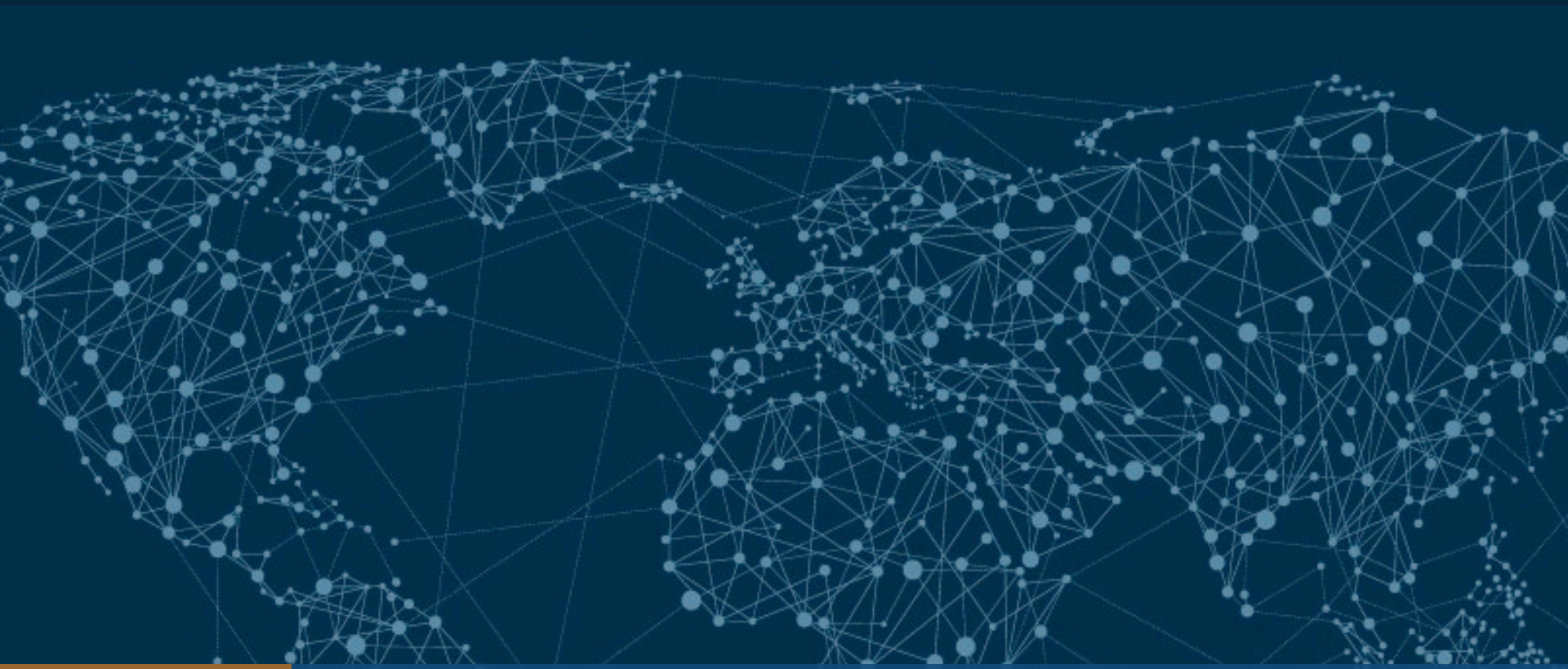


DNS Primer

Matt Larson | GDD Industry Summit | 11 May 2017

Names and Numbers

- IP addresses easy for machines but hard for people
  - IPv4: 192.0.2.7
  - IPv6: 2001:db8::7
- People need to use 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

- Mapping names to IP addresses to names is ***name resolution***
- Name resolution on the early Internet used a ***host file*** named HOSTS.TXT
  - Same function but slightly different format than the familiar */etc/hosts*
- Centrally maintained by the NIC (Network Information Center) at the Stanford Research Institute (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 just to download the file

- A centrally maintained host file just didn't scale

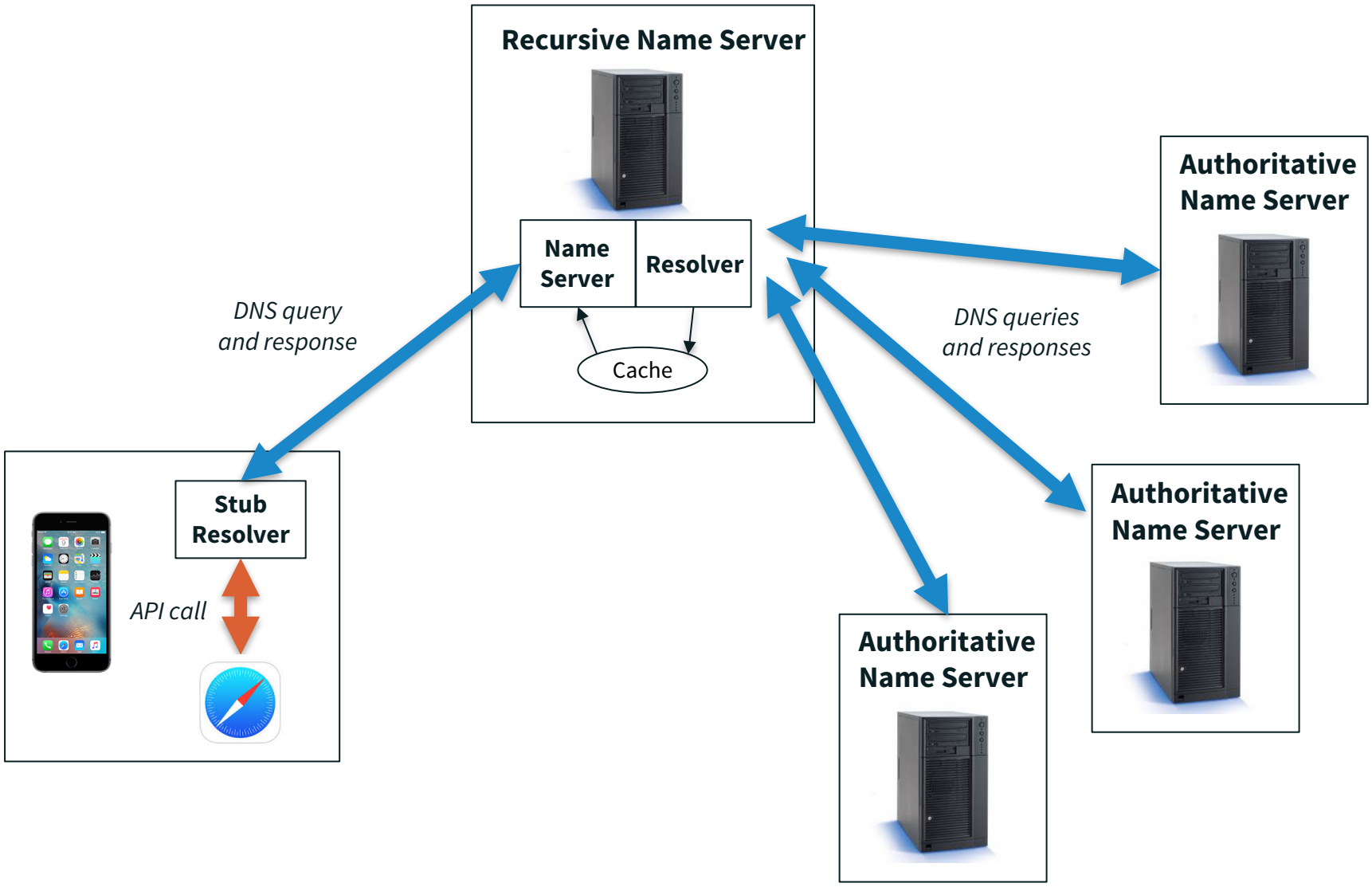
DNS to the Rescue

- Discussion started in the early 1980s on a replacement
- 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”

DNS in a nutshell

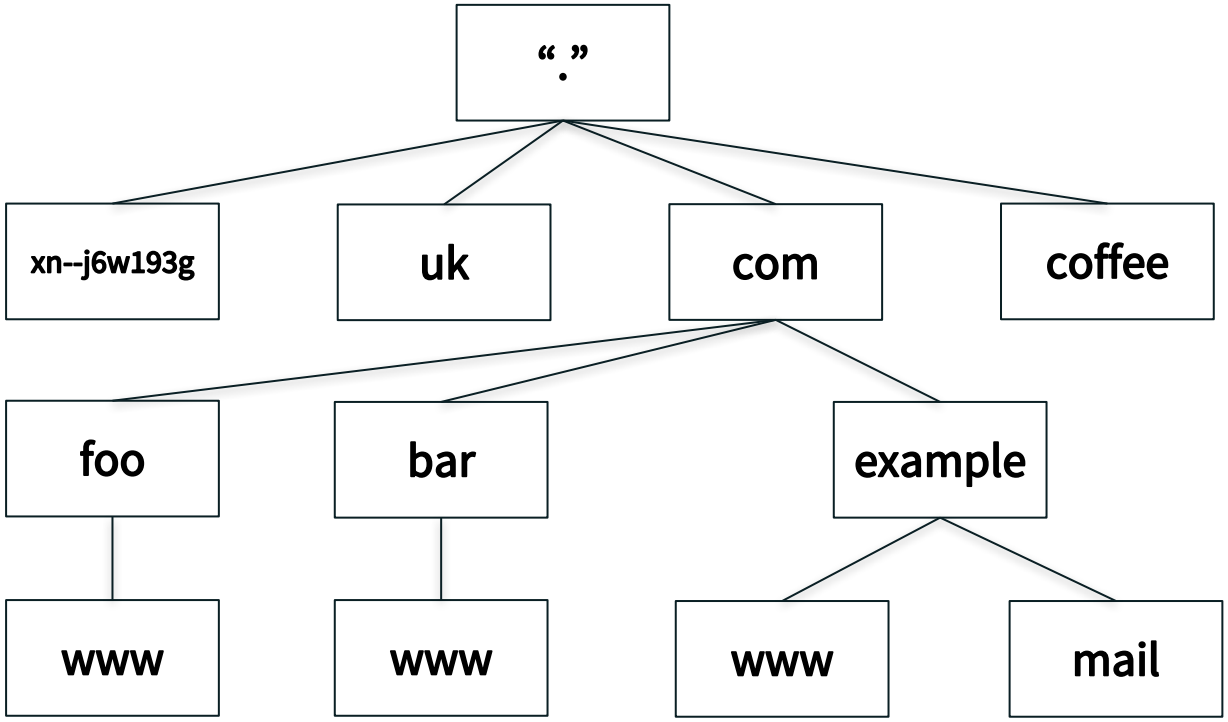
- DNS is a distributed database
  - Data is maintained locally but available globally
- ***Resolvers*** send queries
- ***Name servers*** answer queries
- Optimizations:
  - Caching to improve performance
  - Replication to provide redundancy and load distribution

