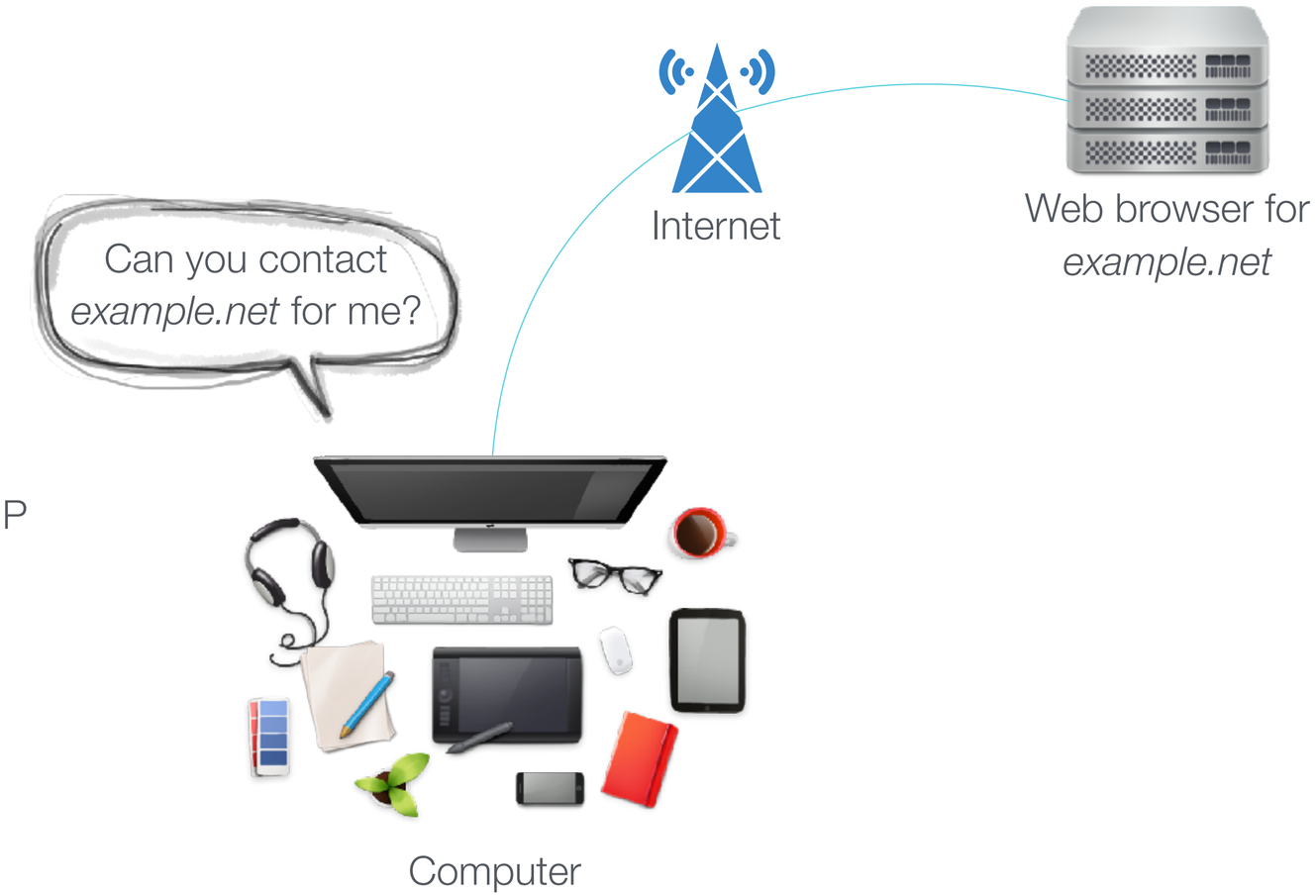
INSIDE THE DOMAIN NAME SYSTEM

Understanding DNS

INTRODUCTION

- Your on your PC in your web browser trying to reach *example.net*
- The PC and the web server are both connected to the Internet
- Usually your browser will automatically contact the web server for you and request the home page
- Although you know the domain name, the Internet runs on IP addresses
- You can't just ask… "Hey can you contact *example.net* for me?"
- The Internet *can't* because it only knows IP addresses, and this is where **DNS** comes in

