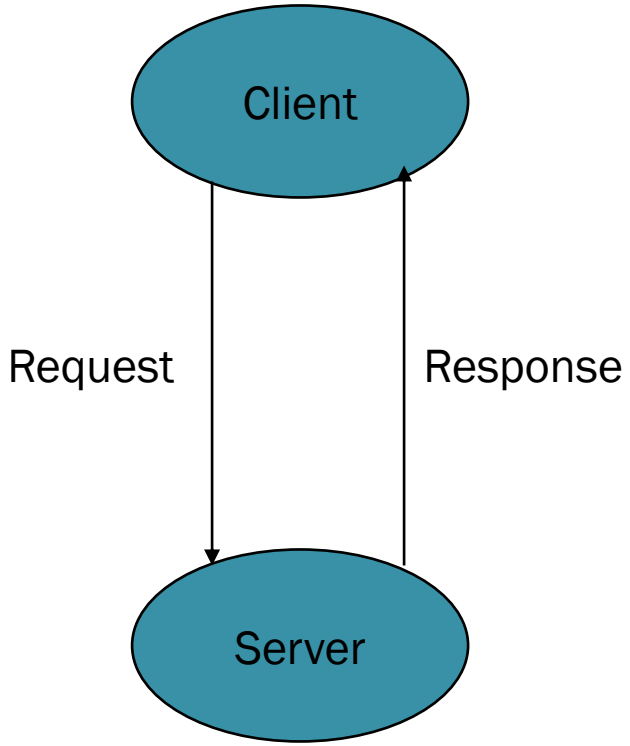
WEB TECHNOLOGY BASIC

Network Application

- An Application over a network is called a network application
- A network application is commonly called a service
- A service comprises of two discrete participants:
  - *A SERVER that provides services*
  - *A CLIENT that requests services from the server*
- Together, the two form a complete computing system with a distinct division of responsibility and clean separation of duty based on the idea of service
- Technically Client /Server computing relates two or more threads of execution using a consumer / producer relationship.

Client/Server Computing

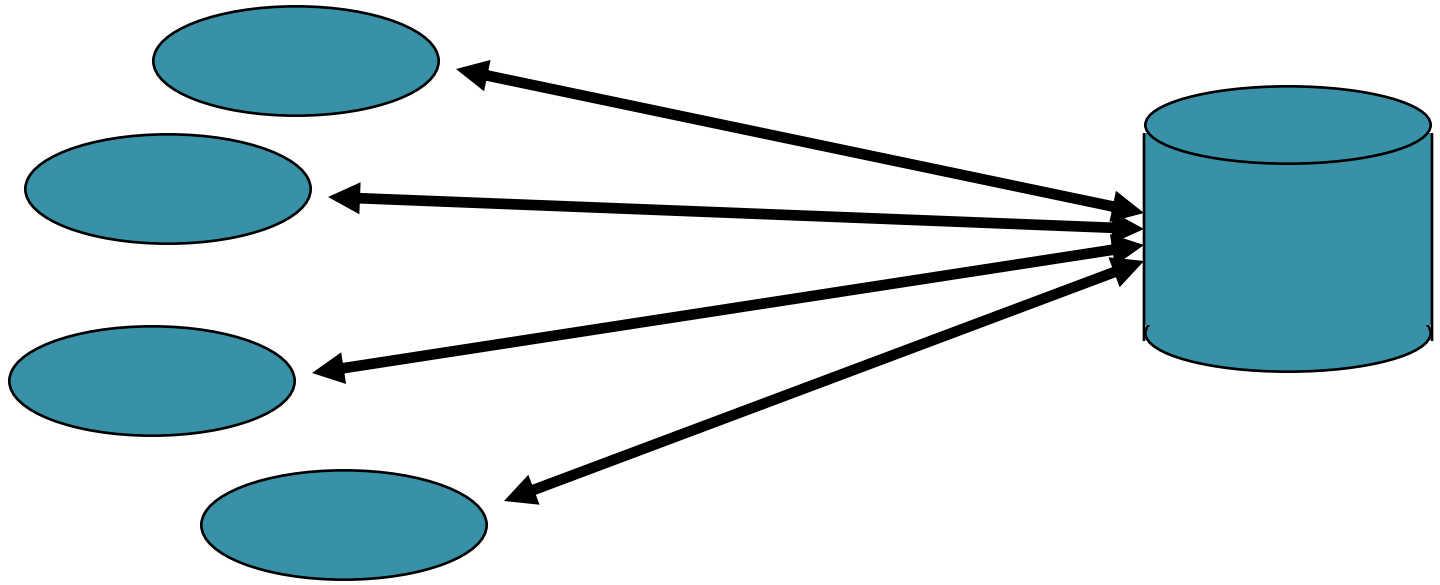
- Client/Server computing is a power full paradigm of computing.
- In this paradigm a client requests a server for a service and the server responds with the service.
- Clients need to be aware of the servers that are available but usually do not know of the existence of other clients.



Client-Server concept

- Server program is shared by many clients
- RPC protocol typically used to issue requests
- Server may manage special data, run on an especially fast platform, or have an especially large disk
- Client systems handle “front-end” processing and interaction with the human user

Server and its clients



Examples of servers

- Network file server
- Database server
- Network information server
- Domain name service
- authentication server

Why use a client-server approach?

- Pricing parameters are “expensive” (in terms of computing resources) to obtain: must monitor many data sources and pre-compute many parameters.
- Computing demands may be extreme: demands a very high performance machine
- Database of bonds is huge: large storage, more pre-computation

On client side

- Need a lot of CPU and graphics power to display the data and interact with the user (Based on Application)
- Dedicated computation provides compressed response time and powerful decision making aids user (Based on Application)
- Can “cache” or “save” results of old computations so that if user revisits them, won’t need to reissue identical request to server
- Some client many only required interact with the user and display result.

Summary of typical split

- Server deals with bulk data storage, high performance computation, collecting huge amounts of background data that may be useful to any of several clients
- Client deals with the “attractive” display, quick interaction times
- Use of caching to speed response time

Server: Passive and Active

Servers may be of two types based on participation :

- *passive* and *active*.
  - *Traditional servers are passive. They passively await requests from clients and then act on them.*
  - *Protocol is initiated by the client.*
  - *It is termed as Client Pull.*

  - *Recent researches conceptualize what is called active servers*
  - *active servers can actively search out changes in the state of clients and take appropriate actions. It is termed as Server push.*
  - ***In Server Push ,server pushes** a resource directly to the client without the client asking for the resource. The **server** is making an assumption here that **pushing** the resource is desirable.*

Server: Stateless and Stateful

- Servers may also be of two types based on forgetfulness: *stateless* and *statefull*.
- A statefull server has stronger memory power. It can not only remember the facts of the ongoing transaction but also history of previous transactions. Therefore, the design of a statefull server is very complex.
- A stateless server has very limited memory power. It can only remember the facts of the ongoing transaction between the server and the client. After the transaction is over, it forgets every thing about the transaction. Its design is comparatively simple.
- Web Server is a stateless server.
- A server can act as a client to other servers. Using this model, a server can execute a task by dividing it into sub-tasks and then having other servers complete the subtasks.

Server: Stateless and Stateful || other concepts

- Client-server system is *stateless* if:

Client is independently responsible for its actions, server doesn't track set of clients or ensure that cached data stays up to date

- Client-server system is *stateful* if:

Server tracks its clients, takes actions to keep their cached states "current". Client can trust its cached data.

- Web Server is a stateless server.
- A server can act as a client to other servers. Using this model, a server can execute a task by dividing it into sub-tasks and then having other servers complete the subtasks.

Typical issues in design

- Client is generally simpler than server: may be single-threaded, can wait for reply to RPC's
- Server is generally multithreaded, designed to achieve extremely high concurrency and throughput. Much harder to develop
- Reliability issue: if server goes down, all its clients may be "stuck". Usually addressed with some form of backup or replication.

Use of caching

- In stateless architectures, cache is *responsibility of the client.* Client decides to remember results of queries and reuse them. Example: caching Web proxies, the NFS client-side cache.
- In stateful architectures, cache is *owned by server.* Server uses “callbacks” to its clients to inform them if cached data changes, becomes invalid. Cache is “shared state” between them.

Characteristics of a Client / Server System

- A client/server application has the following characteristics:
- **shared resources**: A server can service many clients at the same time and regulate their access to shared resources.
- **transparency of location**: The server process can reside on the same machine as the client or on a separate machine across a network. C/S software usually masks the location of the server from the clients by redirecting the service calls when needed.
- **platform independence**: An ideal C/S system is platform independent.
- **message-based exchange**: clients and servers are loosely coupled systems that interact through a message-passing mechanism. The message is the delivery mechanism for the service requests and replies

Contd..

- **encapsulation of services**: A message tells a server what service is requested. It is up to the server to determine how to get the job done. A server can be upgraded without effecting the clients as long as the published message interface is not changed.

- **scalability**: A C/S system can be scaled horizontally or vertically.
  - *Horizontal scaling means adding or removing clients with only little performance impact.*
  - *Vertical scaling means upgrading to a larger or faster server or multi-server*

- **integrity**: The server code and server data is centrally maintained, which results in cheaper maintenance and the guarding of shared data integrity. At the same time, the clients remain personal and independent.

Virtual Hosting

- **Virtual hosting** is a method for **hosting** multiple domain names (with separate handling of each name) on a single server (or pool of servers). This allows one server to share its resources, such as memory and processor cycles, without requiring all services provided to use the same **host** name.
  - *Normally one host has one domain name*
  - *Virtual hosting is a method for hosting multiple domain names (with separate handling of each name) on a single server*
  - *IP address to Domain Name mapping is 1:1*
  - *IP address to DN is 1:M*
  - *One IP address but multiple domain name*
  - *Web Site hosting agencies use virtual hosting*
  - *A host may have more than one IP address, if it is connected to many networks*
  - *There are two main types of virtual hosting*
- name-based
- IP-based.

Name Based Hosting

- Name-based virtual hosts use multiple host names for the same IP address.
- A technical prerequisite needed for name-based virtual hosts is a web browser with HTTP/1.1 support to include the target hostname in the request. This allows a server hosting multiple sites behind one IP address to deliver the correct site's content.
- For instance, a server could be receiving requests for two domains, www.example.com and www.example.net, both of which resolve to the same IP address.
- For www.example.com, the server would send the HTML file from the directory /var/www/user/GU/site/, while requests for www.example.net would make the server serve pages from /var/www/user/AU/site/.

IP Based Hosting

- When IP-based virtual hosting is used, each site (either a DNS host name) points to a unique IP address.
- The web server is configured with multiple physical network interfaces, virtual network interfaces on the same physical interface or multiple IP addresses on one interface.

