Leveraging Multiplatform DNS for Web Application Resiliency

Matthew Torrisi
Solutions Architect - Oracle Dyn
Safe Harbor Statement

The following is intended to outline our general product direction. It is intended for information purposes only, and may not be incorporated into any contract. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. The development, release, and timing of any features or functionality described for Oracle’s products remains at the sole discretion of Oracle.
Program Agenda

1. How Internet Volatility with DNS Is impacting Web Performance (40 min)
2. How to Build Resilient Infrastructure for High-Performance DNS (40 min)
3. Designing a Multilayer, Multiplatform Solution (40 min)
4. WORKSHOP: Optimizing your Zonefile for Multiplatform DNS (30 min)
DNS Volatility Is Impacting Web Performance
DNS - What’s Top of Mind?

- DDoS, Attacks, Threats, Defense
- Cloud, Infrastructure, APIs
- Traffic Steering, Geo, Traffic Management
“Wait, isn’t this just DNS? My DNS has been fine with my registrar/hosting/server-in-basement since forever”

— Approximately Everyone
DNS Is First in the Chain for a User

Dyn.com render on webpagetest.org
Route Early, Route Often

• First in chain allows users to be approximately routed before initial connection

• DNS agnostic to endpoint, good place for cloud management
Everything Starts with DNS

• First in chain means failure in DNS lookup extra critical

• When DNS is down, everything is down
  – Email
  – VoIP
  – Support Pages
Web Performance Starts at Availability

• Layers of the stack build on each other

• If you want to be fast, you need to be available

• This isn’t a security or operations problem, this is a **web performance problem**
Lack of Diversity

- DNS diversity is pitiful
- “[Among the top 100 ranked domains] the percentage of domains using a single provider decreased from 95% in November of 2011 to 86% in October 2016. After the Dyn attack, it plummeted to a low of 78% in February 2017.”
- That’s still bad
- Many vendors make this difficult
What happened to having N +1 Diversity?
It’s not enough that your Ops think like Devs...
It’s not enough that your Ops think like Devs…

Your Devs need to think like Ops
This. Is. Not. New.

Tom Daly, Former CTO and Dyn Founder, Surge 2012
What Has Changed?

• More Internet
  – EVERYTHINGaaS
  – Increased internet buying patterns means the public internet is your primary vehicle to interact with consumers
  – Devops pushing cloud-first stacks

• Bigger Threats
  – Bigger DDoS attacks through amplification, IoT (Mirai), BGP hijack
  – Easier to launch, DDoS for hire services, bigger botnets with increased bandwidth
Begun the IoT Wars Have
If DNS is critical, and is being used in new and important ways...
...But this requires use of vendor specific code for Traffic Steering...
How do I build my diversity at multiple layers?
Today You Will Learn

1. How DNS behaves with diverse delegations in peacetime and outage scenarios

2. The techniques and considerations of building a multiplatform DNS solution

3. How to use DNS for traffic steering AND have redundant multivendor DNS
The DNS Lookup
DNS in Under a Minute

Users

Examples:
You! Trying to access something on the internet
DNS in Under a Minute

Users

Examples:
You! Trying to access something on the internet

Recursivelys

Examples:
Usually local ISP, Comcast, Verizon, BT, PCCW, Google, etc. or could be managed by enterprise
DNS in Under a Minute

**Users**
Examples:
You! Trying to access something on the internet

**Recursives**
Examples:
Usually local ISP, Comcast, Verizon, BT, PCCW, Google, etc. or could be managed by enterprise

**Authoritatives**
Examples:
Internally operated, Registrar/Hosting DNS, Managed providers like Oracle Dyn
DNS in Under a Minute

Users
~4.2B

Recursives
~130k*

Authoritatives
~100*

*Doing any major traffic
DNS in Under a Minute

Users

Initiates request for example.com. Check local, pass to stub resolver, forward to Recursive Resolver

Recursives

Authoritatives

Example.com?
DNS in Under a Minute

Users

Is this in my cache?

Example.com?

Recursives

Check cache, navigate the DNS tree delegation, find authoritative and send query

Authoritatives
DNS in Under a Minute

Users

Recursives

Example.com?

Example.com?

Authoritatives

The authority for all records on the domain.
DNS in Under a Minute

Users

Stores answer locally, uses to perform action such as HTTP request

Example.com?

1.1.1.1

Recursives

Stores answer in cache, responds to user

Example.com?

1.1.1.1

Authoritatives

Responds to Recursive with answer
Review

Users

Know nothing.

Recursives

Ask the **QUESTION**, many many times fetching the answer for the user

Authoritatives

Know everything about a particular domain. Give the **ANSWER.**
10 Minute Break
Nameserver Selection and Multivendor DNS

(why you should stop calling it Secondary DNS)
Delegations on Delegations

example.com.
Delegations on Delegations

example.com.

