

# An Overview of

# DNSSEC

Olaf M. Kolkman  
[olaf@nlnetlabs.nl](mailto:olaf@nlnetlabs.nl)

# Who am I

- Director of NLnet Labs, a charity working on open standards and open source software
  - NSD, Unbound, Idns, Net::DNS, Net::DNS::SEC
  - DNSSEC evangelising
- Previously @ RIPE NCC: responsible for DNSSEC deployment
  - DNSEXT chair 2001-2006
  - IAB member since, 2006 chair since 2007

# DNS

- Domain Name System
- Provides the mapping from names to resources
- A global, distributed, loosely coherent system
- Almost all transactions on the Internet use the DNS

# DNS has a distributed nature

- Authoritative servers all provide part of the name space
- User devices query a local server that maintains a cache
  - For better performance
  - For scalability of the system as a whole



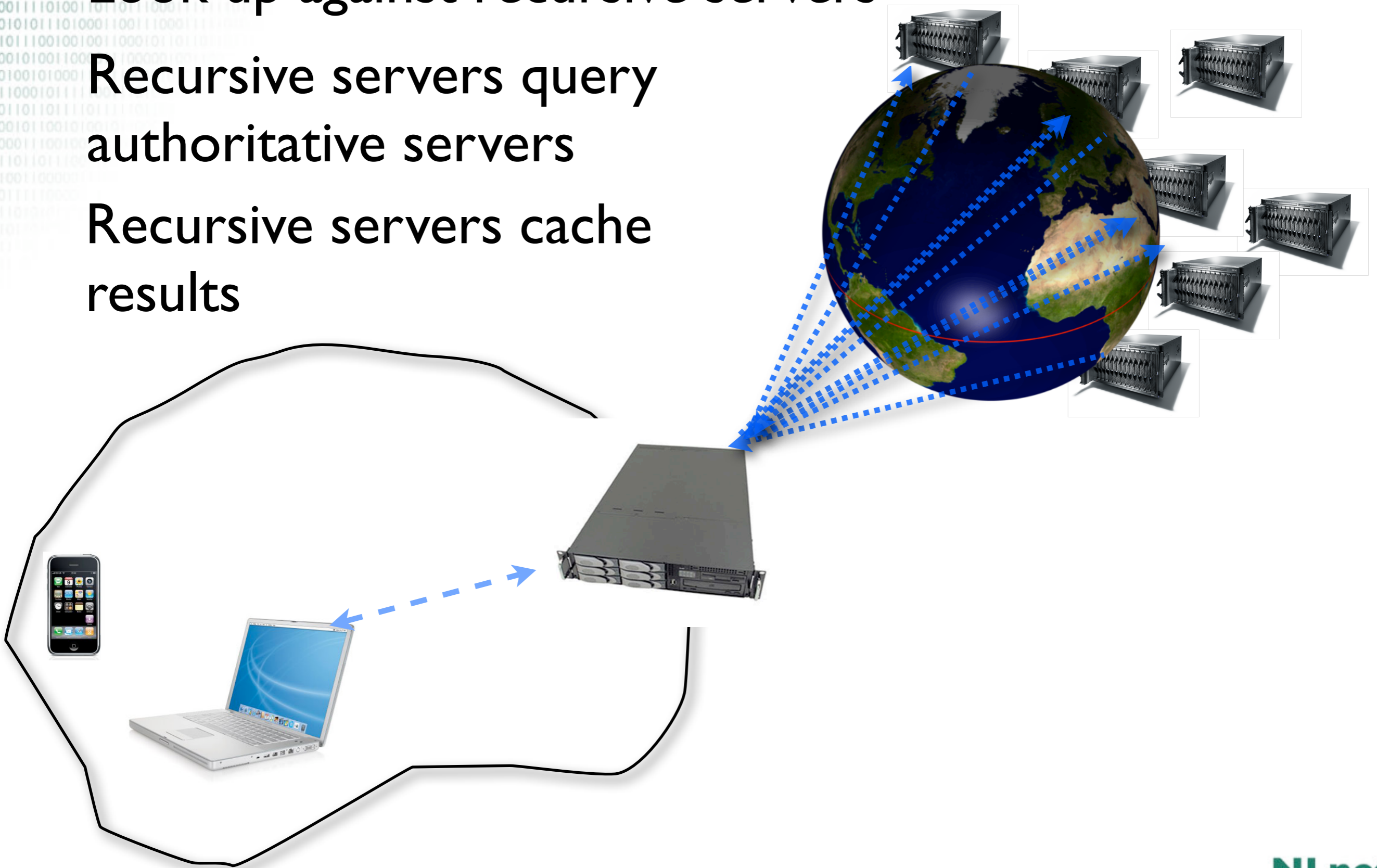




Look up against recursive servers

Recursive servers query authoritative servers

Recursive servers cache results





# When do you use the DNS

- Anytime that you need to know where the other guy is
- DNS is the phone book of the Internet
- So it is used when people make a voice over IP call