DNS Components at a Glance



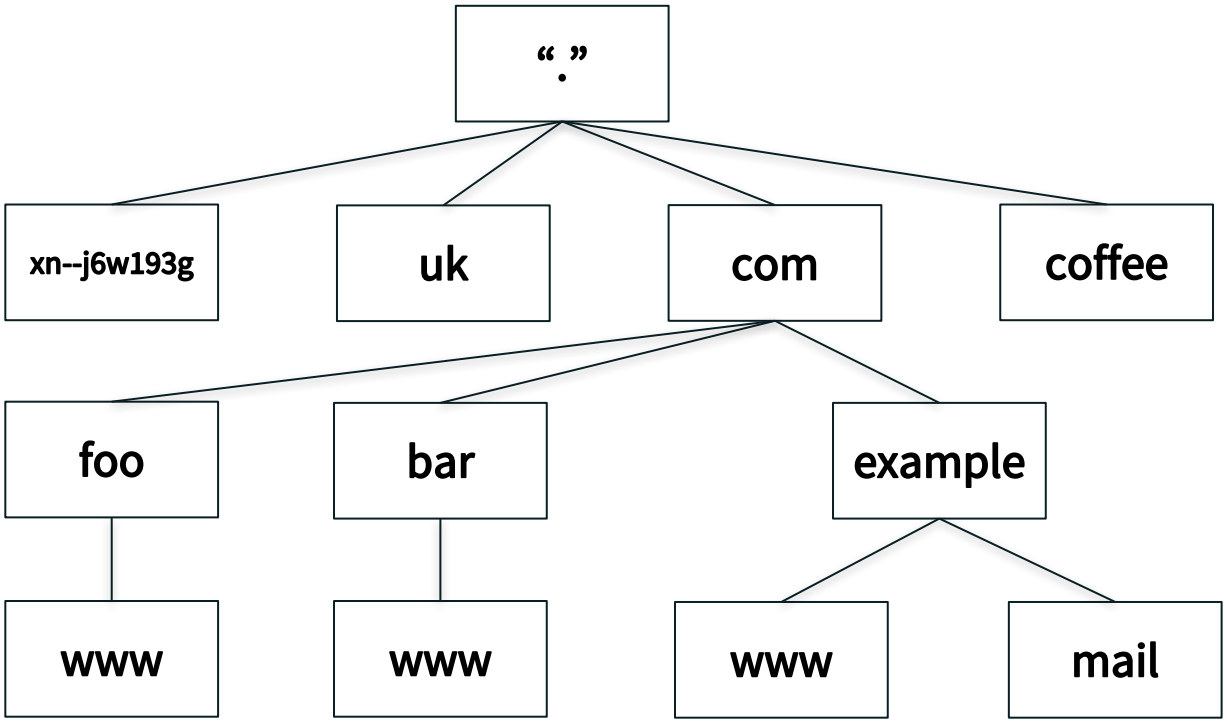
The Name Space

- 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



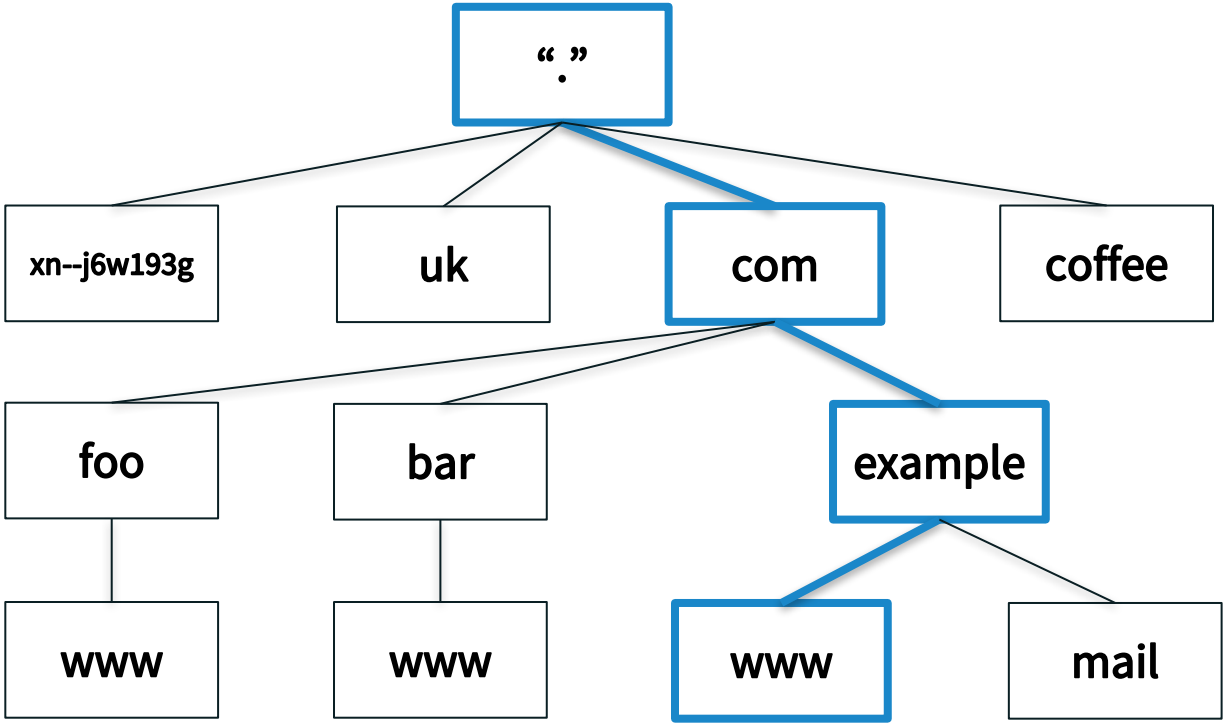
Label Syntax

- Legal characters for labels are “LDH” (letters, digits, hyphen)
- Maximum length 63 characters
- Comparisons of label names are not case sensitive



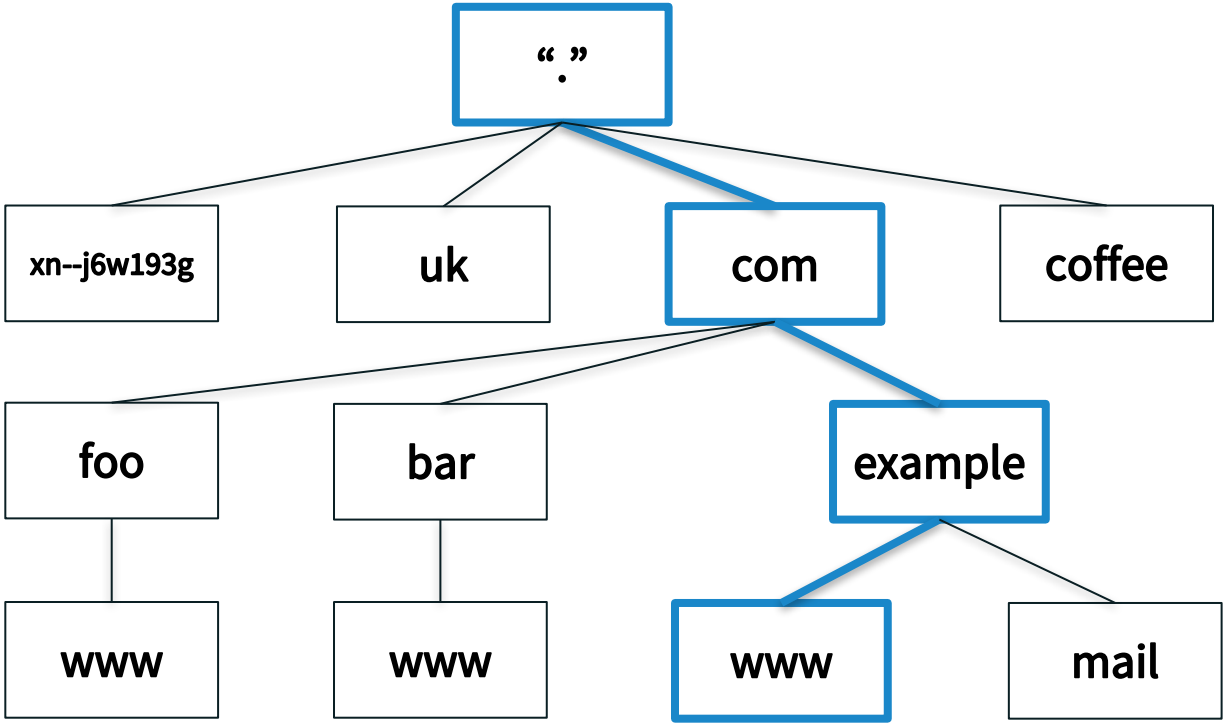
Domain Names

- Every node has a ***domain name***
- Sequence of labels from the node to the root separated by dots
- Highlighted: *www.example.com.*



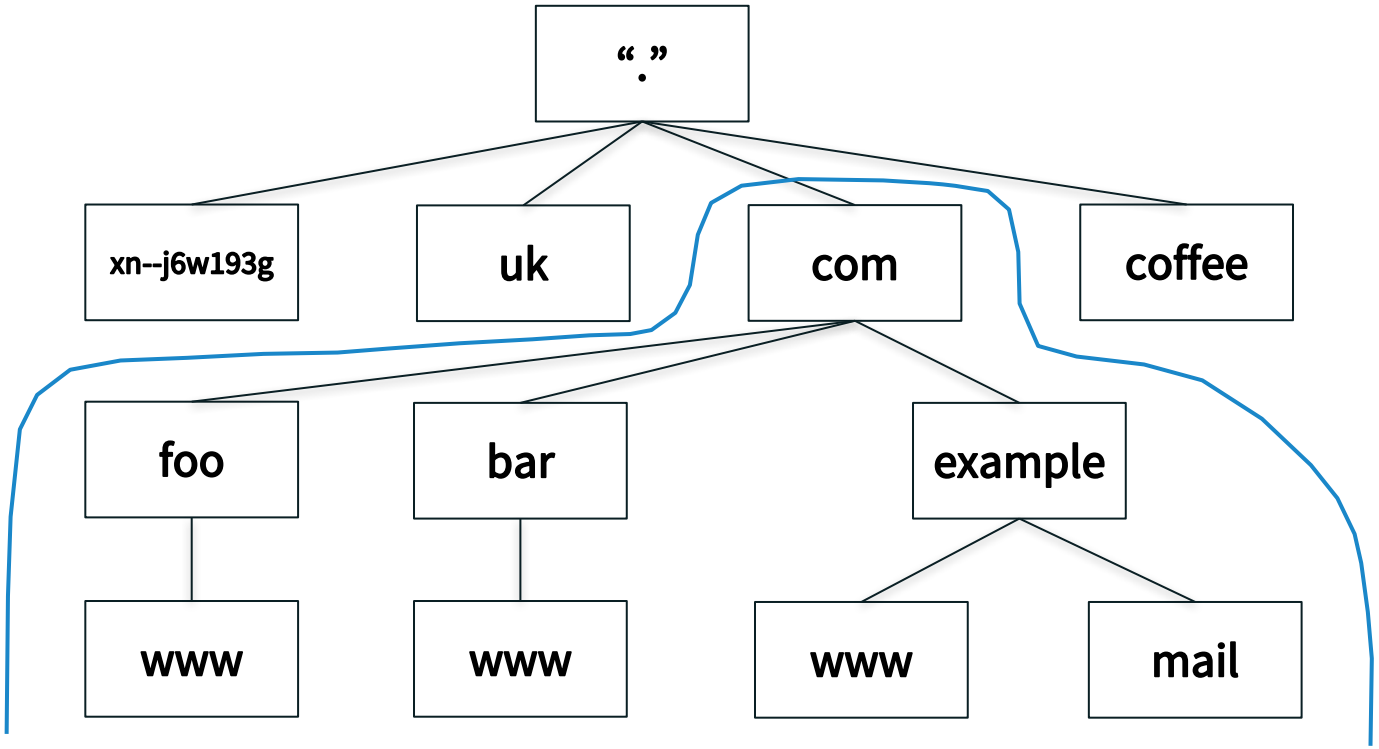
Fully Qualified Domain Names

- A ***fully qualified domain name (FQDN)*** unambiguously identifies a node
  - Not relative to any other domain name
- An FQDN ends in a dot
- Example FQDN: *www.example.com.*



Domains

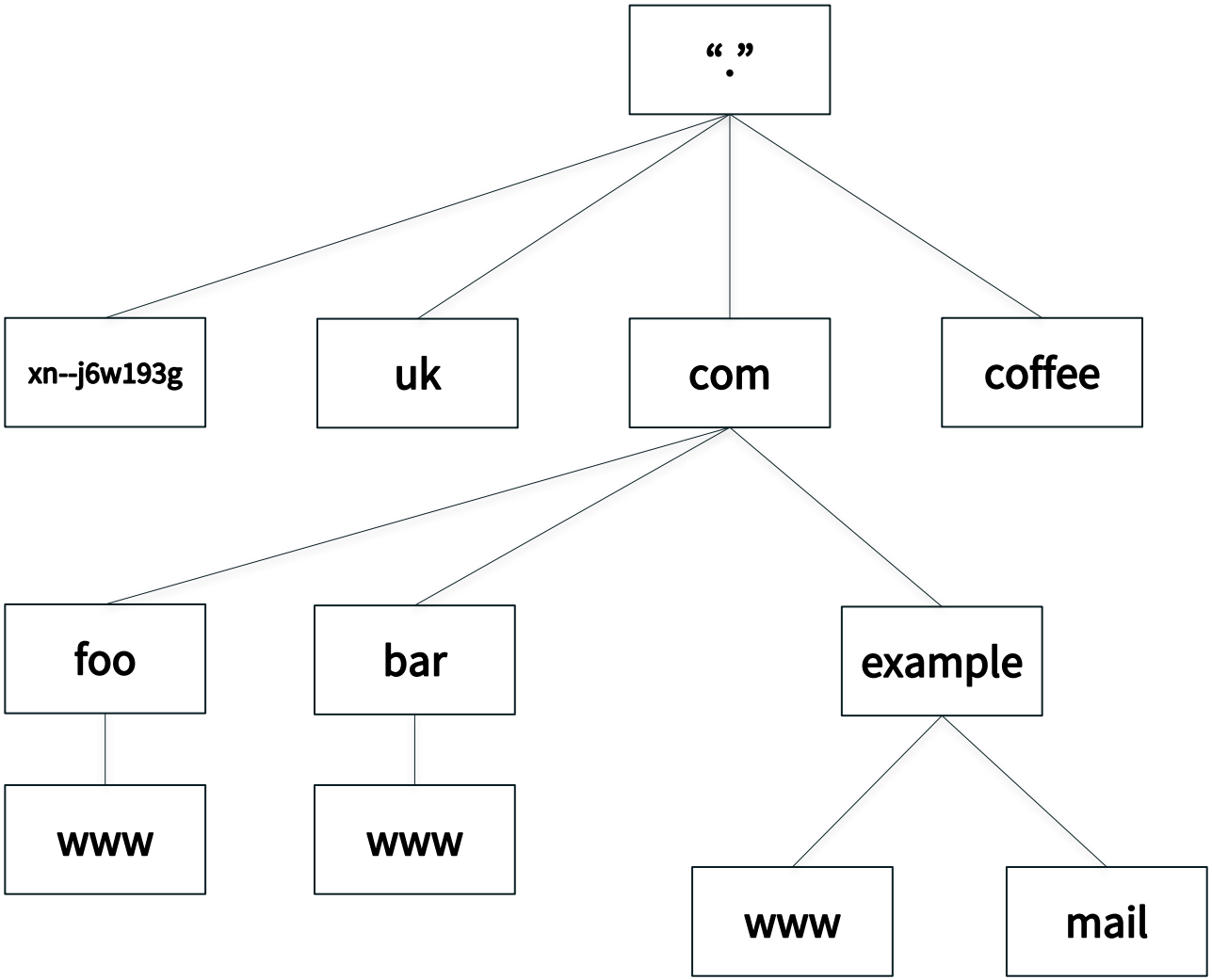
- A ***domain*** is a node and everything below it (its descendants)
- The top node of a domain is the ***apex*** of that domain
- Shown: the *com* domain