Port-based Hosting

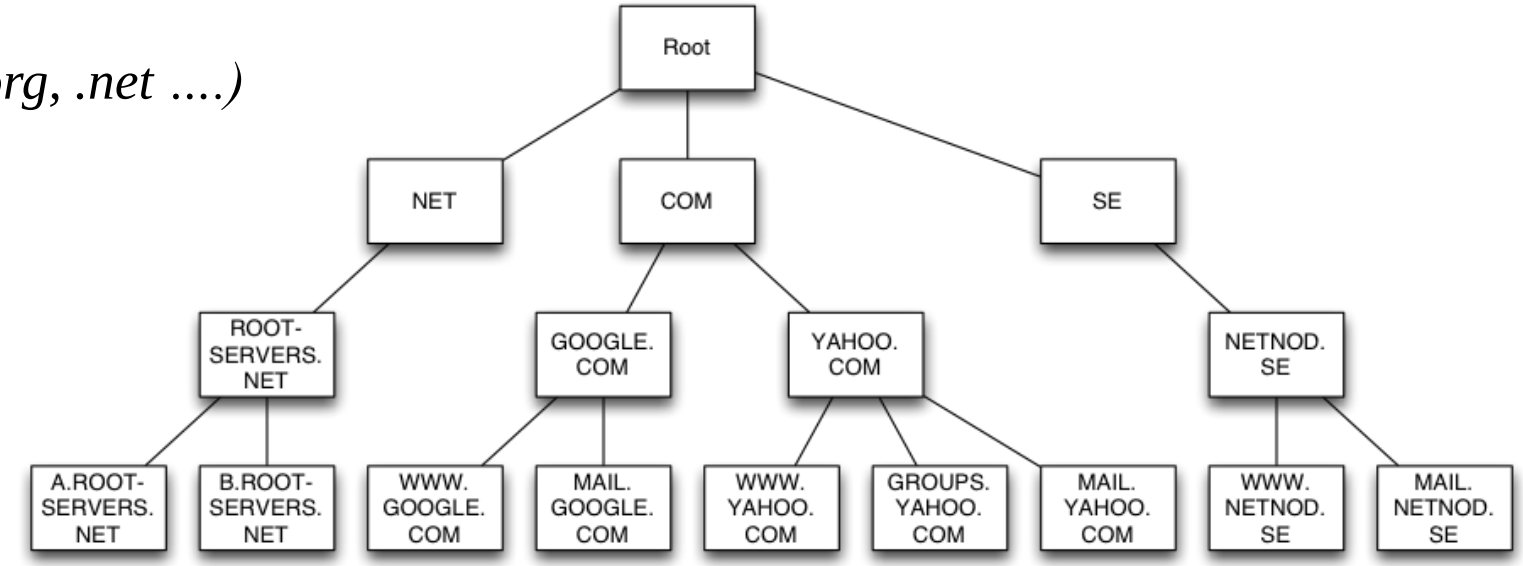
- The default port number for HTTP is 80. However, most web servers can be configured to operate on almost any port number, provided the port number is not in use by any other program on the server.
- For example, a server may host the website www.example.com.
- If the owner wishes to operate a second site, can use another port number, for example, www.example.com:81 for port 81, www.example1.com:8000 for port 8000, or www.example2.com:8080 for port 8080.
- However this is not a user friendly approach. Users cannot reasonably be expected to know the port numbers for their websites and moving a site between servers may require changing the port number.
- Using non-standard port numbers may also be seen as unprofessional and unattractive to users.
- In addition, some firewalls block all but the most common ports, causing a site hosted on a non-standard port to appear unavailable to some users.

Name Service

- For a common user it is hard to remember correctly an IP Address
- An equivalent symbolic name is associated with this number as an identifier of the machine in application layer and such a name is called domain name.
- With domain name a connection cannot be established directly.
- For transport layer connection, the IP address of the host is required
- Name service resolves a domain name to IP address for establishing connection or vice versa

Hierarchical Naming Scheme

- The Domain **Name System** (**DNS**) is a **hierarchical** and decentralized **naming system** for computers, services, or other resources connected to the Internet or a private network. It associates various information with domain **names** assigned to each of the participating entities.
- A hierarchical, domain-based naming scheme
- Root Domain (.)
- Top Level Domains (TLD)
  - *Generic (.com, .edu, .gov, .org, .net ....)*
  - *Country (.in, .jp .us .nl ...)*



Hierarchical Naming Scheme

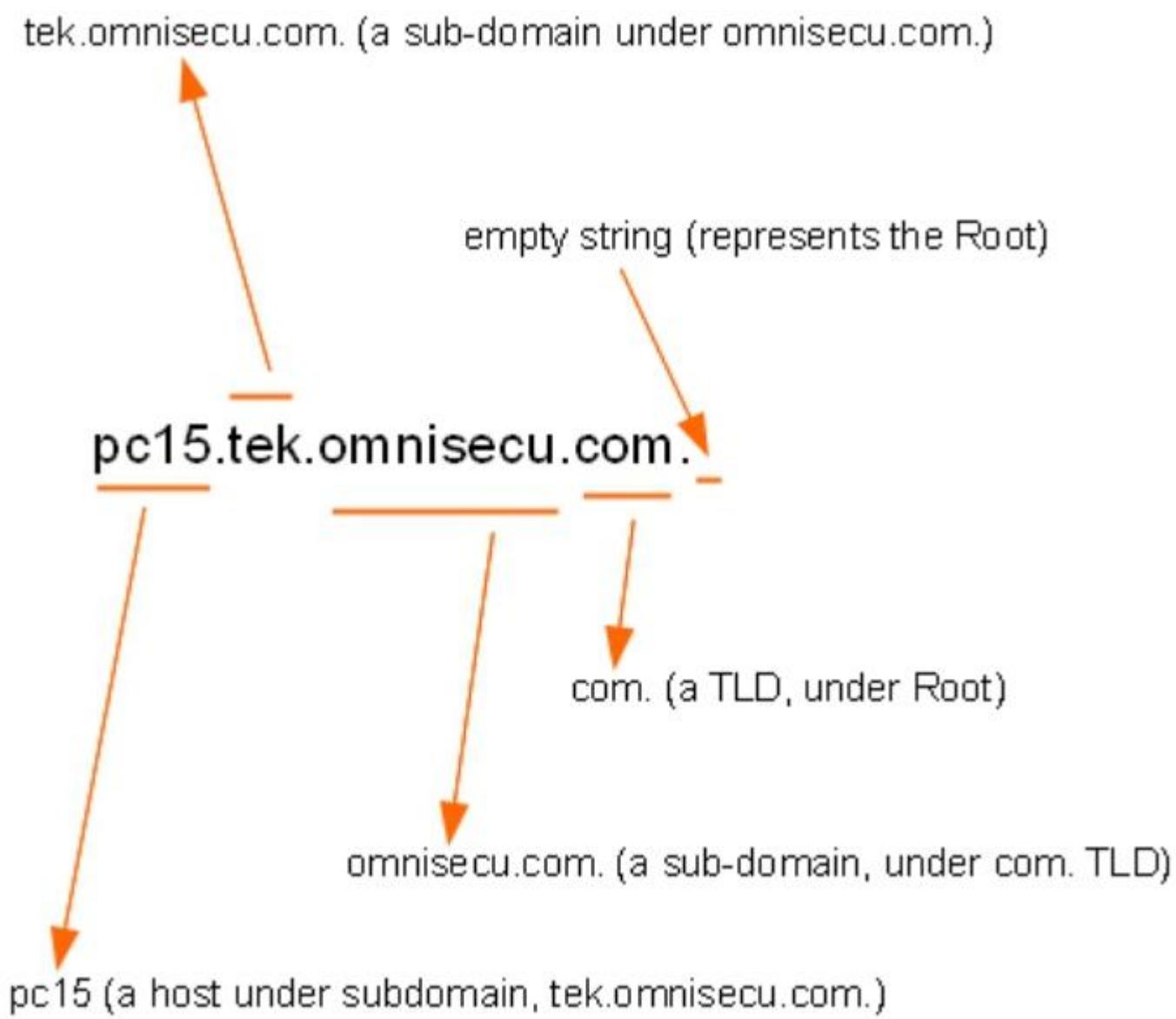
A **top-level domain** (**TLD**) is one of the domains at the highest level in the hierarchical Domain Name System of the Internet

The top-level domain names are installed in the root zone of the name space. For all domains in lower levels, it is the last part of the domain name, that is, the last label of a fully qualified domain name.

- A TLD will have sub-domains and a sub-domain will have sub-sub-domains and so on
- And thus a hierarchy is formed
- Domain name of a host in a network is a path from root to the hostname
- It is expressed in dot-decimal format of path expression
  - *Eg. www.gauhati.ac.in*
- Domain names are case insensitive.
- There is co correspondence among component names and bytes of IP address

Fully Qualified Domain Name (FQDN)

- Fully Qualified Domain Name (FQDN) of a host in the DNS namespace hierarchy consists of all the labels from the node, up to the root of the namespace, separated by periods (".").
- Fully Qualified Domain Name (FQDN) must end with a empty string, which represents the Root. Since there is no need to represent empty string, Fully Qualified Domain Name (FQDN) ends with a period (.). The trailing period (".") for the root domain is usually omitted in day to day use, but the DNS Resolver (Client) and DNS Servers must use it during actual DNS name queries..
- Using Fully Qualified Domain Name (FQDN) , we can identify a host's position from the Root of the DNS namespace. An Fully Qualified Domain Name (FQDN) can unambiguously indicate the position of a host relative to the DNS Root.
- Example of a Fully Qualified Domain Name (FQDN) is **pc15.tek.omnisecu.com.**


tek.omnisecu.com. (a sub-domain under omnisecu.com.)
empty string (represents the Root)
pc15.tek.omnisecu.com.
com. (a TLD, under Root)
omnisecu.com. (a sub-domain, under com. TLD)
pc15 (a host under subdomain, tek.omnisecu.com.)

A Partially Qualified Domain Name (PQDN)

- A Partially Qualified Domain Name (PQDN) is used to specify a portion of a domain name, normally the host portion of it. A Partially Qualified Domain Name (PQDN) starts with a host name, but it may not reach up to the root.

