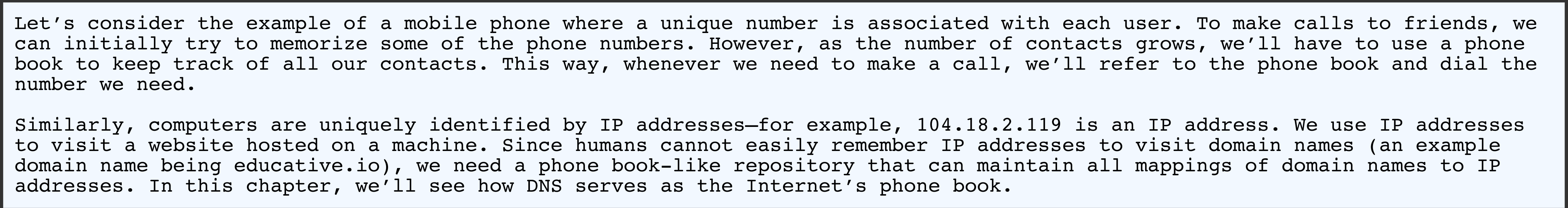
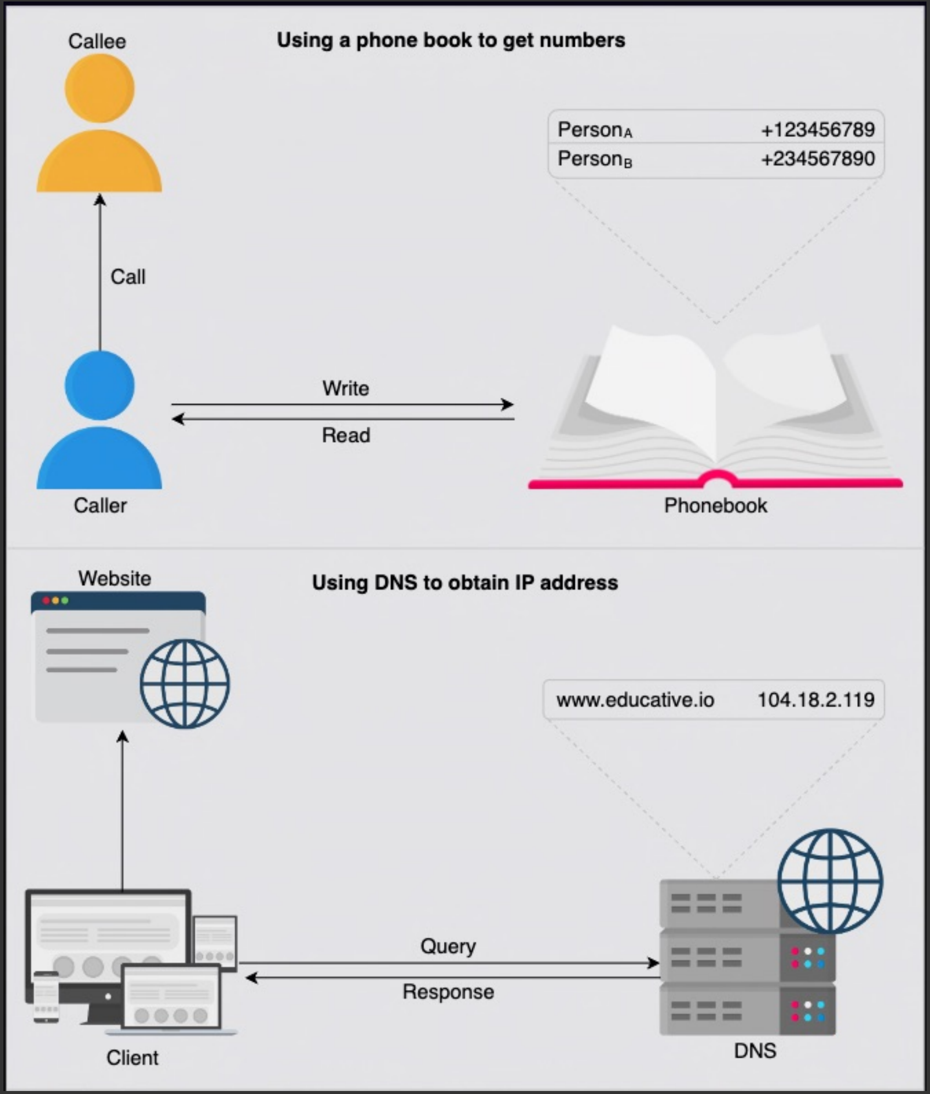
The Origin of DNS

