Application Layer

The **application layer** manages communication between programs or applications running on users' computers and the network itself.

This layer provides high-level services to applications, allowing them to communicate, exchange data and interact with other applications across different networks.

At this layer, data units are called **messages**.

Some common protocols and services associated with the application layer include:

- **HTTP (Hypertext Transfer Protocol)** for the World Wide Web.
- **SMTP (Simple Mail Transfer Protocol)** for sending emails.
- **FTP (File Transfer Protocol)** for transferring files.

Domain Names

A very important application-layer service is the **Domain Name System (DNS)**, which associates domain names with hosts connected to a network.

A **domain name** is a unique, human-readable name used to identify and access online resources such as websites, email servers, online services and other resources.

Domain names are read from right to left and consist of strings separated by dots.

They are mainly used to simplify access to resources on the Internet. Instead of having to remember a sequence of numbers that identifies a server, users can enter a more meaningful domain name into a browser or application.

The DNS therefore translates a domain name into a numerical address, allowing a network device to locate the corresponding host.

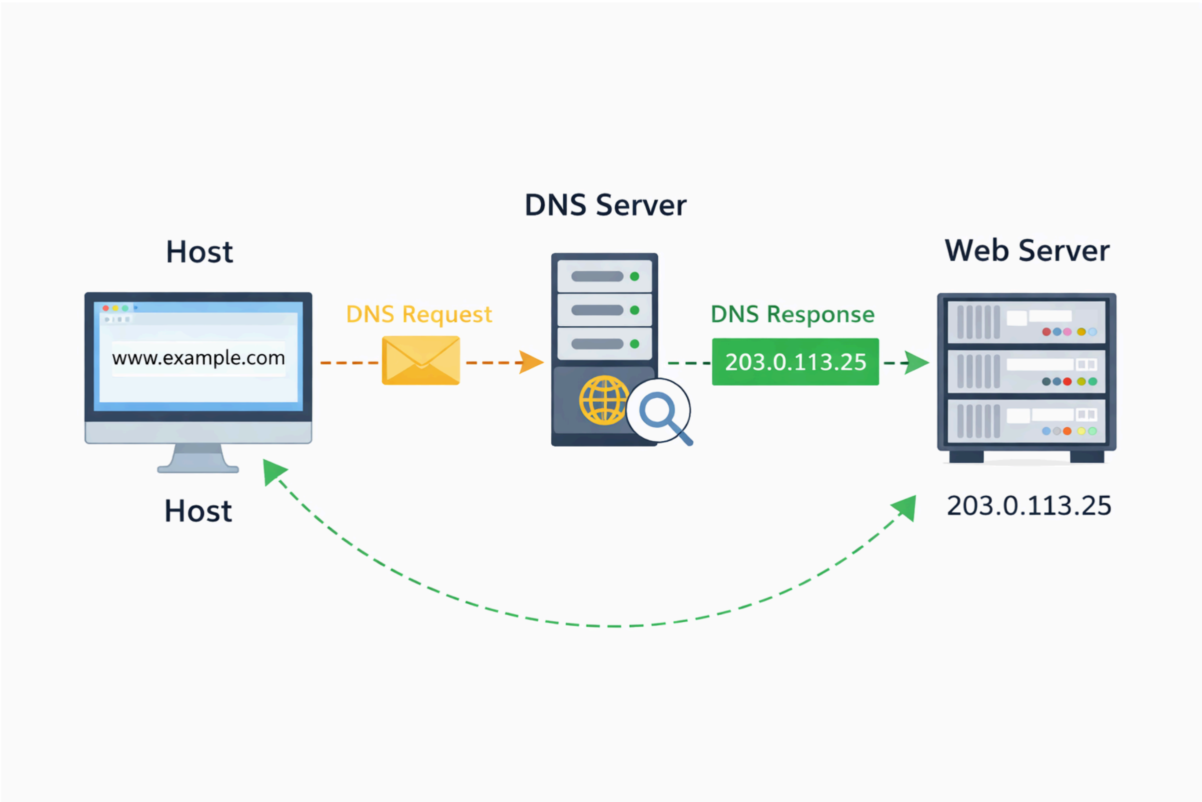
The DNS service is implemented through a globally distributed database made up of DNS servers.