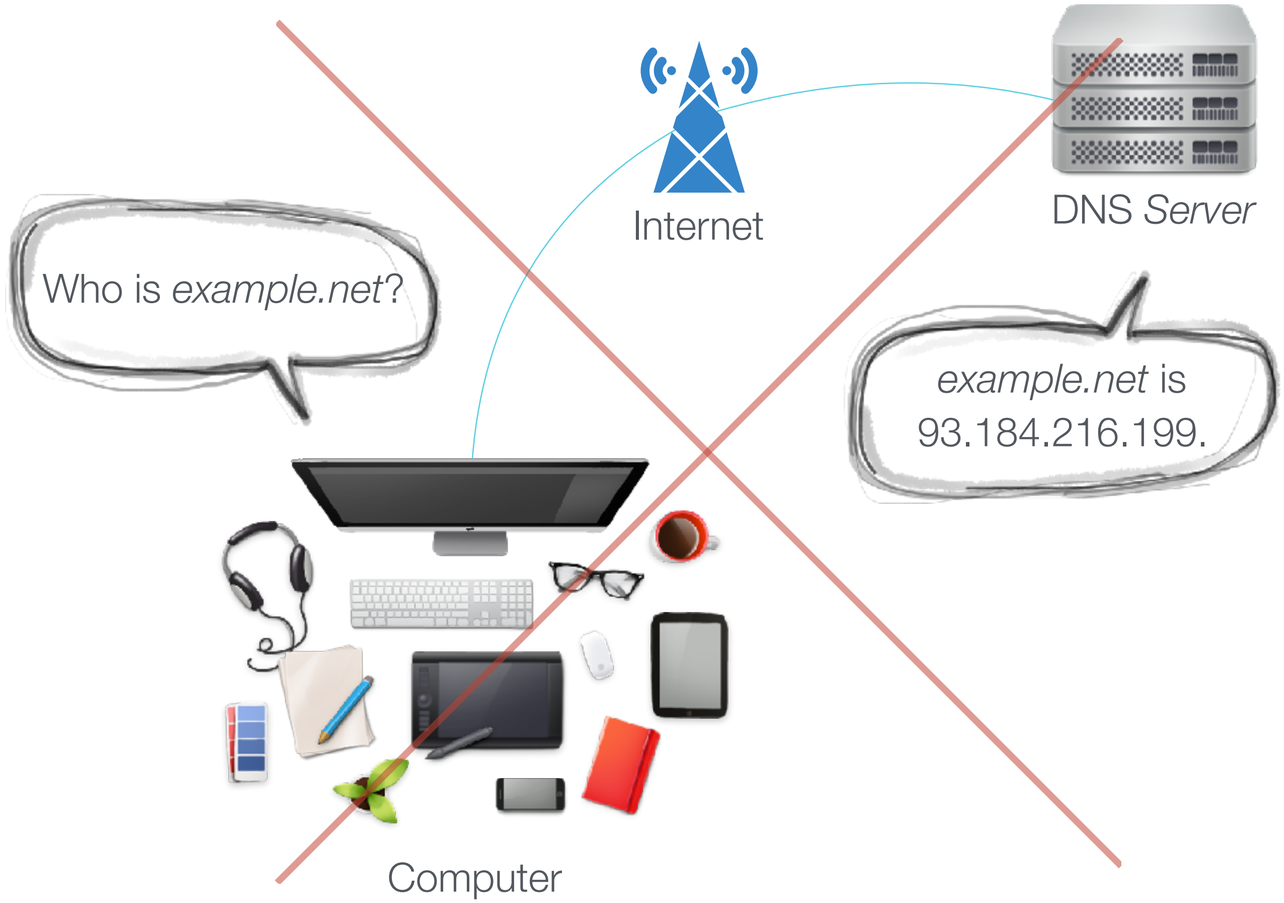


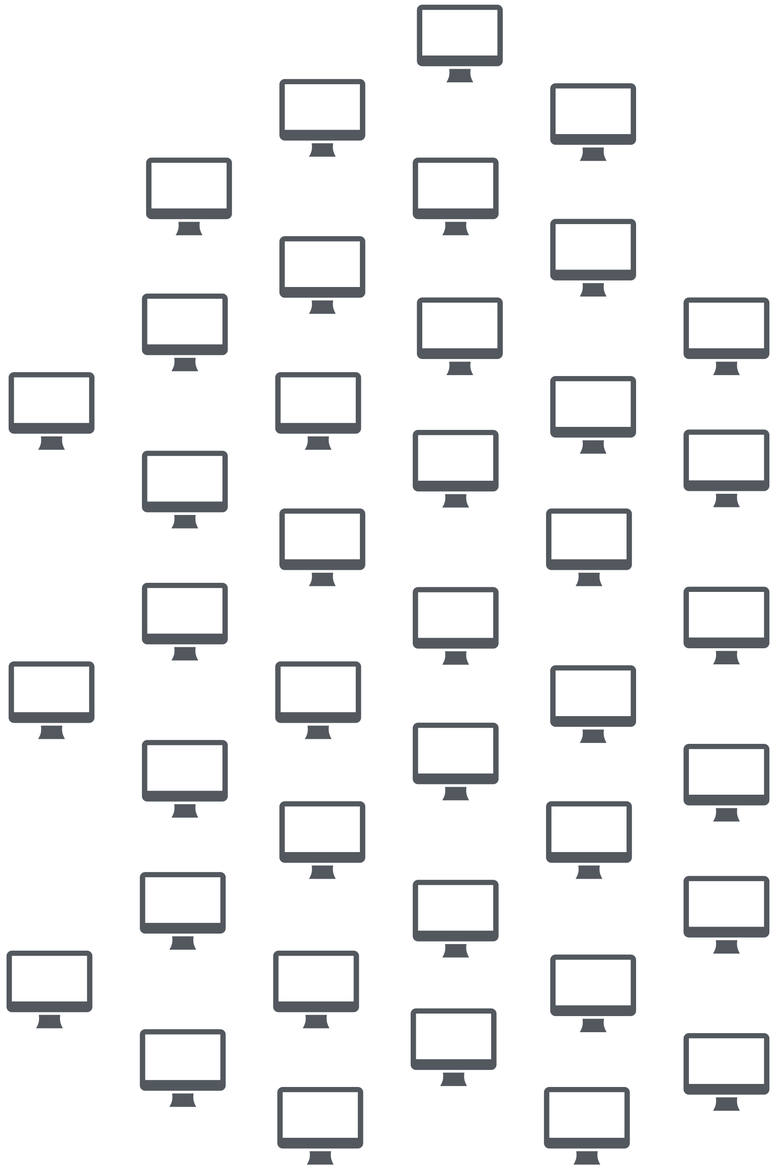
Think of **DNS** as a giant address book that tells you at which IP address any domain name can be contacted.

LOOKING UP AN ADDRESS

To look up an address, you ask a special **DNS server** something like, “Who is example.net?”

- It will tell you the answer right? **Not really.**
- You are not the only one on the Internet…





WHY?

There are **billions** of other Internet connected devices that are also trying to look up IP addresses at the same time.

If there was only *one* DNS server, it would be **overloaded** with requests immediately.

Even with more servers, you do not want **one organization** managing the entire DNS system (a lot of power and responsibility).

