



# UNIT 1.

## INTRODUCTION TO WEB ARCHITECTURES

Web Applications Deployment  
CFGS DAW

Author: Carlos Cacho

Reviewed by: Lionel Tarazón  
[lionel.tarazon@ceedcv.es](mailto:lionel.tarazon@ceedcv.es)

## License



**Attribution-NonCommercial-ShareAlike (by-nc-sa)** You must give appropriate credit, provide a link to the license, and indicate if changes were made. You may not use the material for commercial purposes. If you remix, transform, or build upon the material, you must distribute your contributions under the same license as the original. This work is a derivative of the original work created by Carlos Cacho.

## Nomenclature

During this unit we are going to use special symbols to distinct some important elements.

This symbols are:



Important



Attention



Interesting

# INDEX

<b>1.Introduction.....</b>	<b>4</b>
<b>2.Web architecture features.....</b>	<b>4</b>
2.1 Scalability.....	4
2.1.1 Vertical scalability.....	5
2.1.2 Horizontal scalability.....	5
2.1.3 Servers cluster.....	5
2.2 Portability.....	5
2.3 Design patterns.....	5
2.4 Responsibility separation.....	6
2.5 User's session management.....	6
<b>3.Web architecture models.....</b>	<b>7</b>
3.1 Client/Server Model.....	7
<b>4.Web servers.....</b>	<b>9</b>
<b>5.How to have a web server.....</b>	<b>9</b>
<b>6.Web hosting Control Panels.....</b>	<b>10</b>
6.1 Webmin.....	10
<b>7.Bibliography.....</b>	<b>11</b>

## UT01. INTRODUCTION TO WEB ARCHITECTURES

### 1. INTRODUCTION

When a user needs to execute an application over Internet make use of a **web application**. Typically, it has at least two parts: one in the server side, where the information or data that the user needs are, and the other is executed in a web browser in the client machine.

Some examples of web applications are e-mail, social networks or online shops.

This web application needs a structure to be accessible from different devices. This structure is known as **web architecture**.

So, a web architecture is the design of a web application's structure, including study, analysis, organization, disposition and structure of the information of web pages.

Most of the current web architectures are based on the client/server model. We will explain it in section 3.

### 2. WEB ARCHITECTURE FEATURES

The main web architecture features are:

- Scalability
- Portability
- Design Patterns
- Responsibility separation
- User's session management

#### 2.1 Scalability

An application is scalable when we can increase the data number and the data size. It is important to know that scalability and efficiency is not the same. A web application can have a very good performance but not be scalable.

There are three formats:

### 2.1.1 Vertical scalability

This is the easiest strategy which consists in improving the hardware to increase the performance. The main advantage is that the web application does not need any change, you only have to get a better (and more expensive) machine. Some problems are that the budget we have is the limit to improve the hardware or the nonexistence of a better machine in the market.

### 2.1.2 Horizontal scalability

This strategy implies to add more machines, increasing in this way the capacity. In this case, although the machines we add do not need to have expensive hardware, the web application has to have some changes.

To use it, we must use a web architecture. The most used is **shared nothing**, where every node (or machine) is independent, self-sufficient and autocontent. This means that the application is shared in all the system, it is not stored in the nodes, so if a node fails we will not lose the information and the system will keep working.

Google uses this strategy.

### 2.1.3 Servers cluster

A **cluster** consists of two or more computers working together to provide a higher level of availability, reliability, and scalability than can be obtained by using a single computer.

So, a **server cluster** is a group of independent servers working together as a single system to provide high availability of services for clients. When a failure occurs on one computer in a cluster, resources are redirected and the workload is redistributed to another computer in the cluster. You can use server clusters to ensure that users have constant access to important server-based resources.

## 2.2 Portability

Portability refers to the capacity of a web application to execute it in different platforms or architectures with the minimum modifications.

A web application has to be able to adapt to different possible architectures.

## 2.3 Design patterns

They are general reusable solutions to a commonly occurring problem in the web applications development. They allow the communication between the analyst and the developer, helping to its development and its maintenance.

An example of pattern is the “creational” one. It is the initialization and configuration skeleton for a Model-View-Controller (MVC).

The **MVC** is a software architectural pattern for implementing user interfaces on computers. It divides the application into three layers:

- The model manages the data, logic and rules of the application.
- The view is the representation of the information: charts, graphics, diagrams, etc.

-The controller accepts inputs and converts them to commands for the other two layers.

