# Domain Name System (DNS): Architecture and Record Types

Muskula Rahul

## 1 Domain Name System

The Domain Name System (DNS) serves as a foundational component of the internet, translating human-readable domain names into IP addresses that computers use to communicate. This article explores DNS architecture and the various DNS record types, using the visual representations provided in the diagrams to outline how DNS functions and the roles of specific DNS records.

**Tip**

Think of DNS as the internet's phonebook. It translates human-readable names (like `https://dns.google.com/`) into computer-friendly IP addresses (like 8.8.8.8). DNS records are the building blocks that enable this translation and provide different functionalities.

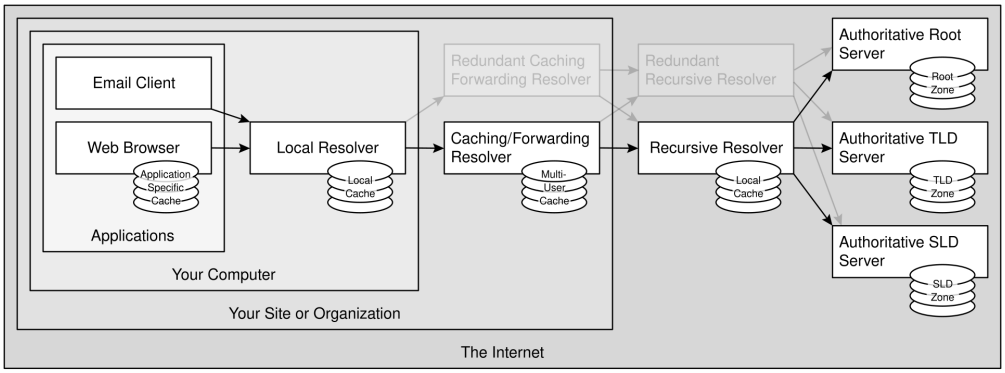


Figure 1: DNS Architecture. Picture By Aaron Filbert - Own work, CC BY-SA 4.0

### DNS Architecture Overview

The above architecture diagram illustrates the flow of DNS resolution, highlighting key components involved in resolving a domain name to an IP address. Here's a breakdown of each component:

#### 1. Applications on the User's Computer

- **Email Client / Web Browser**: These applications initiate DNS queries when users try to access web pages or send emails. Each application often has its own **application-specific cache**, which stores DNS responses temporarily. For instance, a web browser might store DNS records for recently visited websites in its cache for a short period (usually seconds to a few minutes) to avoid repeated DNS queries.

**2. Local Resolver**

- **Local Resolver**: The local resolver is the first DNS resolver encountered when an application initiates a DNS query. It is often part of the operating system's network stack (e.g., `nsswitch.conf` in Unix-like systems or DNS Client in Windows).
- **Local Cache**: This local resolver stores frequently accessed DNS responses in a local cache, which helps reduce DNS lookup time. Cached entries here obey a **Time-To-Live (TTL)** value that controls how long the entry can be kept before it must be refreshed.

**3. Caching/Forwarding Resolver**

- **Caching/Forwarding Resolver**: This resolver is typically provided by the user's organization or ISP. Its primary job is to check if the domain exists in its **multi-user cache** before forwarding queries to an external resolver. This cache stores records accessed by multiple users across the network, allowing faster resolution for commonly accessed domains.
- **Forwarding**: When the domain is not in the cache, the caching resolver forwards the query to an external DNS server (e.g., a public DNS server like Google DNS or Cloudflare DNS).

**4. Recursive Resolver**

- **Recursive Resolver**: The recursive resolver is where the DNS query is broken down into multiple steps and systematically resolved.
- If a domain is not found in any of the prior caches, the recursive resolver performs an iterative lookup, starting with the **Root Server** and moving down the DNS hierarchy until it finds the authoritative answer for the domain. Recursive resolvers are designed to handle large numbers of requests efficiently and can cache responses to improve performance for future queries.

**5. Authoritative Servers**

- **Root Server**: The root server is the highest point in the DNS hierarchy. It directs queries for specific top-level domains (TLDs) to the appropriate TLD servers. Root servers don't store information about individual domains; instead, they only store references to TLD servers, making them essential for initial query redirection.
- **TLD Server**: Each TLD server is responsible for a specific top-level domain (like `.com`, `.org`, `.net`). These servers hold records that direct the query to the appropriate **Second-Level Domain (SLD)** servers.
- **SLD Server**: The SLD server, managed by the domain owner or registrar, holds authoritative DNS records for individual domain names. It provides the final answer in the form of an IP address or other relevant DNS information.

