



# DNS Operations and DNSSEC Tutorial

Champika Wijayatunga | SANOG30 - India | July 12-13, 2017

# Agenda

1

DNS Concepts

2

Registry/Registrar  
Model

3

ccTLD Best  
Practices

4

Managing Zones

5

DNS Security and  
DNSSEC

6

Config Demos and  
Tools



# DNS Overview

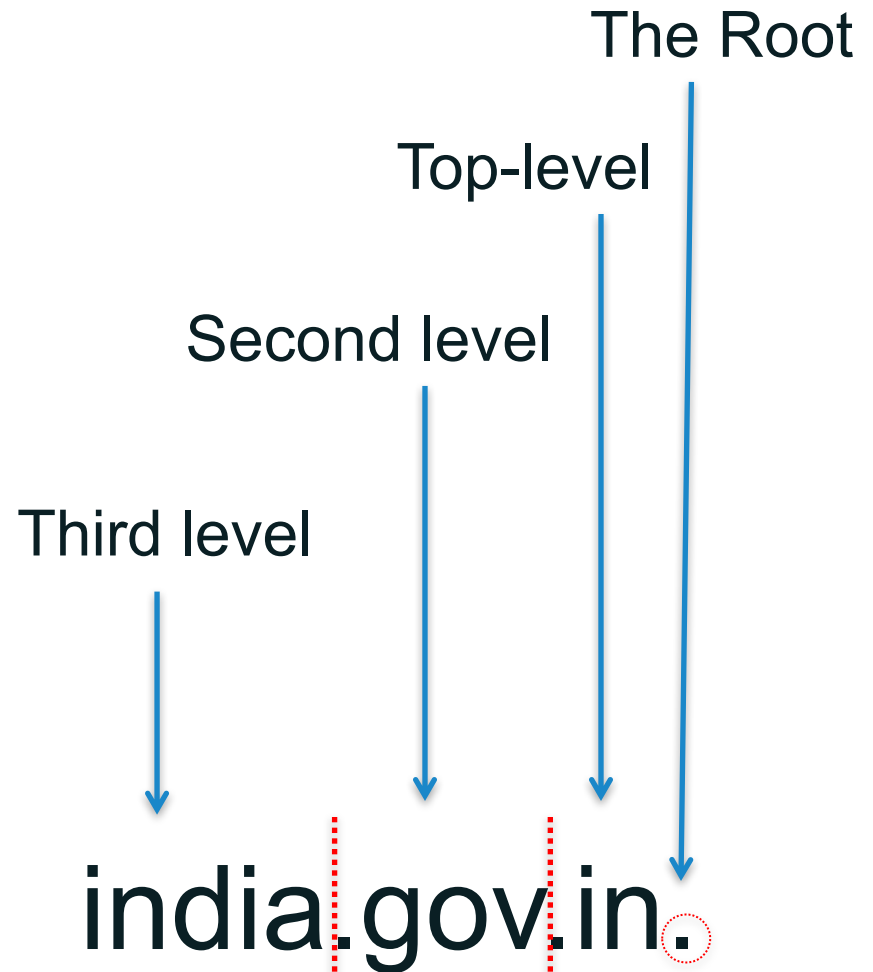


# History

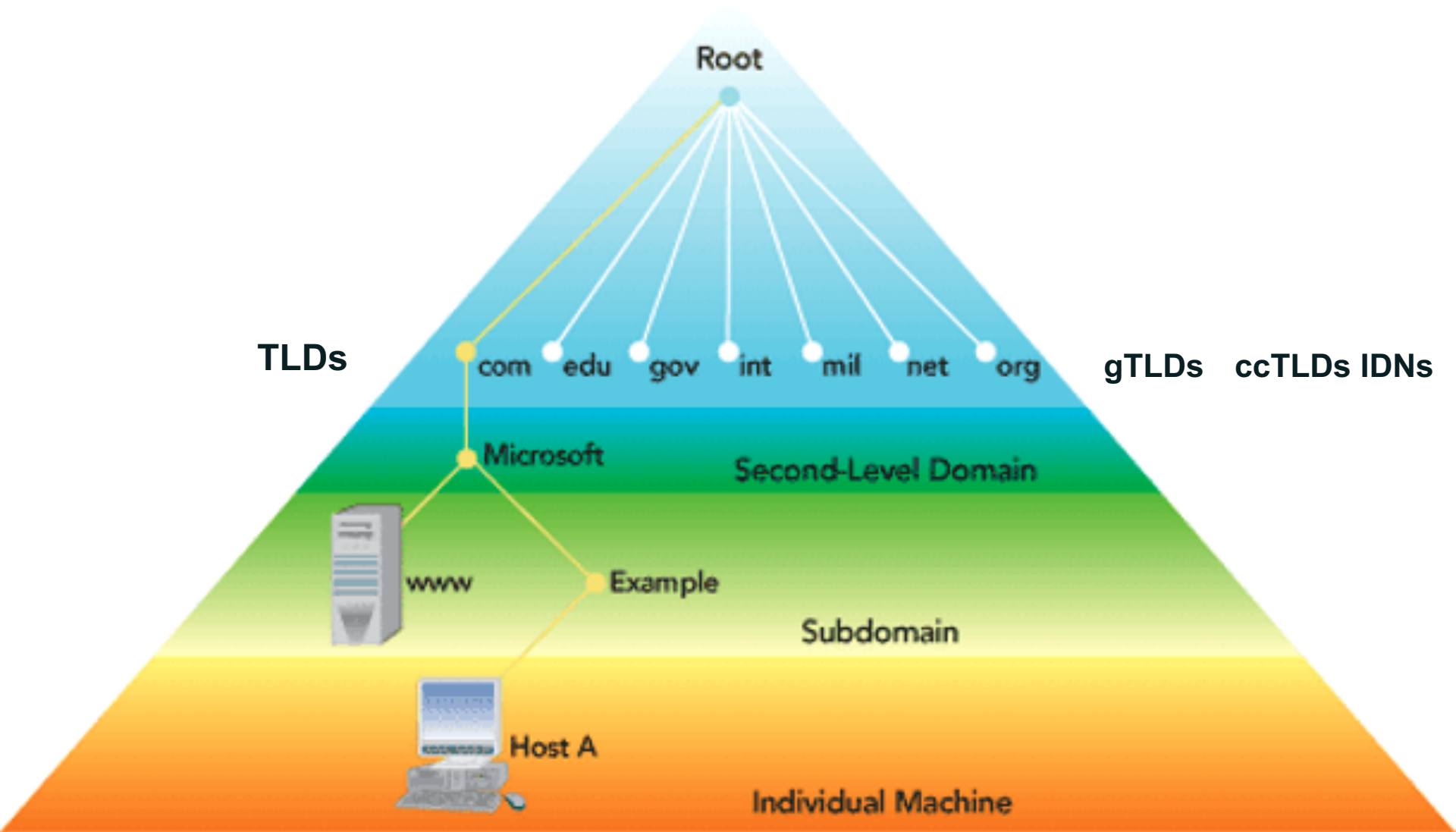
- 1983 DNS was designed/invented by Paul Mockapetris (RFC882 & 883)
- 1984 Berkeley Internet Name Domain (BIND) Server developed  
Original Seven Generic TLDs (.com, .edu, .gov, .int, .mil, .net, and .org)
- 1985 First country codes assigned .us, .uk, and .il
- 1986 .au, .de, .fi, .fr, .jp, .kr, .nl and .se
- 1987 RFC1034 (Considered the first full DNS Specification)
- ..... Country Code TLDs continue to be added....
- 2000 Seven new TLDs added (.aero, .coop, .museum, .biz, .info, .name, and .pro)
- 2012 New round of applications for gTLDs opened by ICANN



# Domain Name's Structure



# Domain Name System (DNS)



# Key Elements of DNS

**Resolution**

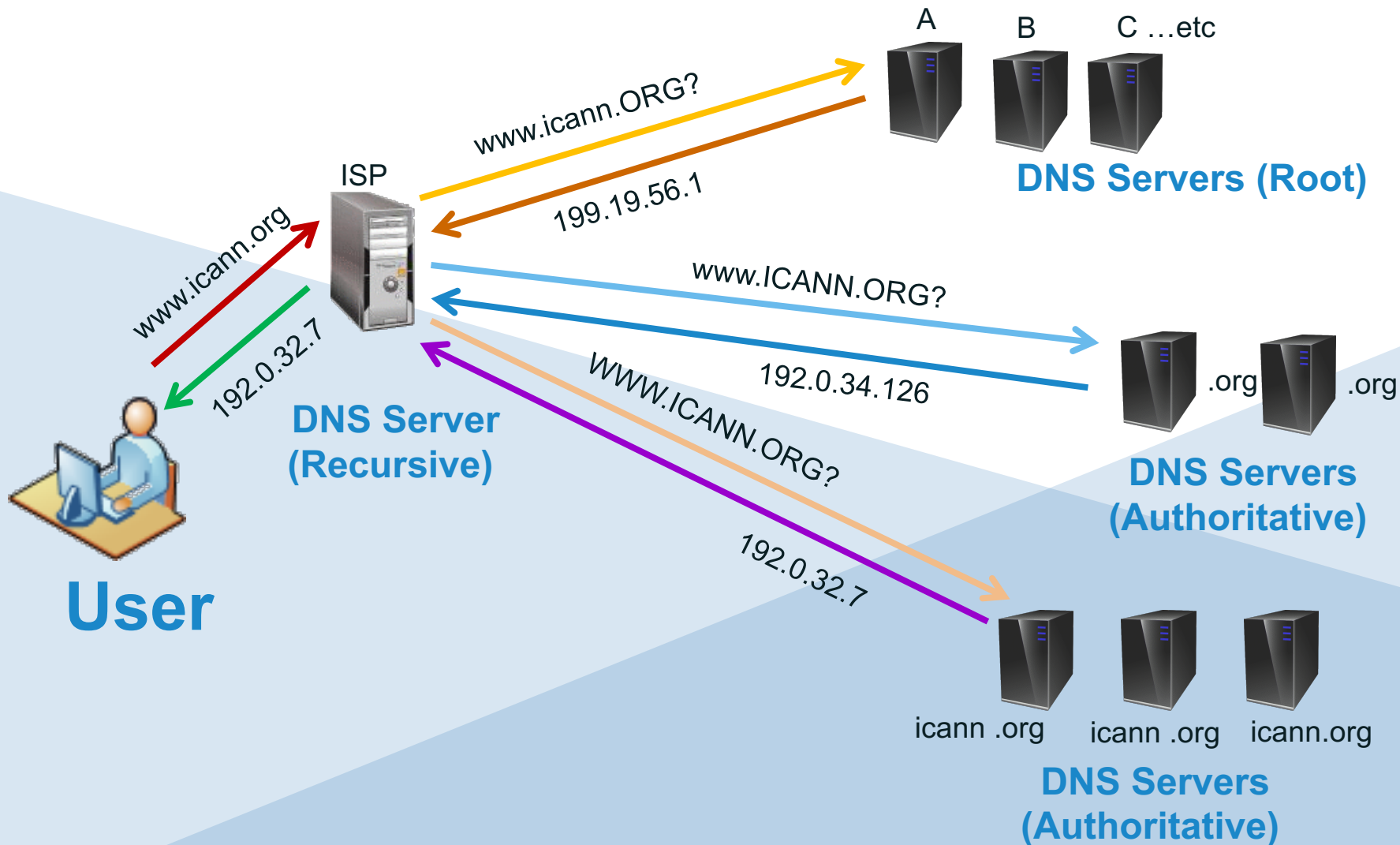
**Distributed**

**Hierarchical**

**Consistent**



# DNS Resolution





# Root Server Operation

# What do the Root-Server Operators do?

- Copy a very small database, the content of which is currently decided by PTI (formerly IANA)
- Put that database in the servers called 'Root Servers.'
- Make the data available to all Internet users
- Work stems from a common agreement about the technical basis
  - Everyone on the Internet should have equal access to the data
  - The entire root system should be as stable and responsive as possible



# What do the Root-Server Operators do not do?

- Interfere with the content of the database
  - E.g. run the printing presses, but don't write the book
- Make policy decisions
  - Who runs TLDs, or which domains are in them
  - What systems TLDs use, or how they are connected to the Internet

# Who are the Root Server operators?

- Not "one group", 12 distinct operators
- Operational and technical cooperation
- Participate in RSSAC as advisory body to ICANN
- High level of trust among operators
  - Show up at many technical meetings, including IETF, ICANN, RIR meetings, NOG meetings, APRICOT etc.

# How Secure are the Root Servers?

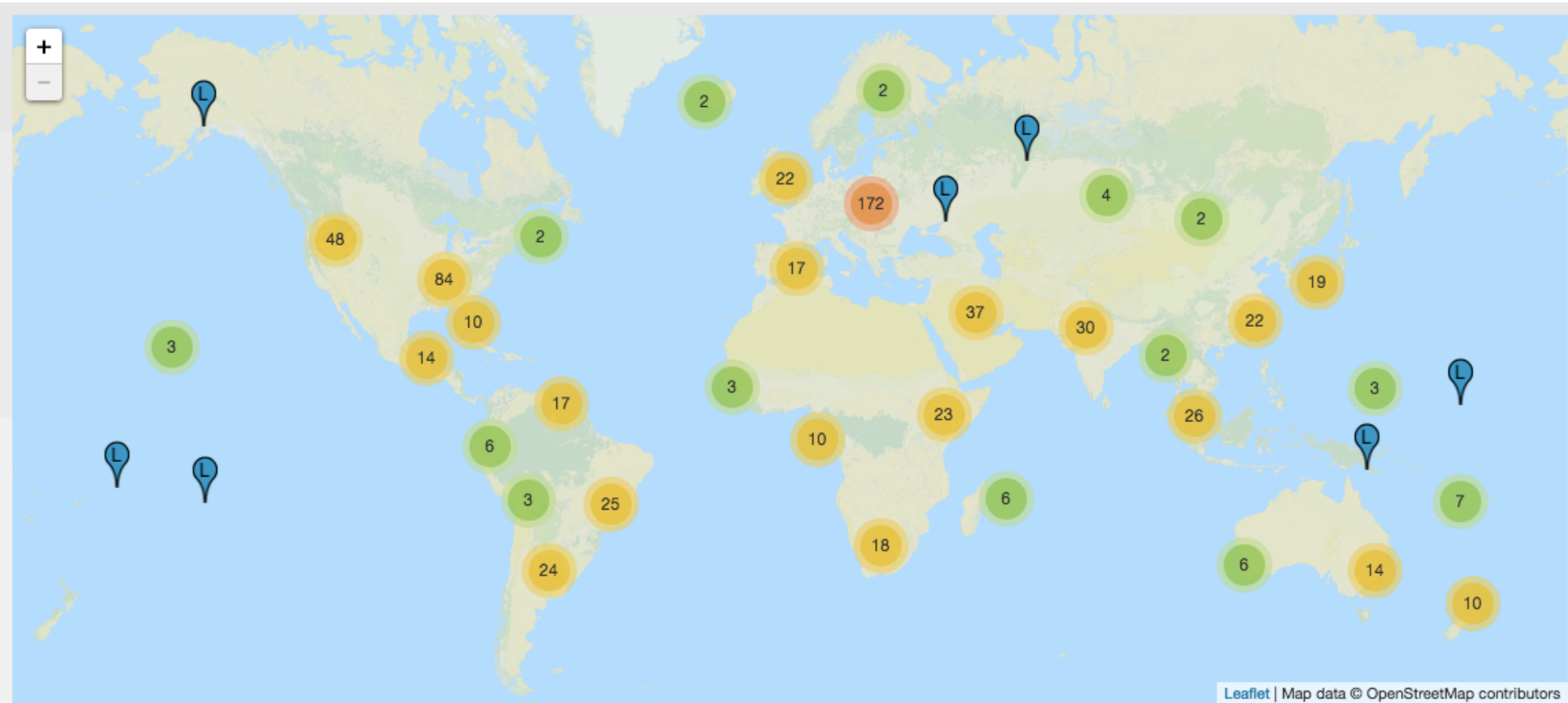
- Physically protected
- Tested operational procedures
- Experienced, professional, trusted staff
- Defense against major operational threat – i.e. DDoS.
  - Anycast
    - Setting up identical copies of existing servers
    - Same IP address
    - Exactly the same data.
    - Standard Internet routing will bring the queries to the nearest server
    - Provides better service to more users.



