

## **CHAPTER 1**

### **WEB APPLICATION OVERVIEW**

#### **1.1 INTRODUCTION**

Software application evolved from character mode terminal applications to desktop applications, where the application logic and data were on the same workstation. Then, with the spread of LAN we had the advent of client/server applications that are network enabled desktop applications and now, with world wide diffusion of Internet, we have the web applications.

As a consequence, the client/server applications gave an impulse to the distributed computing in which a distributed program uses a computer network to achieve a common goal. While the client/server applications connect to the server over a network using full-duplex network connections, with which they have access to server data almost in real-time mode, depending on the network latency, the web applications connect to the server using HTTP protocol by a stateless request and response virtual circuit. With this half-duplex connectivity, web applications take a step back in terms of interactivity. This gap has been covered using numerous innovations from Dynamic HTML to AJAX and now with the HTML 5 WebSockets.

Our everyday life would not be the same without the features provided by the World Wide Web<sup>1</sup>, often abbreviated as WWW. Businesses use the potential of the web and employ it for e-commerce, providing an on-line medium for buying and selling goods interactively. With the new trend arisen in the on-line world, often referred to as Web 2.0, the on-line contents are updated using incorporated applications that support user-generated contents, on-line communities and collaborative mechanisms. In these new generation applications, information flows directly on the web by the surfers. Site visitors add information of their own, ranging from reviews and ratings for movies, music, and book to personal journals. These journals go by the name of blogs (short for “web logs”) and the whole blogging movement has resurrected the idea of the personal web page.

The evolution of web application is always a work in progress. In fact after the Web 2.0 and AJAX, there are already new developments such as HTML 5 and the Semantic Web. For this reason it is important to understand the underlying technology, the architecture and the model of web applications. Otherwise it would be difficult to follow the continuous evolution of the web world.

---

<sup>1</sup> The *World Wide Web*, abbreviated as WWW and commonly known as *The Web*, is a system of interlinked hypertext documents contained on the Internet. The *Internet* is a global system of interconnected computer networks that use the standard Internet Protocol Suite (TCP/IP).

## **1.2 FROM STATIC TO DYNAMIC WEB SITE**

The early web sites were only sets of web pages branching hierarchically from a home page and connected through hypertext links. These web pages:

- maintain thematic consistency of content;
- have a common look and feel. This means that they utilize a common style such as page layout, graphic design, and typographical elements;
- have well-organized interconnections in a manner to facilitate site navigation.

Moreover these web pages and all related resources are delivered to the users as they were filed on the web server. For this reason these web sites are defined static.

With the introduction of CGI technology, web pages are dynamically generated by the web server using a CGI script. This means the advent of dynamic web. In the dynamic web site, the web server have more work to do because information services are generated dynamically often querying a relational database. This evolution step marks the birth of the web application.

### 1.3 WEB APPLICATION

A web application is defined as an application program that runs on the Internet or corporate intranets and extranets. Practically the user of a web application uses a Web browser on a client computer to run a program residing on the server.

We can identify three types of web applications: static, simple interactive and complex web-based.

Static Web applications do not interact or exchange information with their viewers. Their purpose is to share and distribute information to the public.

Simple interactive web applications use response forms to collect feedback or customer evaluation on their products or services.

Complex web applications handle sophisticated business transactions online, such as online banking, stock trading, and interactive database queries. These are the cornerstone technology for e-commerce.

The most common structure of a web application is composed by three tiers as shown in figure 1.1.

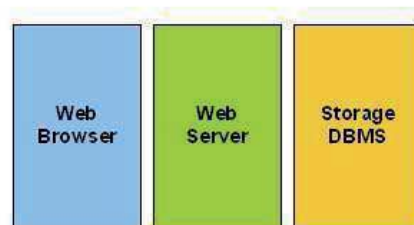


Figure 1.1 – General web application architecture.

These three tiers are called presentation, application and storage, in this order:

1. a web browser is the first tier (presentation),
2. an engine to generate dynamic content technology is the middle tier (application logic),
3. and a database is the third tier (storage).

In this architecture the web browser sends requests to the middle tier which generates a response interacting with the database.

