



# Demystifying DNS

**Allan Hurst**

Partner, Director of Enterprise Strategy: *Keep IT Simple*

allanh@kiscc.com

<http://www.kiscc.com>

Version 4.5– 2015-09-02

Smartphones, dumbphones, pagers (?), et al...set them all to “stun”, please. **No noise is good noise.**

If you have a question, **it’s absolutely OK to ask.** It’ll help if you raise your hand first to get my attention. I’ll try to answer on the fly.

It’s OK to **have fun** in here. Honest.





# Who is this guy, anyway?

## Allan Hurst

Works for KIS (“Keep IT Simple”)

Partner & Technical Principal, Director of Enterprise Strategy

One of five partners at KIS, a VMware/Microsoft/Novell partner in Fremont CA, Kansas City MO, Denver, and Cleveland.

Runs the Enterprise Systems Group (network planning, migrations, upgrades, moves, re-architecting, and clean-up).

Runs “The WAP Squad”. (“WAP” stands for...)



## Who are you?

Network administrator and/or manager, IT executive, or perhaps a security or law enforcement professional.

You use DNS, but don't really understand it. (Maybe you even have to troubleshoot it from time to time, which is really icky.)

You might be seeing network issues, such as:

- Sporadic errors when browsing the web or local network
- Intermittent server communication





## Where did this session come from?

A very small amount of this material originally appeared in a Novell user group session about server migrations.

*After* the session, many attendees came up to ask me privately about DNS.

Most of them were embarrassed to publicly ask about the basics of DNS.

It's OK for you to ask anything about DNS that you wish. In the worst case: I might say, "I don't know."



(I may not always have the answer, but this is how all of my sessions get revised to better meet your needs.)



## Why should I care about DNS?

Familiarity with Domain Name Services (DNS) is now a requirement for administering networks.

If you also need to troubleshoot or analyze networks, and you don't understand DNS, things are going to be pretty tough.

Nearly every contemporary network product assumes and requires solid, working DNS.

This includes Windows and Active Directory, VMware, most databases such as SQL Server and Oracle, web application platforms...the list goes on and on.



## About This Session

- Absolute vanilla basics of DNS (IPv4 only)
- We'll touch on DNS & Active Directory.
- We'll touch on DNS troubleshooting briefly.

(For a detailed session on troubleshooting, ask me back to present “Demystifying *More* DNS”.)






# The Basics of DNS

Stands for “**D**omain **N**ame **S**ystem”.

It's just a database to match names with IP addresses ...

**www.fubar.com**  **158.62.23.47**

The diagram shows the text "www.fubar.com" on the left and the IP address "158.62.23.47" on the right. Between them are two light blue arrows: the top one points from the domain name to the IP address, and the bottom one points from the IP address back to the domain name, indicating a bidirectional relationship.

... and match IP addresses with names. (“Reverse” DNS.)

Think of it as a phone book (and reverse number directory) for the Internet.



## Who are ICANN and IANA?

**ICANN:** Internet Corporation for Assigned Numbers and Names.

A nonprofit organization which assumed responsibility for DNS from the U.S. Government in 1988.

ICANN is the authority for the DNS network “root.”

<http://www.icann.org>

ICANN manages the Internet Assigned Numbers Authority (IANA), who in turn manages the DNS root.

<http://www.iana.org>

## ignatz.fubar.com



Consists of a Top Level Domain (TLD) and at least one second level domain

The leftmost word is called a “host name”

Can consist of a 3rd, 4th, 5th, etc. level ... all the way to 127 levels. (bryant.sousaphone.ignatz.fubar.com)



# What are “Top Level Domains”?

DNS is divided into Top Level Domains (TLDs).

The original TLDs were:

**.com** ... Commercial Enterprises

**.org** ... Noncommercial Organizations

**.net** ... Private Networks

**.edu** ... Educational Institutions

**.mil** ... Military Installations

**.gov** ... Government Installations

... and Country Codes (such as .us, .ca, .uk, etc.)

Many new TLDs have been added since then, such as:

.info, .tv, .work, and .to



## What are “Top Level Domains”?

Originally, there was no way for someone to create a new TLD other than petitioning (begging) ICANN, which had placed strict limits on the creation of Generic Top Level Domains (“gTLDs”)

However, in early 2012, ICANN began accepting applications for new gTLDs.

It’s not cheap; for just \$185,000, any company or organization can apply for their own gTLD, with only a \$25,000/year renewal fee.



## What are “Top Level Domains”?

There are several types of TLDs:

### **The original TLDs:**

.com, .net, .edu, .gov, etc.

### **Country Code TLDs (“ccTLDs”):**

.us, .uk, .to, .au, etc.

### **Geographic TLDs (“geoTLDs”):**

.london, .asia, .quebec, etc.

*The above information is nice to know when tracing email paths and external network traffic.*

## Where are domain names stored?

Public domain names are stored in a huge distributed database.



Each domain name is stored in something called a “zone file”, which lives on a DNS server.  
(We’ll talk more about zone files later.)

## Who controls the DNS databases?



Each TLD is stored in a separate database.

Each database is controlled by one entity.

An entity can be a private company or a government.

Governments generally control “country-code” TLDs.

ICANN controls issuance and admin of generic TLDs.



## What is the DNS Network Root?

The **DNS Network Root** is a “list of lists”.

It's the database showing who controls which Top Level Domain.

It also shows where the machines hosting each TLD database are located.

For a current list of TLD servers, go to:

<http://www.root-servers.org>

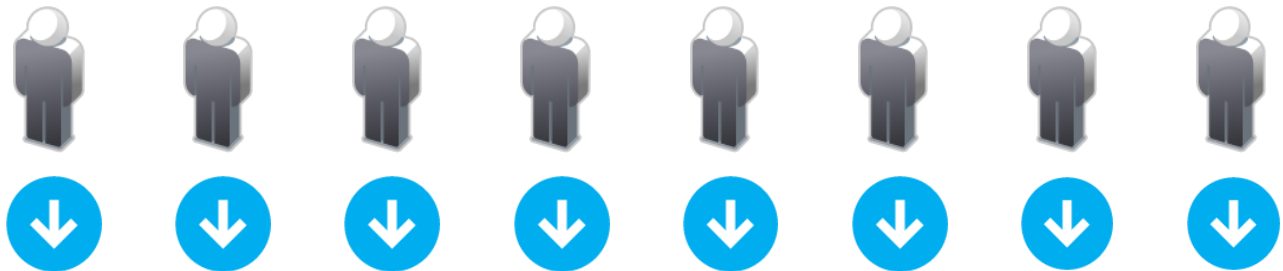
# Where are the DNS Root Servers?



(From <http://www.root-servers.org>)

# How are DNS domains created?

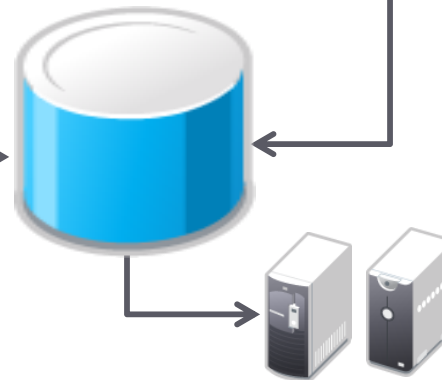
1. Users purchase domains from Registrars



2. Registrars send changes to the TLD



3. The TLD points to each domain's name servers



4. Name servers answer DNS queries for their assigned domains

# How Do I Get My Own Domain?

1. Find a registrar and register the domain with them.
2. Decide where to host your new domain:
  - Your Registrar (many offer DNS hosting for free)
  - A Third-Party DNS Hosting Company
  - Your ISP (if you have a good relationship with them)
  - Your Web Hosting Company (not recommended)
  - On servers within your company (not recommended)
  - On servers within another company (not recommended)
3. Set up the DNS zone on your provider of choice.





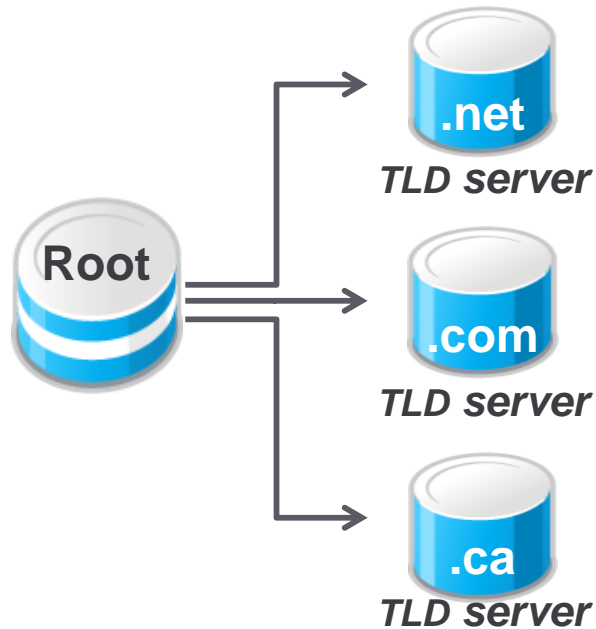
## Where are the DNS records kept?

