DNS 101: Hands-on

Cover Option 1

Dec. 2025

Agenda

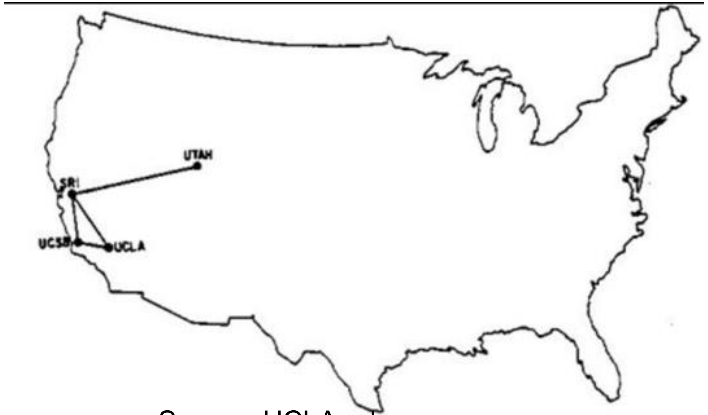
- Once upon a time
- Internet in a nutshell
- DNS architecture and components
- DNS Database and Data
- DNS Resolution process

Once upon a time…

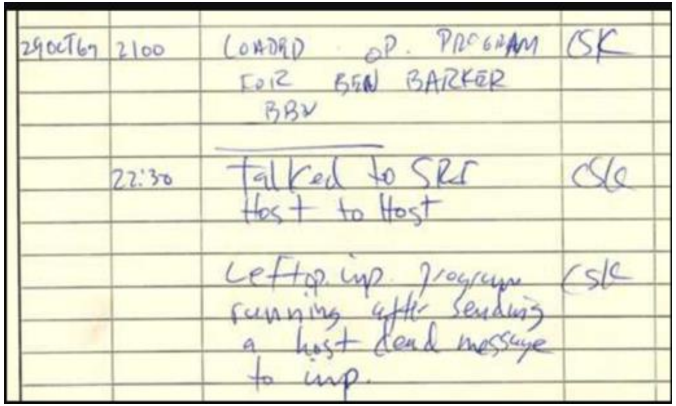
Once upon a time

1969 - ARPANET is Born on October 29th – 4 Participating Institutions:

- UCLA, SRI, University of California Santa Barbara, University of Utah



Source: UCLA.edu



29OCT69	2100	LOADED OP. PROGRAM FOR BEN BARKER BBV	CSK
	22:30	Talked to SRI Host to Host	CSK
		Left op. imp. program running after sending a host dead message to imp.	CSK

Source: edn.com

Internet in a nutshell…

The Internet is millions of devices connected to a network of networks that talk with each other.

Hardware
Software
Protocols

Content
Network
Devices

Naming: DNS/ICANN
Addressing: IP/RIRs
Routing: ISPs

It’s all about Names and Numbers

- Devices are identified over the Internet using IP addresses.
  - IPv4: 192.0.2.7
  - IPv6: 2001:db8::7
- IP addresses are easy for machines to use, humans prefer names.
- In the early days of the Internet, names were simple
  - No domain names yet
  - “Single-label names”, 24 characters maximum
  - Referred to as host names

Name Resolution

- Name resolution is the process of mapping names ↔ IP addresses
- On the early Internet, name resolution used a plain text file: HOSTS.TXT
  - Same function but slightly different format than the former /etc/hosts
  - Centrally maintained by the Network Information Center at the SRI
  - Network administrators sent updates via email
- Ideally everyone had the latest version of the file
  - Released once per week
  - Downloadable via FTP

Problems with HOSTS.TXT

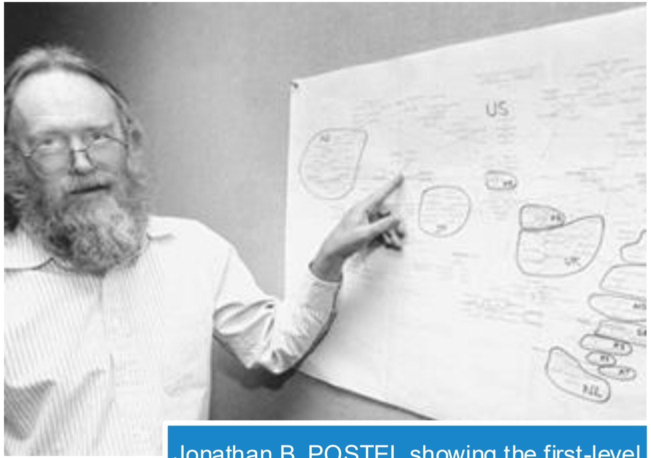
- **Naming contention**
    - Edits made by hand to a text file (no database)
    - No good method to prevent duplicates
- **Synchronization**
    - No one ever had the same version of the file
- **Traffic and load**
    - Significant bandwidth required then just to download the file
- A centrally maintained host file just **didn’t scale.**

DNS to the Rescue

- Discussion started in the early 1980s on a replacement with goals:
  - Address HOST.TXT scaling issues.
  - Simplify email routing.
- Result was the Domain Name System. Requirements in multiple documents:
  - RFC 799, “Internet Name Domains”
  - RFC 819, “The Domain Naming Convention for Internet User Applications”
  - Most referred to: RFC 1034 and RFC 1035

Inventors of the DNS

Paul MOKAPETRIS



Jonathan B. POSTEL showing the first-level domains on a map in 1994

DNS architecture and components

DNS in a nutshell

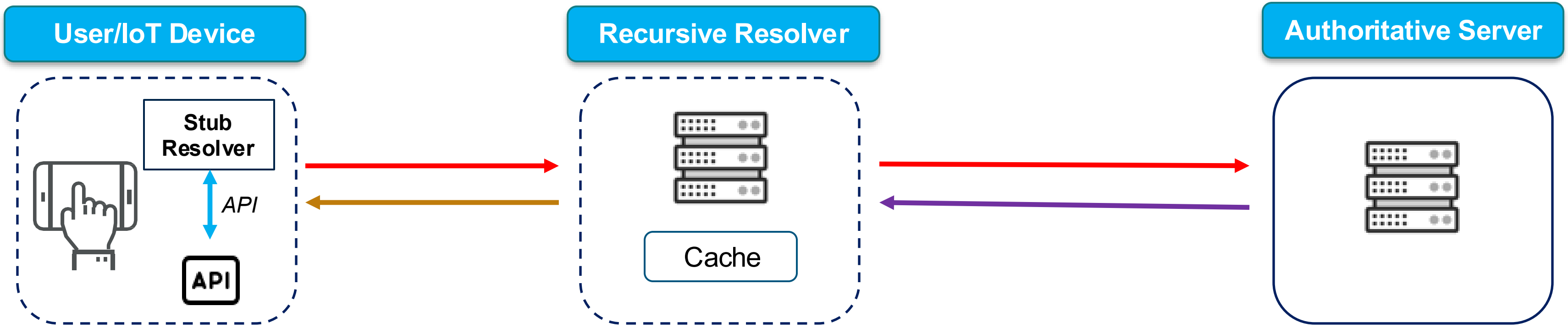
- DNS is a **globally distributed** and **coherent** database.
- A **critical infrastructure** of the Internet, **optimization** and **resilience** are necessary:
    - **caching** to improve performance and
    - Data **replication** to provide redundancy and load distribution.

DNS in a nutshell: components

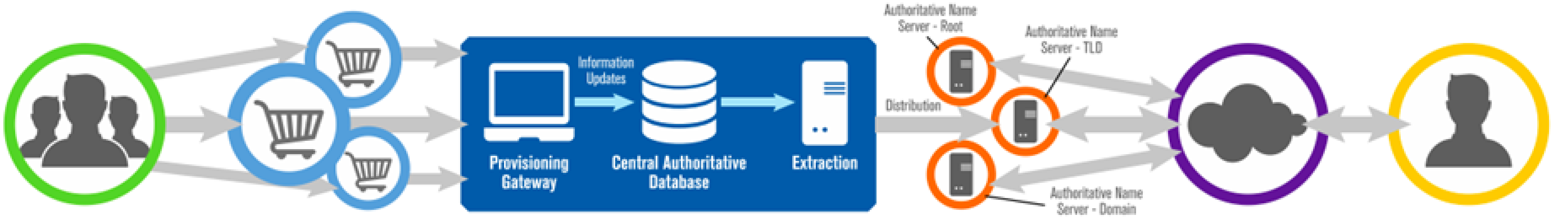
- ”**Name space**”: the **hierarchy** of the nodes and resource records.
- **Recursive resolvers**: run a specific type of service; receive → retrieve → respond to DNS queries.
    - Stub resolver, caches, forwarders.
- **Authoritative name servers**: run a specific type of service to **serve DNS zones** it knows and “**authoritative**” answers to DNS queries.
    - Primary and Secondaries.

DNS in a nutshell: components

- The Stub Resolver [User device]
- The Recursive Resolver [ISP, Public, Corporate]
- The Authoritative [Name/Zone Manager / Operators]



Overview of the DNS Ecosystem


Information Updates
Provisioning Gateway
Central Authoritative Database
Extraction
Distribution
Authoritative Name Server - Root
Authoritative Name Server - TLD
Authoritative Name Server - Domain

Registrants

Customers who register domain names.

Registrars

Organizations accredited by ICANN that process domain name registrations.

TLD Registry

The registry operator generates the authoritative zone for the TLD and distributes to geographically dispersed name servers.

Name Servers

Name servers respond to queries with the appropriate DNS record for the query.

DNS Queries

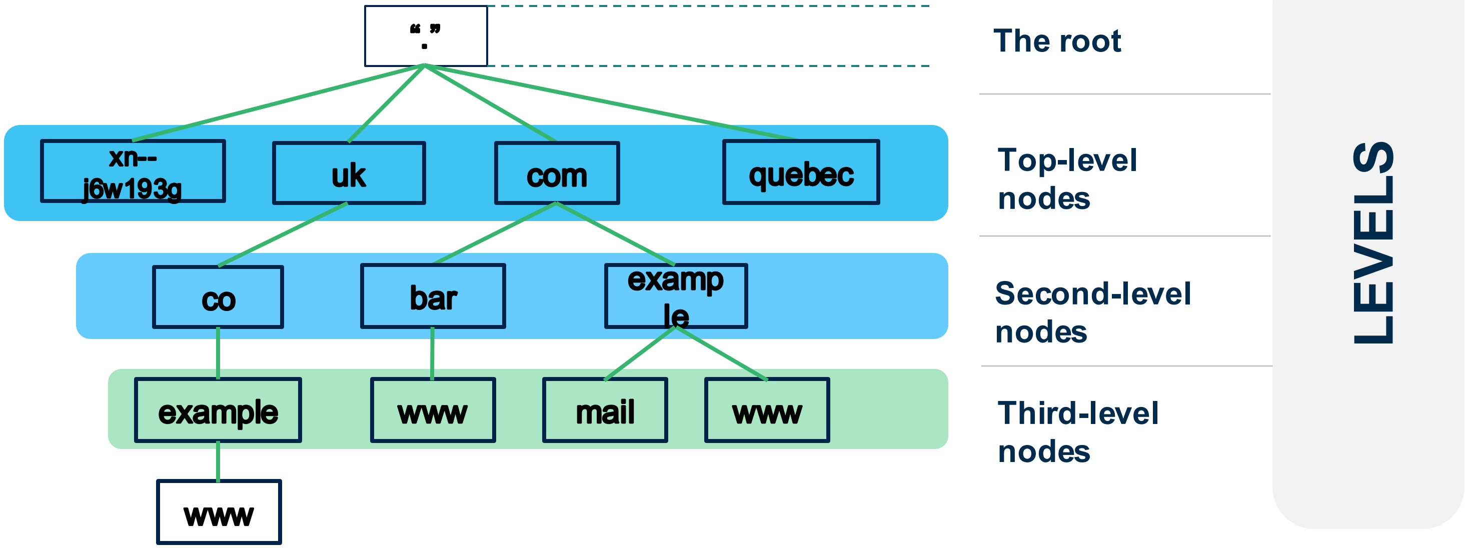
Recursive name servers send the queries to the authoritative name servers for the TLD.

User

The user types a domain name, www.example.com, into a browser. The browser sends the query to recursive name servers in the cloud.

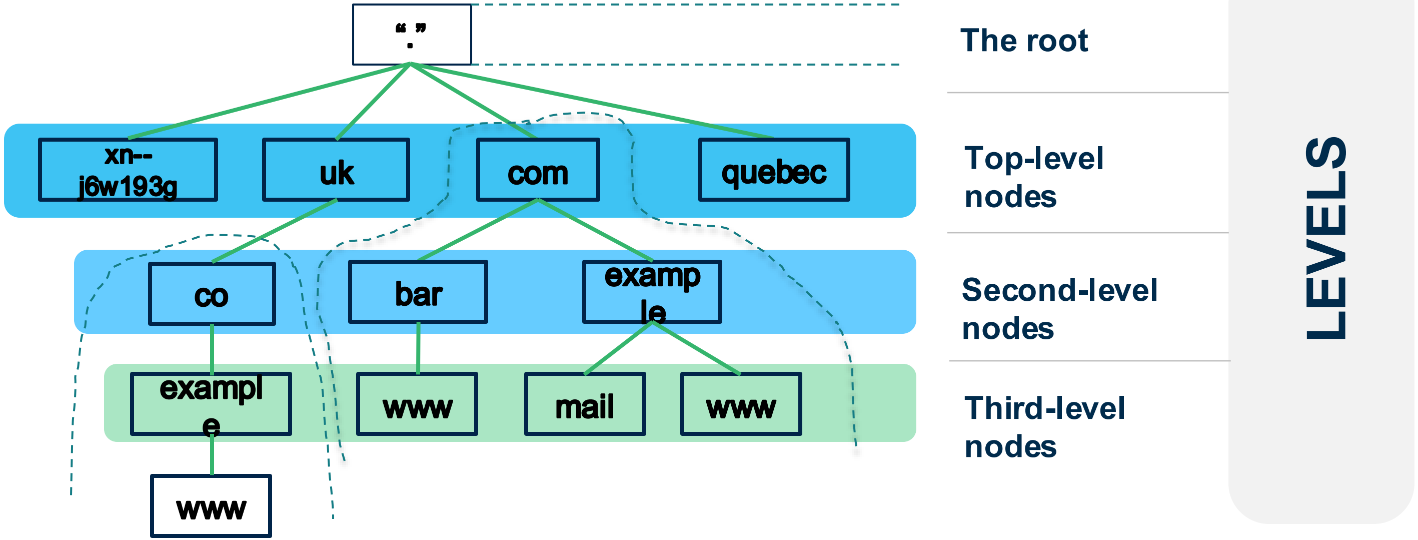
The Name Space

- DNS database structure is an inverted tree called the **name space**
- Each node has a **label**
- The root node (and only the root node) has a **null label**
- The name space is divided up to allow **distributed administration**