- Example of Partially Qualified Domain Name (PQDN) pc15. Usually the computers will add the DNS suffix along with Partially Qualified Domain Name (PQDN) before sending a DNS query for name resolution.
- Example of a Partially Qualified Domain Name (FQDN) is **pc15.**

Domain Name System (DNS)

- Defined in RFC (Request for Comments) 1034,1035

The system comprises of

- Name Client:
  - *Name client requests a name server to resolve the given domain name to its IP address*
  - *It uses UDP and not TCP in the transport layer*
- Name Server:
  - *name server resolves the given name to its IP address*
  - *It listens to port no UDP port no 53*
- Distributed Name Database
  - *name-to-IP address mappings are stored in a name database associated with a name server*

Protocol

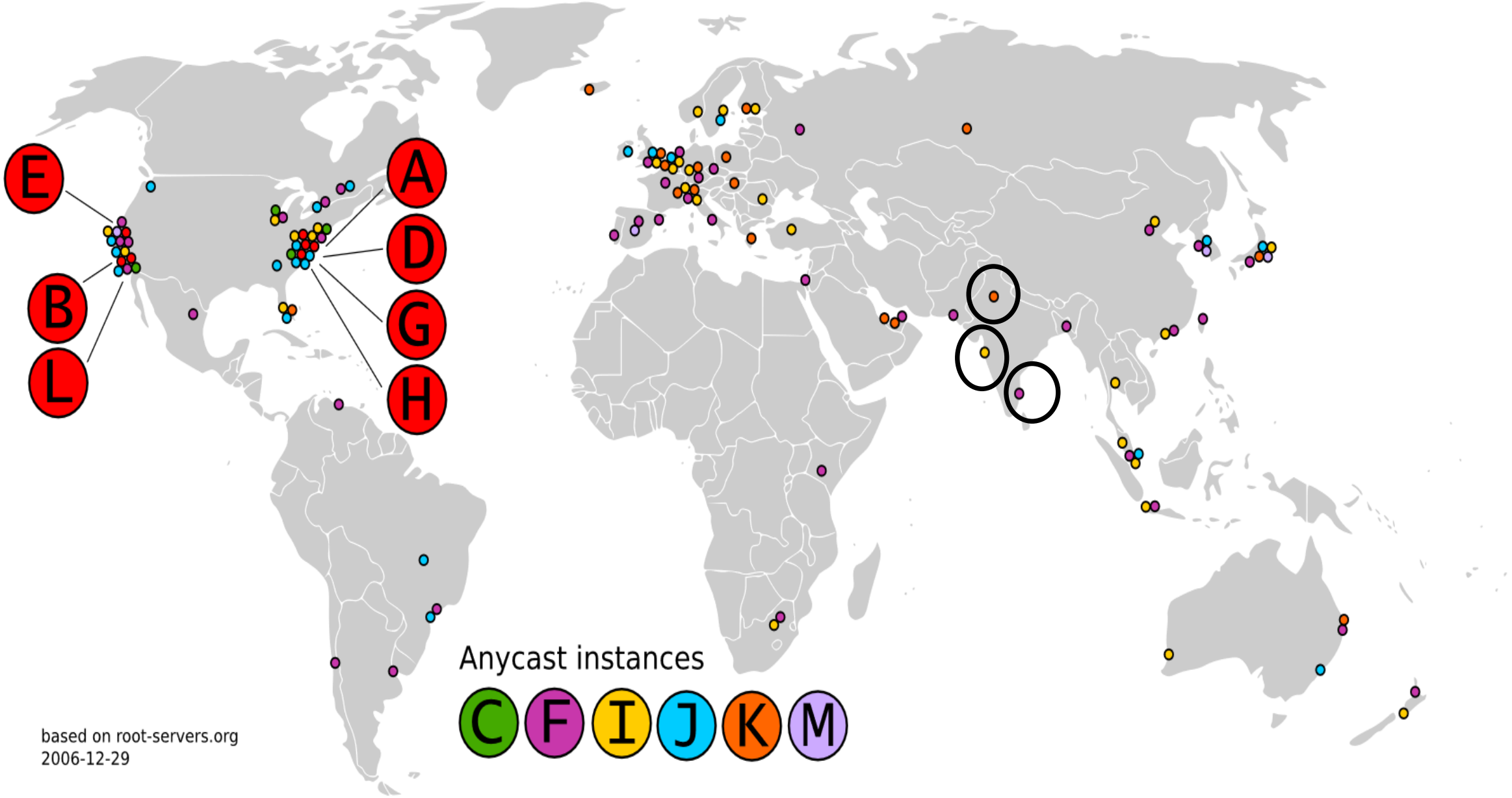
- A client calls a library program, Recursive resolver
- Resolver sends a UDP packet the local name server listening at port no 53
- The server looks into the local name database and returns the ip address to the resolver if it has it in its cache. Resolver forwards it to the client
- If it does not know it will see how closely it can match the requested name and use whatever information it has cached.
- It will remove the leftmost parts one at a time, checking if it knows anything about www.gauhati.ac.in , gauhati.ac.in, ac.in. , in,
- In the worst case there is no match but the `.' (root) of the name, It will then ask a . (root) server about www.gauhati.ac.in
- This . (root) server will not know the answer but it will help your server on its way by giving a referral, telling it where to look instead.
- These referrals will eventually lead your server to a nameserver that knows the answer.

Name Servers

- Local name servers:
  - *each ISP, company has local (default) name server*
  - *host DNS query first goes to local name server and is resolved by using local name database.*
- Authoritative name server:
  - *for a host: stores that host's IP address, name*
  - *can perform name/address translation for that host's name*
- Root Name Servers
  - *Root servers are the servers at the root of the Domain Name System hierarchy*
  - *". " Servers*
  - *13 Nos in the world*

Root Name Servers

- A **root name server** is a name server for the Domain Name System's root zone.
- It directly answers requests for records in the root zone and answers other requests returning a list of the designated authoritative name servers for the appropriate top-level domain (TLD).
- The root name servers are a critical part of the Internet because they are the first step in translating (resolving) human readable host names into IP addresses that are used in communication between Internet hosts.
- There are currently 13 root name servers specified, with names in the form *letter*.root-servers.net, where *letter* ranges from A to M
- In India we have 3 root servers K-server in Delhi , I-server in Mumbai, F-server in Chennai



Mumbai (I root), one in Delhi (K root) and one in Chennai (F root)

Who operates root servers?

The root servers are operated by 12 different organizations:

- A VeriSign Global Registry Services
- B University of Southern California, Information Sciences Institute
- C Cogent Communications
- D University of Maryland
- E NASA Ames Research Center
- F Internet Systems Consortium, Inc.
- G US DoD Network Information Center
- H US Army Research Lab
- I Netnod
- J VeriSign Global Registry Services
- K RIPE NCC
- L ICANN
- M WIDE Project

- There are more than 750 root server instances around the world, on all six populated continents. They are reachable using 13 numeric IP addresses – one per operating organization, except for Verisign, which operates two root servers.
- Most of those addresses are assigned to multiple servers around the world, so DNS queries sent to those addresses get fast responses from local servers.
- Because there are only 13 root server IP addresses, only 13 root servers can be seen from any single location at any given time. Different servers (using the same IP addresses) will be seen from different locations.

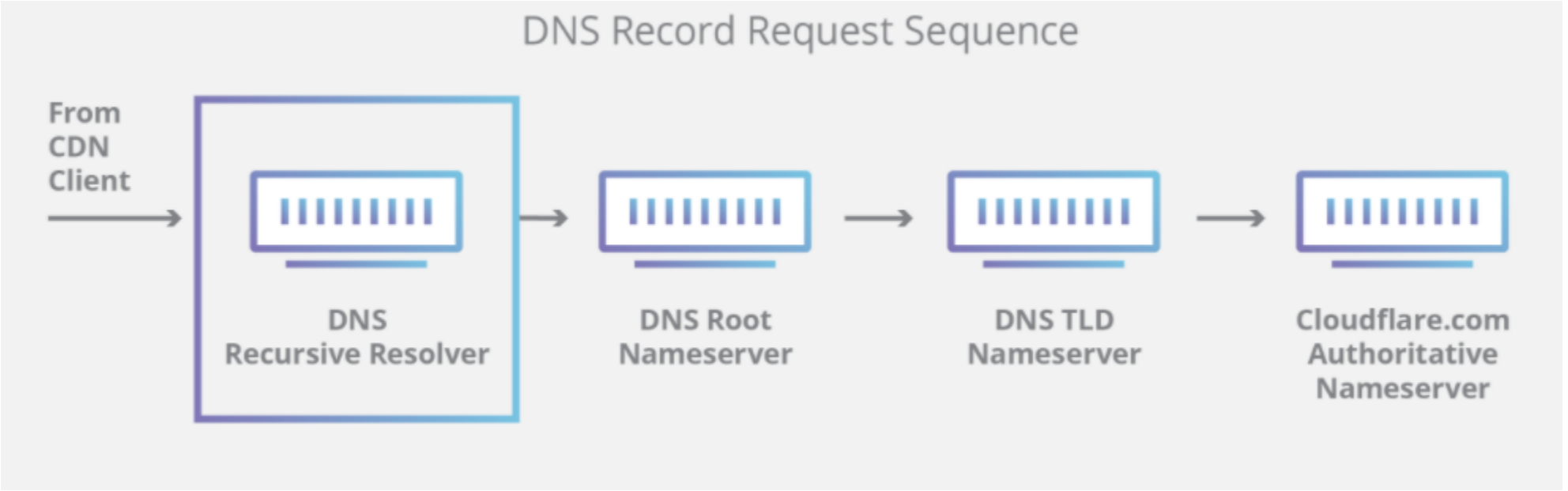
Authoritative Name Server

- An authoritative name server provides actual answer to your DNS queries such as – mail server IP address or web site IP address .
- It provides original and definitive answers to DNS queries.
- It does not provides just cached answers that were obtained from another name server. Therefore it only returns answers to queries about domain names that are installed in its configuration system. There are two types of Authoritative Name Servers:
- **Master server (primary name server)** – A master server stores the original master copies of all zone records.
- A host-master only make changes to master server zone records.
- Each slave server gets updates via special automatic updating mechanism of the DNS protocol. All slave servers maintain an identical copy of the master records.
- **Slave server (secondary name server)** – A slave server is exact replica of master server. It is used to share DNS server load and to improve DNS zone availability in case master server fails.

There are 4 DNS servers involved in loading a webpage:

- DNS recursor - The recursor can be thought of as a librarian who is asked to go find a particular book somewhere in a library. The DNS recursor is a server designed to receive queries from client machines through applications such as web browsers. Typically the recursor is then responsible for making additional requests in order to satisfy the client's DNS query.
- **Root nameserver** - The root server is the first step in translating (resolving) human readable host names into IP addresses. It can be thought of like an index in a library that points to different racks of books - typically it serves as a reference to other more specific locations.
- TLD nameserver - The top level domain server (TLD) can be thought of as a specific rack of books in a library. This nameserver is the next step in the search for a specific IP address, and it hosts the last portion of a hostname (In example.com, the TLD server is "com").
- Authoritative nameserver - This final nameserver can be thought of as a dictionary on a rack of books, in which a specific name can be translated into its definition. The authoritative nameserver is the last stop in the nameserver query. If the authoritative name server has access to the requested record, it will return the IP address for the requested hostname back to the DNS Recursor (the librarian) that made the initial request.

