

Module_6

Client/Server Systems and Multi Tier Architecture

CS 3410

Client/Server intro and background

Client/server topology has been around since the beginning of the internet content model

- One-to-connections are not feasible for web hosted applications where multiple users are connecting to the host for the same content
- This is relevant to the design and purpose of databases do the way the database must store and fetch files for the front end application
 - A database which serves a client/server application would be designed differently than one which operates with applications that utilize a single server for a single client (peer to peer)

Client/Server intro and background

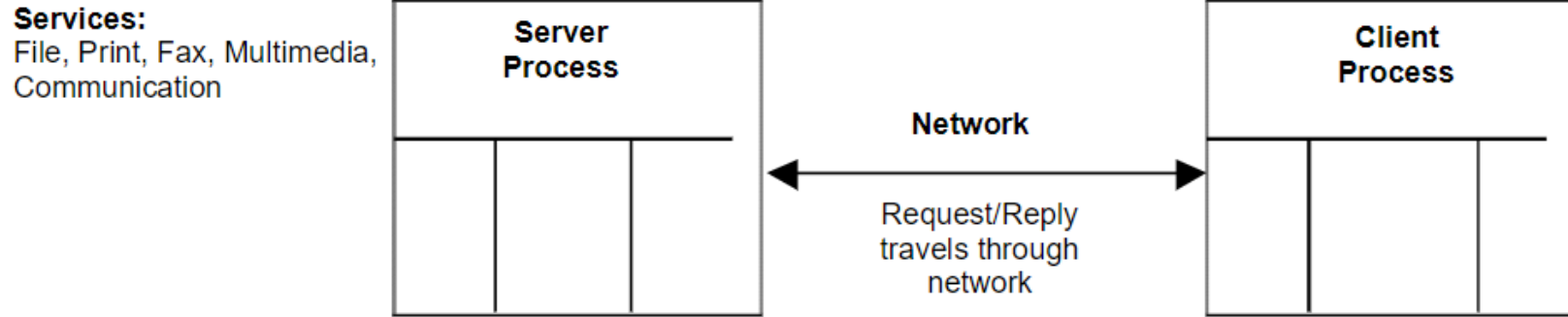


Figure 1. A graphic demonstrating the architecture of client/server systems. From “*An Introduction to Client/Server Computing*” by Yadav SC, Singh SK, 2009.

Multi Tier Architecture - Overview

- Separation and duplication of a server/database system by decentralizing data and compute resources
- Higher reliability, throughput
- No reason to not use this method of system layout
 - Considered the standard for modern computing and storage systems

Multi Tier Architecture - 7 parts

- Persistence: The database which serving application
- Accessor: The SQL server (compute)
- Logic: The applications use data on the database
- Presentation: Web browser languages such as HTML or XML.
- Requester/consumer: The web browser itself (Chrome, IE, Firefox, etc)
- Elsewhere: AWS, Azure, or GCP, or other foreign sources

Multi Tier Architecture - 7 parts

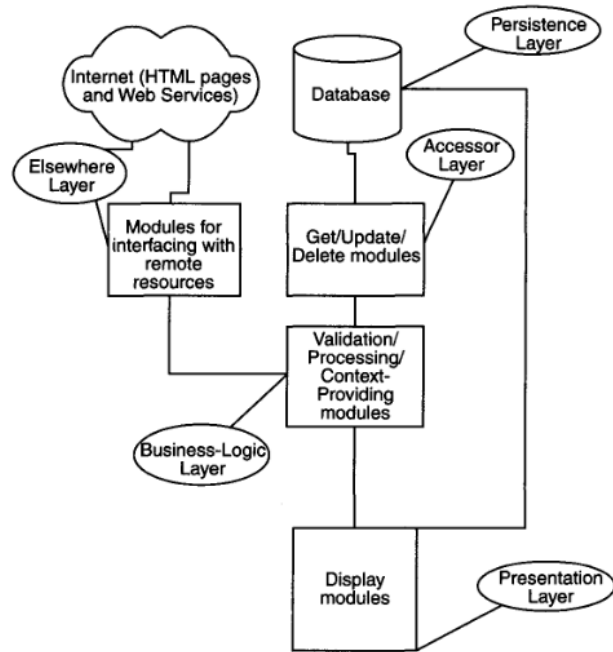


Figure 2. A graphic demonstrating the architecture of multi tier architecture. From “*Multi-Tier Application Programming with PHP : Practical Guide for Architects and Programmers*” by Wall, D, Morgan Kaufmann, 2004.