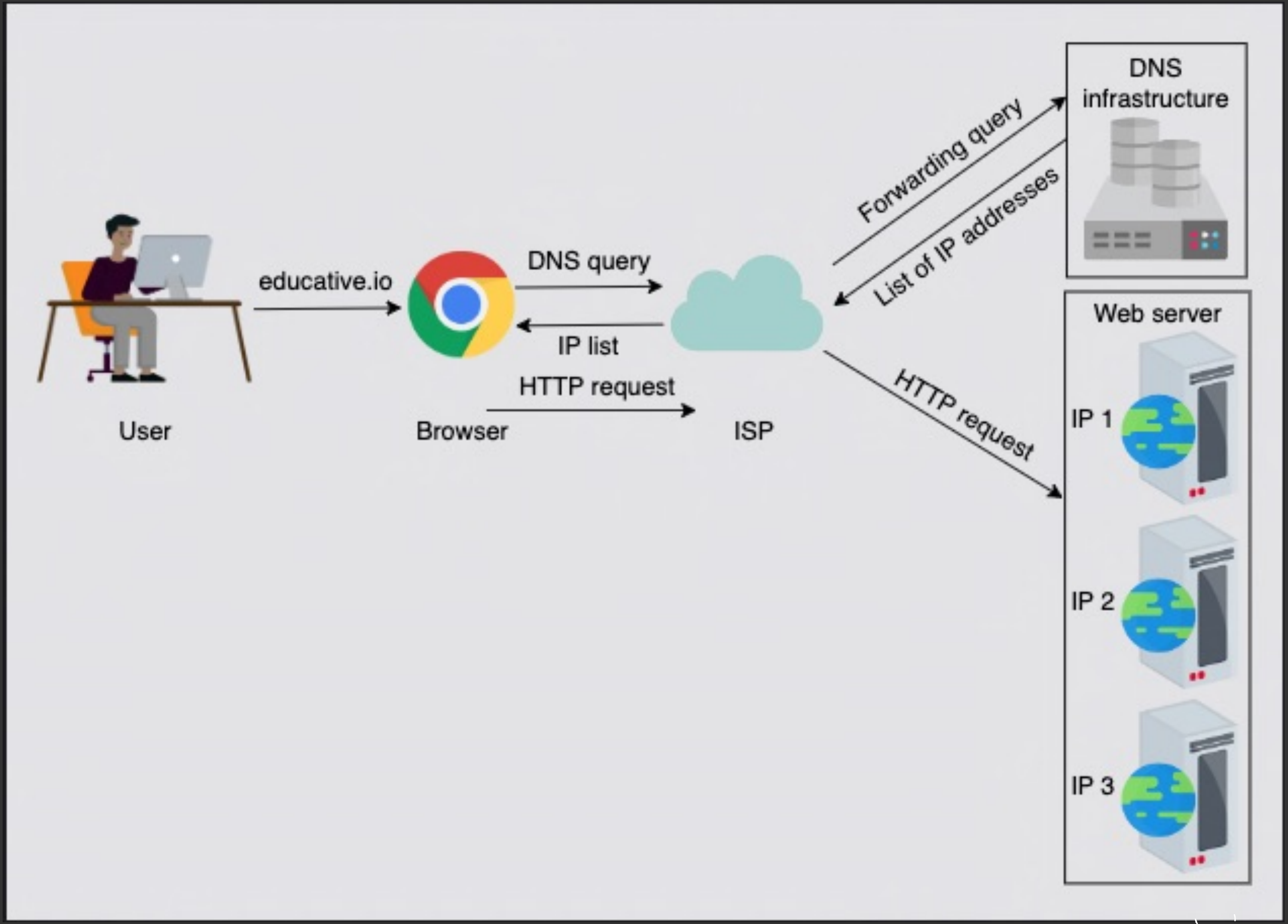


Let’s consider the example of a mobile phone where a unique number is associated with each user. To make calls to friends, we can initially try to memorize some of the phone numbers. However, as the number of contacts grows, we’ll have to use a phone book to keep track of all our contacts. This way, whenever we need to make a call, we’ll refer to the phone book and dial the number we need.