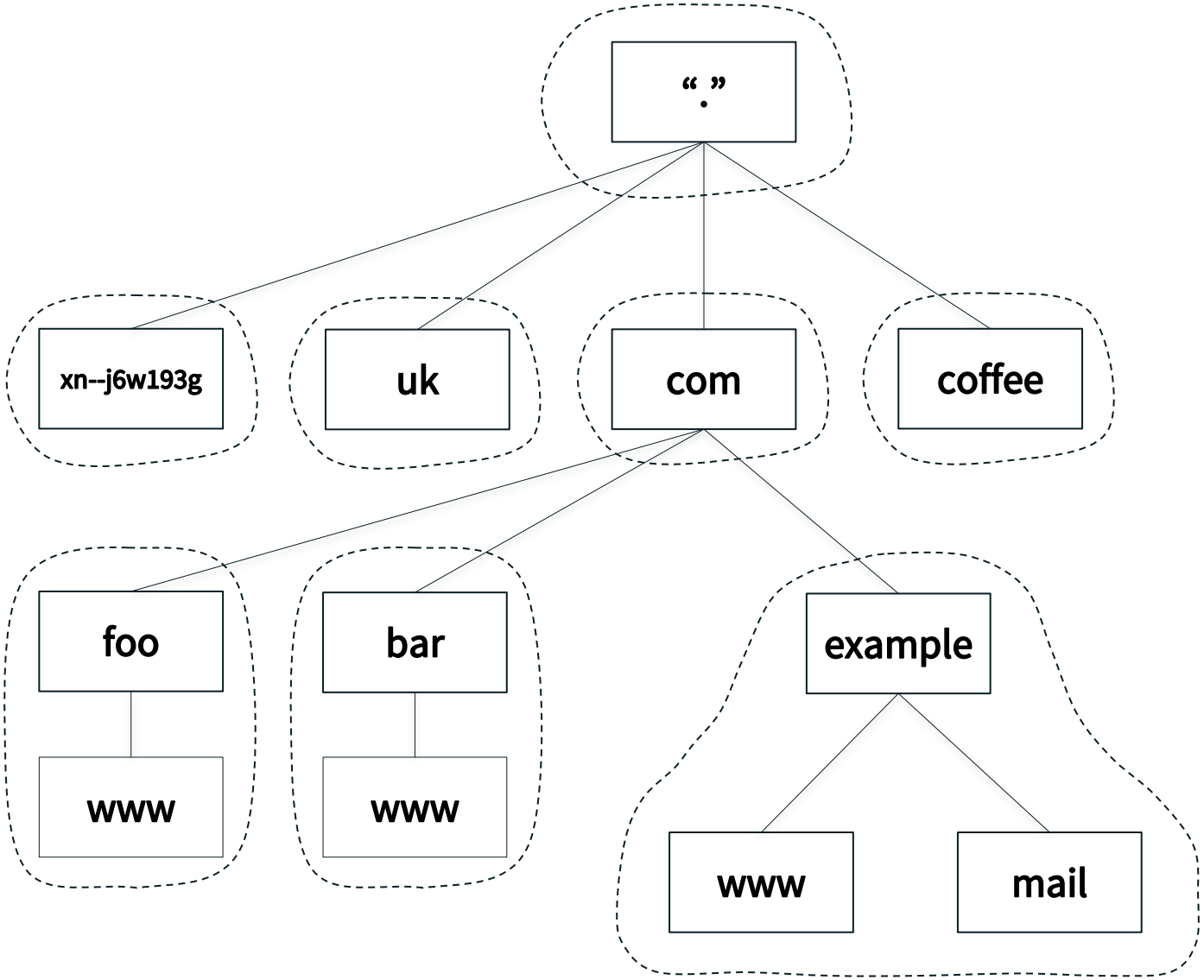
Zones

- The name space is divided up to allow distributed administration
- Administrative divisions are called ***zones***
- ***Delegation*** creates zones
  - Delegating zone is the ***parent***
  - Created zone is the ***child***

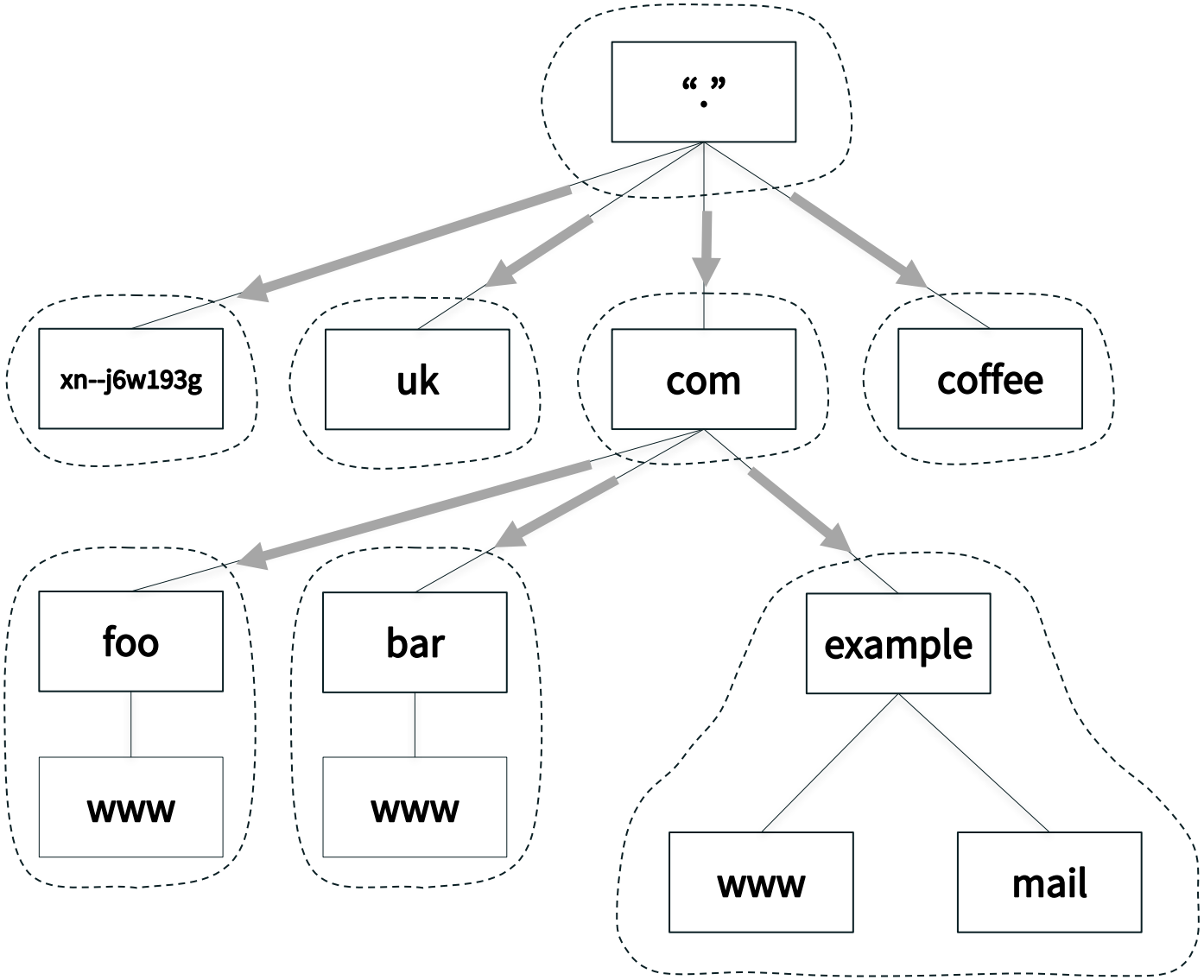
The Name Space



Zones are Administrative Boundaries



Delegation Creates Zones



Name Servers and Zones

- Name servers answer queries
- A name server ***authoritative*** for a zone has complete knowledge of that zone
  - Can provide a definitive answer to queries about the zone
- Zones should have multiple authoritative servers
  - Provides redundancy
  - Spreads the query load

Authoritative Server Synchronization

- How do you keep a zone's data in sync across multiple authoritative servers?
- Fortunately zone replication is built into the DNS protocol
- A zone's ***primary*** name server has the definitive zone data
  - Changes to the zone are made on the primary
- A zone's ***secondary*** or ***slave*** server retrieves the zone data from another authoritative server via a ***zone transfer***
  - The server it retrieves from is called the ***master server***
  - Master server is usually the primary but doesn't have to be
- Zone transfer is initiated by the secondary
  - Secondary polls the master periodically to check for changes
  - The master also notifies the primary of changes
    - RFC 1996, "A Mechanism for Prompt Notification of Zone Changes (DNS NOTIFY)"

DNS Data

- The DNS standard specifies the format of DNS packets sent over the network
  - Informally called “wire format”
- The standard also specifies a text-based representation for DNS data called ***master file format***
- A ***zone file*** contains all the data for a zone in master file format

DNS Resource Records

- Recall every node has a domain name
- A domain name can have different kinds of data associated with it
- That data is stored in ***resource records***
  - Sometimes abbreviated as ***RRs***
- Different record types for different kinds of data

Zone Files

- A zone consists of multiple resource records
- All the resource records for a zone are stored in a ***zone file***
- Every zone has (at least) one zone file
- Resource records from multiple zones are never mixed in the same file

Format of Resource Records

- Resource records have five fields:
  - ***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

Master File Format

- Resource record syntax in master file format:

```
[owner]    [TTL]    [class]    type    RDATA
```

- Fields in brackets are optional
  - Shortcuts to make typing zone files easier on humans
- Type and RDATA always appear

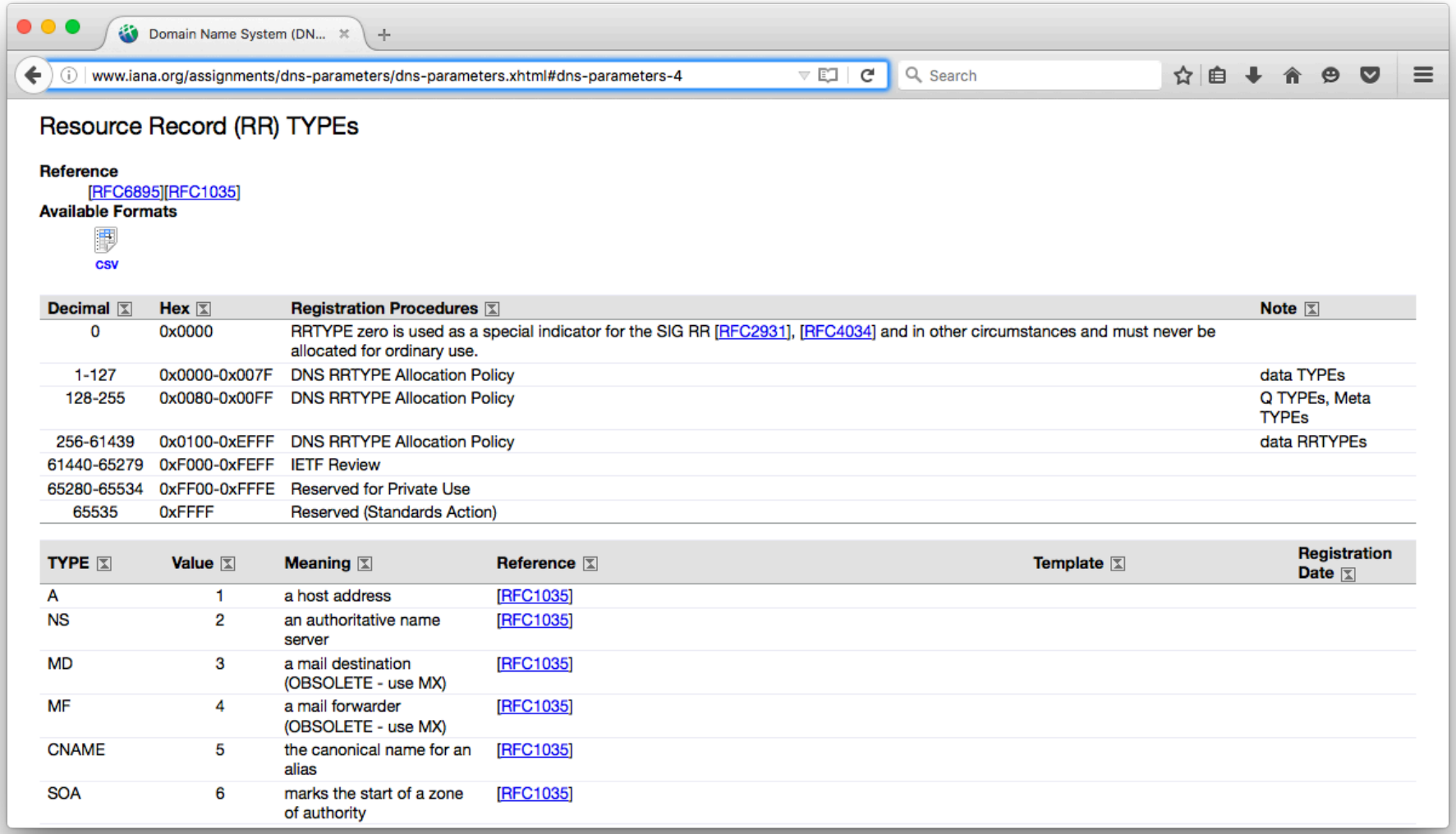
Common Resource Record Types

- **A** IPv4 address
- **AAAA** IPv6 address
- **NS** Name of an authoritative name server
- **SOA** “Start of authority”, appears at zone apex
- **CNAME** Name of an alias to another domain name
- **MX** Name of a “mail exchange server”
- **PTR** IP address encoded as a domain name (for reverse mapping)

Lots of Resource Records

- There are many other resource record types
- 84 types allocated as of August, 2016
- IANA “DNS Resource Record (RR) TYPE Registry” under “Domain Name System (DNS) Parameters”
  - *http://www.iana.org/assignments/dns-parameters/dns-parameters.xhtml#dns-parameters-4*

IANA DNS Resource Record (RR) TYPE Registry



Resource Record (RR) TYPEs

Reference

[RFC6895][RFC1035]

Available Formats

csv

Decimal	Hex	Registration Procedures	Note
0	0x0000	RRTYPE zero is used as a special indicator for the SIG RR [RFC2931], [RFC4034] and in other circumstances and must never be allocated for ordinary use.	
1-127	0x0000-0x007F	DNS RRTYPE Allocation Policy	data TYPEs
128-255	0x0080-0x00FF	DNS RRTYPE Allocation Policy	Q TYPEs, Meta TYPEs
256-61439	0x0100-0xEFFF	DNS RRTYPE Allocation Policy	data RRTYPEs
61440-65279	0xF000-0xFEFF	IETF Review	
65280-65534	0xFF00-0xFFFE	Reserved for Private Use	
65535	0xFFFF	Reserved (Standards Action)	