# Avoiding Common Misconceptions

- Not all internet traffic goes through a root server
- Not every DNS query is handled by a root server
- Root servers are not managed by volunteers as a hobby
  - Professionally managed and well funded
- No single organization(neither commercial nor governmental) controls the entire system
- The "A" server is not special.
- Root Server Operators don't administrate the zone content
  - They publish the IANA-approved data

# Root Server Distribution



Leaflet | Map data © OpenStreetMap contributors



# Recursive and Authoritative Servers

# DNS Servers

- DNS is a distributed database
- Types of DNS servers
  - DNS Authoritative
    - Primary (Master)
    - Secondary (Slaves)
  - DNS Resolver
    - Recursive
    - Cache
    - Stub resolver

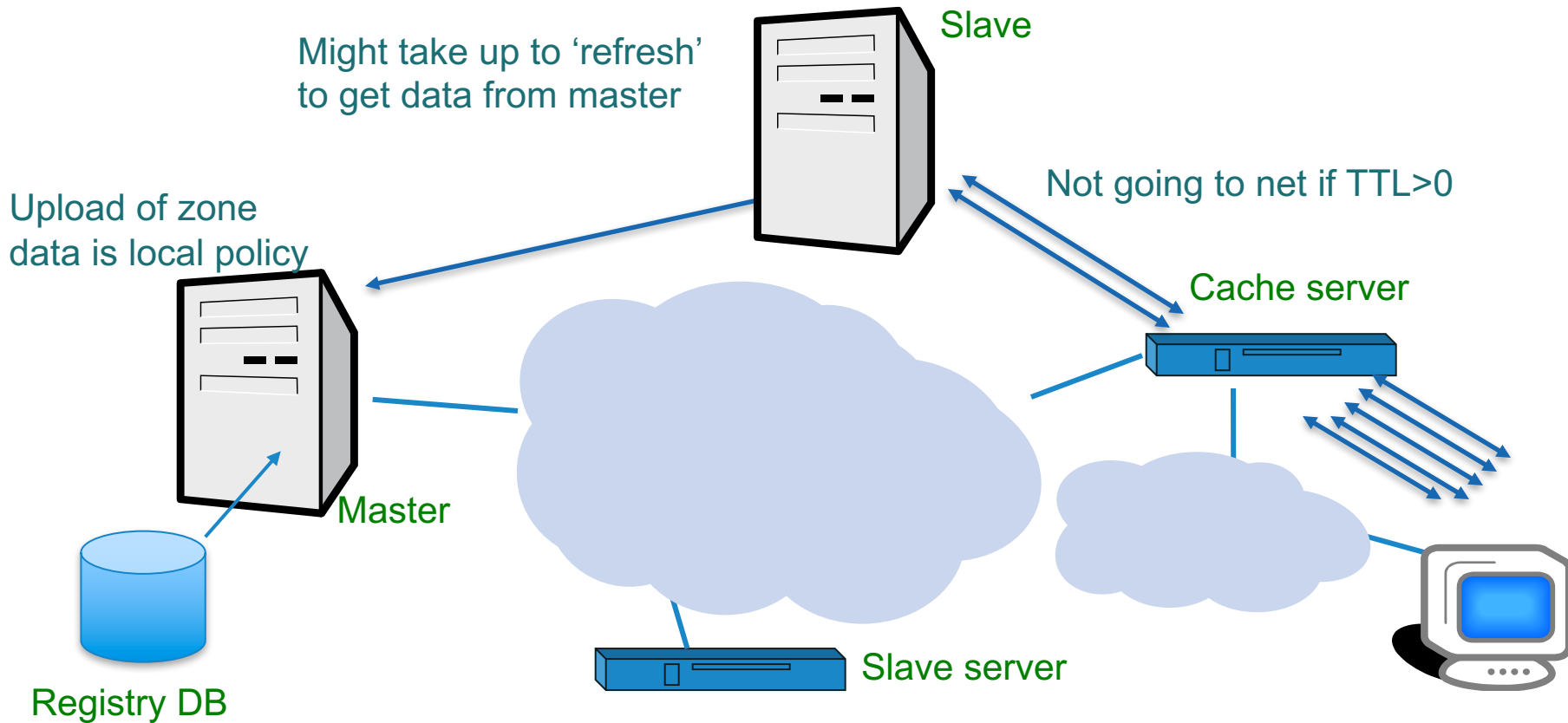
# Operational elements of the DNS

- Authoritative Name Servers host zone data
  - The set of “DNS data” that the registrant publishes
- Recursive Name Resolvers (“resolvers”)
  - Systems that find answers to queries for DNS data
- Caching resolvers
  - Recursive resolvers that not only find answers but also store answers locally for “TTL” period of time
- Client or “stub” resolvers
  - Software in applications, mobile apps or operating systems that query the DNS and process responses



# Places where DNS data lives

Changes do not propagate instantly



# Delegating a Zone

- Delegation is passing of authority for a subdomain to another party
- Delegation is done by adding NS records
  - Ex: if example.in wants to delegate training.example.in

```
training.example.in.    NS ns1.training.example.in.
training.example.in.    NS ns2.training.example.in.
```
- Now how can we go to ns1 and ns2?
  - We must add a Glue Record

# Glue Record

- Glue is a 'non-authoritative' data
- Don't include glue for servers that are not in the sub zones

Only this record needs glue

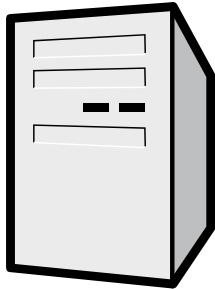
```
training.example.in. NS ns1.training.example.in.  
training.example.in. NS ns2.training.example.in.
```

```
training.example.in. NS ns2.another_example.net.  
training.example.in. NS ns1.another_example.net.
```

Glue  
Record

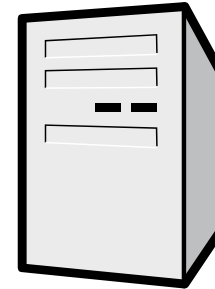
```
Ns1.training.example.in. A 10.0.0.1  
Ns2.training.example.in. A 10.0.0.2
```

# Delegating a child from a parent zone



**ns.example.in**

1. Add NS records and glue
2. Make sure there is no other data from the training.example.in. zone in the zone file



**ns.training.example.in**

1. Setup minimum two servers
2. Create zone file with NS records
3. Add all training.example.in data

The background of the slide is a solid orange color. Overlaid on this is a stylized world map. The map is formed by a complex network of white nodes (small circles) connected by thin white lines, creating a mesh-like structure that follows the general outline of the continents. The nodes are more densely packed in some areas, particularly in North America and Europe, and more sparse in others, like Africa and South America. The overall effect is a digital, network-oriented representation of the world.

# Recursive and Authoritative Server Demo

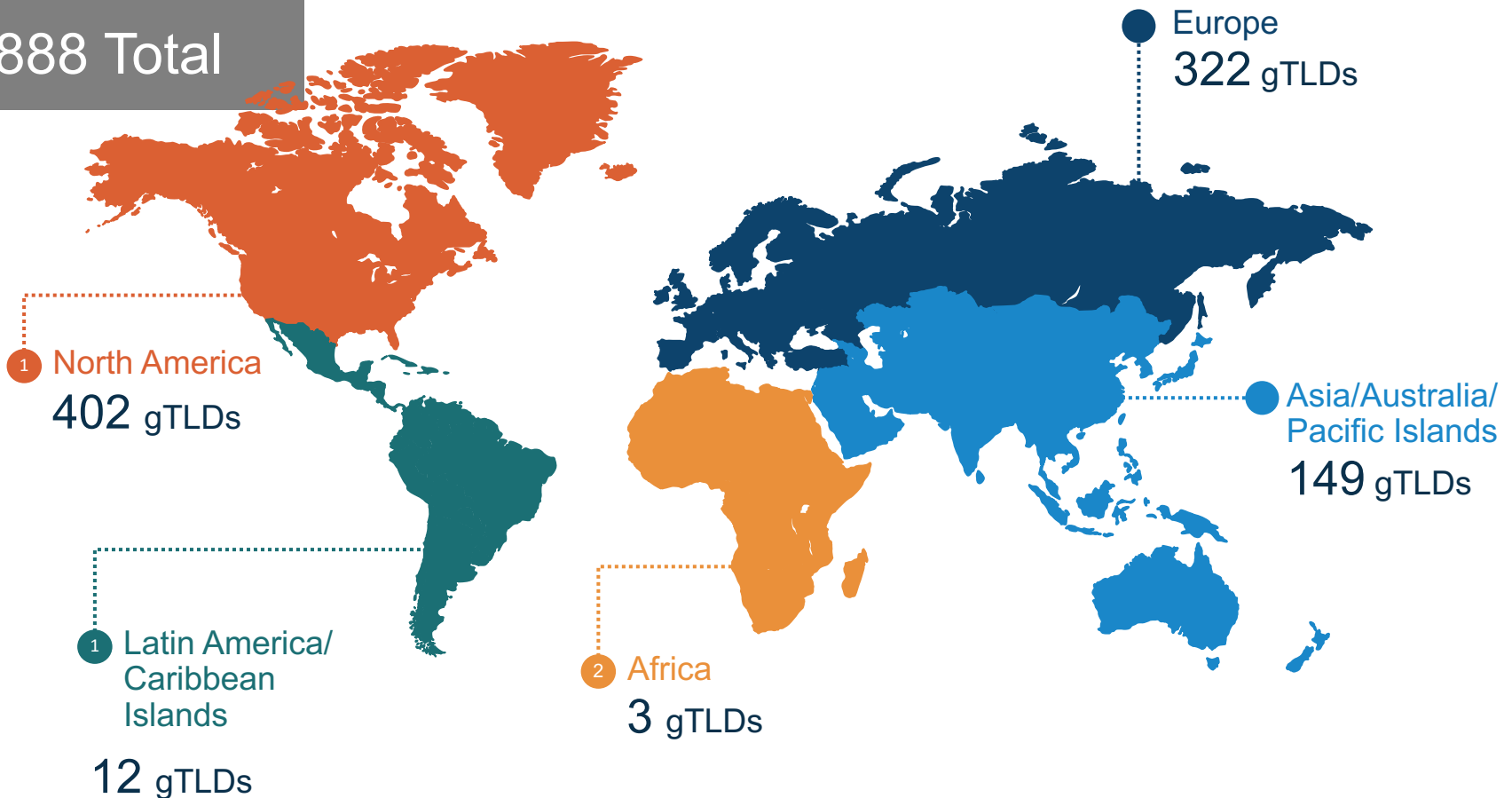


A world map where the continents are defined by a complex network of white nodes and connecting lines, set against a solid teal background. The nodes vary in size, and the lines represent connections between them, creating a digital or network-like representation of the world's geography.

# Registry, Registrar Model

# Regional Distribution of Delegated gTLDs

888 Total

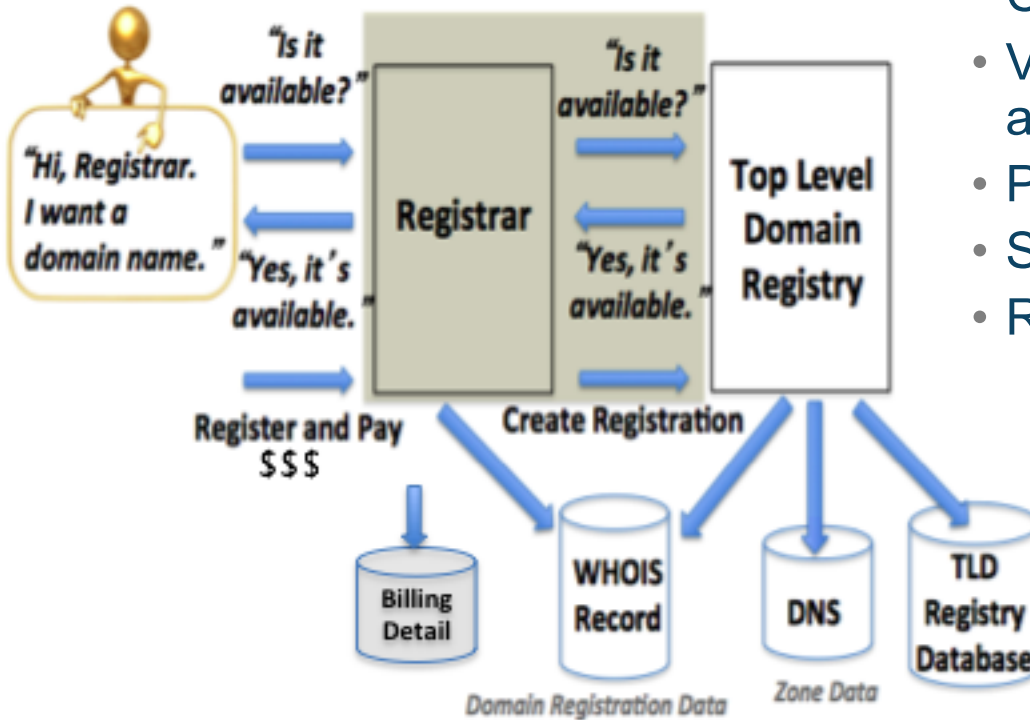


Data as of January 2016  
Categorized by ICANN  
region

# The Registry/Registrar Ecosystem



# Domain Name Registration



How to register a domain:

- Choose a string e.g., example
- Visit a registrar to check string availability in a TLD
- Pay a fee to register the name
- Submit registration information
- Registrar and registries manage:
  - “string” + TLD (managed in registry DB)
  - Contacts, DNS (managed in Whois)
  - DNS, status (managed in Whois DBs)
  - Payment information





# Internationalized Domain Names



# What are IDNs?

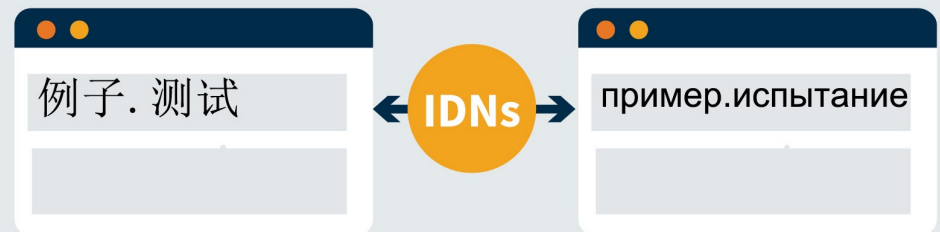
**An Internationalized Domain Name (IDN) uses a particular encoding and format to allow a wider range of scripts to represent domain names.**

Until late 2009, Top-Level Domains were restricted to only the Latin letters a to z without accents or symbols. After 2009, IDN TLDs were introduced in other scripts including Arabic, Chinese and Cyrillic scripts.