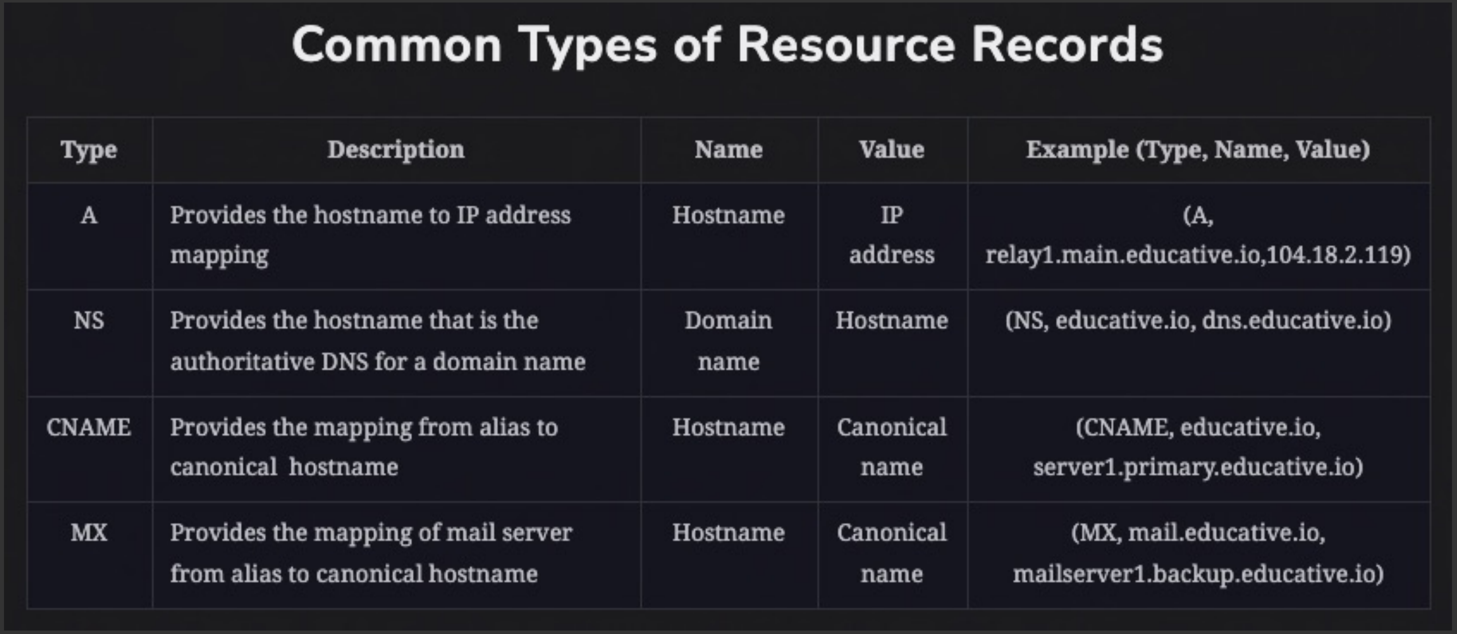
Similarly, computers are uniquely identified by IP addresses—for example, 104.18.2.119 is an IP address. We use IP addresses to visit a website hosted on a machine. Since humans cannot easily remember IP addresses to visit domain names (an example domain name being educative.io), we need a phone book-like repository that can maintain all mappings of domain names to IP addresses. In this chapter, we’ll see how DNS serves as the Internet’s phone book.



The Domain name system (DNS) is the internet naming service that maps human-friendly domain names to machine-readable IP addresses.



The entire process is very fast, so that user have minimum delay.