- Root Servers (keep track of TLD servers)



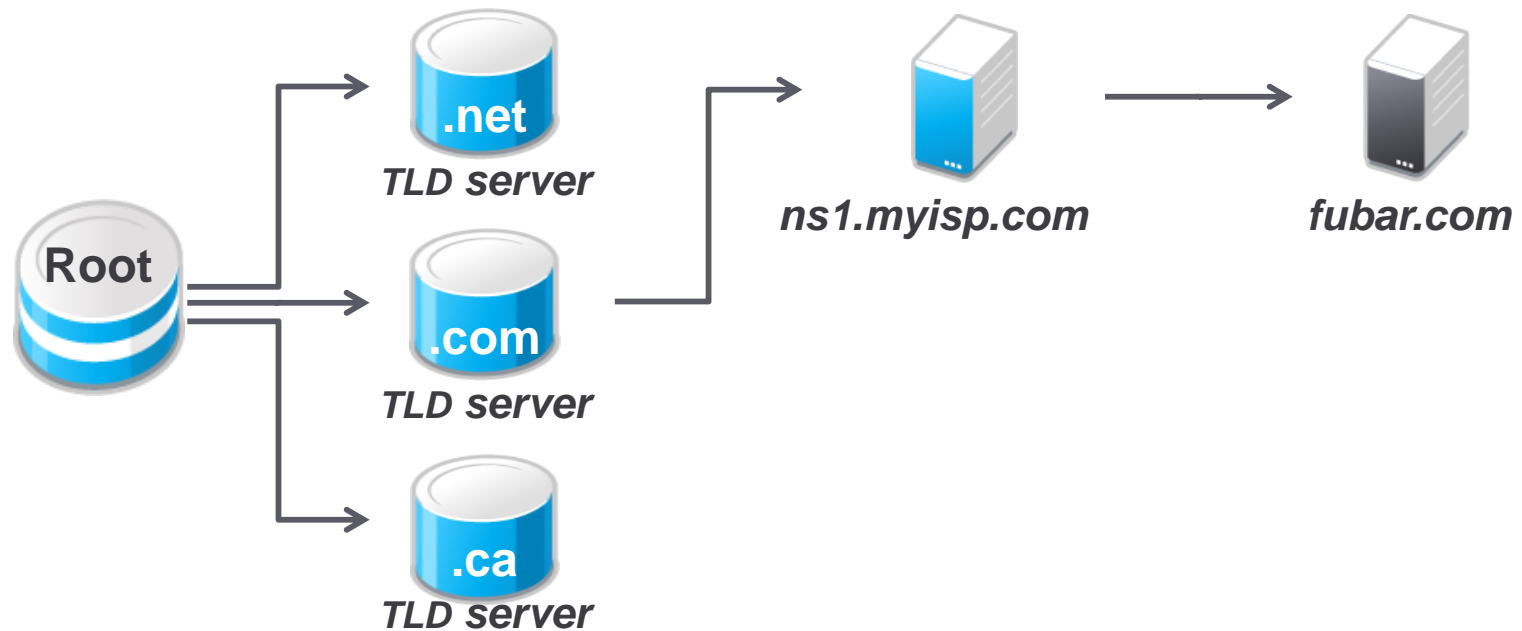
## Where are the DNS records kept?

- Root Servers
- Top Level Domain Servers (keep track of domain NS)



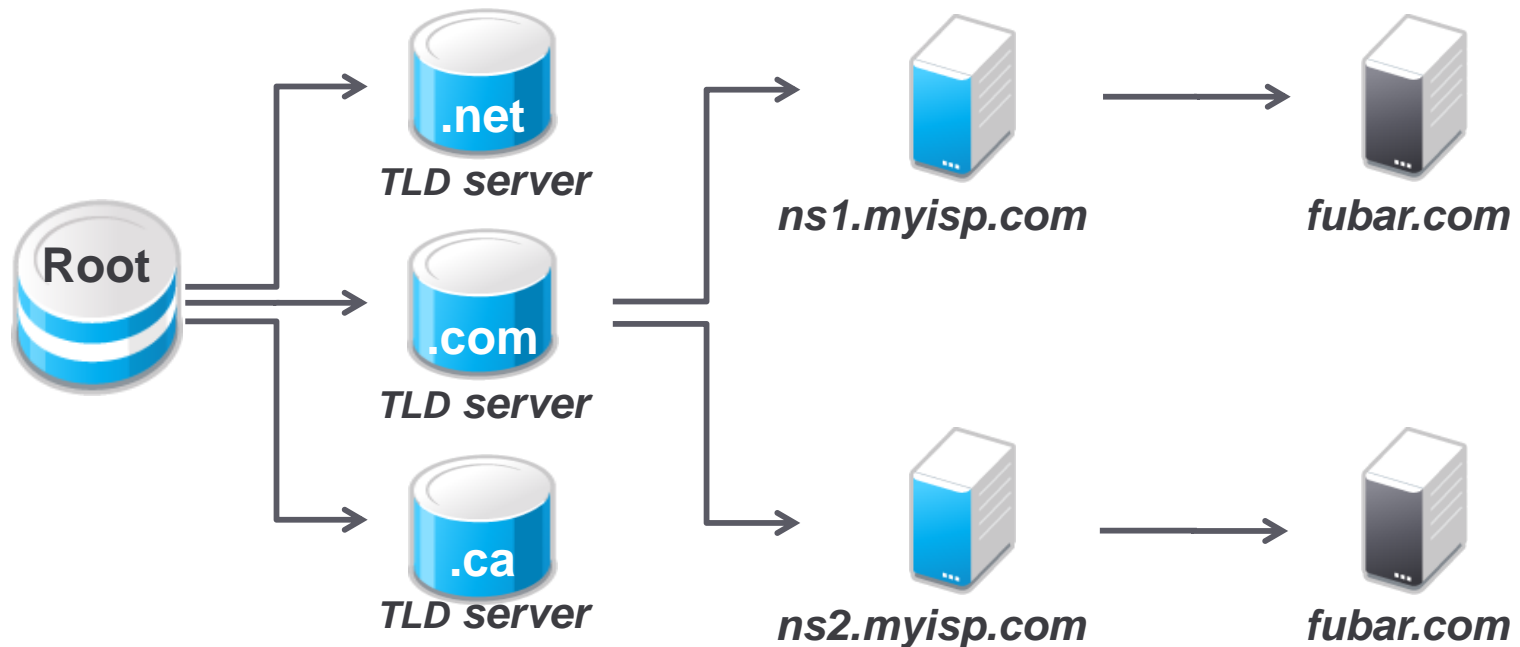
## Where are the DNS records kept?

- Root Servers
- Top Level Domain Servers (keep track of domain NS)
- Name Servers (hold contents of individual domains)



# Where are the DNS records kept?

- Root Servers
- Top Level Domain Servers (keep track of domain NS)
- Name Servers
- Backup Name Servers (same as name servers)



# How do the servers find each other?

There's a simple hierarchy:

**Root Servers** know only where the **TLD Servers** are.

**TLD Servers** know only which **name servers** hold the definitive records for each domain.

**Name Servers** know about IP addresses, hostnames, & sub-domains for the domains for which they're responsible.

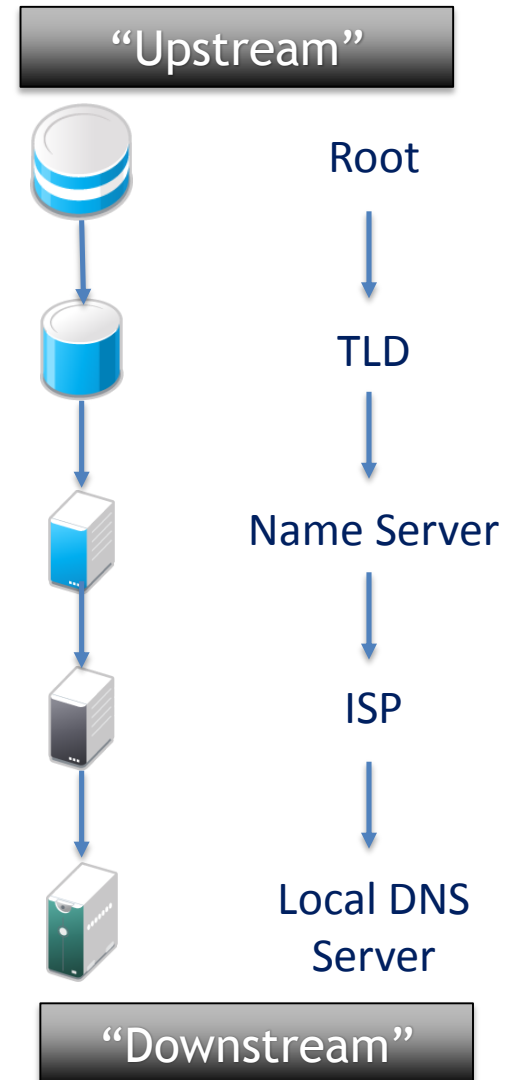


# So who knows what about whom?

All DNS “walks the tree”.

Upstream DNS servers “feed” information to downstream servers.

Upstream servers are sometimes called “resolvers”.