10101110010101110110010110011001011110111  
0011101011111110001111011010001111110111  
111110101000111101010100100100111110111  
0010100101110000111010000100000100001  
000011101110100111010010110110000  
1000101101110010110100001000110010001  
0001110100110110110001111110111  
00101011101000110011100011110111  
010111001001001100010110110111  
1001010011000011100000100111  
001001010001111100101010111

enterprise

# Recursive DNS



SIP server



SIP server

Internet

10101100101011011001011001100101110111  
00111010111111000111101101000111110111  
111110101000111101010100100111110111  
0010100101110000111010001000001000111  
00001110111010011101001011011000111  
10001011011100101101000100011001000111  
0001110100110110110001111110111  
00101011101000110011100011110111  
010111001001001100010110110111111111  
100101001100011100001001111111111111  
001001010011111001010111111111111111

enterprise

# Recursive DNS



1.2.3.4.5.6.7.8..0.2.1.3.e164.arpa



## SIP server



## SIP server

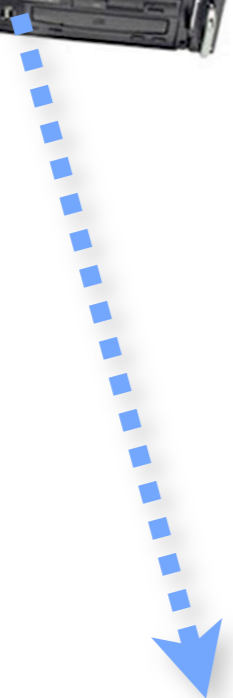


Internet

10101100101011011001011001100101110111  
001101011111100011101101000111110111  
111101010000111101010100100100111110111  
0010100101110000111010000100000100001  
000011101101001110100101101100001111  
1000101101110010110100001000110010001  
0001110100110110110001111110111  
00101011101000110011100011110111  
010111001001001100010110110111  
100101001100001110000010011111  
0010010100011111001010100011111

enterprise

# Recursive DNS



**SIP server**



**SIP server**



Internet

10101110010101110110010110011001011110111  
0011101011111110001111011010001111110111  
11111010100011111010101001001001111110111  
001010010111000011101000010000010000100001  
00001110111010011101001011101100001  
1000101101110010110100001000110010001  
00011101001101101110001111110101  
00101011101000110011100011110101  
010111001001001100010110110101  
10010100110000111000001001001001  
001001010001111100101010001

enterprise

# Recursive DNS



# SIP server



# SIP server



Negotiation and call setup

Internet

10101110010101110110010110011001011110111  
0011101011111110001111011010001111110111  
111110101000111101010100100100111110111  
001010010111000011101000010000010000100001  
00001110111010011101001011101100001  
1000101101110010110100001000110010001  
00011101001101101110001111110111  
00101011101000110011100011110001  
010111001001001100010110110111  
1001010011000011100000100110011  
00100101000111110010101000110011

enterprise

**Recursive DNS**



CALL Established

**SIP server**



Internet

**SIP server**



**DNSSEC**

**NLnet Labs**

© 2006-2008 NLnet Labs

Or they use the DNS  
when sending MAIL

101011100101011101100101100110010111101111  
0011101011111111000111101101000111111111  
111110101000111101010100100100111111111  
0010100101110000111010000100000100001  
1000101101110010110100001000110010001  
0001110100110110111000111111110101  
0101110010010011000101101100010101  
100101001100011100001001000100010001  
00100101001111110101010101010101010101

enterprise

**Recursive DNS**



**Mail server**

Internet

**Mail server**



**DNSSEC**

**NLnet Labs**

© 2006-2008 NLnet Labs



Or they use the DNS  
when browsing the  
Web

10101110010101110110010110011001011110111  
001110101111111000111101101000111111011  
11111010100011110101010010010011111011  
00101001011100001110100001000000103272  
000011101110100111010010110110000  
1000101101110010110100001000110010001  
0001110100110110111000111111011  
00101011101000110011100011110001  
01011100100100110001011011011  
100101001100001110000010011001  
0010010100011111001010100001

enterprise

# Recursive DNS



Web server



Internet

# Or they use the DNS

- When downloading Software upgrades
- Sharing their agenda
- Uploading tax forms
- Instant messaging with friends
- Connect to their security camera
- Figure out the latest news about that merger