Fully Qualified Domain Name (FQDN)
Delegations on Delegations

example.com.

Root
Delegations on Delegations

example.com.

Top Level Domain (TLD)
Delegations on Delegations

example.com.

“Domain”?
Delegations on Delegations

```
  .
 /|
| v
|  |
| example.com.
```
Nameserver Selection

**Any Nameserver in Delegation is a Valid Authority**
When the recursive resolver performs the initial query, it has no context for where those servers are.

<table>
<thead>
<tr>
<th>Time</th>
<th>Nameserver</th>
</tr>
</thead>
<tbody>
<tr>
<td>?</td>
<td>ns1.p43.dynect.net</td>
</tr>
<tr>
<td>?</td>
<td>ns2.p43.dynect.net</td>
</tr>
<tr>
<td>?</td>
<td>ns3.p43.dynect.net</td>
</tr>
<tr>
<td>?</td>
<td>ns4.p43.dynect.net</td>
</tr>
<tr>
<td>?</td>
<td>dns1.example.net.</td>
</tr>
<tr>
<td>?</td>
<td>dns2.example.net.</td>
</tr>
<tr>
<td>?</td>
<td>dns3.example.net.</td>
</tr>
<tr>
<td>?</td>
<td>dns4.example.net.</td>
</tr>
</tbody>
</table>

When the recursive resolver performs the initial query, it has no context for where those servers are. **Whom to ask?**
Nameserver Selection

**Initial Request Essentially Random**

Different implementations could work sequentially or build a random function

---

<table>
<thead>
<tr>
<th>Time</th>
<th>Nameserver</th>
</tr>
</thead>
<tbody>
<tr>
<td>?</td>
<td>ns1.p43.dynect.net</td>
</tr>
<tr>
<td>?</td>
<td>ns2.p43.dynect.net</td>
</tr>
<tr>
<td>?</td>
<td>ns3.p43.dynect.net</td>
</tr>
<tr>
<td>?</td>
<td>ns4.p43.dynect.net</td>
</tr>
<tr>
<td>?</td>
<td>dns1.example.net.</td>
</tr>
<tr>
<td>?</td>
<td>dns2.example.net.</td>
</tr>
<tr>
<td>?</td>
<td>dns3.example.net.</td>
</tr>
<tr>
<td>?</td>
<td>dns4.example.net.</td>
</tr>
</tbody>
</table>
Nameserver Selection

Recursive “Primes” Full Delegation
All nameservers receive a query, regardless of location or availability. Asymmetric delegations not recommended for this reason.

<table>
<thead>
<tr>
<th>Time</th>
<th>Nameserver</th>
</tr>
</thead>
<tbody>
<tr>
<td>?</td>
<td>ns1.p43.dynect.net</td>
</tr>
<tr>
<td>?</td>
<td>ns2.p43.dynect.net</td>
</tr>
<tr>
<td>?</td>
<td>ns3.p43.dynect.net</td>
</tr>
<tr>
<td>?</td>
<td>ns4.p43.dynect.net</td>
</tr>
<tr>
<td>?</td>
<td>dns1.example.net.</td>
</tr>
<tr>
<td>?</td>
<td>dns2.example.net.</td>
</tr>
<tr>
<td>?</td>
<td>dns3.example.net.</td>
</tr>
<tr>
<td>?</td>
<td>dns4.example.net.</td>
</tr>
</tbody>
</table>

Recursives
Nameserver Selection

Preference or “Affinity” Selected
The Recursive will not normally view any single NS as related to any other, so traffic may be spread across providers based on performance in a “band” of acceptable tolerance.

Recursives

Queries sent to fastest performing NS

<table>
<thead>
<tr>
<th>Time</th>
<th>Nameserver</th>
</tr>
</thead>
<tbody>
<tr>
<td>13ms</td>
<td>ns1.p43.dynect.net</td>
</tr>
<tr>
<td>26ms</td>
<td>ns2.p43.dynect.net</td>
</tr>
<tr>
<td>12ms</td>
<td>ns3.p43.dynect.net</td>
</tr>
<tr>
<td>20ms</td>
<td>ns4.p43.dynect.net</td>
</tr>
<tr>
<td>13ms</td>
<td>dns1.example.net.</td>
</tr>
<tr>
<td>13ms</td>
<td>dns2.example.net.</td>
</tr>
<tr>
<td>16ms</td>
<td>dns3.example.net.</td>
</tr>
<tr>
<td>18ms</td>
<td>dns4.example.net.</td>
</tr>
</tbody>
</table>
Nameserver Selection

Failure Initiates Retries
Failure means a user has to wait until a timeout value, before the resolver tries another NS. But that could be another failed NS…