TYPE	Value	Meaning	Reference	Template	Registration Date
A	1	a host address	[RFC1035]		
NS	2	an authoritative name server	[RFC1035]		
MD	3	a mail destination (OBSOLETE - use MX)	[RFC1035]		
MF	4	a mail forwarder (OBSOLETE - use MX)	[RFC1035]		
CNAME	5	the canonical name for an alias	[RFC1035]		
SOA	6	marks the start of a zone of authority	[RFC1035]		

Address Records

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

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

  - “Quad A” (AAAA) record stores an IPv6 address

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

Warehouse Analogy

- Most types are used by consumers of DNS
  - A, AAAA and almost everything else
- Some types are used mostly by DNS itself
  - NS, SOA

- DNS is like a warehouse
  - NS and SOA are the shelves you build…
  - …so you can store stuff you care about (A, AAAA, etc.) in the warehouse

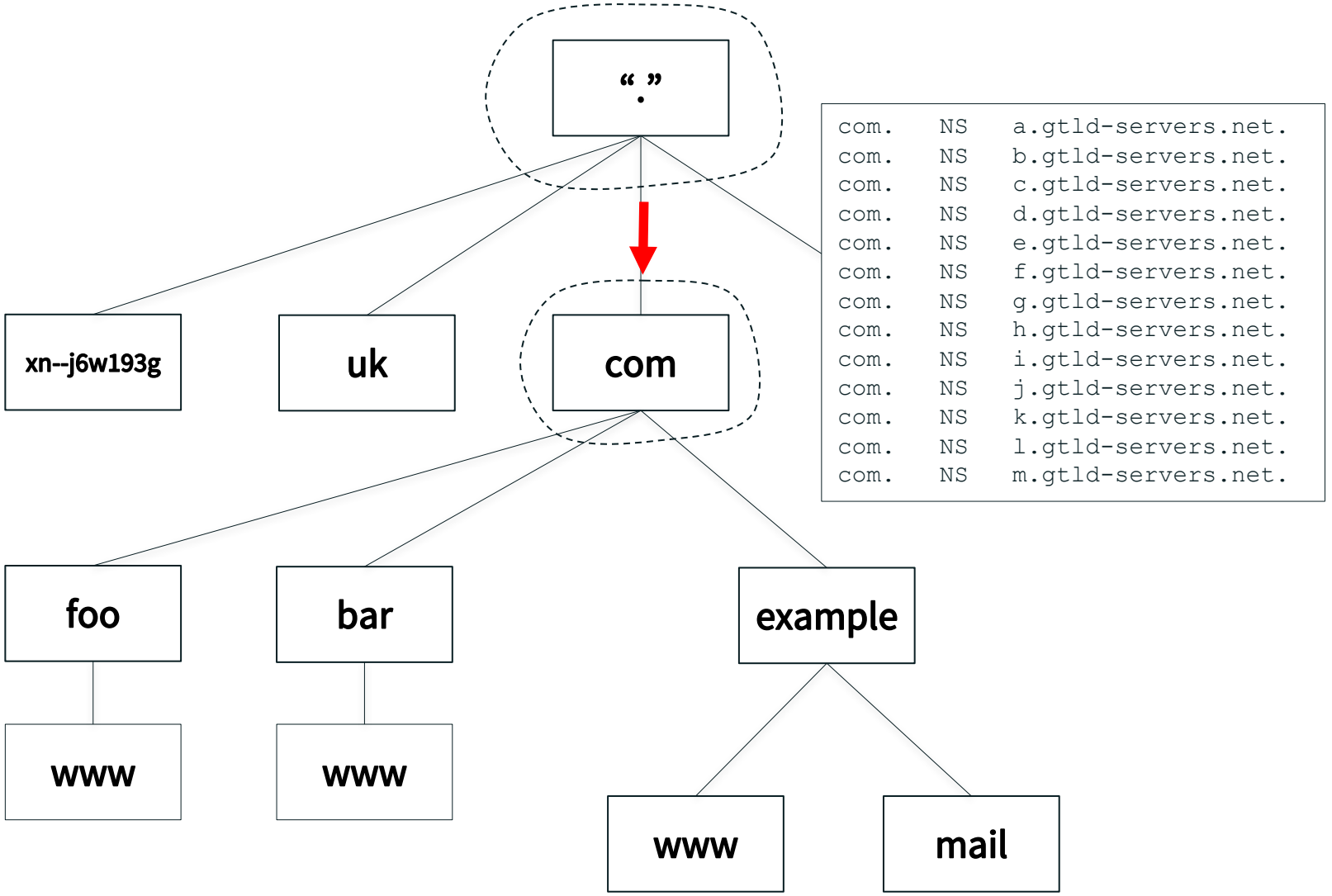
Name Server (NS)

- Specifies an authoritative name server for a zone
- The only record type to appear in two places
  - “Parent” and “child” zones

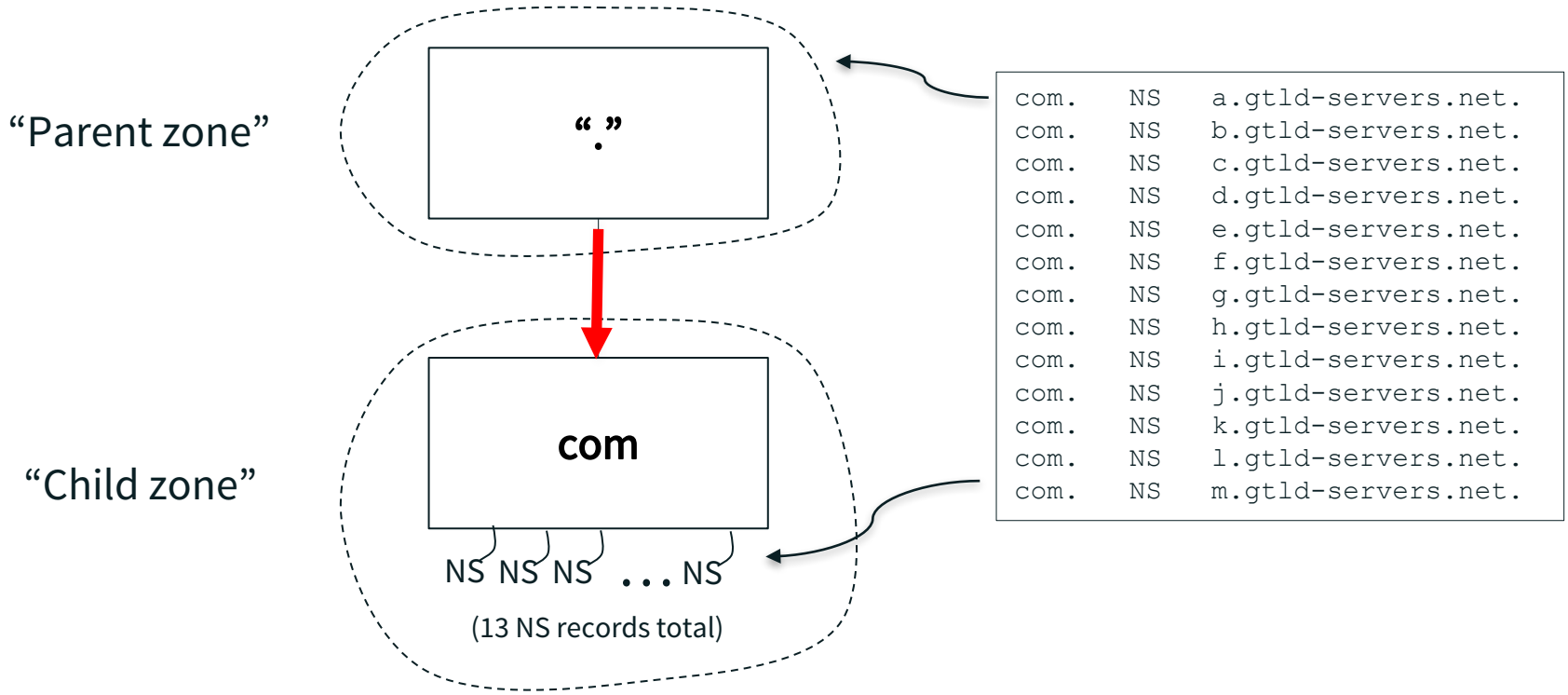
```
example.com.    NS    ns1.example.com.
example.com.    NS    ns2.example.com.
```

- Left hand side is the name of a zone
- Right hand side is the name of a name server
  - Not an IP address!

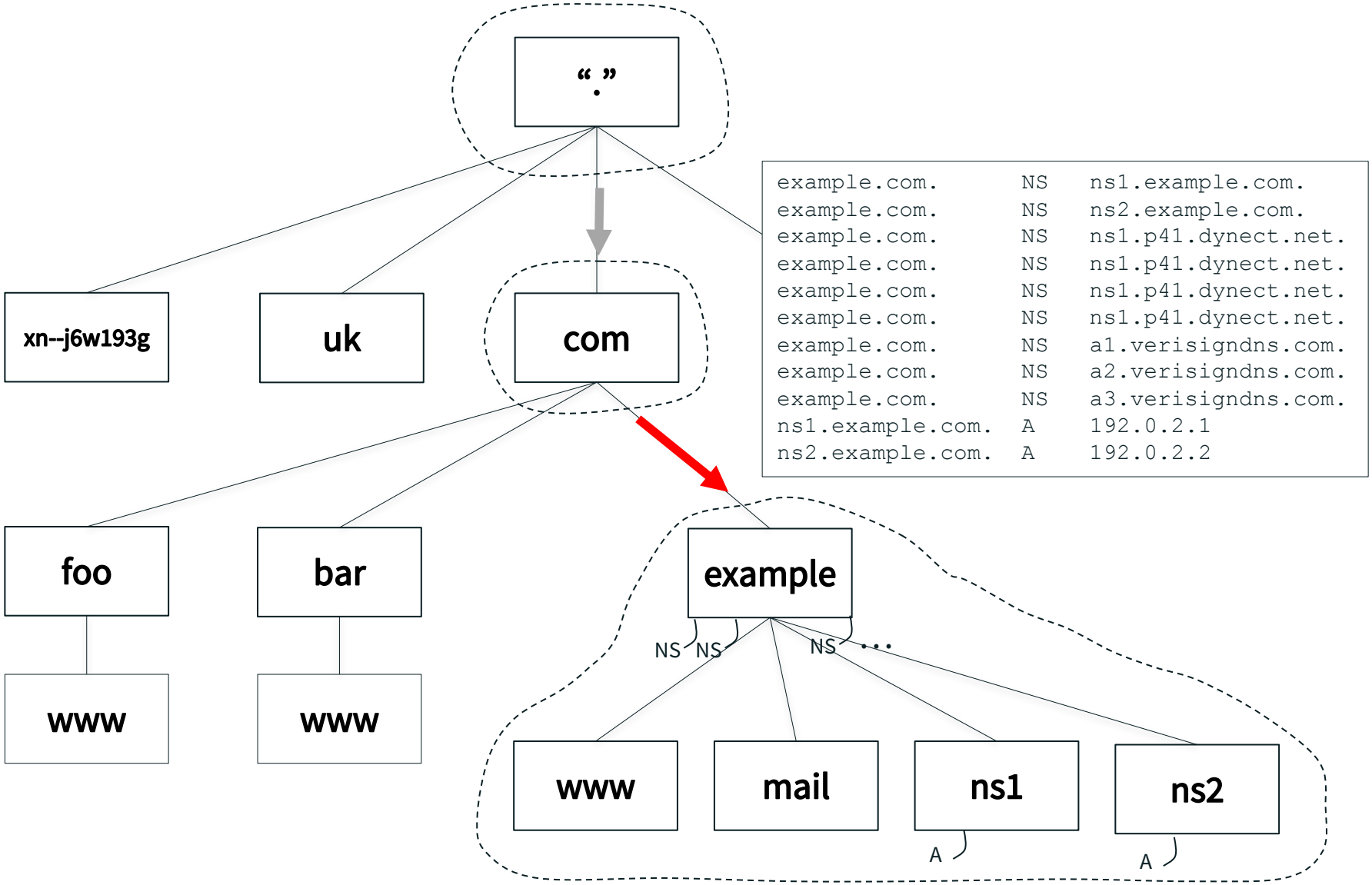
NS Records Mark Delegations



NS Records Appear in Two Places



More Delegation, Including Glue



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.**

  - Also for breaking for complicated dependencies not described here

Sample Zone File: *example.com*

```
example.com.        SOA   ns1.example.com. hostmaster.example.com. (
                           2016050100 ; serial
                           3600       ; refresh (1 hour)
                           600        ; retry (10 minutes)
                           2592000    ; expire (4 weeks 2 days)
                           300 )      ; minimum (5 minutes)
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
```

The Resolution Process

- Stub resolvers, recursive name servers and authoritative name servers cooperate to look up DNS data in the name space
- A DNS query always comprises three parameters:
  - Domain name, class, type
  - E.g., *www.example.com*, IN, A
- Two kinds of queries:
  - Stub resolvers send ***recursive queries***
    - "I need the complete answer or an error."
  - Recursive name servers send ***non-recursive*** or ***iterative queries***
    - "I can do some of the lookup work myself and will accept a ***referral***."