Common Types of Resource Records

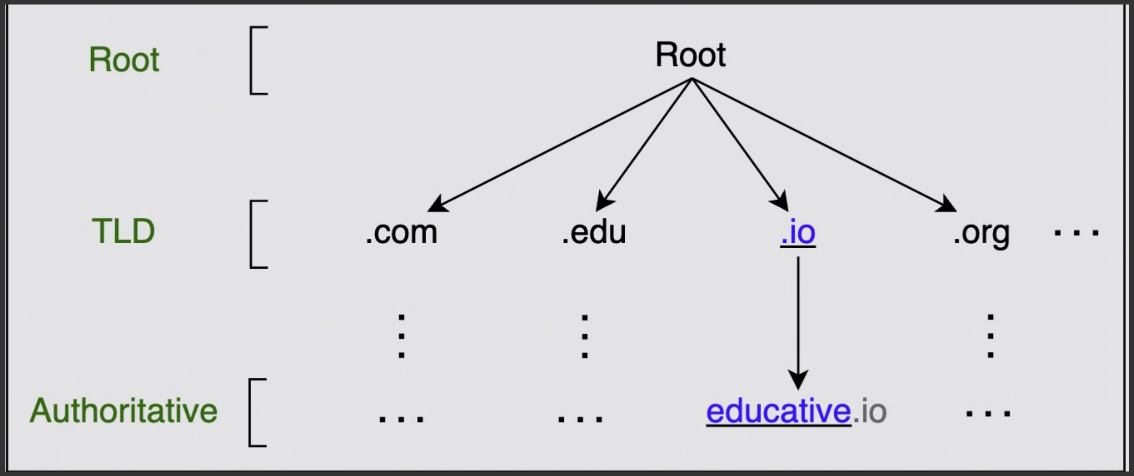
Type	Description	Name	Value	Example (Type, Name, Value)
A	Provides the hostname to IP address mapping	Hostname	IP address	(A, relay1.main.educative.io,104.18.2.119)
NS	Provides the hostname that is the authoritative DNS for a domain name	Domain name	Hostname	(NS, educative.io, dns.educative.io)
CNAME	Provides the mapping from alias to canonical hostname	Hostname	Canonical name	(CNAME, educative.io, server1.primary.educative.io)
MX	Provides the mapping of mail server from alias to canonical hostname	Hostname	Canonical name	(MX, mail.educative.io, mailserver1.backup.educative.io)

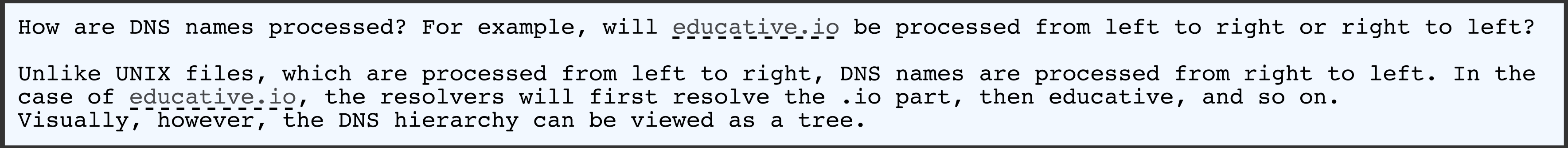
Caching → DNS uses caching at different layers to reduce latency.

Hierarchy → DNS name servers are in a hierarchical form. The hierarchical structure allows DNS to be highly scalable when load increases.

DNS hierarchy -

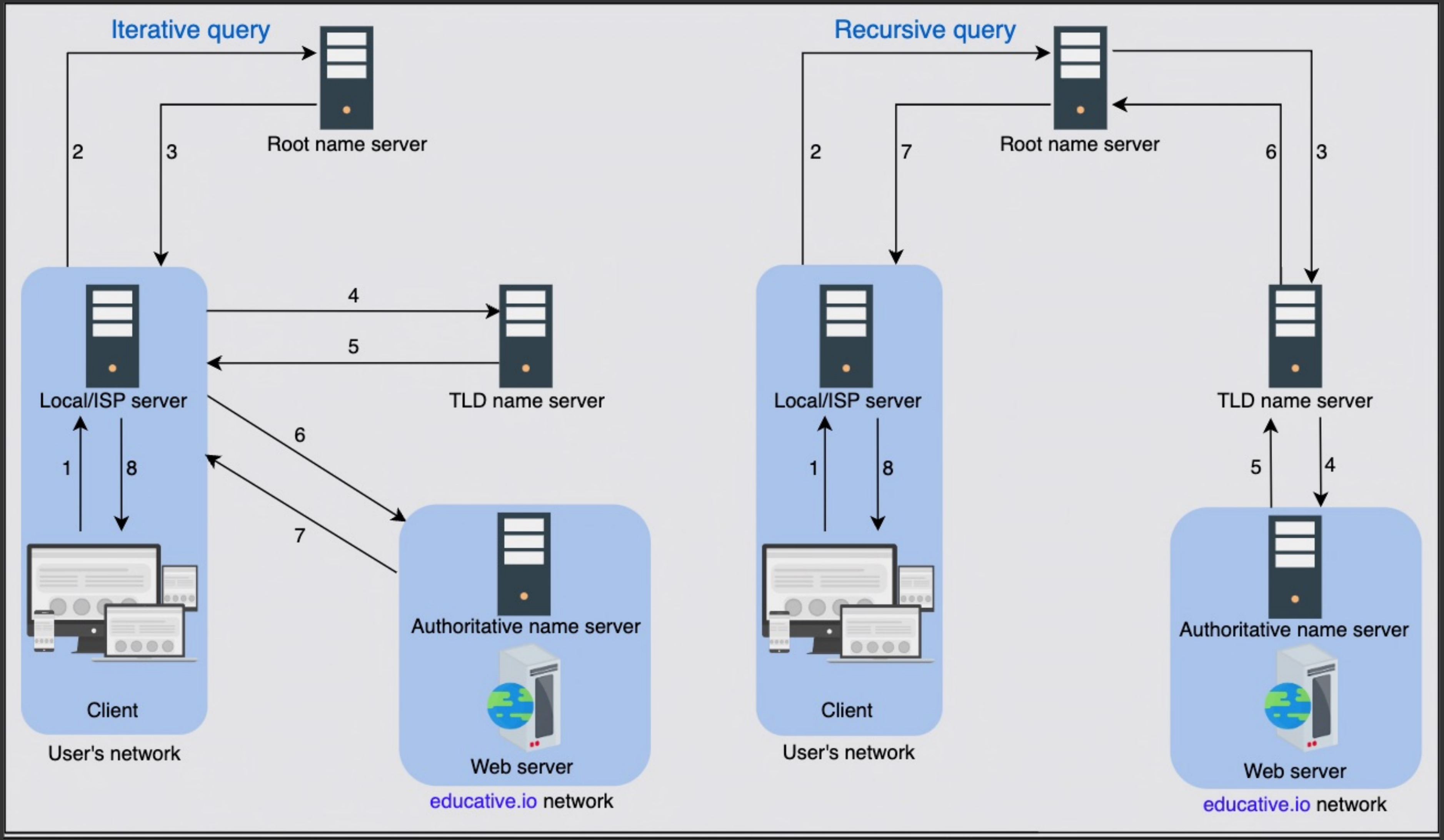
- DNS Resolver
- Root-level name servers
- Top-level Domain servers
- Authoritative name servers.