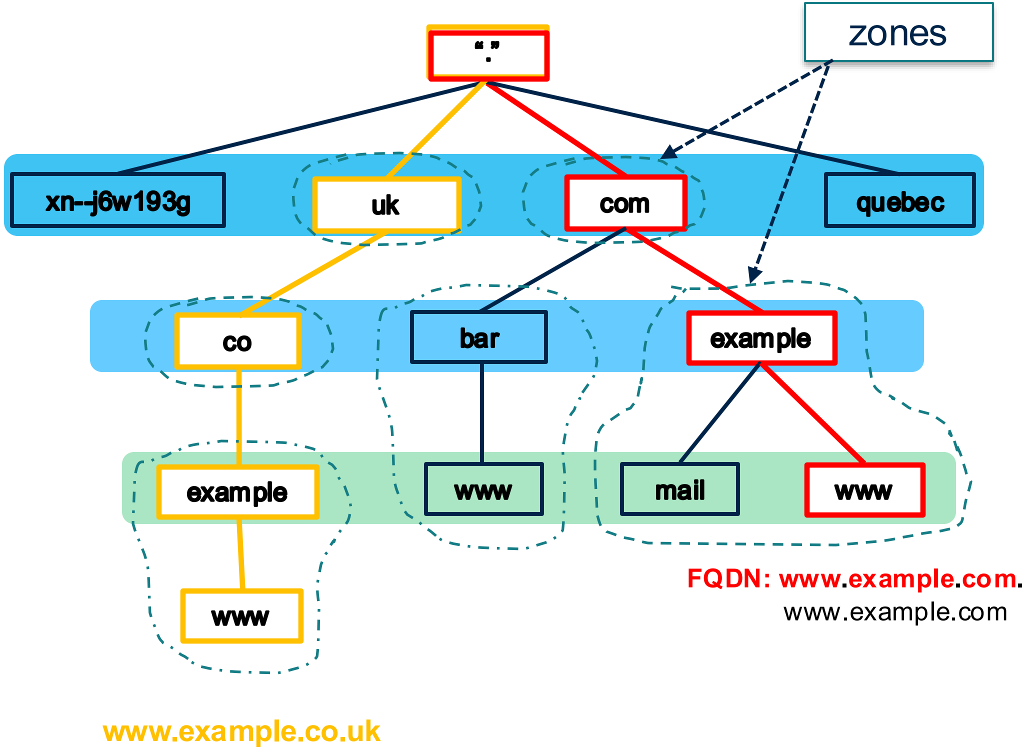


The Name Space

- Allowed characters for labels are “**LDH**” (letters, digits, hyphen)
- Maximum length **63 characters**
- A **domain** is a node and everything below it
- The top node of a domain is the **apex** of that domain



The Name Space: Zones and Administrative boundaries



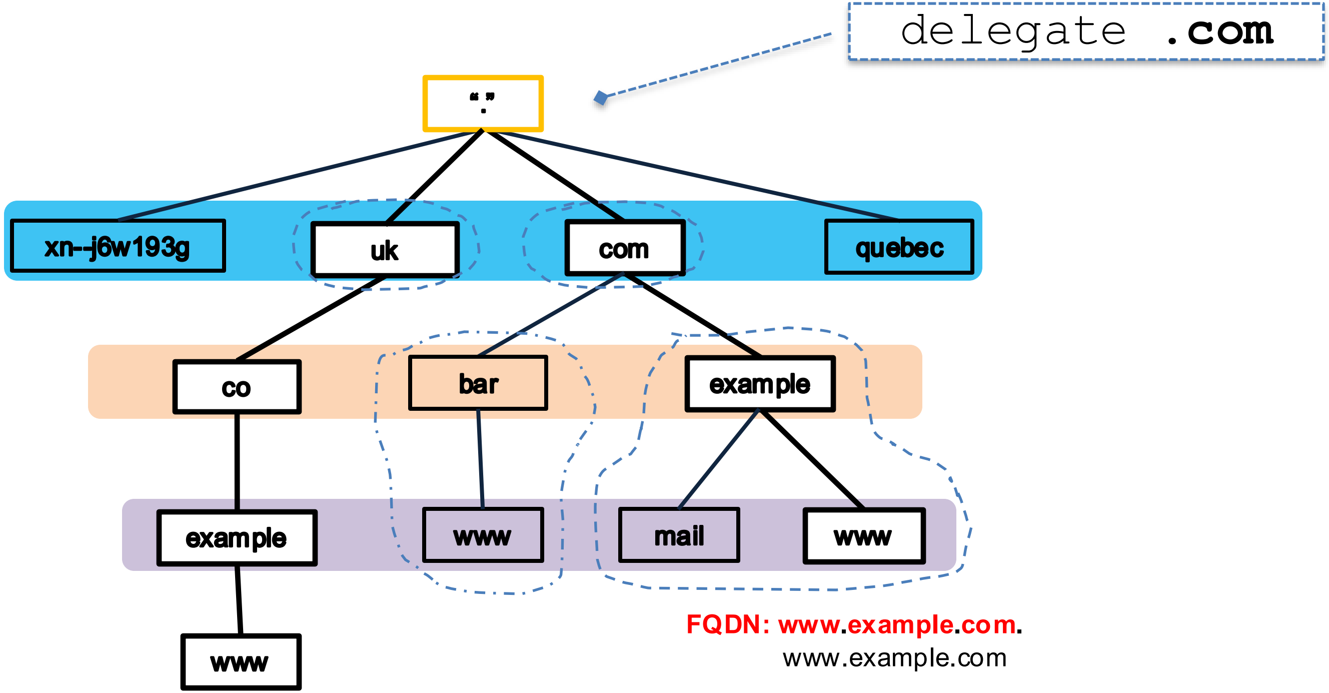
FQDN: www.example.com.

www.example.com

www.example.co.uk

- Delegations create **zones**
- **Administrative divisions** are called zones.
- A zone's administrator may delegate the administration of a subtree of its zone, thus creating a new zone.
- Names are delegated from top to down (starting at the root).
    - Delegating zone is the **parent**
    - Created zone is the **child**
- The full domain names are obtained by concatenating the labels (names) from the last level to the top.

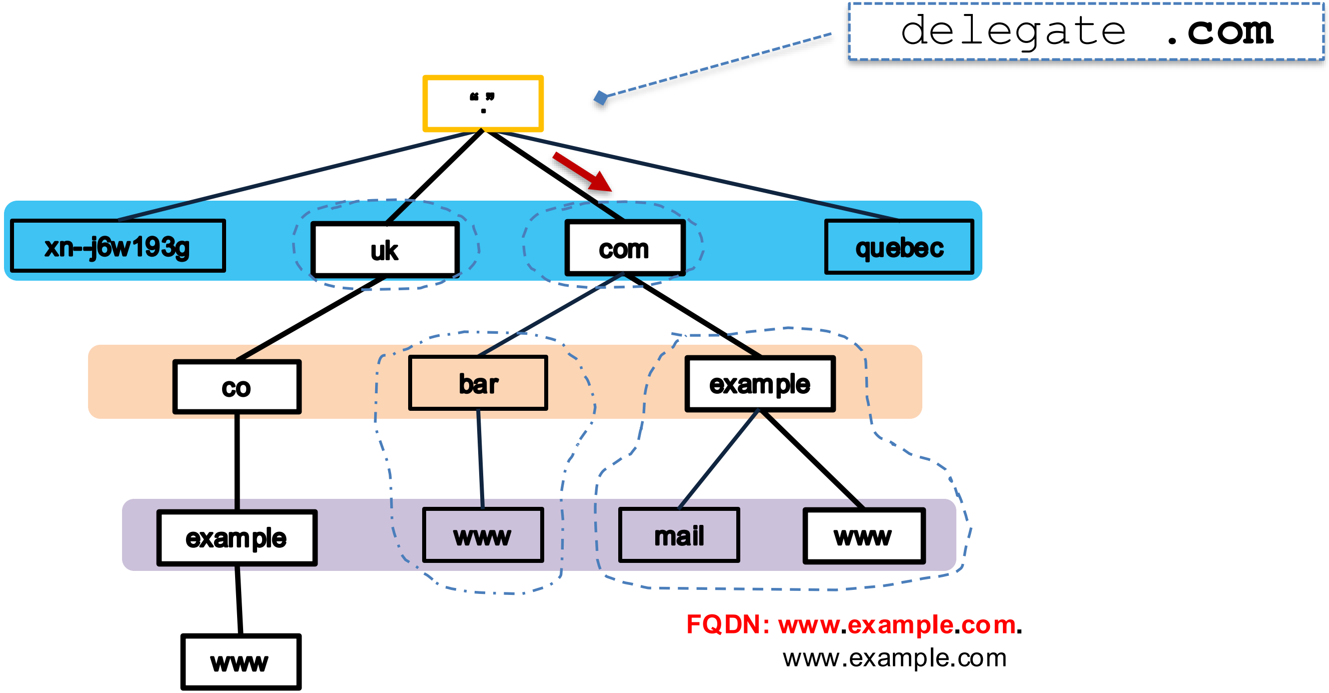
Delegation: How it works



FQDN: www.example.com.

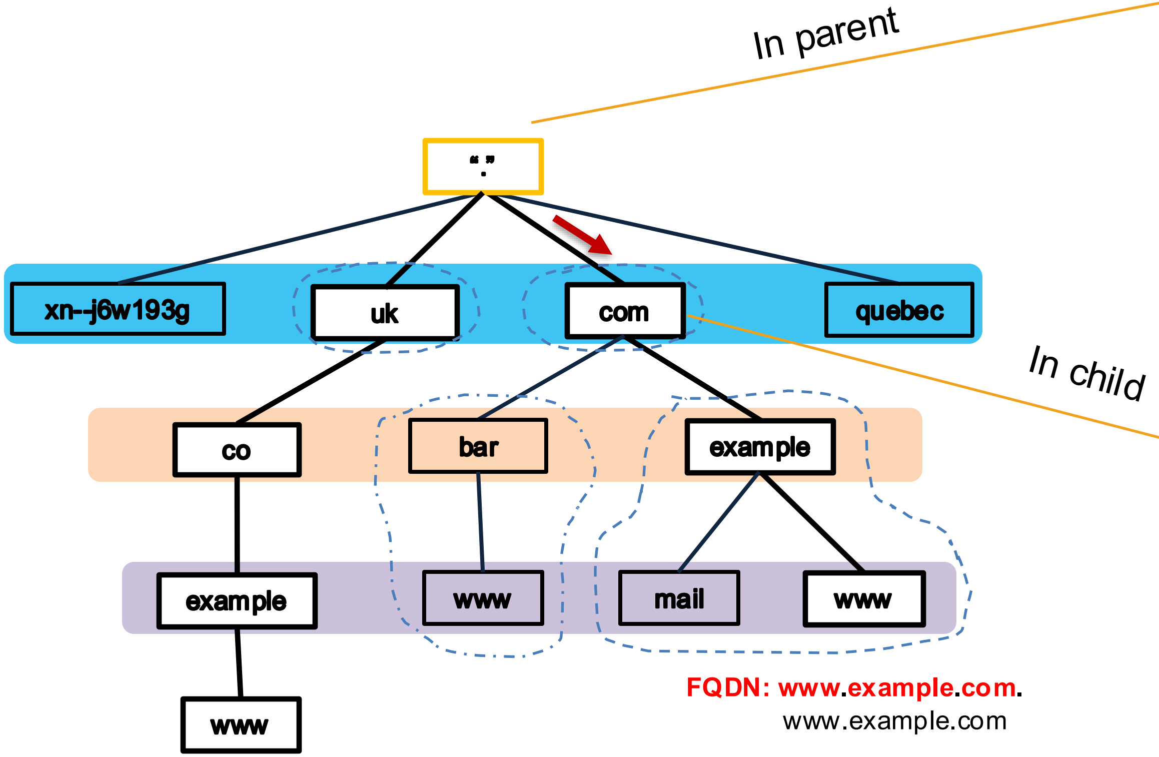
www.example.com

Delegation: How it works



FQDN: www.example.com.
www.example.com

Delegation: How it works

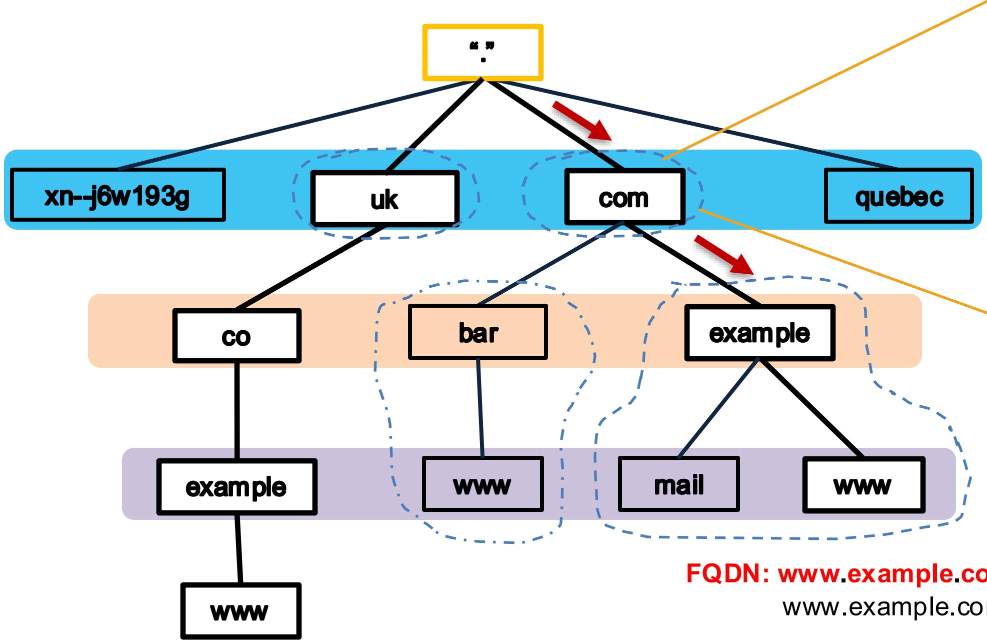


FQDN: www.example.com.
www.example.com

```
com.   NS   a.gtld-servers.net.
com.   NS   b.gtld-servers.net.
com.   NS   c.gtld-servers.net.
com.   NS   d.gtld-servers.net.
com.   NS   e.gtld-servers.net.
com.   NS   f.gtld-servers.net.
com.   NS   g.gtld-servers.net.
com.   NS   h.gtld-servers.net.
com.   NS   i.gtld-servers.net.
com.   NS   j.gtld-servers.net.
com.   NS   k.gtld-servers.net.
com.   NS   l.gtld-servers.net.
com.   NS   m.gtld-servers.net.
```

```
com.   NS   a.gtld-servers.net.
com.   NS   b.gtld-servers.net.
com.   NS   c.gtld-servers.net.
com.   NS   d.gtld-servers.net.
com.   NS   e.gtld-servers.net.
com.   NS   f.gtld-servers.net.
com.   NS   g.gtld-servers.net.
com.   NS   h.gtld-servers.net.
com.   NS   i.gtld-servers.net.
com.   NS   j.gtld-servers.net.
com.   NS   k.gtld-servers.net.
com.   NS   l.gtld-servers.net.
com.   NS   m.gtld-servers.net.
```