Source:: https://www.cloudflare.com/learning/dns/what-is-dns/

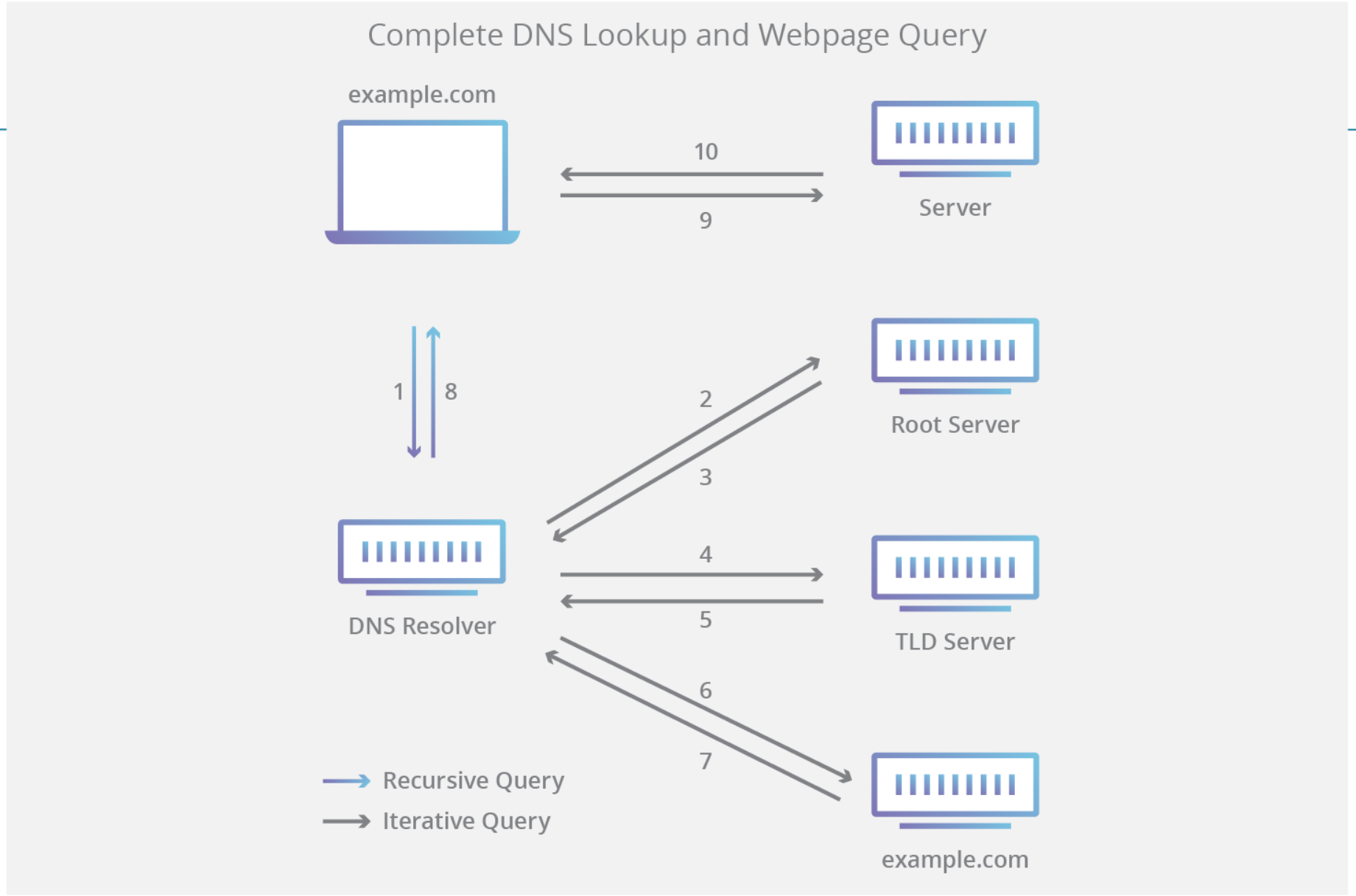


Source:: https://www.cloudflare.com/learning/dns/what-is-dns/

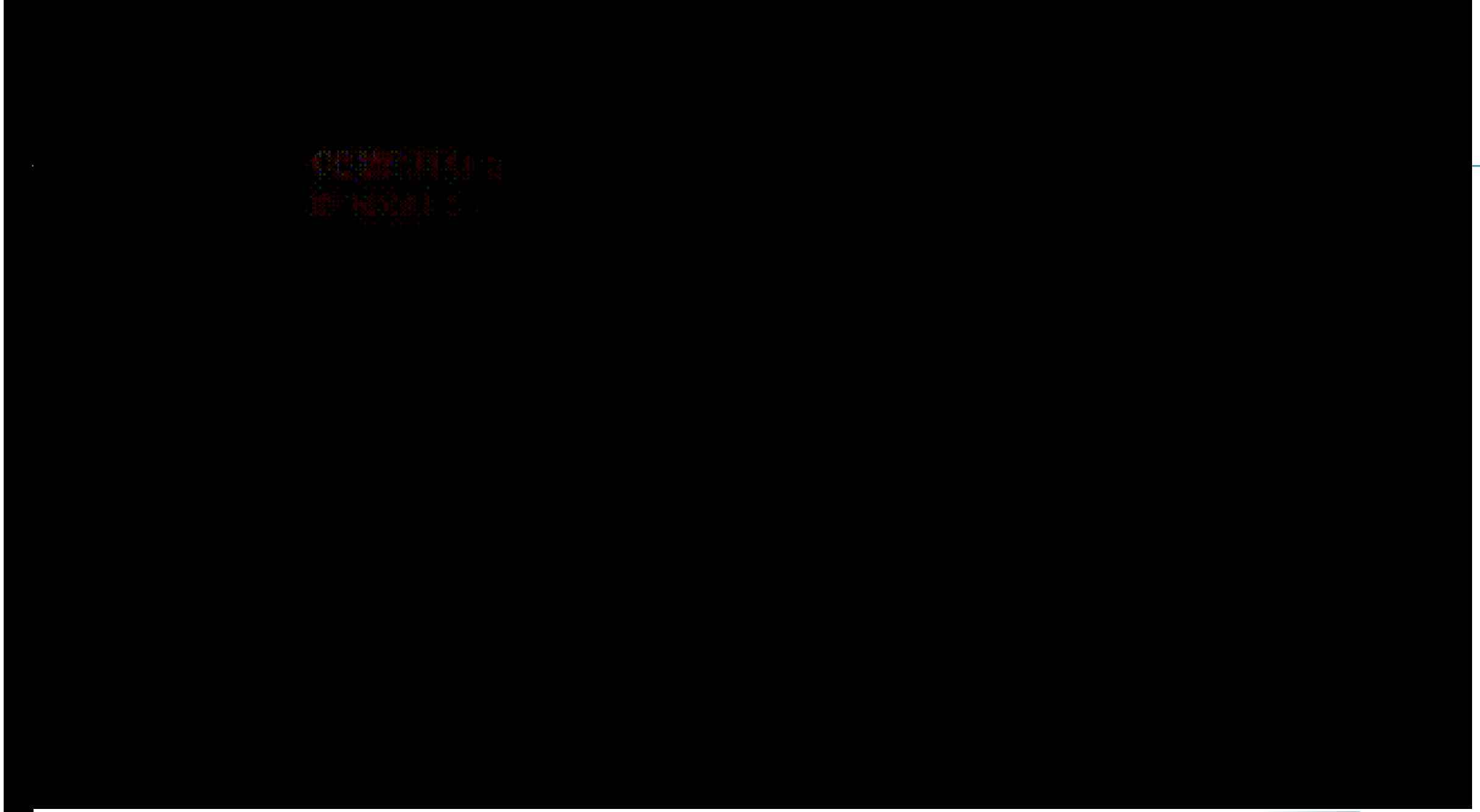
Complete DNS lookup Process:

- A user types ‘example.com’ into a web browser and the query travels into the Internet and is received by a DNS recursive resolver.
- The resolver then queries a DNS root nameserver (.).
- The root server then responds to the resolver with the address of a Top Level Domain (TLD) DNS server (such as .com or .net), which stores the information for its domains. When searching for example.com, our request is pointed toward the .com TLD.
- The resolver then makes a request to the .com TLD.
- The TLD server then responds with the IP address of the domain’s nameserver, example.com.
- Lastly, the recursive resolver sends a query to the domain’s nameserver.
- The IP address for example.com is then returned to the resolver from the nameserver.
- The DNS resolver then responds to the web browser with the IP address of the domain requested initially.
- Once the DNS lookup have returned the IP address for example.com, the browser is able to make the request for the web page:
- The browser makes a HTTP request to the IP address.
- The server at that IP returns the webpage to be rendered in the browser (step 10).

Source:: https://www.cloudflare.com/learning/dns/what-is-dns/

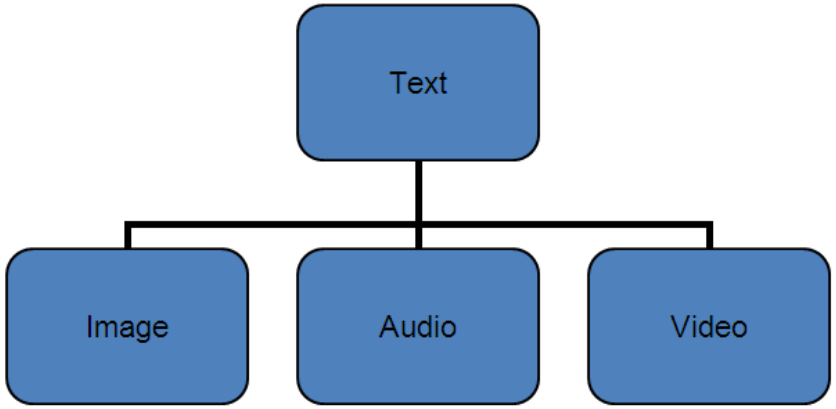


Source:: https://www.cloudflare.com/learning/dns/what-is-dns/



Hypermedia Page

- Hypermedia components
  - *Text (.html, .htm)*
  - *Image (.jpg, .gif, etc)*
  - *Audio (.wav)*
  - *Video (.avi)*



- ✓ Each component is a separate file
- ✓ Text file serves as the binding element of other files
- ✓ Every file is downloaded separately in HTTP

Uniform Resource Locator: URL

protocol://domain_name:target_port/relative_path/file_name.extension#section

- http://www.gauhati.ac.in/ist/prog/btech/index.html#btech_04
- Defaults
  - *Protocol: http*
  - *Domain_name: current*
  - *Target Port:80*
  - *Relative path: wrt DocumentRoot (/var/www/html)*
  - *File: DirectoryIndex (index.html)*
  - *Section : #TOP*
- Server Configutation Directives in /etc/httpd/conf/httpd.conf
- In lieu of domain_name we can use an IP address also.

