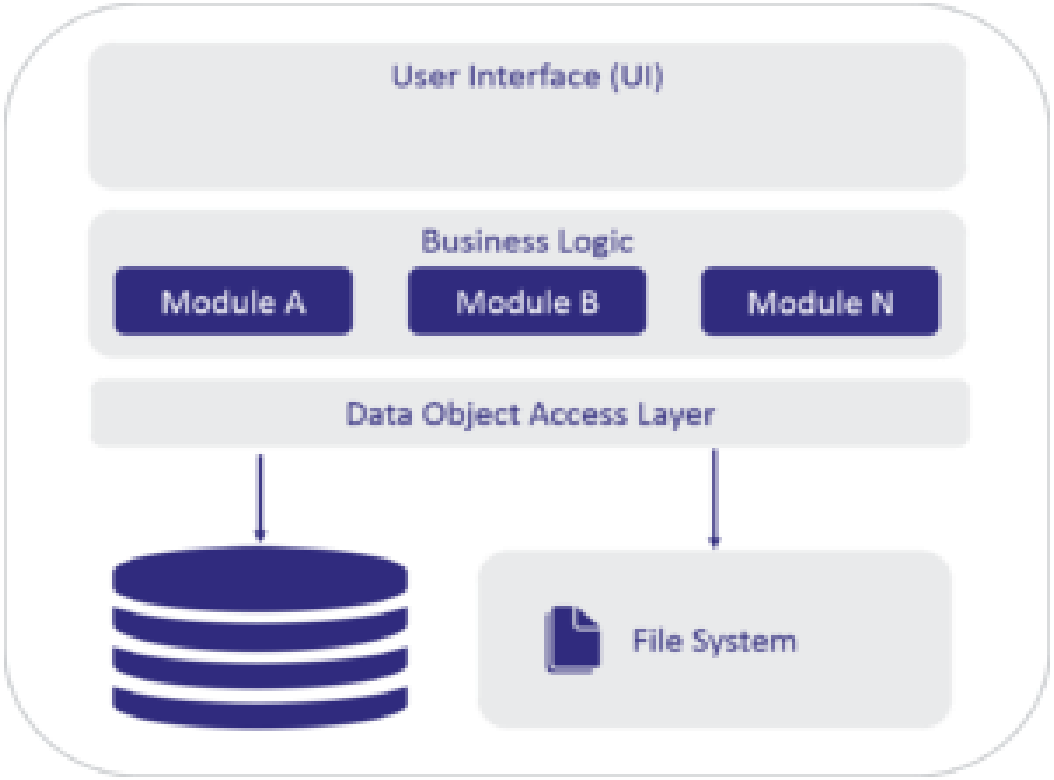
Web Application Architecture

- Client: **Progressive Web Applications (PWA)**
  - A hybrid between a site and a native mobile app
  - Merge the idea of a web app that runs in a browser with a native application installed on the device
    - Such as notifications, data synch, offline mode
  - Advantage
    - Offline performance
    - Improved user experience
  - Disadvantage
    - limited functionality

Web Application Architecture

- Server: **Monolith**
  - Built as a single unit that runs all or most of the functions
  - Advantages
    - Faster to design, develop, test and deploy
  - Disadvantages
    - Scalability
    - Reliability
    - Agility
    - Modifiability