Delegation: How it works



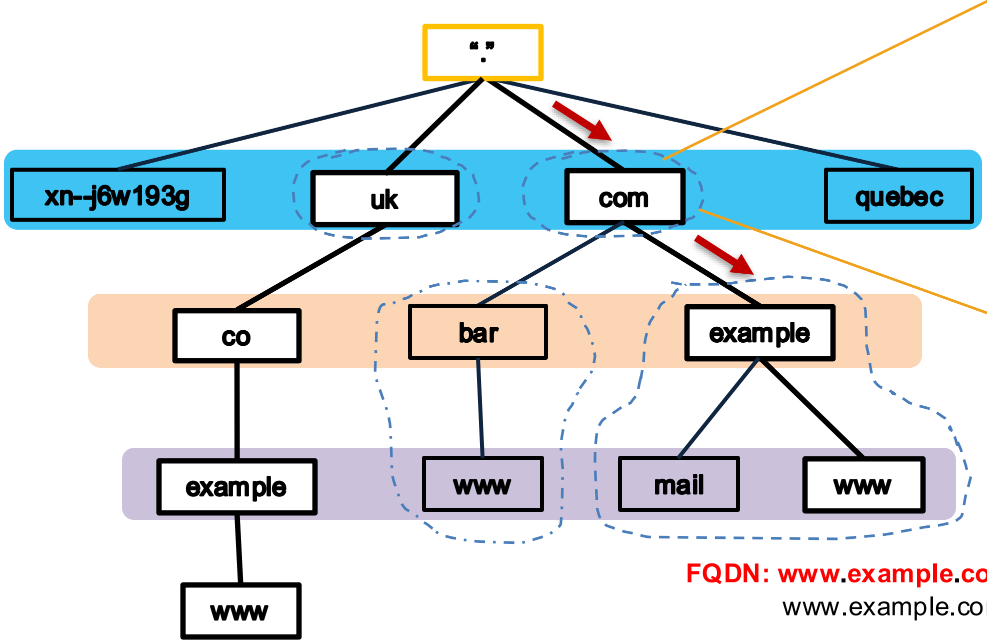
```
com.    NS    a.gtld-servers.net.
com.    NS    b.gtld-servers.net.
com.    NS    c.gtld-servers.net.
com.    NS    d.gtld-servers.net.
com.    NS    e.gtld-servers.net.
com.    NS    f.gtld-servers.net.
com.    NS    g.gtld-servers.net.
com.    NS    h.gtld-servers.net.
com.    NS    i.gtld-servers.net.
com.    NS    j.gtld-servers.net.
com.    NS    k.gtld-servers.net.
com.    NS    l.gtld-servers.net.
com.    NS    m.gtld-servers.net.
```

```
example.com.    NS    ns1.example.com.
example.com.    NS    ns2.example.com.
example.com.    NS    a.gtld-servers.net.
example.com.    NS    b.gtld-servers.net.
```

FQDN: www.example.com.

www.example.com

Delegation: How it works



```
com.    NS    a.gtld-servers.net.
com.    NS    b.gtld-servers.net.
com.    NS    c.gtld-servers.net.
com.    NS    d.gtld-servers.net.
com.    NS    e.gtld-servers.net.
com.    NS    f.gtld-servers.net.
com.    NS    g.gtld-servers.net.
com.    NS    h.gtld-servers.net.
com.    NS    i.gtld-servers.net.
com.    NS    j.gtld-servers.net.
com.    NS    k.gtld-servers.net.
com.    NS    l.gtld-servers.net.
com.    NS    m.gtld-servers.net.
```

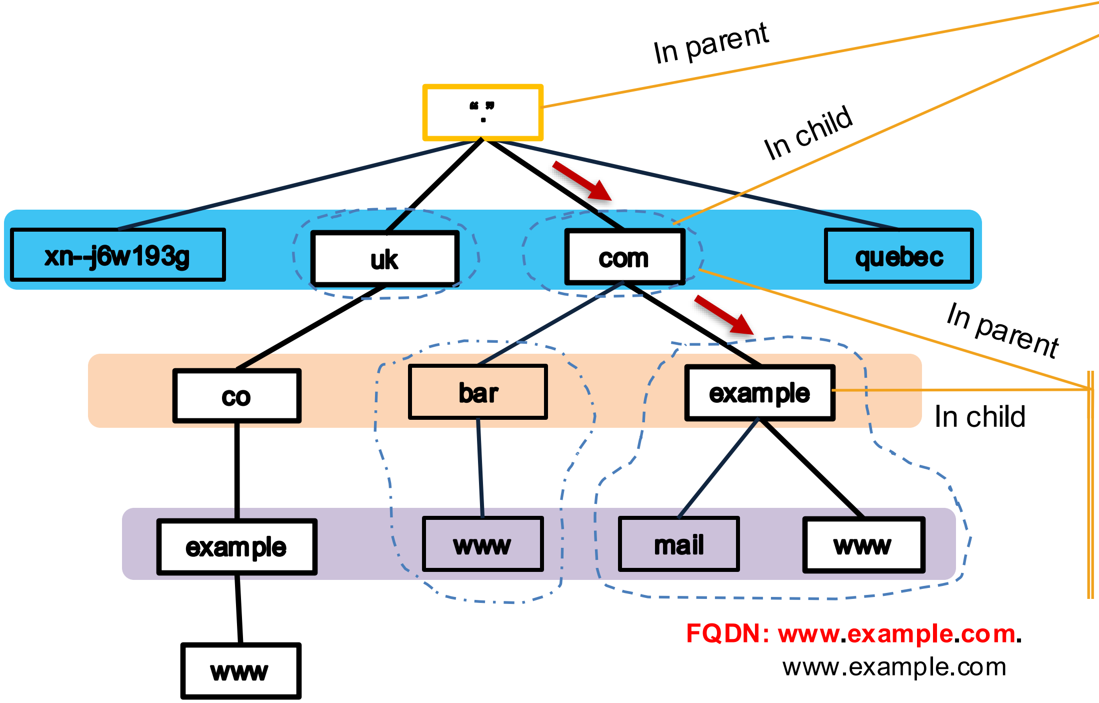
```
example.com.    NS    ns1.example.com.
example.com.    NS    ns2.example.com.
example.com.    NS    a.gtld-servers.net.
example.com.    NS    b.gtld-servers.net.

ns1.example.com.    A     198.51.100.1
ns1.example.com.    A.    203.0.113.150
```

Glue Records

FQDN: www.example.com.

www.example.com

Delegation: How it works



FQDN: www.example.com.
www.example.com

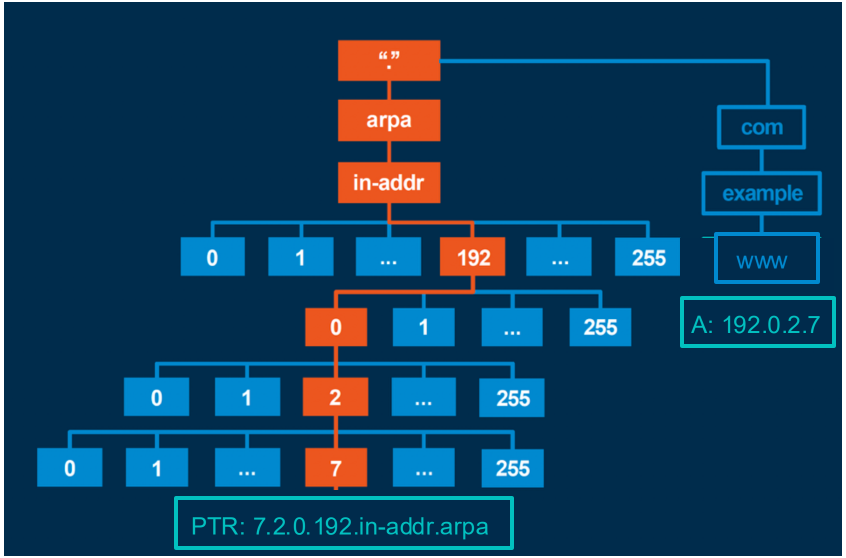
```
com.    NS    a.gtld-servers.net.
com.    NS    b.gtld-servers.net.
com.    NS    c.gtld-servers.net.
com.    NS    d.gtld-servers.net.
com.    NS    e.gtld-servers.net.
com.    NS    f.gtld-servers.net.
com.    NS    g.gtld-servers.net.
com.    NS    h.gtld-servers.net.
com.    NS    i.gtld-servers.net.
com.    NS    j.gtld-servers.net.
com.    NS    k.gtld-servers.net.
com.    NS    l.gtld-servers.net.
com.    NS    m.gtld-servers.net.
```

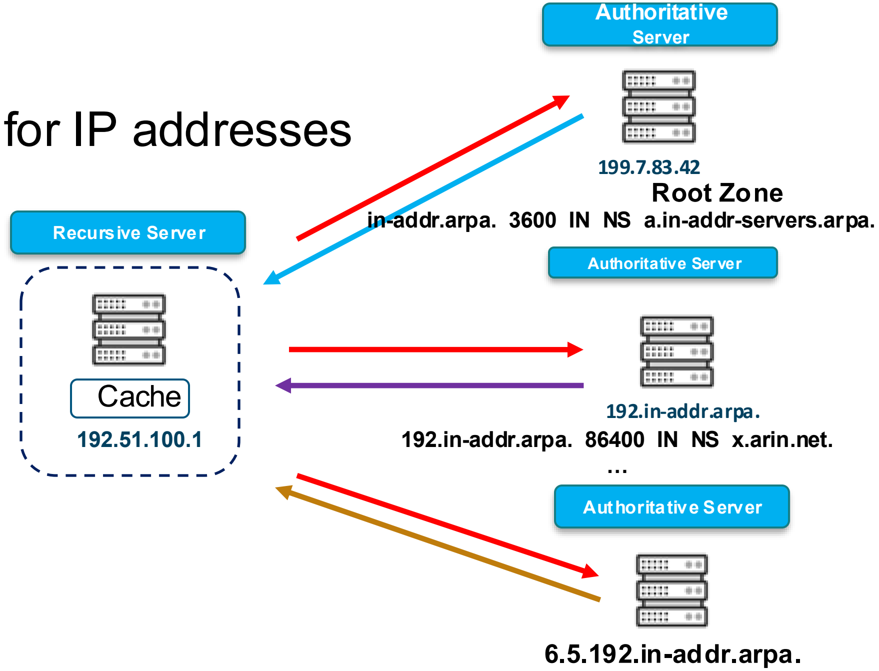
```
example.com.    NS    ns1.example.com.
example.com.    NS    ns2.example.com.
example.com.    NS    a.gtld-servers.net.
example.com.    NS    b.gtld-servers.net.
ns1.example.com.    A     198.51.100.1
ns1.example.com.    A.    203.0.113.150
```

Glue Records

The RIRs & The DNS: reverse DNS

- Assigned IP Addressed to DNS severs
- Run reverse DNS (authoritative NSes) for IP addresses





```
6.5.192.in-addr.arpa.  86400    IN  NS    a3.verisigndns.com.
6.5.192.in-addr.arpa.  86400    IN  NS    a1.verisigndns.com.
6.5.192.in-addr.arpa.  86400    IN  NS    a2.verisigndns.com.
```

DNS Database and Data

DNS Data

- The DNS standard specifies the format of DNS data sent over the network
    - Informally called “wire format”
- The standard also specifies a text-based representation for DNS data called **master file format**, used for storing the data (much like tables in a database)
- A zone file contains all the data for a zone in master file format.

Zone File & DNS Resource Records

- Different kinds of data are associated with a domain name.
- These data are stored as **resource records** (RR) in a DNS database.
- Different types of records (**resource record type**) for different types of data.
- A **zone file** contains all the data for a particular zone (in a specific format).
- At least one zone file associated to each zone.
- Resource records from multiple zones are never mixed in the same file.

Format of Resource Records

- Resource records have **five fields out of which two are mandatory**:
  - **Owner**: Domain name the resource record is associated with
  - **Time to live** (TTL): Time (in seconds) the record can be cached.
  - **Class**: A mechanism for extensibility that is largely unused
  - **Type**: The type of data the record stores
  - **RDATA**: The data (of the type specified) that the record carries
- Resource record syntax in master file format:
  - [owner] [TTL] [class] <type> <RDATA>

Zone File & DNS Resource Records

- Different record types for different type of data in a zone fine. Examples:

A	IPv4 address
AAAA	IPv6 address
NS	Name of an authoritative name server
SOA	“Start of authority”, appears at zone apex
MX	Name of a “mail exchange server”

- Full list at : http://www.iana.org/assignments/dns-parameters/dns-parameters.xhtml#dns-parameters-4

Zone File & DNS Resource Records

As the Internet evolves, new resource record types appear. A few examples:

- **TLSA:** associates a TLS server certificate with the domain name where the record is found.
- **SMIMEA:** associates an end entity certificate with the associated email address.
- **SVCB/HTTPS:** instructions for access to a service with improved performance from DNS – RFC 9460
- **CAA:** CAs authorized to issue certificates for a domain name.

Address Records (A & AAAA)

- Most common use of DNS is mapping domain names to IP addresses.
- Two most common types of resource records are:
  - ○ Address (A) record stores mapping for a domain name to an IPv4 address

```
example.com.        A      192.0.2.7
```

  - ○ “Quad A” (AAAA) record stores mapping for a domain name to an IPv6 address

```
example.com.        AAAA      2001:db8::7
```

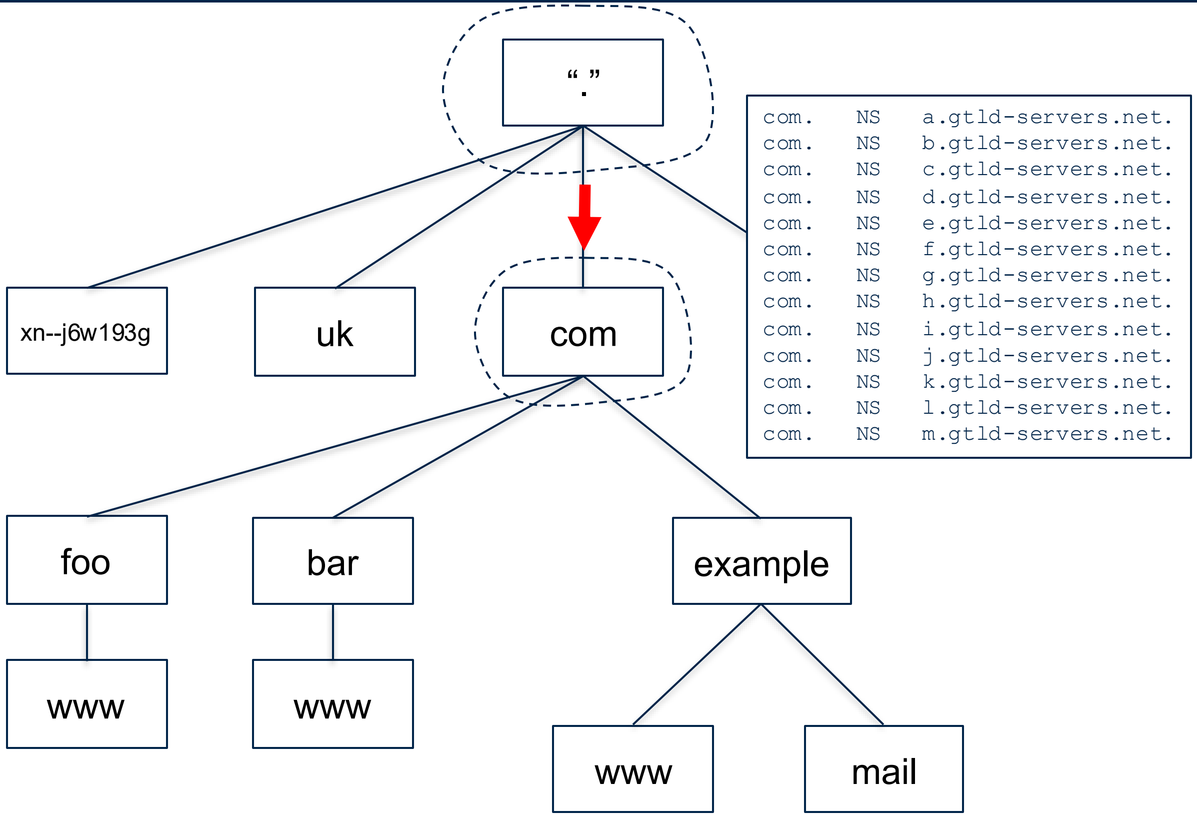
Authoritative Name Server (NS)