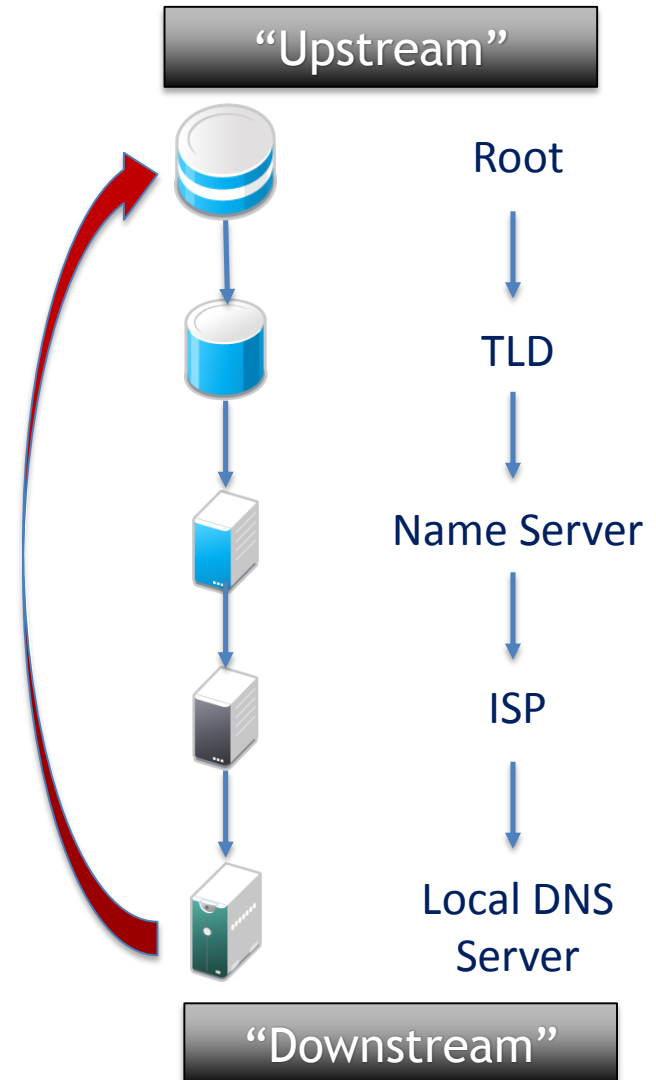
# So who knows what about whom?

All DNS “walks the tree”.

Upstream DNS servers “feed” information to downstream servers.

Upstream servers are sometimes called “resolvers”.

If there’s no response from an upstream server, fault tolerance is provided by jumping “up” to a root server, then walking “down” the tree to get an answer.

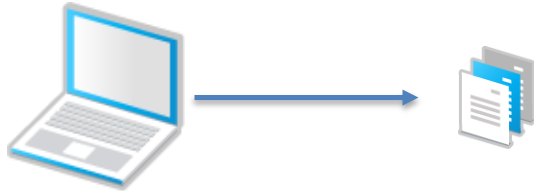




# Resolving DNS Requests



## Resolving using a hosts file.



```
127.0.0.1    localhost  loopback
64.183.75.82 mail    mail.fubar.com
64.183.75.83 ignatz  ignatz.fubar.com
192.168.0.36 crazy  crazy.fubar.com
```

A local file called “hosts” exists on Windows, Macintosh, and Linux systems. This file contains a list of names and IP addresses.

The hosts file was originally intended for environments where DNS wasn’t available, in which users got tired of typing in IP addresses all the time.



# Resolving using a hosts file.

**Windows:** %SystemRoot%\system32\drivers\etc\hosts

**Linux or Mac:** /etc/hosts

```
127.0.0.1      localhost    loopback
64.183.75.82  mail        mail.fubar.com
64.183.75.83  ignatz      ignatz.fubar.com
192.168.0.36  crazy       crazy.fubar.com
```

The hosts file is just a plain text file.

## Resolving using a hosts file.

<pre>127.0.0.1 64.183.75.82 64.183.75.83 192.168.0.36</pre>	<pre>localhost mail ignatz krazy</pre>	<pre>loopback mail.fubar.com ignatz.fubar.com krazy.fubar.com</pre>
IP Address	“short” host name	“long” (or alternate) host name

While there’s no standard for how names be ordered in a hosts file, above is one possible example. You can have multiple names for the same IP address. (Up to 255 characters per line.)



## Resolving using a hosts file.

```
127.0.0.1      localhost  loopback
64.183.75.82  mail      mail.fubar.com
64.183.75.83  ignatz    ignatz.fubar.com
192.168.0.36  crazy     crazy.fubar.com
```

The hosts file was designed to be the only method of name resolution before DNS services became widely available.

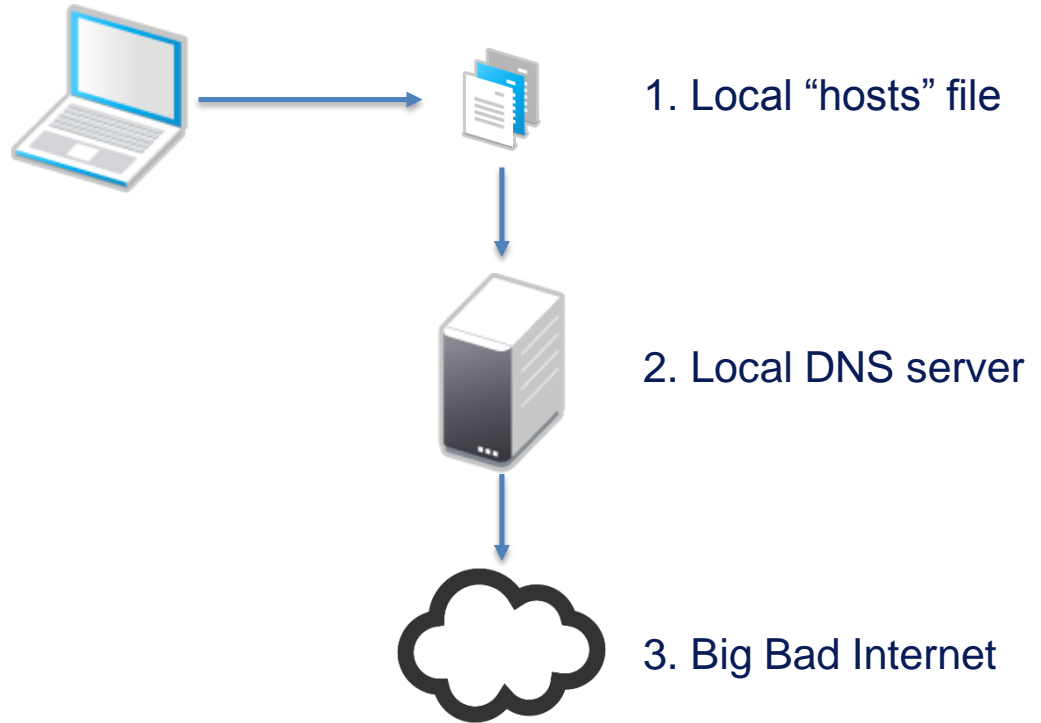
It can also be used to override DNS lookups, though there are easier ways now to do this globally.

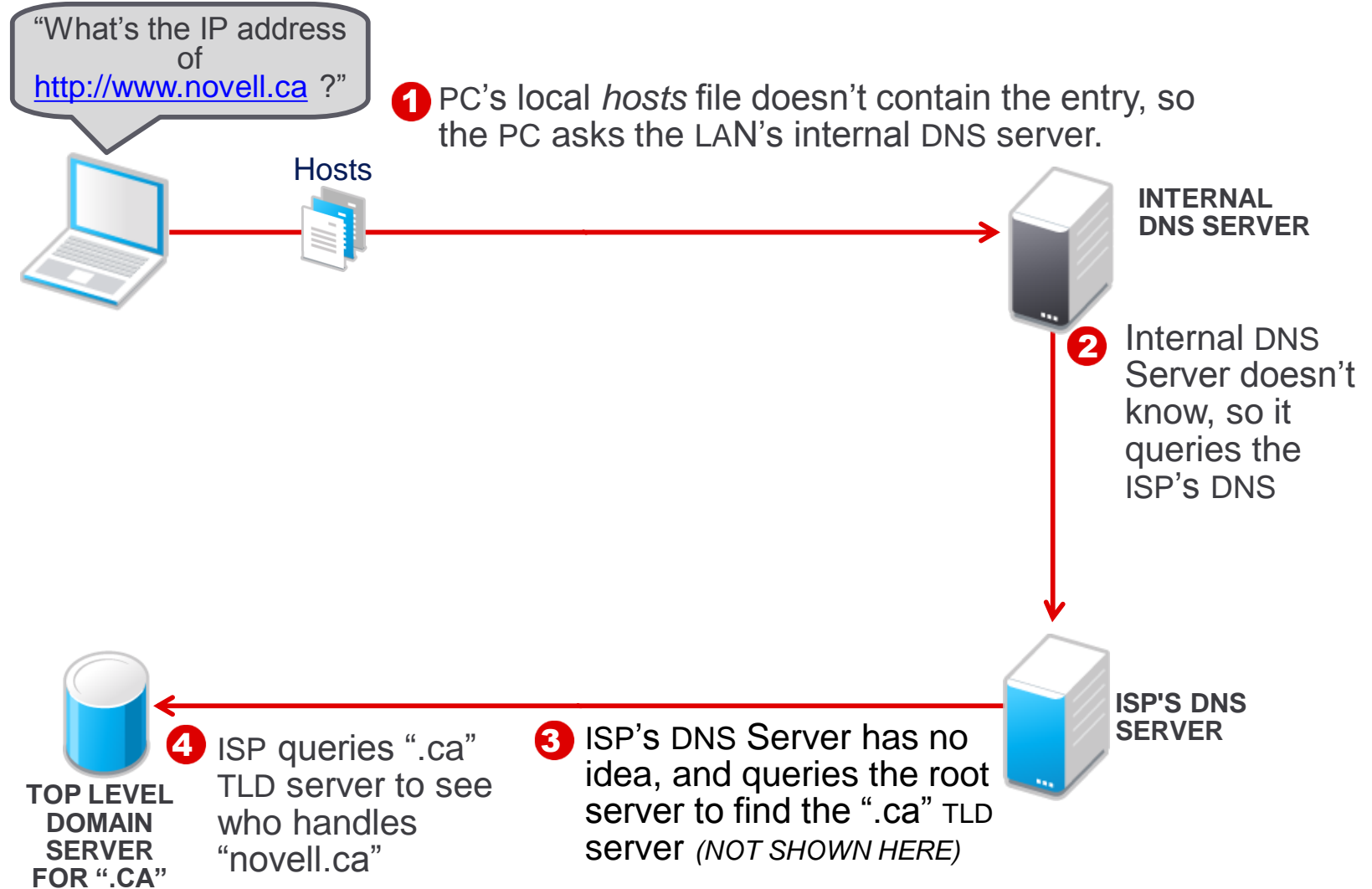
Maintaining hosts files on multiple systems is tedious and difficult...hence, the reason for DNS.

