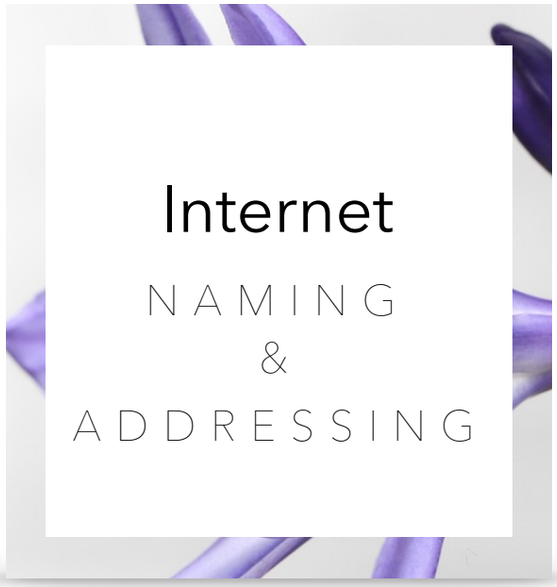


50.005 CSE

Natalie Agus
Information Systems Technology and Design
SUTD

https://websitename.com

↓

THE DOMAIN NAME SYSTEM (DNS)

Application layer protocol,
Distributed database

↓

32-bit IP address

After IP address is obtained, then end hosts can communicate with one another

