

Web design & development

Guidebook



June 5, 2019

IDM-NEGOMBO

K.P.I. Shenesh Perera

Contents

[Domain Name Servers (DNS) 2](#_Toc12382295)

[What is a Domain Name Server? 2](#_Toc12382296)

[Types of Domain Name Servers 2](#_Toc12382297)

[Domain Name Server Management & Organization 3](#_Toc12382298)

[Communication protocols 4](#_Toc12382299)

[Web server operating systems, hardware and software 5](#_Toc12382300)

[Impact of common web development technologies and tools 6](#_Toc12382301)

[Search Engine Optimization (Evidence-based) 7](#_Toc12382302)

[Front/Back-end Web Technologies: Frameworks & Tools 8](#_Toc12382303)

[How do web technologies cater to the OSI model? 8](#_Toc12382304)

[Custom built vs web creation tool built websites 9](#_Toc12382305)

[Justification, recommendation and explanation of various front and backend design and development technologies for custom built websites 9](#_Toc12382306)

# Domain Name Servers (DNS)

## What is a Domain Name Server?

The Domain Name Server is simply described as the phonebook for the globalized network between devices of all types and sizes known as the Internet. The Domain Name Server serves the purpose of translating the human-readable and comprehensible domain address to a machine-readable domain address.

Similar to how a phonebook is used to refer to the phone number of an individual. Just like how the phonebook maps the phone number to the person, the DNS maps the human readable web address to an IP address.

In terms of a phonebook, for example if you had to find the phone number of Jane, then you’d goto the J section and go through the names to find Jane and then look at the phone number.

In terms of the DNS, when you type google.com into a browser’s address bar, and hit enter an immediate request is sent to the DNS, the DNS goes through its large databases to find the domain of google.com and after it’s found, the IP address of google.com is taken and then a request (HTTP GET) to that IP address is sent.

The reason why the Domain Name Server was initially implemented was because humans found it difficult to memorize an IP address that is made up of numbers, and also this became quite inefficient as some web services/applications/sites tend to have more than 1 IP address, and each IP address is specific for a country. If it wasn’t for the DNS, the Internet as we know it, would not be globalized.

## Types of Domain Name Servers

During the whole process of loading a web page into a browser, there are 4 Domain Name Servers involved in the process, which are:

**DNS Recursive Server** - The Domain Name Recursive Server can be thought of as a librarian who is asked to go find a particular book somewhere in a library. The DNSRS is a server designed to receive web address from any machine connected to the internet through an application like a web browser. Usually but not always, the DNSRS is then held responsible for making additional requests in order to satisfy the user’s queried domain address. This usually includes some error handling.

**Root Name Server** - The RNS is the first step in resolving human readable web addresses into IP addresses. Most of the time in the process is usually spent at the Root Name Server.

**Top Level Domain Name Server** - This step of the process is not a mandatory part, think of the Top Level Domain Server as a shelf in a bookshelf that is for very special books. The TLD Name Server is usually the next step after the Root Name Server, if the web address has not already been resolved in the Root Name Server. This can also happen before going through the RNS too. Typical examples of top level domains are those that with .com or .org.

**Authoritative Name Server** – This is the final domain name server involved in the process and often the most complex in the process. Think of it as a dictionary that holds the definitions and authorization details of a web address. If and only if the Authoritative Name Server access to a particular requested web address resolution, then it will return the IP address of the requested web address, back to the Recursive Server that initially made the request from the ANS.

## Domain Name Server Management & Organization

As you may now understand, the Domain Name Server is **very important,** not only in terms of the purpose and functionality it provides to the internet but also because it is the Domain Name Server is the core of the Internet, without it most humans will find the internet and unfriendly service. As such maintaining and organizing is prioritized, although the internet is free the assignment of the free space within the cyber world of the internet is under government control.

Mainly the following organizations are responsible for the maintenance of the DNS:

* Internet Corporation for Assigned Names and Numbers (ICANN)
* Internet Assigned Numbers Authority (IANA)
* Operators of Top Level Domains (Could be any company that holds legal rights to a TLD)
* Accredited registers (Like namecheap or Godaddy)

The DNS is a hierarchical system, at the top of this hierarchy is what is known as the “root zone”. The IANA manages the DNS root zone, alongside the coporation and assistance of ICANN, this is done through the administration and analysis of the data within the root zone file in every root name server. Alongside maintaining the root zone file, ICANN also maintains what’s known as the “root zone database” which is usually what you would see if you did a simple WHOIS on a website.

