

## **General Architecture of Web Browser**

The basic web architecture is two-tiered and characterized by a web client that displays information content and a web server that transfers information to the client. This architecture depends on three key standards: HTML for encoding document content, URLs for naming remote information objects in a global namespace, and HTTP for staging the transfer.

- **HyperText Markup Language (HTML)** - the common representation language for hypertext documents on the Web. HTML files are viewed using a WWW client *browser* (software), the primary user interface to the Web. HTML allows for embedding of images, sounds, video streams, form fields and simple text formatting.
- **Universal Resource Identifier (URI)** - There are two types of URIs, Universal Resource Names (URN) and the Universal Resource Locators (URL). URLs are location dependent and contain four distinct parts: the protocol type, the machine name, the directory path and the file name.
- **HyperText Transfer Protocol (HTTP)** - an application-level network protocol for the WWW. HTTP sets up a new connection for each request, which is not desirable for situations requiring sessions or transactions.

Unlike old web browsers, which support only plain HTML files, today's web browsers are large complex software systems. Current browsers, like Mozilla, are equipped with a fully integrated mail and news reader, a composer that allows a user to create web pages as easily as creating a MS Word document, and support for languages and standards that enable users to interact with a web page.

The top-level conceptual architecture for Mozilla consists of five subsystems. The subsystems and the relationships are illustrated in Figure 1. The arrows in the diagram represent conceptual dependencies, a dependency being a functional relationship. For instance, we say that the User Component subsystem depends on the Network Interface subsystem if the Network Interface provides some service to the User Component.

- The *User Component* Subsystem: This subsystem contains the various user programs that are packaged with the Mozilla source code. The User Component subsystem depends on the Network Interface to establish connection to a remote machine and retrieve the requested file. It depends on the Parser subsystem to parse modified HTML file, and parse mail messages that have embedded HTML.

- The *Parser* Subsystem: This subsystem is responsible for parsing the contents of request files. The parser recognizes HTML files, XML files, and Javascript files. The Parser subsystem depends on the Layout subsystem to determine the orientation of a web page.

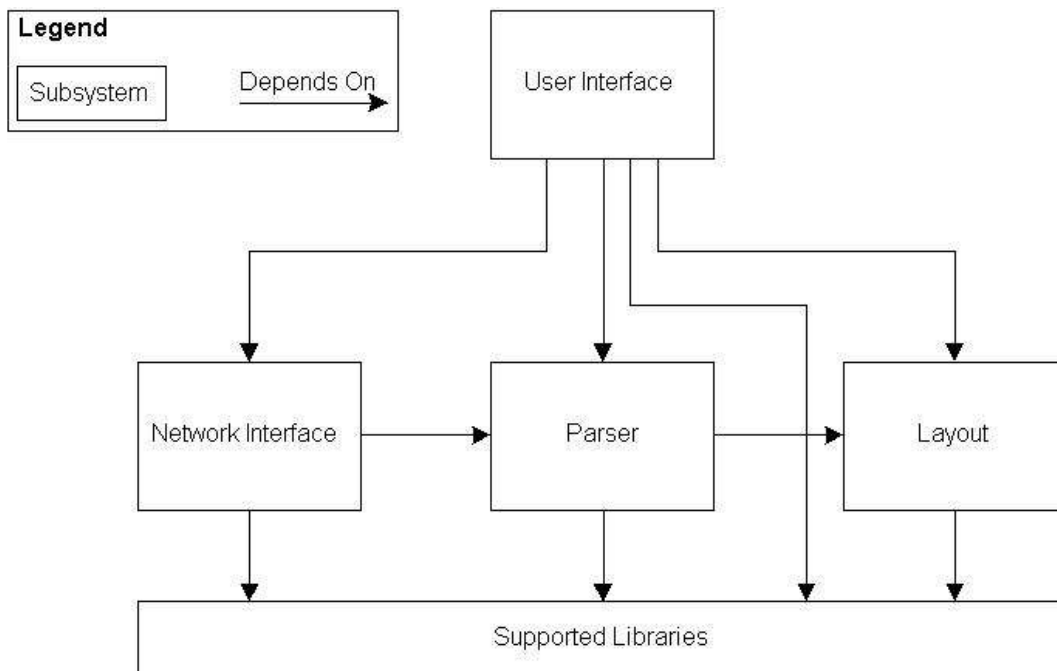


Figure 1: Top-level conceptual architecture for the Mozilla web browser.

- The *Layout* Subsystem: This subsystem handles the presentation of the user components (*i.e.*, browser, composer, mail reader, *etc.*). It also organizes and renders the contents of a web page.
- The *Network Interface* Subsystem: This subsystem handles the data flow across the Internet. It also maintains the cache and the cookies. The Network Interface subsystem depends on the Parser subsystem to parse the input file. Files from the

Internet are received as bytes. The Network Interface does not distinguish an HTML file from a Java applet.

- The *Support Library* Subsystem: This subsystem contains the C runtime library, which provides the interface to platform specific system calls, and the cross-platform component object model (COM) objects. COM allows applications to be built from binary components.