

Web Foundations: History, Architecture & the Web 1.0/2.0/3.0 Story

Web Technologies — Lecture 1

Masoud Hamad

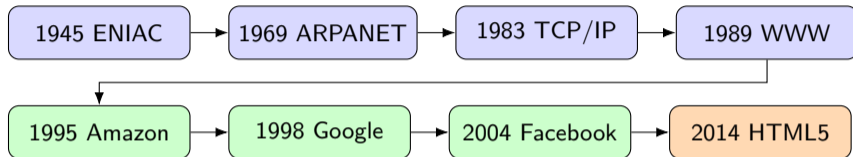
State University of Zanzibar (SUZA)

Semester II, 2025/2026

Outline

- 1 Historical Perspective
- 2 What is a Web Application?
- 3 Client–Server Model
- 4 Web 1.0, 2.0, 3.0 Architectures
- 5 n-Tier Architecture
- 6 What's Next

The Road to the Web



- **Military era** (~1945–1970) — ARPANET, packet switching
- **Business era** (~1970–1990) — TCP/IP, the Internet
- **Web 1.0** (~1990–2003) — read-only, static pages
- **Web 2.0** (~2004–2010) — read-write, social, AJAX
- **Web 3.0** (~2010–) — intelligent web, semantic data, AI

Internet vs Web

Internet

Global system of interconnected computer networks using the TCP/IP protocol suite.

World Wide Web (WWW)

A system of interlinked **hypermedia documents** (web pages) accessed over the Internet using **HTTP**.

The Web is one application running on top of the Internet — email, FTP, and BitTorrent are others.

Web Application

A program accessed by users **over a network**, using a **browser as the client**, consisting of client-side scripts, server-side scripts, HTML pages, and other resources — possibly spread across many servers.

Examples: webmail (Gmail), e-commerce (Amazon, Jumia), banking (CRDB online), social (Facebook, X), wikis (Wikipedia), document storage (Google Docs).

Advantages & Disadvantages

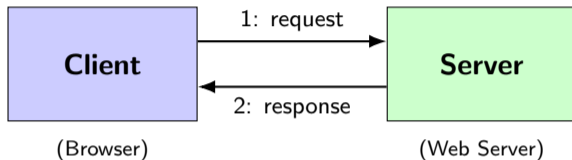
Advantages

- Browser is ubiquitous — no install
- Cross-platform automatically
- Update once, all users get the new version
- Lower IT cost

Disadvantages

- UX gap vs native apps (closing fast)
- Privacy / data sovereignty
- **Many moving parts** — hard to develop and debug
- Network dependency

Client-Server Architecture

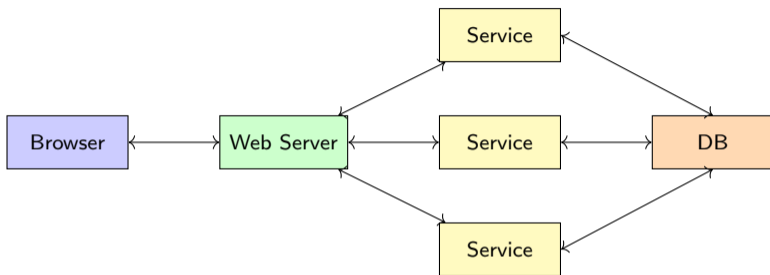


- **Server** — listens for requests, returns resources/services
- **Client** — initiates connection, requests resources
- Communication via **HTTP** (HyperText Transfer Protocol)
- This is the foundation of *every* web application

Web 1.0 — The Read-Only Web



- Server fetches **static HTML** files
- Little interactivity, no separation of data and presentation
- Browser only needs to render HTML
- ~100,000 sites, ~50,000,000 users



- Server-side functionality **partitioned** into services
- Browser is more capable (JS, CSS, multimedia)
- Web 2.0: AJAX, social, UGC, ~1B users
- Web 3.0: semantic web, ML, recommender systems, cloud

n-Tier — Separation of Concerns

Break a system into **tiers**, each with one responsibility, communicating via well-defined interfaces.

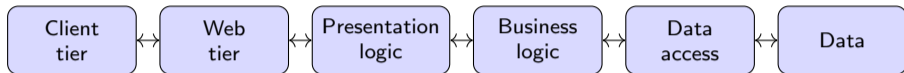
Classic 3-Tier

Presentation (UI) ↔ Application/Logic ↔ Data

Why?

- Manage complexity by abstraction
- Replace one tier without touching others
- Easier to scale, test, and maintain
- Different teams own different tiers

6-Tier Web Application



- **Client** — browser UI components
- **Web** — HTTP server (nginx, Apache)
- **Presentation logic** — server-side templates
- **Business logic** — domain rules, workflows
- **Data access** — ORM, query layer
- **Data** — database (PostgreSQL, MySQL)

Course Roadmap

- 1 **Web Foundations** (today)
- 2 Design Patterns + Agile (Scrum, Kanban, GitHub)
- 3 HTML5 + CSS3
- 4 JavaScript & the DOM
- 5 AJAX, Fetch & REST APIs
- 6 Frontend Frameworks (React)
- 7 Databases for the Web
- 8 Backend REST APIs (Spring Boot, Node, Django)
- 9 Authentication & Web Security
- 10 Deployment & DevOps (Docker, GitHub Actions)

- The Web is a **hypermedia application** on top of the Internet
- Every web app uses the **client-server** model over HTTP
- Web 1.0 → 2.0 → 3.0: from static, to interactive, to intelligent
- Modern apps are organised in **tiers** for manageable complexity

Next: Design Patterns + Agile development practices.

Questions?