CHAPTER 5
NAMING

INTRODUCTION

- Names play a very important role in all computer systems.
- They are used to share resources, to uniquely identify entities, to refer to locations, and more.
- A naming convention is a convention for naming entity.
- Name Resolution is **the process of resolving the name of a computer on a network into its network address**.

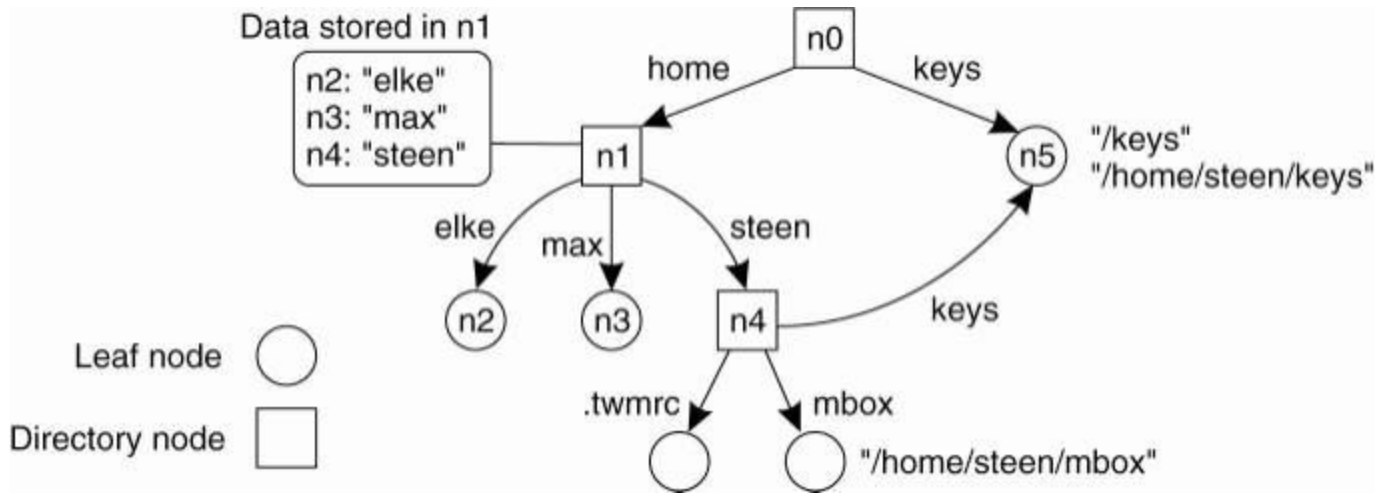
NAMING SYSTEM

Naming Service is a specific service whose aim is :

–to provide a consistent and uniform naming of resources,

–thus allowing other programs or services to localize them and obtain the required metadata for interacting with them.

- For example, a system that implements the DNS is a naming system.


Data stored in n1
n2: "elke"
n3: "max"
n4: "steen"
n0
home
keys
n1
n5
"/keys"
"/home/steen/keys"
elke
max
steen
n2
n3
n4
keys
Leaf node
Directory node
.twmrc
mbox
"/home/steen/mbox"

- Typical examples of such names include those for file systems and the World Wide Web.

- Second, names are used to locate entities in a way that is independent of their current location.

NAMES, IDENTIFIERS, AND ADDRESSES

- A name in a distributed system is a **string of bits** or characters that is **used to refer to an entity.**
- An **entity in a distributed system can be practically anything.**
- Typical examples include resources such as **hosts, printers, disks, and files.**
-
- Other well-known examples of entities that are often explicitly named are:

  – processes, users, mailboxes, newsgroups, Web pages, graphical windows, messages, network connections, and so on.

- Entities can be operated on.
- To operate on an entity, it is necessary to access it, for which we need an access point.
- **The name of an access point is called an address.**
- The address of an access point of an entity is also simply called an **address of that entity.**

EXAMPLE OF ENTITY

- For example, a resource such as a printer offers an interface containing operations for printing a document, requesting the status of a print job, and the like.
-

- An entity can offer more than one access point.
- **As a comparison, a telephone can be viewed as an access point of a person, whereas the telephone number corresponds to an address.**
- **In a distributed system, a typical example of an access point is a host running a specific server, with its address formed by the combination of, for example, an IP address and port number (i.e., the server's transport-level address).**

- An entity may change its access points in the course of time.
- For example, when a mobile computer moves to another location, it is often assigned a different IP address than the one it had before.
- Likewise, when a person moves to another city or country, it is often necessary to change telephone numbers as well.

IDENTIFIER

- A true identifier is a name that has the following properties :
- An identifier refers to at most one entity.
- An identifier – **a name that is used to uniquely identify an entity**

DESIRABLE CHARACTERISTICS OF NAMING SYSTEM

1. Location transparency

the name of an object should not reveal any hint as to the physical location of the object.

2. Location independency

movement and relocation of objects dynamically among the various nodes of a system.

3. Scalability

Distributed systems vary in size ranging from one with a few nodes to one with many nodes

5. Multiple user-defined names for the same object

For a shared object, it is desirable that different users of the object can use their own convenient names for accessing it

6. Group naming

A naming system should allow many different objects to be identified by the same name.

7. Meaningful names

A name can be simply any character string identifying some object.

8. Performance

The most important performance measurement of a naming system is the amount of time needed to map an object's name to its attributes, such as its location.

9. Fault tolerance

A naming system should be capable of tolerating, to some extent, faults that occur due to the failure of a node or a communication link in a distributed system network.

TYPES OF NAMING SYSTEM

-FLAT NAMING

-STRUCTURED NAMING

-ATTRIBUTE BASED NAMING

FLAT NAMING SYSTEM

In a flat naming system, each name is unique and typically not hierarchical.

Names are usually strings or identifiers with no inherent structure or organization beyond their uniqueness.

Advantages:

- Simplicity: The system is easy to understand and implement.
- Direct Access: Names can be resolved directly without navigating a hierarchy.
- Uniqueness: Each name is unique, avoiding ambiguity.

Examples:

- MAC addresses in networking.
- GUIDs (Globally Unique Identifiers).

STRUCTURED NAMING

Structured naming systems organize names into a hierarchy or other structured format.

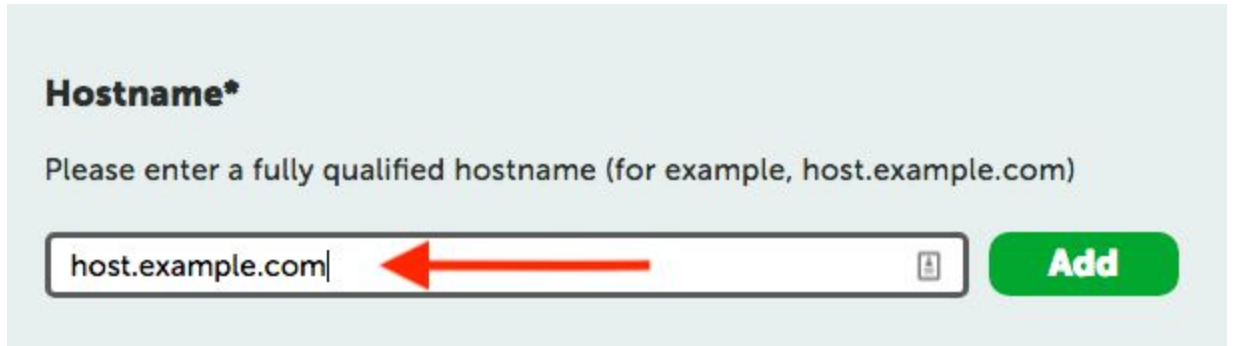
Names often reflect the structure of the organization or the relationships between entities.