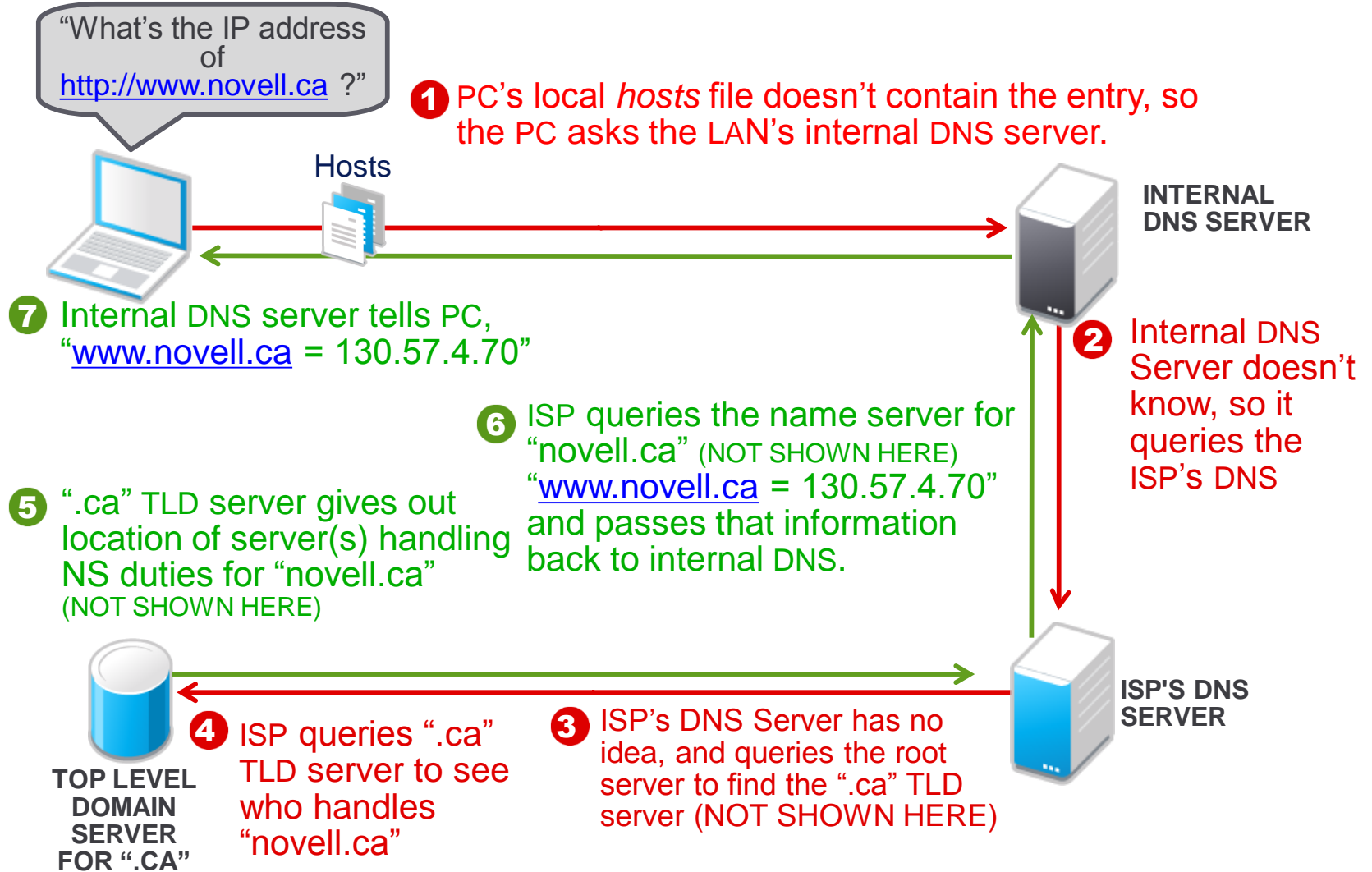
# Resolving Using DNS.

From an end user's standpoint, here's how DNS resolutions work.

(It's not *quite* this simple under the hood, as you'll see momentarily.)