- Specifies an **authoritative name server** for a zone: servers that are expected to provide answers with “**authority**” about a domain.
- The only record type to appear in two places: at “*parent*” and “*child*” zones.

```
example.com.   NS   ns1.example.com.
example.com.   NS   ns2.example.com.
```

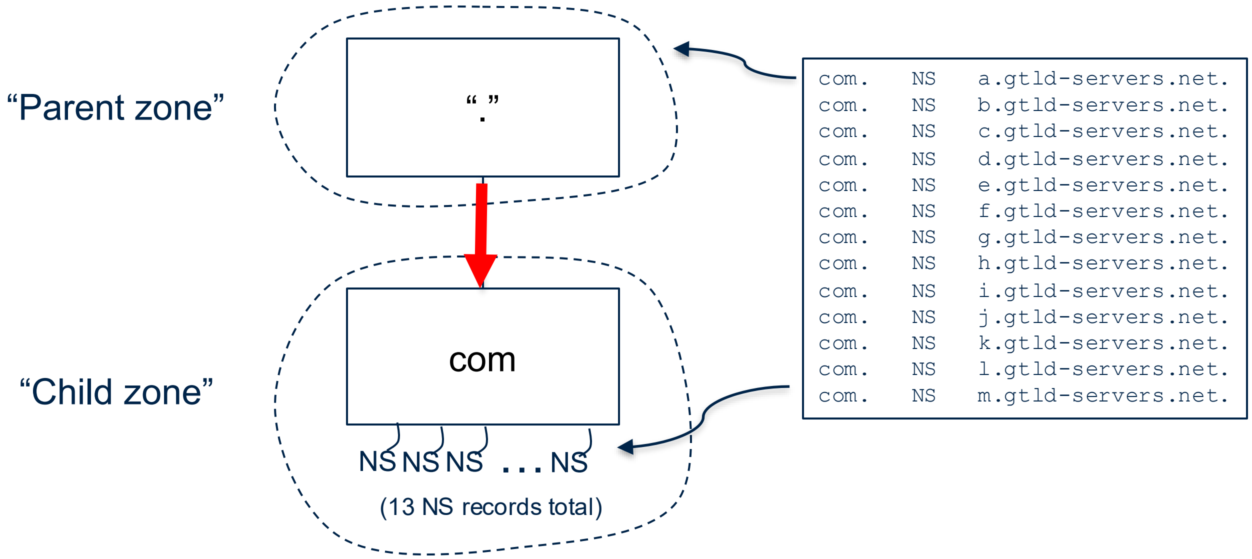
- Left hand side is the name of a zone (Owner).
- Right hand side is the **name** of an authoritative name server for that zone.
    - **Not an IP address**!

NS Records Mark Delegations



```
com.   NS   a.gtld-servers.net.
com.   NS   b.gtld-servers.net.
com.   NS   c.gtld-servers.net.
com.   NS   d.gtld-servers.net.
com.   NS   e.gtld-servers.net.
com.   NS   f.gtld-servers.net.
com.   NS   g.gtld-servers.net.
com.   NS   h.gtld-servers.net.
com.   NS   i.gtld-servers.net.
com.   NS   j.gtld-servers.net.
com.   NS   k.gtld-servers.net.
com.   NS   l.gtld-servers.net.
com.   NS   m.gtld-servers.net.
```

NS Records Appear in Two Places



“Parent zone”

“ . ”

“Child zone”

com

NS NS NS . . . NS

(13 NS records total)

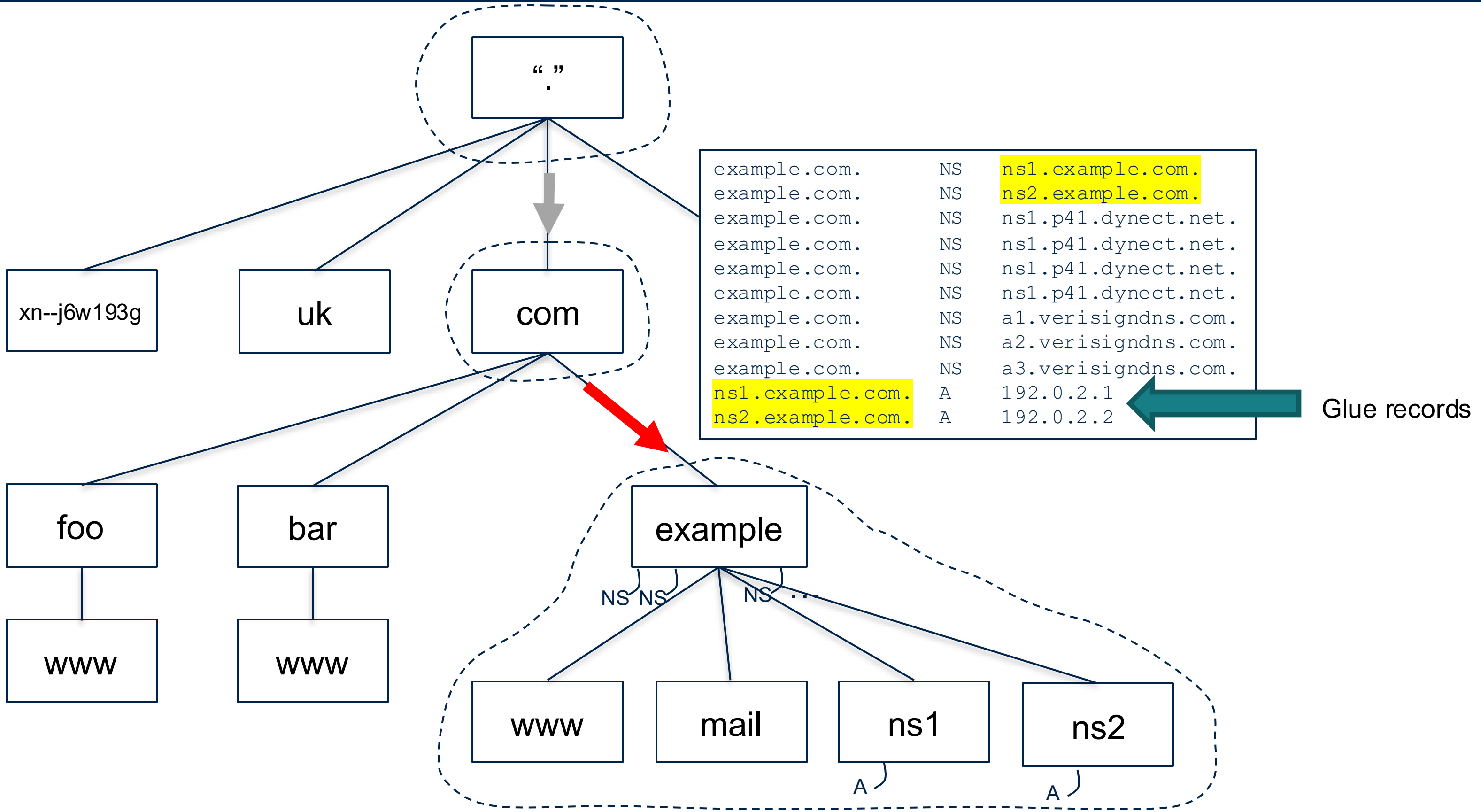
```
com.    NS    a.gtld-servers.net.
com.    NS    b.gtld-servers.net.
com.    NS    c.gtld-servers.net.
com.    NS    d.gtld-servers.net.
com.    NS    e.gtld-servers.net.
com.    NS    f.gtld-servers.net.
com.    NS    g.gtld-servers.net.
com.    NS    h.gtld-servers.net.
com.    NS    i.gtld-servers.net.
com.    NS    j.gtld-servers.net.
com.    NS    k.gtld-servers.net.
com.    NS    l.gtld-servers.net.
com.    NS    m.gtld-servers.net.
```

Glue Records

- A glue record is:
  - An **A** or **AAAA** record.
  - Included in the parent zone as part of the delegation information.
- Glue is needed to **break a circular dependency** when the name of the name server ends in the name of the zone being delegated.

example.com. NS ns1.**example.com.**

More Delegation: Including Glue



```
example.com.        NS    ns1.example.com.
example.com.        NS    ns2.example.com.
example.com.        NS    ns1.p41.dynect.net.
example.com.        NS    ns1.p41.dynect.net.
example.com.        NS    ns1.p41.dynect.net.
example.com.        NS    ns1.p41.dynect.net.
example.com.        NS    a1.verisigndns.com.
example.com.        NS    a2.verisigndns.com.
example.com.        NS    a3.verisigndns.com.
ns1.example.com.    A     192.0.2.1
ns2.example.com.    A     192.0.2.2
```

Glue records

Time for practice : getting familiar with dig

- Use ”dig” tool to:
  - Retrieve the NS records for your ccTLD
  - Retrieve the IP address of those NS
  - Retrieve the IP address records for your organization web site and mail servers.
- Are the TTL for the above records the same ? (use the “aa” flag to answer).
- Check if delegation information for your ccTLD in the root zone matches with the zone’s data.

Start of Authority (SOA)

- Contains **administrative information** about the zone.
- Every domain must have a Start of Authority record at the cutover point where the domain is delegated from its parent domain.
- SOA indicates that a name server is authoritative for a domain. If we do not receive a SOA RR in a query response from a server, that indicates the server is not authoritative for that domain.
- DNS name servers are normally set up in clusters (primary and secondaries). The database for each cluster is synchronized through zone transfers. The data in a SOA record for a zone is **used to control the zone transfer**.

Start of Authority (SOA) record fields

- **Mname**: primary name server for the domain, or the first name server in the name server list.

- **Rname**: mailbox of the responsible party for the domain. For mailbox john.doe@example.com this field will be john\.doe.example.com.

- **Serial**: version number of the original copy of a zone (preserved in zone transfers). If a secondary name server observes an **increase** in this number, it assumes that the zone has been updated and it will initiate a **zone transfer**.

Start of Authority (SOA) record fields

- **Refresh**: duration (seconds) before a secondary NS checks for zone updates
- **Retry**: duration (seconds) before a failed refresh should be retried.
    - Normally set to less than refresh.
- **Expire**: upper limit (in seconds) before a secondary NS should stop answering requests for the zone if the primary NS does not respond.
- **Minimum**: TTL for negative caching purposes.

Start of Authority (SOA) record fields

icann.org. 1909 IN **SOA** **sns.dns.icann.org**. noc.dns.icann.org. 2025121120 10800 3600 1209600 3600

Mname: primary name server for the domain, or the first name server in the name server list.

Start of Authority (SOA) record fields

icann.org. 1909 IN **SOA** sns.dns.icann.org. **noc.dns.icann.org**. 2025121120 10800 3600 1209600 3600

Rname: mailbox of the responsible party for the domain. For mailbox john.doe@example.com this field will be john\.doe.example.com.

Start of Authority (SOA) record fields

icann.org. 1909 IN **SOA** sns.dns.icann.org. noc.dns.icann.org. **2025121120** 10800 3600 1209600 3600

Serial: version number of the original copy of a zone (preserved in zone transfers). If a secondary name server observes an **increase** in this number, it assumes that the zone has been updated and it will initiate a **zone transfer**.

Start of Authority (SOA) record fields

icann.org. 1909 IN **SOA** sns.dns.icann.org. noc.dns.icann.org. 2025121120 **10800** 3600 1209600 3600

Refresh: duration (seconds) before a secondary NS checks for zone updates

Start of Authority (SOA) record fields

icann.org. 1909 IN **SOA** sns.dns.icann.org. noc.dns.icann.org. 2025121120 10800 **3600** 1209600 3600

Retry: duration (seconds) before a failed refresh should be retried.

- Normally set to less than refresh.

Start of Authority (SOA) record fields

icann.org. 1909 IN **SOA** sns.dns.icann.org. noc.dns.icann.org. 2025121120 10800 3600 **1209600** 3600

Expire: upper limit (in seconds) before a secondary NS should stop answering requests for the zone if the primary NS does not respond.

Start of Authority (SOA) record fields

icann.org. 1909 IN **SOA** sns.dns.icann.org. noc.dns.icann.org. 2025121120 10800 3600 1209600 **3600**

Minimum: TTL for negative caching purposes.

Time for practice: SOA record

1. Look for the SOA record for:
   - your ccTLD
   - your organization’s domain name.
2. Comment on the respective values of the various fields.

PTR record in reverse DNS

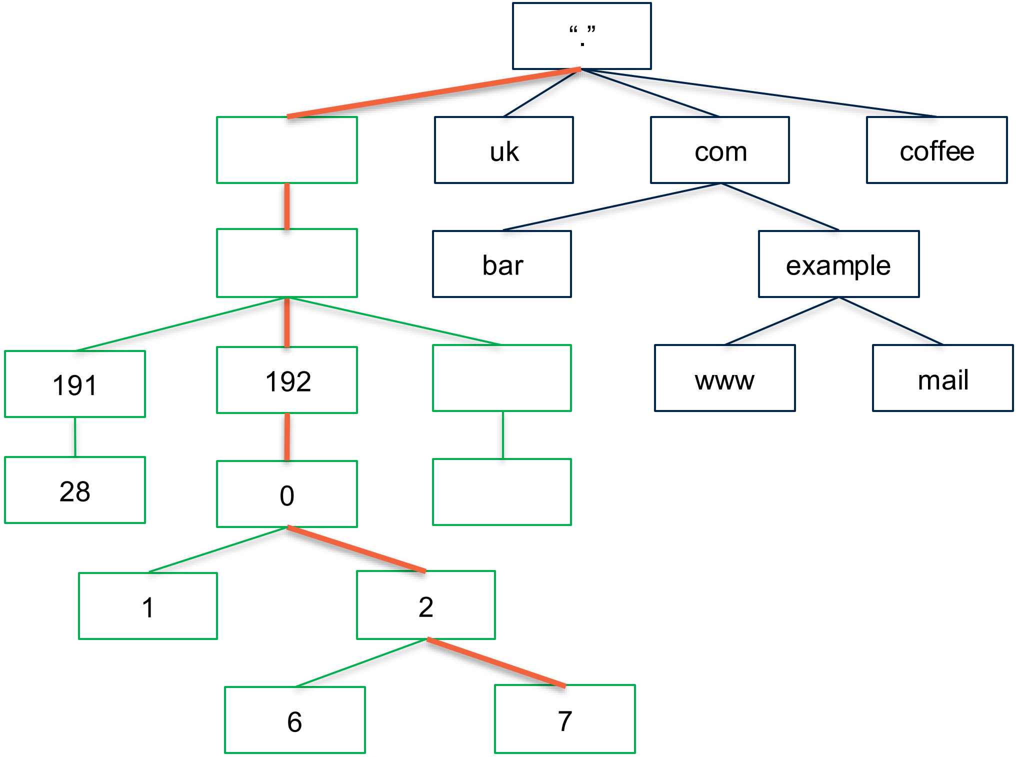
- The most common use of DNS is mapping domain names to IP addresses.
- DNS also maps IP addresses to domain names. This is called reverse DNS and it uses the **PTR RR type**.
- IPv4 reverse DNS is mapped via a special domain called **in-addr.arpa.**
- IPv6 reverse DNS is mapped via a special domain called **ip6.arpa.**
- To represent the IPv4 address 192.0.2.7 of example.com, reverse the IPv4 address and append the suffix in-addr.arpa. at the end, resulting in:

  7.2.0.192.in-addr.arpa.