Three components of client/server systems

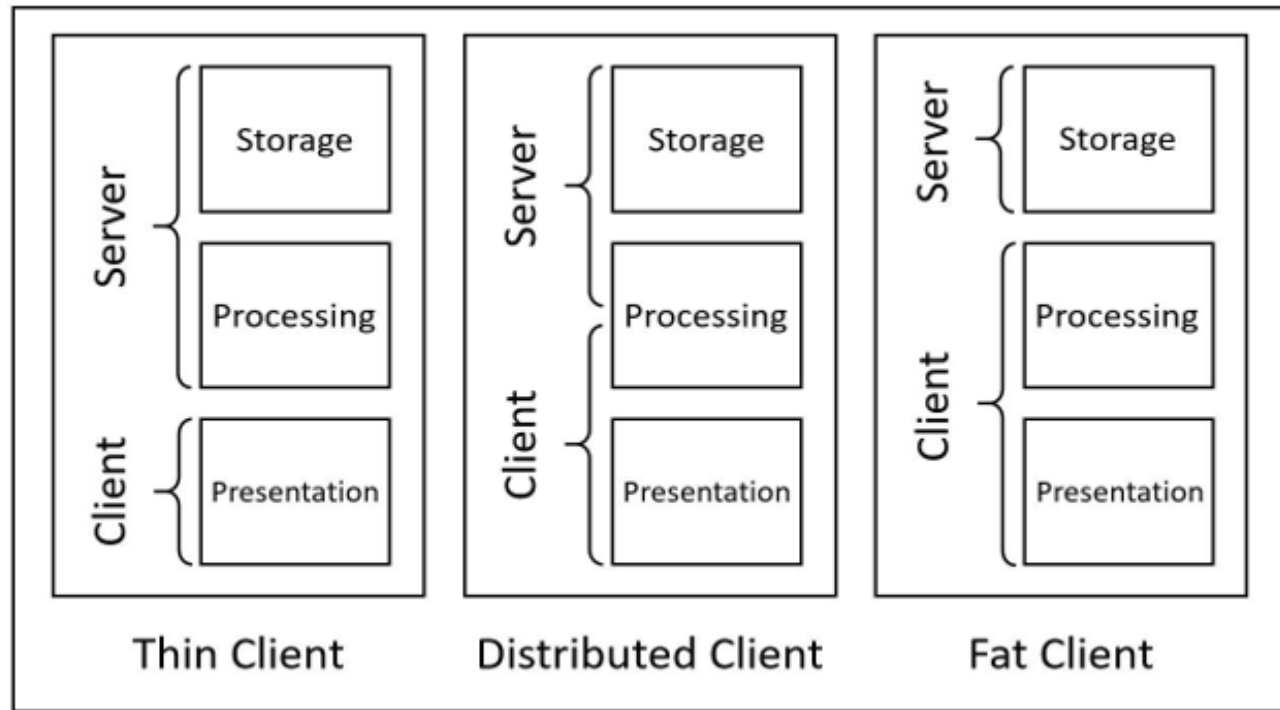


Figure 3. Client distribution models (made via Microsoft PowerPoint, Michael Nolan 2020)

3 Main Components of Client/Server Systems

- Presentation
 - High-level interactive portion of the database. Where the end-user can input requests and receive output on their host device.
 - GUI elements which the user interacts with
 - HTML
- Processing
 - The intermediate coding between the front end and the database itself
 - The “backend”
 - C, C++, Java, Python
- Storage
 - The physical database
 - The DBMS (database management system)

Two-tier vs Three-tier architecture

- In two-tier architectures, applications run on a client and communicate with the server directly
- Data layer or structure is then stored on a server
- Database as server model

Two-tier vs Three-tier architecture

- Three-tier architectures include another layer, the application layer and client layer are broken out
- Developers are able to concurrently work on each tier of the application as they are designing it
- More modern than two-tier architecture

Section Concise Summary

- The client-server model is the standard for modern web-facing servers. The core functionality of data transfer is efficient and well defined.
 - Customization is available to fit most/all application needs
 - More flexible than two-tier architecture.
- Abstraction layers serves to make the application UI as simple to use as possible, making this structure desirable to end users and developers alike.

Reference List

- Akram, Muhammad. “Qualitative & Quantitative analysis of tiered Architecture of Web-Applications.” 2007. PDF.
- Yadav SC, Singh SK. An Introduction to Client/Server Computing. New Delhi: New Age International; 2009.
- Wall, D, Morgan Kaufmann. Multi-Tier Application Programming with PHP : Practical Guide for Architects and Programmers; 2004.