#### **1.4 AJAX-ENABLED WEB APPLICATION**

AJAX (an acronym that stands for Asynchronous JavaScript and XMLHttpRequest) represents a new paradigm for conceiving and developing web applications. With AJAX, web applications can retrieve data from the server asynchronously in the background without interfering with the display and behavior of the existing page. These data are usually retrieved using the XMLHttpRequest object.

Despite the name AJAX, the use of XML is not actually required, nor requests have to be asynchronous.

However the use of AJAX techniques has led to an increase in interactive or dynamic interfaces on web pages. This paradigm will be explained in details in the chapter 4.

### **1.5 RIA (RICH INTERNET APPLICATION)**

The Rich Internet Applications reflect the gradual but inevitable transition of Web applications from the simple thin-client model of a traditional web browser to a richer model that behaves more like a desktop in a client/server model. The Web was originally intended to help researchers to share documents as static pages of linked text formatted in HTML. From there, web pages quickly evolved to include complex structures of text and graphics and to integrate plug-in programs to play audio and video files or to stream multimedia content. Recently, web applications evolve to a new model known as Rich Internet Application (RIA), which is a cross between Web applications and traditional desktop applications, transferring some of the processing to the client and keeping (some of) the processing on the web server.

RIAs are web-based applications which the following characteristics:

- function almost as traditional desktop applications;
- typically are delivered via the Internet to the browser;
- may require additional software in the browser (such as ActiveX, Java Applets, Flash, etc...), but they do not require any software installation.

RIAs introduce an intermediate layer of logic - a client-side engine - between the user and the web server. Downloaded at the start of the session, this client-side engine handles display, changes and communicates with the server. So in the RIA world, you really have two layers of MVC<sup>2</sup>. There is an MVC on the client and an MVC on the web server as well.

The MVC ON THE CLIENT manages the interaction between the user and the interface, handles all requests to the server for data, and controls how the data is presented in the view.

The MVC ON THE SERVER handles requests from the client and delegates actions on the server. Differently from the client, here there is no user interface. Instead of a user interface, the view would be the format of the data that is being returned to the client application.

## 1.6 WEB APPLICATION AND SESSION STATE

We cannot speak of a web application without considering the session state. Before diving into the various technologies behind a web application, it is necessary to explain what a session is. The mechanisms used to implement and manage the session control will be explained in the chapter 5.

The word “session” is used in at several levels and places so in order to avoid confusion we say what a web application session is not and then we define what it is.

---

<sup>2</sup>THE SOFTWARE ARCHITECTURE OF A RIA: Model-View-Controller (MVC) is a software architectural pattern where an application is broken into separate layers:

- MODEL: is the domain-specific representation of the data on which the application operates.
- VIEW: renders the model into a form suitable for interaction, typically a user interface element. Multiple views can exist for a single model for different purposes.
- CONTROLLER: processes and responds to events (typically user actions) and may indirectly invoke changes on the model.

A web application session is not a TCP session, which is a TCP virtual circuit that establishes a point-to-point communication between two hosts on the network using the TCP connection-oriented protocol.

A web application session is not a HTTP session which is a single request-response exchange. HTTP protocol handles each request to the HTTP server by:

- 1) opening a connection with the web server over a TCP session;
- 2) downloading the web document;
- 3) dropping the connection.

in order to decrease the transfer time in the last version 1.1 of HTTP protocol, an HTTP session has changed in:

- opening a connection on a TCP virtual circuit that may be kept open instead of closed;
- requesting from the same browser may reuse this connection instead of starting another one;
- closing the connection after a short period of inactivity e.g. 30 seconds.

A web application session is not a Browser session which lasts for as long as the browser program is running, while HTTP session normally “time out” after a period of inactivity according to the configurations on the web server.

*A web application session is a sequence of HTTP sessions which are connected together using some piece of information (token) and treated as a single interaction. This sequence is associated to one user and it is made from one browser to one or more web sites.*

Moreover when a web application session ends it leaves the state of the web application correctly consistent with its specifications.



## 1.7 SESSION STATE MODELS

Software applications use and maintain the state to drive the interactions with the user. We mean for “state” all the information that is necessary to permit to a user to interact consistently with a software application.

