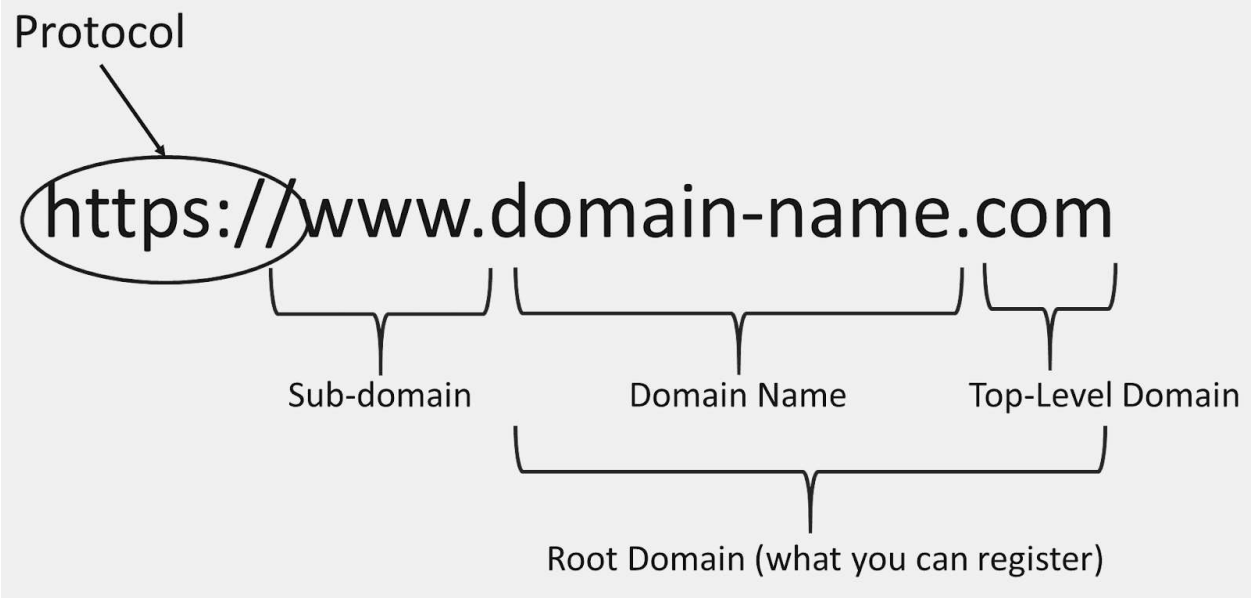
Domain Names and DNS (Domain Name System)

1. Introduction to Domain Names

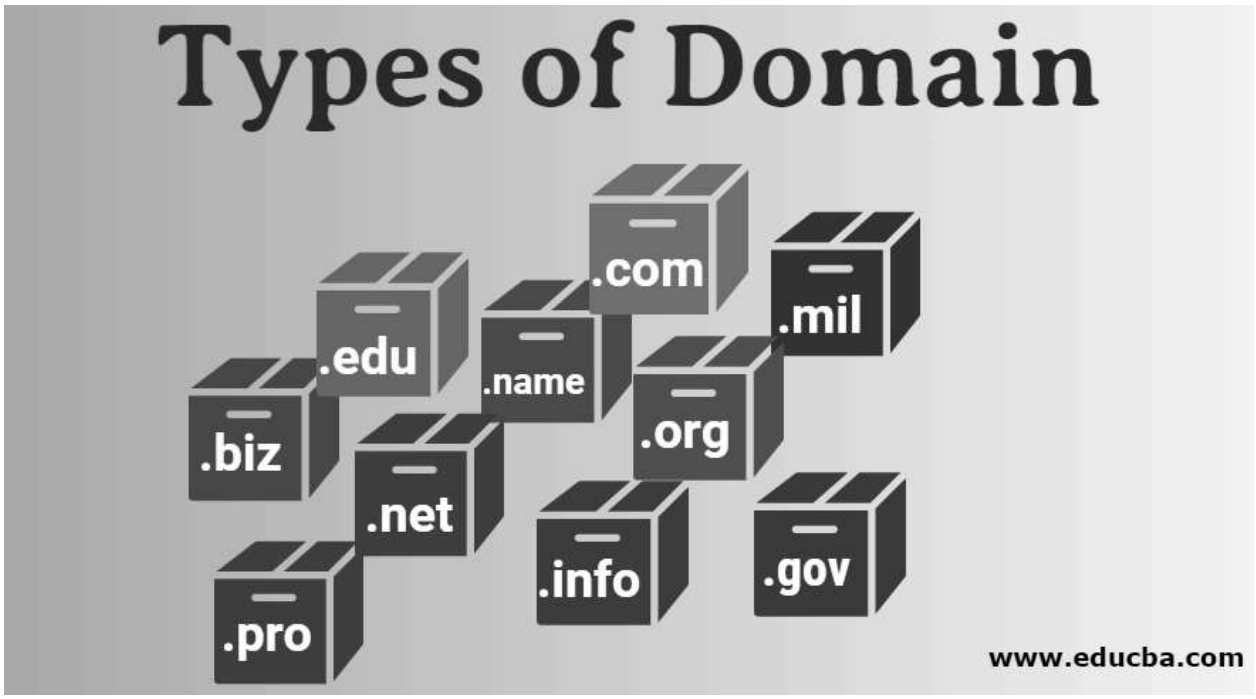
- **Definition**:
  - A domain name is a human-readable address used to access websites or other internet resources.
  - Example: `www.google.com` instead of `142.250.64.78`.
- **Purpose**:
  - Simplifies navigation on the internet.
  - Provides a memorable and user-friendly alternative to IP addresses.