This hierarchical system is essential to the scalability and robustness of DNS, as it allows each layer to manage only a specific part of the domain namespace.

# DNS Record

A `DNS record` is a piece of information that maps a domain name to associated data, such as an IP address, email server, or other resources. For example, A records map a domain name to an IPv4 address, MX records specify the mail server for a domain, and CNAME records allow one domain to be an alias of another. The different types of DNS records have distinct structures and are used to configure the domain and its associated services based on the specific needs of a website or application.

## DNS Record Types

The following diagram categorizes DNS record types. Here's an in-depth look at each category, including use cases, technical explanations, and lesser-known records.

### 1. DNS Meta Records

- **NS (Name Server)**: Specifies the DNS servers that are authoritative for a particular domain. Each domain must have at least one NS record, often listing multiple for redundancy.
- **SOA (Start of Authority)**: Contains crucial metadata about the zone, including:
  - **MNAME**: Primary master name server.
  - **RNAME**: Email of the administrator responsible for the zone.
  - **Serial Number**: Incremented with every update, signaling secondary DNS servers to refresh.
  - **Refresh, Retry, Expire, TTL Values**: Define caching behavior and retry intervals.
- **CNAME (Canonical Name)**: Creates an alias for a domain. CNAMEs are useful for mapping subdomains to other domains or servers (e.g., pointing `www.example.com` to `example.com`).
- **PTR (Pointer)**: Used for reverse DNS lookups, where an IP address is mapped back to a domain name. PTR records are stored in a specially formatted reverse DNS zone.
- **DNAME (Delegation Name)**: Delegates an entire domain subtree to another domain. Unlike CNAME, which only applies to a single hostname, DNAME applies to all names below it.
- **OPT (Option)**: Part of the EDNS (Extension Mechanisms for DNS) specification. It allows DNS packets to exceed 512 bytes and supports new options like DNSSEC.

### 2. IP Address Records

- **A Record**: Maps a domain name to an IPv4 address. It's the most common DNS record and the basis for connecting domain names to servers.
- **AAAA Record**: Similar to an A record but maps a domain to an IPv6 address.
- **APL (Address Prefix List)**: Stores lists of IP prefixes, often used in policy-based systems. APL records can contain both IPv4 and IPv6 prefixes.

### 3. Informational Records

- **TXT (Text)**: Allows administrators to store arbitrary text in DNS. Common uses include storing SPF (Sender Policy Framework) records, DKIM (DomainKeys Identified Mail) keys, and other verification data.
- **HINFO (Host Information)**: Contains details about the hardware and operating system of a host, although it is rarely used due to security and privacy concerns.