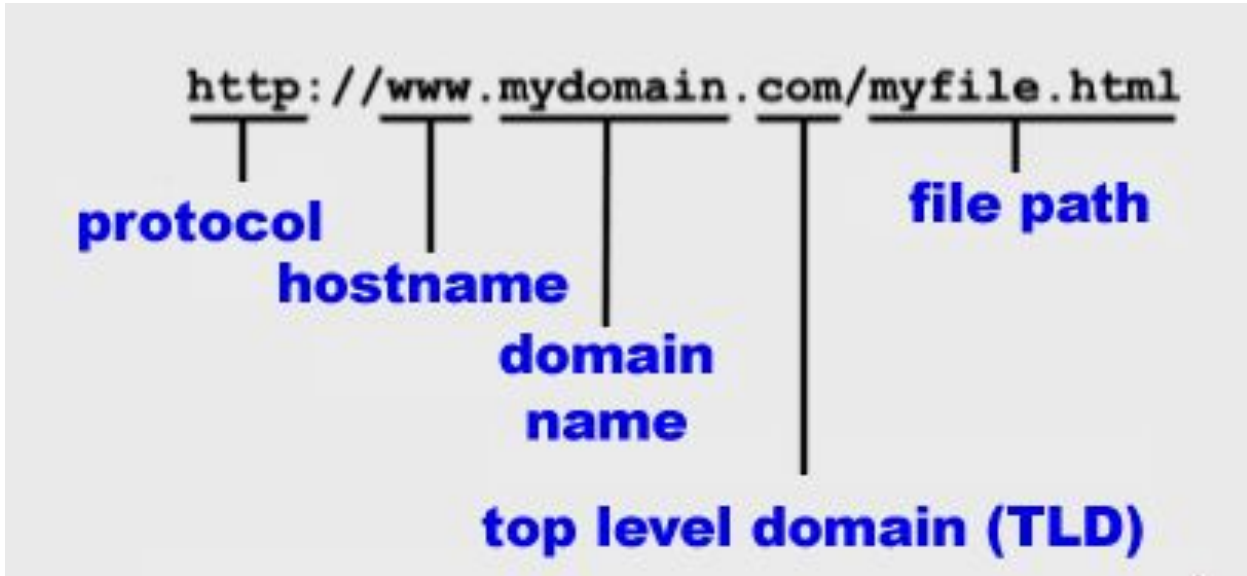
Advantages:

- Organization: Names provide context and information about the structure.
- Scalability: Hierarchical organization can help manage a large number of names.
- Navigability: Easier to navigate and locate resources based on the structure.

Examples:

- File paths in a filesystem (e.g., `/home/user/documents/file.txt`).
- Domain names in DNS (e.g., `www.example.com`).


Hostname*
Please enter a fully qualified hostname (for example, host.example.com)
host.example.com
Add


http://www.mydomain.com/myfile.html
protocol
hostname
domain name
top level domain (TLD)
file path

ATTRIBUTE BASED NAMING

Attribute-based naming (ABN) is a method of identifying and retrieving resources based on a set of attributes rather than a single, unique identifier.

This approach allows for more flexible and powerful querying capabilities, as resources can be found based on various properties or characteristics.

Characteristics of Attribute-Based Naming

1. **Attributes and Values**: Resources are described using attributes, each of which has one or more values. For example, a file might have attributes like `type`, `size`, `date created`, and `owner`.
2. **Flexible Queries**: Users can query the system for resources that match specific attribute criteria. For instance, a search might specify `type = document` and `date created > 2023-01-01`.

A Digital Photo Management System

Imagine a digital photo management system where users can store and retrieve their photos. Instead of organizing photos in a rigid folder structure, the system uses attribute-based naming.

Attributes for Photos:

- `date_taken`: The date the photo was taken.
- `location`: The geographic location where the photo was taken.
- `people`: Names of people in the photo.
- `event`: The event during which the photo was taken (e.g., "vacation", "birthday").
- `tags`: User-defined tags describing the photo (e.g., "sunset", "beach").

Storing Photos: When a user uploads a photo, they can add attributes to it. For example:

- Photo 1:
  - `date_taken = 2023-06-15`
  - `location = Paris`
  - `people = [Alice, Bob]`
  - `event = vacation`
  - `tags = [Eiffel Tower, night]`

Attribute	Value
Country	NL
Locality	Amsterdam
Organization	Vrije Universiteit
OrganizationalUnit	Comp. Sc.
CommonName	Main server
Host_Name	star
Host_Address	192.31.231.42

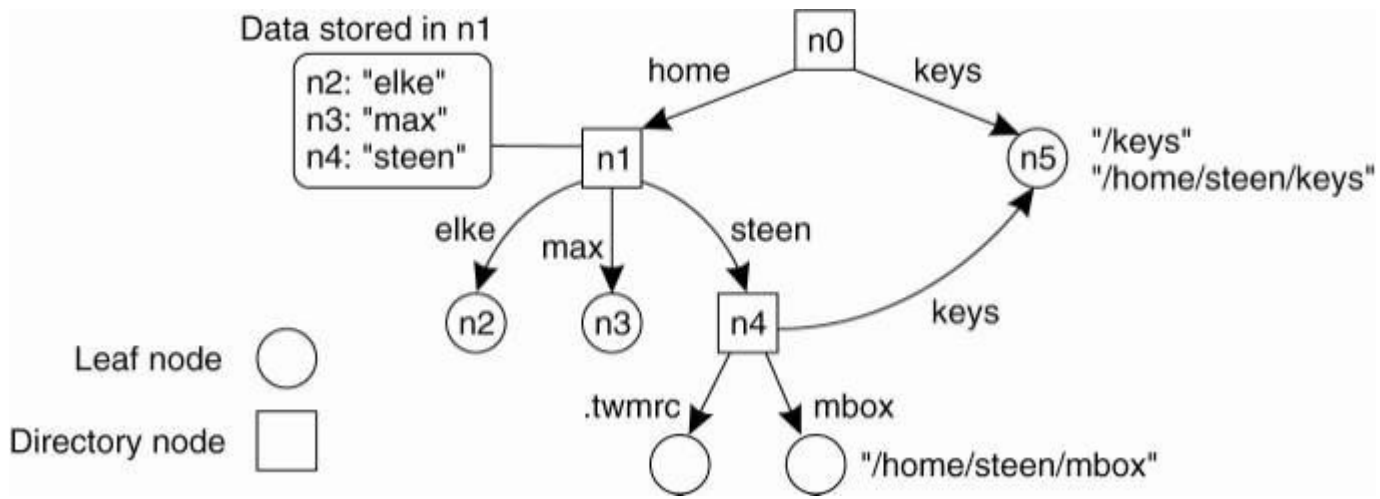
Attribute	Value
Country	NL
Locality	Amsterdam
Organization	Vrije Universiteit
OrganizationalUnit	Comp. Sc.
CommonName	Main server
Host_Name	zephyr
Host_Address	137.37.20.10

(b)

5.3.1. Name Spaces

- Names are commonly organized into what is called a name space.
- Name spaces for structured names can be represented as a labeled, directed graph with two types of nodes.
- A leaf node represents a named entity and has the property that it has no outgoing edges.
- A **leaf node** generally stores information on the entity it is representing—for example, its address—so **that a client can access it.**

- In contrast to a leaf node, a directory node has a number of outgoing edges, each labeled with a name, as shown in Fig.
- A **directory node** stores a table in which an outgoing edge is represented as a pair (edge label, node identifier).
- Such a table is called a directory table.