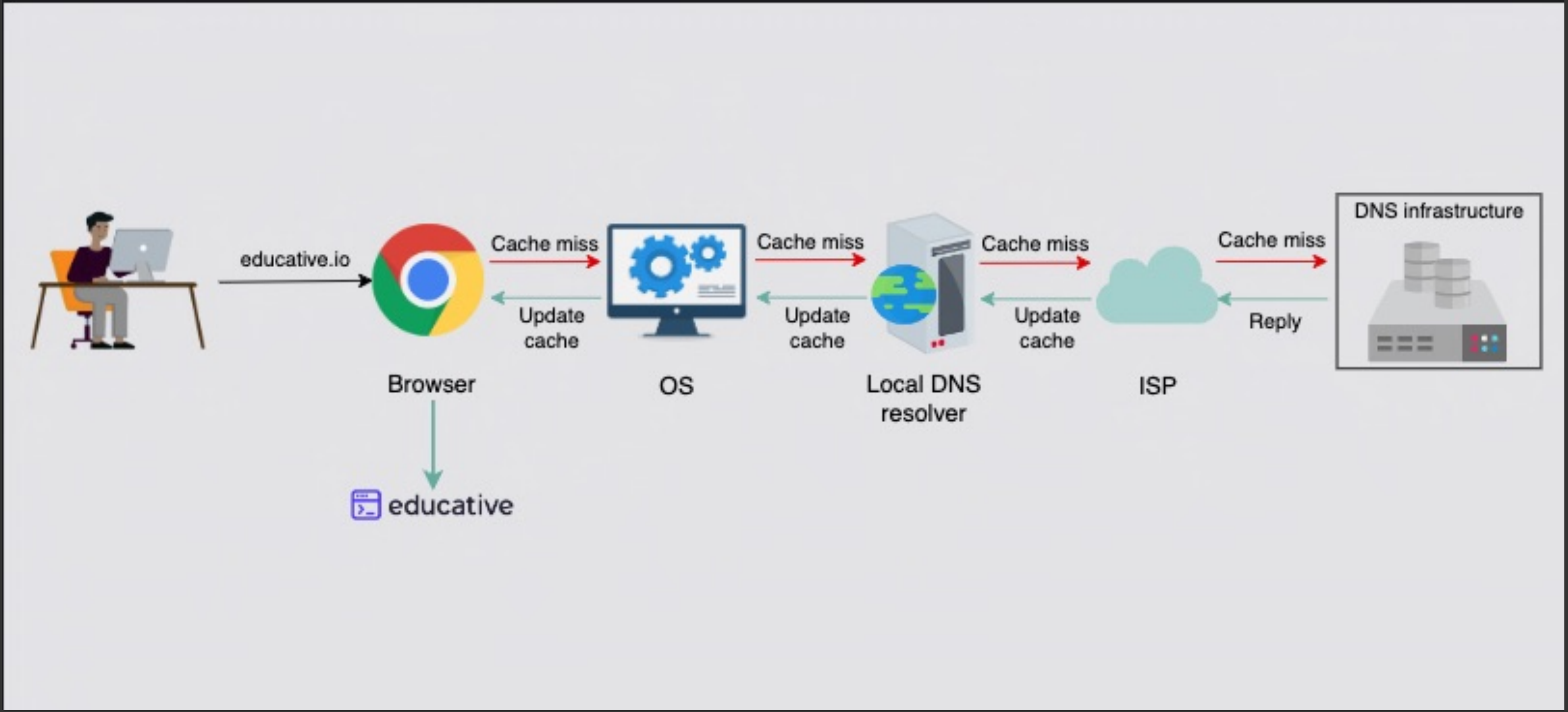
How are DNS names processed? For example, will educative.io be processed from left to right or right to left?