2. Structure of a Domain Name



- Domain names are hierarchical and consist of several components separated by dots:
  - **Top-Level Domain (TLD)**:
    - The highest part of a domain.
    - Examples: `.com`, `.org`, `.edu`, `.gov`.
  - **Second-Level Domain (SLD)**:
    - The main part of the domain name, chosen by the owner.
    - Example: `google` in `google.com`.
  - **Subdomain**:
    - An optional prefix to the SLD.
    - Example: `mail.google.com` (where `mail` is the subdomain).

3. Types of Domains

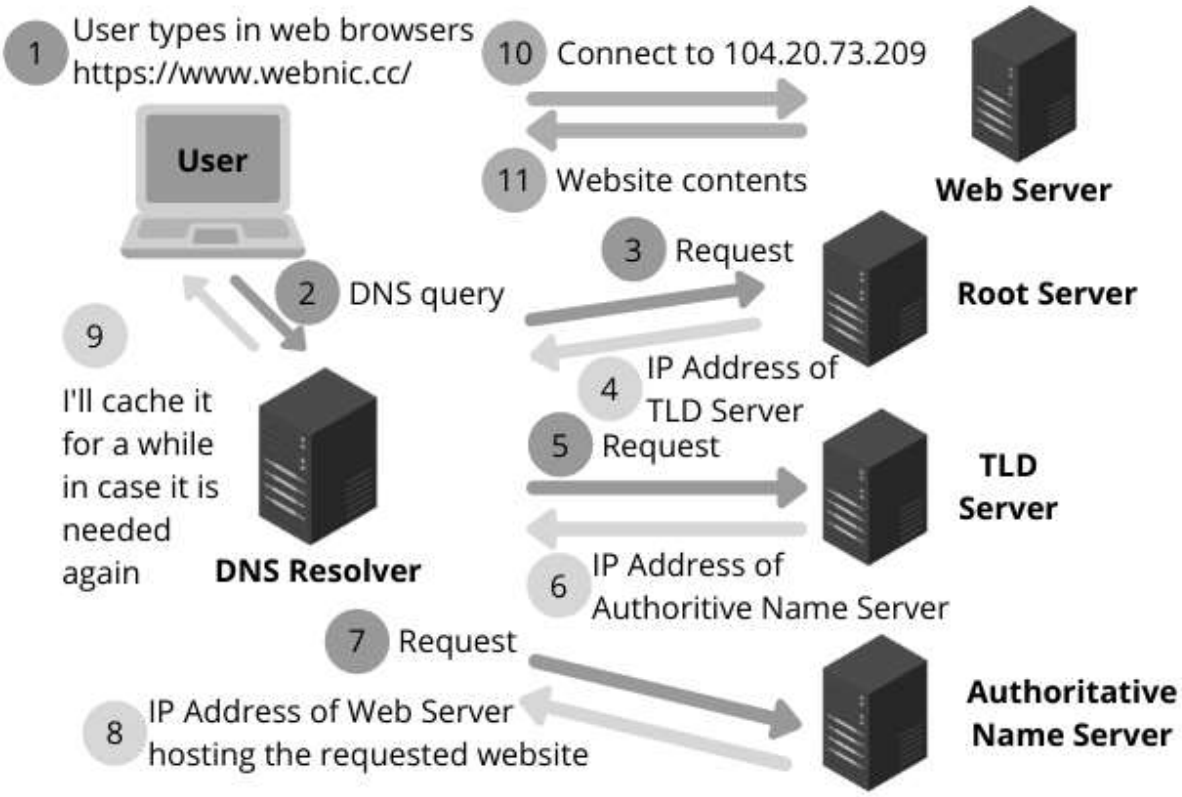


- **Generic Top-Level Domains (gTLDs)**: Common and open to everyone (e.g., `.com`, `.org`, `.net`).
- **Country Code Top-Level Domains (ccTLDs)**: Specific to a country or region (e.g., `.us`, `.in`, `.uk`).
- **Sponsored Top-Level Domains (sTLDs)**: Restricted to specific organizations or industries (e.g., `.edu`, `.gov`).
- **Internationalized Domain Names (IDNs)**: Support non-English scripts like Arabic, Chinese, or Hindi.