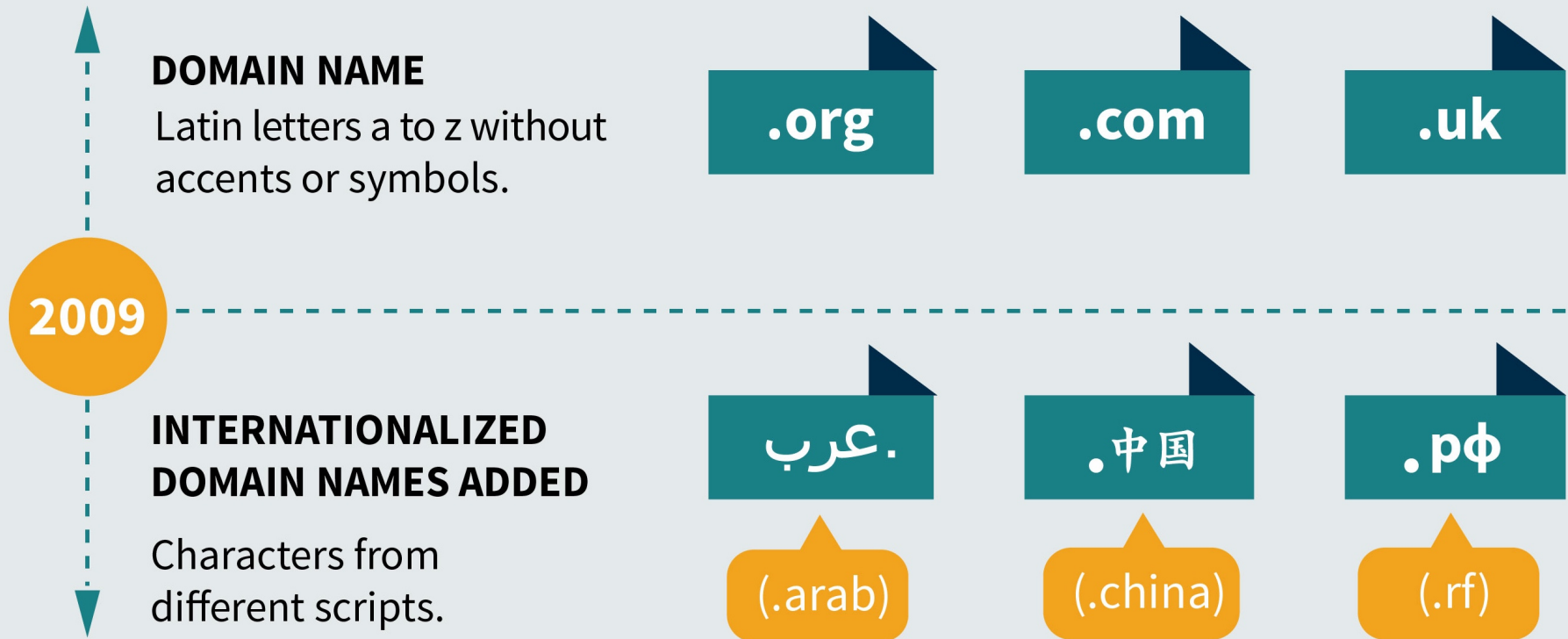
**IDN TLDs can be either ccTLDs or gTLDs.**

## Internationalized Domain Names

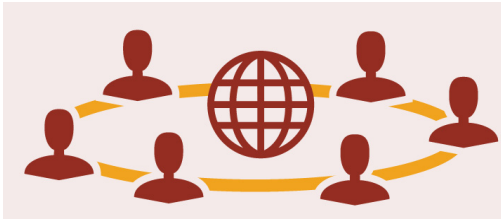
Domain names with non-Latin characters or Latin characters beyond letters (a to z) digits (0 to 9) and hyphen (-), as allowed by relevant protocols.



# What has changed with Top-Level Domains?



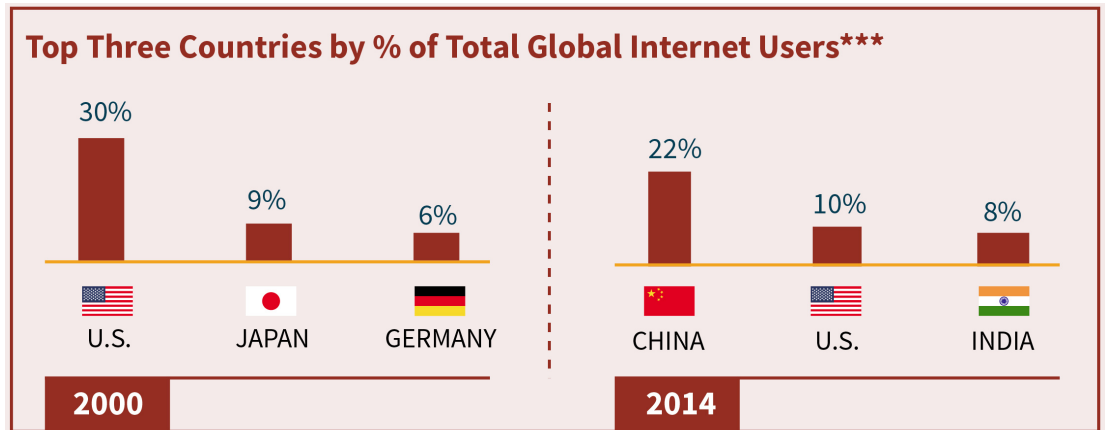
# Why Internationalize Domain Names?



More and more people around the world, once unconnected, are online.



IDNs allow people around the world to access domain names in their local languages.



The background of the slide is a teal color. Overlaid on this is a stylized world map. The map is formed by a network of white dots of varying sizes, connected by thin white lines. The dots are more densely packed in some areas, particularly in North America and Europe, and more sparse in others. The overall effect is a digital, interconnected representation of the world's geography.

# Best practices in ccTLD Management

# Who Currently Operate ccTLDs

- Many of the ccTLDs were assigned in the 1980's.
- They tended to be assigned to whoever was involved in building the Internet in a specific country
- Some changed hands over the years

What types of organisations?

- Universities
- ISPs/Telcos
- Regulators
- Dedicated entities

<http://www.iana.org/domains/root/db>



# What do I mean by “ccTLD policies”

- Anything that defines how and by whom names can be registered.
- Typically ccTLDs have no contract with ICANN and are bound by local rather than ICANN policies
- Can participate in global discussion through ICANN’s ccNSO
  - <http://ccnso.icann.org>

# There is no ONE model for ccTLDs

- Different models work well in different environments.
- This is driven by many things including operational considerations on the ground, local business practices and local culture.
- Policy and operations of a ccTLDs are often built over time and reflect the local environment.

# Who should decide the policies

- Whoever has the role of Sponsoring organisation has the role of ensuring that policies are developed and implemented.
- Many ccTLDs have a model that follow a multi-stakeholder Solution.
- This can take many forms from formal “Policy boards” to processes for gathering public input.
- Often inclusive of Government, Industry and Civil Society as well as registrants

- Which model?

Direct registration

- No middle man - easier to control most aspects of Registration

Registry-registrar model

- Requires an interface between registry and registrar
- Offloads end-user interface from registry

Both

- **Scope of Registrations?**

Local or Global?

There are examples of ccTLDs of both types decide which best serves the community

- Consider that the legal implications are different
- Consider that the risks are different



- **Dispute Resolution:**

Ensure that local law prevails?

You don't want to be arguing in foreign courts

Alternate Dispute Resolution (ADR)?

Design to be lightweight!

UDRP is often used as a base model

<http://www.icann.org/udrp/udrp.htm>

# Some discussions

- Who runs the technical operations?

This is really a business decision.

Policy can define the type of organisation but business decisions should guide the actual choice.

- Technology choices

These are generally operational matters.

The important factor to ensure that the “operator” is bound by the policies created and that choices they make meet those requirements.

# Outsourcing

- There are an increasing number of companies that will provide services to TLD managers.
  - Whole registry back-end providers
  - Authoritative name server providers
- ccTLD managers should understand the basics of how to run the services themselves before they outsource them.
  - Allows you to manage and monitor performance of suppliers
  - Have a back-up strategy! What if your supplier fails?

A world map where the continents are defined by a complex network of white dots and thin white lines, set against a solid teal background. The dots vary in size and are interconnected by a web of lines, creating a digital or network-like appearance of the globe.

# Operational Decisions

## What does it take to run a TLD?

# Technical Requirements for a TLD

- Networks and Servers (redundant)
- Back office systems.
- Physical and Electronic Security
- Quality of Service (24/ 7 availability!)
- Name Servers
- DNS software (BIND, NSD, etc.)
- Registry software
- Diagnostic tools (ping, traceroute, zonecheck, dig)
- Registry Registrar Protocol



# Name Server Considerations

- Support technical standards
- Handle load multiple times the measured peak
- Diverse bandwidth to support above
- Must answer authoritatively
- Turn off recursion!
- Should “NOT” block access from a valid Internet hosts

# Secondary name server choice

Diversity, diversity and diversity!

- Don't place all on the same LAN/building/segment
- Network diversity
- Geographical diversity
- Institutional diversity
- Software and hardware diversity

# Security, Stability & Resiliency Considerations

- Physical security
  - Deploy stringent access controls
  - Fire detection and retardation
  - Other environmental sensors (Flood, Humidity etc.)
  - Power continuity for 48 hours (or more)
- Backups
  - Multiple secure copies locally and offsite
  - Test, test and test!!

# Separations of Services

- Registries generally start small and evolve
- Separation of services means separating the logical functions and elements of the registry
- Two key benefits:
  - **SECURITY:** Clear separation of services is a manner in which to create logical security zones
  - **SCALABILITY:** You can scale only the services that need to grow as they need to grow

# Know your SLAs

- Functioning name servers are the most critical/visible service
- All other services also need to be considered
  - Billing
  - Whois server, webservers
  - Registrar APIs
- Consider your service level targets and how you will meet them
- DNS servers always on, other systems mostly on?



# When it all goes wrong

- DNS is a known target for hackers.
- You will be targeted at some point!
- Have plans in place to deal with attacks, failures and disasters.
- Test those plans regularly!

The background of the slide is a solid orange color. Overlaid on this is a stylized world map. The map is constructed from a network of small white dots connected by thin white lines, creating a mesh-like structure that outlines the continents. The dots are of varying sizes, and the lines are thin and light-colored. The overall effect is a digital, interconnected representation of the world's geography.

# Managing Zones

# DNS Resource Records (RR)

- Unit of data in the Domain Name System
- Define attributes for a domain name

<i>Label</i>	<i>TTL</i>	<i>Class</i>	<i>Type</i>	<i>RData</i>
www	3600	IN	A	192.168.0.1

- Most common types of RR
  - A
  - AAAA
  - NS
  - SOA
  - MX
  - CNAME



A world map where the continents are defined by a complex network of white nodes and connecting lines, set against a dark blue background. The nodes vary in size, and the lines represent connections between them, creating a digital or network-like appearance of the globe.

# Forward Zones

# What is a DNS zone *data*?

- DNS zone data are hosted at an authoritative name server
  - Each “cut” has zone data (root, TLD, delegations)
- DNS zones contain resource records that describe
  - name servers,
  - IP addresses,
  - Hosts,
  - Services
  - Cryptographic keys & signatures...

```
$TTL      86400 ; 24 hours could have been written as 24h or 1d
; $TTL used for all RRs without explicit TTL value
$ORIGIN   example.com.
@ 1D      IN  SOA  ns1.example.com. hostmaster.example.com. (
                                2002022401 ; serial
                                3H ; refresh
                                15 ; retry
                                1w ; expire
                                3h ; minimum
                                )
                                IN  NS   ns1.example.com. ; NS in the domain bailiwick
                                IN  NS   ns2.smokeyjoe.com. ; NS external to domain
                                IN  MX   10 mail.another.com. ; external mail provider
;
; Sender policy framework with hard fail
; Use A and MX resource records for verification and google too
;
example.com. IN TXT "v=spf1 a mx include:google.com -all"
;
; server host definitions
;
ns1          IN  A      192.168.0.1      ;name server definition
www         IN  A      192.168.0.2      ;web server definition
;
; web and ftp server on same address
;
ftp         IN  CNAME  www.example.com. ;ftp server definition
;
; endpoint or non server domain hosts
;
mikeslaptop IN  A      192.168.0.3
fredsipad   IN  A      192.168.0.4
```

Only US ASCII-7 letters, digits, and hyphens can be used as zone data.

In a zone, IDNs strings begin with XN--



# Common DNS Resource Records

```
$TTL      86400 ; 24 hours could have been written as 24h or 1d
; $TTL used for all RRs without explicit TTL value
$ORIGIN example.com.
@ 1D      IN  SOA  ns1.example.com. hostmaster.example.com. (
                2002022401 ; serial
                3H ; refresh
                15 ; retry
                1w ; expire
                3h ; minimum
        )
        IN  NS   ns1.example.com. ; NS in the domain bailiwick
        IN  NS   ns2.smokeyjoe.com. ; NS external to domain
        IN  MX   10 mail.another.com. ; external mail provider
;
; Sender policy framework with hard fail
; Use A and MX resource records for verification and google too
;
example.com. IN  TXT  "v=spf1 a mx include:google.com -all"
;
; server host definitions
;
ns1          IN  A    192.168.0.1      ;name server definition
www         IN  A    192.168.0.2      ;web server definition
;
; web and ftp server on same address
;
ftp         IN  CNAME www.example.com. ;ftp server definition
;
; endpoint or non server domain hosts
;
mikeslaptop IN  A    192.168.0.3
fredsipad   IN  A    192.168.0.4
```

## Time to live (TTL)

- *How long RRs are accurate*

## Start of Authority (SOA) RR

- *Source: zone created here*
- *Administrator's email*
- *Revision number of zone file*

## Name Server (NS)

- *IN (Internet)*
- *Name of authoritative server*

## Mail Server (MX)

- *IN (Internet)*
- *Name of mail server*

## Sender Policy Framework (TXT)

- *Authorized mail senders*

# Common DNS Resource Records

```
$TTL      86400 ; 24 hours could have been written as 24h or 1d
; $TTL used for all RRs without explicit TTL value
$ORIGIN example.com.
@ 1D      IN  SOA  ns1.example.com. hostmaster.example.com. (
                2002022401 ; serial
                3H ; refresh
                15 ; retry
                1w ; expire
                3h ; minimum
        )
        IN  NS   ns1.example.com. ; NS in the domain bailiwick
        IN  NS   ns2.smokeyjoe.com. ; NS external to domain
        IN  MX   10 mail.another.com. ; external mail provider
;
; Sender policy framework with hard fail
; Use A and MX resource records for verification and google too
;
example.com. IN  TXT  "v=spf1 a mx include:google.com -all"
;
; server host definitions
;
ns1          IN  A    192.168.0.1      ;name server definition
www         IN  A    192.168.0.2      ;web server definition
;
; web and ftp server on same address
;
ftp         IN  CNAME www.example.com. ;ftp server definition
;
; endpoint or non server domain hosts
;
mikeslaptop IN  A    192.168.0.3
fredsipad   IN  A    192.168.0.4
```

## Name server address record

- *NS1 (name server name)*
- *IN (Internet)*
- *A (IPv4) \* AAAA is IPv6*
- *IPv4 address (192.168.0.1)*

## Web server address record

- *www (world wide web)*
- *IN (Internet)*
- *A (IPv4) \* AAAA is IPv6*
- *IPv4 address (192.168.0.2)*

## File server address record

- *FTP (file transfer protocol)*
- *IN (Internet)*
- *CNAME means “same address spaces and numbers as www”*

# IPv6 in the DNS

```
;; domain.edu
$TTL          86400
@             IN      SOA      ns1.domain.edu. root.domain.edu. (
                2015050501    ; serial - YYYYMMDDXX
                21600         ; refresh - 6 hours
                1200         ; retry - 20 minutes
                3600000       ; expire - long time
                86400)       ; minimum TTL - 24 hours

;; Nameservers
    IN NS ns1.domain.edu.
    IN NS ns2.domain.edu.

;; Hosts with just A records
host1      IN  A    1.0.0.1

;; Hosts with both A and AAAA records
host2      IN  A    1.0.0.2
          IN  AAAA  2001:468:100::2
```

A world map where the continents are defined by a complex network of white nodes and connecting lines, set against a dark blue background. The nodes vary in size, and the lines represent connections between them, creating a mesh-like structure that outlines the major landmasses.