## 2.4 Responsibility separation

It means that different responsibilities must go to different layers.

Nowadays, the trend is to separate as much as possible the responsibilities in layers, so each layer works in a part of the application. In particular, the most used web architectures is the 3-independent layers:

- Presentation** (Client): the web browser is executed in the user's machine.
- Business** (Web server): the presentation layer can access to the logic and procedures of this layer, they can be HTML files, Java Servlets, etc. Some technologies used in this layer are PHP, ASP, Perl, etc.
- Persistence** (Data): is a management system to access to the different data. Usually is a Database Management System (DBMS) but also XML files.

🔊 It is interesting to know the web applications evolution from the responsibility point of view:

### -Mainframe applications

They are monolithic systems, all the applications are inside of a high performance machine. There is only one layer. Logic and physical layers are not separate.

### -Client/server applications

These applications have two layers: one with the presentation of the information and the other with the business and the persistence of the data.

### -3-tier applications

As we said, these applications have three layers: presentation, business and persistence.

### -N-tier applications

They are known as Brown's model. They are multiplatform distributed systems which have a high modularity, scalability and flexibility. Some examples are Sun Tone or Microsoft .Net.

## 2.5 User's session management

A user session is the activity time that a user, with an IP address, spends in a web site. The number of user sessions are used to calculate the traffic of data in that application.

### 3. WEB ARCHITECTURE MODELS

As we said above, there are several web architecture models, for example, the W3C (World Wide Web Consortium) describes four different types: Message Oriented Model, Service Oriented Model, Resource Oriented Model and Policy Model.

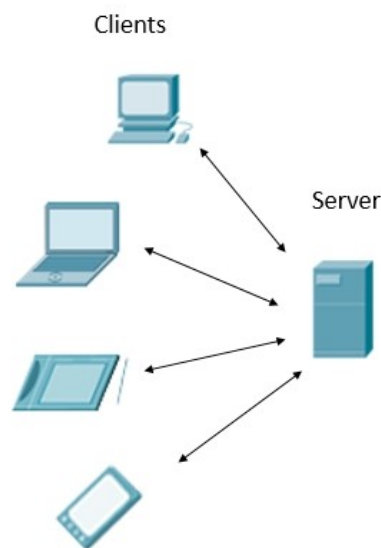
You can read more about these models in: <https://www.w3.org/TR/ws-arch/#concepts>

We are going to focus on Client/Server Model.

#### 3.1 Client/Server Model

This model works like a distributed application which distinguish providers (**servers**) from petitioners (**clients**). So a client requests some information to a server, and this answer it.

Normally, the client and the server are not in the same machine, but it is possible.



Client/Server model

A basic example is when some clients request a web page to a server. These are the steps:

1. From the client web browser, the user request a web page writing the URL (Uniform Resource Locator). For example, *http://www.ceedcv.org*
2. The server receives the request.
3. The server looks for the web page in its management system.
4. The server sends the source code of the web page to the client.
5. The client receives the source code and the web browser shows the web page.

As we said, nowadays this model is divided into three layers: Presentation, Business and Persistence.

It is interesting to know that the Business layer can be divided into two more layers:

- Presentation layer**: it is in charge of integrating the dynamic and the static part of the web page. Some solutions to do this is to use languages like ASP, JSP or PHP.

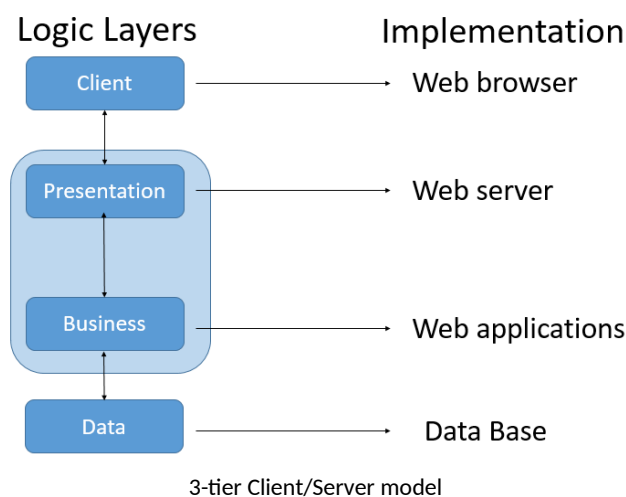


A **static web page** is delivered to the user exactly as stored. They are HTML documents stored as files in the file system and made available by the web server over HTTP.