Recursives

That was unpleasant

Most timeout values at 2s / 2000ms

Example.com Delegation

<table>
<thead>
<tr>
<th>Time</th>
<th>Nameserver</th>
</tr>
</thead>
<tbody>
<tr>
<td>13ms</td>
<td>ns1.p43.dynect.net</td>
</tr>
<tr>
<td>26ms</td>
<td>ns2.p43.dynect.net</td>
</tr>
<tr>
<td>12ms</td>
<td>ns3.p43.dynect.net</td>
</tr>
<tr>
<td>20ms</td>
<td>ns4.p43.dynect.net</td>
</tr>
<tr>
<td>13ms</td>
<td>dns1.example.net.</td>
</tr>
<tr>
<td>13ms</td>
<td>dns2.example.net.</td>
</tr>
<tr>
<td>16ms</td>
<td>dns3.example.net.</td>
</tr>
<tr>
<td>18ms</td>
<td>dns4.example.net.</td>
</tr>
</tbody>
</table>
Nameserver Selection

DNS Failover is Automatic
Although some traffic will be affected, with multiple providers the cutover will be incredibly quick for the majority of traffic. Happens automatically and on a local level, Recursive-by-Recursive.

Example.com Delegation

<table>
<thead>
<tr>
<th>Time</th>
<th>Nameserver</th>
</tr>
</thead>
<tbody>
<tr>
<td>13ms</td>
<td>ns1.p43.dynect.net</td>
</tr>
<tr>
<td>26ms</td>
<td>ns2.p43.dynect.net</td>
</tr>
<tr>
<td>12ms</td>
<td>ns3.p43.dynect.net</td>
</tr>
<tr>
<td>20ms</td>
<td>ns4.p43.dynect.net</td>
</tr>
<tr>
<td>13ms</td>
<td>dns1.example.net.</td>
</tr>
<tr>
<td>13ms</td>
<td>dns2.example.net.</td>
</tr>
<tr>
<td>16ms</td>
<td>dns3.example.net.</td>
</tr>
<tr>
<td>18ms</td>
<td>dns4.example.net.</td>
</tr>
</tbody>
</table>

You're dead to me

Most timeout values at 2s / 2000ms
Things We Learned

• Secondary != Backup DNS

• “Secondary” = MULTIVENDOR DNS

• Failover automagically handled by Recursive, not the operator

• Routes based on performance to each individual nameserver
Then Why “Secondary”?
What Is a Primary?

• Primary, also known by “Master”, is a DNS service which has an administrative Read/Write function for a particular zonefile

• Because it is zonefile specific, it is possible for zones of both primary and secondary configuration to coexist on the same platform

• Zones generally need to be Primary in order to operate
What is a Secondary?

- Secondary, also known as “Slave”, is a DNS service which has an administrative Read Only function for a particular zonefile.

- Because it is zonefile specific, it is possible for zones of both primary and secondary configuration to coexist on the same platform.

- Because secondary, generally cannot perform Traffic Steering.
How an Update Works

- **Step 1 - Initiate**
  - Option A - the zonefile hits the Refresh TTL on the SOA record at the secondary
  - Option B - the Primary initiates a NOTIFY to the secondary
How an Update Works

• Step 2 – Validate Serial
  – Even though the Primary just sent the SOA in the NOTIFY, the Secondary will immediately forget it and check the SOA directly. Whatever.
  – If the Primary serial is HIGHER, proceed. The Secondary could be the same or even lower and will not proceed to sync zone.
How an Update Works

- Step 3 – XFR
  - The secondary will request the needed updates to the zone to get in sync. There are two methods
    - AXFR - The whole zone. Used for initial zone load into the secondary, as well as with older systems that do not support IXFR
    - IXFR - Incremental XFR, where the only records given are those since the provided serial of the Secondary
  - Because the Secondary initiates, the XFR is more of a PULL than a PUSH
Break It Down

- Primary and Secondary are methods of zonefile management, used to keep a zone in sync.
- The nameservers present in delegation govern traffic, which is based on performance.
- These are **two distinct concepts** that are related but not linked. Each possibility has a legitimate use.

<table>
<thead>
<tr>
<th></th>
<th>Primary</th>
<th>Secondary</th>
</tr>
</thead>
<tbody>
<tr>
<td>Controls Zone</td>
<td>Receive Traffic</td>
<td>Copy of Zone</td>
</tr>
<tr>
<td></td>
<td>No Traffic</td>
<td>Receive Traffic</td>
</tr>
<tr>
<td></td>
<td></td>
<td>No Traffic</td>
</tr>
</tbody>
</table>

Present | Hidden
Remember our Recursive?

How would the recursive know which nameserver (if any) is Master and which is/are secondary?