Unlike UNIX files, which are processed from left to right, DNS names are processed from right to left. In the case of educative.io, the resolvers will first resolve the .io part, then educative, and so on.
Visually, however, the DNS hierarchy can be viewed as a tree.

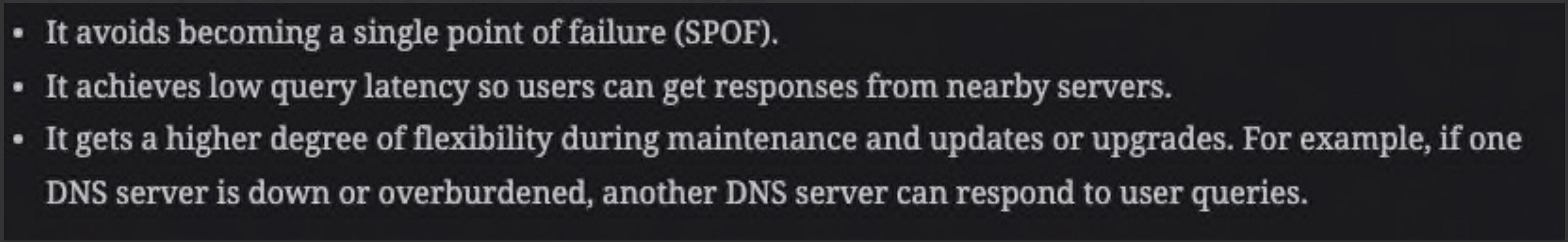


How Does DNS query is resolved.