ICANN also manages the Key Signing Key (KSK) which provides the security that the Domain Name Servers needs, this segment of ICANN is known as the DNSSEC. ICANN creates different policies for the management of the root zone through advice provided by two large technical bodies. The Root Server System Advisory Committee (RSSAC) and the Security and Stability Advisory Committee (SSAC).

ICANN also manages the Top Level Domains (like .com, .org., .gov) which are usually given special treatment, and also credits any and all registrars that buy and manage the free space of the internet. There are over 100s of such registrars (like NameCheap, Godaddy) who are known as 3rd party registrars that is accessible to anyone in the whole word. The global protocols for Top Level Domains are applied and maintained by another two large segments within the ICANN known as the Generic Names Supporting Organization (GNSO) and Country Code Names Supporting Organization (CCNSO).

# Communication protocols

Communication protocols are the set of rules that allow for the functionality within the cyber space for web applications that have to communicate among each other. This part of the guidebook will describe of 5 largely used and commonly used communication protocols that nearly every large platform out there in the world use which are:

* **Simple Mail Transfer Protocol (SMTP**) – SMTP is a protocol that is widely used in a lot of industrial organizations. This protocol plays the role of sending, authorizing, managing, organizing and establishing a connection in order for an electrical mail can be sent from one person to another. SMTP is widely used in any designing and functionality that involves mailing servers, most websites nowadays have newsletters, these newsletter services are done with the help of SMTP
* **Transmission Control Protocol/Internet Protocol (TCP/IP)** – The TCP/IP protocol is a large suite, it is also called the IP suite. TCP/IP brings what’s known as a session to a web application, in terms of design and purpose within a web service, sessions allow a web service to store, manage and manipulate data specific for each instance a user creates of a particular web site or web application.
* **Voice over Internet Protocol (VoIP)** – As you may see in many social media platforms, games and several other communication platforms, just like the name of this protocol suggest it is a protocol that allows transmission of audio as data packets through the internet from one user to another. In terms of design and purpose, VoIP is widely used in the industry to have conferences, security applications and even simple server side audio rendering. VoIP is a one of the protocols that has started to see large amounts of attention.
* **Hypertext Transfer Protocol (HTTP)** – The oldest and the most fundamental communication protocols of the internet, this protocol allows for transmission of data in the form of text in the internet. In terms of design and purpose, this protocol does nearly everything from assisting rendering of static and dynamic text content to maintaining the repetitive transmission of text data from one place to another.
* **User Datagram Protocol (UDP)** – The UDP protocol is similar in terms of design and purpose to the TCP/IP protocol suite, but without the sessions. There is no additional overhead to create sessions between users of the internet when it comes to UDP, it is usually used in situations where performance is a critical component and security isn’t a concern, for example streaming videos.

# Web server operating systems, hardware and software

A web server is the device that would allow for hosting your web application or your web site, in such a way that the world is capable of accessing your web site. There are a many operating systems, software and hardware that can be discussed but most of them are not industrially used.

In terms of pioneers in operating systems which are also called server system software, there are two large branches in both industrial and commercial web hosting and publishing:

* **Microsoft Windows Server** – This is the most commonly used web hosting and publishing operating system in organizations where technical expertise is lacking. Ofcourse, with this given addition to ease of use, there is a disadvantage. The resource consumption of Microsoft Windows web servers is usually very high. Most viruses, malware and malicious software target web servers that use Microsoft Windows Servers. However though, there are very competent software that can help resolve these situations, but these programs too are quite resource consuming. Most startups, companies that do not have solid technical background use web servers with the Microsoft Windows Server operating system.
* **Linux** – Mostly used by organizations that have been on the map for more than a few years and have a solid technical background. Usually organizations switch to linux based web servers for more efficient resource consumption and other concerns like security. However though, Linux based web servers usually have no graphical user interface, or in otherwords they are operated through the command line.