<table>
<thead>
<tr>
<th>Time</th>
<th>Nameserver</th>
</tr>
</thead>
<tbody>
<tr>
<td>?</td>
<td>ns1.p43.dynect.net</td>
</tr>
<tr>
<td>?</td>
<td>ns2.p43.dynect.net</td>
</tr>
<tr>
<td>?</td>
<td>ns3.p43.dynect.net</td>
</tr>
<tr>
<td>?</td>
<td>ns4.p43.dynect.net</td>
</tr>
<tr>
<td>?</td>
<td>dns1.example.net.</td>
</tr>
<tr>
<td>?</td>
<td>dns2.example.net.</td>
</tr>
<tr>
<td>?</td>
<td>dns3.example.net.</td>
</tr>
<tr>
<td>?</td>
<td>dns4.example.net.</td>
</tr>
</tbody>
</table>
Insource vs Outsource

Pick your poison.
Many organizations will start out doing DNS in-house, often with hardware DDI Devices that take care of DNS, DHCP, and IP Address Management for the whole network.

**Pros:**
- Inexpensive to throw BIND on a server laying around
- Total control over tech
- Add on if already using DDI device

**Cons:**
- Expensive to do well, especially to build a comparable network
- Full force of attack rests on your security team
- Hard to combat large and targeted threats
- Much higher labor
Outsourcing

On the other hand, managed DNS companies offer a way to serve DNS globally; it’s highly scalable, with value-added features, and expertise

**Good:**
- Improved performance and global coverage
- Traffic Steering capabilities
- DNS Expertise, automatic updates

**Bad:**
- Hard to create budget if new realized expense
- Loss of visibility at edge
- Increased collateral risk
Let’s Talk Risk

• While countless things *could* go wrong, most outages are the result of a DDoS attack

• With Insourcing, there is decreased chance of an attack, because an attack must be directed towards you. However, when an attack does hit, you have fewer resources

• With Outsourcing, you gain the risk of each customer the vendor adds, but have vastly more resources in bandwidth, servers, and staff experienced with attacks
## Risk Review

<table>
<thead>
<tr>
<th></th>
<th>Insourcing</th>
<th>Outsourcing</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Pro</strong></td>
<td>No collateral impact means less risk of attack</td>
<td>Large network and teams mitigate attacks effectively</td>
</tr>
<tr>
<td><strong>Con</strong></td>
<td>Decreased resources means increased chance of attack causing outage</td>
<td>Increased risk from attacks on other customers within vendor portfolio</td>
</tr>
</tbody>
</table>
Trust no one.
Trusting yourself is still trusting someone.
Nobody.
Option 0: Full Insourse

How it works:
The PRIMARY is on premises, maintaining existing workflow, often a DDI device. Primary receives all traffic and there are no external Secondary solutions.
Option 0: Full Insourcing is a terrible idea

It is unlikely an internally managed solution will adequately respond to widespread and advanced threats.
Option 1: Hidden Master

How it works:
The PRIMARY is on premises, maintaining existing workflow, often a DDI device. SECONDARY is cloud provider receiving updates. Only provider is in delegation to receive traffic.
Option 1: Hidden Master

How it works:
The PRIMARY is on premises, maintaining existing workflow, often a DDI device. SECONDARY is cloud provider receiving updates. Only provider is in delegation to receive traffic.
This was a test.
Trust. No. One.
Option 1a: Hidden Master with Two Cloud Providers

How it works:
Internal PRIMARY can update as many SECONDARIES as one would like

- Users
- Recursives
- Cloud Provider 1
  - Internal Master
  - Updates
- Cloud Provider 2

1.1.1.1
Example.com?
Option 2: Hybrid Cloud

How it works:
The PRIMARY is on premises, maintaining existing workflow. SECONDARY is cloud provider receiving updates. Both are in delegation providing redundancy from provider outage.
Option 3: Multicloud

How it works:
Multiple cloud providers either in PRIMARY-PRIMARY relationship with API, or PRIMARY-SECONDARY using NOTIFY / IXFR to stay in sync.
Considerations for a DNS Partner
Considerations for a DNS partner

1. Network
   Global Network with good multihomed connectivity and multiple everything in their stack
Considerations for a DNS partner

1. **Network**
   Global Network with good multihomed connectivity and multiple everything in their stack

2. **Support**
   Global, reachable, and experienced
Considerations for a DNS partner

1. **Network**
   Global Network with good multihomed connectivity and multiple everything in their stack

2. **Support**
   Global, reachable, and experienced

3. **Plays Nice with Others**
   Ability for secondary - XFR, NOTIFY, TSIG, APIs, able to add NS records, doesn’t force solo delegation
Considerations for a DNS partner

1. **Network**
   Global Network with good multihomed connectivity and multiple everything in their stack

2. **Support**
   Global, reachable, and experienced

3. **Plays Nice with Others**
   Ability for secondary - XFR, NOTIFY, TSIG, APIs, able to add NS records, doesn’t force solo delegation

4. **Traffic Steering Services**
   Availability of services to manage traffic based on health, GeoIP, Round Robin, etc.
Considerations for a DNS partner

1. **Network**
   Global Network with good multihomed connectivity and multiple everything in their stack

2. **Support**
   Global, reachable, and experienced

3. **Plays Nice with Others**
   Ability for secondary - XFR, NOTIFY, TSIG, APIs, able to add NS records, doesn’t force solo delegation

4. **Traffic Steering Services**
   Availability of services to manage traffic based on health, GeoIP, Round Robin, etc.