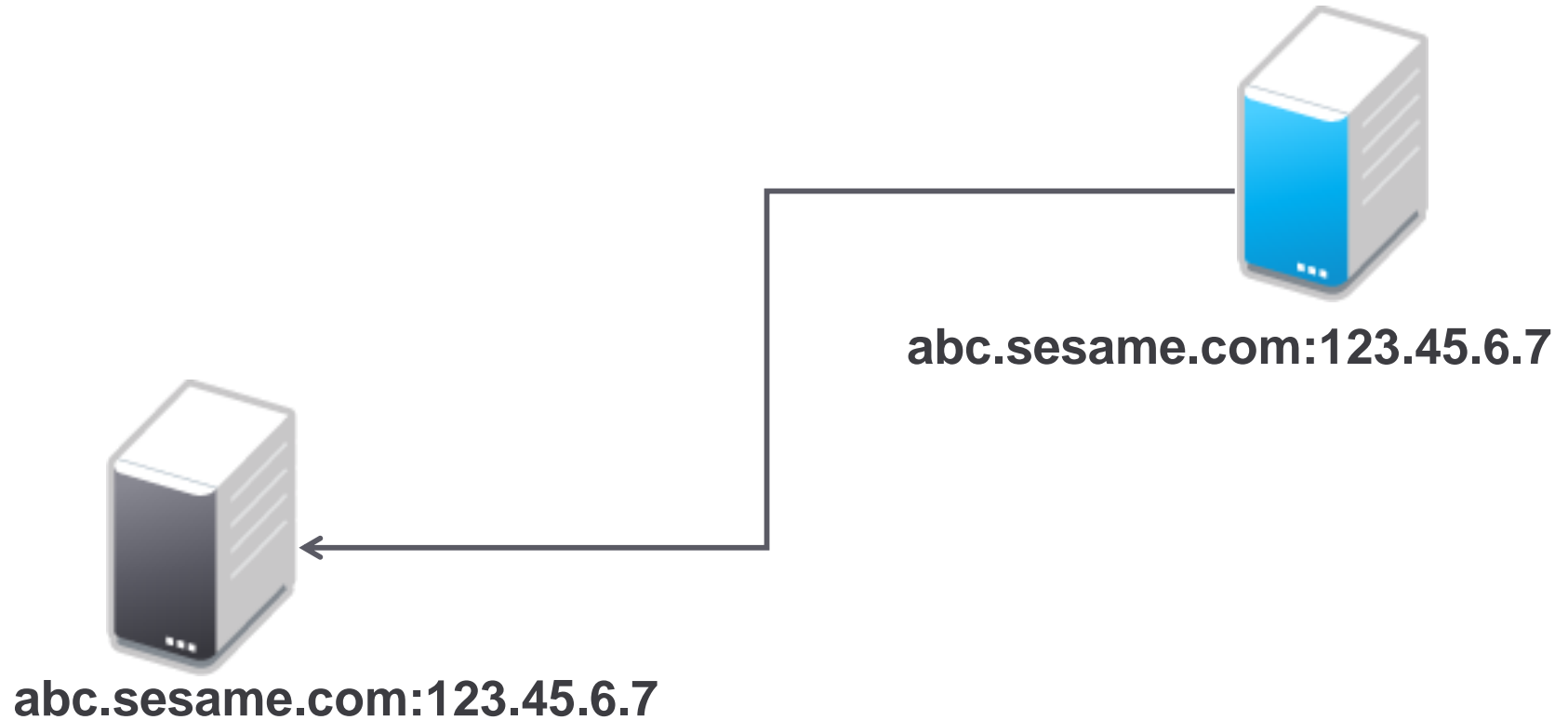




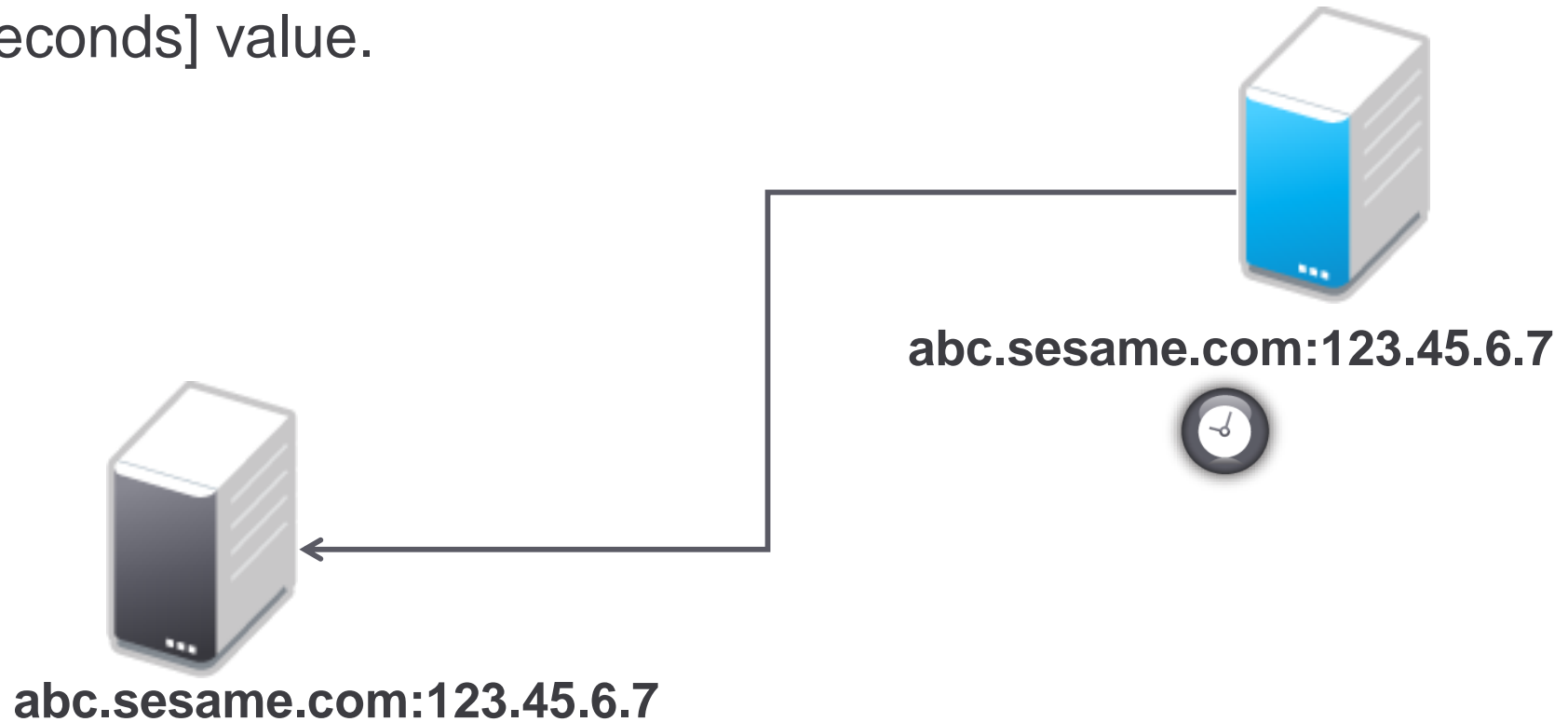


# Cache & Carry

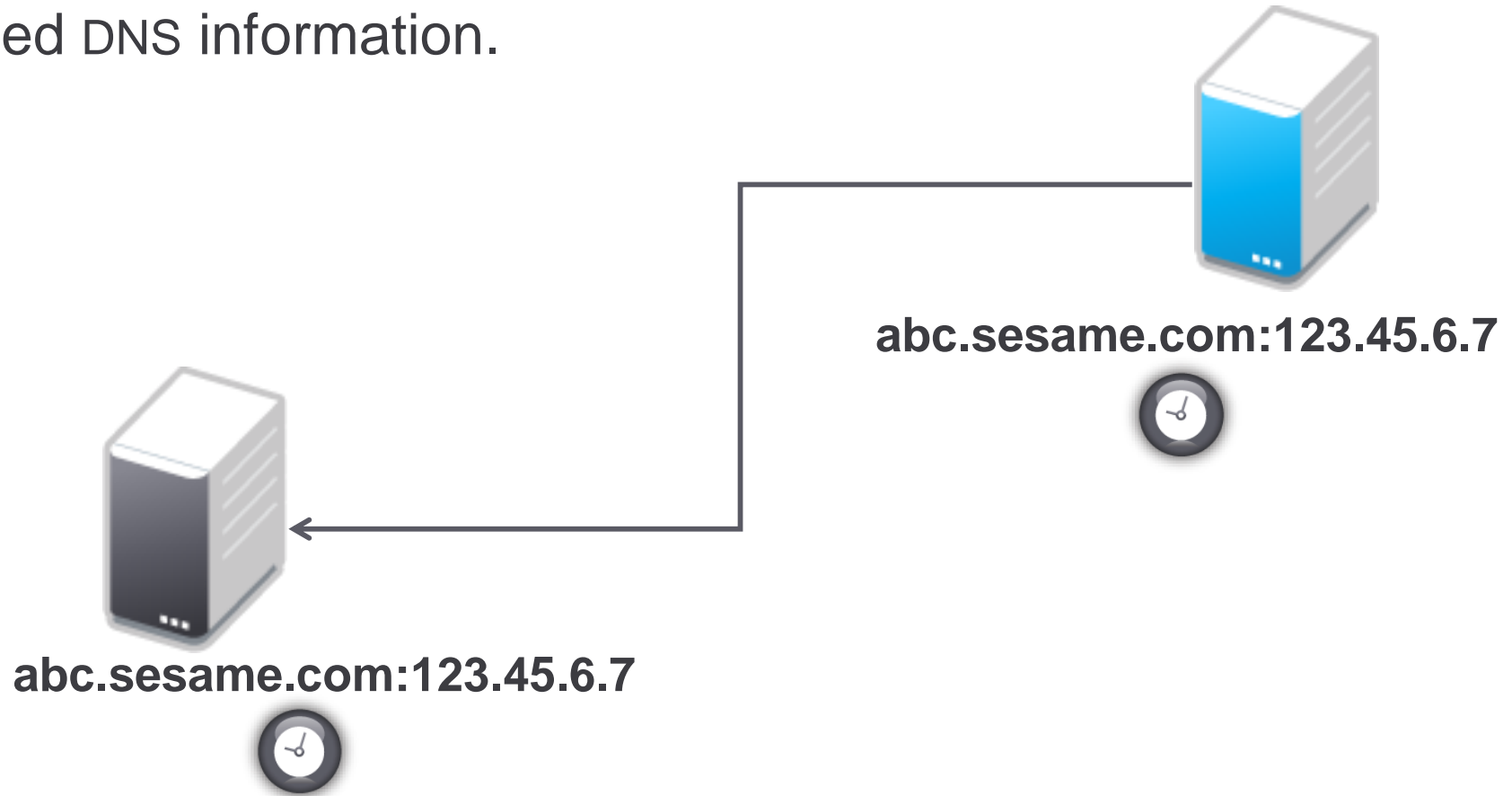
DNS servers cache information...



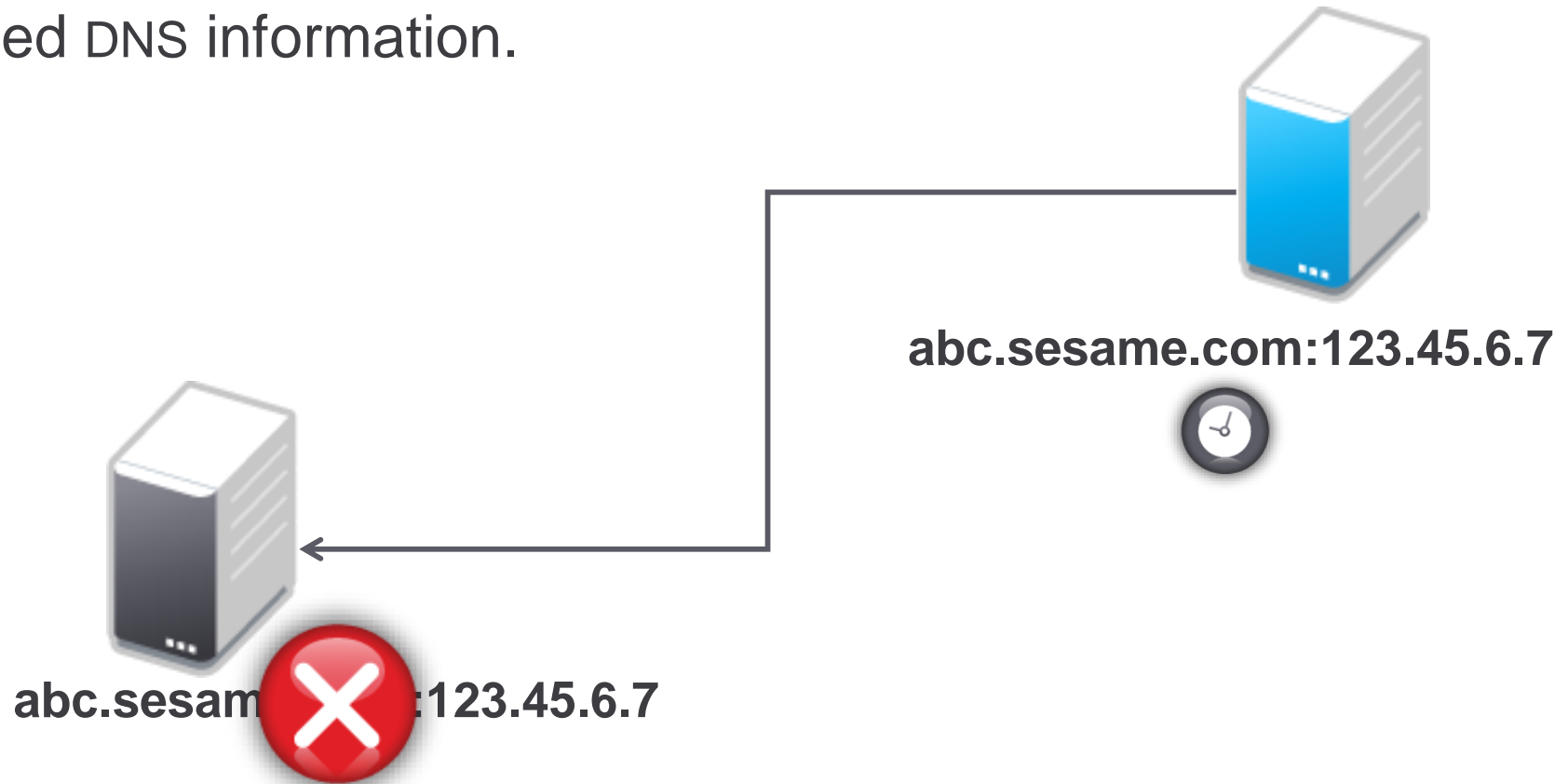
Cached information has a “Time To Live”  
[in seconds] value.