The following table will describe a multitude of different hardware and software that’s associated with web servers:

|  |  |
| --- | --- |
| **Web Servers** | |
| **Hardware** | **Software** |
| * **Server towers** – These are large and highly efficient computer systems that are specially built for hosting and publishing web applications or web sites. They consume large amounts of electricity, often they have more than 1 CPU, GPU or RAM cards. They go beyond just normal performance ranges, they have high amounts of resources and usually have backup power sources as a fallback procedure. * **Load balancers** – Load balancers are devices that sits between the clients and the servers. The purpose of a load balancer is to distribute traffic in a particular preconfigured pattern among large clusters of web servers, which leads to higher performance and other benefits. This reduces the resource consumption of each component of a cluster of web servers. * **Hardware Firewalls** – Hardware firewalls interpret and police internet traffic that clients send to web servers. According to a preconfigured set of inbound and outbound rules, a firewall will allow or deny web traffic from particular clients to or from a web server. * **Proxy Servers** – These devices also sit between clients and web servers, but proxy servers usually act as the intermediate or the mediator between the two ends. For particular clients, a proxy server may decide to send a different set of responses unlike another set of particular clients. | * **Nginx –** This is a multi-functional piece of software that can act as a reverse proxy, load balancer, HTTP cache and a mail proxy. It promises to accelerate security, responsiveness and overall efficiency of a web server. * **NodeJS –** A server runtime that allows web developers to write web server logic and functionality with Javascript. It is a C/C++ program that binds Javascript to allow to create server instances. * **Apache HTTP Server –** A quite legacy level web server software that allows for hosting web applications. It can act as a simple File server, mail server or a complex server that has multiple other functions. * **Mongoose –** This is a cross-platform embedded web server and networking librabry that promises to provide functions like TCP, HTTP client servers, Web Socket client and server and MQTT client/broker services. It allows literally any device out in the world that have basic processing rules and technology to act as a web server. It is an application that is quite fundamental in the industrial web hosting and publishing sphere. * **WampServer –** This is a software stack that was created by Microsoft Windows for the Microsoft Windows server operating systems, which consists of the Apache web server, Open SSL, MySQL and PHP. This is counted as an obsolete and legacy stack and is no longer widely used in the industry anymore. |

# Impact of common web development technologies and tools

There are a plenty of web development technologies and tools out there, the main purpose of these technologies or tools is to ease the process of creating a custom built web application or website.

Industrially such technologies are used for the following reasons:

* Performance
* Security
* Efficiency
* Ease of implementation
* Validity
* Higher Exposure

Using technologies or tools like this ensure the security of the web application or web site as this allows for an organization to use something that has been tested and accepted by many other similar organizations as solution for their web development needs and requirements.

Performance and efficiency are almost always guaranteed as these technologies have survived the test of time and have been improved in various aspects by large numbers of other more experienced web developers.

It becomes easy to implement your web application or web site when you use web technologies and tools as there are multiple levels of abstraction over complex and usually logic demanding concepts that are quite difficult to implement.

Using such technologies and tools gives an organization higher exposure and more validity as fellow organizations who have been known to succeed using said technologies know that the tools you use are more than just a fluke.

# Search Engine Optimization (Evidence-based)

It does not matter how good your web application is if your web application doesn’t have a good Search Engine ranking. A search engine is the piece of software allows users to go through large databases of web sites to reach the web sites they wish to find.

Having good SEO is very important to the influence of the website, as this dictates whether or not users of your organization are aware of your web site’s existence and rely on your website. Otherwise, if you have poor SEO then your users will be unaware and you won’t have high internet traffic to your website. This will result in a waste of resources, money and time spent on producing a website at all.