5. **Charming Solutions Engineers**
10 Minute Break
Using DNS for Operational Control
Dyn to DC | LB to Asset

Federated Load Balancing

- Often used to provide a GLB solution to drive users to a DC or cloud solution
- Utilize local LB to control assets within DC
- No integration required. Independent health monitoring
- Perform lift and shift of GSLB or provide redundancy for DDoS protection
End-to-end Health

Check Health Outside System

- Most premium DNS vendors will offer feature to check health
- Being outside of system, can catch scenarios where system is available but not reachable
- Either replicate user or hit monitoring page to initiate series of checks
Failover with DNS

Steer Around Issues

- Because DNS is first, user never attempts to connect to failed DC
- DNS doesn’t care what an asset is. Could be any service hosted anywhere
- Add service whenever, wherever

Traffic Director

US-based DNS requests

2.2.2.2

www.example.com

EU-based DNS requests

2.2.2.2

*Example scenario

IAD

lb01-iad
2.2.2.1

lb02-iad
2.2.2.2

web01-iad
web01-01-iad
web01-02-iad

web02-iad
web02-01-iad
web02-02-iad

LON

lb01-lon
1.1.1.1

lb02-lon
1.1.1.2

web01-lon
web01-01-lon
web01-02-lon

web02-lon
web02-01-lon
web02-02-lon

*Example scenario
Middle Man

Users

Recursives

Authoritatives

Remember Me?

Example.com?

1.1.1.1
Middle Man Blues

Recursive Hides Actual User

- Good for GDPR!
- Means highly specific routing with DNS especially involving sessions is a bad idea
- Takes a little time to propagate changes, usually a few minutes
- EDNS0 Client Subnet (ECS) helps this, but not widely adopted
Traffic Steering Must Be Primary
Why Primary for Traffic Steering?

- Technically, there is nothing stopping this from happening in the future. The problem is standards.

- Because there is no clear standard, each vendor has to develop Traffic Steering with value-added features independently. This metadata is stored however the vendor decides to store it.

- That metadata may/may not transfer from an XFR, but the Secondary won’t know how to interpret the metadata anyway.
Multiple DNS Dilemma

• Using DNS for Traffic Steering allows failover and steering at the first interaction point for a user

• Each traffic steering is vendor specific, zone will have to be Primary

• DNS is critical, so multiple providers are needed, but Secondary can’t be used to keep the zone files in sync

• Because DNS nameserver selection is based on performance, all zone files must be in sync with same response behavior
Option A - Perform Updates Manually
Option B - Integrate Everything
Option C - What’s Behind Door Number 3?
Focus on the Hostnames That Matter

- Zones contain hundreds of records performing various functions

<table>
<thead>
<tr>
<th>example.com.</th>
<th>10 mail.example.com</th>
</tr>
</thead>
<tbody>
<tr>
<td>@</td>
<td>MX</td>
</tr>
<tr>
<td>@</td>
<td>TXT</td>
</tr>
<tr>
<td>_sip._tls</td>
<td>SRV</td>
</tr>
<tr>
<td>_dmarc</td>
<td>TXT</td>
</tr>
<tr>
<td>api</td>
<td>A</td>
</tr>
<tr>
<td>blog</td>
<td>A</td>
</tr>
<tr>
<td>mail</td>
<td>A</td>
</tr>
<tr>
<td>www</td>
<td>A</td>
</tr>
<tr>
<td>vpn</td>
<td>A</td>
</tr>
</tbody>
</table>

```
example.com.
@      MX      10 mail.example.com
@      TXT     "v=spf1 a mx -all"
_sip._tls SRV    100 1 443 sip.ex.com.
_dmarc   TXT     v=DMARC1; p=none; pct=100;..
api      A       1.1.2.3
blog     A       1.2.3.4
mail     A       6.7.8.9
www      A       2.1.2.3
vpn      A       3.1.2.3
```
Focus on the Hostnames That Matter