Reverse DNS entries (PTR)

- Subtree for 192.0.2.7:

7.2.0.192.in-addr.arpa.



Time for practice !

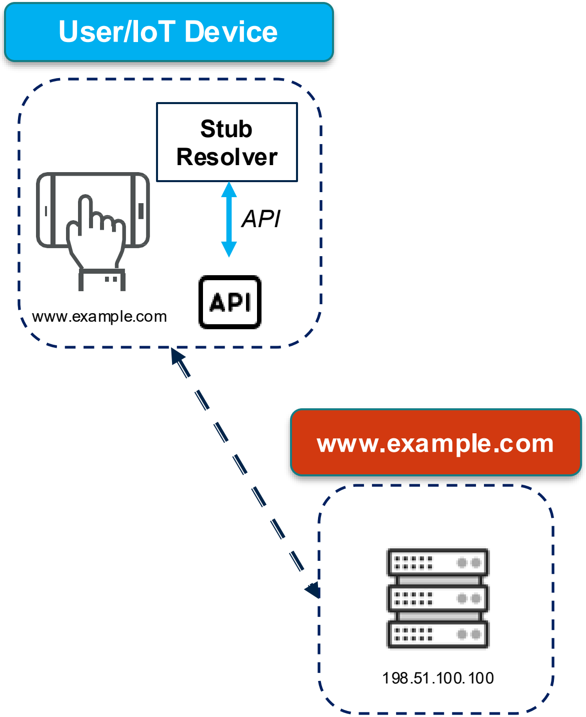
1. Identify if reverse DNS entries exist for your organization’s:
   - email servers
   - Web server
   - Authoritative nameservers.
2. If reverse DNS entry does not exist, how to create ?

Sample Zone File: example.com

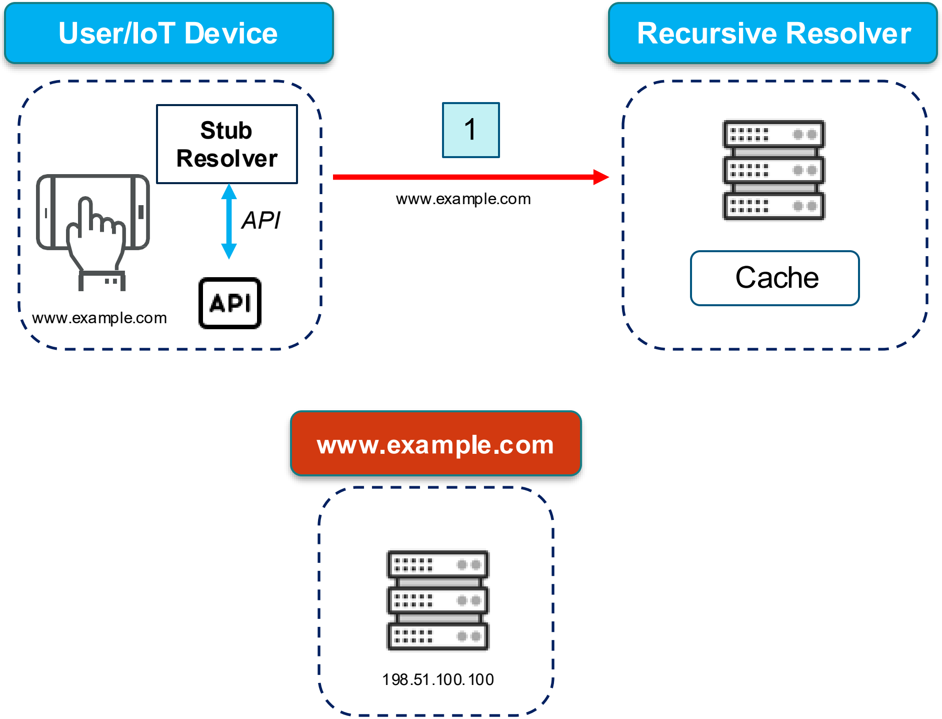
```
example.com.      SOA   ns1.example.com. hostmaster.example.com. (
                         20200316155500 ; serial
                         86400          ; refresh (1 hour)
                         7200           ; retry (2 hour)
                         2592000        ; expire (4 weeks 2 days)
                         172800 )       ; minimum (2 days)
example.com.      NS    ns1.example.com.
example.com.      NS    ns2.example.com.
example.com.      NS    ns1.p41.dynect.net.
example.com.      NS    ns1.p41.dynect.net.
example.com.      NS    ns1.p41.dynect.net.
example.com.      NS    ns1.p41.dynect.net.
example.com.      NS    a1.verisigndns.com.
example.com.      NS    a2.verisigndns.com.
example.com.      NS    a3.verisigndns.com.
example.com.      A     192.0.2.7
example.com.      AAAA  2001:db8::7
example.com.      MX    10 mail.example.com.
example.com.      MX    20 mail-backup.example.com.
www.example.com.  CNAME example.com.
ns1.example.com.  A     192.0.2.1
ns2.example.com.  A     192.0.2.2
```