The Resolution Process

- High-level algorithm for processing a query:
  - Answer exact match from local data (authoritative or cache), if possible
  - If no exact answer possible, walk up the name space tree in local data from the queried name to find the best match, the ***closest enclosing zone***
  - Is it a recursive query?
    - Send the query to a name server for the closest enclosing zone
    - Keep following referrals down the tree until the zone with the answer (which could be “doesn’t exist”)
  - Is it a non-recursive query?
    - Return a referral to the closest enclosing zone

The Resolution Process

- How do you start the resolution process if there's no local data?
  - Empty cache, or
  - Not authoritative for any zones
- No choice but to start at the root zone
  - The ***root name servers*** are the servers authoritative for the root zone
- How does a name server find the root name servers?
  - They must be configured
  - No way to discover them
- The ***root hints file*** contains the names and IP addresses of the root name servers
  - *http://www.internic.net/domain/named.root*

List of Root Name Servers and Root Hints File

```
.                       NS      a.root-servers.net.
.                       NS      b.root-servers.net.
.                       NS      c.root-servers.net.
.                       NS      d.root-servers.net.
.                       NS      e.root-servers.net.
.                       NS      f.root-servers.net.
.                       NS      g.root-servers.net.
.                       NS      h.root-servers.net.
.                       NS      i.root-servers.net.
.                       NS      j.root-servers.net.
.                       NS      k.root-servers.net.
.                       NS      l.root-servers.net.
.                       NS      m.root-servers.net.
a.root-servers.net.     A       198.41.0.4
b.root-servers.net.     A       192.228.79.201
c.root-servers.net.     A       192.33.4.12
d.root-servers.net.     A       199.7.91.13
e.root-servers.net.     A       192.203.230.10
f.root-servers.net.     A       192.5.5.241
g.root-servers.net.     A       192.112.36.4
h.root-servers.net.     A       198.97.190.53
i.root-servers.net.     A       192.36.148.17
j.root-servers.net.     A       192.58.128.30
k.root-servers.net.     A       193.0.14.129
l.root-servers.net.     A       199.7.83.42
m.root-servers.net.     A       202.12.27.33
a.root-servers.net.     AAAA    2001:503:ba3e::2:30
b.root-servers.net.     AAAA    2001:500:84::b
c.root-servers.net.     AAAA    2001:500:2::c
d.root-servers.net.     AAAA    2001:500:2d::d
e.root-servers.net.     AAAA    2001:500:a8::e
f.root-servers.net.     AAAA    2001:500:2f::f
h.root-servers.net.     AAAA    2001:500:1::53
i.root-servers.net.     AAAA    2001:7fe::53
j.root-servers.net.     AAAA    2001:503:c27::2:30
k.root-servers.net.     AAAA    2001:7fd::1
l.root-servers.net.     AAAA    2001:500:9f::42
m.root-servers.net.     AAAA    2001:dc3::35
```

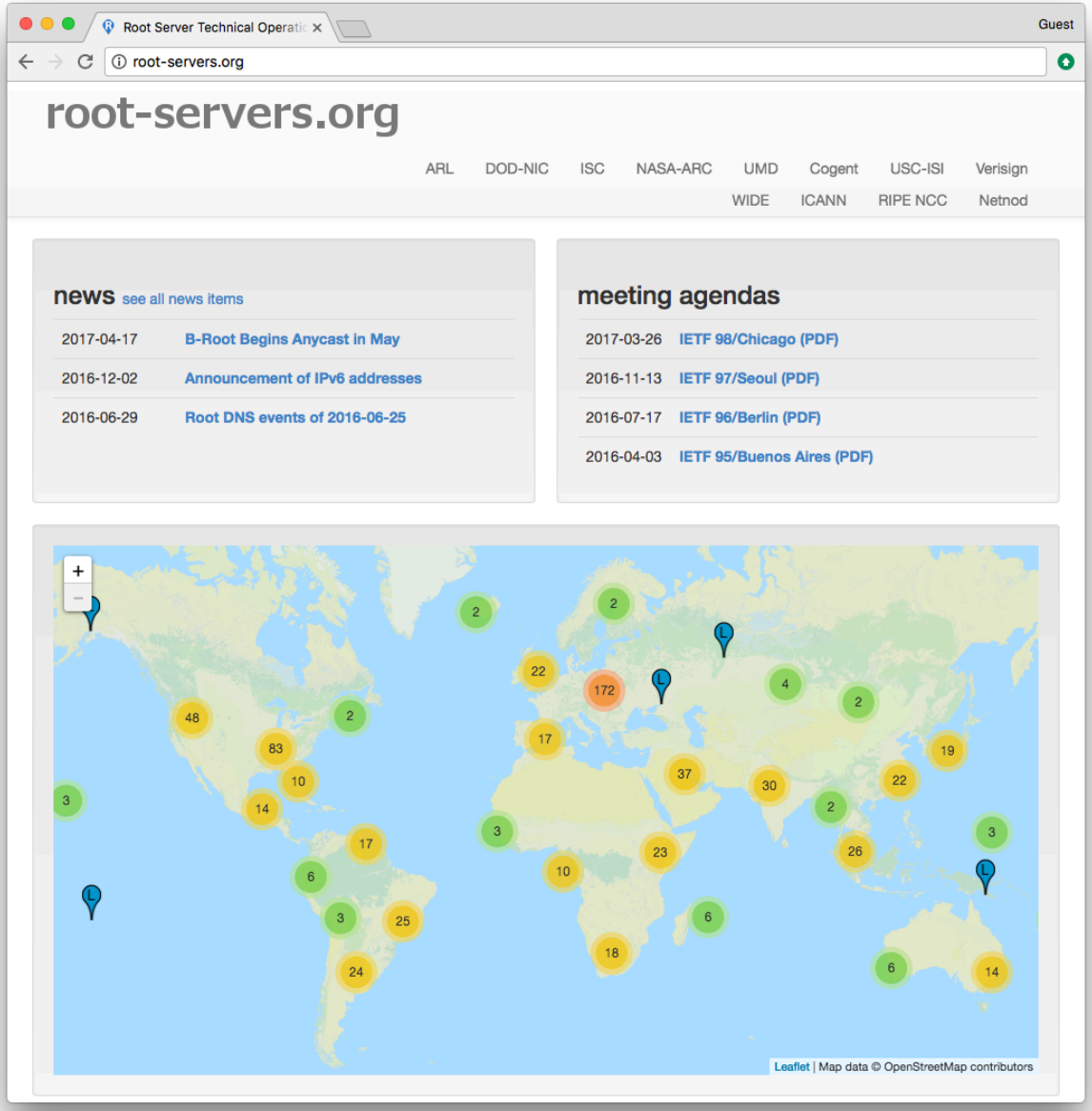
Root Zone Administration

- Administration of the root zone is complicated
- Two organizations cooperate to administer the zone's contents
  - Public Technical Identifiers (PTI), an ICANN affiliate, is the IANA Functions Operator
  - Verisign is the Root Zone Maintainer
- Twelve organizations operate authoritative name servers for the root zone

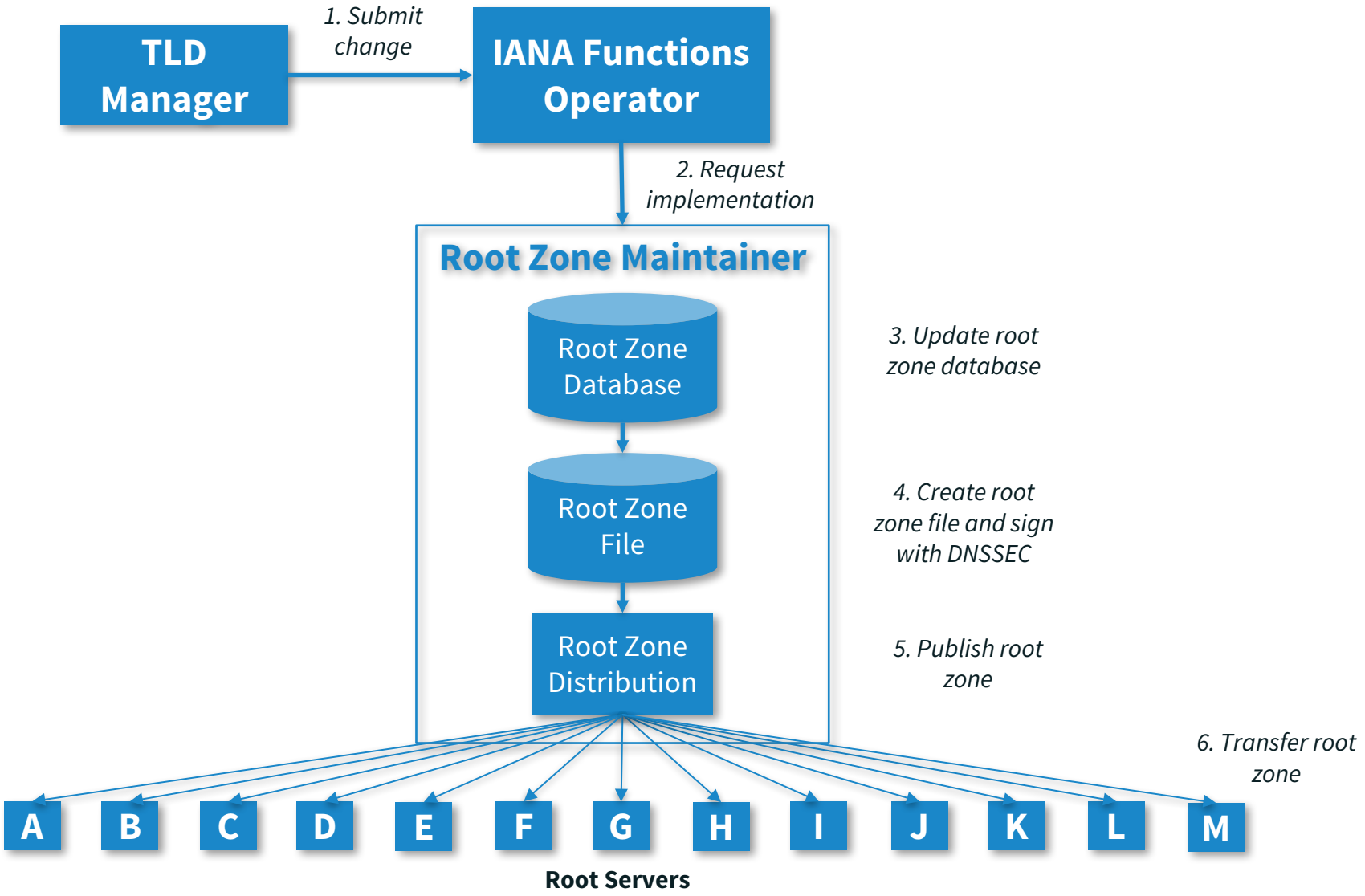
The Root Servers and Operators

- **A** Verisign
- **B** University of Southern California Information Sciences Institute
- **C** Cogent Communications, Inc.
- **D** University of Maryland
- **E** United States National Aeronautics and Space Administration (NASA) Ames Research Center
- **F** Information Systems Consortium (ISC)
- **G** United States Department of Defense (US DoD) Defense Information Systems Agency (DISA)
- **H** United States Army (Aberdeen Proving Ground)
- **I** Netnod Internet Exchange i Sverige
- **J** Verisign
- **K** Réseaux IP Européens Network Coordination Centre (RIPE NCC)
- **L** Internet Corporation For Assigned Names and Numbers (ICANN)
- **M** WIDE Project (Widely Integrated Distributed Environment)

The *root-servers.org* Web Site



Root Zone Change Process



Resolution Process

The phone is configured to send queries to the recursive name server with IP address 4.2.2.2

Recursive Name Server
4.2.2.2

4.2.2.2 is a recursive server run by Level 3 Communications

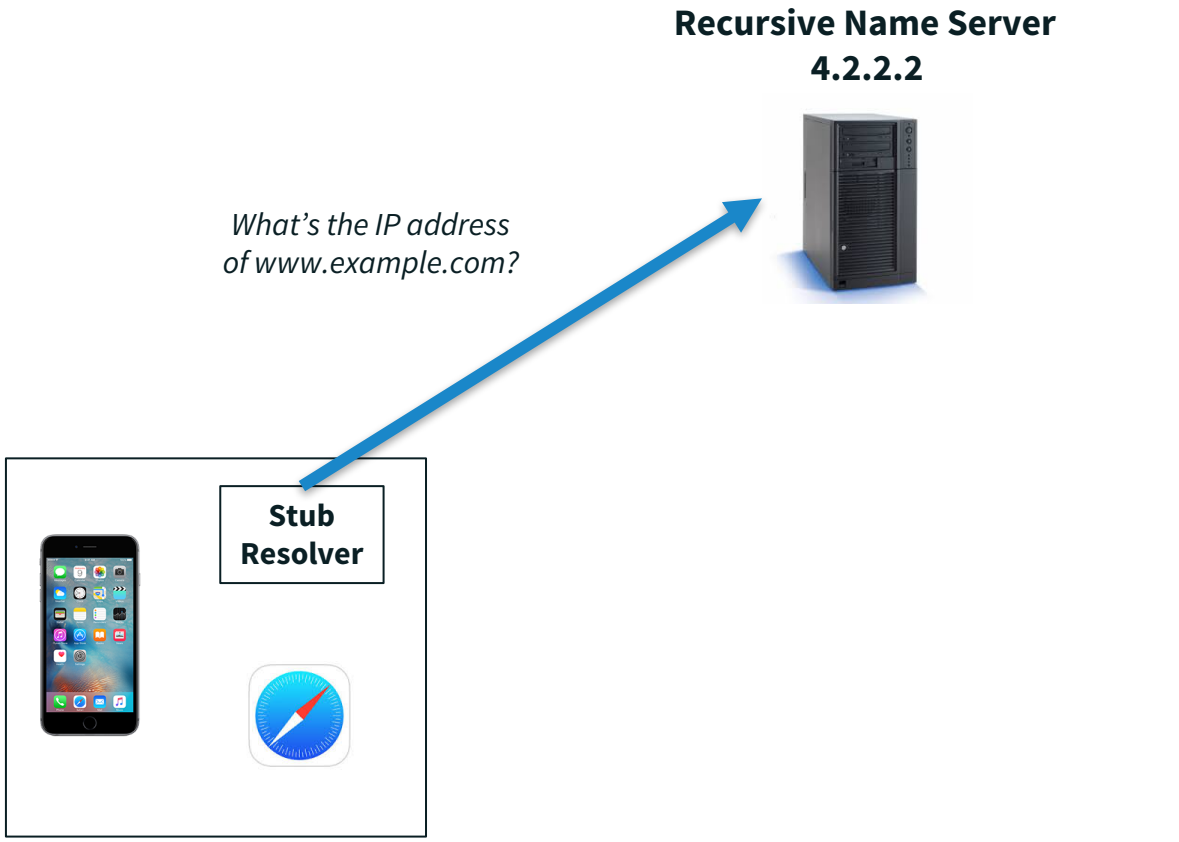
Resolution Process

A user types *www.example.com* into Safari on her phone
Safari calls the stub resolver function to resolve the name