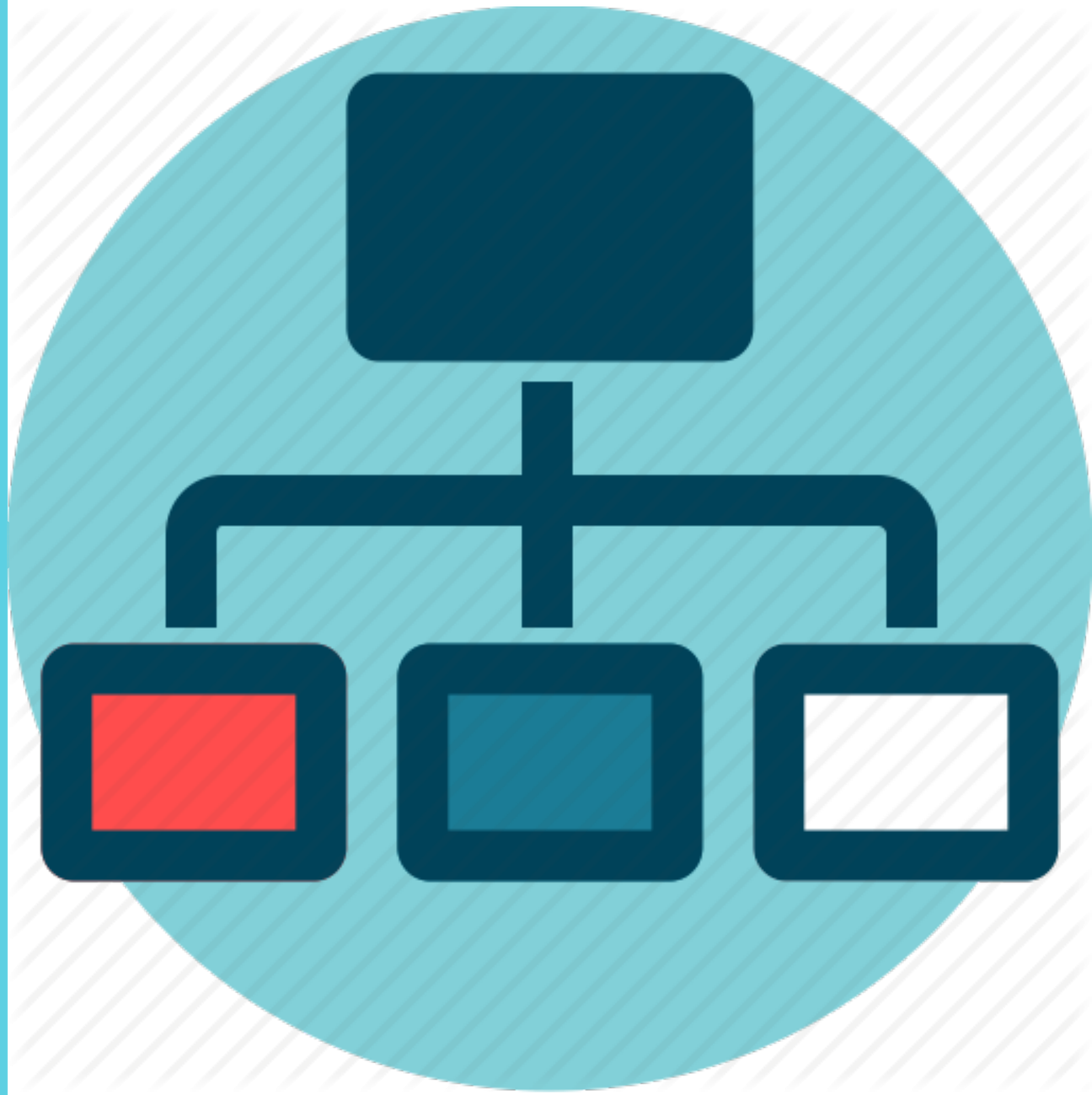
Targeting this one organization would be too risky for the **security** of DNS across the web.

It also makes the Internet more susceptible to **government interference**.

IN REALITY DNS IS MORE OF A **DISTRIBUTED GROUP EFFORT**.

THERE IS STILL A **HIERARCHY** BUT RESPONSIBILITIES ARE **DIVIDED**.