Time To Live (TTL) provides periodic dumping of old cached DNS information.



Time To Live (TTL) provides periodic dumping of old cached DNS information.



## KEEP IN MIND...

DNS servers may not honor TTLs

It may take several days for everyone on the Internet to request/receive modified DNS records

Most ISPs seem to cache data for 2 to 3 days

This is the origin of that *most* annoying phrase, “it may take anywhere from 24 to 72 hours for your DNS changes to fully propagate”





# **DNS Zone Files & Record Types**



## What's in a zone file?

Zone files were designed to be processed by a UNIX program called ***bind*** (“Berkeley Internet Name Domain”)

Bind is maintained by the Internet Systems Consortium  
<http://www.isc.org/>

Each zone file is comprised of many types of *resource records*

Each *resource record* describes a different aspect of DNS

Full list: <http://www.iana.org/assignments/dns-parameters>

### Some common record types:

**SOA** – “Start Of Authority” – contains zone info

**A** – contain IP address/name matching info

**MX** – “Mail eXchange” – contains mail server info

**CNAME** – “Canonical Name” – an alias



# Sample Resource Record (RR)

Ignatz



Name

10

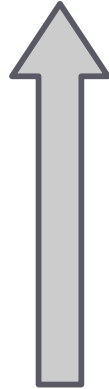


TTL

(Seconds)

(Blank="Don't Cache")

IN



Class

(IN = Internet)

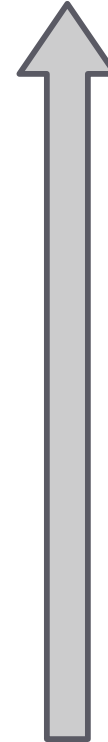
A



Record Type

(A, MX, CNAME, etc.)

183.27.12.2



Data

(in this case, IP address)



# **Dissecting a Zone File**

**(comparatively painlessly)**



## What's in a Zone File?

Here's a file for the mythical external domain, "fubar.com":

```
@ NS      ns1.myisp.com
@ NS      ns2.myisp.com
@ MX      10      mail
mail     A       64.183.75.82
ratso    A       64.183.75.83
www      CNAME   ratso
```

(Not shown here: the SOA record. We'll cover that next.)



## “Start of Authority” Record (SOA)

First record in each zone file.

Contains global information about the zone.

Lists the authoritative server for the zone.

Defines record management parameters.

---

```
@    IN          SOA      ns1.fubar.com      admin.fubar.com. (
2002040103 ; Serial Number [10 digits, changes per rev.]
7200      ; Slave refreshes from Master NS after 2 hours
3600      ; Slave will retry master after 1 hour
604800    ; If no answer from Master, Slave stops
           ;           responding to domain queries after 1 week
86400     ; Minimum TTL for all records in domain )
```



## Dissecting a Zone File (comparatively painlessly)

Here's a file for the mythical external domain,  
“fubar.com”:

---

```
@           NS           ns1.myisp.com
@           NS           ns2.myisp.com
@           MX           10                mail
mail       A            64.183.75.82
ratso      A            64.183.75.83
www        CNAME        ratso
```

---

The “@” signs in column 1 are placeholders for the domain name (fubar.com).



## Dissecting a Zone File (comparatively painlessly)

Here's a file for the mythical external domain,  
“fubar.com”:

---

```
@      NS      ns1.myisp.com
@      NS      ns2.myisp.com
@      MX      10          mail
mail   A       64.183.75.82
ratso  A       64.183.75.83
www    CNAME   ratso
```

---

Host names are the leftmost portions of the FQDN (Fully Qualified Domain Name)

For example, “mail.fubar.com”



## Dissecting a Zone File (comparatively painlessly)

Here's a file for the mythical external domain,  
“fubar.com”:

---

```
@      NS      ns1.myisp.com
@      NS      ns2.myisp.com
@      MX      10          mail
mail   A       64.183.75.82
ratso  A       64.183.75.83
www    CNAME   ratso
```

---

Column 2 is the Record Type:

Name Servers (NS)



## Dissecting a Zone File (comparatively painlessly)

Here's a file for the mythical external domain, "fubar.com":

---

@	NS	ns1.myisp.com	
@	NS	ns2.myisp.com	
@	<b>MX</b>	10	mail
mail	A	64.183.75.82	
ratso	A	64.183.75.83	
www	CNAME	ratso	

---

Column 2 is the Record Type:

Name Servers (NS)

Mail eXchange (MX) servers



## Dissecting a Zone File (comparatively painlessly)

Here's a file for the mythical external domain, "fubar.com":

---

```
@           NS           ns1.myisp.com
@           NS           ns2.myisp.com
@           MX           10           mail
mail       A           64.183.75.82
ratso      A           64.183.75.83
www        CNAME        ratso
```

---

Column 2 is the **Record Type**:

Name Servers (NS)

Mail eXchange (MX) servers

**Hosts (A)** ← Stands for "A"ddress record



## Dissecting a Zone File (comparatively painlessly)

Here's a file for the mythical external domain, "fubar.com":

---

@	NS	ns1.myisp.com	
@	NS	ns2.myisp.com	
@	MX	10	mail
mail	A	64.183.75.82	
ratso	A	64.183.75.83	
www	<b>CNAME</b>	ratso	

---

Column 2 is the **Record Type**:

Name Servers (NS)

Mail eXchange (MX) servers

Hosts (A)

**Canonical names (also known as "aliases")**



## Dissecting a Zone File (comparatively painlessly)

Here's a file for the mythical external domain,  
“fubar.com”:

---

@	NS	<b>ns1.myisp.com</b>	
@	NS	<b>ns2.myisp.com</b>	
@	MX	10	mail
mail	A	64.183.75.82	
ratso	A	64.183.75.83	
www	CNAME	ratso	

---

Column 3 is the Data Field:

**NS** = name server



## Dissecting a Zone File (comparatively painlessly)

Here's a file for the mythical external domain,  
“fubar.com”:

---

@	NS	ns1.myisp.com	
@	NS	ns2.myisp.com	
@	MX	10	mail
mail	A	64.183.75.82	
ratso	A	64.183.75.83	
www	CNAME	ratso	

---

Column 3 is the **Data Field**:

NS = name server

MX = mail priority (10) and mail host name (mail)



## Dissecting a Zone File (comparatively painlessly)

Here's a file for the mythical external domain,  
“fubar.com”:

---

```
@           NS           ns1.myisp.com
@           NS           ns2.myisp.com
@           MX           10           mail
mail       A           64.183.75.82
ratso     A           64.183.75.83
www       CNAME        ratso
```

---

Column 3 is the **Data Field**:

NS = name server

MX = mail priority (10) and mail host name (mail)

A = IP address



## Dissecting a Zone File (comparatively painlessly)

Here's a file for the mythical external domain,  
“fubar.com”:

---

```
@      NS      ns1.myisp.com
@      NS      ns2.myisp.com
@      MX      10          mail
mail   A       64.183.75.82
ratso  A       64.183.75.83
www    CNAME   ratso
```

---

Column 3 is the **Data Field**:

NS = name server

MX = mail priority (10) and mail host name (mail)

A = IP address

**CNAME = aliased host**



# **(Very) Basic DNS Troubleshooting**

## nslookup

“Built-in” lookup tool for Windows.

Linux version is deprecated, succeeded by “dig”...yet most people still use nslookup, because it’s quicker and easier than dig.

## dig

Officially preferred tool in Linux.

Has been ported to Windows; Google “*dig for windows*”





## Sample session with nslookup

```
C:\>nslookup
```

```
Default Server:  dns-rs1
```

```
Address:  12.127.17.71
```

```
> www.novell.ca
```

```
Server:  dns-rs1
```

```
Address:  12.127.17.71
```

```
Non-authoritative answer:
```

```
Name:      redirector.novell.com
```

```
Address:   130.57.5.70
```

```
Aliases:   www.novell.ca
```



## Sample session with nslookup

```
C:\>nslookup
```

```
Default Server:  dns-rs1  
Address:  12.127.17.71
```

```
> www.novell.ca
```

```
Server:  dns-rs1  
Address:  12.127.17.71
```

```
Non-authoritative answer:
```

```
Name:      redirector.novell.com  
Address:   130.57.5.70  
Aliases:   www.novell.ca
```

```
> 130.57.5.70
```

```
Server:  rmtu.mt.rs.els-gms.att.net  
Address:  12.127.16.67
```

```
Name:      redirector.novell.com  
Address:   130.57.5.70
```

← We can also put in an IP address to get a DNS name (Reverse DNS lookup).