[Moz](https://moz.com) is a company that does Search Engine Optimization professionally, the following procedures are what Moz does for all their customers, these have been proven to work per the reviews at their official homepage. Search Engine Optimization according to Moz focuses on the following points:

* **Crawl Accessibility –** Crawling is the process by which an application goes through the text based informative text within a website. Moz recommends that crawl accessibility is a key component so that search engines can see your web site and then rank your content.
* **Appealing Content –** According to Moz, the content audio, video or text based must be appealing and relevant to the user.
* **Keyword Optimization –** Having very perculiar keywords that are definitely bound to have search engines understand the relevance of your web site’s content is a must.
* **Quality User experience(UX) –** Having high load speeds of images and the overall web site, having compressed images that do not sacrifice quality, be responsive in order to look good in multiple devices.
* **Curious Content –** All your content according to Moz, must be share worthy so that people would take content off your website and cite it somewhere else which will lead other people to your website.
* **Appealing title and web address –** This is the most important and the most effective method of improving SEO optimization according to Moz, having a web address like “JanesFood” is better than a web address like “DeliciousFoods”, as a name like “JanesFood” is unique and is bound to grant you high ranking in the search engine.

# Front/Back-end Web Technologies: Frameworks & Tools

## How do web technologies cater to the OSI model?

A web technology may come in the form of a framework or a tool, that has been developed by a large community of developers in order to ease and improve the process of web development. In web development there are notably two different sections that are concerned:

* **Front-end**: The seeable representation that is rendered on to your browser is called the front end of a website or a web application. Frontend design and development is completely focused on logic and functional implementations of various different concepts that allow for a good User Experience(UX). The front-end is also called the User Interface(UI). The frontend is what acts as the presentation layer, compression algorithms like JPEG, GIF, PNG are used for visualizing image data, although nowadays SVG is much more popular. The presentation layer is what the user directly interacts with in the OSI model and the frontend is what gives fruition to the functionality of the presentation layer in the OSI model.
* **Back-end:** The un-seeable part of a website or web application. This handles the view, controls it and manipulates the data that is visualized in the UI. Usually computational logic that consumes large amounts of resources that would otherwise freeze the UI are done in the backend. Backend web development is much more difficult than frontend development in most case scenarios as it involves the cumulative focus on a multitude of concepts that include maintaining the frontend. The backend is what focuses on the application layer, through the manipulation of the application layer, different behaviours and responses are produced. When you interact with the presentation layer, the application layer’s components are manipulated which is the backend.

# Custom built vs web creation tool built websites

Websites can be created in a multiple ways, through the evolution of various technologies and tools, a lot of different solutions have been produced in order to ease the development procedure. However currently there are 2 main ways to build websites:

* Use a web creation tool – Doing so will make sure that you don’t have to write any code or design anything. Everything from designing the website to hosting the website is done by the web creation tool, your only purpose is to fill in the contents and drag and drop components on to your website. While this is a very easy thing to do, it is extremely unflexibile. The UI/UX flexibility is really low you can not flexibly create a website that is per your needs. The components that you drag and drop can only be used as they are and you can’t alternate their design or behavior unless they explicitly give you an interface to do so.
* Custom built – Everything from designing to hosting your web application must be done on your own. This is actually more resource and time consuming but ensures security and countless other key features of a good website like mobile first design, using design patterns, privacy, data flows, implementing ISO standards and efficient UI/UX design.

Some examples of web creation tools are:

* Wix
* Weebly
* Webs
* Wordpress drag-n-drop tool
* Bootstrap creation tool

All these creation tools have major problems when it comes to proper UI/UX design, also the reputation you will have as an organization if you use one of these creation tools to create your website will significantly low as the perspective most developers and industrial experts have of web creation tools are quite low.

As such, use web creation tools only when flexibile UI/UX design is not your concern but ease of deployment is. Usually web creation tools are used by individuals not by organizations, usually they are used for creating blogs, image portfolios and diaries.

# Justification, recommendation and explanation of various front and backend design and development technologies for custom built websites

The design section of each category includes tools used for designing the frontend or the backend, while the development section will include frameworks and runtimes that are used for backend development.