DNS Resolution process

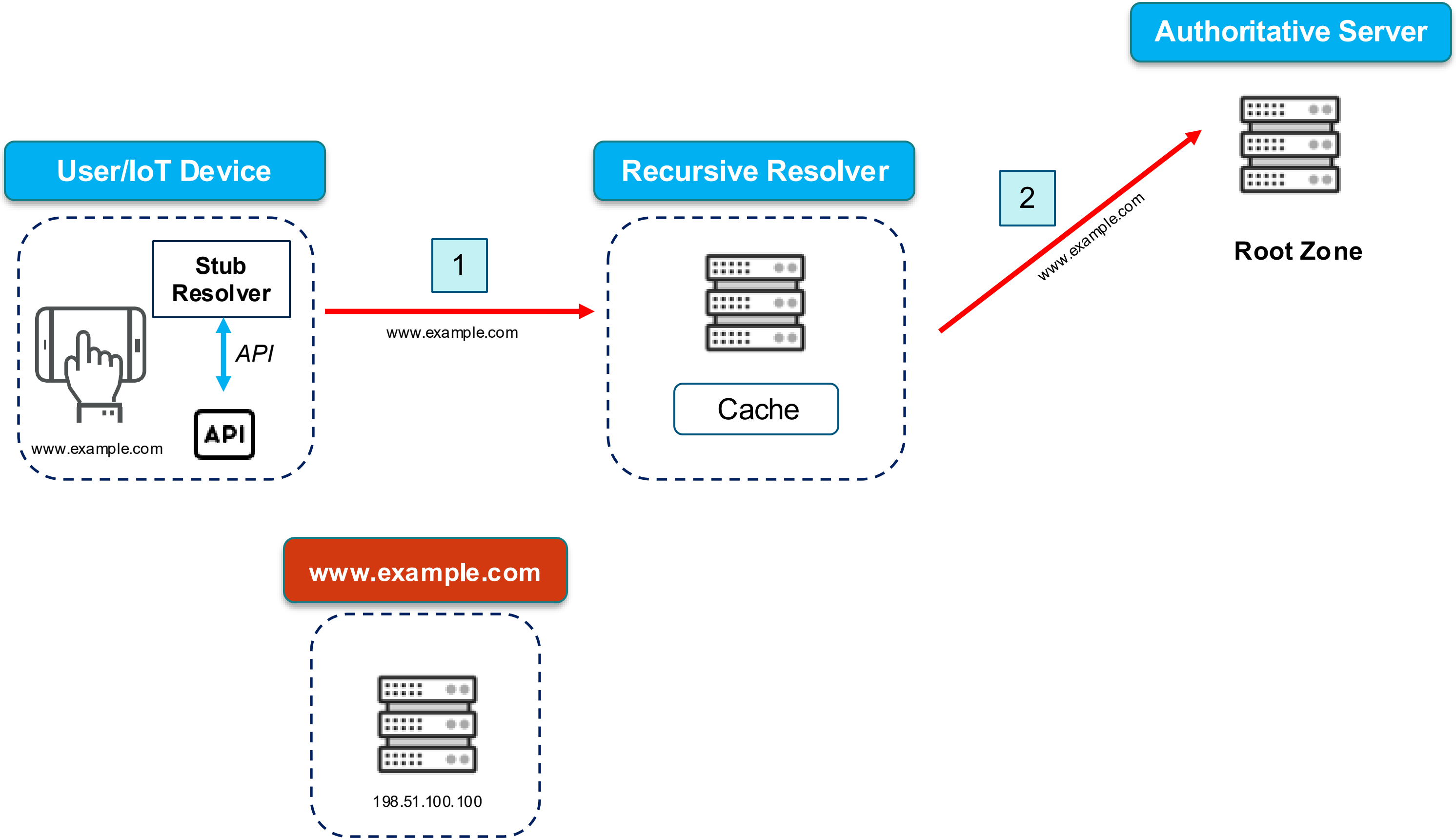
DNS Resolution process



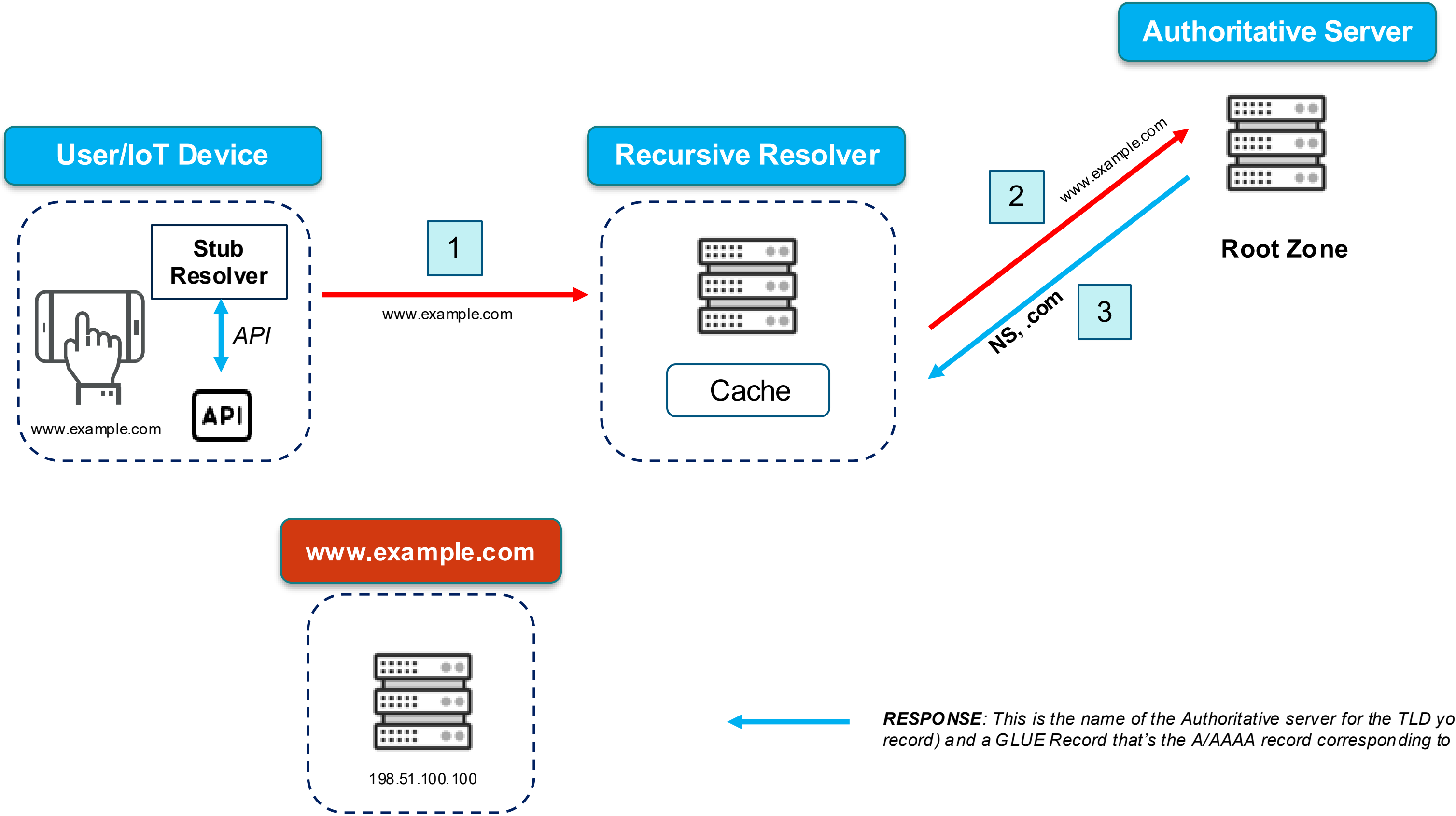
DNS Resolution process



DNS Resolution process

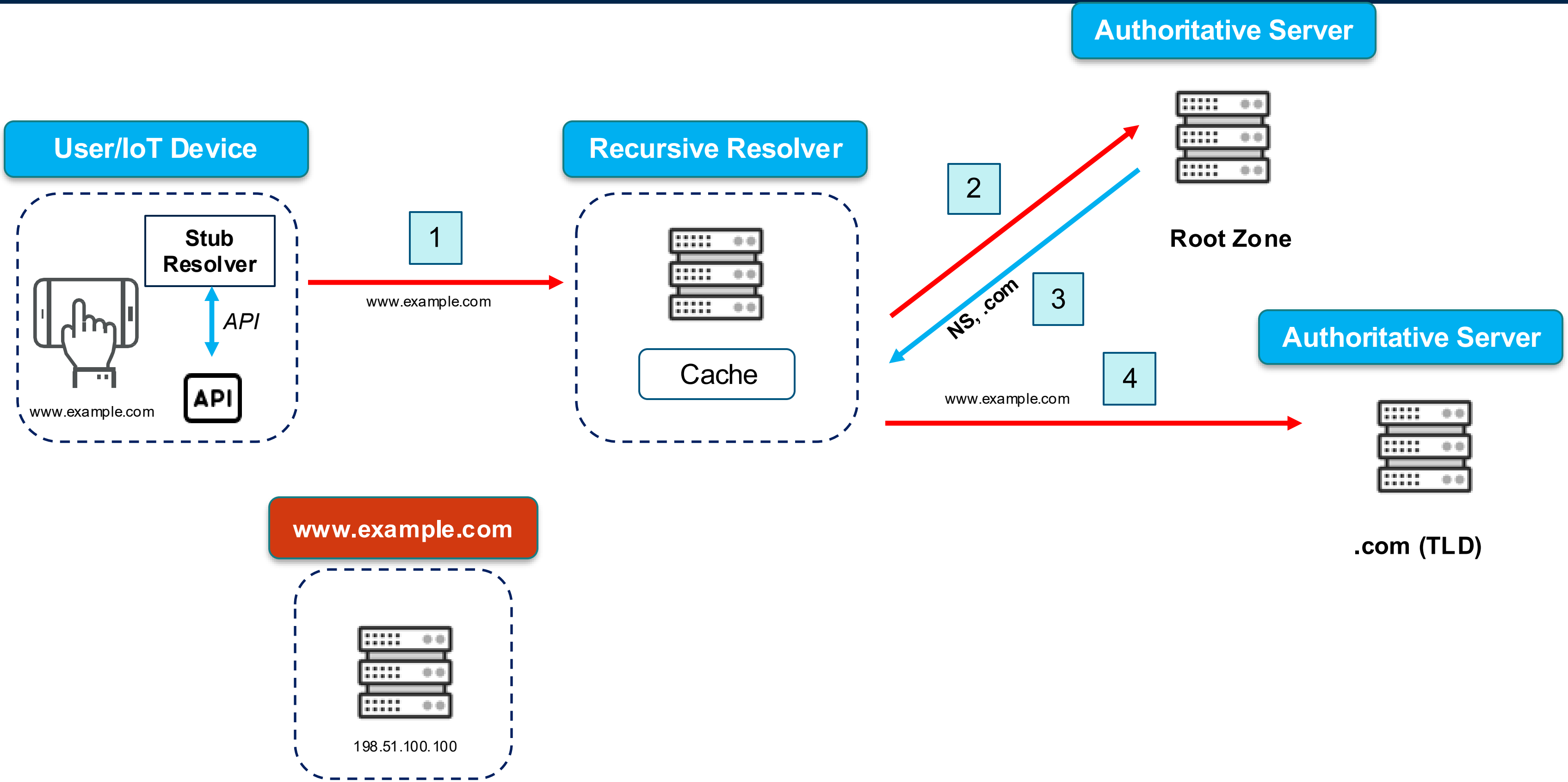


DNS Resolution process

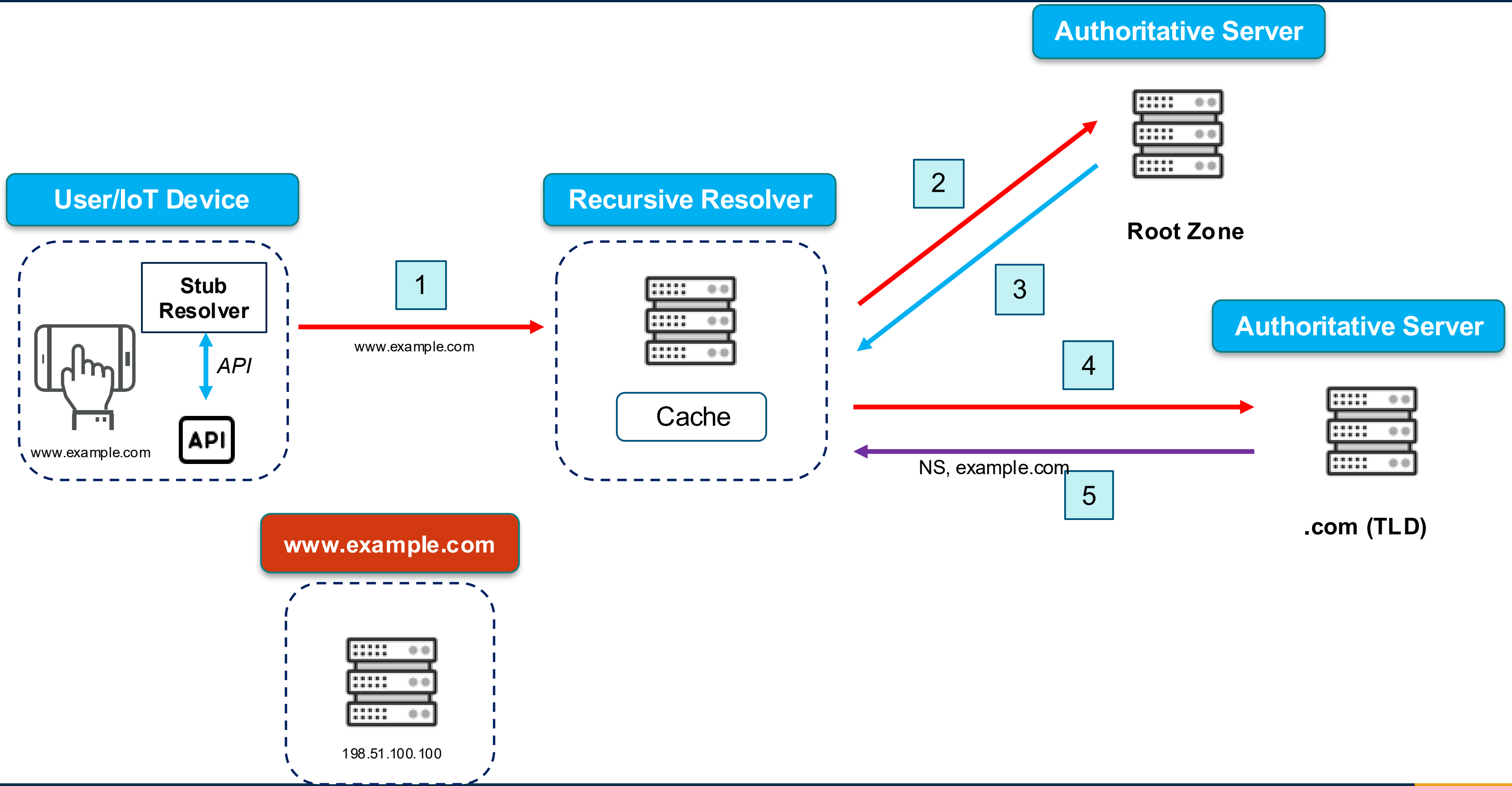


RESPONSE*: This is the name of the Authoritative server for the TLD you're looking for (NS record) and a GLUE Record that's the A/AAAA record corresponding to that NS record.*