## Sample session with nslookup

```
C:\>nslookup
```

```
Default Server:  dns-rs1  
Address:  12.127.17.71
```

```
> www.novell.ca
```

```
Server:  dns-rs1  
Address:  12.127.17.71
```

```
Non-authoritative answer:
```

```
Name:      redirector.novell.com  
Address:   130.57.5.70  
Aliases:   www.novell.ca
```

```
> 130.57.5.70
```

```
Server:  rmtu.mt.rs.els-gms.att.net  
Address:  12.127.16.67
```

```
Name:      redirector.novell.com  
Address:   130.57.5.70
```

← This shows who's hosting the reverse DNS records for that IP address range. Generally, this is the "owner" of the address range. *In this case, the IP address appears to be hosted by AT&T.*



## Sample session with dig

```
C:\dig>dig www.novell.ca.
```

```
; <<>> DiG 9.3.2 <<>> www.novell.ca.
;; global options:  printcmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 1478
;; flags: qr rd ra; QUERY: 1, ANSWER: 2, AUTHORITY: 3,
   ADDITIONAL: 3

;; QUESTION SECTION:
;www.novell.ca.                IN      A

;; ANSWER SECTION:
www.novell.ca.                86348  IN      CNAME
   redirector.novell.com.
redirector.novell.com.       86348  IN      A      130.57.5.70

;; AUTHORITY SECTION:
novell.com.                   83897  IN      NS      NS.UTAH.EDU.
novell.com.                   83897  IN      NS      ns.novell.com.
novell.com.                   83897  IN      NS
   NS1.WESTNET.NET.
```

**And there's still more information that won't fit on the page...**



# **Taking Control of Your Own DNS**



# How many DNS servers do I need?

At least two internal DNS servers.

(Note: I used to say “one”, but DNS has become important enough that now I strongly suggest two.)

At least two external DNS servers

DNS services don't add a large CPU or traffic load to internal servers





## **Do I *need* an internal DNS server?**

These days ... yes. In fact...you need two of 'em.

New web-based services require DNS entries.

Contemporary networks won't function efficiently without some kind of internal DNS.

Microsoft networks require two Domain Controllers, and each DC is also a DNS server.



## Should I host my external DNS?

External DNS needs fewer hosts than internal DNS.

Keep only the most essential hosts in external DNS.

Let an outside entity manage your external DNS.

For security, keep external DNS outside your firewall.

Have someone else host your DNS, which is an “attractive nuisance”. (Let hackers target *their* systems, not *yours*.)



## Should I host my external DNS?

Some domain registrars provide free self-service DNS hosting. (e.g., GoDaddy, Network Solutions)

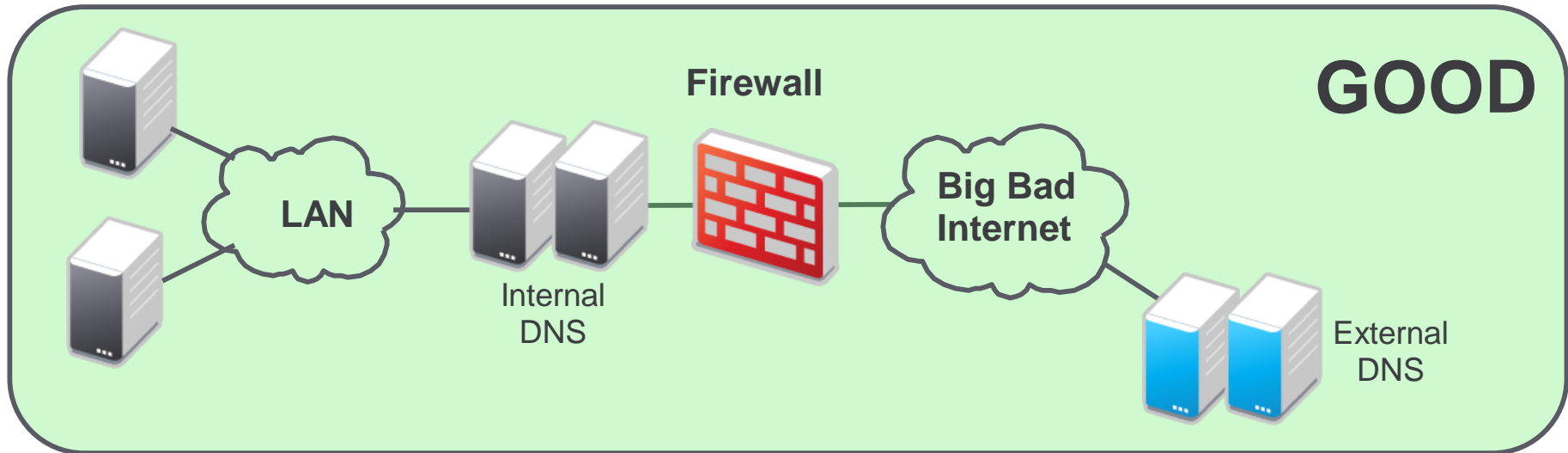
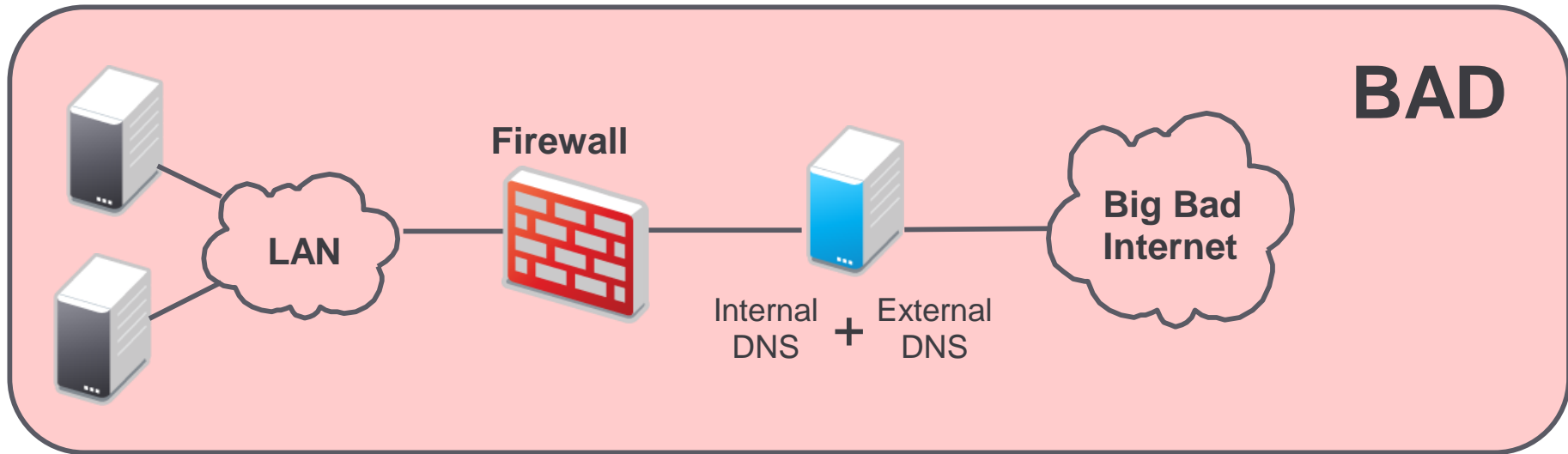
Only use external DNS providers that give you control over your DNS via a web-based, self-service control panel.

Avoid DNS providers that require changes to be emailed or called in. (There are still a few of these around.)

External DNS belongs on a dedicated set of servers outside your firewall.

*Never* connect your internal and external DNS systems.

# Do. Not. Multihome. DNS. Ever.





## **Do. Not. Multihome. DNS. Ever.**

Let's revisit that last statement again, shall we?

***Never connect internal and external DNS servers.***

The security risk is high. There's nothing to be gained in efficiency by using the same server for your internal and external DNS.

Keeping internal and external DNS on separate sets of servers provides security through compartmentalization; external entities can only see the few records you've placed in your external DNS.



# Ransoming Back DNS From Your Web Hosting Company or ISP ...

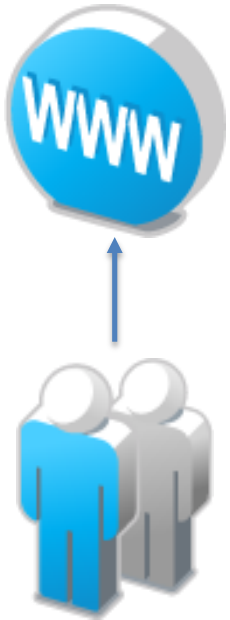
Difficult web hosting company? Try this...

1. Create & register a new domain name. Append the word “web” to your existing domain (e..g, “**fubar.com**” becomes “**fubarweb.com**”).
2. Have your web hosting company move to the new domain, which they can control all they like.
3. Transfer your original domain name (fubar.com) to any registrar providing self-service DNS.
4. Create a CNAME record for your original domain name (www.fubar.com) pointing to your new domain name (www.fubarweb.com), or use “Domain Forwarding”.