Web Application Architecture

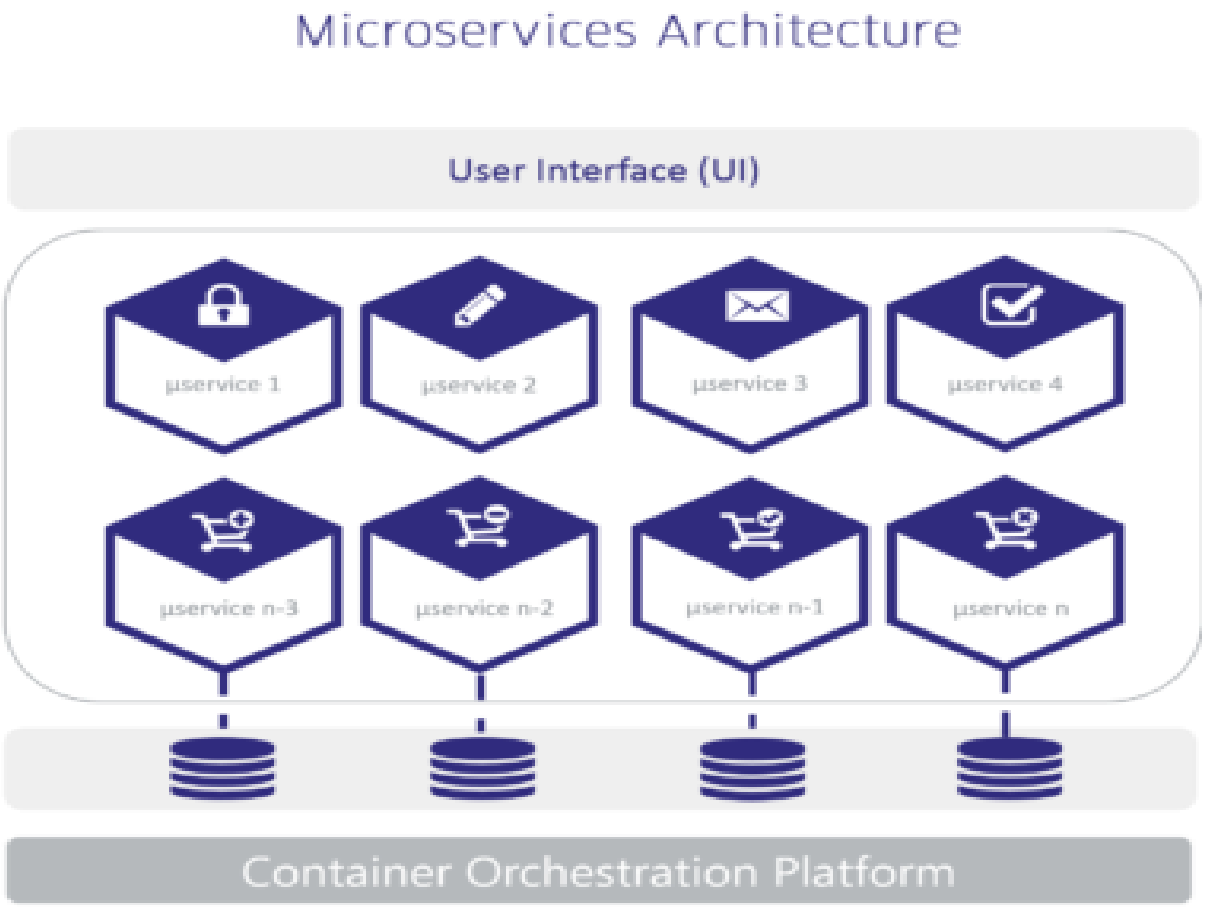


Source: https://www.sumerge.com/microservices-vs-monolithic-architecture/

Web Application Architecture

- Server: **Microservices**
  - Based on the principles of decentralization and modularity
  - Each microservice is built around a single business function and deployed independently
  - The most popular widely-applicable approaches to build highly-scalable web applications
  - Advantages
    - High reliability
    - Scalability
    - Faster time-to-market
    - Modifiable & Agile
  - Disadvantage
    - Development
    - Testing

Web Application Architecture



Source: https://www.sumerge.com/microservices-vs-monolithic-architecture/

Web Application Architecture

- Server: **Serverless**
  - Applications are developed as a set of functions
  - No need to maintain server infrastructure
    - Handled by Cloud service provides (AWS, Google Cloud,…)
  - Advantage:
    - Cost

The Web Programming Tools

- Client side:
  - HTML
  - CSS
  - JavaScript
    - React
      - https://reactjs.org/
    - Angular
      - https://angular.io
    - Vue
      - https://vuejs.org/
    - jQuery
      - https://jquery.com/
    - Bootstrap
      - https://getbootstrap.com/
    - Etc.

The Web Programming Tools

- Server side
  - Phyton – Django, Flask, …
  - Java - JSP/Servlet, Spring, …
  - Ruby - Ruby on Rails
  - PHP – Laravel, CakePHP, …
  - Node.js, Express.js, …
  - Etc.

Discussion

- What do you think the future of Web app development will be?
  - Front-end?
  - Back-end?

References

- Andrew S. Tanenbaum, David J. Wetherall **Computer Networks**. 5th ed.
- The Web and Web Standards
  - https://developer.mozilla.org/en-US/docs/Learn/Getting_started_with_the_web/The_web_and_web_standards