4. What is DNS (Domain Name System)?

- **Definition**:
  - DNS is a hierarchical system that translates human-readable domain names into machine-readable IP addresses.
  - Example: `www.google.com` → `142.250.64.78`.
- **Purpose**:
  - Acts as the internet's "phonebook."
  - Enables seamless navigation without needing to remember numerical IP addresses.

5. How DNS Works



1. **User Request**: A user enters a domain name (e.g., `www.google.com`) into their browser.
2. **DNS Query**: The browser sends a request to a DNS resolver to find the IP address associated with the domain.
3. **Recursive Resolver**:
    - The resolver queries DNS servers in a hierarchical manner:
        1. **Root DNS Server**: Directs the resolver to the appropriate TLD server (e.g., `.com`).
        2. **TLD DNS Server**: Points to the authoritative server for the domain (e.g., `google.com`).
        3. **Authoritative DNS Server**: Provides the IP address for the domain.
4. **Response**: The resolver sends the IP address back to the browser.
5. **Connection**: The browser uses the IP address to connect to the web server and retrieve the website.

6. Components of DNS

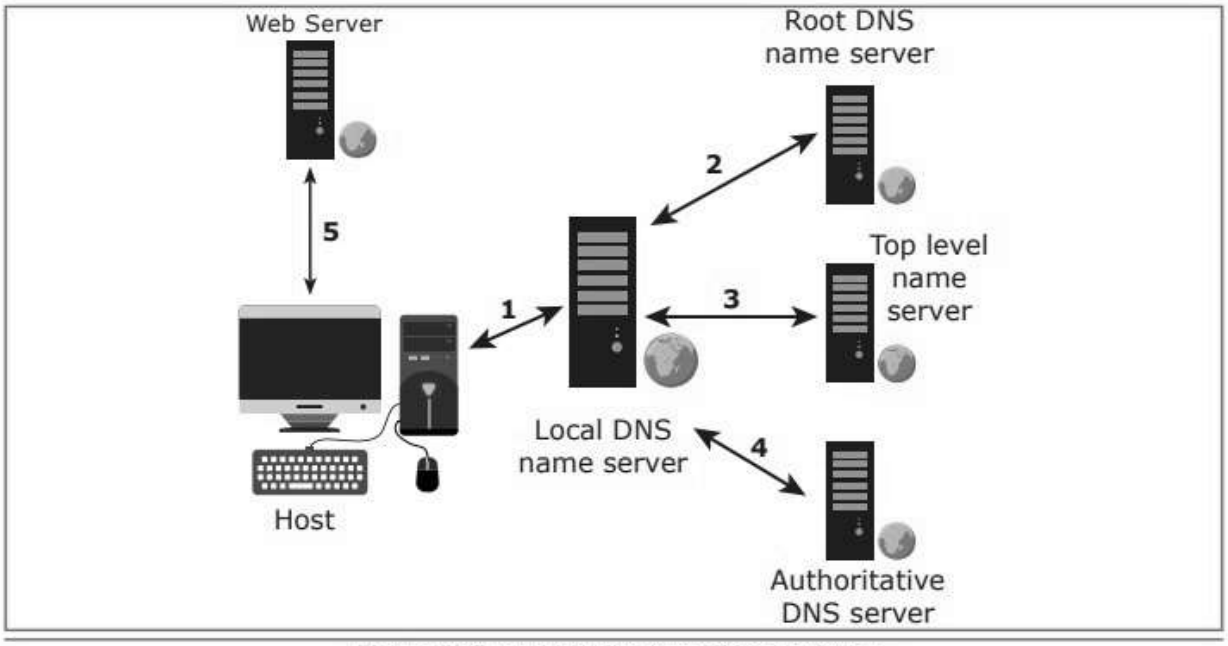
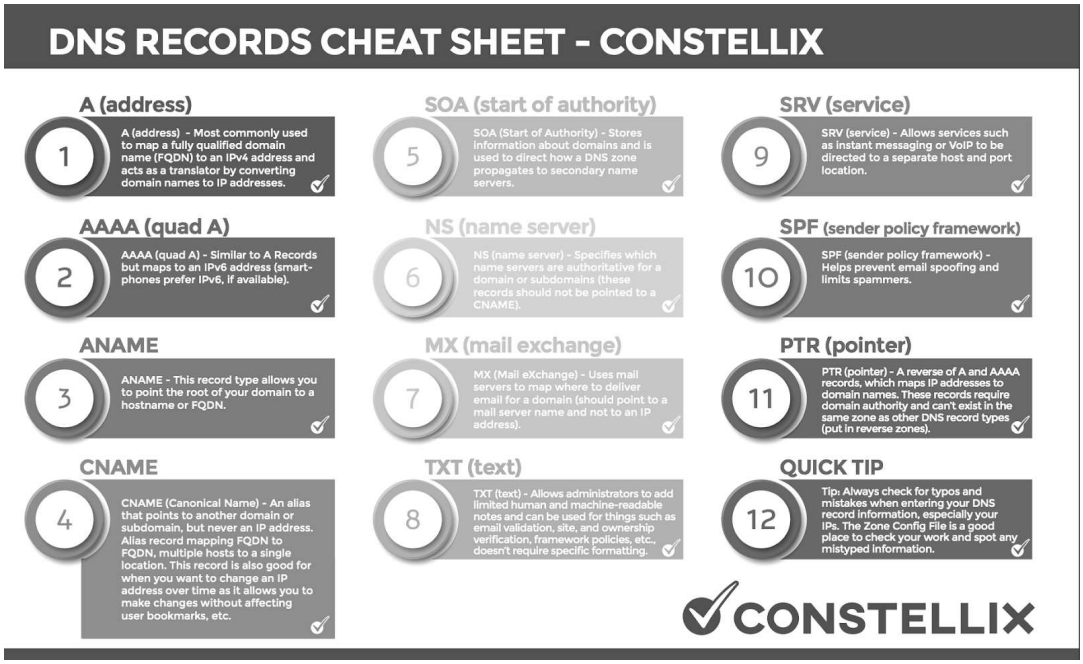