Recursive Name Server
4.2.2.2

Resolution Process

The phone's stub resolver sends a query for *www.example.com*, IN, A to 4.2.2.2

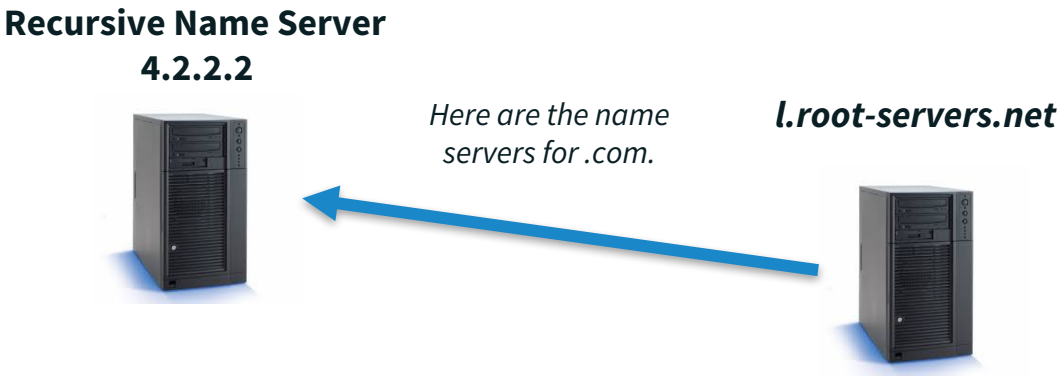


Resolution Process

Empty cache, so recursive server queries a root server

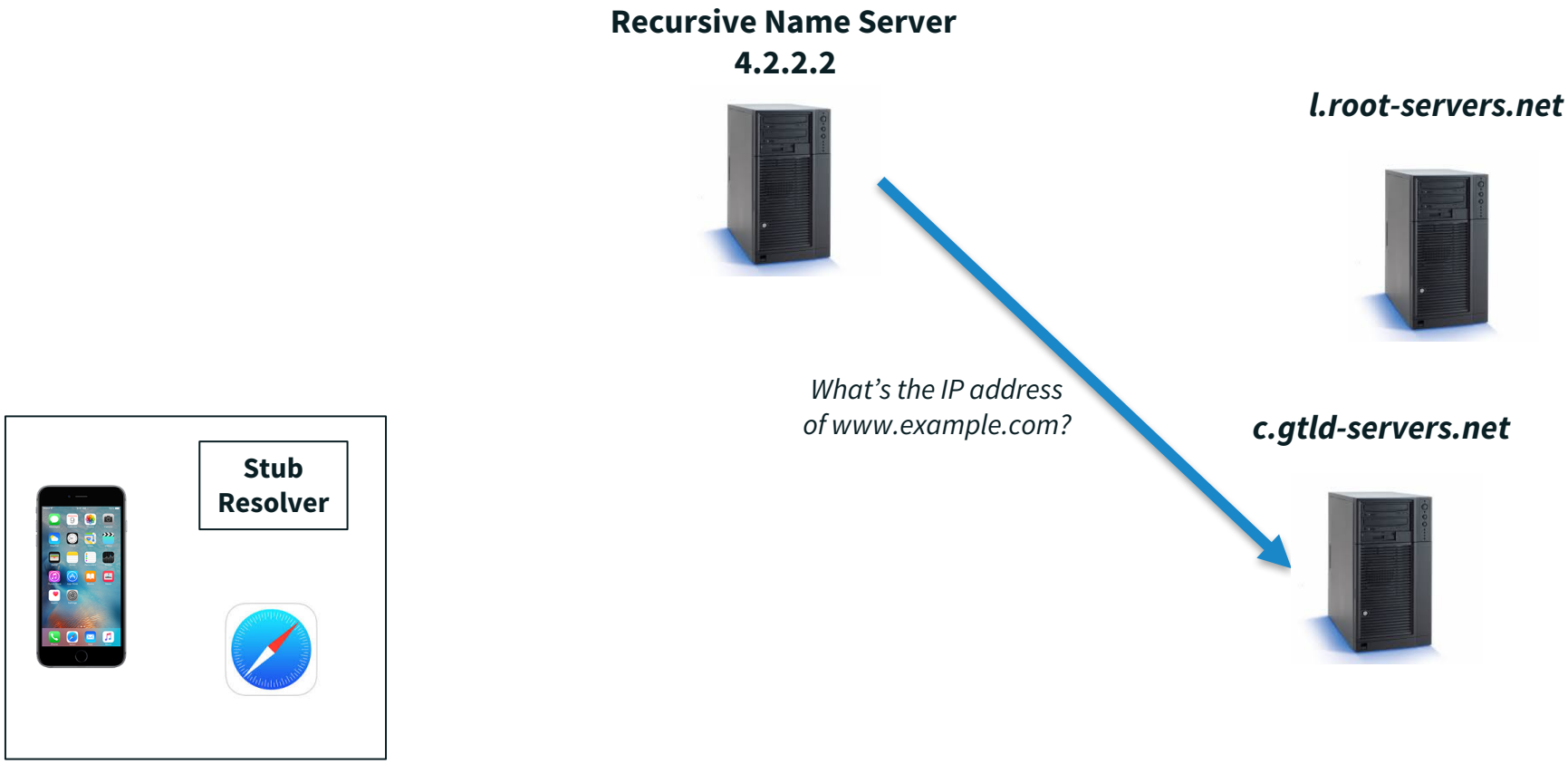
Resolution Process

Root server returns a referral to *.com*



Resolution Process

Recursive server queries a *.com* server

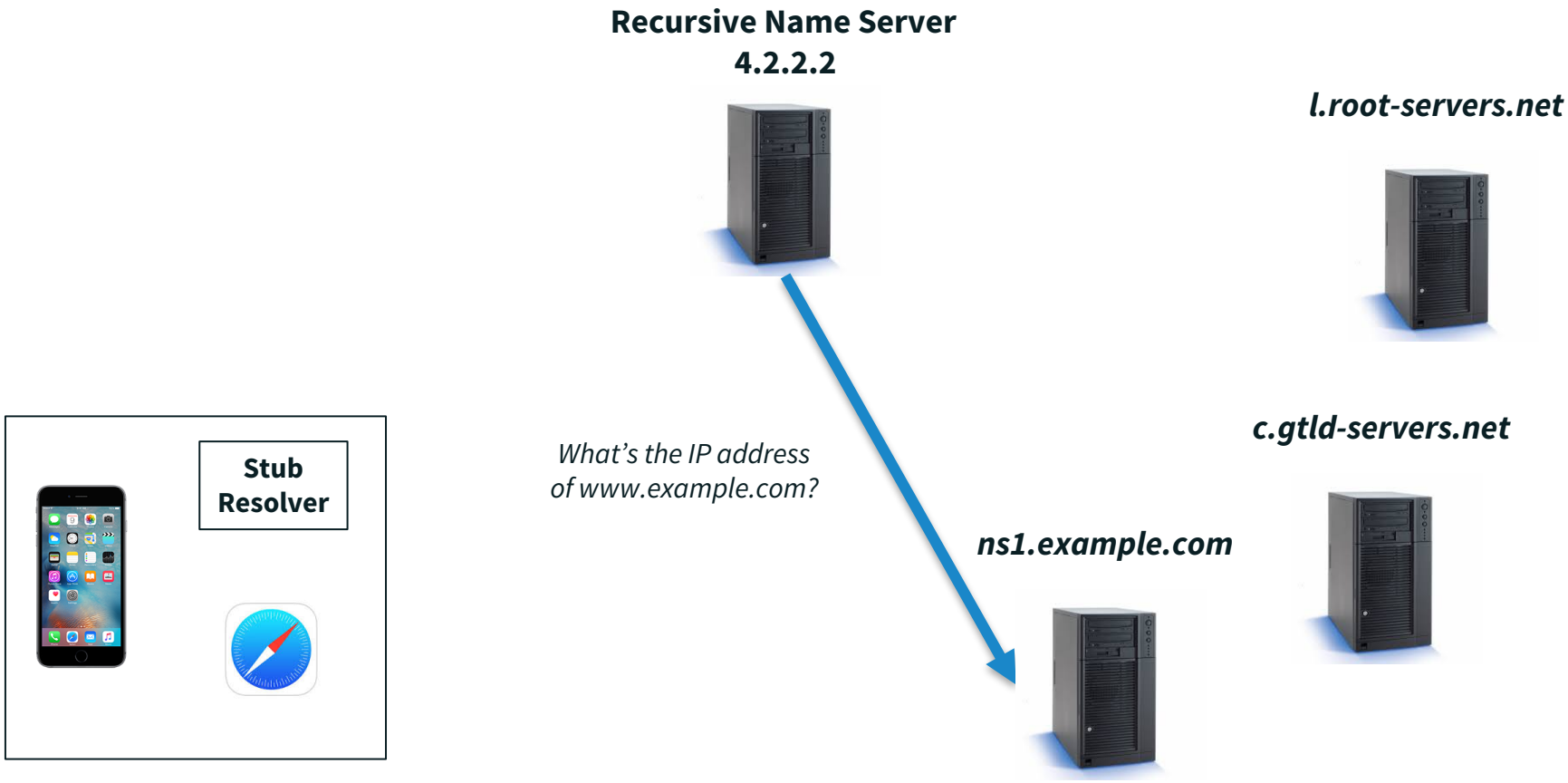


Resolution Process

.com server returns a referral to *example.com*

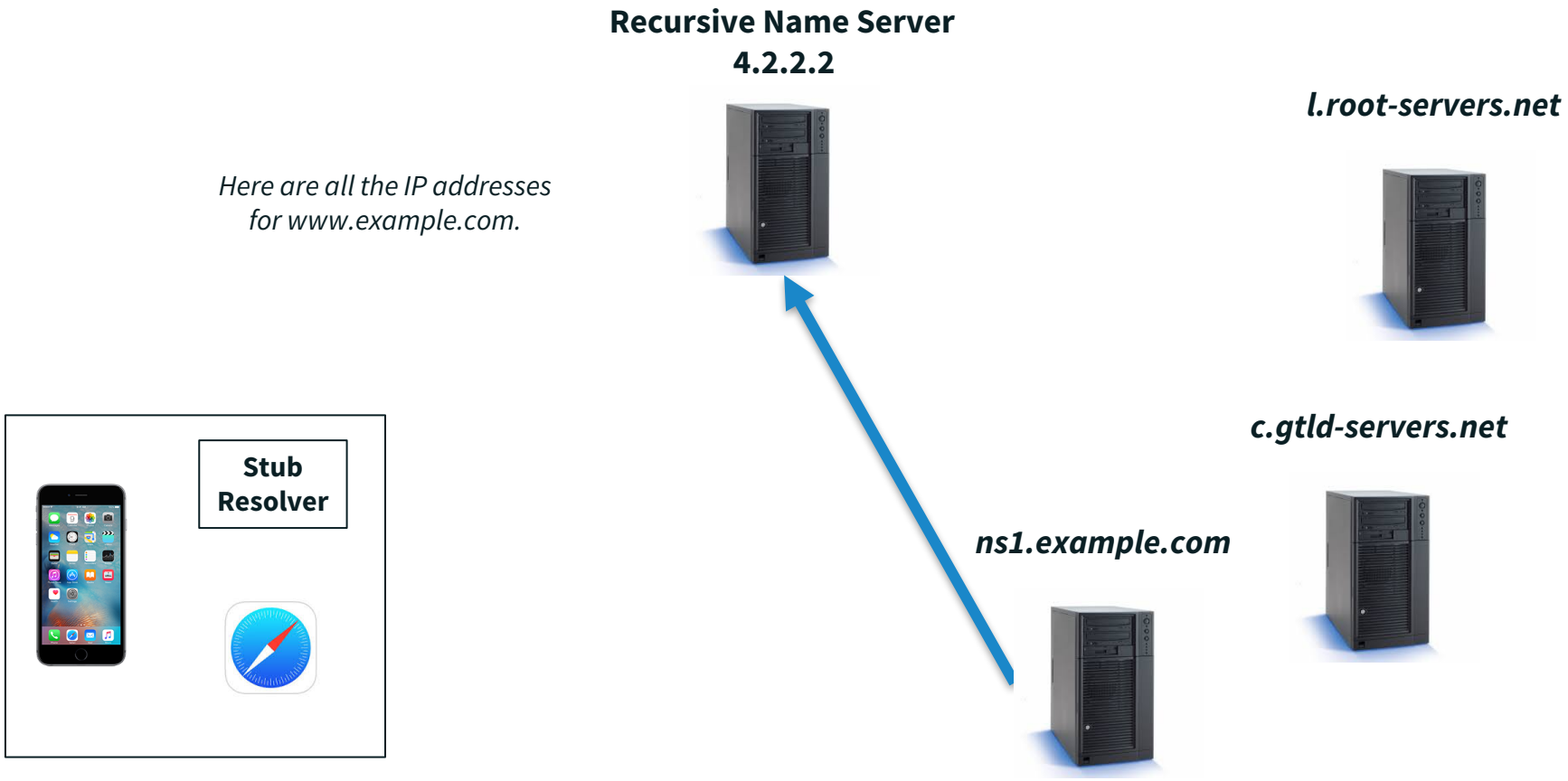
Resolution Process

Recursive server queries an *example.com* server



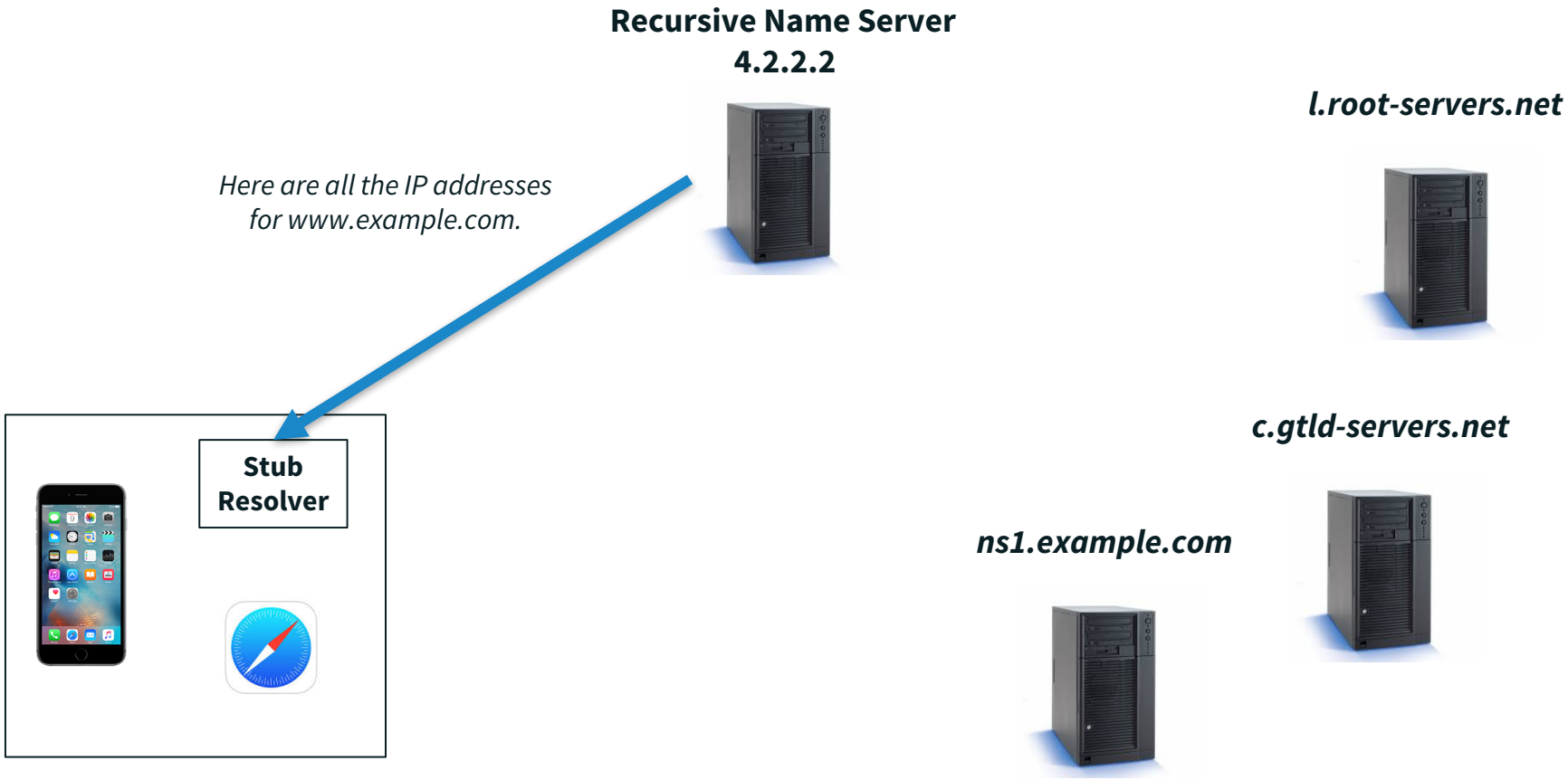
Resolution Process

example.com server returns the answer to the query