The DNS is similar to a mobile phone's contact list. Once Mario Rossi's telephone number has been saved, there is no need to enter the number manually each time: the user can simply search for the name Mario Rossi in the contact list.

In the same way, the DNS converts a domain name, such as `example.it`, into a numerical address, such as `190.12.13.54`. This number corresponds to the actual address of a specific host on the network. In later chapters, we will see that this address is called an **IP address**.

When a user enters an address into a browser, the DNS server searches its records for the corresponding numerical address.

If the address is not found, the DNS server forwards the request to another server that is more likely to have the required information. This process continues until an answer is found or it is confirmed that the name does not exist.



Example

Suppose that a user wants to access the web address `www.example.com`.

The user's device sends a DNS request to obtain the numerical address associated with that domain name.

- **DNS request:** The user enters `www.example.com`, starting the name-resolution process.
- **Sending the request:** The device sends the DNS request to a local DNS server or to the DNS server configured by the ISP.
- **Receiving DNS server:** The DNS server receives the request and searches its database for the address associated with `www.example.com`.
- **DNS server response:** The DNS server replies with the address corresponding to the requested domain name.
- **Receiving the response:** The device receives the numerical address returned by the DNS query.
- **Connecting to the web server:** Now that the device has the numerical address, it can use it to establish a connection with the web server associated with `www.example.com`.

A **nameserver** is an authoritative DNS server. This means that it directly holds the official information for one or more domains—or, more precisely, for one or more DNS zones.

Domain Name Hierarchy

A **Fully Qualified Domain Name (FQDN)** is a domain name specified in full, including all the labels in the DNS hierarchy, with no parts omitted.

An FQDN is therefore an absolute name that identifies a resource precisely within the DNS hierarchy, all the way to the root.

A domain name is hierarchically divided into three main levels. Reading from right to left, each level is separated by a dot:

- Top-level domain
- Second-level domain
- Subdomain

Note: A final dot may sometimes appear on the right-hand side of a domain name. This represents the DNS root and makes the name complete and absolute. Without the final dot, the name may be interpreted as relative in certain contexts.

Top-Level Domain

The most significant part of a domain name is the first part encountered when reading from right to left.

This is called the **Top-Level Domain**, or **TLD**, for example `.org` or `.it`.

TLDs are coordinated and assigned within the system overseen by **ICANN**, the Internet Corporation for Assigned Names and Numbers, which is responsible for coordinating unique identification systems on the Internet.

There are two main types of TLD:

- Generic TLDs
- Country-code TLDs

Generic TLDs
Generic TLDs are generally associated with broad categories rather than specific geographical locations.

Examples include:

- `.com` **— commercial:** Originally intended for commercial organisations, it is now widely used without specific restrictions.
- `.org` **— organisation:** Originally intended for non-profit organisations, it is now used by a wide variety of entities.
- `.net` **— network:** Originally intended for network infrastructure organisations, it is now used by many different types of entities.

Country-Code TLDs
Country-code TLDs are associated with particular countries or geographical areas.

Each country has its own TLD representing the geographical identity of the domain.

Examples include:

- `.us` — United States
- `.uk` — United Kingdom
- `.fr` — France
- `.de` — Germany
- `.jp` — Japan

Second-Level Domain

A Second-Level Domain, or **SLD**, is the part of a domain name located immediately to the left of the Top-Level Domain.

It is often simply called the **domain name**.

The domain name is the name chosen for a website, service or online resource, following the structure:

SLD.TLD

For example, consider the domain `example.com`:

- `example` is the Second-Level Domain.
- `com` is the Top-Level Domain.

The Second-Level Domain is the part that makes a domain name distinct from others.

It is the customisable and identifying part of the web address.

It may contain words or expressions representing a brand, business, website content or anything else the organisation or owner wishes to communicate.

Together, the Second-Level Domain and Top-Level Domain form the **root domain**, meaning the complete domain name without any subdomains.

Subdomain

A **subdomain** is the part of a domain name located to the left of the Second-Level Domain.

Subdomains are mainly used to provide additional context, structure and organisation within the domain-name architecture.