DNS Hierarchy



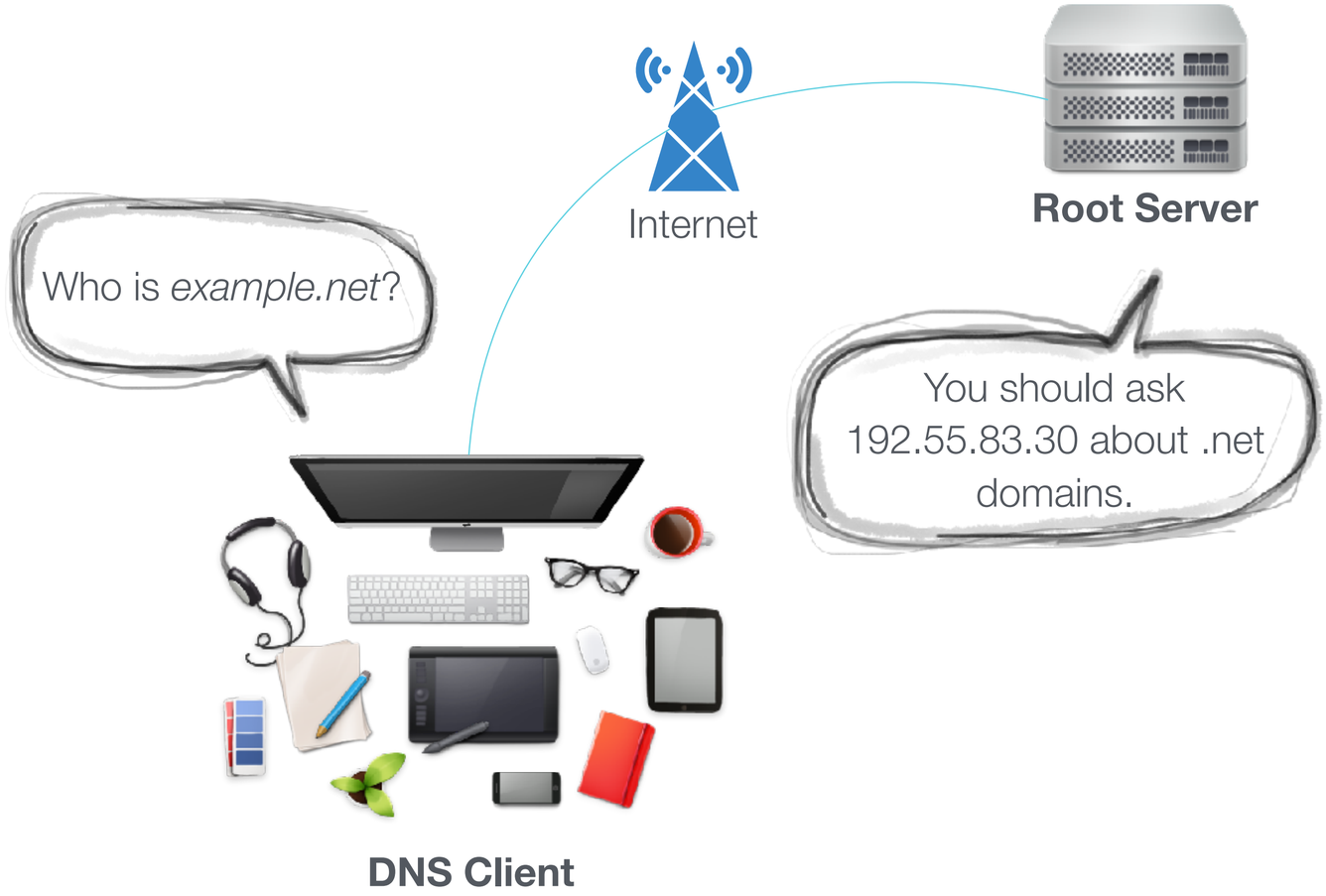
DNS SERVER HIERARCHY

There are more than 380 **root servers** worldwide (as of August 2014) which are divided into **13 groups**.

Root servers know other DNS servers that can help you with specific **top-level domain names**.