Figure 12.8 working structure of Name server

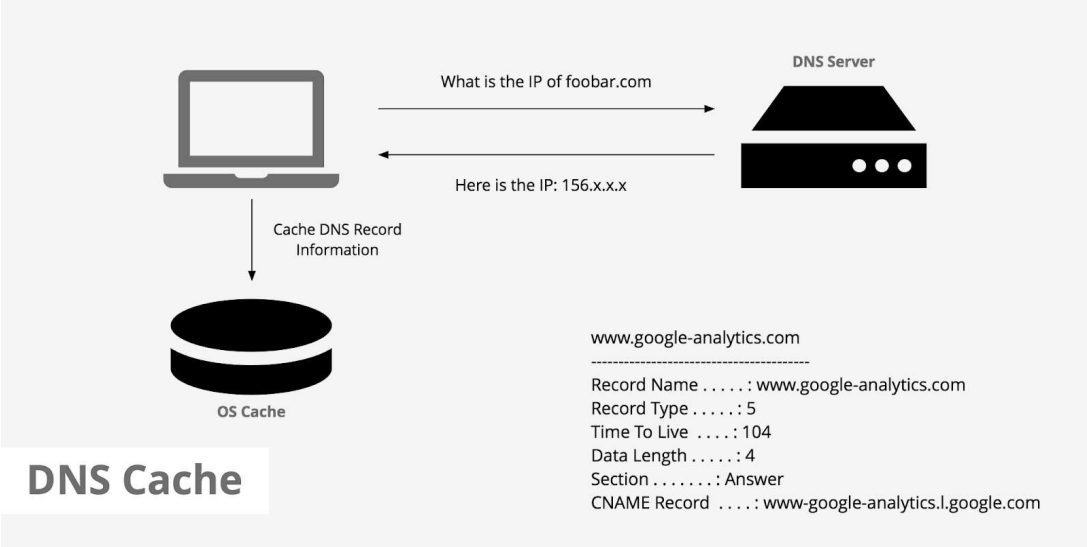
- **DNS Resolver**: The first point of contact for a DNS query, often managed by the user's ISP.
- **Root Server**: Directs queries to the appropriate TLD server; there are 13 root server clusters globally.
- **TLD Server**: Manages domains within a specific TLD (e.g., `.com`, `.org`).
- **Authoritative DNS Server**: Holds the actual mapping of domain names to IP addresses.

7. Types of DNS Records



- **A Record (Address Record)**: Maps a domain to an IPv4 address.
- **AAAA Record**: Maps a domain to an IPv6 address.
- **CNAME Record (Canonical Name)**: Points a domain to another domain (used for subdomains).
- **MX Record (Mail Exchange)**: Directs emails to mail servers.
- **TXT Record**: Stores text information for verification purposes (e.g., SPF, DKIM for email security).