Data stored in n1
n2: "elke"
n3: "max"
n4: "steen"
n0
home
keys
n1
n5
"/keys"
"/home/steen/keys"
elke
max
steen
n2
n3
n4
keys
Leaf node
Directory node
.twmrc
mbox
"/home/steen/mbox"

- Although it is possible for a naming graph to have several root nodes, for simplicity, many naming systems have only one.
- Each path in a naming graph can be referred to by the sequence of labels corresponding to the edges in that path, such as
- N:<label-1, label-2, ..., label-n>
- where N refers to the first node in the path.
- Such a sequence is called a path name. If the first node in a path name is the root of the naming graph, it is called an absolute path name.
- Otherwise, it is called a relative path name.

- However, instead of writing the sequence of edge labels to reprepresent a path name, path names in file systems are generally represented as a single string in which the labels are separated by a special separator character, such as a slash ("/").
- This character is also used to indicate whether a path name is absolute. For example, in Fig. , instead of using n0:<home, steen, mbox>, that is, the actual path name, it is common practice to use its string representation /home/steen/mbox.

NAME RESOLUTION

- Name resolution: **process to determine the actual entity that a name refers to**.

- To operate on an entity, we need to access it at an access point.
- Name Space offer a convenient mechanism for storing and retrieving information about entities by means of names.

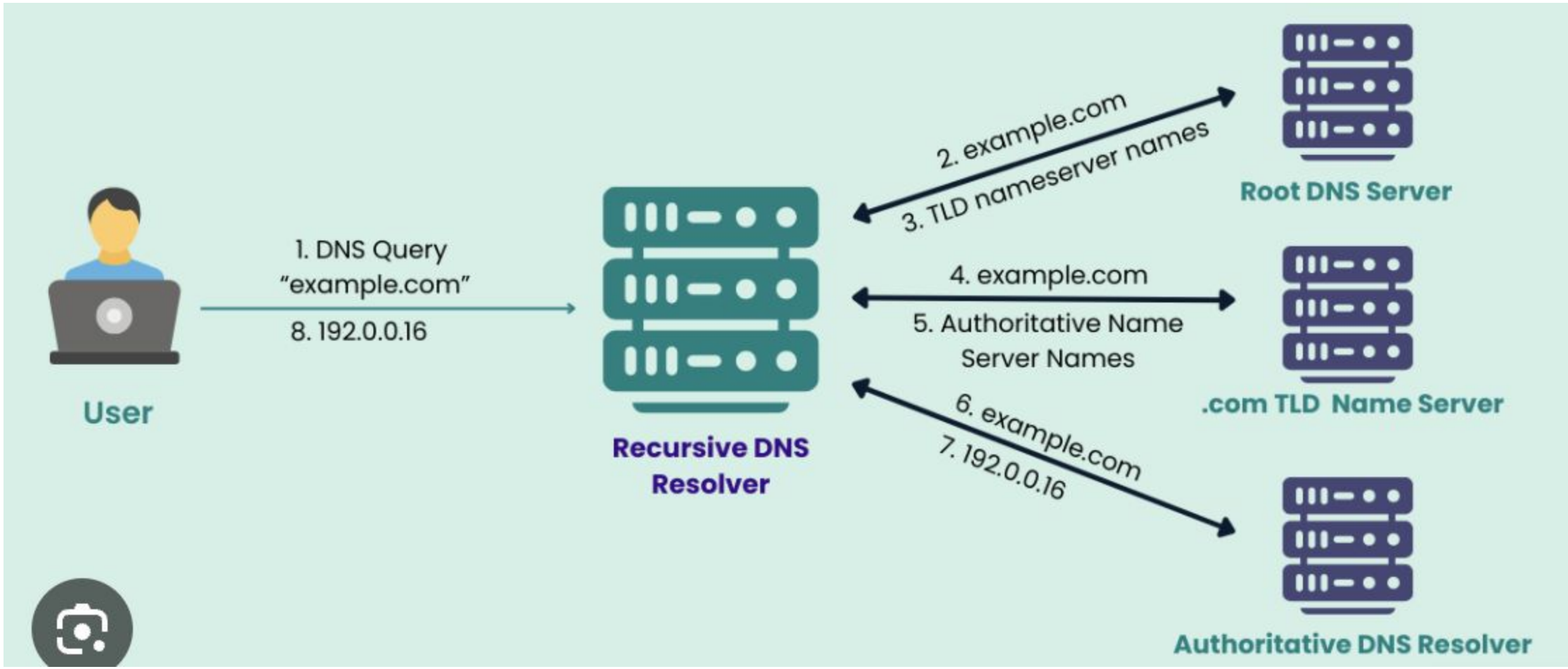
- Given a path name, name space can look up information stored in the node referred to by that name.
- This process of looking up a name is called **name resolution.**

Domain Name System (DNS)

DNS translates human-readable domain names (like `www.google.com`) into the numerical IP addresses that computers use to communicate. This translation is essential for accessing websites and other internet resources.

Process:

1. **User Request**: A user types `www.example.com` into a web browser.
2. **Local Cache**: The browser first checks its local cache to see if it already knows the IP address for `www.example.com`.
3. **DNS Resolver**: If the address is not in the local cache, the browser contacts a DNS resolver (usually provided by the Internet Service Provider).


1. DNS Query
"example.com"
8. 192.0.0.16
User
Recursive DNS Resolver
2. example.com
3. TLD nameserver names
Root DNS Server
4. example.com
5. Authoritative Name Server Names
.com TLD Name Server
6. example.com
7. 192.0.0.16
Authoritative DNS Resolver

4. Recursive Query: The DNS resolver performs a recursive query. It starts by querying the root DNS servers, which then refer it to the appropriate top-level domain (TLD) servers (e.g., `.com` servers).

5. Authoritative DNS Server: The TLD servers direct the resolver to the authoritative DNS server for `example.com`.

6. IP Address: The authoritative DNS server returns the IP address associated with `www.example.com`.

7. Access Resource: The browser uses the IP address to establish a connection to the web server and retrieves the web page.

-

Example:

- User inputs: `www.example.com`
- DNS resolver queries and eventually retrieves: `93.184.216.34`
- Browser connects to: `93.184.216.34` to load the website.

HIERARCHAIL STRUCTURE

- To explain how name resolution works, let us consider a path name such as N:<label1, label2, ..., labeln>.
- Resolution of this name starts at node N of the naming graph, where the name label1 is looked up in the directory table, and which returns the identifier of the node to which label1 refers.
- Resolution then continues at the identified node by looking up the name label2 in its directory table, and so on.