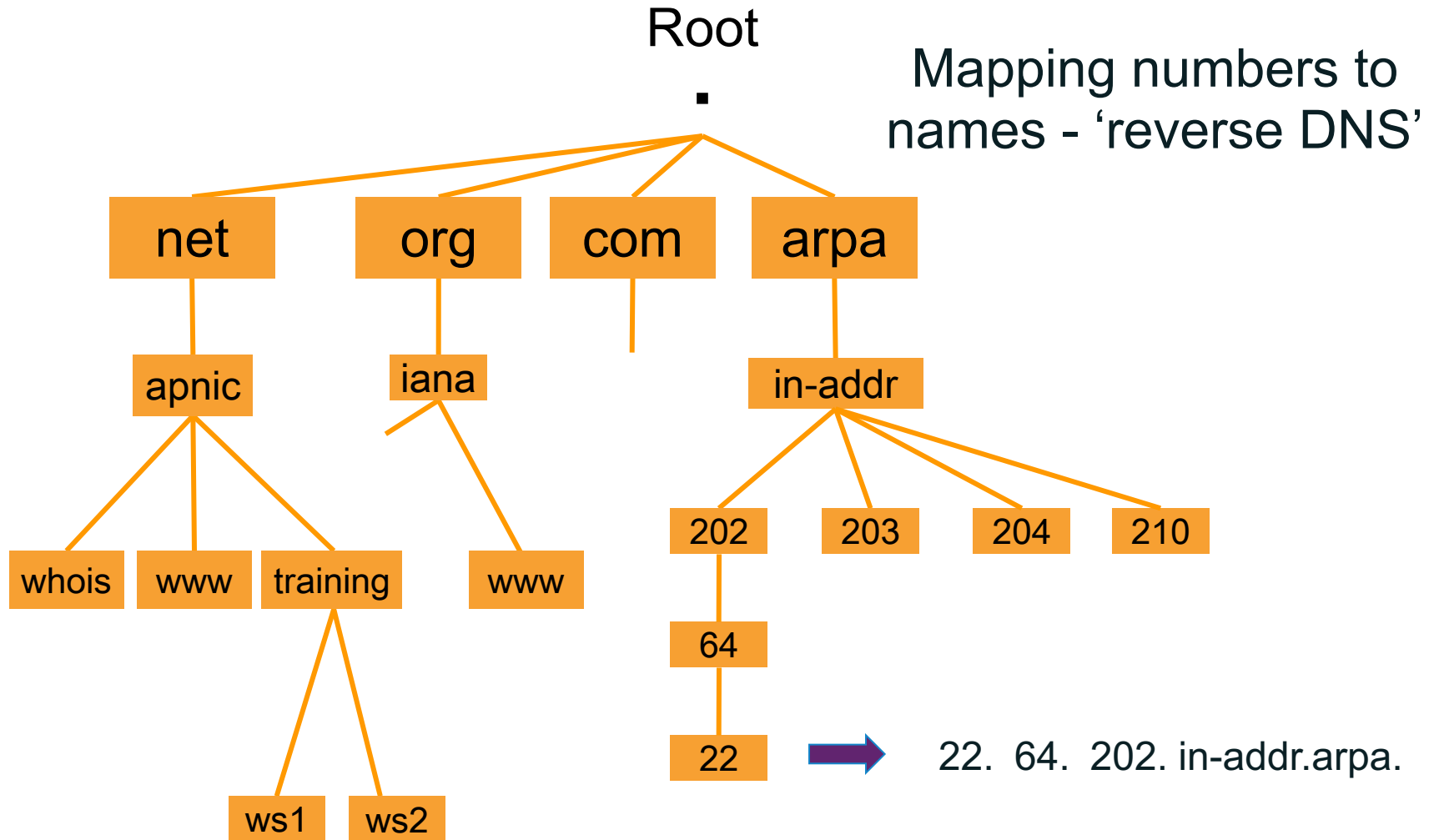
# Reverse Zones

# Why bother?

- Service denial
  - only allow access when fully reverse delegated
  - Example: anonymous ftp
- Diagnostics
  - Used in tools such as traceroute
- Spam identifications
  - Failed reverse lookup results in a spam penalty score
- RIR/NIR Registration responsibilities
  - Make sure all your address spaces are properly reverse delegated



# Principles – Reverse DNS Tree



# Creating Reverse Zones

- Same as creating a forward zone file
  - SOA and initial NS records are the same as forward zone
- Create additional PTR records
- In addition to the forward zone files, you need the reverse zone files
  - Ex: for a reverse zone on a 203.176.189.0/24 block, create a zone file and name it as “db.203.176.189” (make it descriptive)

# Reverse Zone Example

```
$ORIGIN 1.168.192.in-addr.arpa.  
@ 3600 IN SOA test.company.org. (  
    sys\admin.company.org.  
    2017021301 ; serial  
    1h        ; refresh  
    30M       ; retry  
    1W        ; expiry  
    3600 )    ; neg. answ. ttl
```

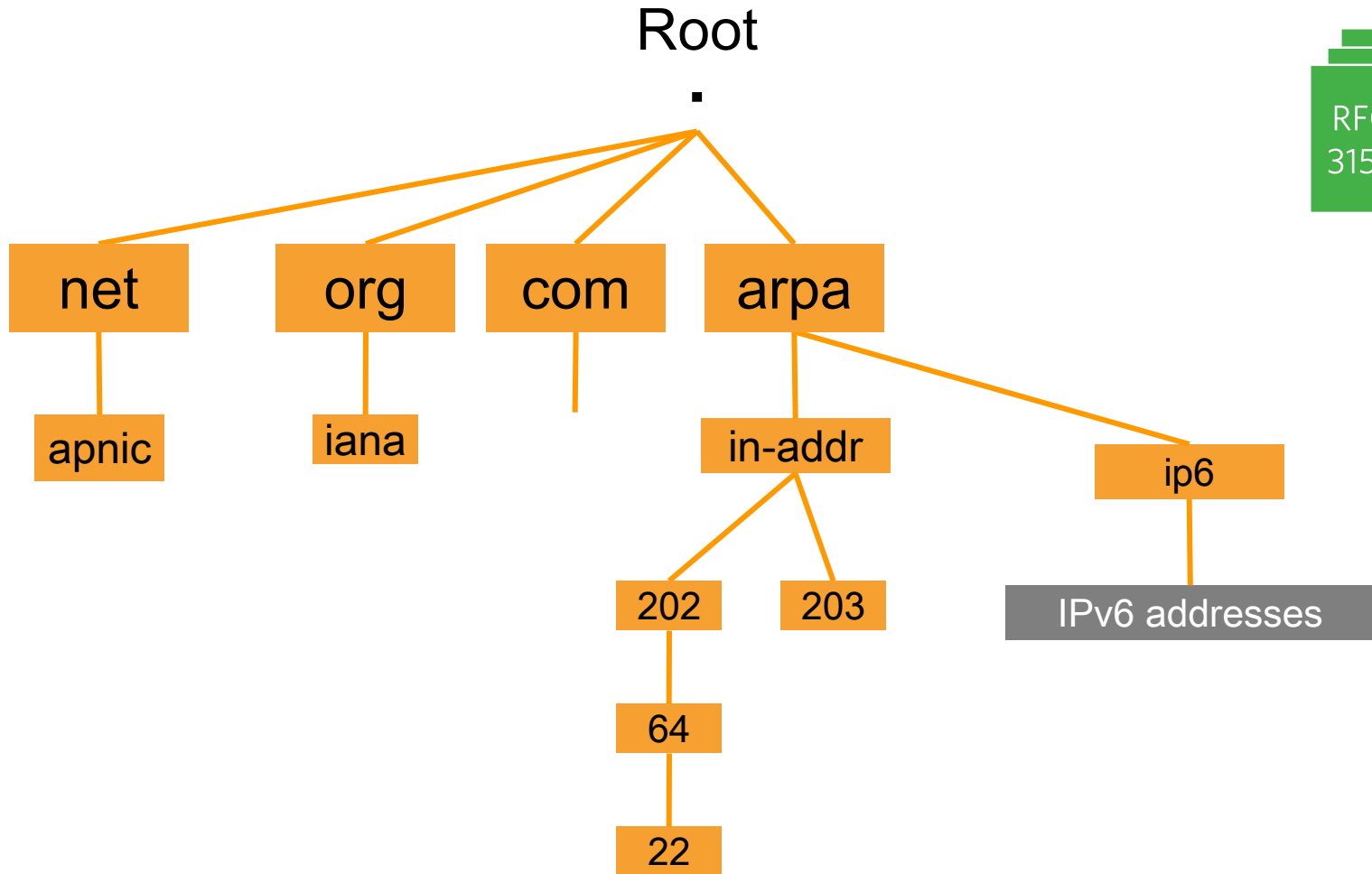
```
NS ns1.company.org.
```

```
NS ns2.company.org.
```

```
1 PTR gw.company.org.  
    router.company.org.
```

```
2 PTR ns1.company.org.
```

# Reverse DNS Tree – with IPv6



# IPv6 Reverse Lookups – PTR records

- Similar to the IPv4 reverse record

```
b.a.9.8.7.6.5.0.4.0.0.0.3.0.0.0.2.0.0.0.1.0.0.0.0.0.0.0.1.2.3.4.ip6.arpa.
```

```
IN PTR test.ip6.example.com.
```

- Example: The reverse name lookup for a host with address

```
3ffe:8050:201:1860:42::1
```

```
$ORIGIN 0.6.8.1.1.0.2.0.0.5.0.8.e.f.f.3.ip6.arpa.
```

```
1.0.0.0.0.0.0.0.0.0.0.0.0.2.4.0.0 14400 IN PTR  
host.example.com.
```



# Example: Reverse Zone

```
;; 0.0.0.0.0.0.1.0.8.6.4.0.1.0.0.2.rev  
;; These are reverses for 2001:468:100::/64)
```

```
$TTL          86400
```

```
@          IN          SOA      ns1.domain.edu. root.domain.edu. (  
          2015050501; serial - YYYYMMDDXX  
          21600        ; refresh - 6 hours  
          1200         ; retry - 20 minutes  
          3600000      ; expire - long time  
          86400)      ; minimum TTL - 24 hours
```

```
;; Nameservers
```

```
      IN NS ns1.domain.edu.
```

```
      IN NS ns2.domain.edu.
```

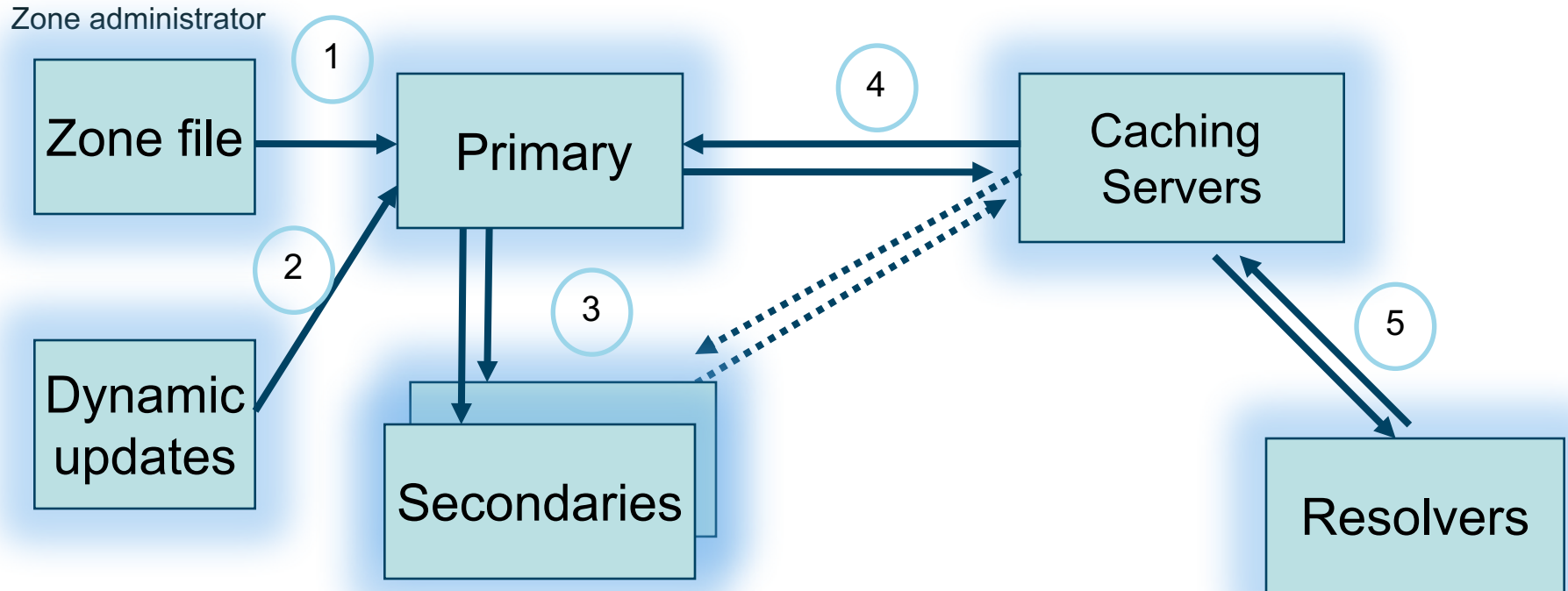
```
1.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0 IN PTR  
  host1.ip6.domain.edu.
```

```
2.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0 IN PTR host2.domain.edu
```