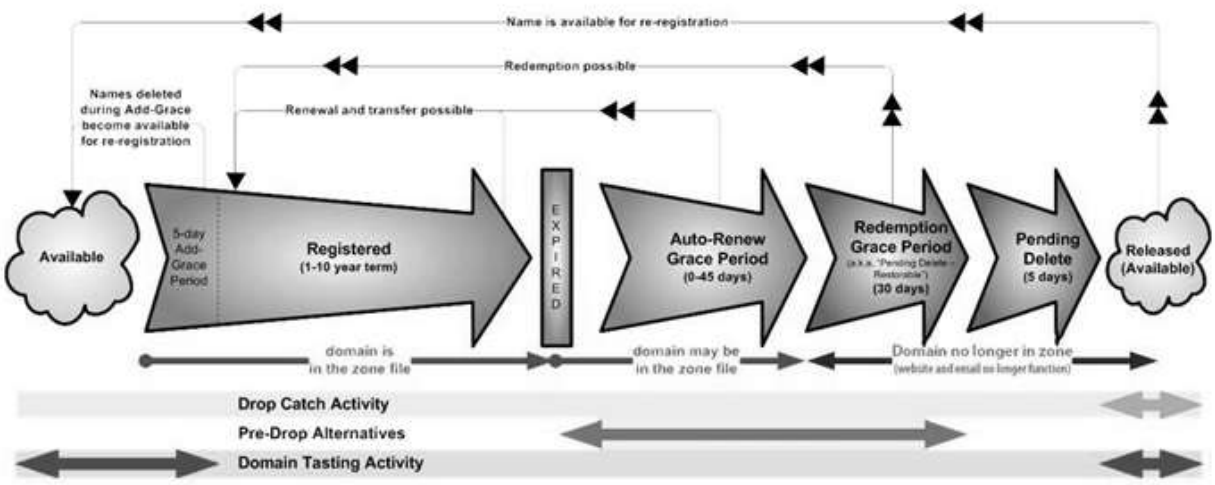
8. DNS Cache



- **Definition**: Temporary storage of DNS query results on devices or servers to speed up subsequent requests.
- **Benefits**:
    - Reduces query time.
    - Decreases load on DNS servers.
- **Potential Issues**: Cached data can become outdated, leading to resolution errors.

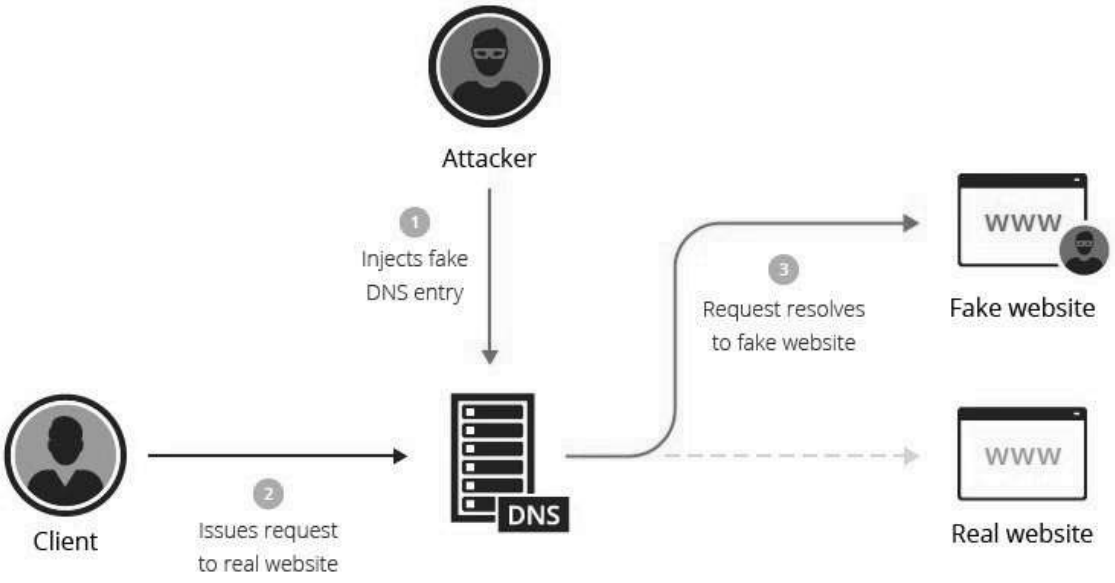
9. Domain Registration

- **Process**:
    - Choose a domain name.
    - Check its availability through a registrar (e.g., GoDaddy, Namecheap).
    - Register the domain and configure DNS settings.
- **Domain Lifecycle**: **Available** → **Registered** → **Active** → **Expired** → **Grace Period** → **Deleted**.