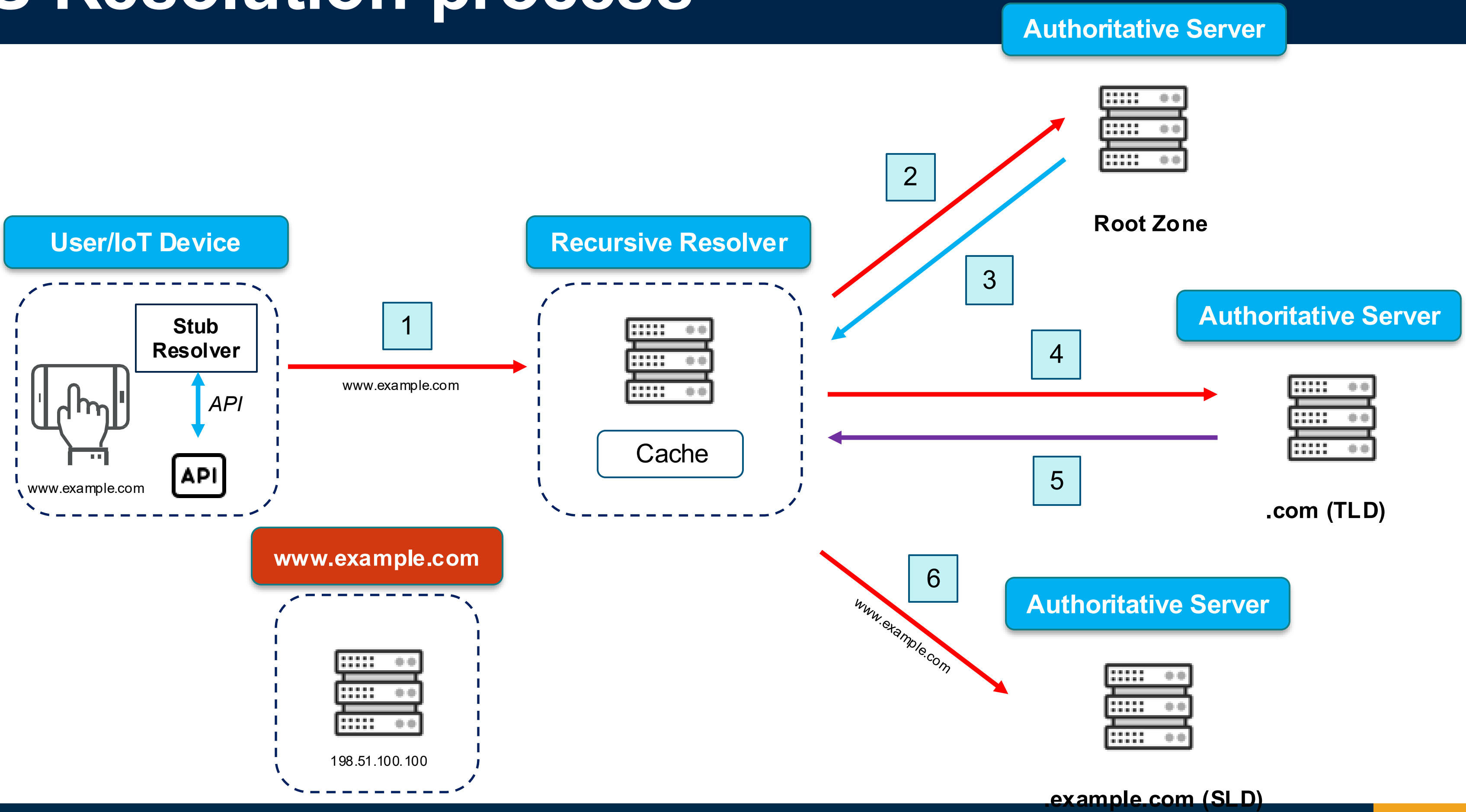
DNS Resolution process



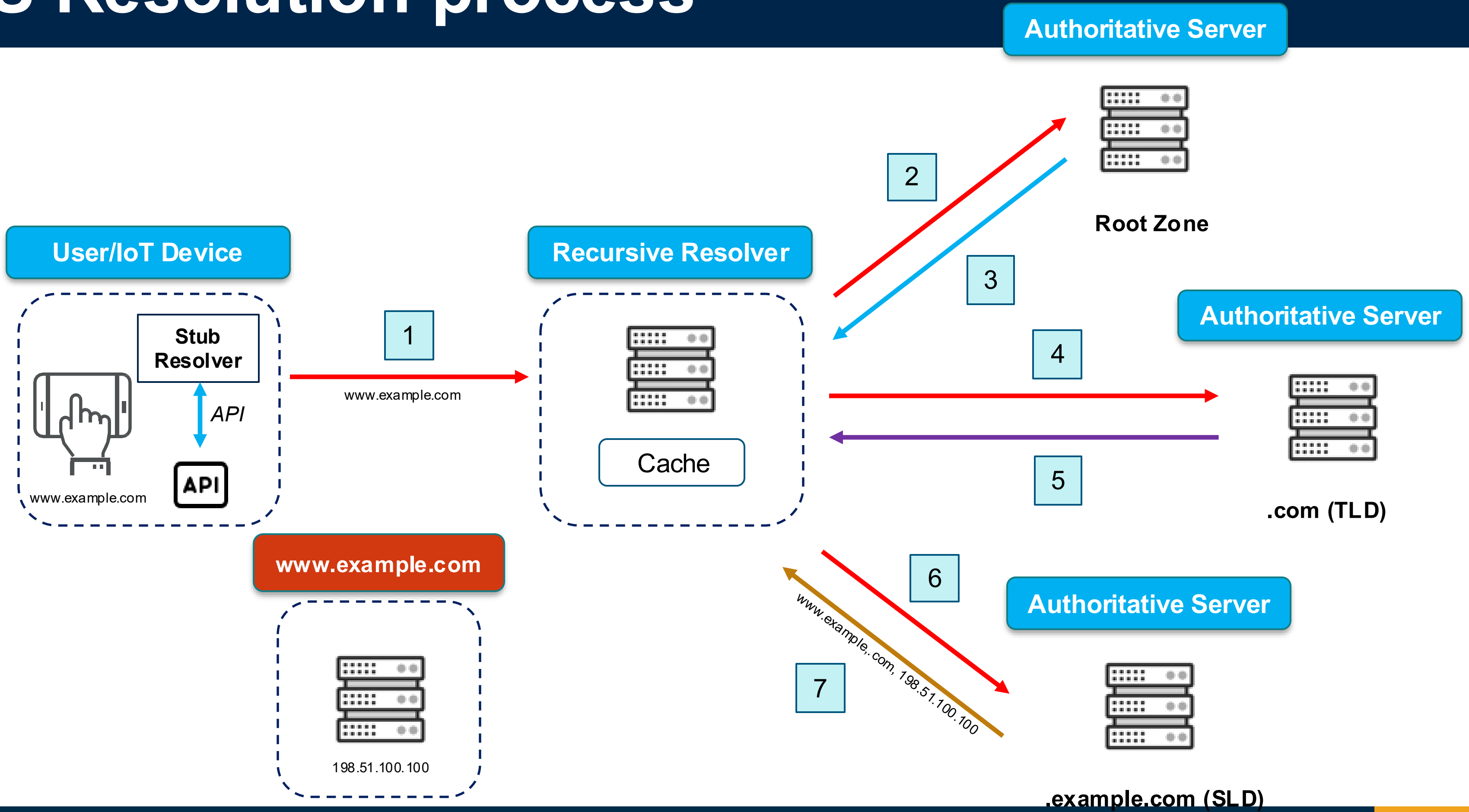
DNS Resolution process



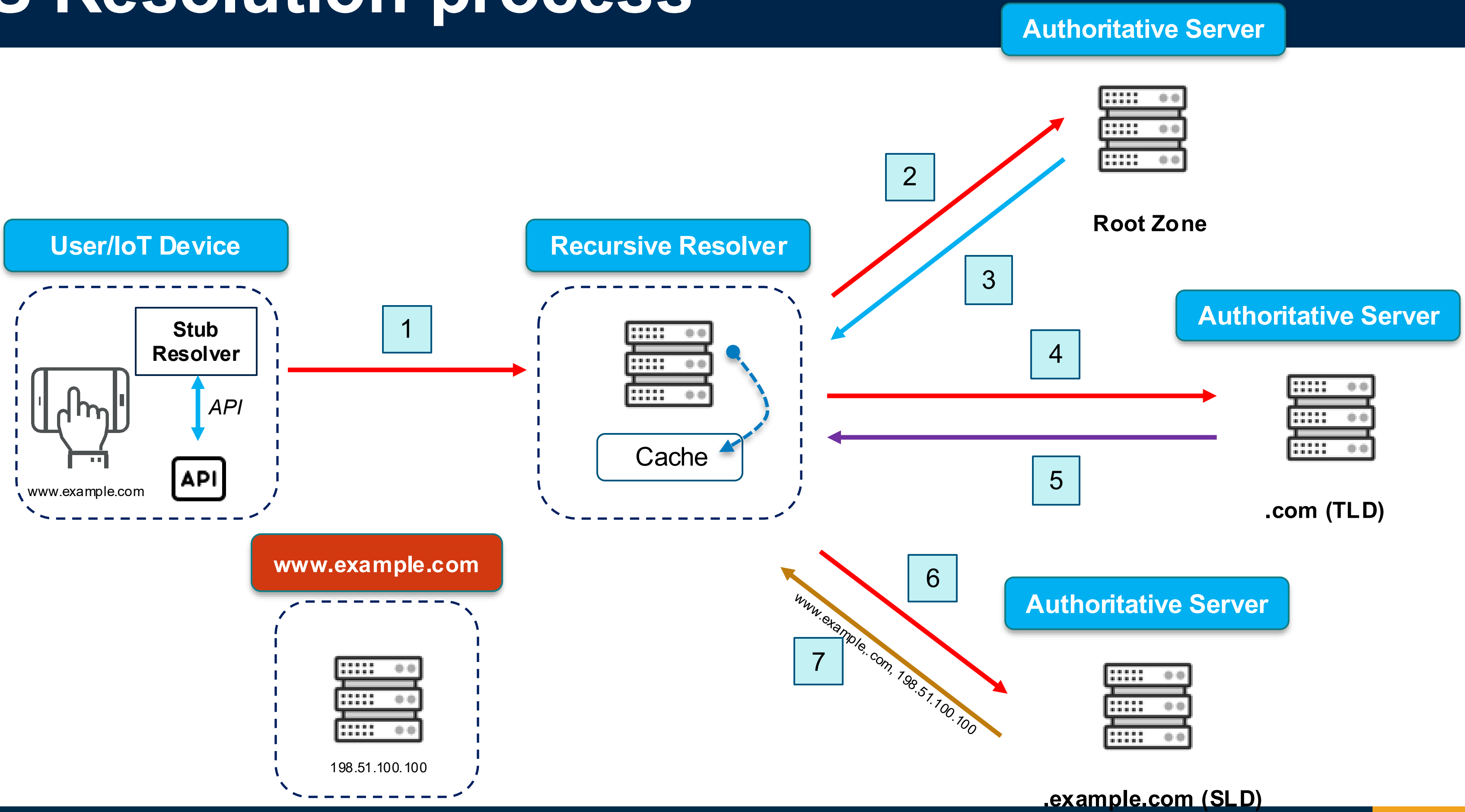
DNS Resolution process



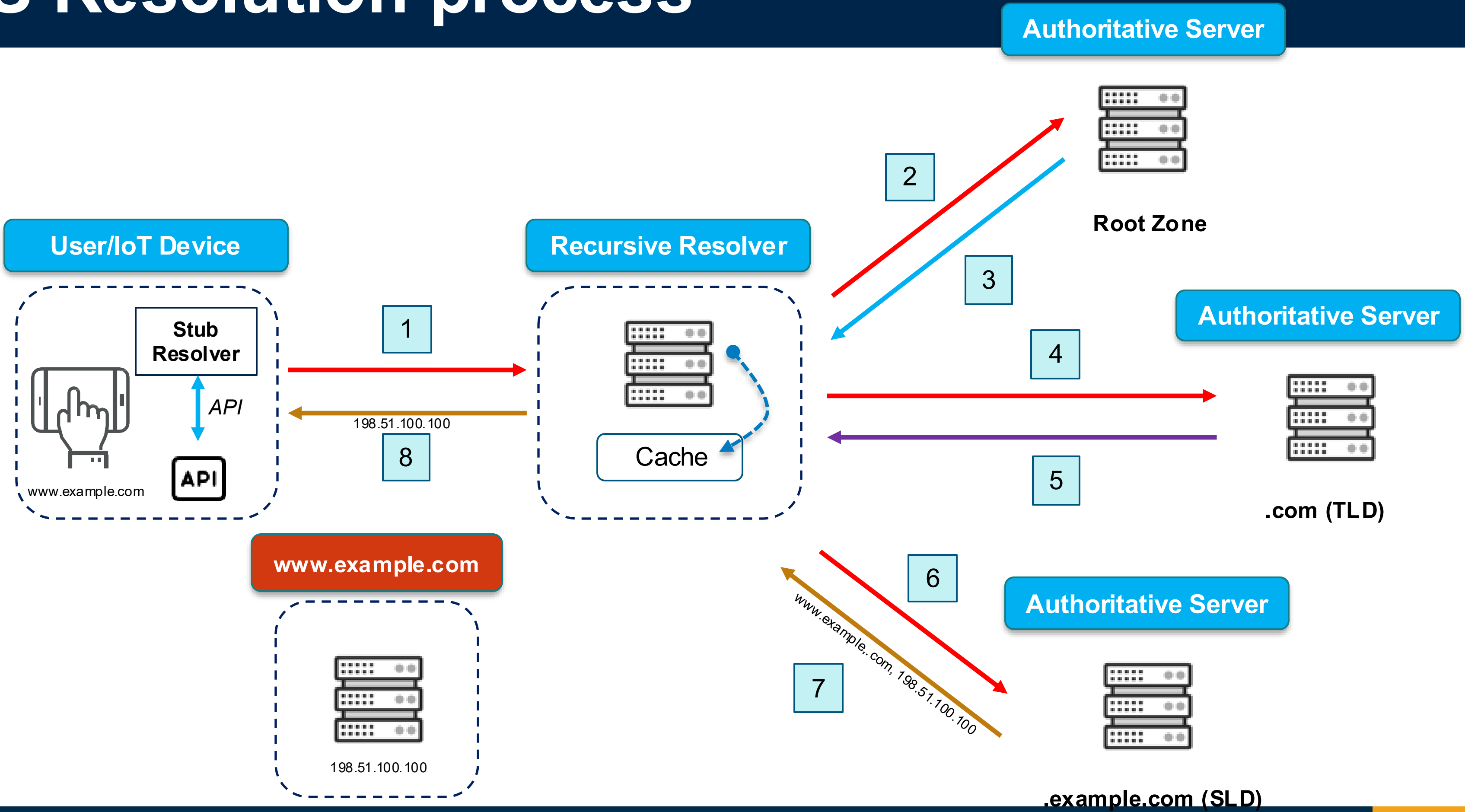
DNS Resolution process



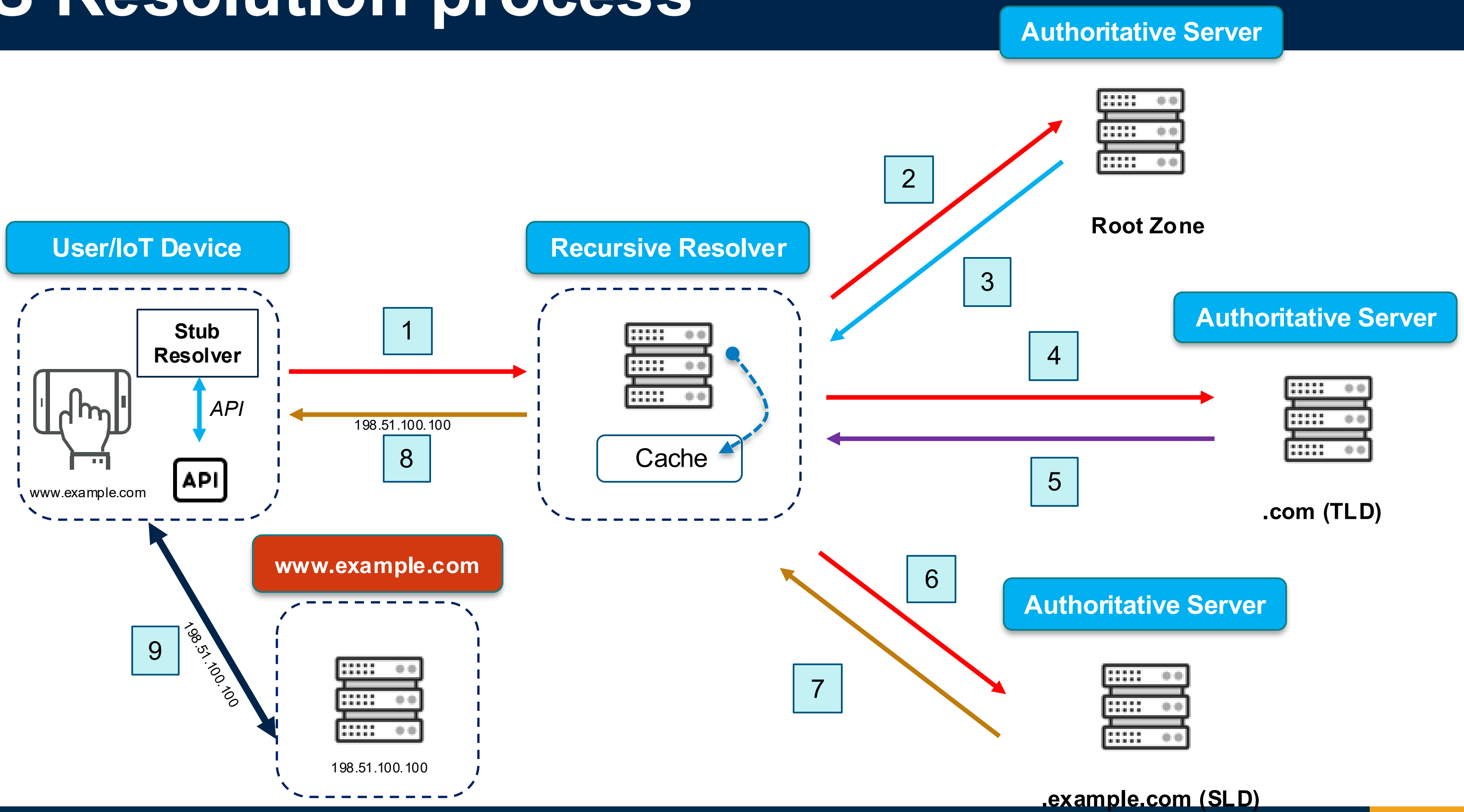
DNS Resolution process



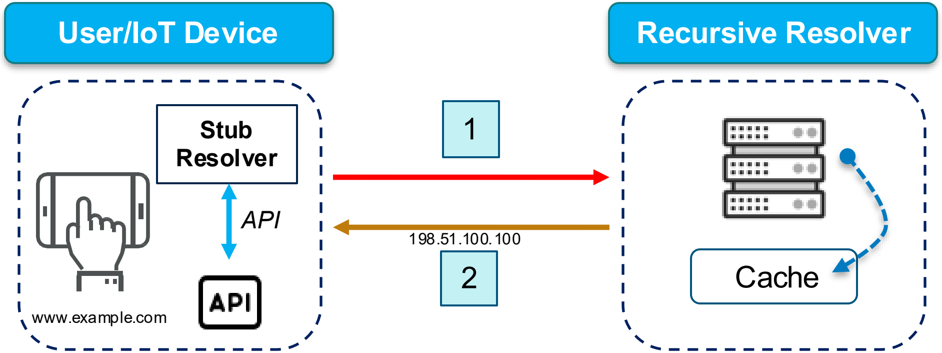
DNS Resolution process



DNS Resolution process

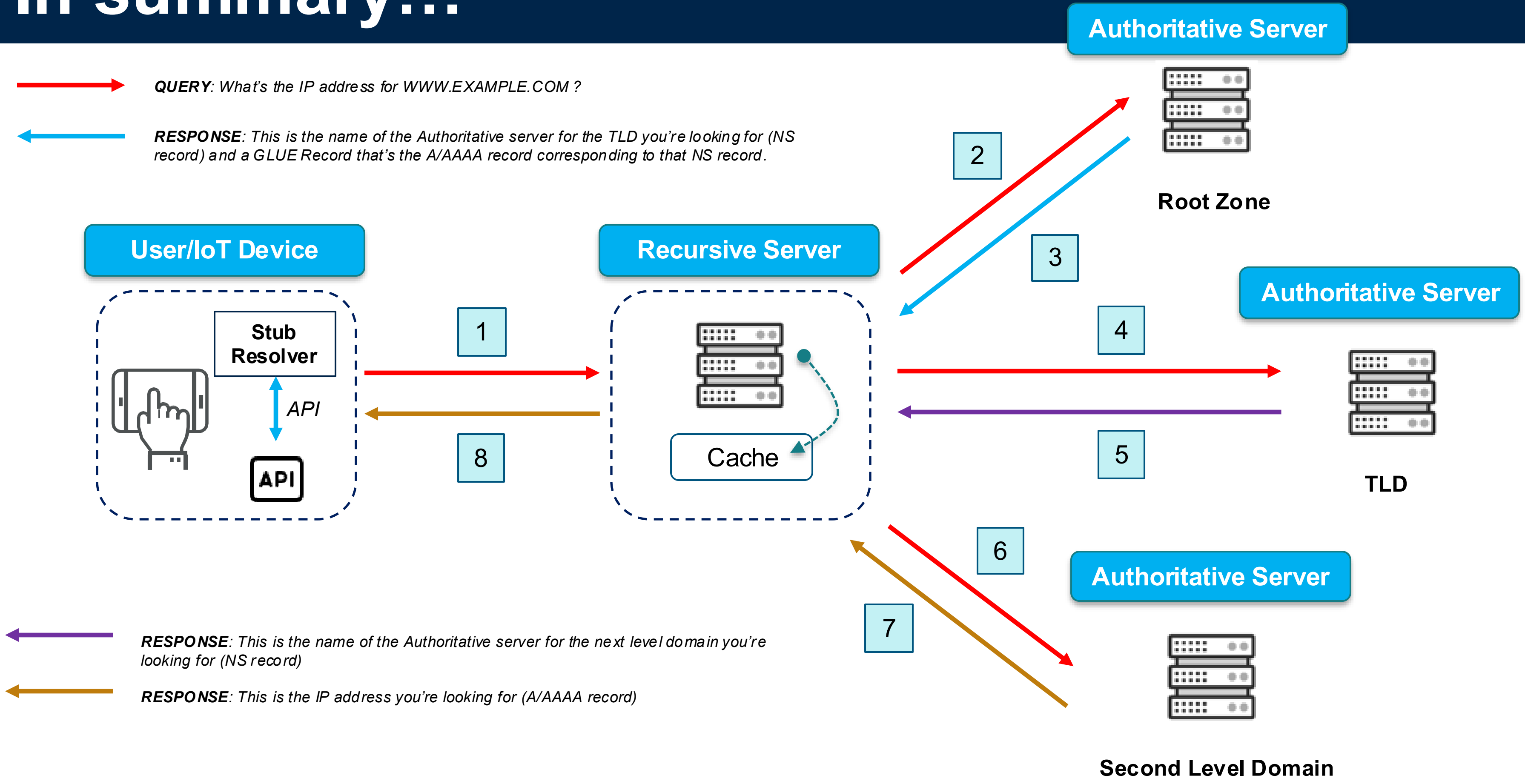


Caching



Next time someone using the same resolver ask the question, things are going be different

In summary…



QUERY*: What’s the IP address for WWW.EXAMPLE.COM ?*

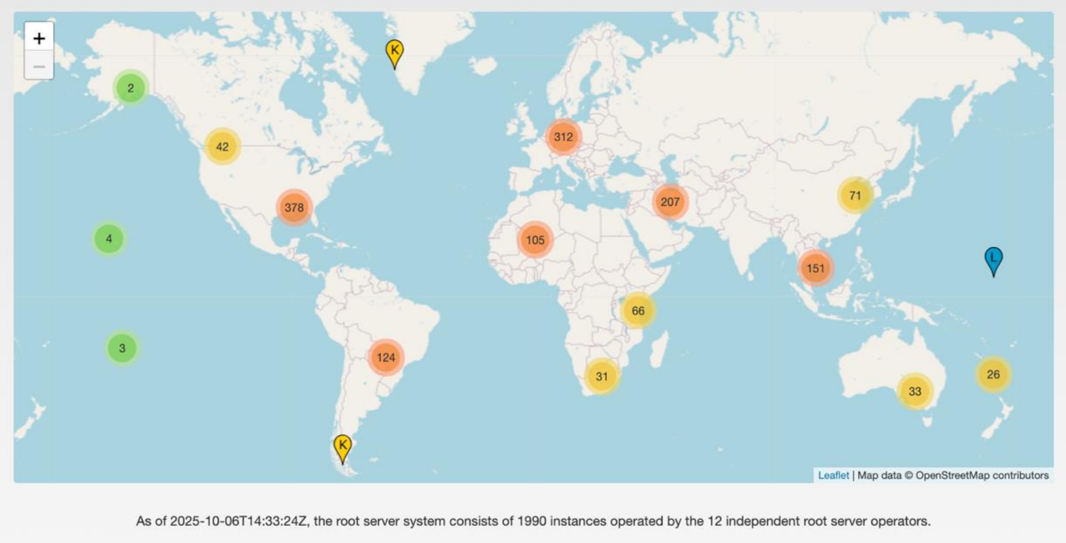
RESPONSE*: This is the name of the Authoritative server for the TLD you’re looking for (NS record) and a GLUE Record that’s the A/AAAA record corresponding to that NS record.*