CLOSURE MECHANISM

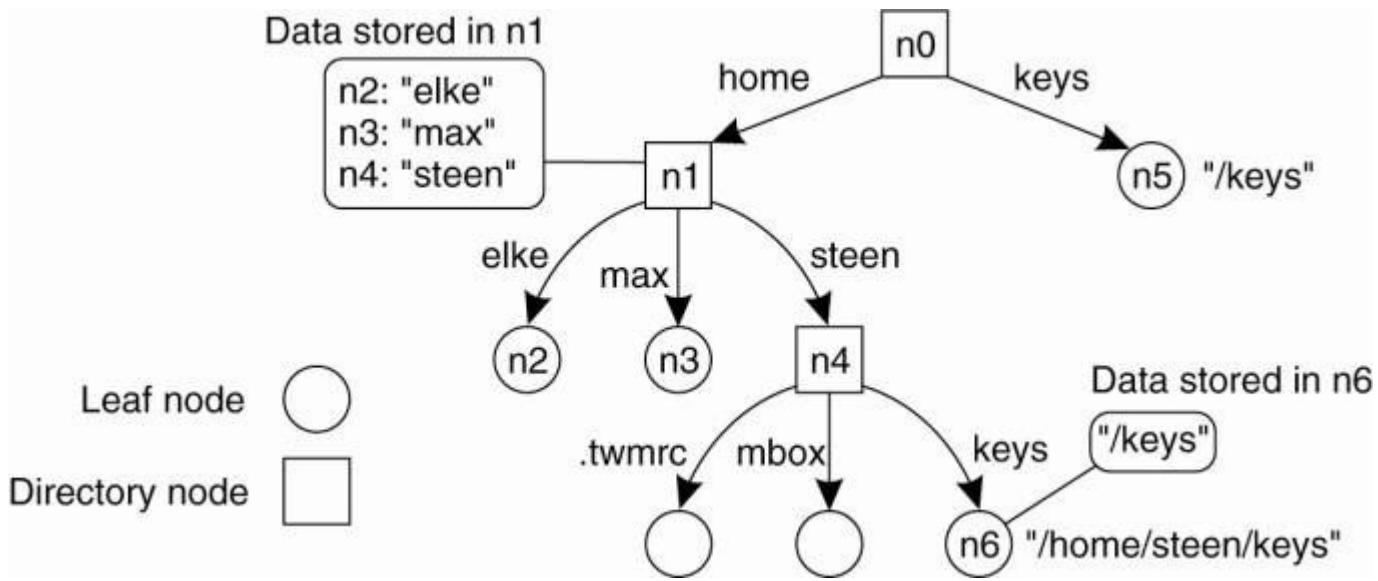
- Name resolution can take place only if we know how and where to start.
- In our example, the starting node was given, and we assumed we had access to its directory table.
- Knowing how and where to start name resolution is generally referred to as a c**losure mechanism.**
- Essentially, **a closure mechanism deals with selecting the initial node in a name space from which name resolution is to start.**

- A completely different example is the use of the string "003120443078 4".
- Many people will not know what to do with these numbers, unless they are told that the sequence is a telephone number.
- That information is enough to start the resolution process, in particular, by dialing the number.
- The telephone system subsequently does the rest.

• LINKING AND MOUNTING

- Strongly related to name resolution is the use of aliases.
- An alias is another name for the same entity.
- In terms of naming graphs, there are basically two different ways to implement an alias.
- The first approach is to simply allow multiple absolute paths names to refer to the same node in a naming graph.
- This approach is illustrated in Fig. in which node n5 can be referred to by two different path names.
- *Note: An absolute path name* ***represents the complete name of a directory or file from the /(root) directory downward***.

- The second approach is to represent an entity by a leaf node, say N, but instead of storing the address or state of that entity, the node stores an absolute path name.


Data stored in n1
n2: "elke"
n3: "max"
n4: "steen"
n0
home
keys
n1
n5
"/keys"
elke
max
steen
n2
n3
n4
Data stored in n6
"/keys"
Leaf node
Directory node
.twmrc
mbox
keys
n6
"/home/steen/keys"

DIRECTORY SERVICES

A directory service is a database for storing and maintaining information about users and resources.

Directory Services are often referred to as directories, user stores, Identity Stores, or LDAP Directory, and they store information such as usernames, passwords, user preferences, information about devices, and more.

An example of a directory service is a domain name system (DNS) server.

A DNS server maps computer hostnames to IP addresses.

Thus, all of the computing resources (hosts) become clients of the DNS server.

Directory services can be either centralized or distributed. In a centralized service, all the information is stored on one server. With a distributed directory service, information is stored on multiple servers.

LDAP

LDAP (Lightweight Directory Access Protocol) is a software protocol for enabling anyone to locate data about organizations, individuals and other resources such as files and devices in a network -- whether on the public internet or a corporate intranet.

LDAP has two main goals: to store data in the LDAP directory and authenticate users to access the directory.

LDAP can also tackle authentication, so users can sign on just once and access many different files on the server. LDAP is a protocol, so it doesn't specify how directory programs work.

Directory Structure:

- LDAP directories are organized in a hierarchical tree structure. Each entry in the directory is identified by a distinguished name (DN), which uniquely identifies the entry in the directory hierarchy.

Entries and Attributes:

- An entry represents a single unit of information, such as a user, group, or device. Each entry consists of a set of attributes, each of which has a type and one or more values.

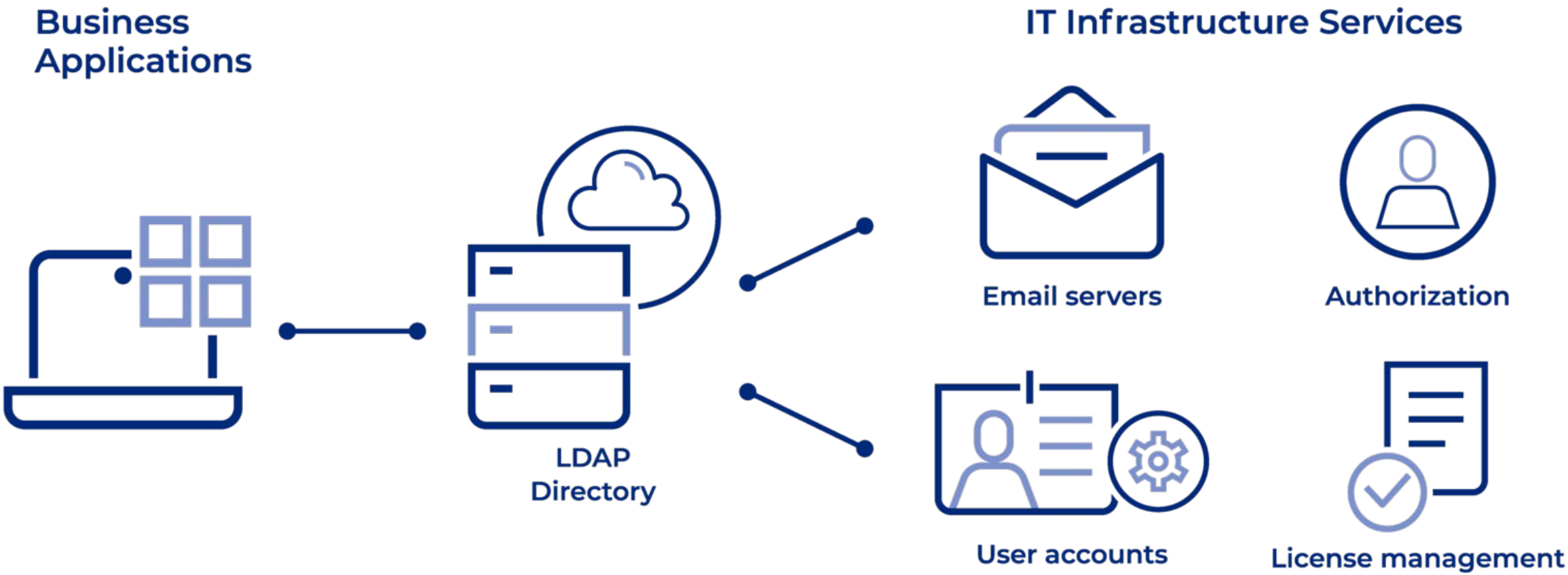
1. **Distinguished Names (DN)**:
    - The DN is a unique identifier for an entry in the directory. It is composed of attribute-value pairs separated by commas.
2. **Object Classes**:
    - Object classes define the types of entries and the set of attributes that entries of that type can have.
3. **Schema**:
    - The schema defines the rules for the directory, including object classes, attribute types, and syntax rules.