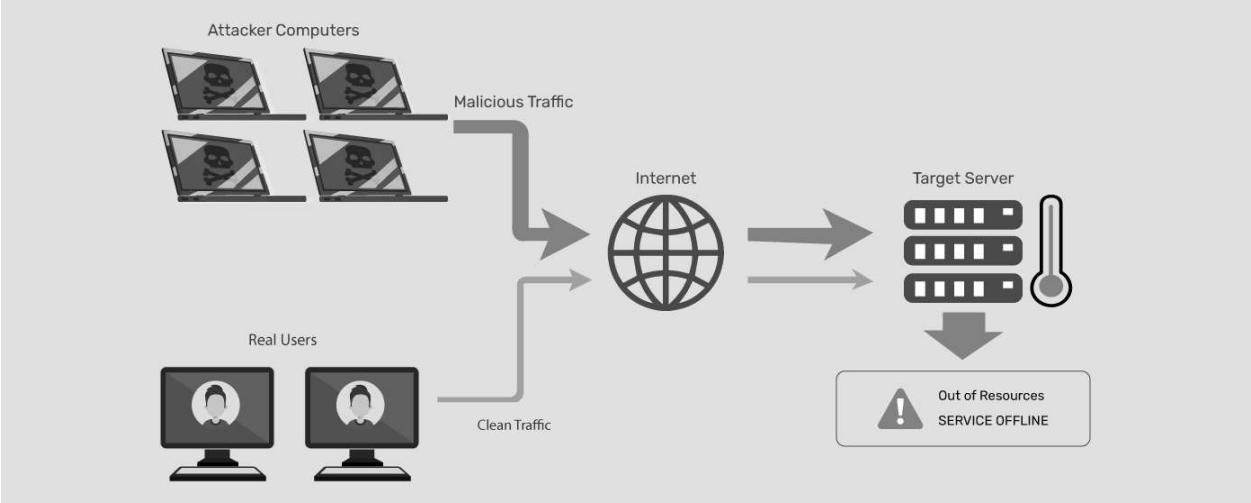


10. Security in DNS

- **DNS Vulnerabilities**:

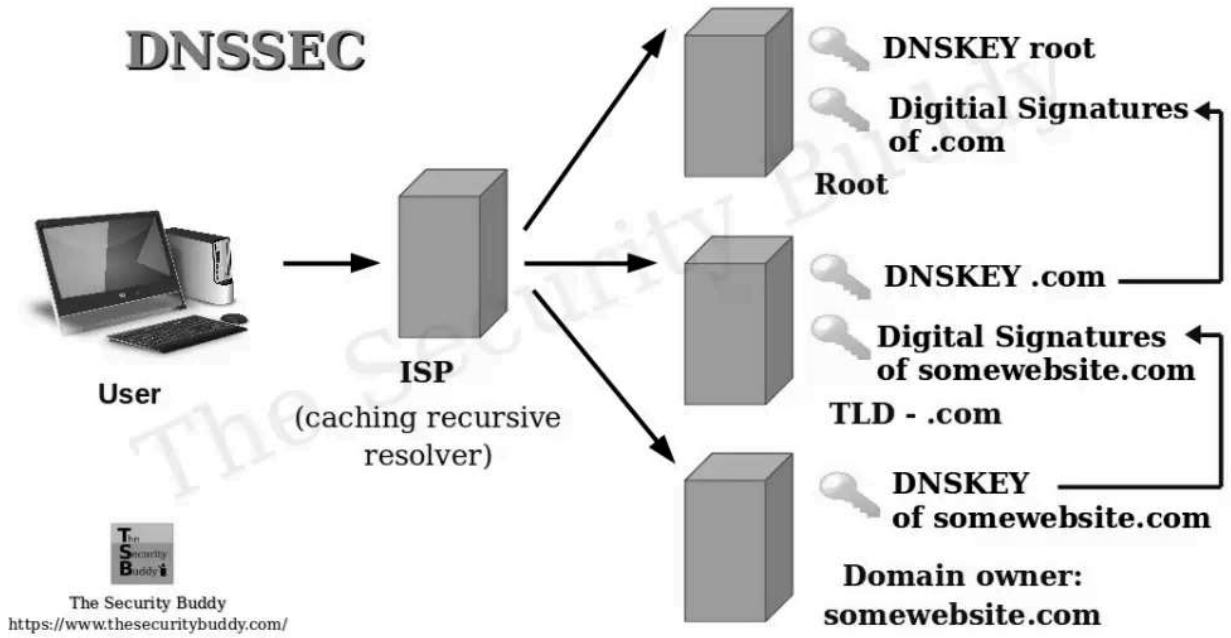


  - **DNS Spoofing/Cache Poisoning**: Attackers provide false DNS responses to redirect users to malicious sites.