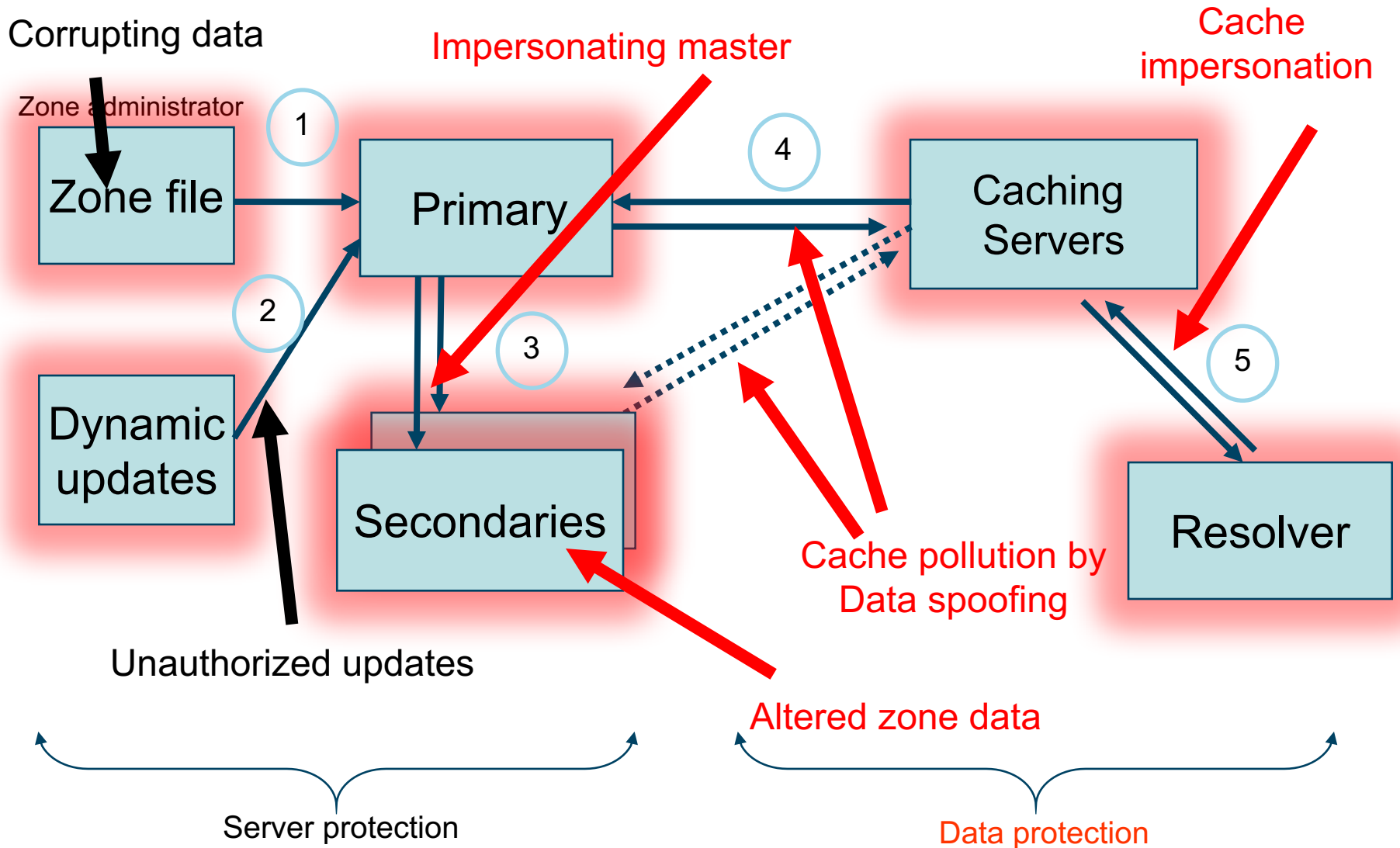


# DNS Security

# DNS: Data Flow



# DNS Vulnerabilities



# The Bad

- DNSChanger\*
  - Biggest Cybercriminal Takedown in History
  - 4M machines, 100 countries, \$14M
- And many other DNS hijacks in recent times\*\*
- SSL / TLS doesn't tell you if you've been sent to the correct site, it only tells you if the DNS matches the name in the certificate. Unfortunately, majority of Web site certificates rely on DNS to validate identity.
- DNS is relied on for unexpected things though insecure.

\* [http://www.fbi.gov/news/stories/2011/november/malware\\_110911/malware\\_110911](http://www.fbi.gov/news/stories/2011/november/malware_110911/malware_110911)  
End-2-end DNSSEC validation would have avoided the problems

\*\* A Brief History of DNS Hijacking - Google  
<http://costarica43.icann.org/meetings/sanjose2012/presentation-dns-hijackings-marquis-boire-12mar12-en.pdf>





# Basic Cache Poisoning

## Attacker

- Launches a spam campaign where spam message contains <http://loseweightfastnow.com>
- Attacker's name server will respond to a DNS query for loseweightnow.com with malicious data about ebay.com
- Vulnerable resolvers add malicious data to local caches
- The malicious data will send victims to an eBay phishing site for the lifetime of the cached entry



My Mac



My local resolver

What is the IPv4 address for loseweightfastnow.com

I'll cache this response... and update www.ebay.com

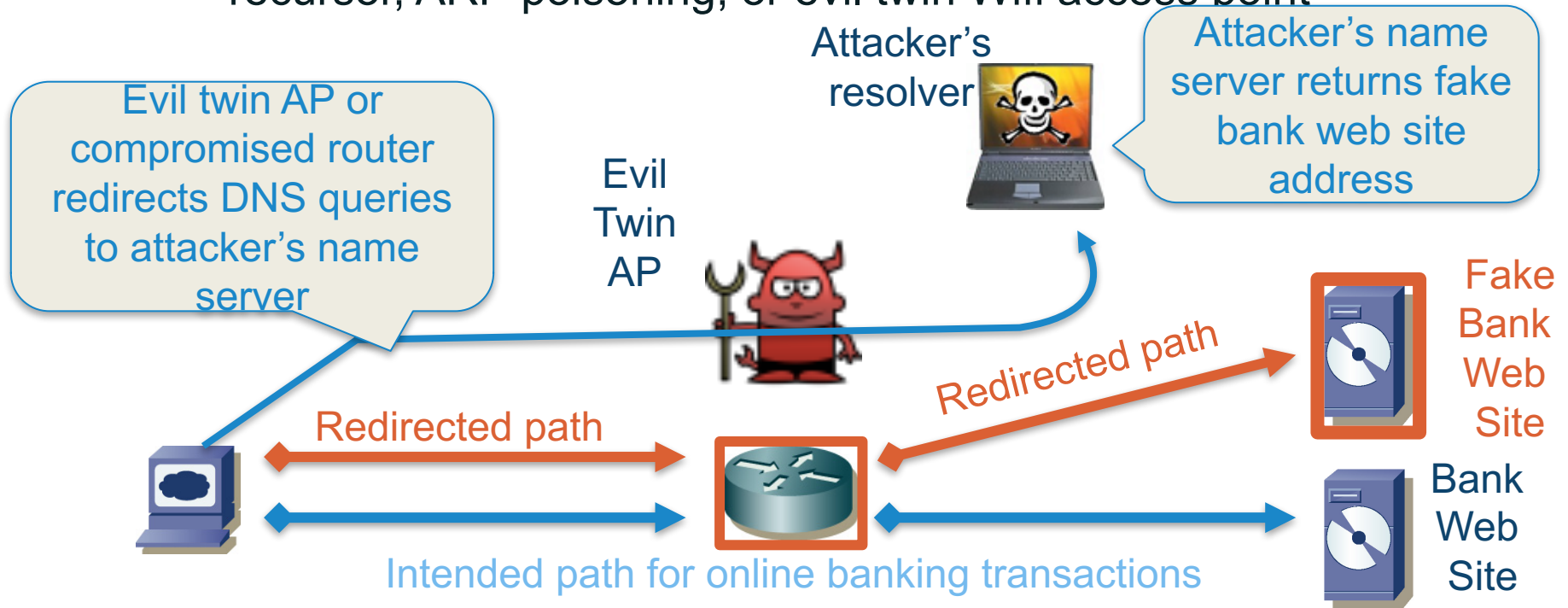
loseweightfastnow.com IPv4 address is 192.168.1.1  
ALSO www.ebay.com is at 192.168.1.2



ecrime name server

# Query Interception (DNS Hijacking)

- A man in the middle (MITM) or spoofing attack forwards DNS queries to a name server that returns forged responses
  - Can be done using a DNS proxy, **compromised** access router or recursor, ARP poisoning, or evil twin Wifi access point





# Protecting DNS Servers and Transactions



A world map where the continents are defined by a complex network of white nodes and connecting lines, set against a dark blue background. The nodes vary in size, and the lines are thin and light blue. The overall effect is a digital or network-based representation of the world's geography.

# ACLs and Views

# Elements in an address match list

- Individual IP addresses
- Addresses/netmask pairs
- Names of other ACLs
- In some contexts, key names



# Purposes in Bind

- Restricting queries & zone xfer
  - Authorizing dynamic updates
  - Selecting interfaces to listen on
- \* Address match lists are always enclosed in curly braces.

# Notes on Address Match list

- Elements must be separated by “ ; ”
- The list must be terminated with a “ ; ”
- Elements of the address match list are checked sequentially.
- To negate elements of the address match list prepend them with “!”
- Use *acl* statement to name an address match list.
- *acl* must be define before it can be used elsewhere.

# Example: Address match lists

- For network 192.168.0.0 255.255.255.0  
  { 192.168.0.0/24; }
- For network plus loopback  
  { 192.168.0.0/24; 127.0.0.1; }
- Addresses plus key name  
  { 192.168.0.0/24; 127.0.0.1; example.myzone.net; }

# The *acl* Statement

- Syntax:

```
acl <acl name> { address match list};
```

- Example:

```
acl internal { 127.0.0.1; 192.168.0/24; };
```

```
acl dynamic-update { key  
dhcp.myzone.net; };
```

# Notes on the *acl* Statement

- The *acl* name need not be quoted.
- There are four predefined ACLs:
  - any* (Any IP address)
  - none* (No IP address)
  - localhost* (loopback, 127.0.0.1)
  - localnets* (all networks the name server is directly connected to)



# Blackhole

```
options {  
    blackhole { ACL-name or itemized list; };  
};
```

# Allow-transfer

```
zone "myzone.example." {  
type master;  
file "myzone.example."  
allow-transfer { ACL-name or  
itemized list; };  
};
```

# Allow-Query

```
zone "myzone.example." {  
type master;  
file "myzone.example."  
allow-query { ACL-name or  
itemized list; };  
};
```

```
options {  
  listen-on port # { ACL-  
name or itemized list;};  
};
```

The view statement is a powerful feature that lets a name server answer a DNS query differently depending on who is asking. It is particularly useful for implementing split DNS setups without having to run multiple servers.



- view view\_name  
[class] {  
match-clients { address\_match\_list } ;  
match-destinations {  
address\_match\_list } ;  
match-recursive-only yes\_or\_no ;  
[ view\_option; ...]  
[ zone\_statement; ...]  
};

# Example Config

- view "internal" {  
    // This should match our internal networks.  
    match-clients { 10.0.0.0/8; };  
  
    // Provide recursive service to internal clients only.  
    recursion yes;  
  
    // Provide a complete view of the example.com zone  
    // including addresses of internal hosts.  
    zone "example.com" {  
        type master;  
        file "example-internal.db";  
    };  
};

```
view "external" {  
    // Match all clients not matched by the previous view.  
    match-clients { any; };  
  
    // Refuse recursive service to external clients.  
    recursion no;  
  
    // Provide a restricted view of the example.com zone  
    // containing only publicly accessible hosts.  
    zone "example.com" {  
        type master;  
        file "example-external.db";  
    };  
};
```



TSIG

# What is TSIG - Transaction Signature?

- A mechanism for protecting a message from a primary to secondary and vice versa
- A keyed-hash is applied (like a digital signature) so recipient can verify the message
  - DNS question or answer
  - & the timestamp
- Based on a shared secret - both sender and receiver are configured with it
  - TSIG/TKEY uses DH, HMAC-MD5, HMAC-SHA1, HMAC-SHA224, HMAC-SHA512 among others



# What is TSIG - Transaction Signature?

- TSIG (RFC 2845)
  - authorizing dynamic updates & zone transfers
  - authentication of caching forwarders
- Used in server configuration, not in zone file

# TSIG Steps

1. Generate secret
2. Communicate secret
3. Configure servers
4. Test

- TSIG name
  - A name is given to the key, the name is what is transmitted in the message (so receiver knows what key the sender used)
- TSIG secret value
  - A value determined during key generation
  - Usually seen in Base64 encoding

# TSIG – Generating a Secret

- `dnssec-keygen`

- Simple tool to generate keys
- Used here to generate TSIG keys

```
> dnssec-keygen -a <algorithm> -b  
  <bits> -n host <name of the key>
```

# TSIG – Generating a Secret

- **Example**

```
> dnssec-keygen -a HMAC-SHA1 -b 128 -n HOST ns1-  
ns2.myzone.net
```

This will generate the key

```
> Kns1-ns2.myzone.net.+157+15921
```

```
>ls
```

```
Kns1-ns2.myzone.net.+157+15921.key
```

```
Kns1-ns2.myzone.net.+157+15921.private
```

# TSIG – Generating a Secret

- TSIG should never be put in zone files
  - might be confusing because it looks like RR:

```
ns1-ns2.myzone.net. IN KEY 128 3 157 nEfRX9...bbPn7lyQtE=
```



# TSIG – Configuring Servers

- Configuring the key
  - in named.conf file, same syntax as for rndc
  - `key { algorithm ...; secret ...; }`
- Making use of the key
  - in named.conf file
  - `server x { key ...; }`
  - where 'x' is an IP number of the other server

# Configuration Example – named.conf

## Primary server 10.33.40.46

```
key ns1-ns2.myzone.net {
    algorithm hmac-sha1;
    secret "APlaceToBe";
};
server 10.33.50.35 {
    keys {ns1-ns2.myzone.net;};
};
zone "my.zone.test." {
    type master;
    file "db.myzone";
    allow-transfer {
        key ns1-ns2.myzone.net ;};
};
```

## Secondary server 10.33.50.35

```
key ns1-ns2.myzone.net {
    algorithm hmac-sha1;
    secret "APlaceToBe";
};
server 10.33.40.46 {
    keys {ns1-ns2.myzone.net;};
};
zone "my.zone.test." {
    type slave;
    file "myzone.backup";
    masters {10.33.40.46;};
```

You can save this in a file and refer to it in the named.conf using 'include' statement:

```
include "/var/named/master/tsig-key-ns1-ns2";
```

- You can use dig to check TSIG configuration

```
– dig @<server> <zone> AXFR -k <TSIG keyfile>
```

```
$ dig @127.0.0.1 example.net AXFR \  
–k Kns1-ns2.myzone.net.+157+15921.key
```

- Wrong key will give “Transfer failed” and on the server the security-category will log this.