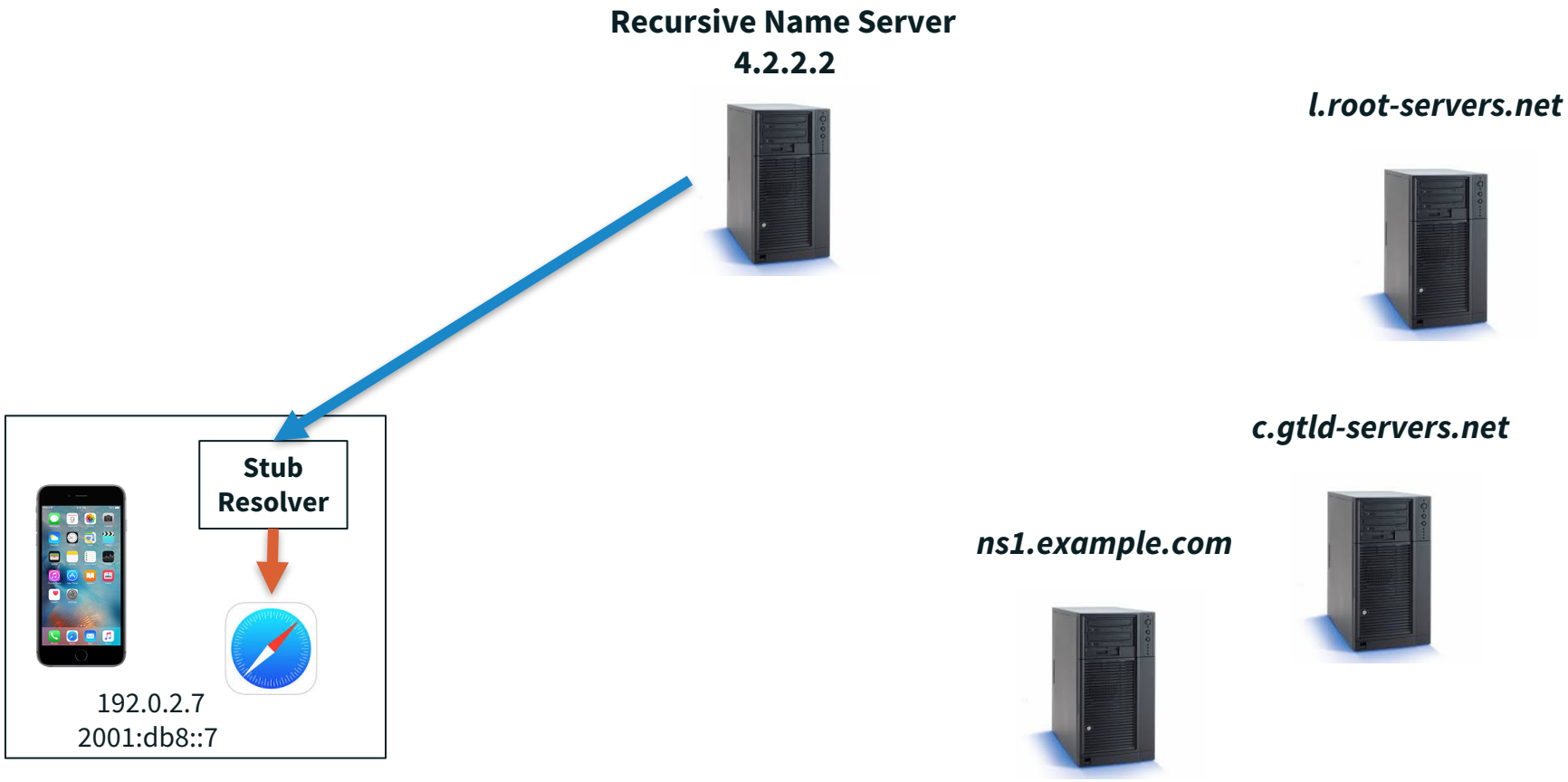
Resolution Process

Recursive server returns the answer to the query to the stub resolver



Resolution Process

Stub resolver returns the IP addresses to Safari



Caching

- Caching speeds up the resolution process
- After the previous query, the recursive server at 4.2.2.2 now knows:
  - Names and IP addresses of the *.com* servers
  - Names and IP addresses of the *example.com* servers
  - IP addresses for *www.example.com*

- Let's look at another query following immediately the first

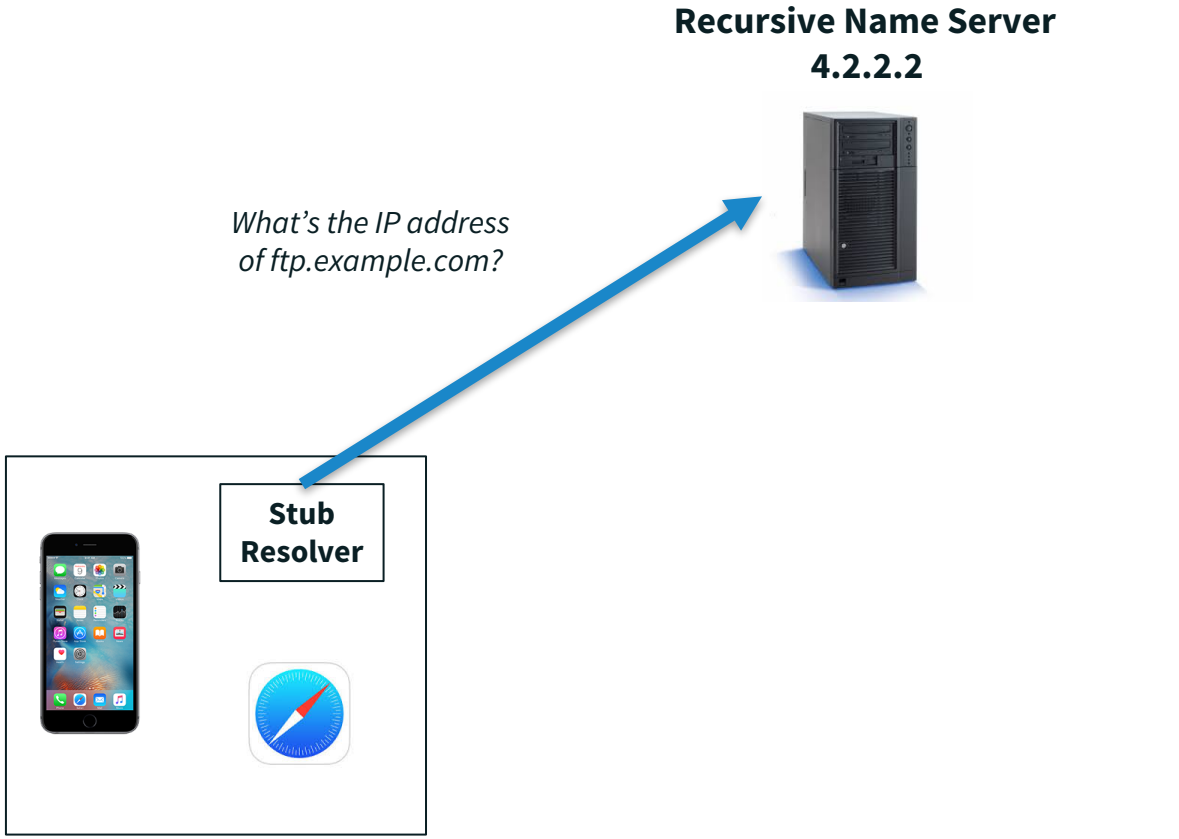
Resolution Process

A user types *ftp.example.com* into Safari on her phone
Safari calls the stub resolver function to resolve the name

