Client-Server “Pattern”

Intent and Motivation

The client-server pattern describes a type of interaction between two computers programs or modules, where one is termed “the client” and the other as “the server”. The client initializes a communication cycle by requesting a service or data transformation from the server. The server listens for client requests and services requests as they are made. The client-server pattern allows for an asymmetric distribution of computing resources, leveraging the benefits of multiple computing platforms. Client applications are typically run on low resource environments such as laptops, mobile devices, or virtual machines which trade computational power for mobility or flexibility. Servers are typically high resource environments such as mainframes or parallel processing clusters; these processing-intensive nodes may host databases, web services, or complex cloud / data services. In general, servers have more powerful hardware than clients such as: higher-powered central multi processors, disk drives in the Terabyte scale and more memory as compared to the clients. Additionally, the client-server pattern may be combined in a tiered approach where the server may complete client requests by acting as a client and making resource requests to servers in other tiers. By relying on the exchange and transformation of information by a server, a client application can appear to complete computations far more difficult than would ordinarily be possible on the client platform. This pattern relies on a computer network data protocol to exchange data requests and responses.

Applicability

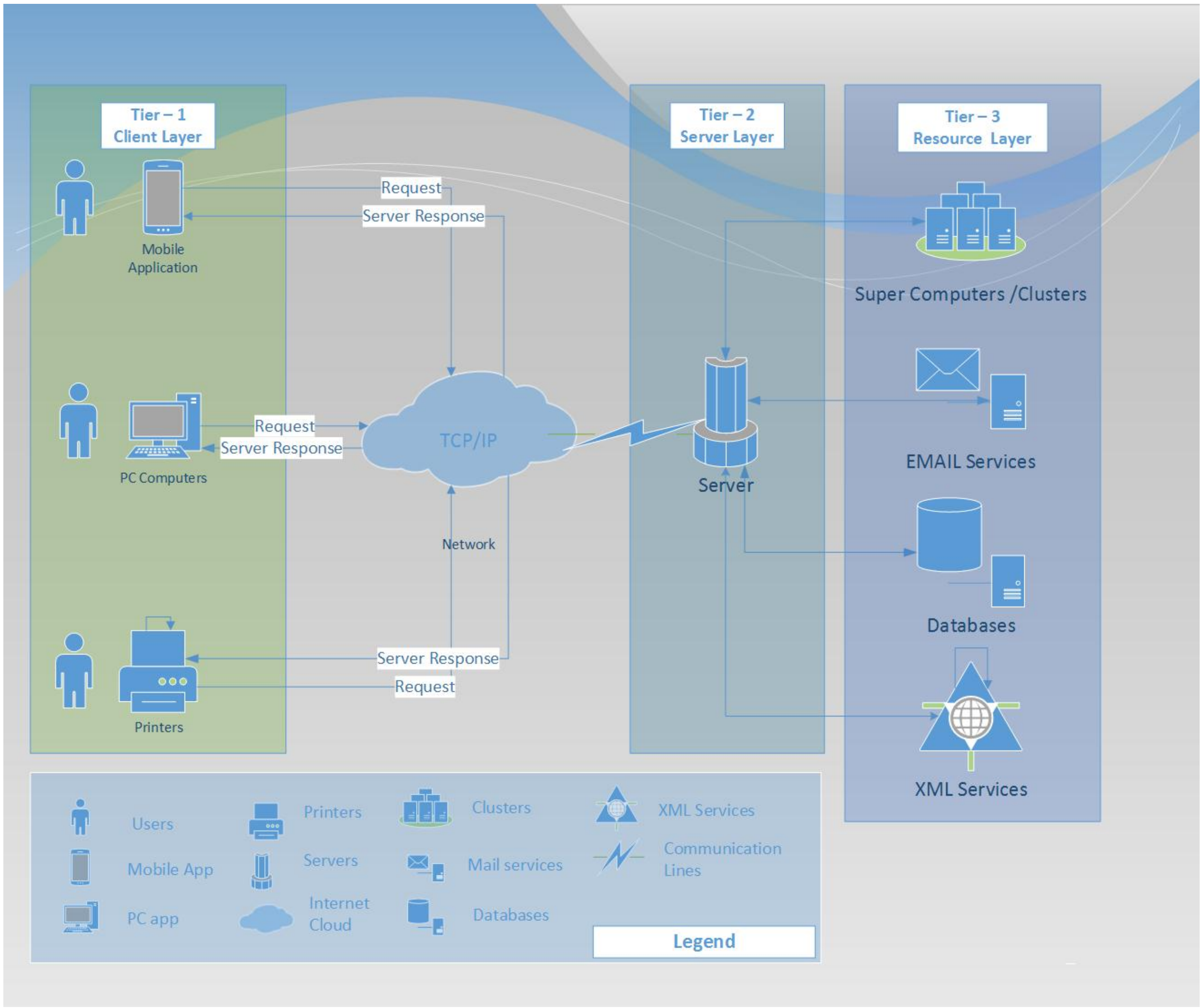
Advantages:

- *Parallelism* - Server is able to handle multiple clients in parallel
- *Flexibility* - Server can manage different types of clients (e.g., different browsers)
- *Reliability* - Infrastructure reliability is possible using redundant servers and back ups

Disadvantages:

- Relies on a network infrastructure to function properly
- May be more exposed to some security issues
- Multiple request at the same time may cause lag in the whole system
- Server is a single point of failure – clients require a working server to transform data

Structure



In tiered client-server architecture, a system element may act as both a server and a client. This is the case for the Tier 2 Server layer in the diagram above.

Known Uses

The client-server model is the architecture behind our emails, webpages, databases requests and mobile applications. All this applications provide us with a local client front-end that communicates with a server over a network which is waiting for client's request.

Client Application	*Server Application*	*Description Application*
HTML-Browsers – Chrome, Firefox	Webserver- e.g. Apache	This server runs a little different than the standard client-server architecture. The main webserver delivers requested information to client browsers via the internet but browsers may invoke third party software that is needed to transform and display some information that is transmitted from the server to the client. An example of this would be Adobe Flash video. This video can served from a webserver but the client also needs a special non-client/server application (Adobe Flash player) for the information to be displayed to the user.
Mobile applications	Google Store	This server transfers binary application data to mobile/handheld devices. The client device requests descriptions of applications and binary application data. The client displays the transferred information and potentially stores executable application data for the user.
Online-Games- WOW	Game Service Services- Steam framework	With online games, there is a distributed architecture on the server which is known as the Coliseums. This is the design and implementation protocol that is used on the server that is used to predict different checkpoints that may be accomplished by the client also known as the player.
Mail Applications- Gmail , Outlook	Mail servers	Mail servers store, relay, receive and deliver mail to client mail applications. This is done via different mail protocols. Clients make requests for sending mail via the Simple Mail Transfer Protocol. Clients make requests to receive mail with the Post Office Protocol. An alternative to

		the POP protocol is called the Internet Message Access Protocol.
Database applications-Banking, etc…	Data Servers/ Cloud Servers	The server is used to store the shared data so multiple clients can connect and add information to the database. This makes the information consistent across all of the client computers in the model.
FTP-Client e.g Filezilla client	Ftp Server e.g Filezilla server	Filezilla Server, Interface used for the remote client to connect and perform file transfers in both directions.