How Does Caching Working

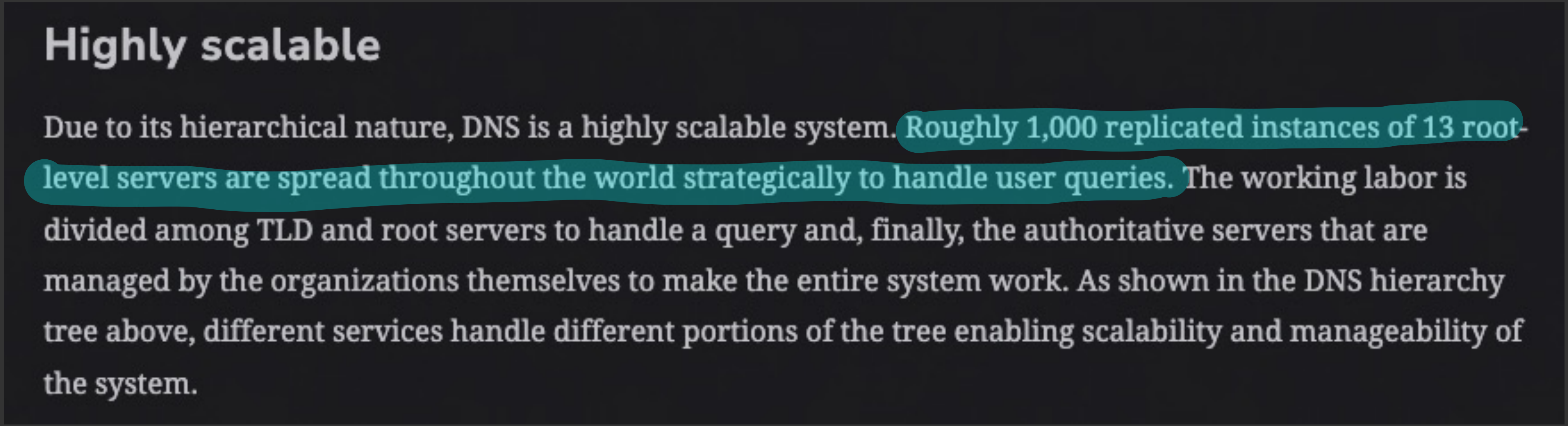


DNS as a Distributed System



- It avoids becoming a single point of failure (SPOF).
- It achieves low query latency so users can get responses from nearby servers.
- It gets a higher degree of flexibility during maintenance and updates or upgrades. For example, if one DNS server is down or overburdened, another DNS server can respond to user queries.