LDAP OPERATION

1. **Bind**:
   - Establishes a connection to the LDAP server and authenticates the client.
2. **Search**:
   - Searches for entries in the directory that match specified criteria. It can be used to retrieve specific attributes or a set of entries.
3. **Compare**:
   - Compares an attribute value in an entry to a given value to see if they match.
4. **Add**:
   - Adds a new entry to the directory.

1. **Delete**:
    - Removes an entry from the directory.
2. **Modify**:
    - Updates an existing entry by adding, deleting, or modifying its attributes.
3. **Modify DN**:
    - Changes the distinguished name of an entry, effectively moving it within the directory hierarchy.

How LDAP Works



okta

- Session connection. The user connects to the server via an LDAP port.
- Request. The user submits a query, such as an email lookup, to the server.
- Response. The LDAP protocol queries the directory, finds the information, and delivers it to the user.
- Completion. The user disconnects from the LDAP port.

Example LDAP Use Case:

Scenario: User Authentication in an Enterprise Environment.

Steps:

1. **User Login**: A user attempts to log in to a corporate application.
2. **Bind Operation**: The application performs an LDAP bind operation to authenticate the user. It sends the user's DN and password to the LDAP server.
3. **Authentication**: The LDAP server verifies the user's credentials. If they are correct, the bind operation succeeds, and the user is authenticated.
4. **Search Operation**: The application performs an LDAP search operation to retrieve additional user information, such as group memberships and profile details.
5. **Access Control**: Based on the retrieved information, the application determines the user's access rights and grants or denies access to resources accordingly.

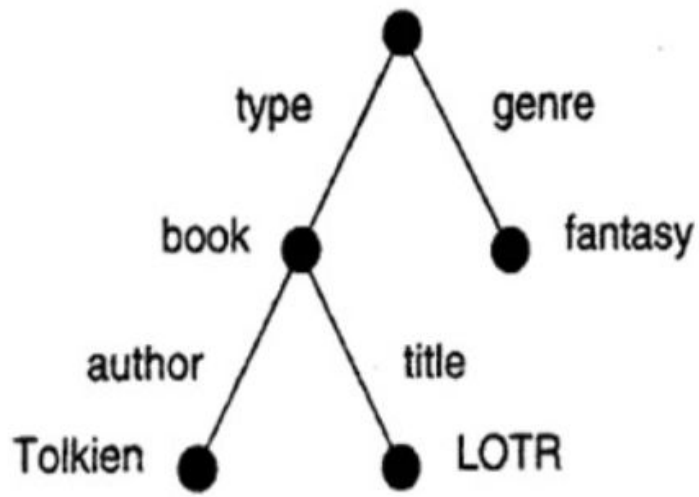
5.4.3 Decentralized Implementations

- Decentralized attribute-based naming systems
  - Peer to peer
- The key issue here is that *(attribute, value) pairs need to be efficiently mapped so that searching can* be done efficiently, that is, by avoiding an exhaustive search through the entire attribute space

Mapping to Distributed Hash Tables

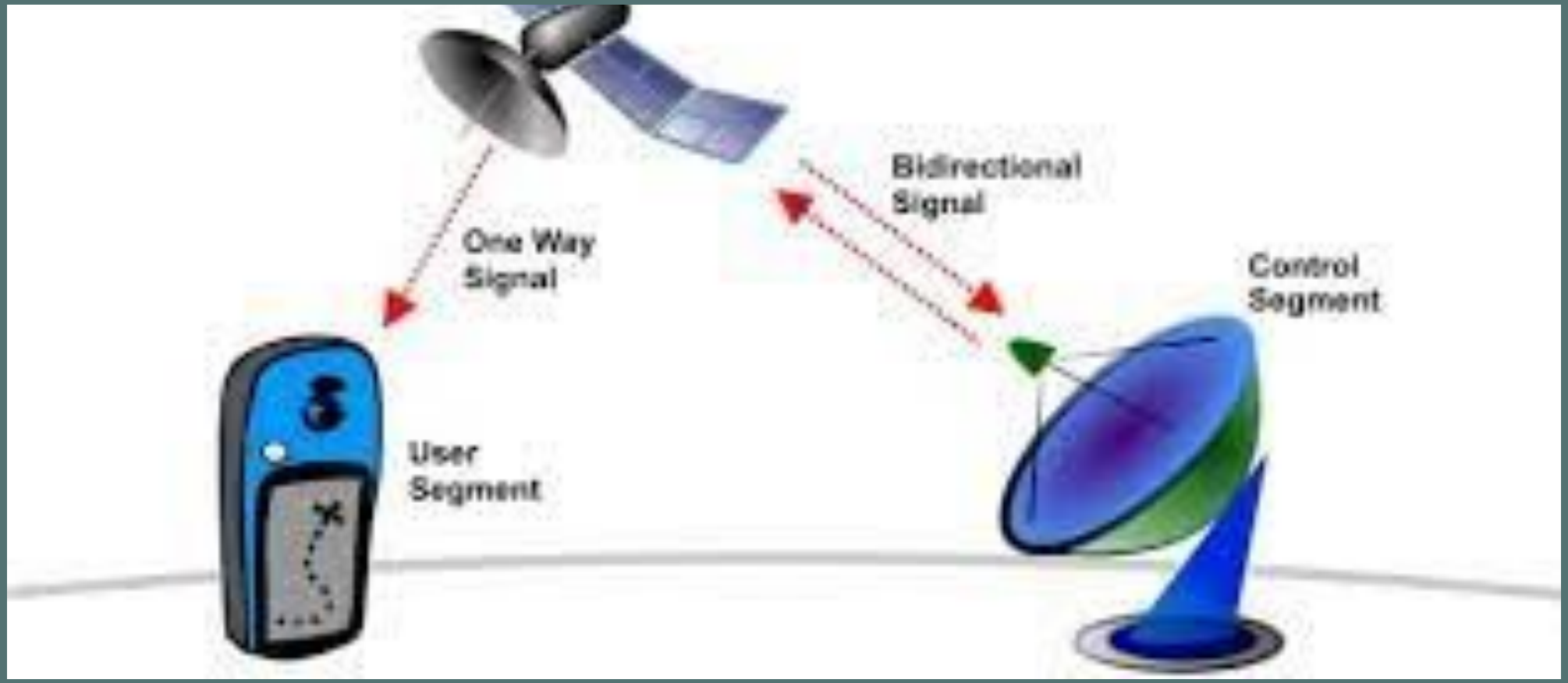
- Queries consist of a conjunction of pairs
  - specifies a list of attributes, along with the unique value
  - Each entity is assumed to be described by hierarchically organized attributes

```
description {
    type = book
    description {
        author = Tolkien
        title = LOTR
    }
    genre = fantasy
}
```



GPS

- Stands for "Global Positioning System." GPS is a satellite navigation system used to determine the ground position of an object.
- GPS technology was first used by the United States military in the 1960s and expanded into civilian use over the next few decades.
- Today, GPS receivers are included in many commercial products, such as automobiles, smartphones, exercise watches, and GIS devices.
- The GPS system includes 24 satellites deployed in space about 12,000 miles (19,300 kilometers) above the earth's surface.
- They orbit the earth once every 12 hours at an extremely fast pace of roughly 7,000 miles per hour (11,200 kilometers per hour). The satellites are evenly spread out so that four satellites are accessible via direct line-of-sight from anywhere on the globe.