• Zones contain hundreds of records performing various functions

• Most people only want traffic steering on some hosts, not the whole domain

| example.com. |
| @ MX 10 mail.example.com |
| @ TXT "v=spf1 a mx -all" |
| _sip._tls SRV 100 1 443 sip.ex.com. |
| _dmarc TXT v=DMARC1; p=none; pct=100;..
api A 1.1.2.3 |
| blog A 1.2.3.4 |
| mail A 6.7.8.9 |
| www A 2.1.2.3 |
| vpn A 3.1.2.3 |
Focus on the Hostnames That Matter

- Zones contain hundreds of records performing various functions
- Most people only want traffic steering on some hosts, not the whole domain
- The whole zone doesn’t need to be primary, only the bits that need traffic steering

```plaintext
example.com.
@     MX       10 mail.example.com
@     TXT      “v=spf1 a mx -all”
_sip._tls SRV   100 1 443 sip.ex.com.
_dmarc TXT      v=DMARC1; p=none; pct=100;..
api    A        1.1.2.3
blog   A        1.2.3.4
mail   A        6.7.8.9
www    A        2.1.2.3
vpn    A        3.1.2.3
```
Enter the Management Zone

- Build zone which contains the intelligent records as a PRIMARY

### Core Zone

<table>
<thead>
<tr>
<th>example.com.</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>@</td>
<td>MX</td>
<td>10 mail.example.com</td>
</tr>
<tr>
<td>@</td>
<td>TXT</td>
<td>“v=spf1 a mx -all”</td>
</tr>
<tr>
<td>_sip._tls</td>
<td>SRV</td>
<td>100 1 443 sip.ex.com</td>
</tr>
<tr>
<td>_dmarc</td>
<td>TXT</td>
<td>v=DMARC1; p=none; pct=100;..</td>
</tr>
<tr>
<td>api</td>
<td>A</td>
<td>1.1.2.3</td>
</tr>
<tr>
<td>blog</td>
<td>A</td>
<td>1.2.3.4</td>
</tr>
<tr>
<td>mail</td>
<td>A</td>
<td>6.7.8.9</td>
</tr>
<tr>
<td>www</td>
<td>A</td>
<td>2.1.2.3</td>
</tr>
<tr>
<td>vpn</td>
<td>A</td>
<td>3.1.2.3</td>
</tr>
</tbody>
</table>

### Management Zone

<table>
<thead>
<tr>
<th>example-mgt.com.</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>api</td>
<td>IN</td>
<td>A</td>
</tr>
<tr>
<td>www</td>
<td>IN</td>
<td>A</td>
</tr>
<tr>
<td>vpn</td>
<td>IN</td>
<td>A</td>
</tr>
</tbody>
</table>
Enter the Management Zone

- Build zone which contains the intelligent records as a PRIMARY
- Change the host from an A record to a CNAME pointing to the corresponding host on the management zone

### Core Zone

<table>
<thead>
<tr>
<th>Domain</th>
<th>Type</th>
<th>Record</th>
</tr>
</thead>
<tbody>
<tr>
<td>example.com</td>
<td>MX</td>
<td>@ 10 mail.example.com</td>
</tr>
<tr>
<td>api</td>
<td>CNAME</td>
<td>api.example-mgt.com</td>
</tr>
<tr>
<td>blog</td>
<td>A</td>
<td>1.2.3.4</td>
</tr>
<tr>
<td>mail</td>
<td>A</td>
<td>6.7.8.9</td>
</tr>
<tr>
<td>www</td>
<td>CNAME</td>
<td><a href="http://www.example-mgt.com">www.example-mgt.com</a></td>
</tr>
<tr>
<td>vpn</td>
<td>CNAME</td>
<td>vpn.example-mgt.com</td>
</tr>
<tr>
<td>_sip._tls</td>
<td>SRV</td>
<td>100 1 443 sip.ex.com</td>
</tr>
<tr>
<td>_dmarc</td>
<td>TXT</td>
<td>v=DMARC1; p=none; pct=100</td>
</tr>
</tbody>
</table>

### Management Zone

<table>
<thead>
<tr>
<th>Domain</th>
<th>Type</th>
<th>Record</th>
<th>IP Address</th>
</tr>
</thead>
<tbody>
<tr>
<td>example-mgt.com</td>
<td>IN</td>
<td>A 1.1.2.3 OR 1.4.5.6</td>
<td></td>
</tr>
<tr>
<td>api</td>
<td>IN</td>
<td>A 1.2.3.4</td>
<td></td>
</tr>
<tr>
<td>www</td>
<td>IN</td>
<td>A 2.1.2.3 OR 2.4.5.6</td>
<td></td>
</tr>
<tr>
<td>vpn</td>
<td>IN</td>
<td>A 3.1.2.3 OR 5.4.5.6</td>
<td></td>
</tr>
</tbody>
</table>
Enter the Management Zone

- Build zone which contains the intelligent records as a PRIMARY
- Change the host from an A record to a CNAME pointing to the corresponding host on the management zone