If you ask it, “Who is *example.net*?” it will not know the answer…

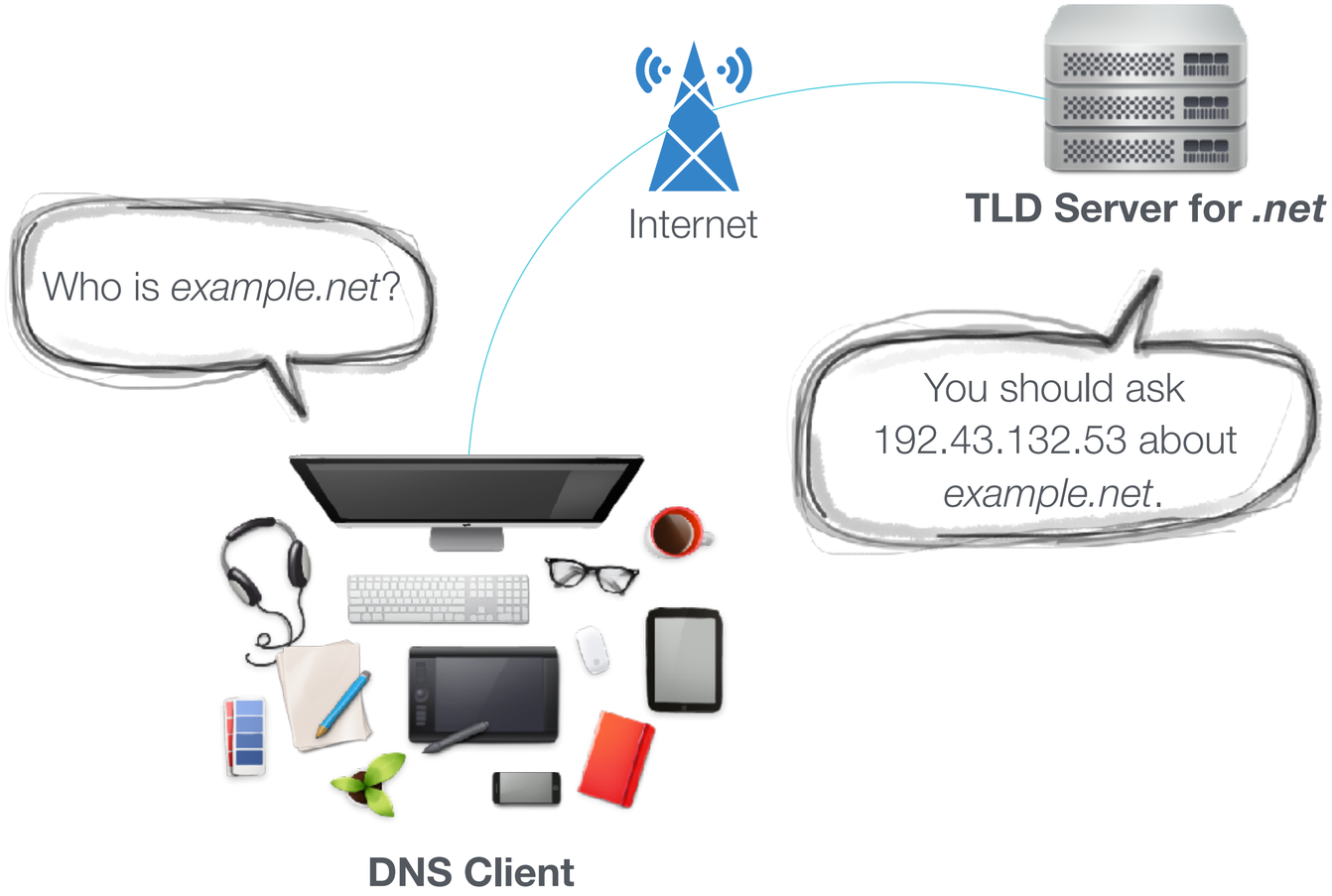
But it ***will know*** which servers are responsible for all the .net domains:



THE TLD LAYER

Your PC has arrived at the **Top-Level Domain Layer** of the hierarchy and can move on to find it's destination.

You will get a **list of IP addresses** of servers that are responsible for queries regarding *example.net* and it's **subdomains**.