A subdomain may consist of a third-level domain or of several additional levels, such as fourth-level or fifth-level domains.

Example
Consider the domain:

`materials.students.university.it`

- `it` is the Top-Level Domain.
- `university` is the Second-Level Domain.
- `students` is a subdomain.
- In `materials.students.university.it`, both `materials` and `students` are additional levels placed to the left of the main domain `university.it`. In this case, the domain contains multiple levels of subdomains.

In many cases, the leftmost part of a domain name identifies a specific host or service, such as `www`, `mail`, `files` or `db`.

Examples include:

- **Web server:** `www` is often used to identify a web server hosting a website.
- **Email server:** `mail` may be used to identify an email server.
- **File server:** `files` may be used to identify a file or storage server.

- **Application server:** `app` may be used to identify a server hosting a particular application.
- **Database server:** `db` may be used to identify a database server.
- **Game server:** `game` may be used to identify an online game server.

Example
The domain `email.example.com` indicates that the host being accessed provides an email-related service.

The domain `www.example.com` indicates that the host being accessed provides a website or web page.

Registrar

A **domain registrar** is an organisation or company authorised to facilitate the registration of Internet domain names.

It acts as an intermediary between end users and domain-name registries.

The registrar provides domain registration, management and renewal services, allowing users to establish an online identity through domain names.

To register a Second-Level Domain, users must request the service from a registrar.

Important aspects of domain registrars include:

- **Authorisation:** Domain registrars must be accredited by the organisations responsible for Top-Level Domains or by national registries. This authorisation ensures that the registrar follows specific standards and regulations.
- **Choosing domains:** Registrars allow users to search for and select available domain names across different Second-Level Domains and Top-Level Domains.
- **Registration:** Registrars allow users to register domain names by collecting the necessary information and managing the registration process through the relevant TLD registry.

It is important to clarify that domain names are not permanently sold. They are granted for temporary use.

When a user “purchases” a domain, they are effectively entering into a rental agreement: they obtain the exclusive right to use that name for a limited period, usually one or more years.

When this period expires, the domain must be renewed. Otherwise, the right to use it ends and the name may become available to other users.

A registrar therefore does not sell permanent ownership. It manages a temporary right of use, while the domain remains part of the wider global DNS governance system.

Registry

Having explained the role of the registrar, we can introduce the other fundamental participant in the domain-name system: the **registry**.

A registry is the organisation that officially manages a Top-Level Domain, such as `.it`, `.com`, `.org`, `.net` or `.edu`.

While the registrar is the intermediary that communicates with the end user, the registry is the technical and administrative authority responsible for maintaining and governing the domain extension.

In simple terms:

- The **registrar** provides services to users.
- The **registry** manages the domain's infrastructure and rules.

A registry has three main responsibilities.

Maintaining the Domain Database

The first responsibility is maintaining the database of all domain names registered under a particular TLD.

This database contains information such as:

- Which domains are registered.
- Their status, such as active, expired or locked.
- The nameservers associated with each domain.

Managing TLD Nameservers

The second responsibility is managing the nameservers of the TLD.

When the DNS needs to resolve a name such as `www.example.it`, the resolution process is eventually directed to the `.it` TLD servers.

These servers are managed by the registry and respond by identifying the authoritative nameservers for the requested domain.

Defining and Applying Domain Rules

The third responsibility is defining and enforcing the rules of the domain.

Each TLD has its own policies governing:

- Who may register it.
- How long it may be registered for.
- What restrictions apply.
- Which renewal, transfer and revocation procedures are available.

For example:

- Some TLDs have specific eligibility requirements, such as residence in a particular country or belonging to a certain type of organisation.
- Other TLDs are open to anyone.
- Expiration and recovery periods may vary.

The registry does not usually sell domain names directly to end users.

Instead, it operates through a network of accredited registrars, which are authorised to interact with its database using standardised protocols.

When a user registers, renews or transfers a domain:

- The user communicates with the registrar.
- The registrar sends the operation to the registry.
- The registry updates the official TLD database.

Within this model, the registry ensures consistency, uniqueness and global reliability, while the registrar provides the commercial and operational interface used by the customer.