# TSIG Testing - TIME!

- TSIG is time sensitive - to stop replays
  - Message protection expires in 5 minutes
  - Make sure time is synchronized
  - For testing, set the time
  - In operations, (secure) NTP is needed



# Protecting DNS Data

# Brief reminder on Cryptography

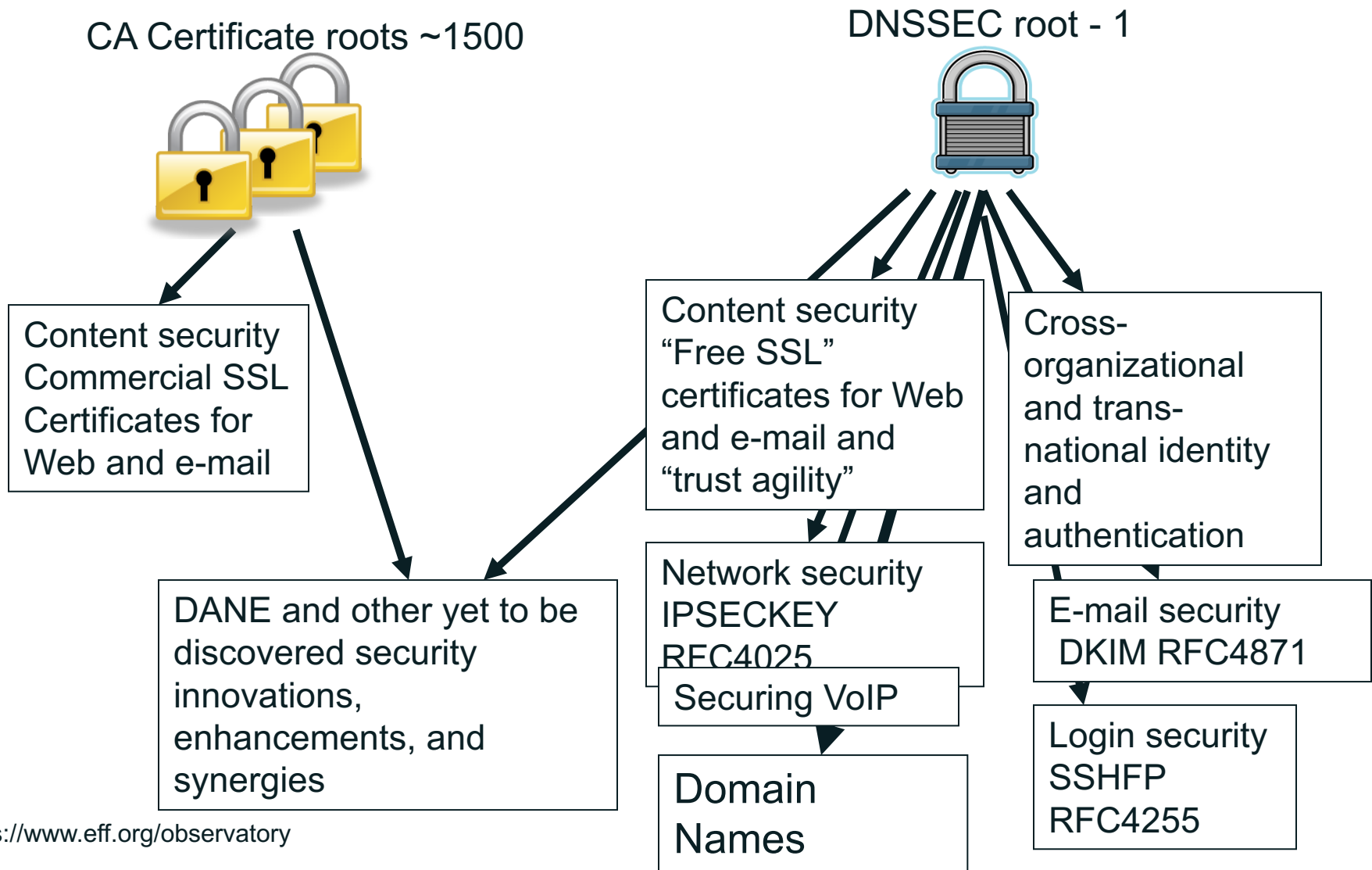
- Nowadays most of our Security Services are based in one (or a combination) of the following areas:
  - One-way hash functions
  - Symmetric key crypto
  - Public-key crypto (or asymmetric)



# Where DNSSEC fits in

- CPU and bandwidth advances make legacy DNS vulnerable to MITM attacks
- DNS Security Extensions (DNSSEC) introduces digital signatures into DNS to cryptographically protect contents
- With DNSSEC fully deployed a business can be sure a customer gets un-modified data (and visa versa)

# Too many CAs. Which one can we trust?



<https://www.eff.org/observatory>

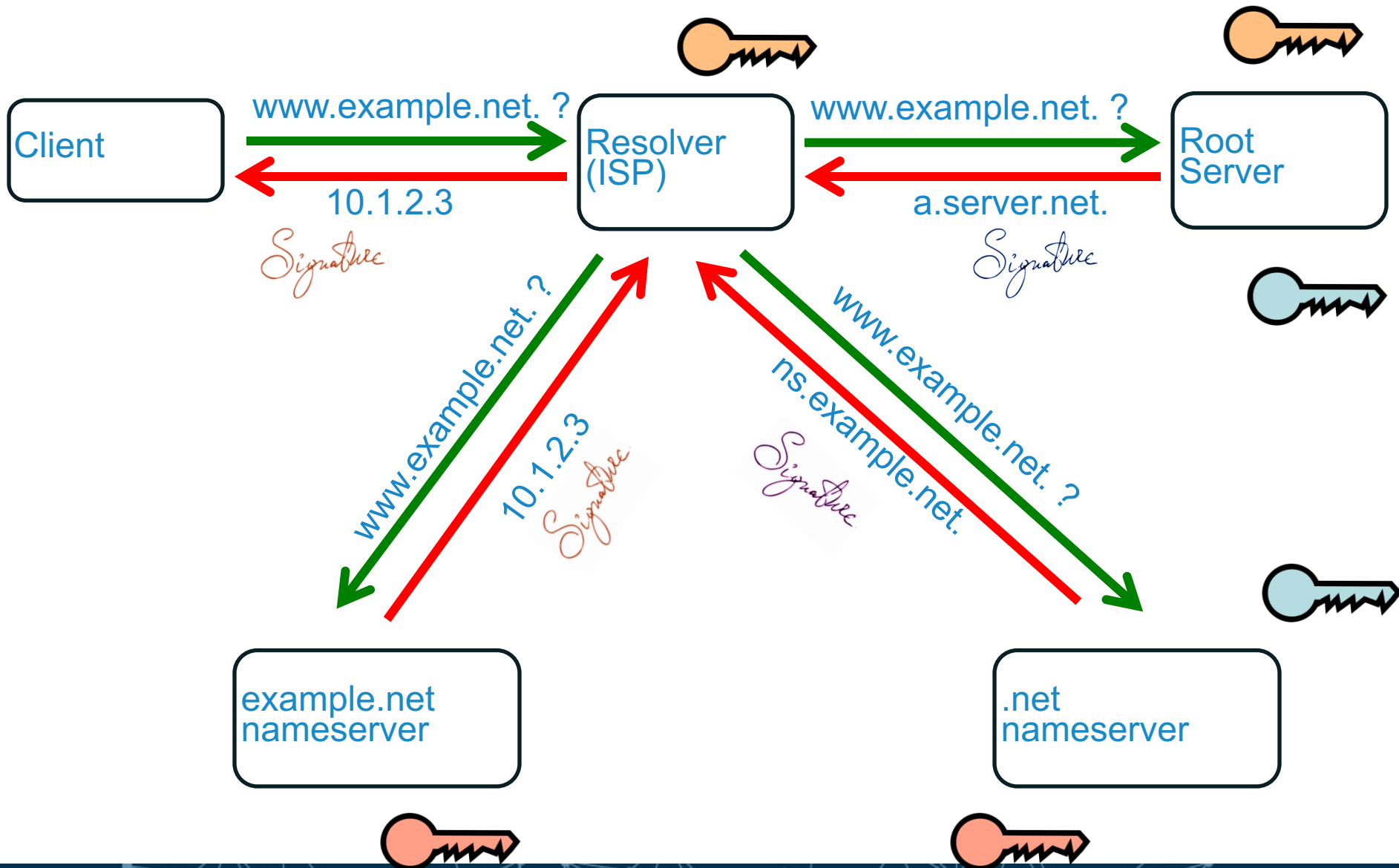
# Opportunity: New Security Solutions

- Improved Web SSL and certificates for all
- Secured e-mail (S/MIME) for all
- Validated remote login SSH, IPSEC
- Securing VoIP
- Cross organizational digital identity systems
- Secured content delivery (e.g. configurations, updates, keys)
- Securing Smart Grid efforts
- First global FREE PKI
- Increasing trust in e-commerce



# How DNSSEC Works?

# How DNSSEC Works





# How DNSSEC Works

- Data authenticity and integrity by signing the Resource Records Sets with a private key
- Public DNSKEYs published, used to verify the RRSIGs
- Children sign their zones with their private key
  - Authenticity of that key established by parent signing hash (DS) of the child zone's key
- Repeat for parent...
- Not that difficult on paper
  - Operationally, it is a bit more complicated
  - $DS_{KEY} \rightarrow KEY \text{ --signs--} \rightarrow \text{zone data}$



# The Business Case for DNSSEC

- Cyber security is becoming a greater concern to enterprises, government, and end users. DNSSEC is a key tool and differentiator.
- DNSSEC is the biggest security upgrade to Internet infrastructure in over 20 years. It is a platform for new security applications (for those that see the opportunity).
- DNSSEC infrastructure deployment has been brisk but requires expertise. Getting ahead of the curve is a competitive advantage.

# DNSSEC ccTLD Map

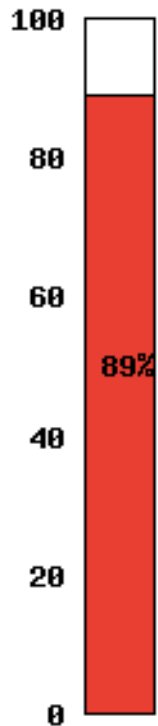


<https://rick.eng.br/dnssecstat/>

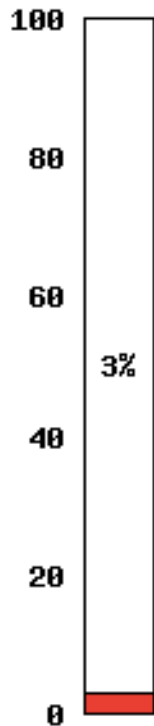
# DNSSEC Deployment

## Trend

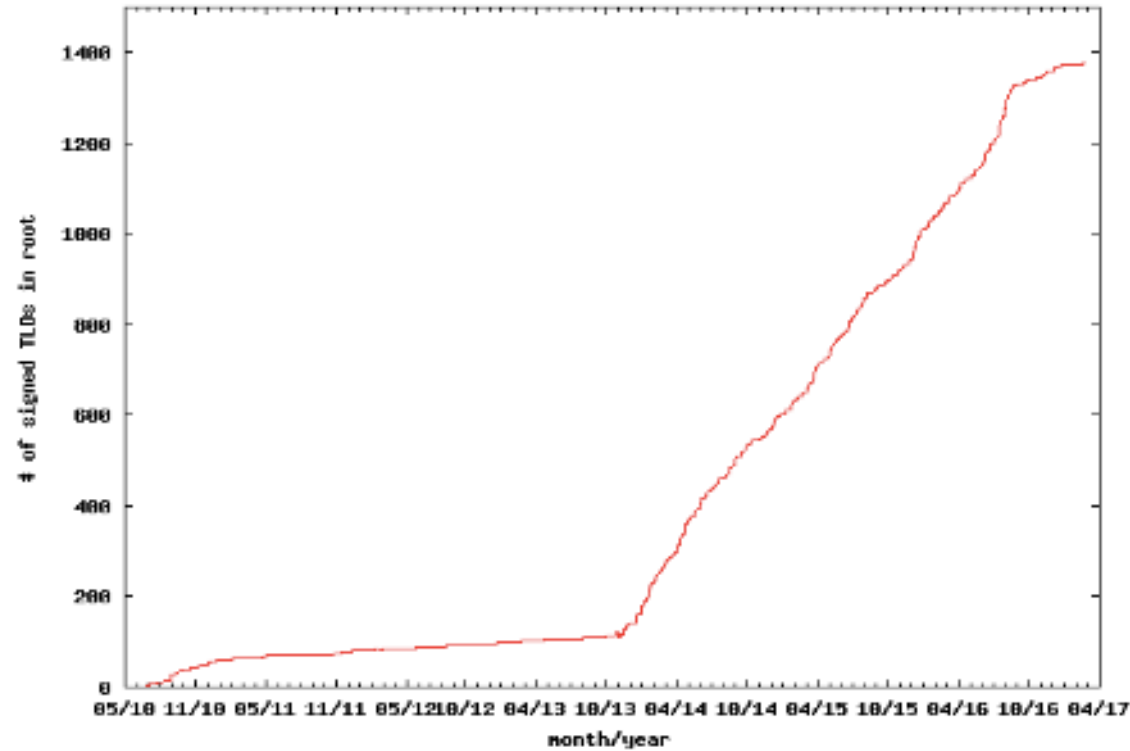
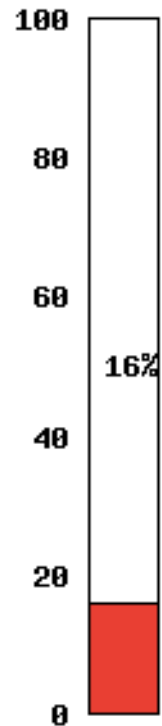
% of TLDs signed in root



Approx % of these 2LDs signed



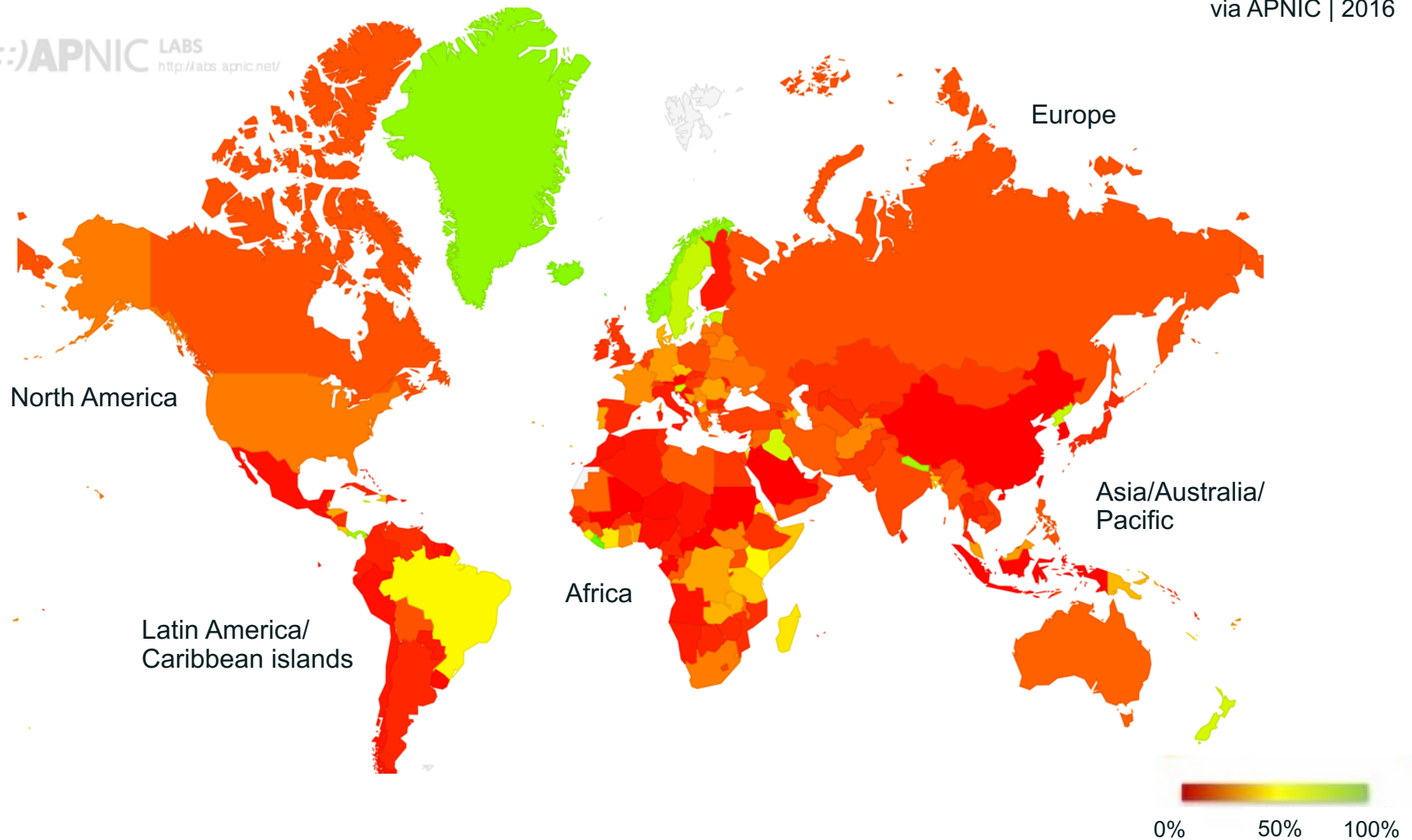
Approx\* % Users Validating



<https://rick.eng.br/dnssecstat/>

# Global State of DNSSEC Validation

via APNIC | 2016



# DNSSEC: So what's the problem?

- Not enough IT departments know about it or are too busy putting out other security fires.
- When they do look into it they hear old stories of FUD and lack of turnkey solutions.
- Registrars\*/DNS providers see no demand leading to “chicken-and-egg” problems.