A *distributed application* is software that executes on two or more computers in a network. It is made up of two parts the '*front-end*' and the '*back-end*'. The first one runs on one computer while the second one runs on one or more suitably equipped server computers.

A *web application* is a distributed application which is expected to maintain the state. A typical web application is the shopping cart, where the server is expected to keep a list of items in the cart, and to present this list on demand.

According the aforementioned scenario we now consider various possible models of storing the state in a web application considered as distributed client-server application. This topic will be better covered throughout this book starting from the chapter 5.

### Stateless Server

In the “*stateless server model*” server doesn't maintain any state information of each active client. The amount of exchanged data is high and as a consequence the response time is long because each client keeps all the state.

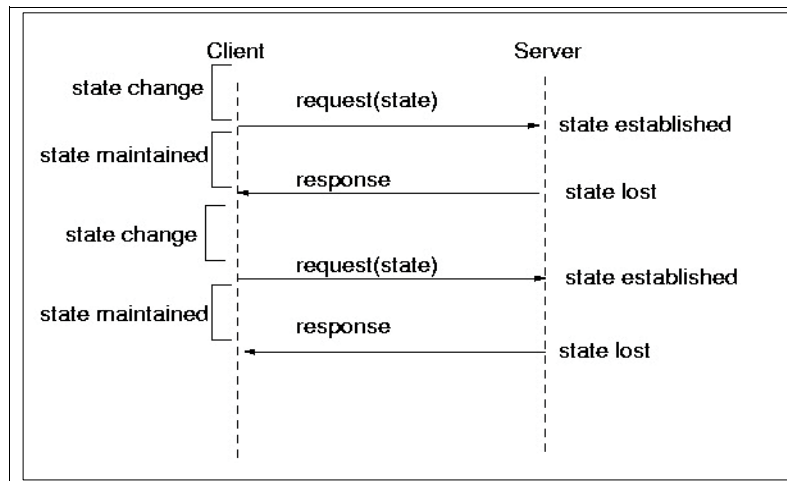


Figure 1.2 – Session Management: Stateless server

Distributed State

In the “*distributed state model*”, the client of a web application generally uses the minimum data (a token like a Session ID) to identify the session.

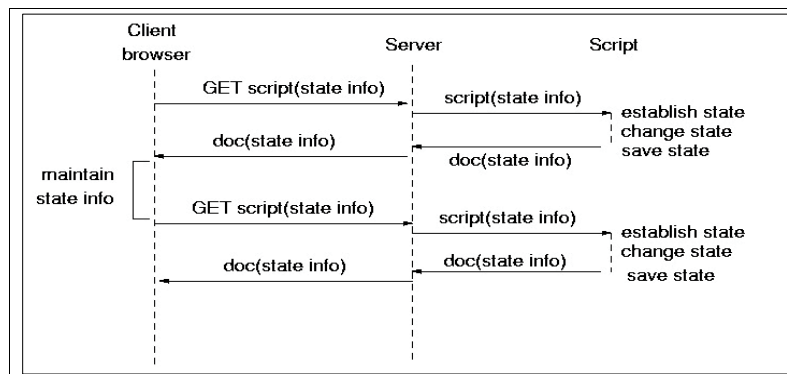


Figure 1.3 – Web Application Session Management

These models are only an idea of the state management of a web application before diving into more analytic aspects related to its operation on the Internet infrastructure.

## **Bibliography**

- [1.01] Leon Shklar, Rich Rosen, *Web Application Architecture: Principles, Protocols and Practices*, Second Edition John Wiley & Sons Ltd, 2009;
- [1.02] Vito Roberto, Marco Frails, Alessio Gugliotta, Paolo Omero, *Introduzione alle Tecnologie Web*, McGraw-Hill, 2005;
- [1.03] Wikipedia, *World Wide Web*,  
[http://en.wikipedia.org/wiki/World\\_Wide\\_Web](http://en.wikipedia.org/wiki/World_Wide_Web);
- [1.04] Wikipedia, *Web Application*,  
[http://en.wikipedia.org/wiki/Web\\_application](http://en.wikipedia.org/wiki/Web_application);