Example

URI

- A Uniform Resource Identifier (URI) is a compact sequence of characters that identifies an abstract or physical resource.
- A URI can be further classified as **a locator, a name**, or both.
- The term “Uniform Resource Locator” (URL) refers to the subset of URIs that, in addition to identifying a resource, provide a means of locating the resource by describing its primary access mechanism (e.g., its network “location”).

URL

- A URL is a URI that, in addition to identifying a web resource, specifies the means of acting upon or obtaining the representation, specifying both its primary access mechanism and network location.
- For example, the URL http://guist.ac.in/index.html refers to a resource identified as index.html whose representation, in the form of HTML and related code, is obtainable via HyperText Transfer Protocol (http) from a network host whose domain name is guist.ac.in

URN

- A URN is a URI that identifies a resource by name in a particular namespace *.
- A URN can be used to talk about a resource without implying its location or how to access it.
  - *The International Standard Book Number (ISBN) system for uniquely identifying books provides a typical example of the use of URNs.*
  - *The URN for that edition would be urn:isbn:0-486-27557-4.*

*** Namespaces**

- In order to ensure the global uniqueness of URN namespaces, their identifiers (NID [Namespace Identifier]) are required to be registered with the IANA(Internet Assigned Numbers Authority) .

Contd..

Syntax

- The syntax of a URN

<URN> ::= "urn:" <NID> ":" <NSS>

- This renders as: urn:<NID>:<NSS>
- The leading

– *urn: sequence is case-insensitive.*

– *<NID> is the namespace identifier, which determines the syntactic interpretation of <NSS>, the namespace-specific string.*

– *Example : urn:isbn:0451450523*

Conceptual distinctions

- "URL is a useful but informal concept: a URL is a type of URI that identifies a resource via a representation of its primary access mechanism (e.g., its network "location"), rather than by some other attributes it may have". A URL is simply a URI that happens to point to a physical resource over a network.

- A URN is a URI that identifies a resource by name in a particular namespace.

Examples

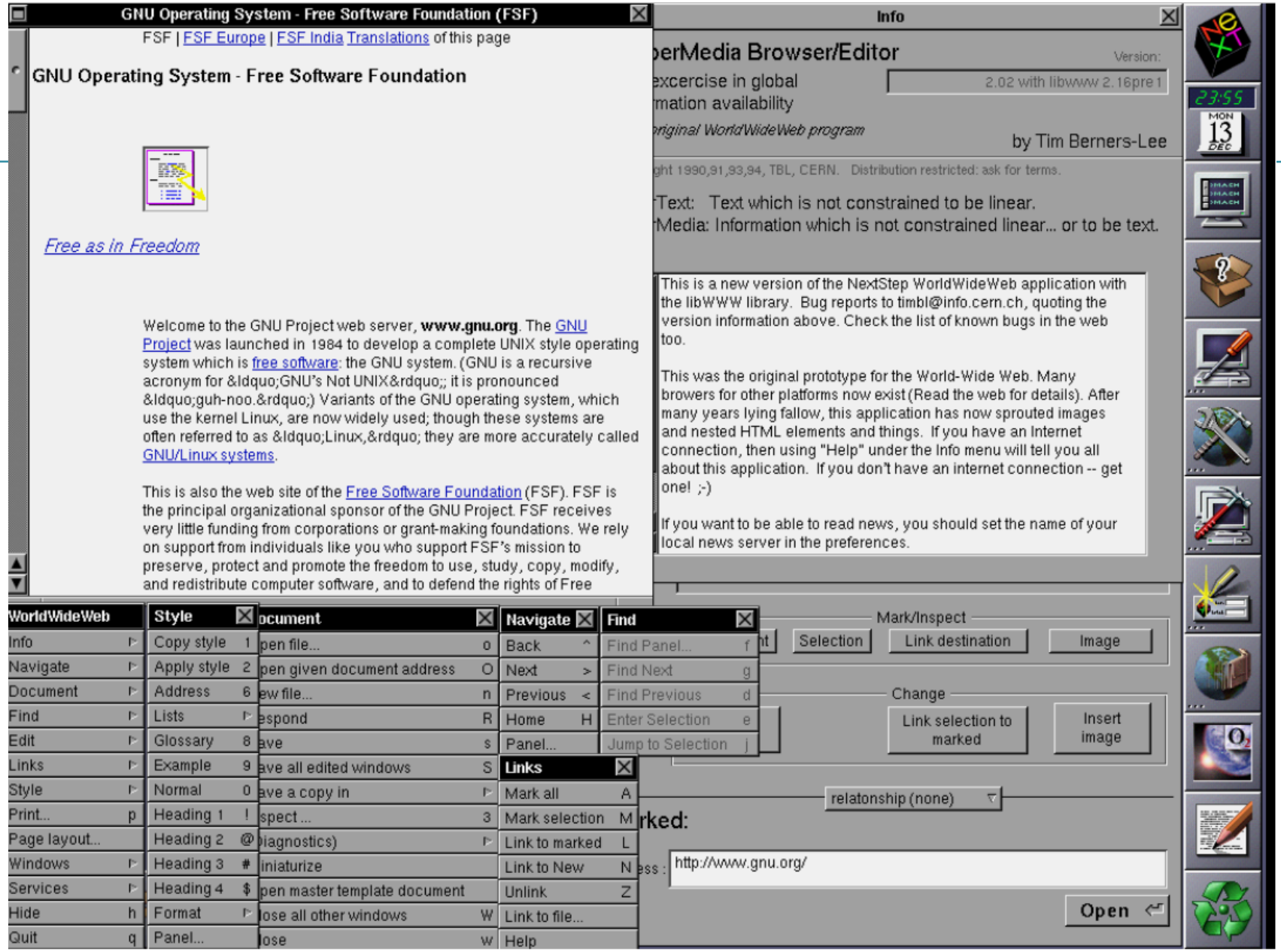
1. ftp:is.co.za/rfc/rfc1808.txt
2. http://www.ietf.org/rfc/rfc2396.txt
3. mailto:John.Doe@example.com
4. 192.0.2.16:80/ text.html
5. urn:oasis:names:specification:docbook:dtd:xml:4.1.2

Web Browser

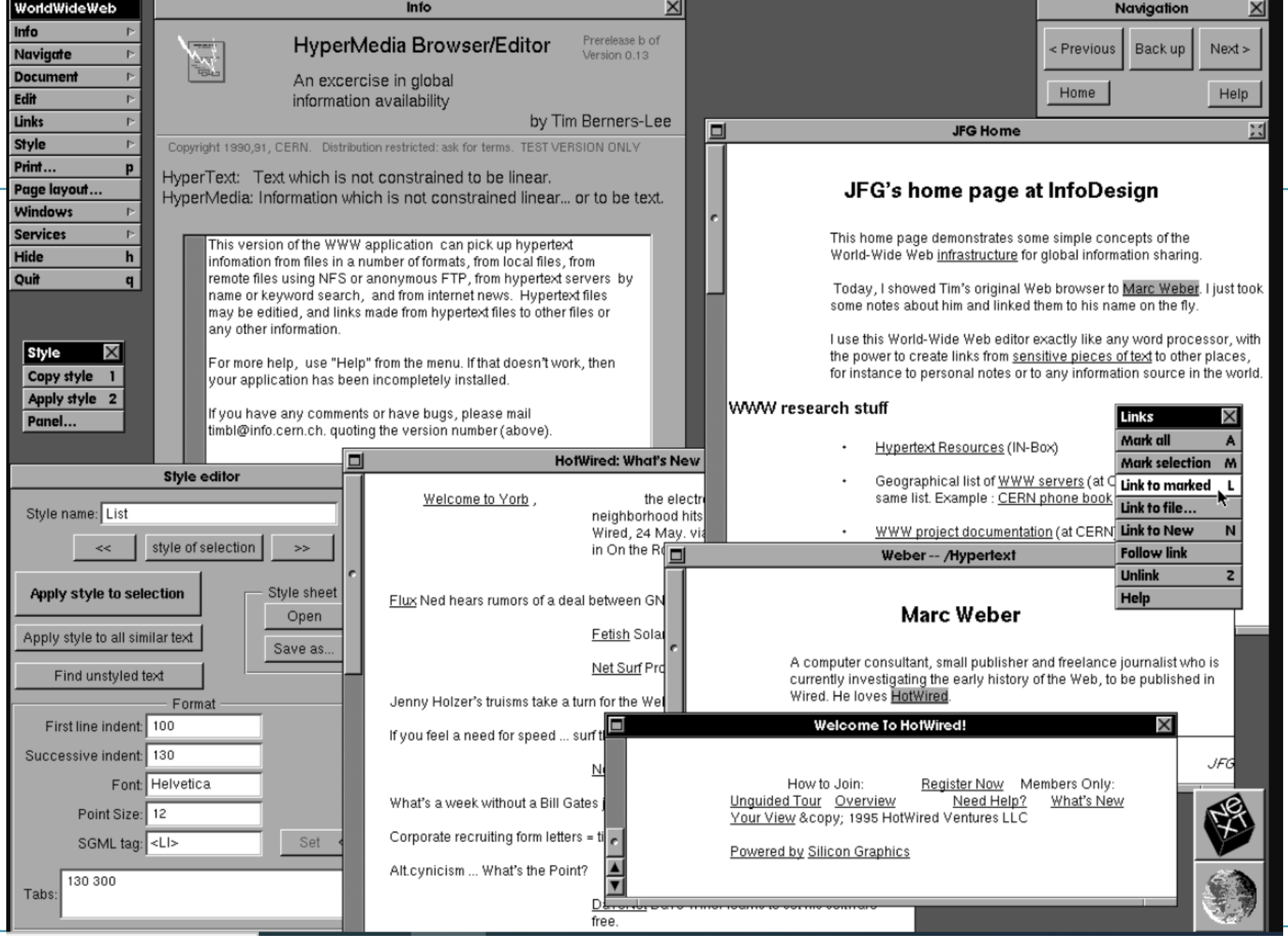
- A **web browser** is a software application for retrieving, presenting and traversing information resources on the World Wide Web.
- An *information resource* is identified by a Uniform Resource Identifier (URI) and may be a web page, image, video or other piece of content.
- Hyperlinks present in resources enable users easily to navigate their browsers to related resources.
- A web browser can also be defined as an application software or program designed to enable users to access, retrieve and view documents and other resources on the Internet.
- Although browsers are primarily intended to use the World Wide Web, they can also be used to access information provided by web servers in private networks or files in file systems.
- Browsers:
  - *Internet Explorer*
  - *Netscape Navigator*
  - *Mozilla Firefox*
  - *Chrome*
  - *Opera*
  - *Safari*