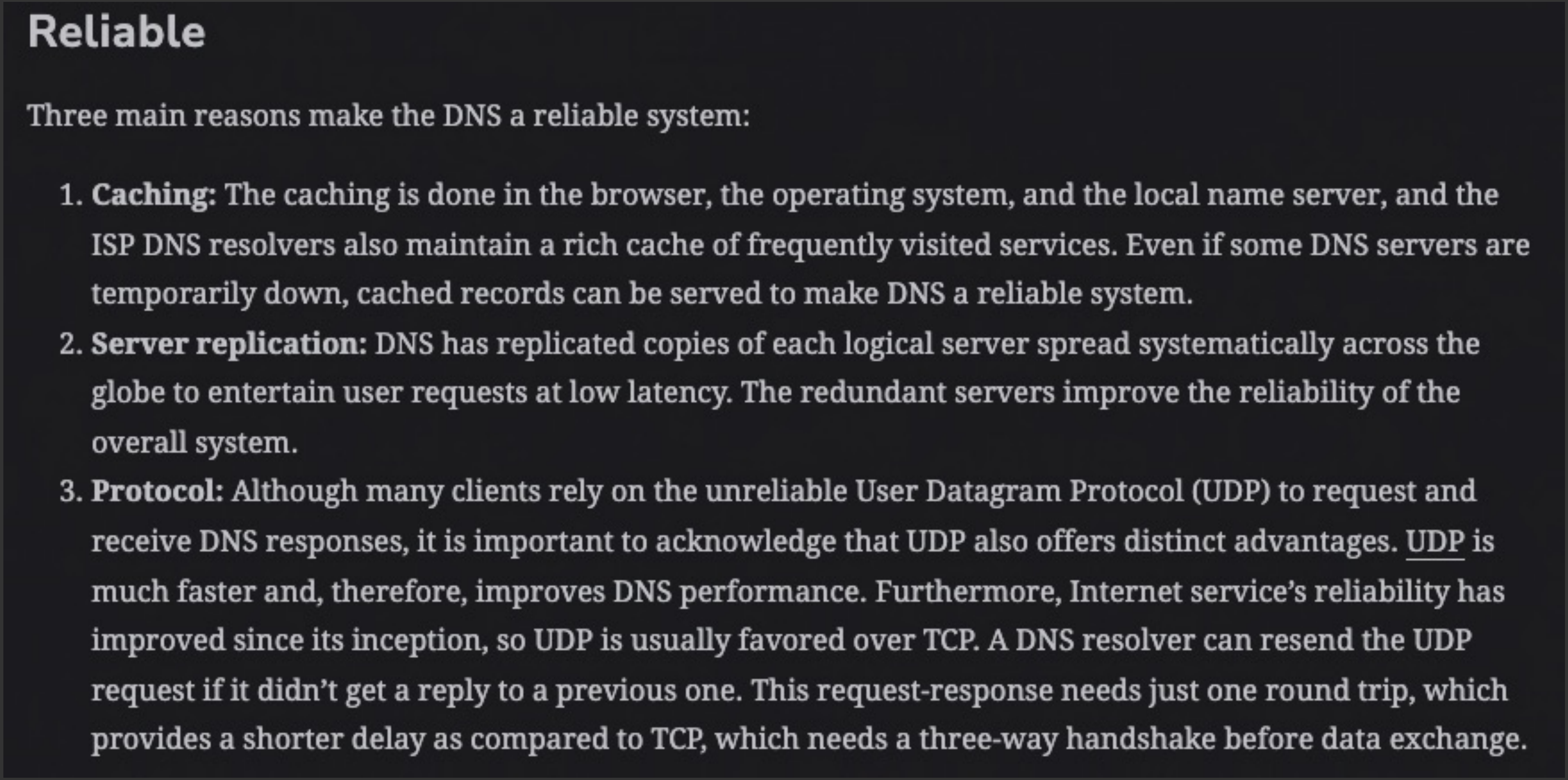
How DNS is highly Scalable



Highly scalable

Due to its hierarchical nature, DNS is a highly scalable system. Roughly 1,000 replicated instances of 13 root-level servers are spread throughout the world strategically to handle user queries. The working labor is divided among TLD and root servers to handle a query and, finally, the authoritative servers that are managed by the organizations themselves to make the entire system work. As shown in the DNS hierarchy tree above, different services handle different portions of the tree enabling scalability and manageability of the system.

How does DNS is reliable ?

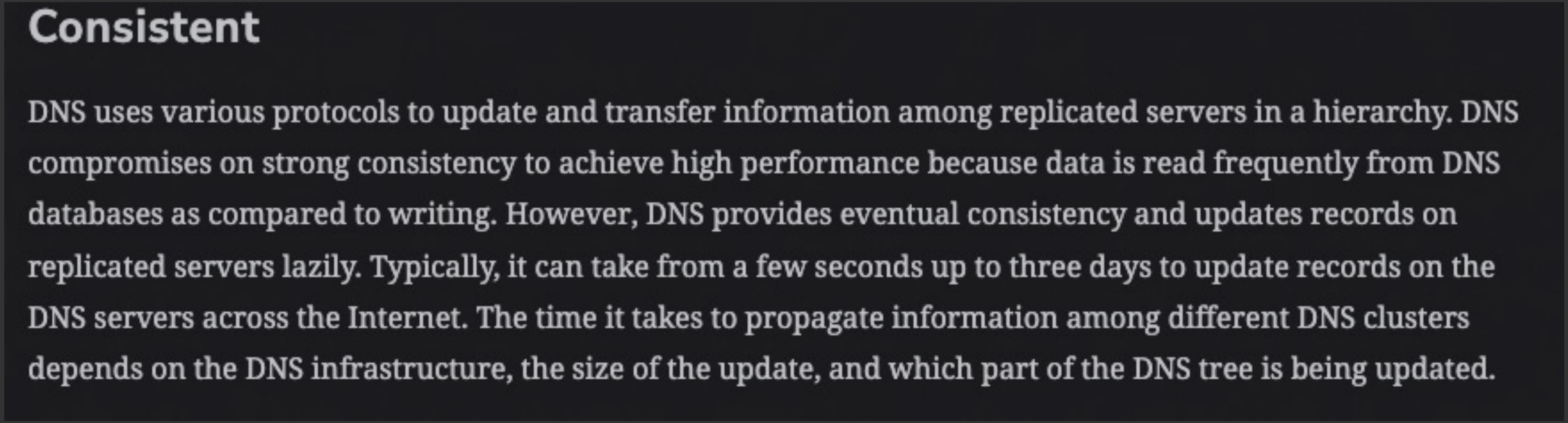


Reliable

Three main reasons make the DNS a reliable system:

1. **Caching:** The caching is done in the browser, the operating system, and the local name server, and the ISP DNS resolvers also maintain a rich cache of frequently visited services. Even if some DNS servers are temporarily down, cached records can be served to make DNS a reliable system.
2. **Server replication:** DNS has replicated copies of each logical server spread systematically across the globe to entertain user requests at low latency. The redundant servers improve the reliability of the overall system.
3. **Protocol:** Although many clients rely on the unreliable User Datagram Protocol (UDP) to request and receive DNS responses, it is important to acknowledge that UDP also offers distinct advantages. UDP is much faster and, therefore, improves DNS performance. Furthermore, Internet service's reliability has improved since its inception, so UDP is usually favored over TCP. A DNS resolver can resend the UDP request if it didn't get a reply to a previous one. This request-response needs just one round trip, which provides a shorter delay as compared to TCP, which needs a three-way handshake before data exchange.

How DNS is consistent ?



Consistent

DNS uses various protocols to update and transfer information among replicated servers in a hierarchy. DNS compromises on strong consistency to achieve high performance because data is read frequently from DNS databases as compared to writing. However, DNS provides eventual consistency and updates records on replicated servers lazily. Typically, it can take from a few seconds up to three days to update records on the DNS servers across the Internet. The time it takes to propagate information among different DNS clusters depends on the DNS infrastructure, the size of the update, and which part of the DNS tree is being updated.