\*but required by new ICANN registrar agreement

# What you can do

- For Companies:
  - Sign your corporate domain names
  - Just turn on validation on corporate DNS resolvers
- For Users:
  - Ask ISP to turn on validation on their DNS resolvers
- For All:
  - Take advantage of DNSSEC education and training

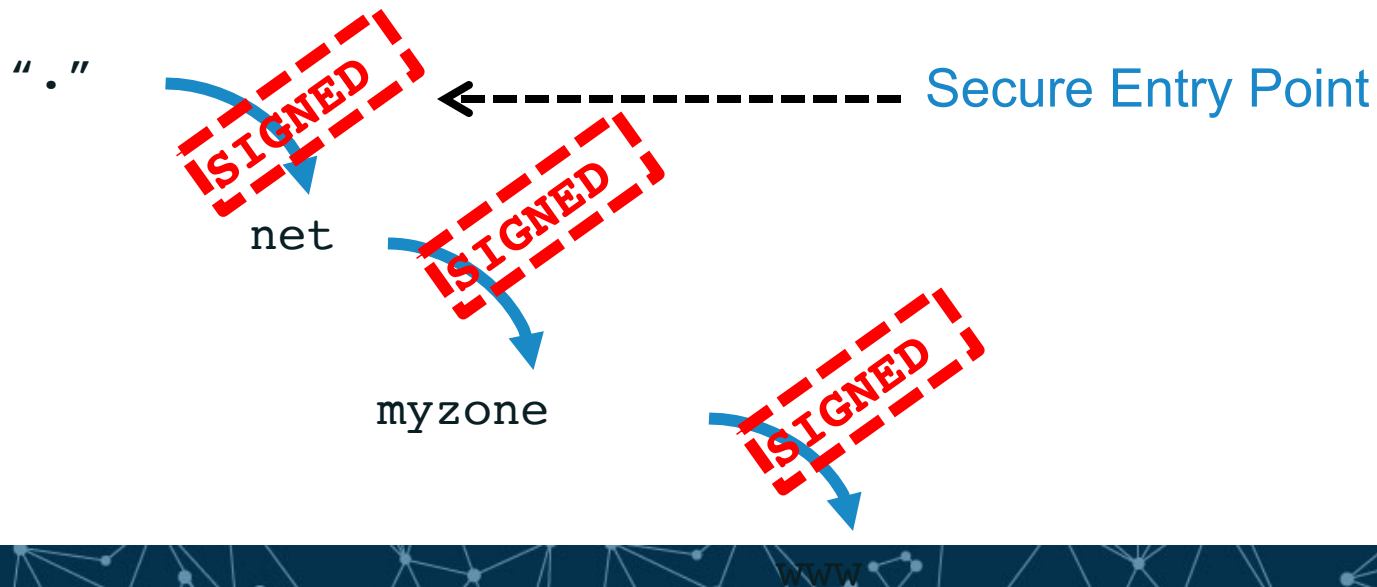


# New Concepts

- Secure Entry Point and Chain of Trust
  - Delegating Signing Authority
- New packet options (flags)
  - CD, AD, DO
- New RRs
  - DNSKEY, RRSIG, NSEC/NSEC3 and DS
- Signature expiration
- Key Rollovers

# Chain of Trust and Secure Entry Point

- Using the existing delegation based model of distribution
- Don't sign the entire zone, sign a RRset
- Parent DOES NOT sign the child zone. The parent signs a pointer (hash) to the key used to sign the data of the child zone (DS record)
- Example with **www.myzone.net**.



A world map where the continents are defined by a complex network of white nodes and connecting lines, set against a dark blue background. The nodes vary in size and are densely packed in some areas, creating a digital or network-like appearance of the globe.

# New Resource Records

# New RRs

- Adds five new DNS Resource Records:
  1. DNSKEY: Public key used in zone signing operations.
  2. RRSIG: RRset signature
  3. NSEC &
  4. NSEC3: Returned as verifiable evidence that the name and/or RR type does not exist
  5. DS: Delegation Signer. Contains the hash of the public key used to sign the key which itself will be used to sign the zone data. Follow DS RR's until a "trusted" zone is reached (ideally the root).

# New RR: DNSKEY

```
OWNER          TYPE          FLAGS          PROTOCOL
example.net.   43200        DNSKEY         256           3           7 (
AwEAAbinasY+k/9xD4MBBa3QvhjuOHipe319SFbWYIRj    PUBLIC KEY
/nbmVZfJnSw7By1cV3Tm7ZlLqNbcB86nVFMSQ3JjOFMr    (BASE64)
....) ; ZSK; key id = 23807          KEY ID
```

- FLAGS determines the usage of the key
- PROTOCOL is always 3 (DNSSEC)
- ALGORITHM can be (3: DSA/SHA-1, 5: RSA/SHA1, 8: RSA/SHA-256, 12: ECC-GOST)
  - <http://www.iana.org/assignments/dns-sec-alg-numbers/dns-sec-alg-numbers.xml>

# DNSKEY: Two Keys, not one...

- There are in practice at least **two** DNSKEY pairs for every zone
- Originally, one key-pair (public, private) defined for the zone
  - private: key used to sign the zone data (RRsets)
  - public: key published (DNSKEY) in the zone
- DNSSEC works fine with a single key pair
- Problem with using a single key:
  - Every time the key is updated, the DS record must be updated on the parent zone as well
  - Introduction of **Key Signing Key** (flags=257)



# KSK and ZSK

- Key Signing Key (KSK)
  - Pointed to by parent zone in the form of DS (Delegation Signer). Also called Secure Entry Point.
  - Used to sign the Zone Signing Key
  - Flags: 257
- Zone Signing Key (ZSK)
  - Signed by the KSK
  - Used to sign the zone data RRsets
  - Flags: 256
- This decoupling allows for independent updating of the ZSK without having to update the KSK, and involve the parents (i.e. less administrative interaction)

# New RR: RRSIG (Resource Record Signature)

```
example.net. 600 A 192.168.10.10  
example.net. 600 A 192.168.23.45
```

TYPE COVERED #LABELS

OWNER

TYPE

ALG

TTL

```
example.net. 600 RRSIG A 7 2 600 (
```

SIG. EXPIRATION

SIG. INCEPTION

KEY ID SIGNER NAME

```
20150115154303
```

```
20141017154303
```

```
23807
```

```
example.net.
```

SIGNATURE

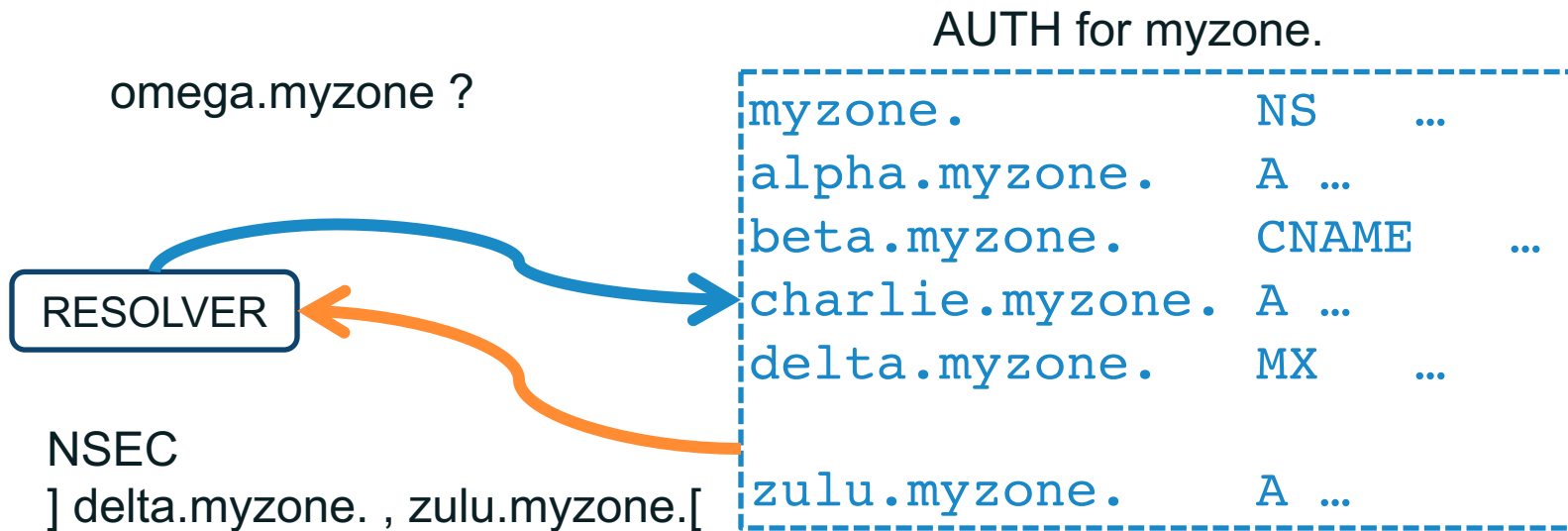
```
CoYkYPqE8Jv6UaVJgRrh7u16m/cEFGtFM8TArbJdaiPu  
W77wZhrvonoBEyqYbhQ1yDaS74u9whECEe08gfoe1FGg
```

```
. . .  
)
```

- Typical default values
  - Signature inception time is 1 hour before.
  - Signature expiration is 30 from now
  - Proper timekeeping (NTP) is required
- What happens when signatures run out?
  - SERVFAIL
  - Domain effectively disappears from the Internet for validating resolvers
- Note that keys do not expire
- No all RRsets need to be resigned at the same time

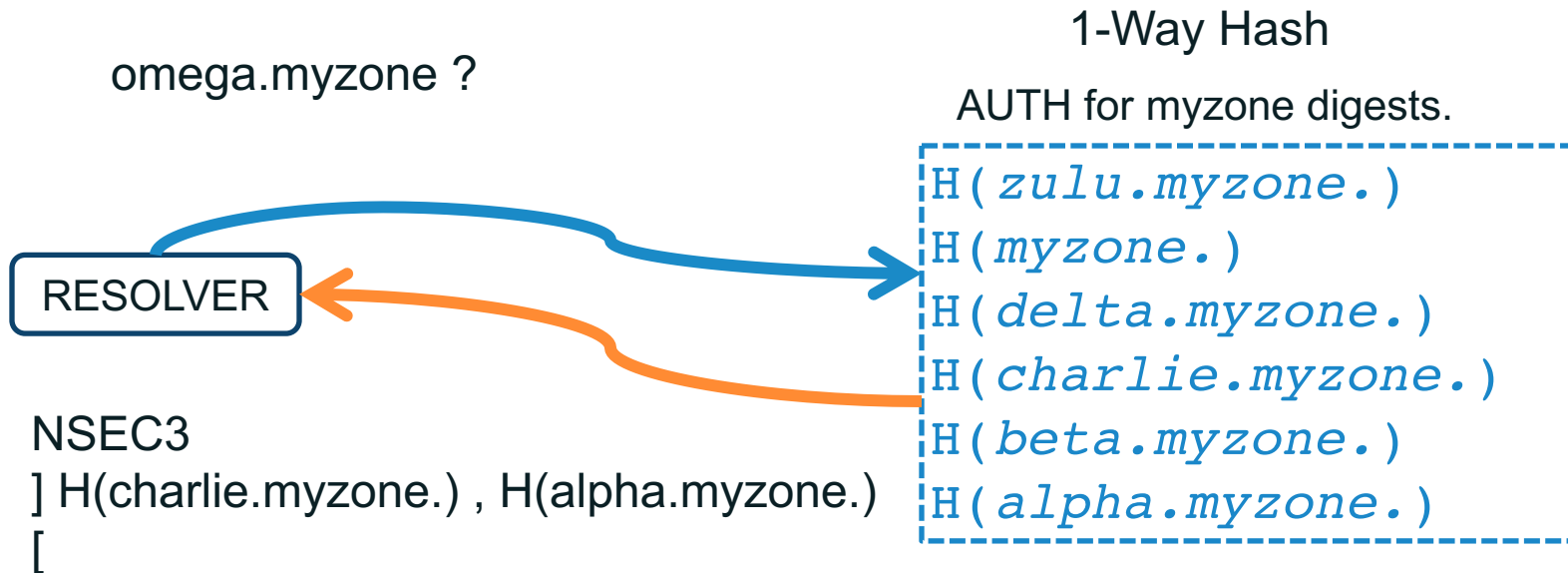
# New RR: NSEC

- NXDomains also must be verified
- NSEC provides a pointer to the Next SECure record in the chain of records.



# New RR: NSEC3

- To avoid concerns about “zone enumeration”
- To avoid large zone-files: opt-out concept



# New RR: DS (Delegation Signer)

- Hash of the KSK of the child zone
- Stored in the parent zone, together with the NS RRs indicating a delegation of the child zone.
- The DS record for the child zone is signed together with the rest of the parent zone data
- NS records are NOT signed (they are a hint/pointer)

Digest type 1 = SHA-1, 2 = SHA-256

```
myzone. DS 61138 5 1  
F6CD025B3F5D0304089505354A0115584B56D683
```

```
myzone. DS 61138 5 2  
CCBC0B557510E4256E88C01B0B1336AC4ED6FE08C8268CC1AA5FBF00 5DCE3210
```



A world map where the continents are defined by a complex network of white nodes and connecting lines, set against a dark blue background. The nodes vary in size and are densely packed in some areas, creating a mesh-like structure that outlines the major landmasses.