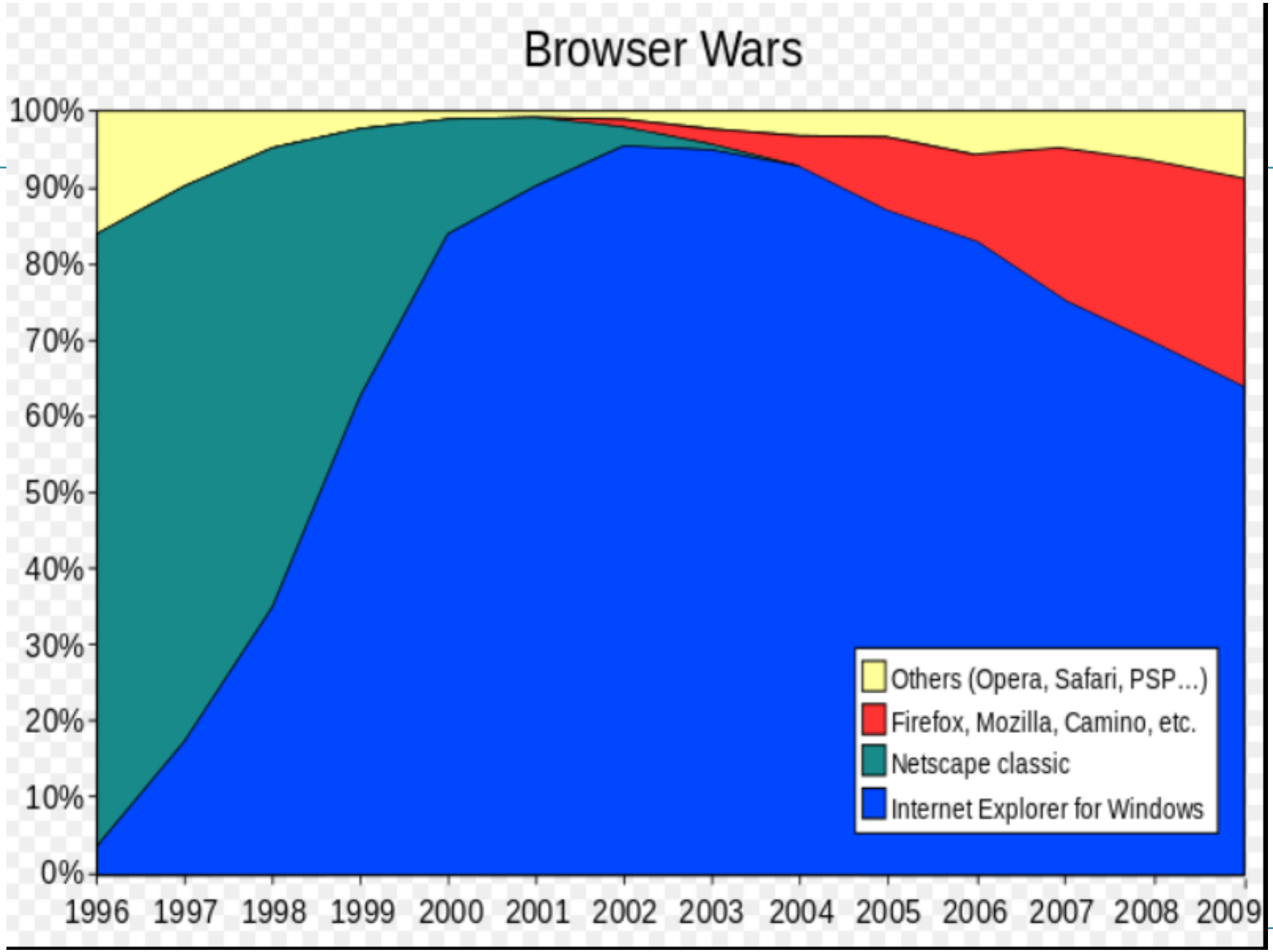
History - web browser

- The first web browser was invented in 1990 by Sir Tim Berners-Lee. It was called WorldWideWeb (no spaces) and was later renamed Nexus.
- In 1993, Mosaic (later Netscape), browser software was innovated by Marc Andreessen ,"the world's first popular browser which made the World Wide Web system easy to use and more accessible to the average person
- Andreessen, the leader of the Mosaic team at NCSA, soon started his own company, named Netscape, and released the Mosaic-influenced Netscape Navigator in 1994, which quickly became the world's most popular browser, accounting for 90% of all web use at its peak
- Microsoft responded with its Internet Explorer in 1995, also heavily influenced by Mosaic, initiating the industry's first browser war.
- Internet Explorer gained dominance in the web browser market; Internet Explorer usage share peaked at over 95% by 2002.



WorldWideWeb for NeXT, released in 1991, was the first web browser.


WorldWideWeb
Info
Navigate
Document
Edit
Links
Style
Print... p
Page layout...
Windows
Services
Hide h
Quit q
Info
HyperMedia Browser/Editor
Prerelease b of Version 0.13
An excercise in global information availability
by Tim Berners-Lee
Copyright 1990,91, CERN. Distribution restricted: ask for terms. TEST VERSION ONLY
HyperText: Text which is not constrained to be linear.
HyperMedia: Information which is not constrained linear... or to be text.
This version of the WWW application can pick up hypertext infomation from files in a number of formats, from local files, from remote files using NFS or anonymous FTP, from hypertext servers by name or keyword search, and from internet news. Hypertext files may be editied, and links made from hypertext files to other files or any other information.
For more help, use "Help" from the menu. If that doesn't work, then your application has been incompletely installed.
If you have any comments or have bugs, please mail timbl@info.cern.ch. quoting the version number (above).
Navigation
< Previous
Back up
Next >
Home
Help
JFG Home
JFG's home page at InfoDesign
This home page demonstrates some simple concepts of the World-Wide Web infrastructure for global information sharing.
Today, I showed Tim's original Web browser to Marc Weber. I just took some notes about him and linked them to his name on the fly.
I use this World-Wide Web editor exactly like any word processor, with the power to create links from sensitive pieces of text to other places, for instance to personal notes or to any information source in the world.
WWW research stuff
Hypertext Resources (IN-Box)
Geographical list of WWW servers (at C
same list. Example : CERN phone book
WWW project documentation (at CERN
Links
Mark all A
Mark selection M
Link to marked L
Link to file...
Link to New N
Follow link
Unlink Z
Help
Style
Copy style 1
Apply style 2
Panel...
Style editor
Style name: List
<<
style of selection
>>
Apply style to selection
Apply style to all similar text
Find unstyled text
Style sheet
Open
Save as...
Format
First line indent: 100
Successive indent: 130
Font: Helvetica
Point Size: 12
SGML tag: <LI>
Set
Tabs: 130 300
HotWired: What's New
Welcome to Yorb ,
Flux Ned hears rumors of a deal between GN
Fetish Sola
Net Surf Pro
Jenny Holzer's truisms take a turn for the We
If you feel a need for speed ... surf t
What's a week without a Bill Gates
Corporate recruiting form letters = ti
Alt.cynicism ... What's the Point?
free.
Weber -- /Hypertext
Marc Weber
A computer consultant, small publisher and freelance journalist who is currently investigating the early history of the Web, to be published in Wired. He loves HotWired.
Welcome To HotWired!
How to Join:
Register Now
Members Only:
Unguided Tour
Overview
Need Help?
What's New
Your View
© 1995 HotWired Ventures LLC
Powered by Silicon Graphics


Browser Wars
100%
90%
80%
70%
60%
50%
40%
30%
20%
10%
0%
1996
1997
1998
1999
2000
2001
2002
2003
2004
2005
2006
2007
2008
2009
Others (Opera, Safari, PSP...)
Firefox, Mozilla, Camino, etc.
Netscape classic
Internet Explorer for Windows

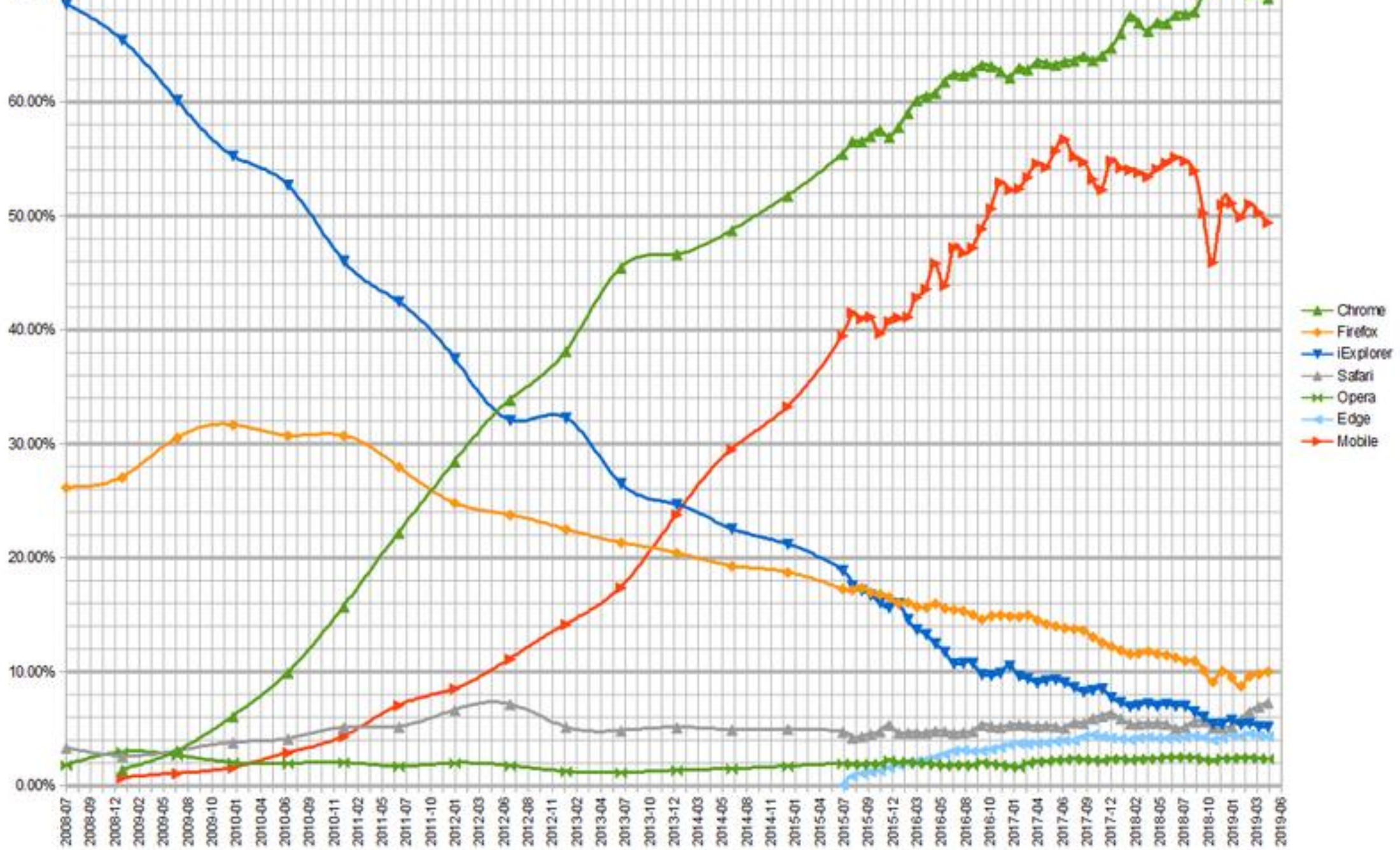
Contd..