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Class</th>
<th>A-Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>api</td>
<td>IN</td>
<td>A</td>
<td>1.1.2.3 OR 1.4.5.6</td>
</tr>
<tr>
<td>www</td>
<td>IN</td>
<td>A</td>
<td>2.1.2.3 OR 2.4.5.6</td>
</tr>
<tr>
<td>vpn</td>
<td>IN</td>
<td>A</td>
<td>3.1.2.3 OR 5.4.5.6</td>
</tr>
</tbody>
</table>
Enter the Management Zone

- Build zone which contains the intelligent records as a PRIMARY
- Change the host from an A record to a CNAME pointing to the corresponding host on the management zone
- Flat zone can now be XFR to SECONDARY

<table>
<thead>
<tr>
<th>Zone</th>
<th>Primary</th>
<th>Secondary</th>
</tr>
</thead>
<tbody>
<tr>
<td>CNAME</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Management Zone

<table>
<thead>
<tr>
<th>Host</th>
<th>Type</th>
<th>IP Address</th>
</tr>
</thead>
<tbody>
<tr>
<td>api</td>
<td>IN</td>
<td>1.1.2.3 OR 1.4.5.6</td>
</tr>
<tr>
<td>www</td>
<td>IN</td>
<td>2.1.2.3 OR 2.4.5.6</td>
</tr>
<tr>
<td>vpn</td>
<td>IN</td>
<td>3.1.2.3 OR 5.4.5.6</td>
</tr>
</tbody>
</table>
Enter the Management Zone

- Build zone which contains the intelligent records as a **PRIMARY**
- Change the host from an A record to a CNAME pointing to the corresponding host on the management zone
- Flat zone can now be XFR to **SECONDARY**
- Two management zones are better than one!
Management Zone Options

• Same technique CDNs and other cloud services will use CNAME as a pointer to a FQDN which is easier to change for the provider

• Technique is also deployed by hardware GSLB services common in traditional enterprise

• Two major methods of execution
  – CNAME Method
  – Delegation Method
CNAME Method

• This method has a single management domain for all Traffic Steering / GSLB hosts which is CNAME’d to/from the core zone.

• Could be entirely new domain (example-mgt.com) or could be a new zone delegated from the core domain (mgt.example.com)

• Easy to move services over one at a time, but adding second management zone will require second management zone to replicate ALL services before entering delegation. Could be eased by making a new management zone.

Core Zone

| example.com. | @ | MX | 10 mail.example.com |
| @ | TXT | “v=spf1 a mx -all” |
| _sip._tls | SRV | 100 1 443 sip.ex.com. |
| _dmarc | TXT | v=DMARC1; p=none; pct=100; |
| api | CNAME | api.example-mgt.com |
| blog | A | 1.2.3.4 |
| mail | A | 6.7.8.9 |
| www | CNAME | www.example-mgt.com |
| vpn | CNAME | vpn.example-mgt.com |

Management Zone

| example-mgt.com. | api | IN | A | 1.1.2.3 OR 1.4.5.6 |
| www | IN | A | 2.1.2.3 OR 2.4.5.6 |
| vpn | IN | A | 3.1.2.3 OR 5.4.5.6 |
Delegation Method

- This method has an individual zone for every host. Traffic Steering is desired for.
- Most common with groups which started with only a single host that needed steering, then another... and another...
- Large amounts of administrative overhead in ballooning number of individual zones. Has the benefit that all zones have individual delegations so moving to multiple management zones can be done individually.

---

**Core Zone**

<table>
<thead>
<tr>
<th>example.com.</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>@</td>
<td>MX</td>
</tr>
<tr>
<td>@</td>
<td>TXT</td>
</tr>
<tr>
<td>_sip._tls</td>
<td>SRV</td>
</tr>
<tr>
<td>_dmarc</td>
<td>TXT</td>
</tr>
<tr>
<td>api</td>
<td>NS</td>
</tr>
<tr>
<td>api</td>
<td>NS</td>
</tr>
<tr>
<td>api</td>
<td>NS</td>
</tr>
<tr>
<td>api</td>
<td>NS</td>
</tr>
<tr>
<td>blog</td>
<td>A</td>
</tr>
<tr>
<td>mail</td>
<td>A</td>
</tr>
<tr>
<td>www</td>
<td>NS</td>
</tr>
<tr>
<td>vpn</td>
<td>NS</td>
</tr>
</tbody>
</table>

**Management Zone**

<table>
<thead>
<tr>
<th>api.example.com.</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>@</td>
<td>IN</td>
</tr>
</tbody>
</table>
How Will You Sync the Management Zone?

- While we still have two primary zones, the number and type of records should be greatly reduced.
- Manual is not nearly as scary as it once was.
- Easier to build scripts for only traffic steering vs. day-to-day record changes.
What if the Features Are Different?