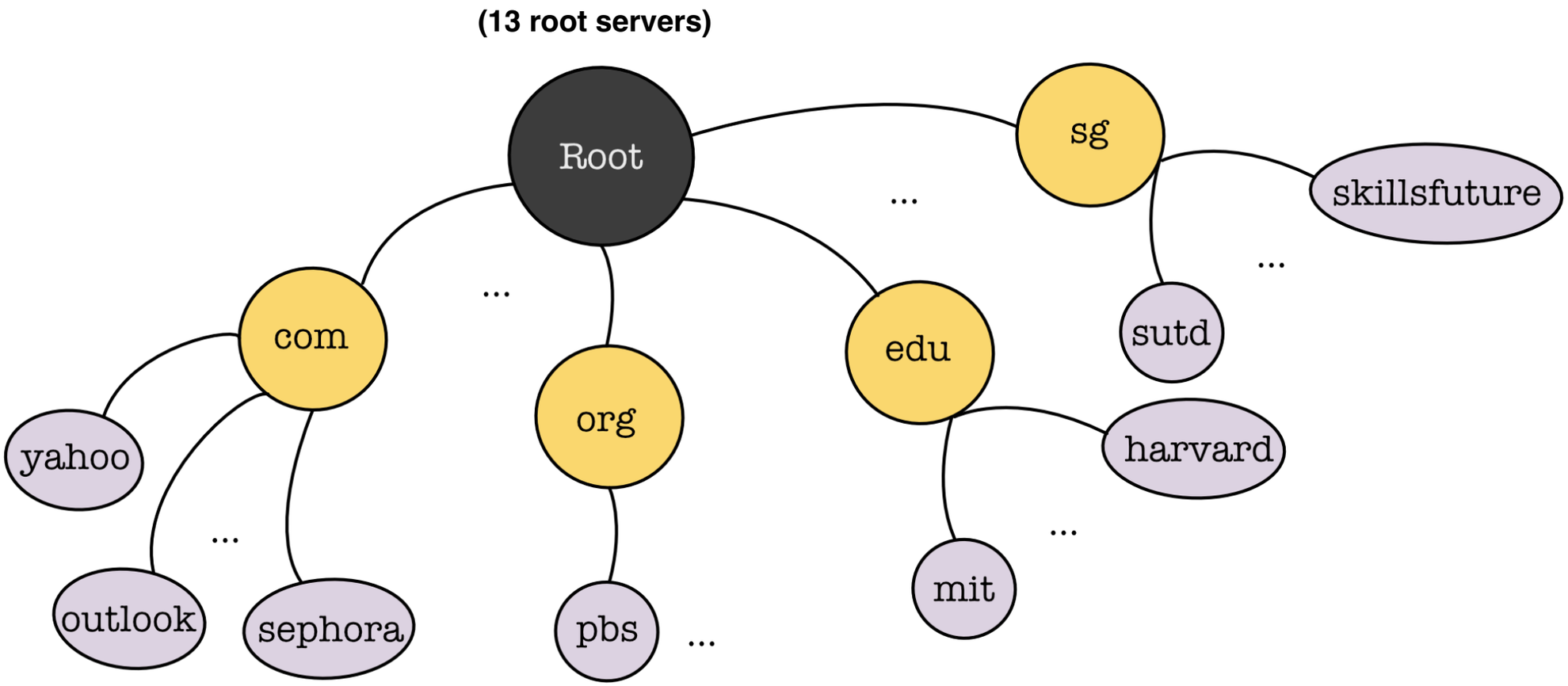
DNS SERVICES

1 **Hostname to IP translation**

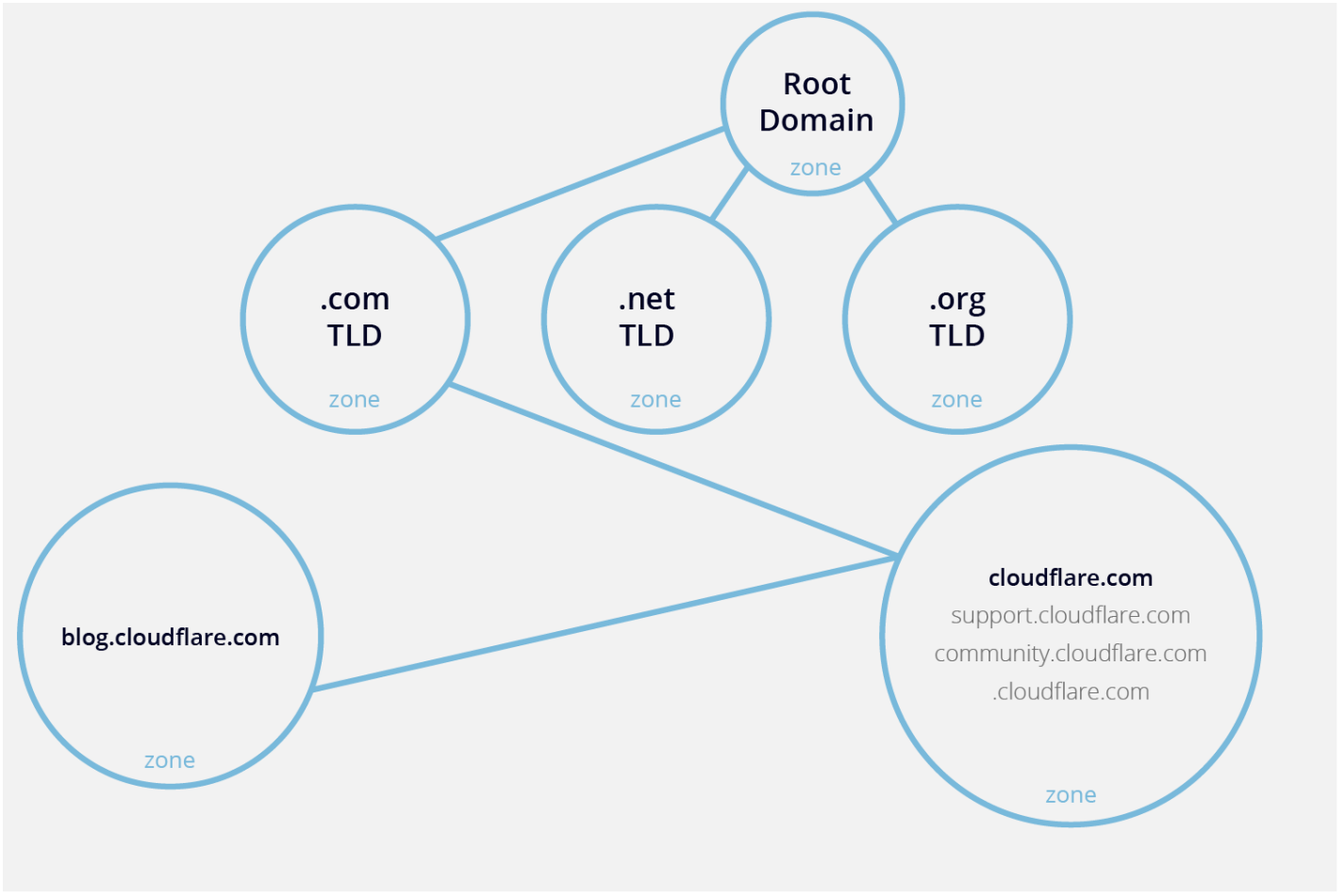
2 **Host aliasing**

3 **Mail server aliasing**

4 **Load distribution**



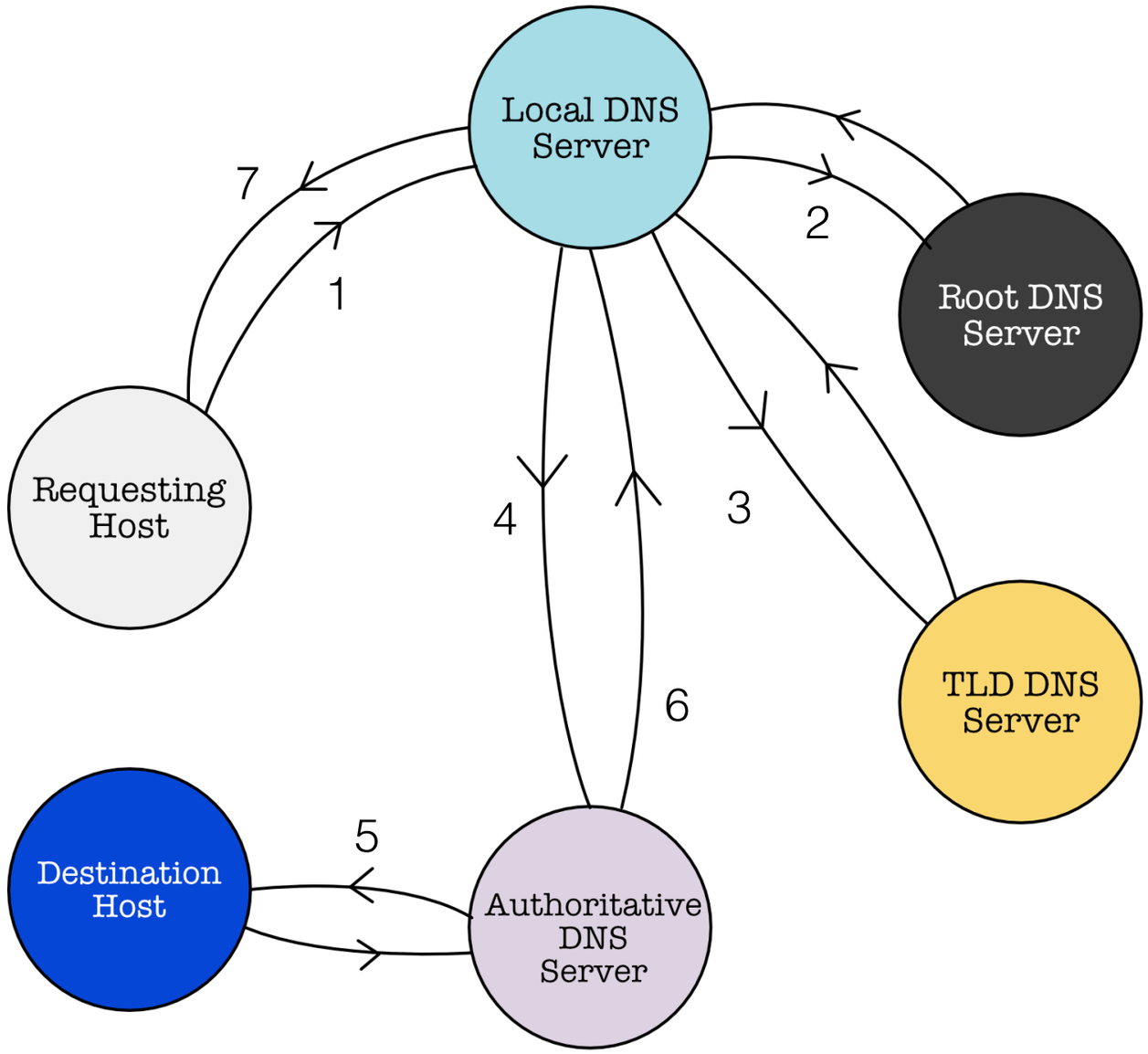
DISTRIBUTED, HIERARCHICAL DNS DATABASE



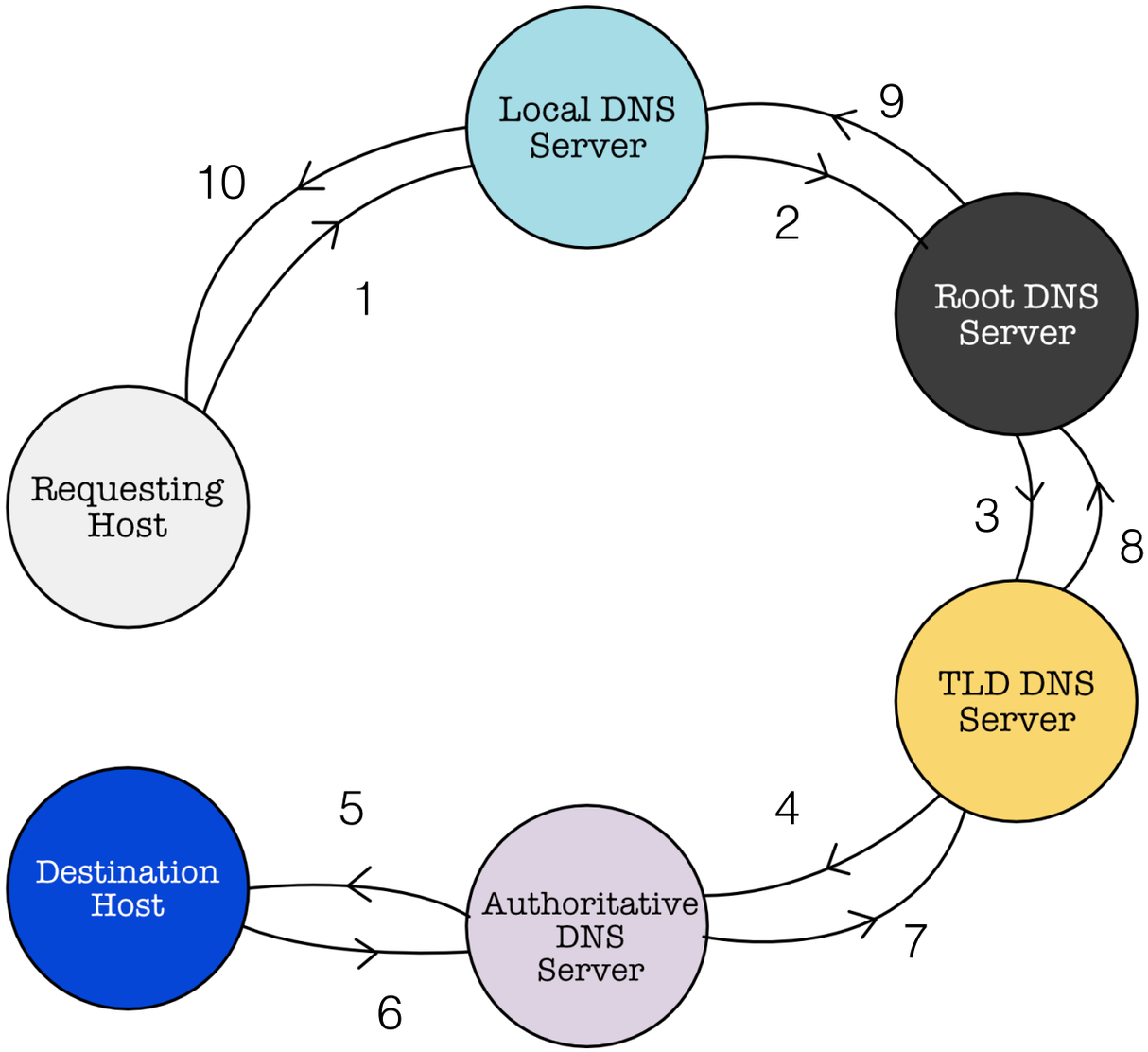
SOME TERMINOLOGIES

1. **DNS Namespace:** The entire tree (all nodes)
2. **Domain:** represents a the entire set of names / machines that are contained under an organizational domain name (subtree), eg: ".com" websites are part of the "com" domain, **en.wikipedia.com** and **id.wikipedia.com** websites are part of the "**wikipedia.com**" domain.
3. **Zone:** DNS namespace can be broken into zones for which **individual** DNS servers are **responsible**. Unlike "domain" that's a subtree, a zone can just be a bunch of sibling nodes. A DNS zone is not necessarily associated with one server, so two servers or more can **store the same copy of the zone**. A DNS server can also contain multiple zones.

