

[Next](#) [Up](#) [Previous](#) [Contents](#)

**Next:** [One Tier Architecture](#) **Up:** [Architectures for Distributed Computing](#) **Previous:** [Architectures for Distributed Computing](#) [Contents](#)

## Early approaches to Distributed Applications

According to George Schussel [\[Sch95\]](#), the following major approaches have been made:

### **Mainframe architecture:**

With mainframe software architectures all intelligence is within the central host computer. Users interact with the host through a terminal that captures keystrokes and sends that information to the host. Mainframe software architectures are not tied to a hardware platform. User interaction can be done using PCs and UNIX workstations.

### **File-sharing architecture:**

The original PC networks were based on file sharing architectures, where the server downloads files from the shared location to the desktop environment. The requested user job is then run (including logic and data) in the desktop environment. File sharing architectures work if shared usage is low, update contention is low, and the volume of data to be transferred is low. In the 1990s, PC LAN (local area network) computing changed because the capacity of the file sharing was strained as the number of online users grew (it can only satisfy about 12 users simultaneously) and graphical user interfaces (GUIs) became popular (making mainframe and terminal displays appear out of date). PCs are now being used in client/server architectures.

The *Client/server architecture* emerged as a result of the limitations of file sharing architectures. This approach introduced a database server to replace the file server. Using a relational database system (RDBS), user queries could be answered directly. The client/server architecture reduced network traffic by providing a query response rather than total file transfer. It improves multi-user updating through a GUI front end to a shared database. In client/server architectures, Remote Procedure Calls (RPCs) or standard query language (SQL) statements are typically used to communicate between the client and the server [\[Sch95\]](#).

For reasons of classifying the client/server-architectures, it has to be stated that most applications are made up of three fundamental types of application components [\[Ass96\]](#):

- A *presentation component* contains the logic which presents information to an external source and obtains input from that source. The presentation logic generally provides menus of options to allow the user to navigate through the different parts of the application, and it manipulates the input and output fields on the display device. Frequently the presentation component also performs a limited amount of input data validation.
- A *business component* obtains the application logic which governs the business function and processes are invoked either by a presentation component when a user requests an option, or by another business function. The business functions generally perform some type of data manipulation.
- A *data access component* contains the logic which interfaces either with a data storage system such as database systems or hierarchical file systems, or with some other type of external data source as a data feed or an external application system. Data access functions are generally invoked by a business function, although in simple applications they may be invoked directly by a presentation component.

When all three components are combined into a tightly integrated single executable program on a single machine, co-resident with its associated data files, the resulting architecture is called *one tier architecture*.

---

[Next](#) [Up](#) [Previous](#) [Contents](#)

**Next:** [One Tier Architecture](#) **Up:** [Architectures for Distributed Computing](#) **Previous:** [Architectures for Distributed Computing](#) [Contents](#)