Recursive Name Server
4.2.2.2

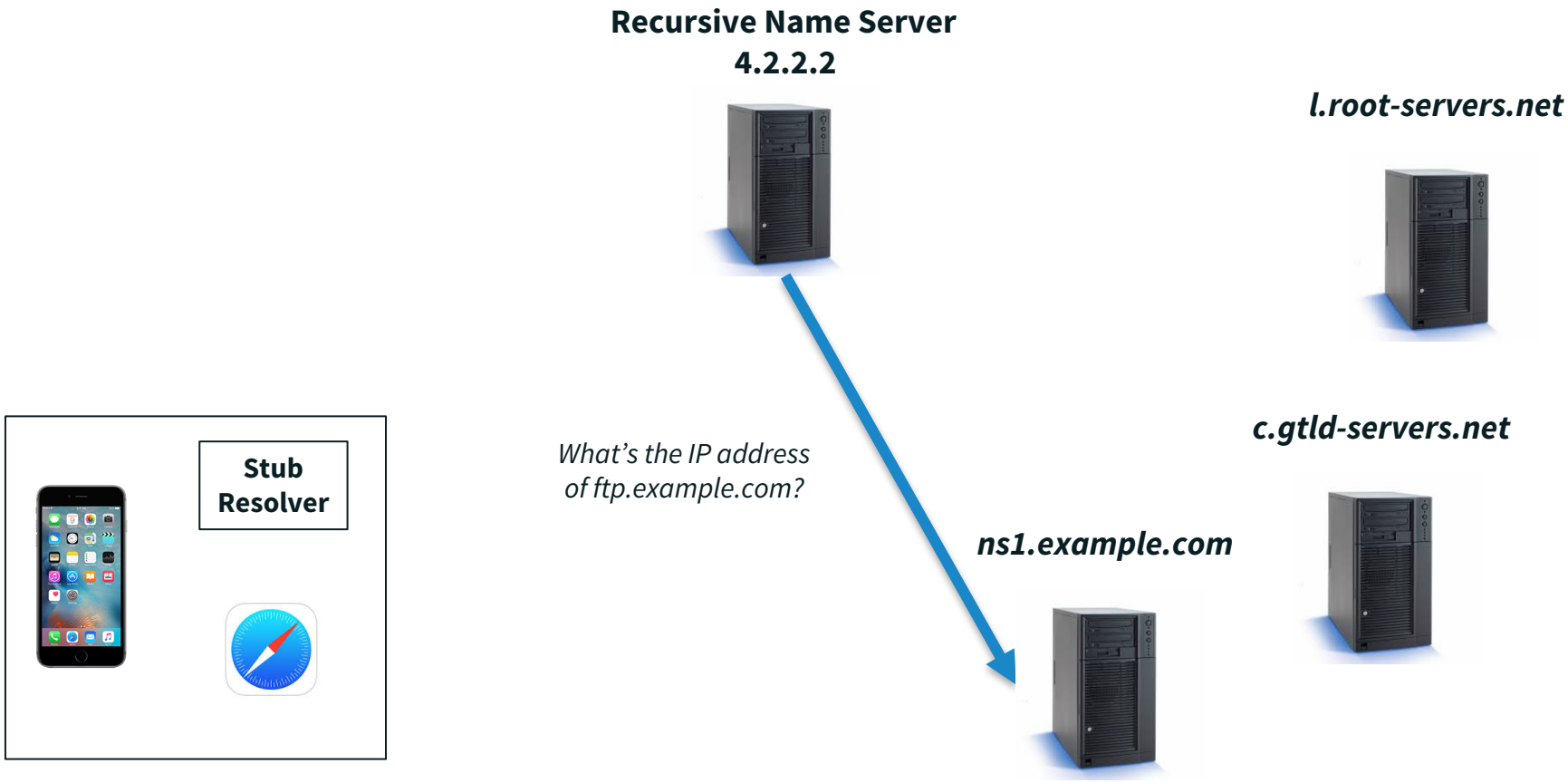
Resolution Process

The phone's stub resolver sends a query for *ftp.example.com*/IN/A to 4.2.2.2



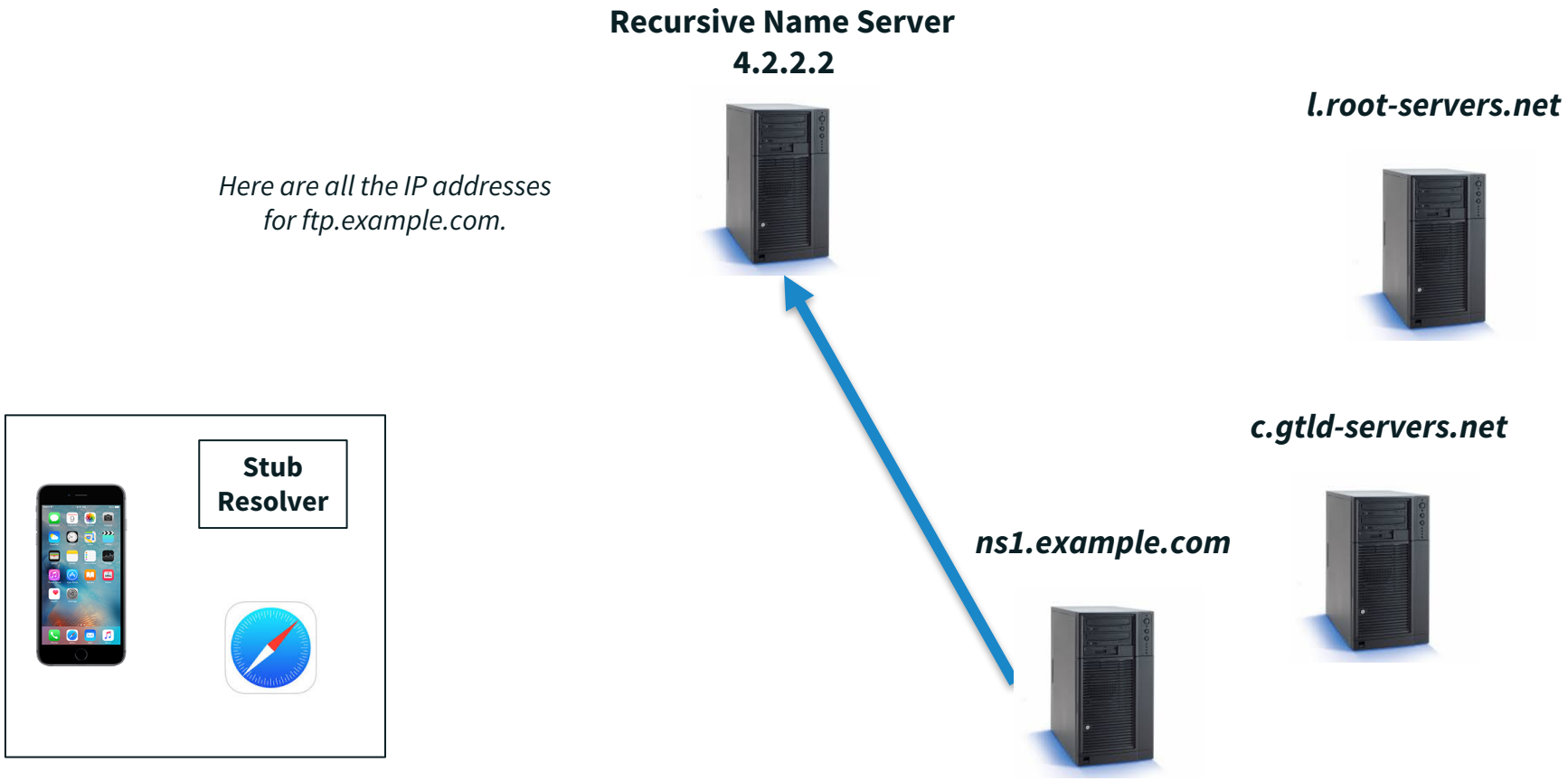
Resolution Process

Recursive server queries an *example.com* server



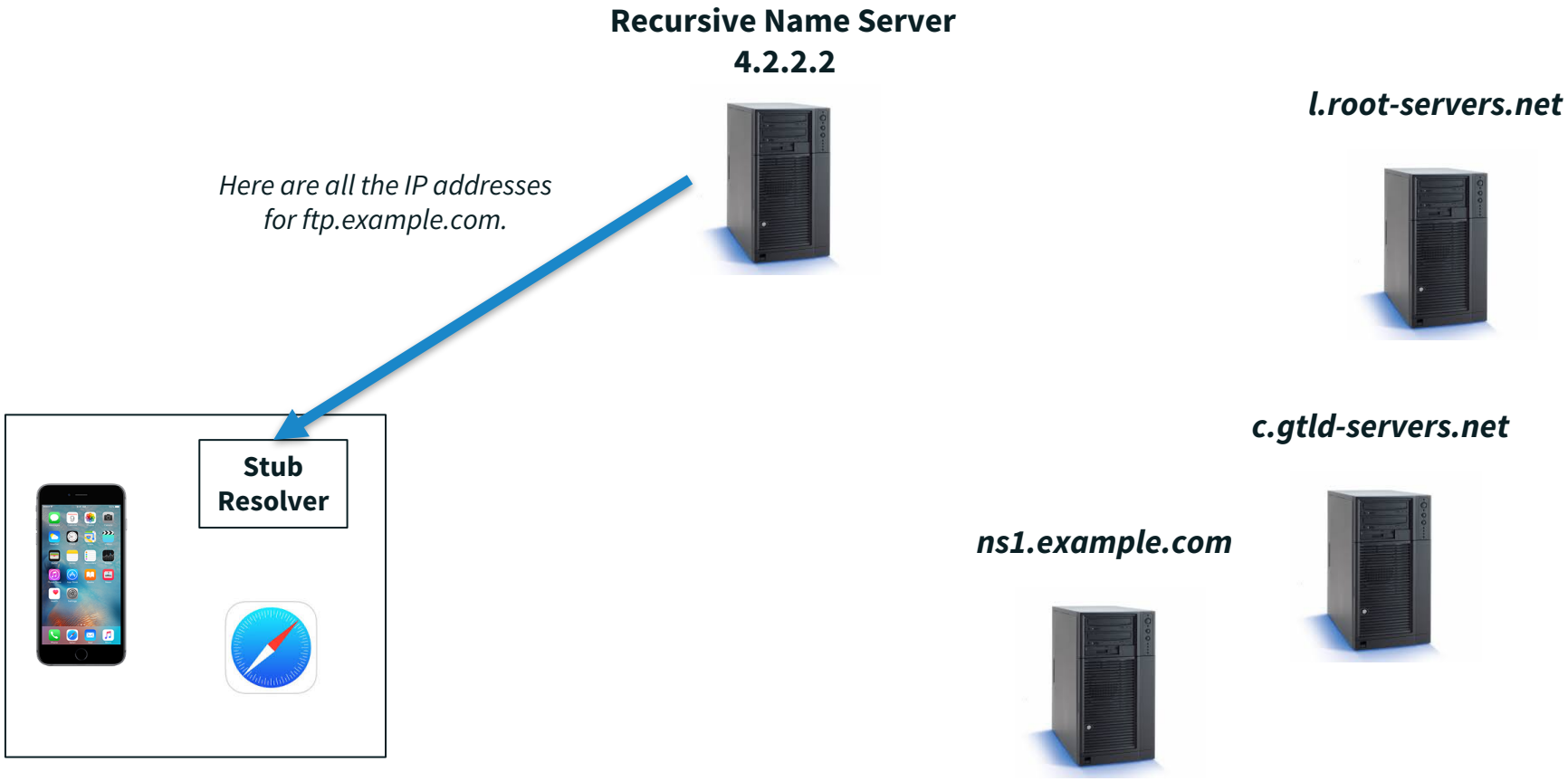
Resolution Process

example.com server returns the answer to the query



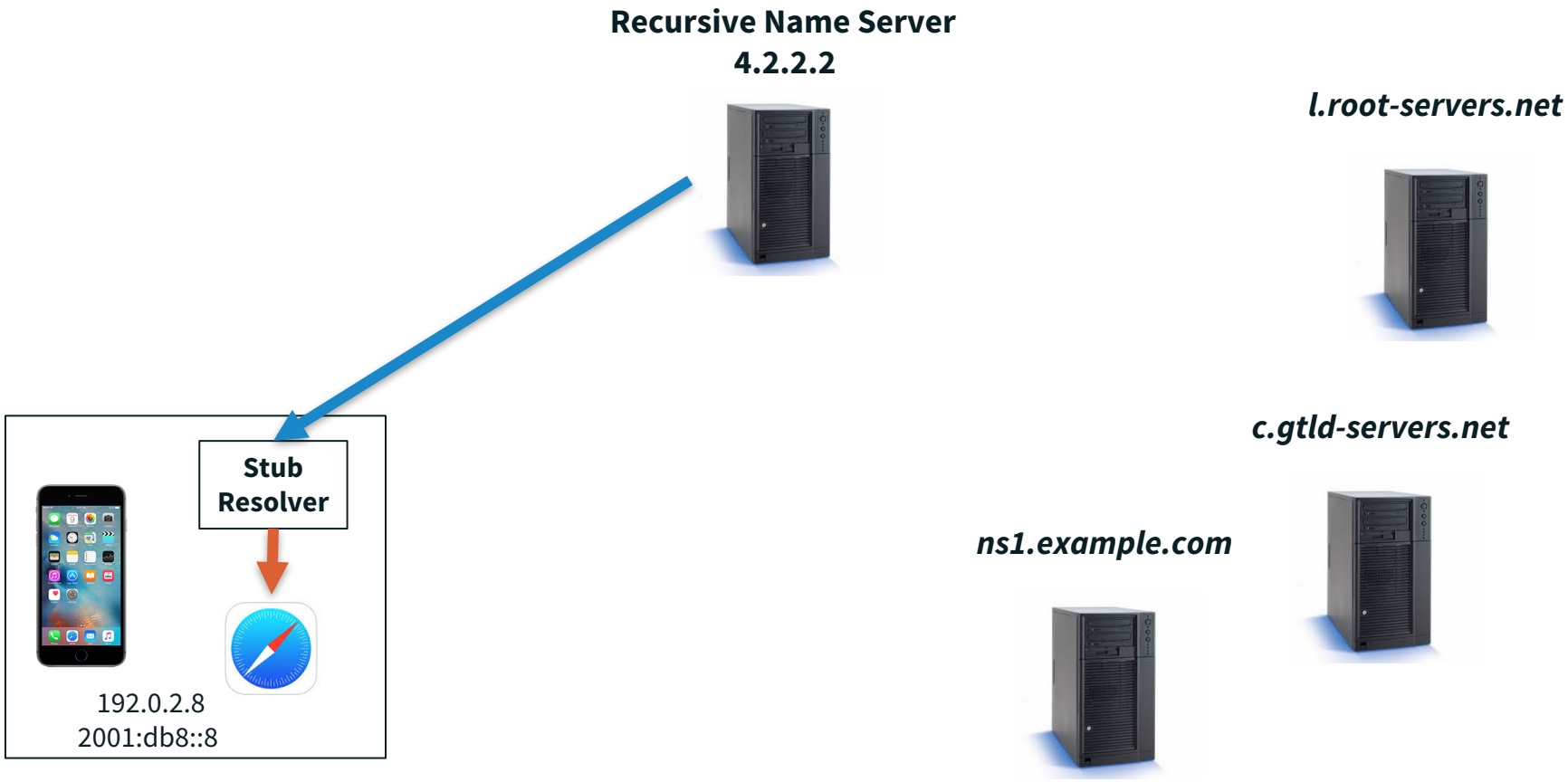
Resolution Process

Recursive server returns the answer to the query to the stub resolver



Resolution Process

Stub resolver returns the IP addresses to Safari



Engage with ICANN – http://www.icann.org

Thank You and Questions

Reach us at:
Email: matt.larson@icann.org

twitter.com/icann

gplus.to/icann

facebook.com/icannorg

weibo.com/ICANNorg

linkedin.com/company/icann

flickr.com/photos/icann

youtube.com/user/icannnews

slideshare.net/icannpresentations