A **dynamic web page** is a web page whose construction is controlled by an application server processing server-side scripts (PHP, ASP, etc.). It processes the web pages that are send by the client, for example the data of a form.

The **HTTP** protocol (Hyper Text Transfer Protocol) is the most used in the Internet. Its objective is to allow the files transfer between a browser and a web server through an URL address.

- Business logic layer**: it is the most complex. It controls the results of the functional analysis (it realizes all the operations and validations, manages the workflow and manages the sessions) and manages all the data access operations to the Presentation Layer, showing the set of business rules. It used to be located in an “Applications Server”.





Some **advantages** of this model are:

- Centralization of the information: the access, resources and data are controlled by the server avoiding that an unauthorized client can enter to the system and damage it.
- Scalability: you can increase the number of clients or servers separately.
- Easy maintenance: all the functions and responsibilities are distributed between some independent machines so is easy to replace, repair or update any server.
- There are many technologies based on this model.

And some **disadvantages** are:

- Management of the traffic congestion: if many clients send petitions to the same server, it can be down.
- Resource access: if a server falls down, its resources are not available.
- Dependence of hardware and software: normally we need specific software and hardware for each case.

An example of this model is a LAMP server.

## 4. WEB SERVERS

A **web server** is a program or a set of them, which offer services over a network. Its aim is to offer necessary resources in order that two or more programs could communicate independently.

The web servers belong to the system called **distributed**, which means that the system is made up of more than one machine or a set of hardware. Thus, the resources must distribute among them. The technology that takes carry of this is known as load balancing.

The **load balancing** is implemented by an algorithm that distributes the tasks in the most efficient and equitable possible way, avoiding the **bottlenecks**.

In unit 3 we will talk about Apache and Nginx, the most used web servers.

## 5. HOW TO HAVE A WEB SERVER

First of all, we need to have a powerful machine, dedicated and able to attend all the requests.

If it is not possible to have this kind of machine, the web hosting is a good solution to a reasonable price, in exchange for a specialized company that looks after our resources.

So we will need:

- A stable operating system, with security and permissions control.
- A static IP address (an IP address which never changes), to allow the remote access.
- A Domain Name Server (DNS) to accede easier to our server.

- 24 Internet Connection to be accessible from another networks.
- Specific software of web server.

In Unit 5 we will talk about web hosting.

## 6. WEB HOSTING CONTROL PANELS

Nowadays, the hosting companies allow to costumers to manage their hosted services using a web-based interface.

Some available modules are:

- Access to server logs.
- Create and configure virtual web servers.
- DNS Server settings and routing configuration.
- Details of available and used webspace and bandwidth.
- Email account configuration.
- Maintaining FTP users' accounts.
- Managing database.
- Visitor statistics using web log analysis software.
- Web-based file manager.

Here you can find a list of them and their comparison:

[https://en.wikipedia.org/wiki/Comparison\\_of\\_web\\_hosting\\_control\\_panels](https://en.wikipedia.org/wiki/Comparison_of_web_hosting_control_panels)

During this course we will use Webmin.

### 6.1 Webmin

Webmin is a program that simplifies the process of managing a Linux or Unix system. Normally you need to manually edit configuration files and run commands to create accounts, set up a web server or manage email forwarding. Webmin lets you perform these tasks through an easy to use web interface, and automatically updates all of the required configuration files for you. This makes the job of administering your system much easier.

Some of the things that you can do with Webmin are:

- Create, edit and delete Unix accounts on your system.
- Export files and directories to other systems with the NFS protocol.

- Set up Disk Quotas to control how much space users can use up with their files.
- Install, view and remove Software Packages in RPM and other formats.
- Change your system's IP address, DNS Server settings and routing configuration.
- Set up a Linux Firewall to protect your computer, or to give hosts on an internal LAN access to the Internet.
- Create and configure virtual web servers for the Apache Webserver.
- Manage databases, tables and fields in a MySQL or PostgreSQL Database Server
- Share files with Windows systems by configuring Samba Windows File Sharing.

In this unit we will learn how to install it and during the course we will configure several web hosting modules.

## 7. BIBLIOGRAPHY

- [1] Álvarez Villanueva, Cristina (2015): "Implementation of web architectures", CEEDCV
- [2] Cuesta, Sergio (2014): *Despliegue de aplicaciones web*
- [3] Gallego, Micael (2014): *Desarrollo de aplicaciones Web*.
- [4] Official documentation of Webmin: <http://doxfer.webmin.com/Webmin>