**Application Layer**

An **application layer** is an [abstraction layer](https://en.wikipedia.org/wiki/Abstraction_layer) that specifies the shared [communications protocols](https://en.wikipedia.org/wiki/Communications_protocol) and [interface](https://en.wikipedia.org/wiki/Interface_(computing)) methods used by [hosts](https://en.wikipedia.org/wiki/Host_(network)) in a communications network. The application layer abstraction is used in both of the standard models of [computer networking](https://en.wikipedia.org/wiki/Computer_networking): the [Internet Protocol Suite](https://en.wikipedia.org/wiki/Internet_Protocol_Suite) (TCP/IP) and the [OSI model](https://en.wikipedia.org/wiki/OSI_model). Although both models use the same term for their respective highest level layer, the detailed definitions and purposes are different.

## **Application layer protocols**

The IETF definition document for the application layer in the Internet Protocol Suite is [RFC 1123](https://tools.ietf.org/html/rfc1123). It provided an initial set of protocols that covered the major aspects of functionality of the early [Internet](https://en.wikipedia.org/wiki/Internet).

* Remote login to hosts: [Telnet](https://en.wikipedia.org/wiki/Telnet)
* File transfer: [File Transfer Protocol](https://en.wikipedia.org/wiki/File_Transfer_Protocol) (FTP), [Trivial File Transfer Protocol](https://en.wikipedia.org/wiki/Trivial_File_Transfer_Protocol) (TFTP)
* Electronic mail transport: [Simple Mail Transfer Protocol](https://en.wikipedia.org/wiki/Simple_Mail_Transfer_Protocol) (SMTP)
* Networking support: [Domain Name System](https://en.wikipedia.org/wiki/Domain_Name_System) (DNS)
* Host initialization: [BOOTP](https://en.wikipedia.org/wiki/BOOTP)
* Remote host management: [Simple Network Management Protocol](https://en.wikipedia.org/wiki/Simple_Network_Management_Protocol) (SNMP), [Common Management Information Protocol](https://en.wikipedia.org/wiki/Common_Management_Information_Protocol) over.

**Design patterns for application layer protocols**

There are commonly reoccurring problems that occur in the design and implementation of communication protocols that can be addressed by patterns from several different pattern languages: *Pattern Language for Application-level Communication Protocols (CommDP)*, *Service Design Patterns*, *Patterns of Enterprise Application Architecture*, and *Pattern-Oriented Software Architecture: A Pattern Language for Distributed Computing*. The first of these pattern languages focuses on the design of protocols and not their implementations. The others address issues in either both areas or just the latter.

## **What is a network-based application?**

When we speak of **applications**, we simply mean the things a computer or other tool can be used for. You are familiar with many applications, like word processing, email, Web browsing, and spreadsheet processing.

We can distinguish between **network** and **stand-alone** applications. For example, if you use Microsoft Word to write a letter and save it on your PC, both the program and the data are stored on your computer. Since your computer does not have to be connected to a network, this is an example of a stand-alone application.

In this class, our focus is on network applications -- applications in which *either the program you are using or the data you are working with or both* reside on a network (often, but not always, the Internet).

Network applications use a **client-server** architecture, where the **client** and **server** are two computers connected to the network. The server is programmed to provide some **service** to the client.

The client is typically a desktop, laptop or portable device like an Apple iPhone. The server can be any of these, but is typically a computer in a data center.

In most (though not all) network applications, the client computer runs a **Web client** program like Firefox or Internet Explorer, and the server runs a Web server program like Apache or Internet Information Server. Shared data would be stored on the server or a computer it could access.

Note that the **user** of a network application might be a computer, not a person. For example, a computer might query the server in a vending machine, checking to see if it was low on Doctor Pepper or Coke. Search engine companies like Google run client programs that constantly scan the Web, checking for new pages which can be indexed.

For eg-  
[Eggtimer](http://e.ggtimer.com/) is a simple application where the program is stored on the network and downloaded each time it is used. Go to <http://e.ggtimer.com/5> for a five-second timer or go to [the home page](http://e.ggtimer.com/) for other options.