- Opera was released in 1996; although it has never achieved widespread use, having less than 2% browser usage share as of February 2012 according to Net Applications.
- Its Opera-mini version has an additive share, in April 2011 amounting to 1.1% of overall browser use.
- In 1998, Netscape launched what was to become the Mozilla Foundation in an attempt to produce a competitive browser using the open source software model.
- That browser would eventually evolve into Firefox
- As of August 2011, Firefox has a 28% usage share.
- Apple's Safari had its first beta release in January 2003; as of April 2011, it had a dominant share of Apple-based web browsing, accounting for just over 7% of the entire browser market.

Contd..

- The most recent major entrant to the browser market is Chrome, first released in September 2008. Chrome's take-up has increased significantly year on year, by doubling its usage share from 8% to 16% by August 2011.

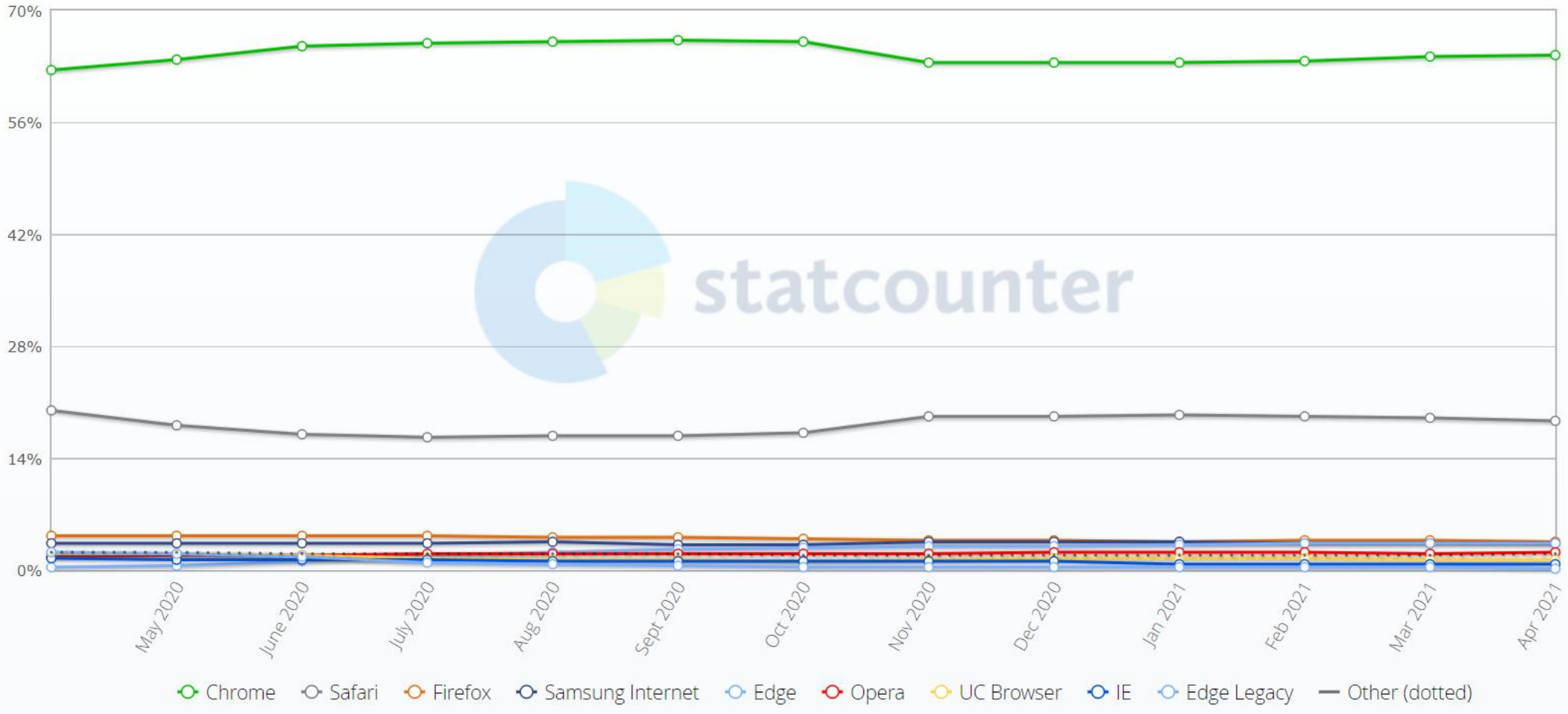
- This increase seems largely to be at the expense of Internet Explorer, whose share has tended to decrease from month to month.
- In December 2011, Chrome overtook Internet Explorer 8 as the most widely used web browser but still had lower usage than all versions of Internet Explorer combined.
- Chrome's user-base continued to grow and in May 2012, Chrome's usage passed the usage of all versions of Internet Explorer combined.
- By April 2014, Chrome's usage hit 45%.


60.00%
50.00%
40.00%
30.00%
20.00%
10.00%
0.00%
2008-07
2008-09
2008-12
2009-02
2009-05
2009-08
2009-10
2010-01
2010-04
2010-06
2010-09
2010-11
2011-02
2011-05
2011-07
2011-10
2012-01
2012-03
2012-06
2012-08
2012-11
2013-02
2013-04
2013-07
2013-10
2013-12
2014-03
2014-05
2014-08
2014-11
2015-01
2015-04
2015-07
2015-09
2015-12
2016-03
2016-05
2016-08
2016-10
2017-01
2017-04
2017-06
2017-09
2017-12
2018-02
2018-05
2018-07
2018-10
2019-01
2019-03
2019-06
Chrome
Firefox
iExplorer
Safari
Opera
Edge
Mobile