• Because there is not a clear-cut standard of what Traffic Steering looks like, the response behavior could be different depending on which provider you hit.

• Most offer State/Country level GeoIP with Weighted Round Robin, and HTTP(S) Health Monitoring. Can be configured approximately the same, but always lowest common denominator.
What if the Features Are Different?

- Goal is to have equivalent, but not necessarily equal experiences across providers.
- Monitoring could be offloaded to a single central source of truth which informs DNS vendors via API calls.
- Keep GeoIP maps large to limit impact from different maps, use DNS for high rough steering to DCs and keep fine tuning to local LBs.
Has Anyone Built This Integration Before?

• Different cloud-native orgs have attempted this across different DNS providers with different goals.

Oracle does not officially endorse any of the following projects:

• Netflix - Denominator
  Original attempt, some code may be stale
  https://github.com/Netflix/denominator

• Github - OctoDNS
  https://github.com/github/octodns

• Spotify Gordon
  Event-driven record management solution.
  https://github.com/spotify/gordon
Happy Little Clouds

Let's reflect on what we accomplished

- Primary-Secondary DNS with diverse delegation enables multiple DNS solutions in delegation for diversity and resiliency
Happy Little Clouds

Let's reflect on what we accomplished

- Primary-Secondary DNS with diverse delegation enables multiple DNS solutions in delegation for diversity and resiliency
- Using DNS-based Traffic Steering solutions allows first-in-the-chain routing, pushing users to live assets
Happy Little Clouds

**Let's reflect on what we accomplished**

- Primary-Secondary DNS with diverse delegation enables multiple DNS solutions in delegation for diversity and resiliency
- Using DNS-based Traffic Steering solutions allows first-in-the-chain routing, pushing users to live assets
- Agnostic routing enables deployments in multicloud or hybrid cloud deployments
10 Minute Break
Workshop!

Optimizing your Zonefile for Multiplatform DNS
The Soccer Mom of Infrastructure

- Look, let’s call a spade a spade. DNS is business-critical, but not exactly the best test drive, amirite?
The Soccer Mom of Infrastructure

- Look, let’s call a spade a spade. DNS is business-critical, but not exactly the best test drive, amirite?
- We will be going through the major design considerations and drafting a proposal for how your organization might want to operate
The Soccer Mom of Infrastructure

• Look, let’s call a spade a spade. DNS is business-critical, but not exactly the best test drive, amirite?

• We will be going through the major design considerations and drafting a proposal for how your organization might want to operate

• BONUS – if some lucky volunteer is feeling bold... We could add Dyn as secondary with a management zone and traffic steering. Sky is the limit here folks.
Workshop Project

1. **Determine Zonefile Management and Delegation**
   Based on what you have today, what will make the most sense to the business? Do you have budget for 1 or 2 providers? Will you keep management internal or move that to a cloud solution with a UI?
Workshop Project

1. Determine Zonefile Management and Delegation

2. Determine Which Hosts Need Traffic Steering
   Even if this is not currently the case today, it is good to understand what FQDNs of the zone are business-critical that might warrant advanced services in the future
Workshop Project

1. **Determine Zonefile Management and Delegation**

2. **Determine Which Hosts Need Traffic Steering**

3. **Determine Management Zone Style**
   - Does it make sense to utilize a CNAME method, or a Delegation method management zone? What will the management zone be? Is it easier to get new domains in your org or delegate from a known one?
Workshop Project

1. **Determine Zonefile Management and Delegation**

2. **Determine Which Hosts Need Traffic Steering**

3. **Determine Management Zone Style**

4. **Determine Primary -> Secondary Sync Parameters**
   - Does your current DNS support the ability to have a Secondary? Will you use TSIG for security? What are the Master IPs? Who do you see about opening a firewall?
Workshop Project

1. Determine Zonefile Management and Delegation

2. Determine Which Hosts Need Traffic Steering

3. Determine Management Zone Style

4. Determine Primary -> Secondary Sync Parameters

5. Determine Management Zone Sync Parameters
   How will the management zones stay in sync? Are there few enough to do manually? Who owns the playbook? Will you automate? Fork existing or build new? Who owns development?
Workshop Project

1. Determine Zonefile Management and Delegation
2. Determine Which Hosts Need Traffic Steering
3. Determine Management Zone Style
4. Determine Primary -> Secondary Sync Parameters
5. Determine Management Zone Sync Parameters
Appendix
Sources and Links

• www.dyn.com

• 2012 ~ Tom Daly ~ The Architecture Behind Super Fast DNS - Routing, Redundancy & Geography https://youtu.be/iCfuKZUyTko


• https://www.internetworldstats.com/stats.htm
Vendors make it hard