RESPONSE*: This is the name of the Authoritative server for the next level domain you’re looking for (NS record)*

RESPONSE*: This is the IP address you’re looking for (A/AAAA record)*

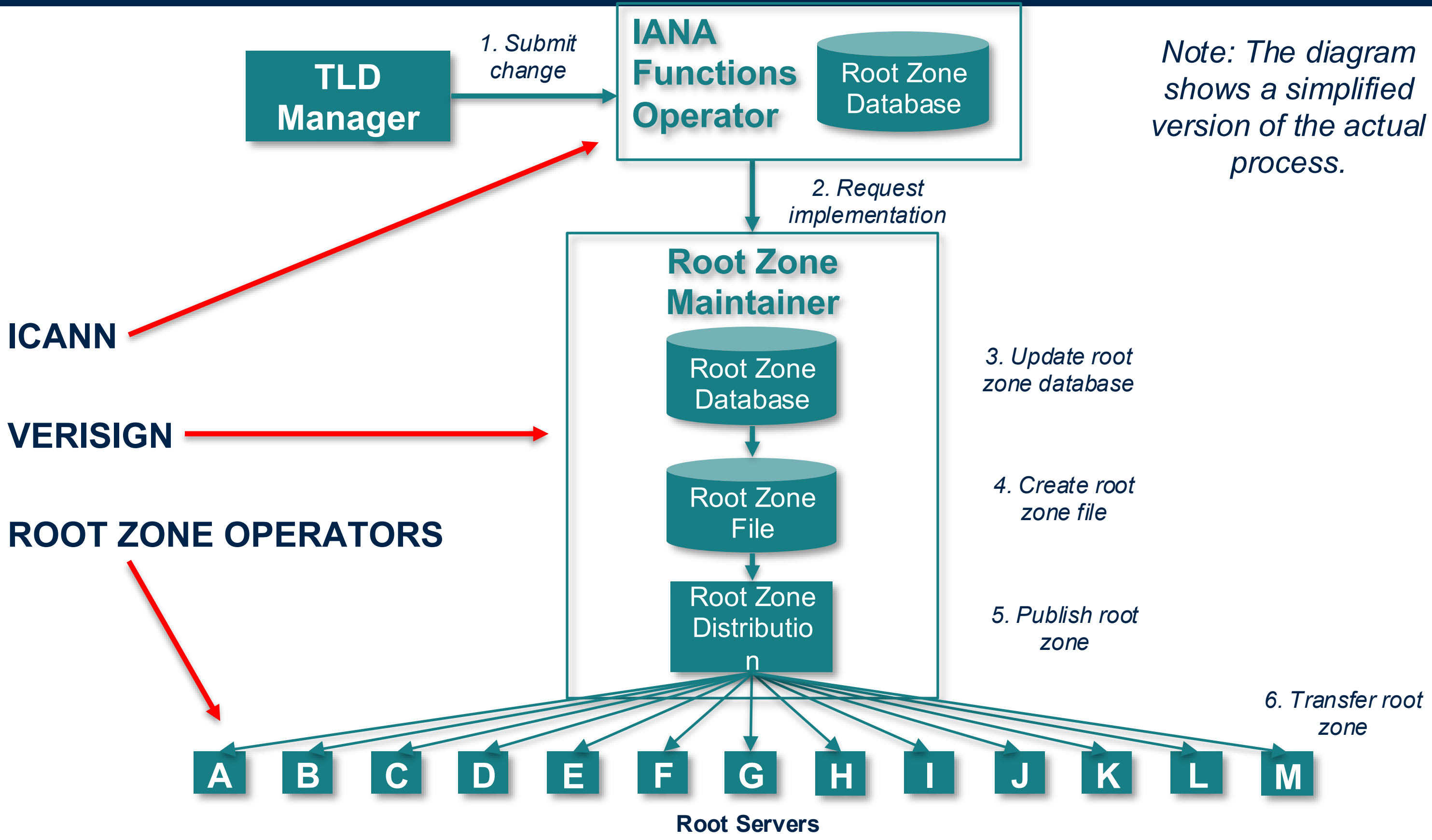
The Root Server System (RSS)

- 13 identities labeled from A to M.
- Operated by 12 Root Server Operators (RSO).
- 26 IP addresses: 13 IPv4 + 13 IPv6
- Almost 2,000 instances: anycast.
- More at https://root-servers.org/



As of 2025-10-06T14:33:24Z, the root server system consists of 1990 instances operated by the 12 independent root server operators.

Root Zone Change Process



Note: The diagram shows a simplified version of the actual process.

DNS Software overview

- Diversity of software platforms and vendors: commercial and open source.
  - Open Source: BIND, Unbound, Knot Resolver, PowerDNS Recursor, DNSMASQ, …
  - Commercial: Windows, Nominum Vantio (now part of Akamai), Secure64 DNS Cache, …
  - Overview : https://en.wikipedia.org/wiki/Comparison_of_DNS_server_software
- This list is not exhaustive and does not imply endorsement of any specific package.

ISC's BIND

- Authoritative server and cache all-in-one
- http://isc.org/
- Always changing, see current version on web site: https://www.isc.org/downloads/bind/
- Longest track record in DNSSEC

NLnetLab's Unbound

- a caching-only name server with DNSSEC built in
- http://www.unbound.net/
- "unbound" is a play on the word "bind"

cz.nic’s Knot Resolver

- a caching-only name server with DNSSEC built in
- http://www.knot-resolver.cz/download/
- “Knot" is a play on the words "bind” and “unbound” (see a trend?)

PowerDNS Recursor

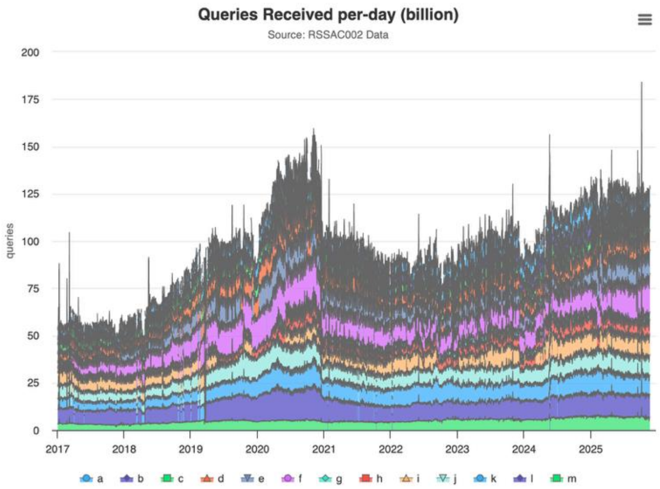
- Caching resolver
- Supports DNSSEC validation
- https://www.powerdns.com/documentation.html
- Also, an authoritative server
- name is not related to BIND, unbound, Knot

DNSMASQ

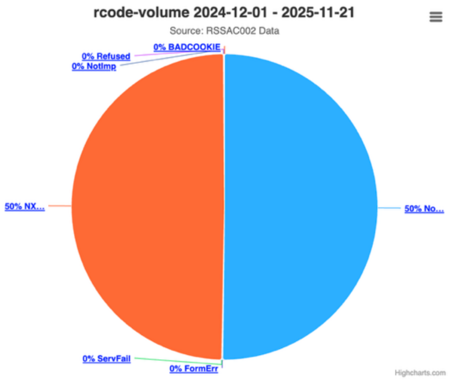
- provides network infrastructure for small networks: DNS, DHCP, router advertisement and network boot
- authoritative and cache
- Supports DNSSEC (validation)
- main page: http://www.thekelleys.org.uk/dnsmasq/doc.html

A few statistics

Traffic to root servers
https://rssac002.root-servers.org

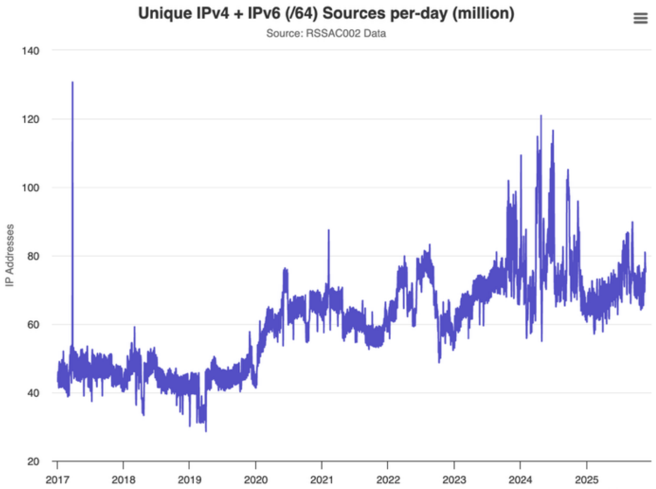

Queries Received per-day (billion)

Source: RSSAC002 Data


rcode-volume 2024-12-01 - 2025-11-21

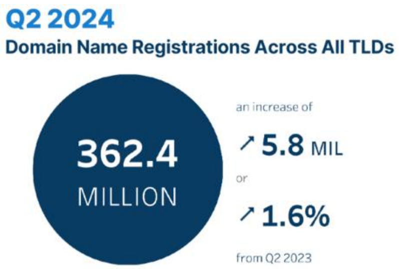
Source: RSSAC002 Data

Start 2024-12-01 End 2025-11-21 RSIs a-m


Unique IPv4 + IPv6 (/64) Sources per-day (million)

Source: RSSAC002 Data

A few statistics



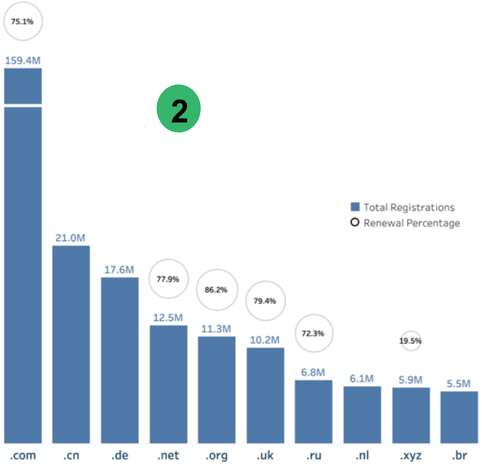
Q2 2024

Domain Name Registrations Across All TLDs

362.4 MILLION

an increase of ↗ **5.8** MIL or ↗ **1.6%** from Q2 2023

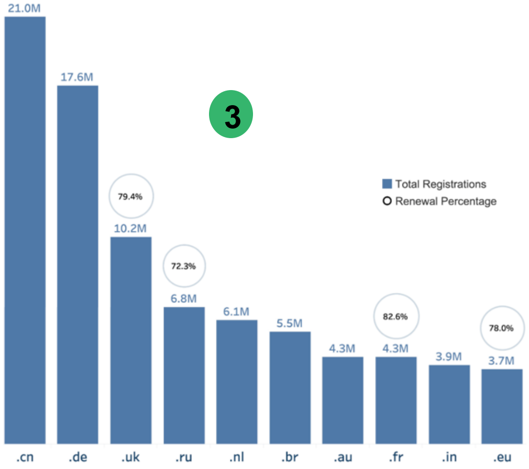
1



2

Top 10 largest TLDs by # of reported domain names

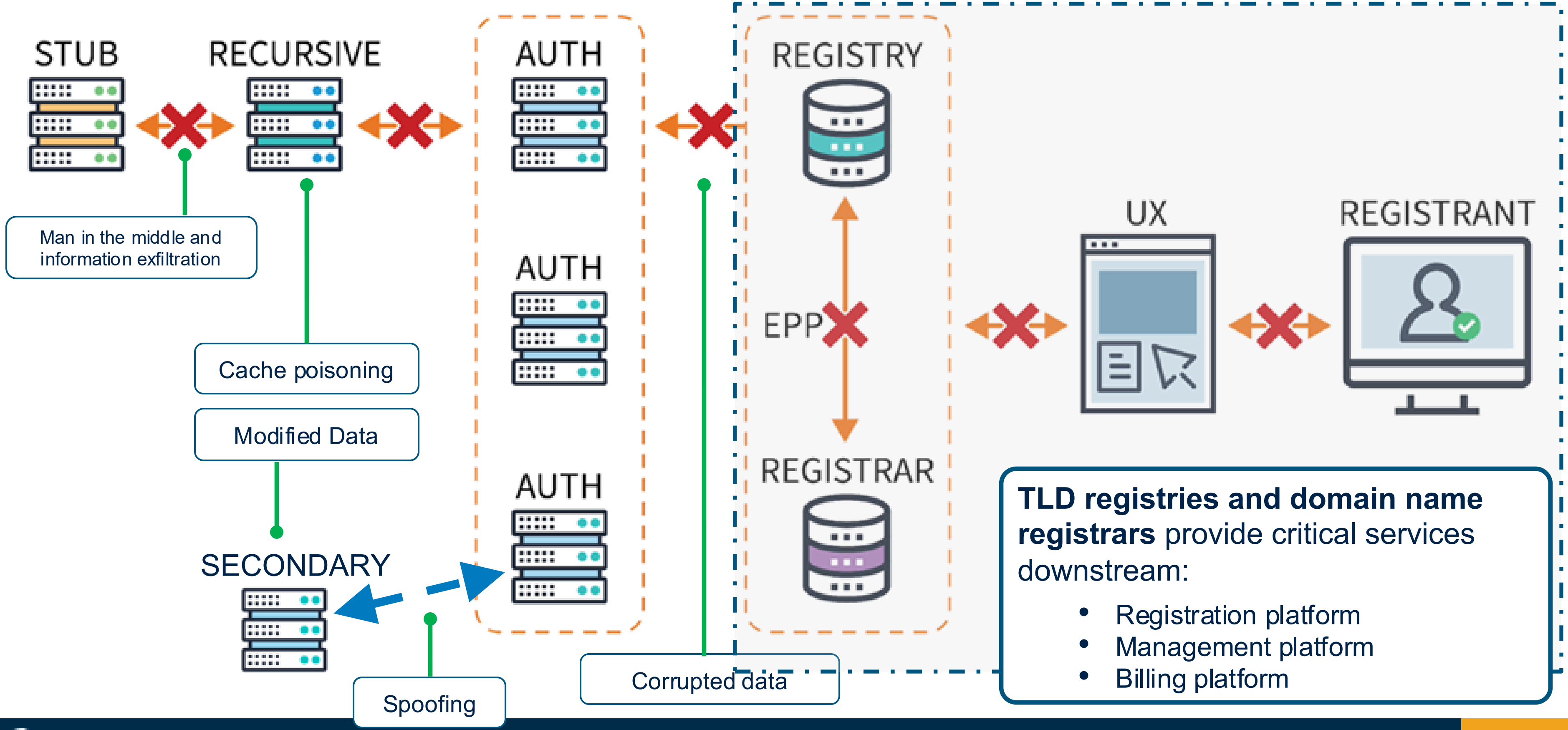
The DNIB Quarterly Report :
https://dnib.com/articles/latest-report



3

Top 10 largest ccTLDs by # of reported domain names

Potential Target Points of the DNS



Man in the middle and information exfiltration

Cache poisoning

Modified Data

Spoofing

Corrupted data

TLD registries and domain name registrars provide critical services downstream:

- Registration platform
- Management platform
- Billing platform

Time for practice !

1. Configure your own zone’s primary and secondaries NS
2. Confirm they are all in sync, serving and responding well for the zone.

ICANN Webpage and Social Media Links

icann.org

@icann

facebook.com/icannorg

youtube.com/icannnews

flickr.com/icann

linkedin.com/company/icann

instagram.com/icannorg