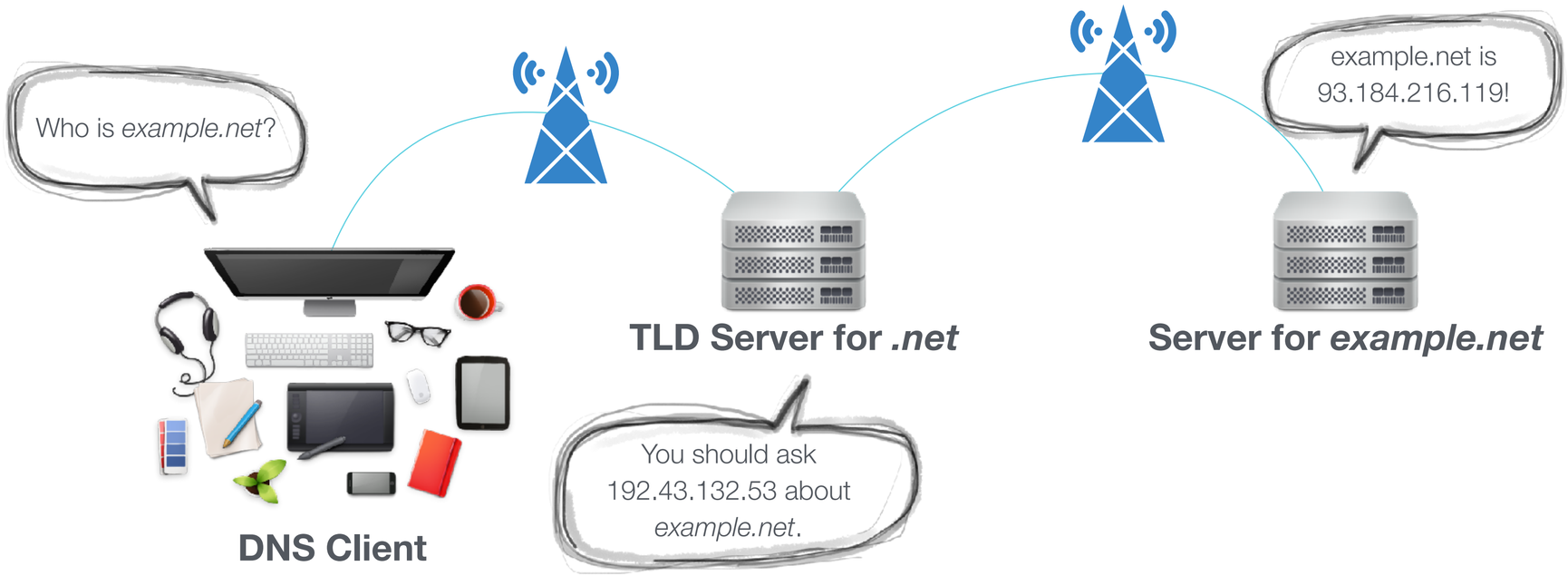


THE DOMAIN NAME LAYER

Your PC then moves on to the **Domain Name Layer**.

This is the final step.

The next server you reach will give you the **correct IP address**.



BIRD'S EYE VIEW OF DNS HIERARCHY

Layer 1 - Root Servers

Root Servers

A **root server** told us what DNS servers are responsible for .net domains.