One Way
Signal
Bidirectional
Signal
Control
Segment
User
Segment

- Each GPS satellite broadcasts a message that includes the satellite's current position, orbit, and exact time.
- A GPS receiver combines the broadcasts from multiple satellites to calculate its exact position using a process called triangulation.
- Three satellites are required in order to determine a receiver's location, though a connection to four satellites is ideal since it provides greater accuracy.

- 21 GPS satellites and three spare satellites are in orbit at 10,600 miles above the Earth. The satellites are spaced so that four satellites will be above the horizon from any point on Earth.
- Each satellite contains a computer, an atomic clock and a radio.
- With an understanding of its orbit and the clock, it continually broadcasts its changing position and time.
- On the ground, any GPS receiver contains a computer that "triangulates" its position by getting bearings from three satellites. If the receiver is also equipped with a display screen and a map, the position can be shown as well.
- If a fourth satellite can be received, the receiver/computer can figure out the altitude as well as the geographic position.
- If you are moving, your receiver may also calculate your speed and direction of travel and give you estimated arrival times to specified destinations.

ABSOLUTE AND RELATIVE PATH

Absolute Path

An absolute path (or full path) specifies a location from the root directory (the topmost directory in a hierarchy). It provides the complete address to a file or directory, ensuring that the location is always the same regardless of the current working directory.

Characteristics

- **Starts from the Root**: It always begins from the root directory (e.g., `/` in Unix-like systems or `C:\` in Windows).
- **Unambiguous**: Since it specifies the complete path, it uniquely identifies a file or directory.
- **Length**: Typically longer and more detailed.

Examples

- **Unix-like Systems**:
  `/home/user/documents/report.txt`
- **Windows**:
  `C:\Users\user\Documents\report.txt`

Relative Path

A relative path specifies a location in relation to the current working directory. It does not start from the root directory but rather from the current directory context.

Characteristics

- **Context-Dependent**: The meaning of a relative path changes based on the current working directory.
- **Shorter and Flexible**: Often shorter than absolute paths and more convenient when working within a specific directory structure.

From Current Directory: `documents/report.txt`

Example in Unix-like System

Absolute Path:

1. `/etc/nginx/nginx.conf`
    - Starts from the root directory `/`.
    - Always points to the same file regardless of the current working directory.

Relative Path:

1. `nginx/nginx.conf`
    - If the current directory is `/etc`, this path points to `/etc/nginx/nginx.conf`.

Absolute Path

http://www.computerhope.com/logo.gif

Relative Path

Relative path of above file locally

/logo.gif

Location of current HTML file

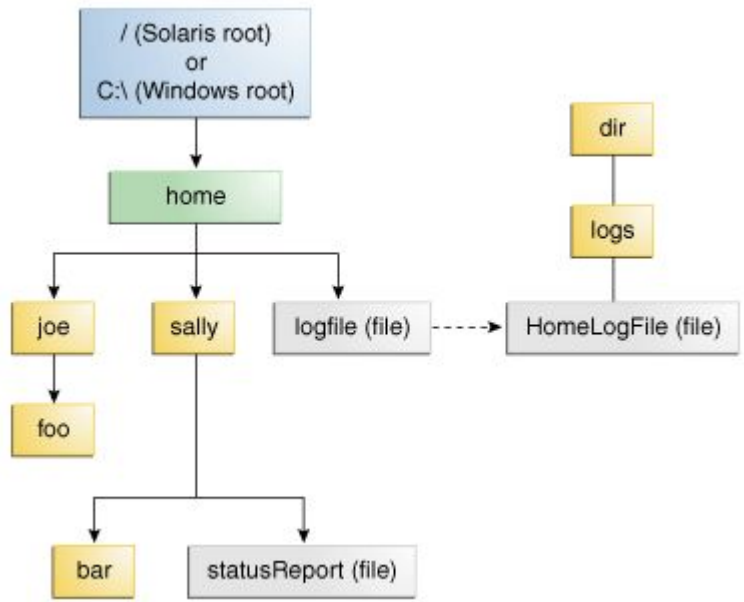
http://www.computerhope.com/issues/ch001708.htm

Which is locally

/issues/ch001708.htm

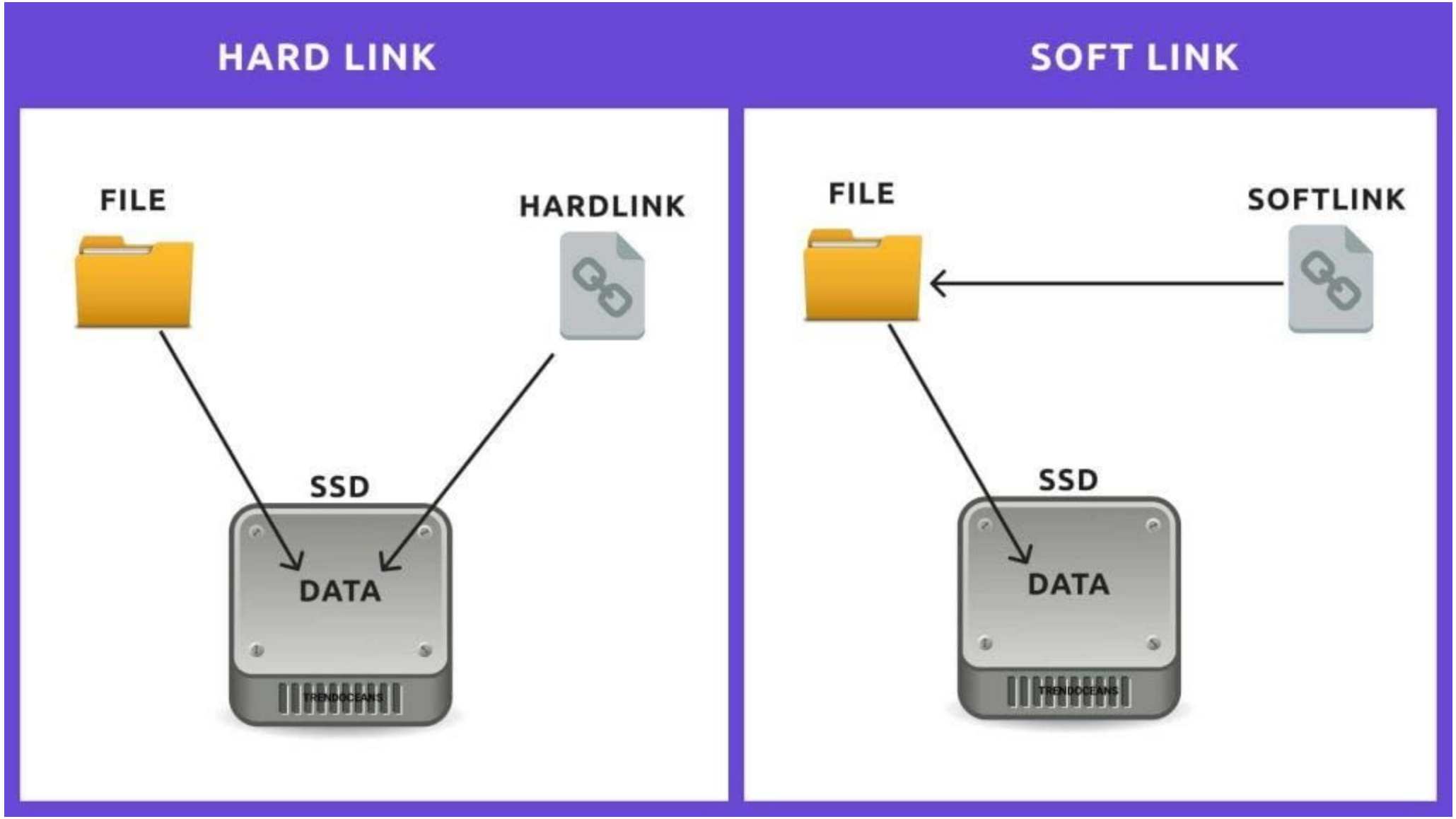
Relative path to file from current HTML file

../logo.gif


/ (Solaris root)
or
C:\ (Windows root)
home
joe
sally
logfile (file)
foo
bar
statusReport (file)
dir
logs
HomeLogFile (file)

HARD LINK AND SOFT LINK

- **A hard link is a file all its own**, and the file references or points to the **exact spot on a hard drive** where the Inode stores the data.
- A soft link isn't a separate file, **it points to the name of the original file,** rather than to a spot on the hard drive.