DNS NAME RESOLUTION
1. ITERATIVE QUERY



DNS NAME RESOLUTION
2. RECURSIVE QUERY



DNS CACHING

Once local name servers learns hostname-ip mapping, it **caches** the mapping

- TLD servers are typically cached in DNS local name servers, **hence faster resolution**
- Cached entries may be **out of date**
- **DNS authoritative name server (the one that hosted the website) decides the DNS record TTL**
- DNS local name server re-queries when TTL expires

DNS RECORDS

The “data structure” of a DNS resource record (RR). RR is used by clients who query hostname-IP resolution

RR = (name, value, type, TTL)

Name	Value	Type	TTL
hostname, e.g en.wikipedia.com, or mywebsite.net	IP address	**A (Authoritative)**	TTL
A domain or hostname in question, e.g wikipedia.com (for wiki domain), or a.gtld.net (for .net domain)	A nameserver that serves this domain, e.g: en.wikipedia.com, ns1.mywebsite.net — they MIGHT know the IP of this hostname	**NS (Name Server)**	TTL
alias name for some canonical (real) hostname, e.g: alias.mywebsite.net	canonical hostname name, e.g: mywebsite.net	**CNAME (Canonical Name)**	TTL
domain, e.g: example.com	mailserver name, e.g: mail.example.com	**MX (Mail)**	TTL

DNS RECORDS

The “data structure” of a DNS resource record (RR). RR is used by clients who query hostname-IP resolution

RR = (name, value, type, TTL)

Types:

1. **A: Authoritative,** it contains the IP address of the hostname in question. No longer delegates you to further probing the query and you should go to that IP address
2. **NS: Name Server,** it tells which nameservers are authoritative for a **zone (a collection of computer network, e.g: SUTD.com zone, Wiki zone.com, etc).** You can have multiple NS records for load distribution (increasing availability)
3. **CNAME: Canonical Name**, just another alias for a hostname. Requires more probing (queries) to resolve to an IP eventually. It is like the same website with two or more URLs.
4. **MX: Mail server**, points you to the mailserver name responsible for your domain.

DNS RECORDS

An example: You want to know the IP of **b.example.com**. You ask **localDNSserver.com** (A local DNS nameserver whose IP you know, usually a public knowledge), it has the following records:

Name	Value	Type	TTL
b.example.com	NS2.server.com	**NS**	TTL
NS2.server.com	111.222.125.124	**A**	TTL

So your **localDNSServer.com** do not know what is **b.example.com**'sIP, but it has an NS entry regarding it. It means that we should ask **NS2.server.com** on where is **b.example.com**. It has an A record of **NS2.server.com** so we (or the localDNSServer) can go and ask **NS2.server.com** if it has the A record of **b.example.com**. Now suppose **NS2.server.com** has the following record:

Name	Value	Type	TTL
a.example.com	10.12.14.145	**A**	TTL
b.example.com	a.example.com	**CNAME**	TTL