Layer 2 - TLD Servers

TLD Servers for .net

A **TLD server** told us which DNS servers are responsible for *example.net.*

Layer 3 - DNS Servers

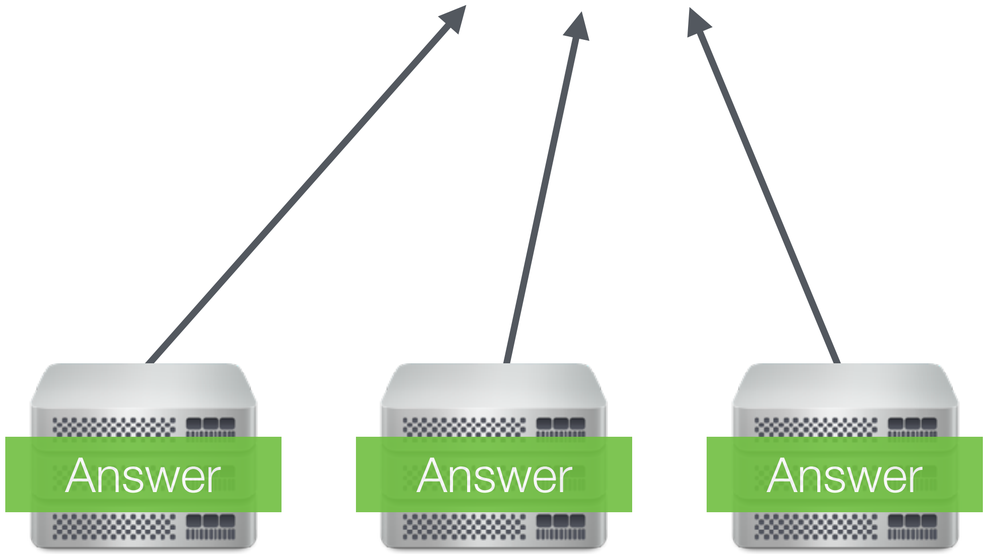
Servers for *example.net*

Finally, one of *example.net*'s **DNS servers** told us which IP address is associated with that website.

BIRD'S EYE VIEW OF DNS HIERARCHY

After going through this maze of DNS servers, the servers at the last level are the ones that have the IP address a client is looking for.

They are set to be **Authoritative** servers for the site you are looking for.



Layer 3 - DNS Servers

Client

Request

Local DNS Server (Usually located at your ISP)

Your device will hand off the request to your **local DNS server**.

This server will **play the DNS client on your behalf** and send the answer when it is ready.

The client contacts a root server. The client comes pre-equipped with a list of root servers as well as their IP addresses.

If the client did not have this list of root servers, it would not know where to start looking. The client picks a root server **at random** and starts looking for the domain name.

FILING A REQUEST

Query

```
example.net                IN                A
```

example.net: the domain to look up

IN: the DNS class (can have a bunch of values but is almost always IN which stands for Internet.

A: the record type (we as for type A because we are looking for the address of a top-level domain server.

THE QUERY: CONTACTING A ROOT SERVER

Query

```
example.net          IN      A
```

The first section contains general information about the TLD servers for .net

DNS Client

How long the info remains valid (in secs)

DNS class

Record type (NS = name server, points to DNS server) that is authoritative about hosts in the .net domain.

Here's what I know…

Authority section:

```
net.    172800  IN  NS  m.gtld-servers.net
net.    172800  IN  NS  l.gtld-servers.net
net.    172800  IN  NS  k.gtld-servers.net
net.    172800  IN  NS  j.gtld-servers.net
(list continues)
```

Contains the domain name for each of the name servers… **NEED IP ADDRESSES!**

Root Server

THE QUERY: ROOT SERVER RESPONSE

How long the info remains valid (in secs)

DNS class

Record type (NS = name server, points to DNS server) that is authoritative about hosts in the .net domain.

Here's what I know…

Authority section:

net.	172800	IN	NS	m.gtld-servers.net
net.	172800	IN	NS	l.gtld-servers.net
net.	172800	IN	NS	k.gtld-servers.net
net.	172800	IN	NS	j.gtld-servers.net

(list continues)

Contains the domain name for each of the name servers

Root Server

Additional section:

m.gtld-servers.net	172800	IN	A 192.55.83.30
l.gtld-servers.net	172800	IN	A 192.41.162.30
k.gtld-servers.net	172800	IN	A 192.52.178.30
j.gtld-servers.net	172800	IN	A 192.48.79.30

(list continues)

Tells us where to find the domain names that were previously mentioned… Without them, we would would be stuck in a **loop**. (Would have to look up **m.gtld-servers.net** first but without any IPs for any TLD server for .net, we cannot look up .net domains)

ADDITIONAL SECTION (AKA GLUE RECORDS)

Query

```
example.net        IN      A
```

More specific information about the target website.

The server crunches some numbers and sends back the following list

192.55.83.30

m.gtld-servers.net

THE QUERY: TLD SERVER