HARD LINK
FILE
HARDLINK
SSD
DATA
TRENDOCEANS
SOFT LINK
FILE
SOFTLINK
SSD
DATA
TRENDOCEANS

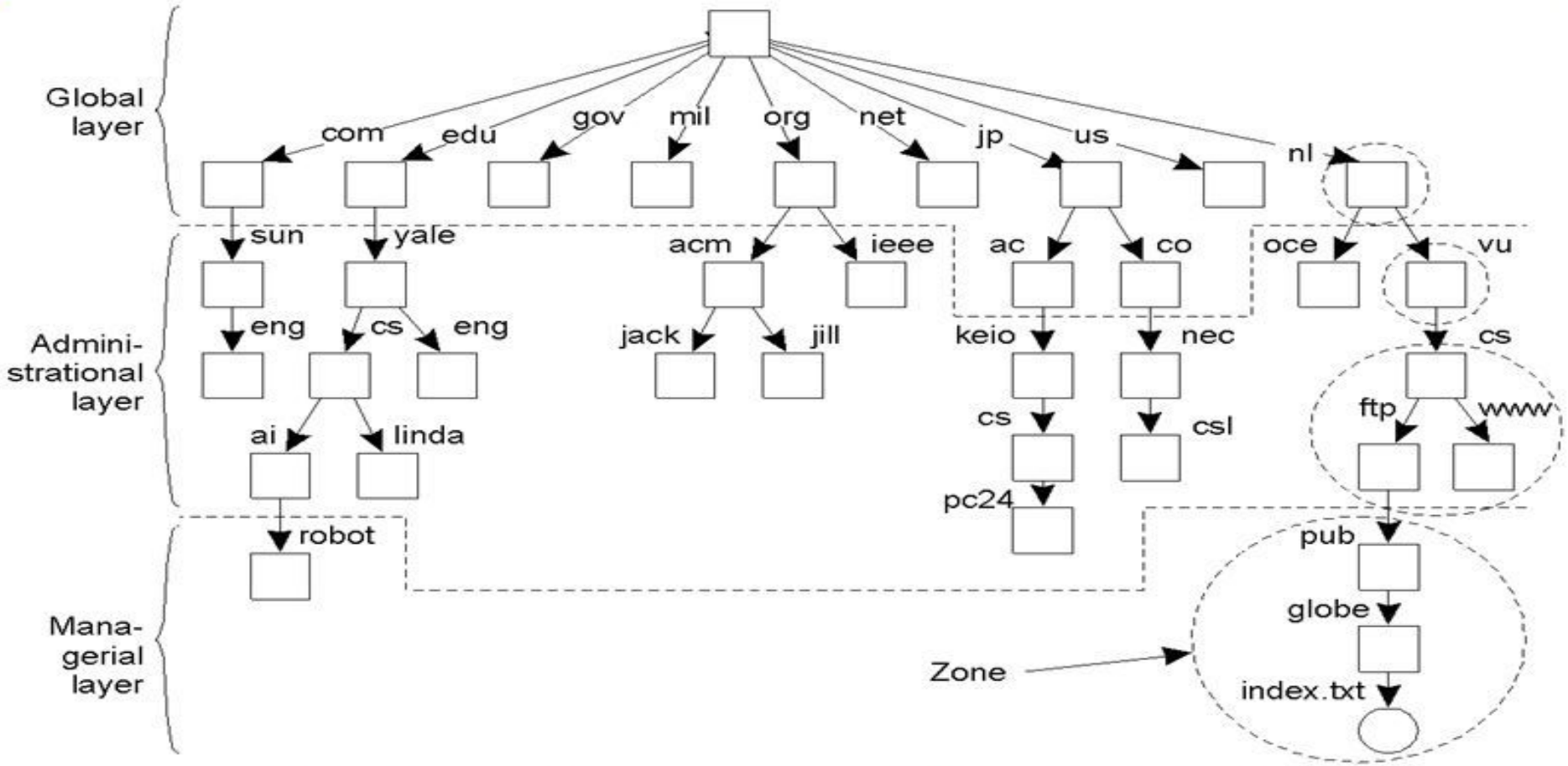
MOUNT POINT

- A mount point is **a directory in a file system where additional information is logically connected from a storage location outside the operating system's root drive and partition**.

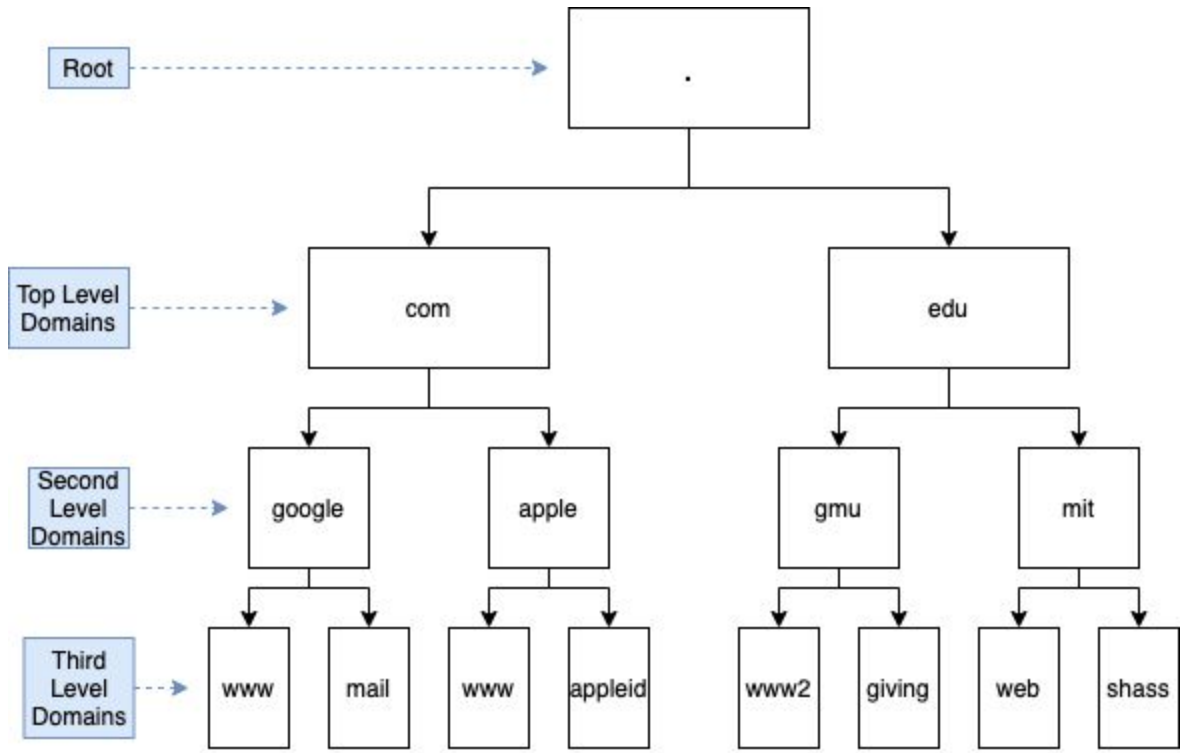
Name Space Distribution

- Name spaces for a large-scale, possibly worldwide distributed system, are usually organized hierarchically. The name space is partitioned into three logical layers:
  - *The name space is partitioned into three logical layers:*
    - ***The global layer*** *is formed by highest-level. This layer is often characterized by its stability; the directory tables in this layer are rarely changed (19)*
    - ***The administrational layer*** *is formed by directory nodes that together are managed within a single organization. A characteristic feature of the directory nodes in the administrational layer is that they represent groups of entities that belong to the same organization or administrational unit.*
    - ***The managerial layer*** *consists of nodes that may typically change regularly. The nodes in this layer are maintained not only by system administrators, but also by individual end users of a distributed system.*