|  |  |
| --- | --- |
| **Frontend** | |
| **Design** | **Development** |
| * **Adobe Xd** – A industrially accepted design, wireframing and prototyping application produced by Adobe. It is slowly getting very high exposure and popularity in the industry. It is very effective and has a lot of components that already come with it. The already existing functionality of Adobe Xd can be extended using community plugins as such the power you have with Adobe Xd is endless. * **Microsoft Publisher** – Quite the obsolete and legacy type of application that is used to design websites. It has a very simple interface like most of the Microsoft applications but the functionality and the purpose it has only stretches so much. It is not as industrially popular as it was back in the 2007s. * **Sketch** – A MacOS only digital design tool, which is very famous among frontend designers, as sketch provides various UI frameworks and widgets that allow designing responsive and appealing frontends in very short periods of time. * **Adobe Illustrator** – Although not majorly used for UI/UX design, illustrator is quite often used by beginner frontend designers for design phase of the frontend alone. It does not provide any features like wireframing or prototyping by default, as such it is not industrially used for frontend design. * **Adobe Photoshop** – Like Adobe Illustrator, it is also just used to make mockups of frontend UI/UX designs, it too, does not provide any features like wireframing or prototyping. However though, as graphic designers use Adobe Photoshop intensively, Adobe Photoshop sees large industrial exposure for frontend design. | * **JQuery** – A very famous and almost obsolete frontend development framework that is focused on DOM manipulation, and doing it safely while ensuring that you do not sacrifice performance. JQuery has survived the test of time and is quite often used in the industry due to its extreme simplicity and performant interfaces. It uses Javascript concepts like function currying and composition and ES6 symbols. Usually for support for various technologies that are not supported in Internet Explorer, JQuery is used to write efficient polyfills. * **ReactJS** – An extremely popular and the frontend development giant in the development world, produced and used by Facebook, it has multiple features like one-way data flow/binding, stateful and functional components, hooks and lifecyle methods. ReactJS is described as the V in the MVC design pattern. * **AngularJS** – A frontend framework that has a much more higher resource consumption and composition than ReactJS produced by Google specifically for Single Page Applications (SPAs). It has a multiple features like server-side rendering, stateful components. AngularJS is also described as the V in the MVC design pattern. * **VueJS** – Another frontend framework that was produced to combine the features of AngularJS and ReactJS but in a smaller minified size so that your bundle size is as less as possible. It is seeing a lot of industrial exposure and is widely used in China as the developer/founder of VueJS is Chinese. * **Supercharged CSS** – SCSS is a CSS preprocessor. SCSS allows for multiple things that are usually not possible with pure CSS like mixins, blends, conditional statements, loops etc. |
| **Backend** | |
| **Design** | **Development** |
| There are not a many ways that you can “design” a backend, however there are two very effective methods which are:   * **Visualization through rough sketches** - by doing so you will try to visually describe every component of the backend, how data flows, how each component of the backend interacts with each other and how the backend architecture must be implemented. * **UML Diagrams** – Using multiple UML diagrams like flowcharts, data flow diagrams, entity relationship diagrams etc. you can describe and express the architecture and the logistics within the backend of an application. There are many industrially accepted UML diagrams that have been proven to focus and efficiently demonstrate the backend design of a web application. | * **PHP –** One of the oldest languages to exist that have been used for backend development. It isn’t used as intensively anymore as the way of programming is no longer synchronous. PHP has proven to be a rabbit hole of a lot of problems, and it is extremely challenging to create an efficient architecture using PHP. It is however an extremely simple language to develop the backend as such the industrial exposure it has seen reduces by each month. * **ExpressJS –** It is a backend web application framework developed for NodeJS based solutions. It is widely used by developers for creation of RESTFul APIs and it is extremely efficient and performant. It is widely used by both experts and beginners in the industry. It has been ranked as one of the most used technologies for backend development by the Developer Economics Survey 2018. * **Flask –** A python based web application framework for Python based solutions. It is very simple to get started with. Although it’s features are limited, it is very efficient in what it does. Most startups and developers that are just starting web developers reap benefits from PyFlask, however there is usual a transition from Flask to something more capable like Django. * **Django –** A python based web application framework for Python based solutions. This isn’t as user-friendly and more difficult to get started with when it comes to backend development. However it has a lot of features for heavy industrial complications and features and has been a framework that has survived the test of time. * **Spring –** A Java based web application framework for Java based solutions. It has a lot features and mainly focuses on the Object Oriented Programming paradigm as Java revolves around this programming paradigm. |

**Disclaimer: All above content are entirely my own, justifications and explanations are not taken from anywhere, as such they are my personal opinions not of any external source or person.**

|  |  |
| --- | --- |
| **Bibliography** | |
| Moz | <https://moz.com/> |
| Reddit (r/webdev) | <https://www.reddit.com/r/webdev/> |