# Ransoming Back DNS From Your Web Hosting Company or ISP ...

<http://www.fubar.com>



Web hosting company

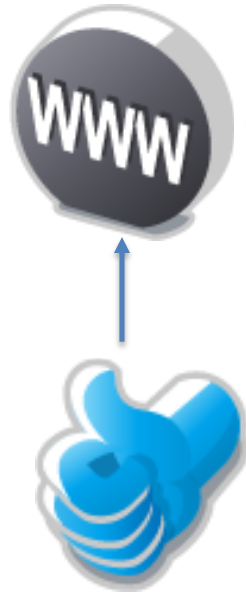


# Ransoming Back DNS From Your Web Hosting Company or ISP ...

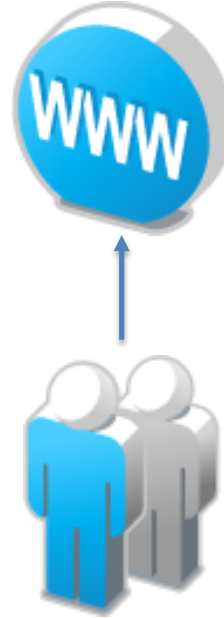
fubar.com

*www CNAME fubarweb.com*

<http://www.fubarweb.com>



You control **your** domain.



Web hosting company controls this domain.



# **DNS & Other Systems**



## DNS & DHCP:

**“You got peanut butter in my chocolate!”**

DNS & DHCP work together quite nicely

DNS by itself is called “static DNS”.

DNS linked to DHCP is called “dynamic DNS” (or “DDNS”).

DDNS creates “instant” host name and reverse DNS records whenever a workstation acquires an IP address via DHCP





# DNS & Active Directory

There are two types of DNS zones in Active Directory:

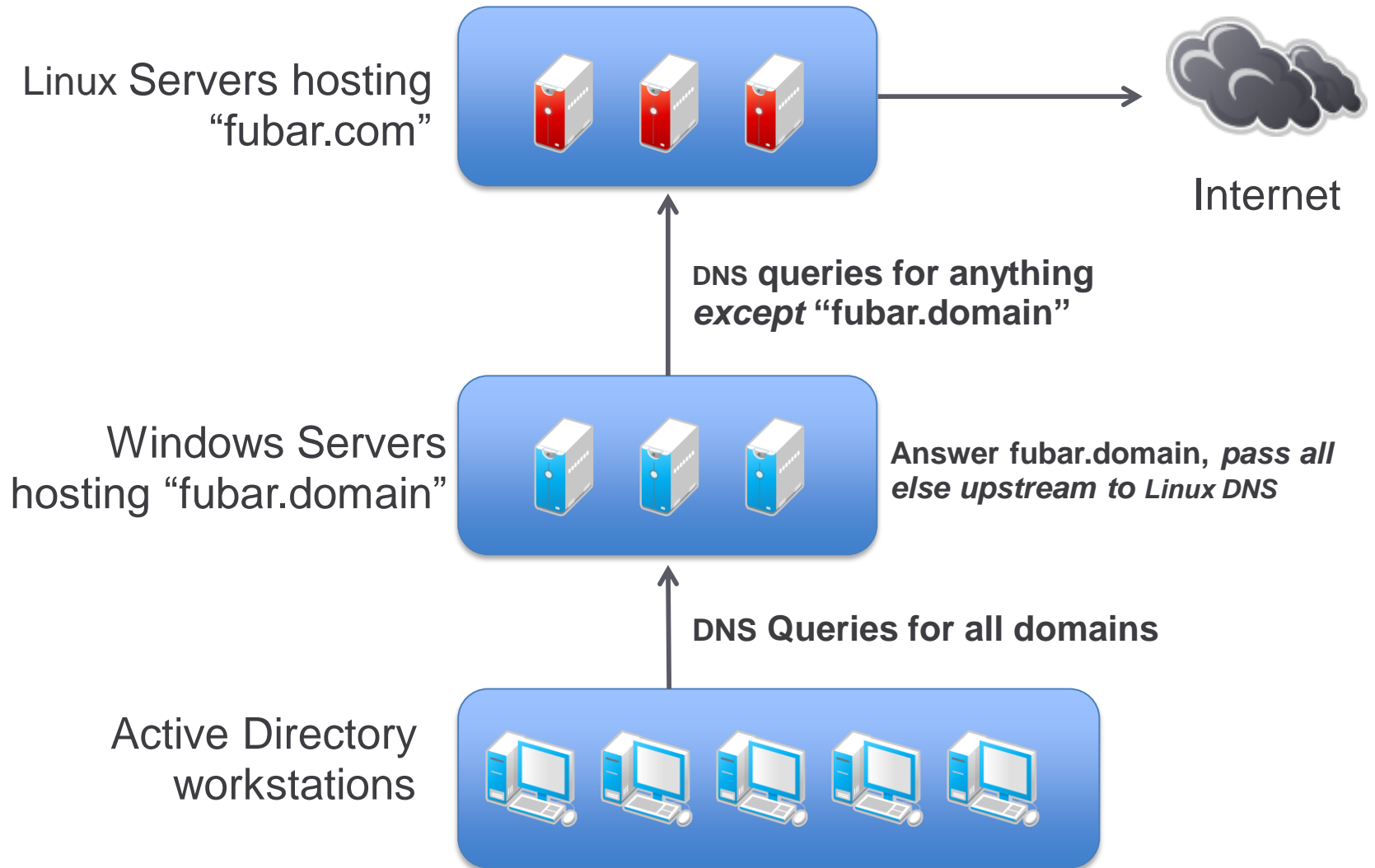
Integrated Zone	Non-Integrated (“Regular”) Zone
Stores DNS zone information in Active Directory.	Stores DNS zone information on a local DNS server.
Replicates zone information automatically between all Active Directory domain controllers.	Replicates zone information only if manually configured to do so.
Can be used with a mixture of AD and non-AD servers.	Can be used with a mixture of AD and non-AD servers.
Required for Active Directory domains, optional and OK for non-AD domains.	Used for non-Active Directory domains only.
Can use a “non-routeable” TLD such as .domain or .ad	Generally uses a standard TLD such as .com, .net, .edu, etcetera.
Should <i>never</i> use a TLD of .local	Should <i>never</i> use a TLD of .local
Active Directory domains should <i>never</i> be exactly the same as your external DNS zone name. It <i>can</i> , however, be a <i>sub</i> -domain, such as “ad.fubar.com”	A non-AD domain can be (and usually is) the same as your external DNS zone name.



## DNS & Active Directory

Some “old-school” network administrators don’t trust AD DNS. While I don’t think such mistrust is justified any more, here’s how to mix-and-match AD/non-AD DNS.

1. Create your MS network’s *integrated* DNS using Active Directory. (e.g., “fubar.ad”)
2. Create/maintain your network's non-AD DNS domain using something else, such as Linux. (e.g., “fubar.com”)
3. Point AD DNS to your Linux DNS server for resolution of your “real” DNS domain (e.g., “fubar.com”)





## Best Practices for AD DNS

DNS is installed as part of Active Directory; figure out your DNS *before* grabbing the installation CD/DVD.

### Useful AD DNS Resources

- Setting up Dynamic Name Services for AD:
  - Microsoft Knowledgebase article 237675
- Dan DiNicolo's "Quick Start Guide to Setting Up AD":
  - <http://www.serverwatch.com/tutorials/article.php/1474461>



## Best Practices for AD DNS

Let's say your company's domain is "sample.com". Here's what your Active Directory domain name should NOT be:

- sample.com ... Don't use your public facing domain for Active Directory.
- sample.local ... ".local" is a device type in Linux/Unix.

Here are some possibilities of what your AD name COULD be:

- ad.sample.com ... Using a subdomain for AD is OK.
- ad.sample.domain ... "Non-routable" domains are more secure than "routable".
- ad.domain ... "Non-routable", no connection at all to your primary domain.

*Keep the leftmost subdomain short.* This makes it easier for your users.

For example, in all three of the "could" cases above, the user would log on as "AD\username".



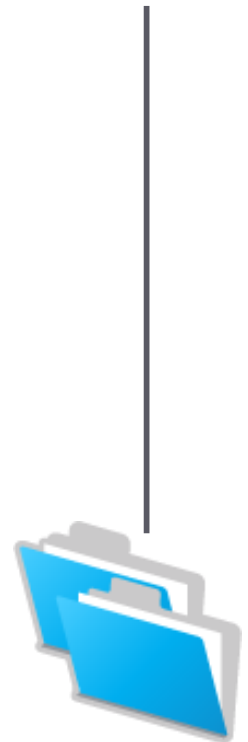
# What other DNS records do I need?

## www / ftp

Points to the address of your public web & ftp server(s), so internal users can get there as easily as external users.

## External Resources:

- Webmail
- Employee Portal
- Exchange Client Access
- SharePoint

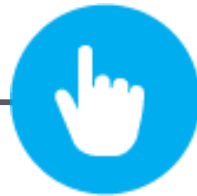


<http://www.dnsstuff.com>

Extended DNS utilities.

<http://www.zytrax.com/books/dns/>

“DNS For Rocket Scientists”... my favorite DNS reference text.





# Thank You!

**Allan Hurst**

Director of Enterprise Strategy

510.933.7555

[allanh@kiscc.com](mailto:allanh@kiscc.com)

<http://www.kiscc.com>



*The Virtualization & IT Solution Experts*