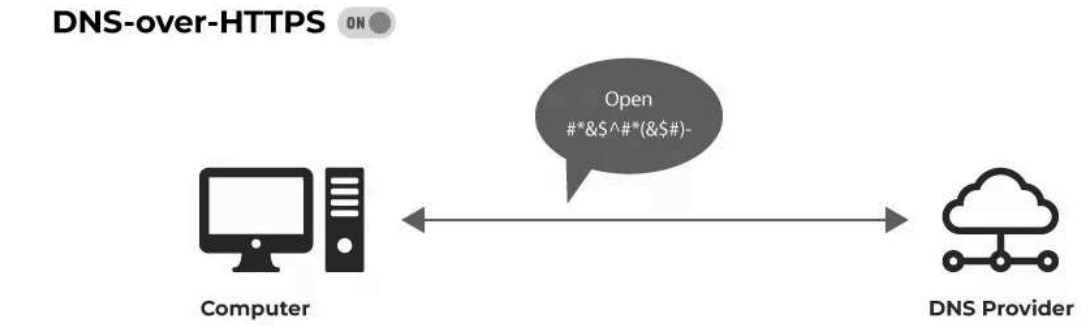


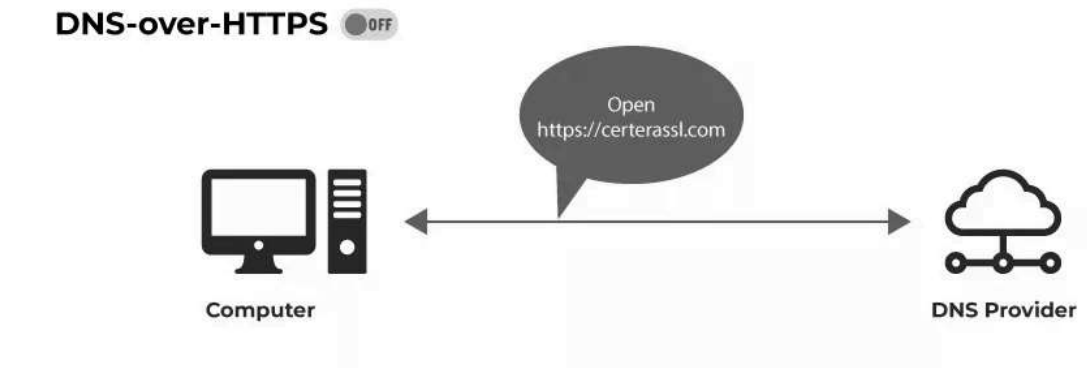
- **DDoS Attacks**: Overwhelming DNS servers with traffic to disrupt service.

- **Mitigation Techniques**:



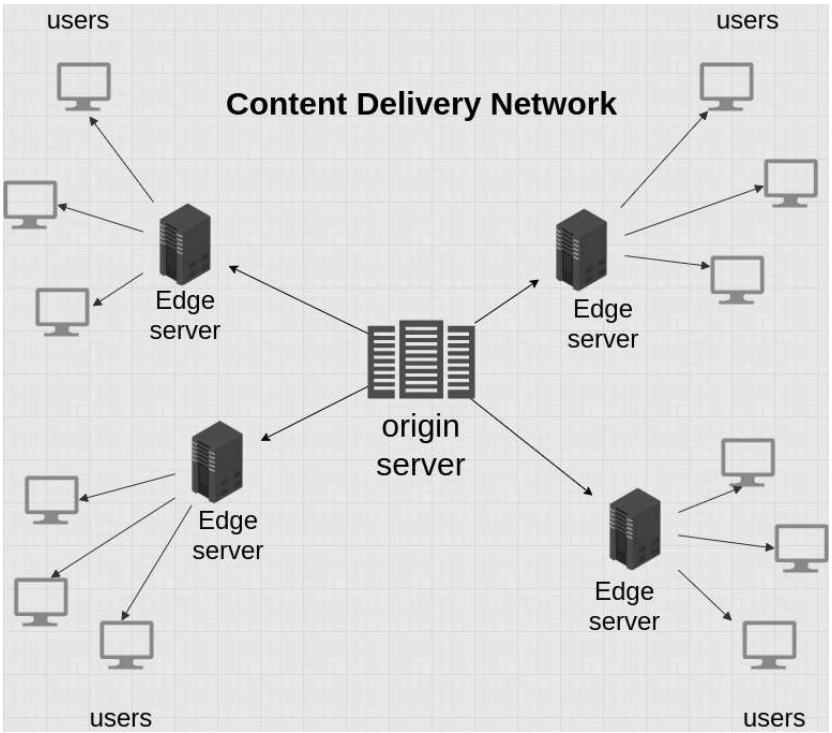
- **DNSSEC (Domain Name System Security Extensions)**: Adds authentication to DNS responses using digital signatures.