# Signatures expiration and Key Rollovers

# Signature Expiration

- Signatures are per default 30 days (BIND)
- Need for regular resigning:
  - To maintain a constant window of validity for the signatures of the existing RRset
  - To sign new and updated Rrsets
  - Use of jitter to avoid having to resign all expiring RRsets at the same time
- The keys themselves do NOT expire...
- But they may need to be rolled over...

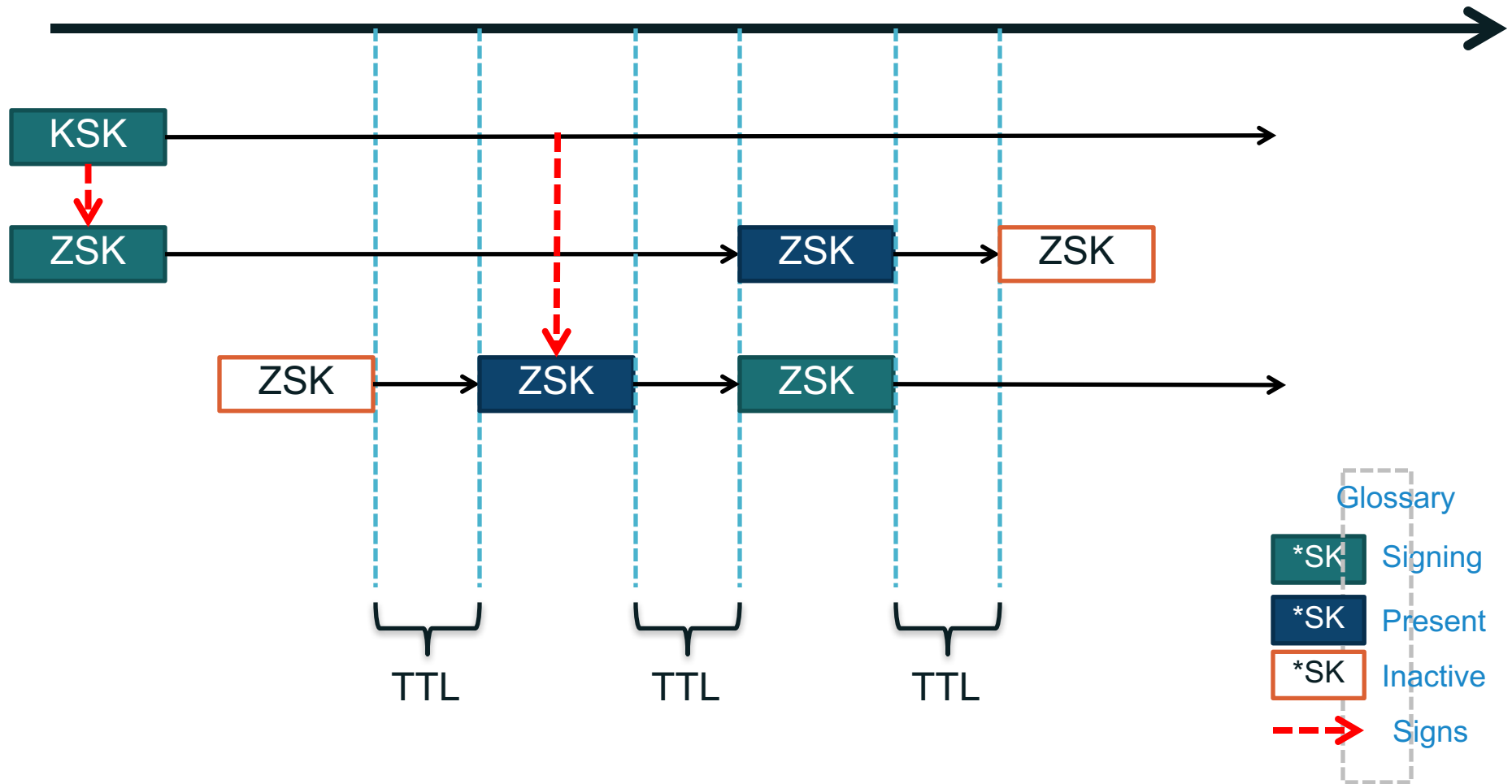
# Key Rollovers

- Try to minimise impact
  - Short validity of signatures
  - Regular key rollover
- Remember: DNSKEYs do not have timestamps
  - the RRSIG over the DNSKEY has the timestamp
- Key rollover involves second party or parties:
  - State to be maintained during rollover
  - Operationally expensive

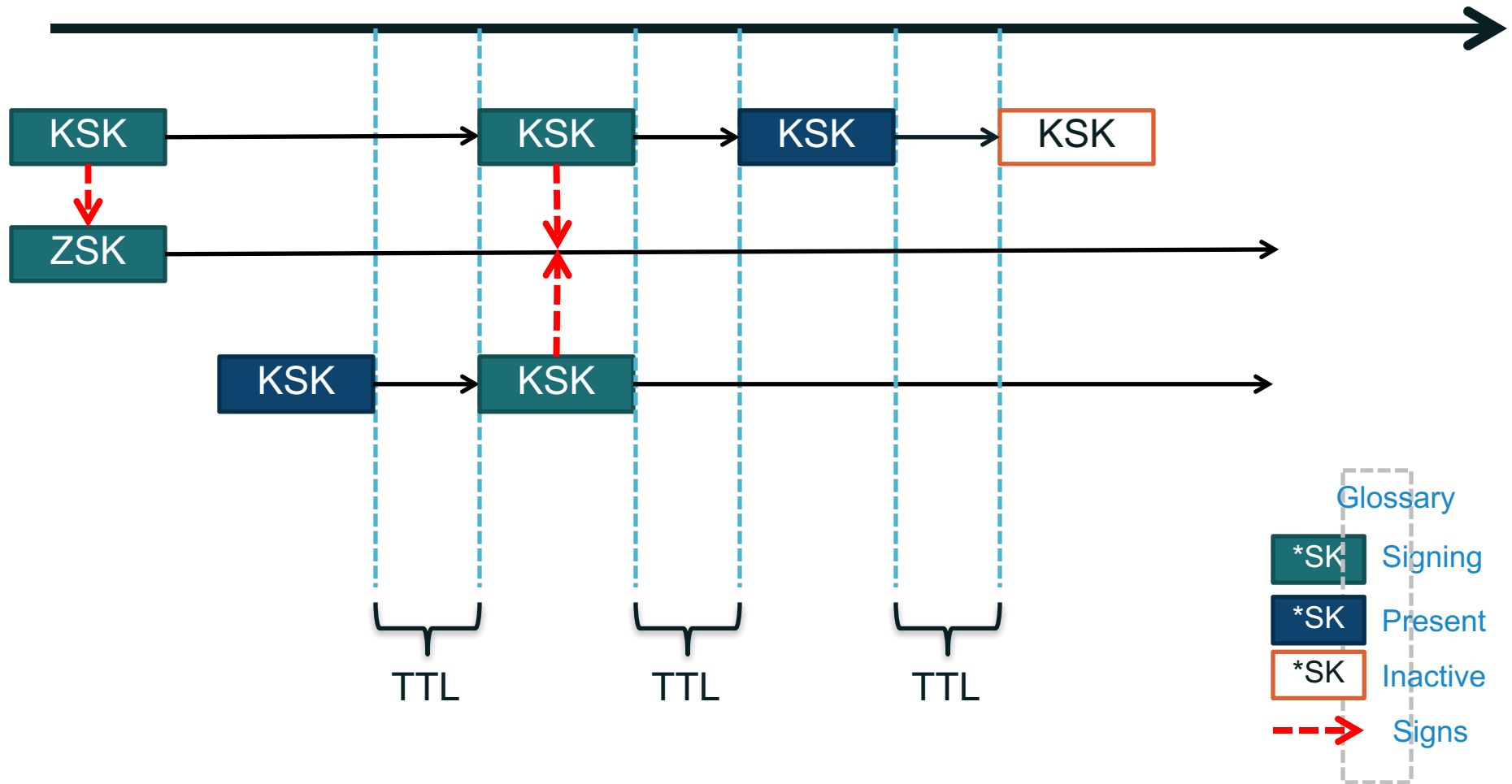
# Key Rollovers

- Two methods for doing key rollover
  - Pre-Publish
  - Double Signature
- KSK and ZSK rollover use different methods.
  - Remember that KSK needs to interact with parent zone to update DS record.

# Key Rollovers: Pre-Publish method



# Key Rollovers: Double Signature





# Steps

- Enable DNSSEC in the configuration file (named.conf)

```
dnssec-enable yes;  
dnssec-validation yes;
```


- Create key pairs (KSK and ZSK)

```
dnssec-keygen -a rsasha1 -b 1024 -n zone myzone.net  
dnssec-keygen -a rsasha1 -b 1400 -f KSK -n zone myzone.net
```

- Publish your public key

```
$INCLUDE /path/Kmyzone.net.+005+33633.key ; ZSK  
$INCLUDE /path/Kmyzone.net.+005+00478.key ; KSK
```

- Signing the zone
- Update the config file
  - Modify the zone statement, replace with the signed zone file
- Test with dig



**DNSSEC: Internet infrastructure  
upgrade to help address today's needs  
and create tomorrow's opportunity.**



A world map where the continents are defined by a complex network of white nodes and connecting lines, set against a dark blue background. The nodes vary in size and are densely packed in some areas, creating a digital or network-like appearance of the globe.

Hmm...how do I trust it?

# ICANN DNSSEC Deployment @Root

- Multi-stakeholder, bottom-up trust model\* /w 21 crypto officers from around the world
- Broadcast Key Ceremonies and public docs
- SysTrust audited
- FIPS 140-2 level 4 HSMs

Root DNSSEC Design Team

F. Ljunggren  
Kirei  
T. Okubo  
VeriSign  
R. Lamb  
ICANN  
J. Schlyter  
Kirei  
May 21, 2010

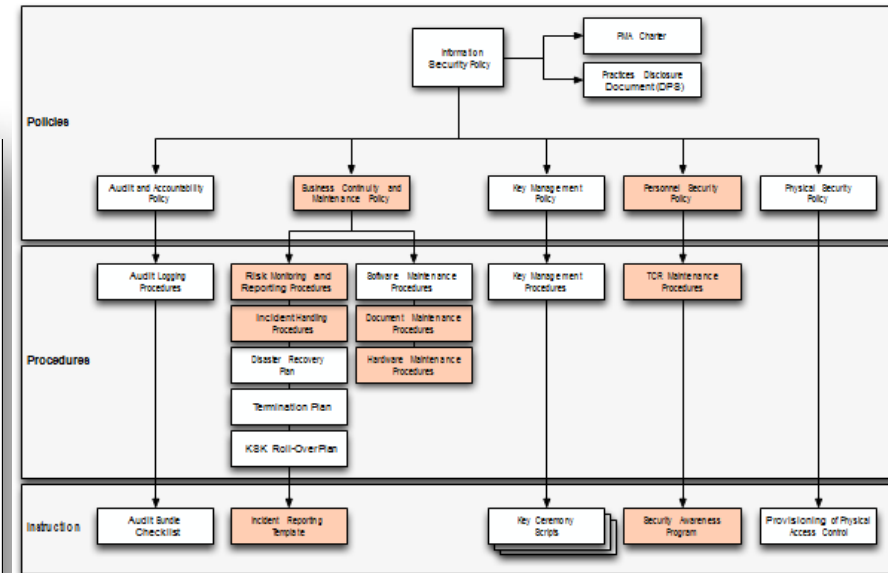
DNSSEC Practice Statement for the Root Zone KSK Operator

## Abstract

This document is the DNSSEC Practice Statement (DPS) for the Root Zone Key Signing Key (KSK) Operator. It states the practices and provisions that are used to provide Root Zone Key Signing and Key Distribution services. These include, but are not limited to: issuing, managing, changing and distributing DNS keys in accordance with the specific requirements of the U.S. Department of Commerce.

## Copyright Notice

Copyright 2009 by VeriSign, Inc., and by Internet Corporation For Assigned Names and Numbers. This work is based on the Certification



## Root DPS

### DNSSEC Practice Statement

\*Managed by technical community+ICANN

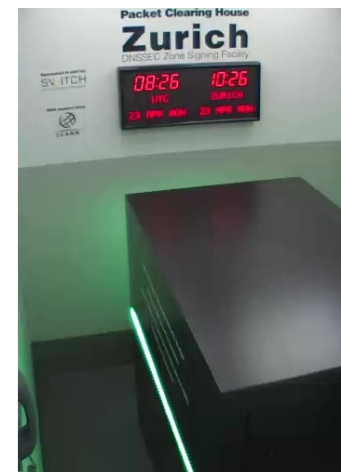
# ICANN DNSSEC Deployment @Root (and elsewhere)



FIPS 140-2 level 4



DCID 6/9





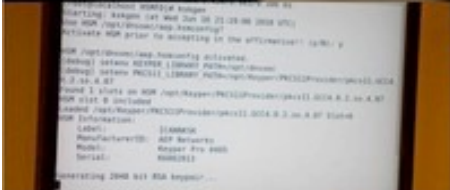


January 27, 2010









Photos: Kim Davies



A world map where the continents are defined by a complex network of white dots and thin white lines. The dots vary in size, and the lines connect them to form a web-like structure. The background is a solid dark blue color.

# Tools to help the process

# Tools to use in DNSSEC

- Authoritative Servers that support DNSSEC
  - NSD (by NLNetLabs)
  - Knot (by CZ NIC Labs)
  - BIND (by ISC)
  - Vantio (by Nominum)
  - YADIFA (by EURid)
  - MS DNS Server (by Microsoft)
  - TinyDNSSEC (based on tinydns by D.J. Bernstein)

# Tools to use in DNSSEC

- Resolvers that support DNSSEC
  - Unbound (by NLNetLabs)
  - BIND (by ISC)
  - MS Windows Server (by Microsoft)
- Tools to automate DNSSEC
  - OpenDNSSEC (by NLnetLabs, .SE, Nominet...et al)
  - DNSSEC-Tools (by Sparta)
  - BIND (by ISC)





**DNSSEC: Internet infrastructure  
upgrade to help address today's needs  
and create tomorrow's opportunity.**





# DNSSEC Demo

# Thank you and Questions



Email: <[champika.wijayatunga@icann.org](mailto:champika.wijayatunga@icann.org)>

Website: [icann.org](http://icann.org)



[twitter.com/icann](https://twitter.com/icann)  
[twitter.com/icann4biz](https://twitter.com/icann4biz)



[gplus.to/icann](https://plus.google.com/icann)



[facebook.com/icannorg](https://facebook.com/icannorg)



[weibo.com/ICANNorg](https://weibo.com/ICANNorg)



[linkedin.com/company/icann](https://linkedin.com/company/icann)



[flickr.com/photos/icann](https://flickr.com/photos/icann)



[youtube.com/user/icannnews](https://youtube.com/user/icannnews)



[slideshare.net/icannpresentations](https://slideshare.net/icannpresentations)