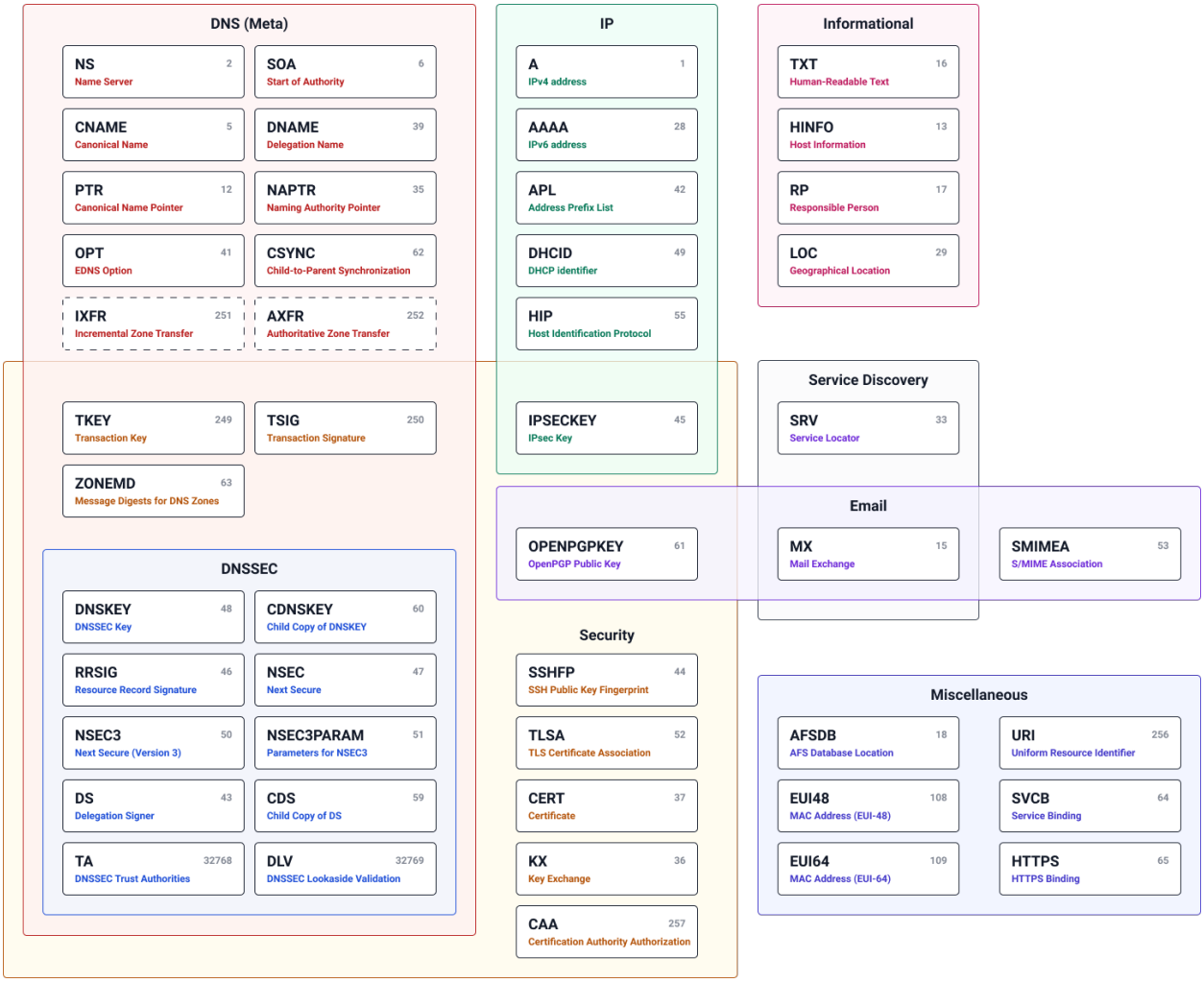


Figure 2: DNS records

- **RP (Responsible Person)**: Specifies the email address of the person responsible for a domain. The format is often a domain name where the user can retrieve the email address.
- **LOC (Location)**: Defines the geographical location (latitude, longitude, altitude) of a domain, though it is not widely implemented.

**4. Service Discovery Records**

- **SRV (Service)**: Used to define the location of servers for specific services, such as SIP or LDAP. SRV records include priority, weight, port, and target.

**5. Email Records**

- **MX (Mail Exchange)**: Specifies mail servers responsible for receiving email on behalf of a domain. Includes a priority value, with lower values indicating higher priority.
- **SMIMEA (S/MIME Association)**: Associates an S/MIME certificate with a domain for secure email transmission.

**6. Security Records**

- **SSHFP (SSH Fingerprint)**: Stores SSH public key fingerprints, allowing secure SSH access verification.
- **TLSA (TLS Authentication)**: Used for DANE (DNS-based Authentication of Named Entities) to bind SSL/TLS certificates to a domain for added security.
- **CERT (Certificate)**: Stores certificates in the DNS, enabling the publication of X.509, OpenPGP, or other public keys.
- **CAA (Certification Authority Authorization)**: Specifies which Certificate Authorities (CAs) are authorized to issue SSL/TLS certificates for a domain, enhancing SSL security.

**7. DNSSEC (DNS Security Extensions) Records**

- **DNSKEY**: Holds public keys used in DNSSEC for authenticating DNS responses.
- **RRSIG (Resource Record Signature)**: Stores cryptographic signatures for DNS records, verifying their authenticity.
- **DS (Delegation Signer)**: Links a child zone to its parent in DNSSEC, creating a chain of trust.
- **NSEC and NSEC3**: Used to provide authenticated denial of existence for DNS records.
- **CDNSKEY/CDS**: Allow automatic DNSSEC delegation updates in the parent zone, easing key management.

**8. Miscellaneous Records**

- **AFSDB (AFS Database)**: Used by the Andrew File System (AFS) to locate database servers, though it's rarely encountered in modern usage.
- **URI (Uniform Resource Identifier)**: Stores URIs associated with a domain for various service endpoints.
- **SVCB (Service Binding)** and **HTTPS**: Emerging records designed to make DNS service-aware, particularly for directing HTTPS traffic.
- **EUI48 and EUI64 (MAC Addresses)**: Store device-specific MAC addresses in the DNS for identification purposes.

## Summary

DNS architecture, with its tiered resolver system and reliance on recursive querying, supports the seamless and efficient lookup of domain names across a globally distributed database. Each DNS record type serves a specific role, from basic IP resolution to security, service discovery, and metadata. These details are critical for network engineers, as they provide tools to manage DNS traffic, enhance security, and optimize performance.