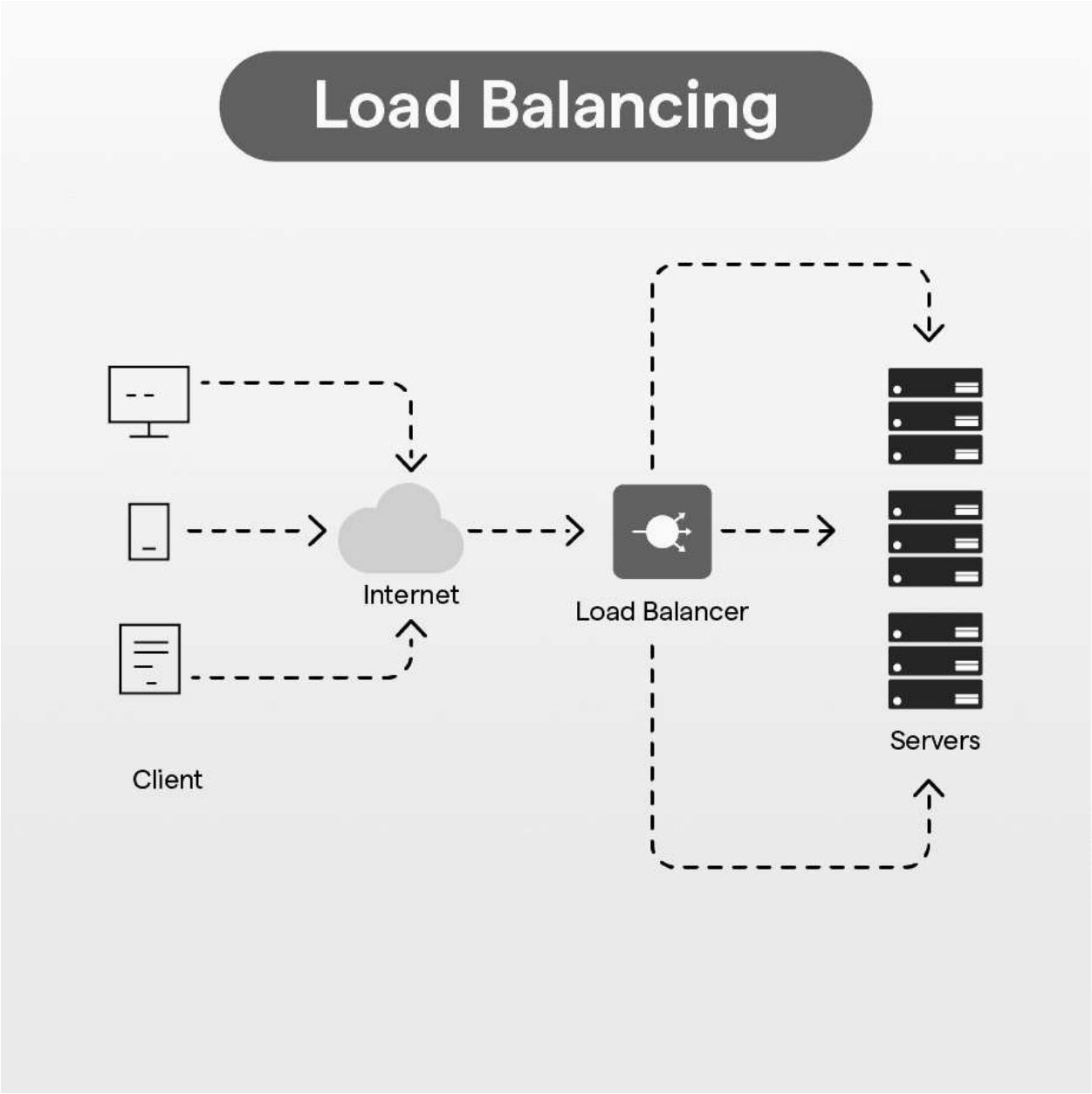


- **HTTPS DNS (DoH)**: Encrypts DNS queries for privacy.

11. DNS in Modern Applications



- **Content Delivery Networks (CDNs)**: Use DNS to route users to the nearest server for faster content delivery.



- **Load Balancing**: DNS helps distribute traffic among multiple servers to ensure reliability.
- **Cloud Computing**: Domain names map to virtual resources in cloud environments.