DNS RECORDS

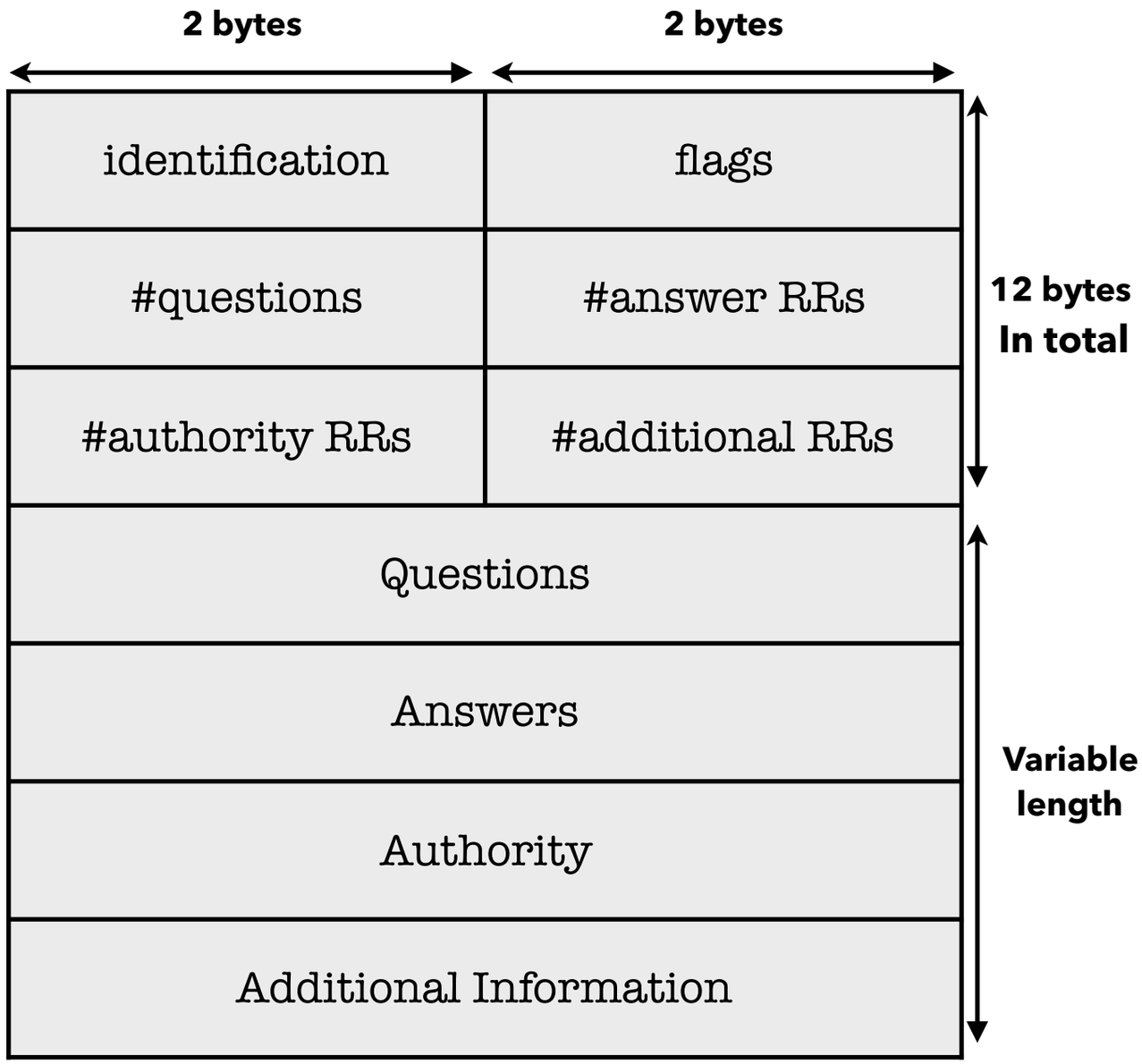
An example (continued): Now the first query to **NS2.server.com** shows that **b.example.com** is a CNAME record, so it is simply an alias of another hostname called **a.example.com**

Name	Value	Type	TTL
a.example.com	10.12.14.145	**A**	TTL
b.example.com	a.example.com	**CNAME**	TTL

Hence, you, or the local DNS server has to continue probing and send another DNS query asking where is a.example.com, and finally NS2.server.com returns the A record with IP of 10.12.14.145.

DNS PROTOCOL

This is the protocol to make a DNS query or reply.
Both query and reply has the same message format



- Identification: 16bit # for query, reply uses the same #
- Flags:
  - Q/R
  - Recursion desired / available
  - Reply is authoritative?

INSERTING DNS RECORDS

How do you insert (register) DNS records so people in the internet can find your website?

1. Register yournewwebsite.com at **DNS registrar,** e.g: Verisign registry
2. You need to provide names, IP addresses of authoritative name server (**primary and secondary server as backup**)
3. Verisign stores the A records of your website in its name servers: ns01 and ns02
4. Verisign then inserts the RRs into **.com TLD server,** e.g:
    - yournewwebsite.com, ns01-verisign.com , NS
    - yournewwebsite.com, ns02-verisign.com NS
    - ns01-verisign.com, ZZZ.ZZZ.ZZZ.Z, A
    - ns02-verisign.com, ZZZ.ZZZ.ZZZ.Z, A

ATTACKING DNS

- **DDoS Attack**
  - Bombard root server with traffic:
    - Not successful to date, due to implemented traffic filtering
    - Local DNS servers also caches TLD servers, hence bypass root
  - Bombard TLD servers: potentially more dangerous

- **Redirect Attack**
  - Man-in-the-middle: intercept queries
  - DNS poisoning: sends bogus replies to DNS servers and affect their cache

- **Exploit DNS for DDoS:**
  - Send queries with spoofed source address, because you want to target it.
  - Requires amplification