Name Space Distribution (1)

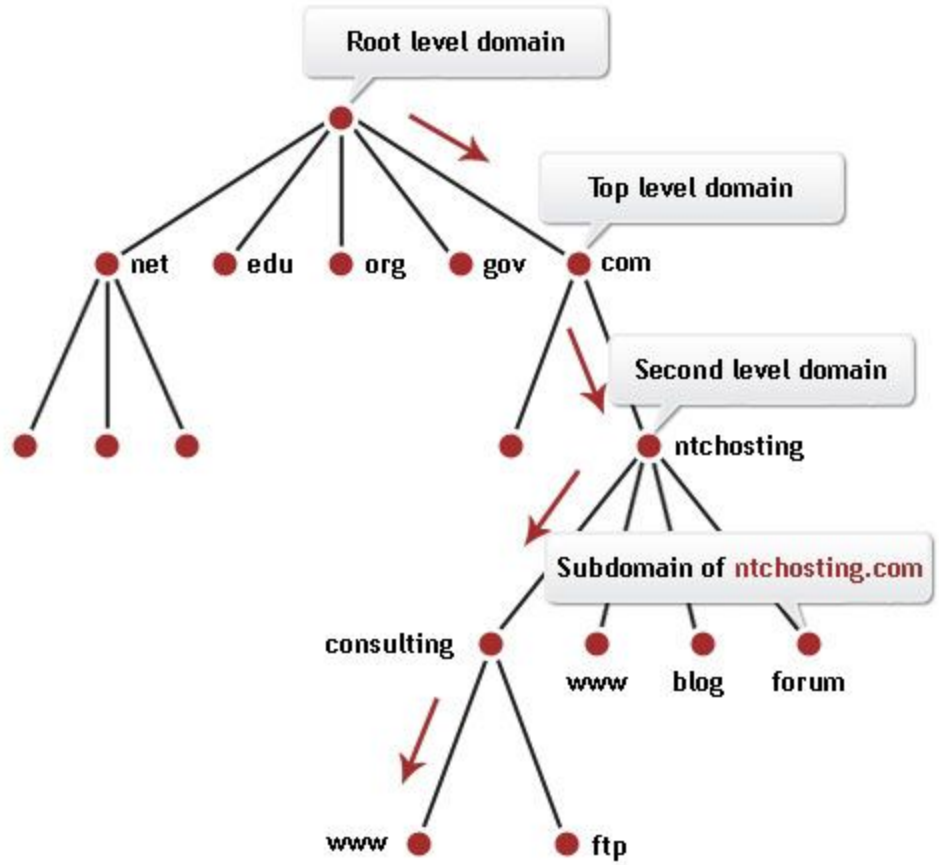


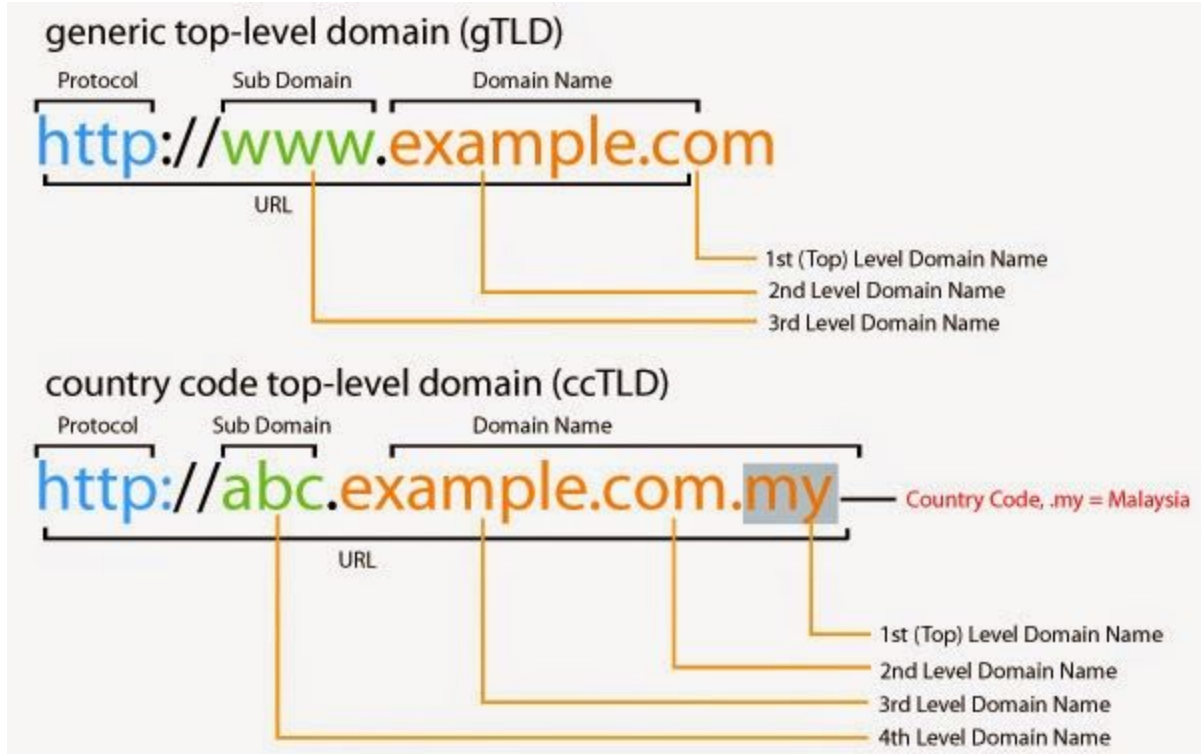
- An example partitioning of the DNS name space, including Internet-accessible files, into three layers.



Graphic created by Blake Khan (blakekhan.com)

gTLDs	Meaning	Intended for
.com	Commercial	Commonly used for Commercial websites.
.org	Organization	Typically used for non-profit organizations/Charitable Institutions.
.net	Network	Originally designed for Network providers, but currently in wide use.
.info	Information	For websites providing Informational Content.
.biz	Business	For websites that are business related.


Root level domain
Top level domain
net
edu
org
gov
com
Second level domain
ntchosting
Subdomain of ntchosting.com
consulting
www
blog
forum
www
ftp


generic top-level domain (gTLD)
Protocol
Sub Domain
Domain Name
http://www.example.com
URL
1st (Top) Level Domain Name
2nd Level Domain Name
3rd Level Domain Name
country code top-level domain (ccTLD)
Protocol
Sub Domain
Domain Name
http://abc.example.com.my
Country Code, .my = Malaysia
URL
1st (Top) Level Domain Name
2nd Level Domain Name
3rd Level Domain Name
4th Level Domain Name