Browser Market Share Worldwide

Apr 2020 - Apr 2021

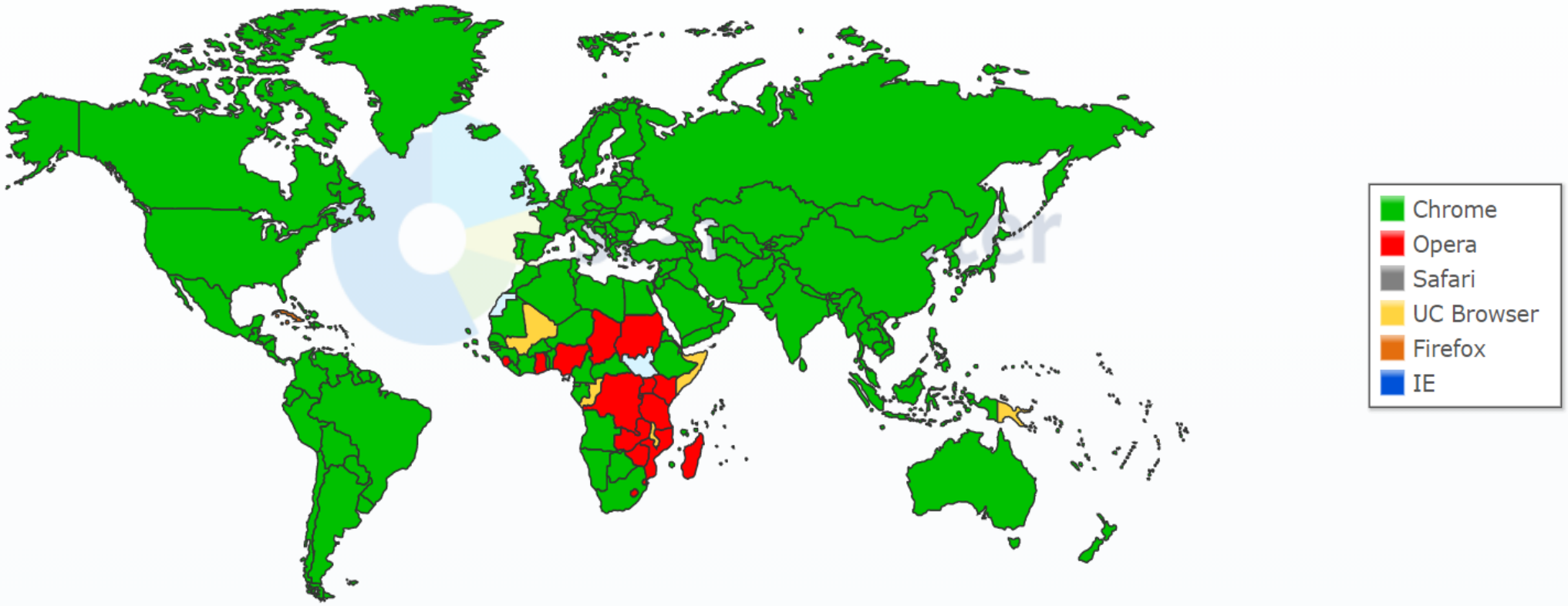
Edit Chart Data



Browser Market Share Worldwide

Apr 2017

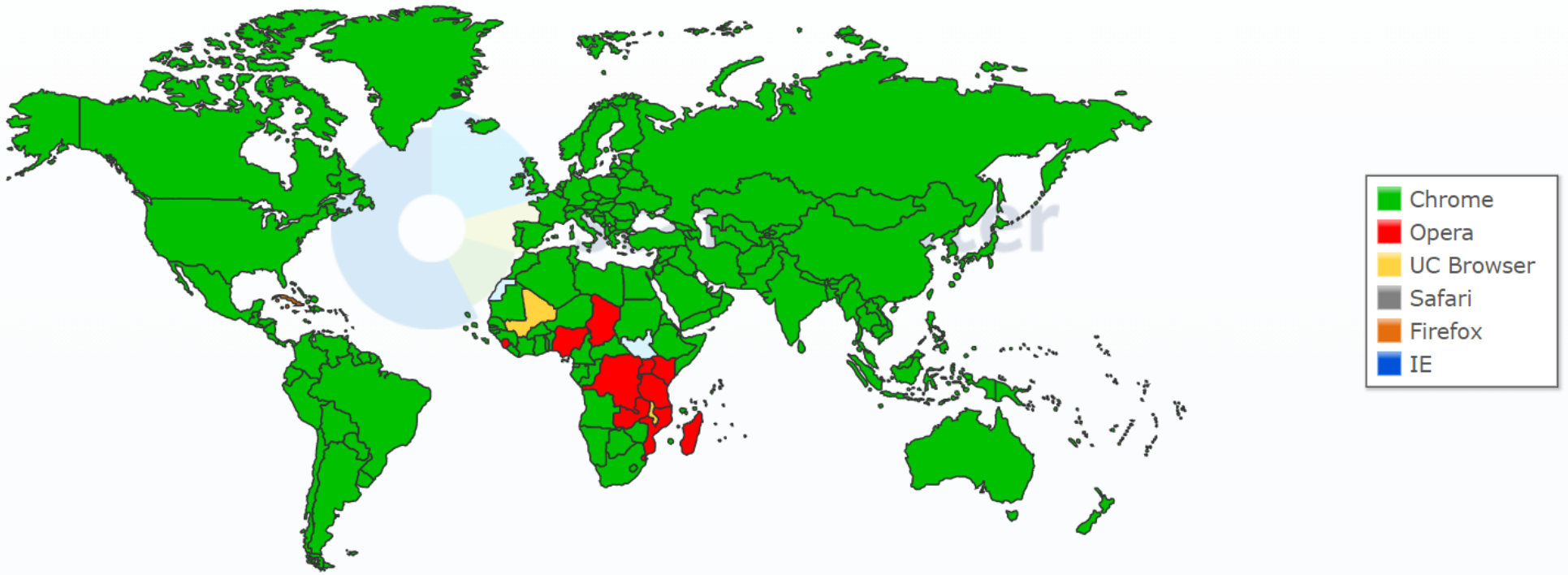
Edit Chart Data



Browser Market Share Worldwide

Apr 2018

Edit Chart Data



Browser Market Share Worldwide

Apr 2019

Edit Chart Data


StatCounter Global Stats
Browser Market Share Worldwide, Apr 2021
Chrome
Safari
Coc Coc

Web Browser – Function

- Function of a web browser is to bring information resources to the user ("retrieval" or "fetching"), allow them to view the information ("display", "rendering"), and then access other information ("navigation", "following links").
- This process begins when the user inputs a Uniform Resource Locator (URL), for example *http://en.wikipedia.org/*, into the browser.
- The prefix of the URL, the Uniform Resource Identifier or URI, determines how the URL will be interpreted.
- The most commonly used kind of URI starts with *http:* and identifies a resource to be Retrieved over the Hypertext Transfer Protocol (HTTP).
- Many browsers also support a variety of other prefixes, such as *https:* for HTTPS, *ftp:* for the File Transfer Protocol, and *file:* for local files.
- Prefixes that the web browser cannot directly handle are often handed off to another application entirely. For example, *mailto:* URIs are usually passed to the user's default e-mail application, and *news:* URIs are passed to the user's default newsgroup reader.

Contd..

- In the case of *http*, *https*, *file*, and others, once the resource has been Retrieved the web browser will display it.
- HTML and associated content (image files, formatting information such as CSS, etc.) is passed to the browser's layout engine to be transformed from markup to an interactive document, a process known as "rendering".
- Aside from HTML, web browsers can generally display any kind of content that can be part of a web page.
- Most browsers can display images, audio, video, and XML files, and often have plug-ins to support Flash applications and Java applets.
- Upon encountering a file of an unsupported type or a file that is set up to be downloaded rather than displayed, the browser prompts the user to save the file to disk.
- Information resources may contain hyperlinks to other information resources. Each link contains the URI of a resource to go to. When a link is clicked, the browser navigates to the resource indicated by the link's target URI, and the process of bringing content to the user begins again.

HTTP Commands

- GET :Request for obtaining a Web page
- HEAD :Request to read the header of a Web page. If a browser wants to know the last modified date of a Web page
- PUT :Request the server to store a Web page
- POST : Similar to PUT, but is used for updating a existing Web page
- ~~DELETE~~ : Remove a Web page (Obsolete)
- ~~LINK~~ : Connects two resources (Obsolete)
- ~~UNLINK~~ : Disconnects two resources (Obsolete)

- Note that GET is the most common command sent by a client browser as apart of the HTTP request to a Web server.
- This is because not many Web servers would allow a client to delete/add/link/unlink files.

Web Content

- Now the issue is how to encode web content for presentation and for processing.
- Programming languages are used to encode logic of processing data.
- Markup languages are used for encoding data for presentation and processing.

HTTP

- The Hypertext Transfer Protocol (HTTP) is an application-level protocol for distributed, collaborative, hypermedia information systems. This is the foundation for data communication for the World Wide Web (i.e. internet) since 1990. HTTP is a generic and stateless protocol which can be used for other purposes as well using extensions of its request methods, error codes, and headers
- Basically, HTTP is a TCP/IP based communication protocol, that is used to deliver data (HTML files, image files, query results, etc.) on the World Wide Web. The default port is TCP 80, but other ports can be used as well. It provides a standardized way for computers to communicate with each other.

HTTP Basic Features

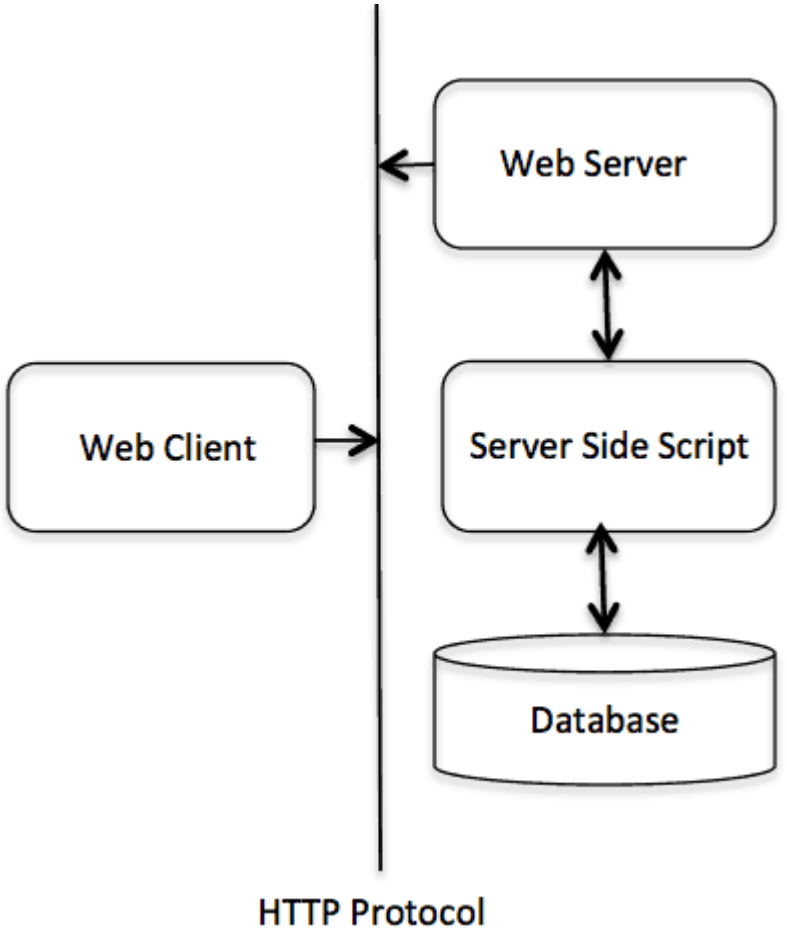
- **HTTP is connectionless:** The HTTP client, i.e., a browser initiates an HTTP request and after a request is made, the client waits for the response. The server processes the request and sends a response back after which client disconnect the connection. So client and server knows about each other during current request and response only. Further requests are made on new connection like client and server are new to each other.

- **HTTP is media independent:** It means, any type of data can be sent by HTTP as long as both the client and the server know how to handle the data content. It is required for the client as well as the server to specify the content type using appropriate MIME-type.

- **HTTP is stateless:** As mentioned above, HTTP is connectionless and it is a direct result of HTTP being a stateless protocol. The server and client are aware of each other only during a current request. Afterwards, both of them forget about each other. Due to this nature of the protocol, neither the client nor the browser can retain information between different requests across the web pages.

HTTP Basic Architecture

- The HTTP protocol is a request/response protocol based on the client/server based architecture where web browsers, search engines, etc. act like HTTP clients, and the Web server acts as a server.



HTTP - Messages

- HTTP is based on the client-server architecture model and a stateless request/response protocol that operates by exchanging messages across a reliable TCP/IP connection
- HTTP makes use of the Uniform Resource Identifier (URI) to identify a given resource and to establish a connection. Once the connection is established, HTTP messages are passed in a format similar to that used by the Internet mail [RFC5322] and the Multipurpose Internet Mail Extensions (MIME) [RFC2045]. These messages include requests from client to server and responses from server to client which will have the following format:

  <Request> | <Response> ;

  HTTP/1.1 messages

- HTTP requests and HTTP responses use a generic message format of RFC 822 for transferring the required data. This generic message format consists of the following four items.

1. *A Start-line*
2. *Zero or more header fields followed by CRLF ("Carriage Return" and "Line Feed")*
3. *An empty line (i.e., a line with nothing preceding the CRLF)*
4. *indicating the end of the header fields*
5. *Optionally a message-body*

Message Start-Line

- A start-line will have the following generic syntax:

- start-line = Request-Line | Status-Line

Thank You

TELNET as Alternative Web Browser

- TELNET software can be used as poor alternative to a web browser.
- It can be used to request an HTML page from Web server and then interpret the HTML page and display its contents on the user's screen
- If the user's , for some reason, does not have Web Browser, but knows how to enter TELNET commands, and has some software that can interpret HTML pages.
- In such case, the user can actually type TELNET commands that mimic the function of Web Browser, by requesting web pages from a web server.
- However, the point to note is that TELNET can actually be used to send HTTP commands to a Web Server

telnet 172.16.0.2 80

GET /index.html