# So DNS is IMPORTANT

- How would an attacker use the DNS for attacks?
- By fooling the receiver that a service lies elsewhere

# So DNS is IMPORTANT

- How would an attacker use the DNS for attacks?
- By fooling the receiver that a service lies elsewhere

Back to our VOIP example

10101110010101110110010110011001011110111  
0011101011111110001111011010001111110111  
111110101000111101010100100100111110111  
0010100101110000111010000100000100001  
0000111011101001110100101101100001  
1000101101110010110100001000110010001  
0001110100110110110001111110111  
00101011101000110011100011110111  
010111001001001100010110110111  
1001010011000011100000100110111  
0010010100011111001010110111

enterprise

# Recursive DNS



SIP server



SIP server



Internet

10101100101011011001011001100101110111  
001101011111100011101101000111110111  
11110101000111101010100100100111110111  
00101001011100001110100001000001000011  
0000111011010011101001011011000011  
1000101011100101101000010001100100011  
0001110100110110110001111110111  
00101011101000110011100011110111  
01011001001001100010110110111  
1001010011000011100000100111  
001001010001111100101010111

enterprise

# Recursive DNS



SIP server



SIP server



Internet



Labs

10101100101011011001011001100101110111  
001101011111100011101101000111110111  
11110101000111101010100100100111110111  
0010100101110000111010000100000100001  
000011101101001110100101101100001  
100010110111001011010000100010010001  
0001110100110110110001111110101  
00101011101000110011100011110101  
010111001001001100010110110101  
100101001100001110000010010001  
00100101000111110010101010101010101

enterprise

# Recursive DNS



1.2.3.4.5.6.7.8..0.2.1.3.e164.arpa



SIP server



SIP server



Internet



Labs





1010110010101101100101001100101110111  
001101011111100011101101000111110111  
111101010001111010101001001011110111  
0010100101110000111010000100000100001  
00001110110100111010010110110000  
1000101101100101010000100010010001  
000111010011011011000111110101  
001010110100011001100011110101  
0101100100100100010110110101  
1001010010000110000010010001  
0010010100011110010101010101010101

enterprise

# Recursive DNS



Location of destination Server



SIP server



SIP server

Negotiation and call setup



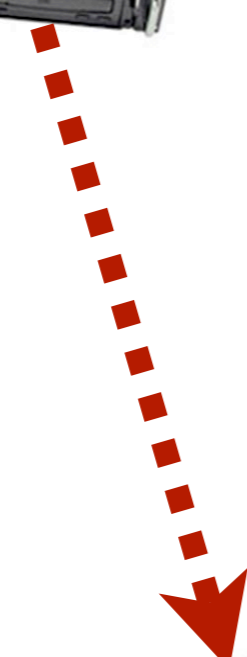
Internet

Labs

1010110010101101100101001100101110111  
001101011111100011101101000111110111  
11110101000111101010100100111110111  
001010010111000011101000100000100011  
00001110110100111010010110110000  
1000101011100101010000100010010001  
000111010011011011000111110101  
0010101101000110011100011110101  
01011001001001000101101101010101  
10010100100001100000100100010001  
00100101000111100101010101010101

enterprise

# Recursive DNS



**SIP server**



**SIP server**



Negotiation and call setup

Internet

Labs

1010111001010111011001011001100101110111  
0011101011111110001111011010001111110111  
111110101000111101010100100100111110111  
0010100101110000111010000100000100001  
000011101110100111010010110110000  
1000101101110010110100001000110010001  
00011101001101101110001111110101  
00101011101000110011100011110101  
010111001001001100010110110101  
100101001100001110000010010001  
001001010001111100101010001

enterprise

# Recursive DNS



CALL Established



# SIP server



# SIP server



Internet



Labs

# Cache Poisoning

- The attack you just saw is called cache poisoning
- Inserting false data into the cache of recursive name servers
- This form of attack has been known for years
- One of the reasons to work on DNSSEC

# Kaminsky's variant

- Classic cache poisoning gave you 'a few tries' to get in between the outgoing question and incoming answer
- Kaminsky came with a scheme where the culprit can keep trying
- Surprisingly simple, a wonder nobody thought of the variety before



# There is Recognition



## US-CERT

UNITED STATES COMPUTER EMERGENCY READINESS TEAM

[Vulnerability Notes Database](#)

[Search Vulnerability Notes](#)

[Vulnerability Notes Help Information](#)

## Vulnerability Note VU#800113

### Multiple DNS implementations vulnerable to cache poisoning

#### Overview

Deficiencies in the DNS protocol and common DNS implementations facilitate DNS cache poisoning attacks.

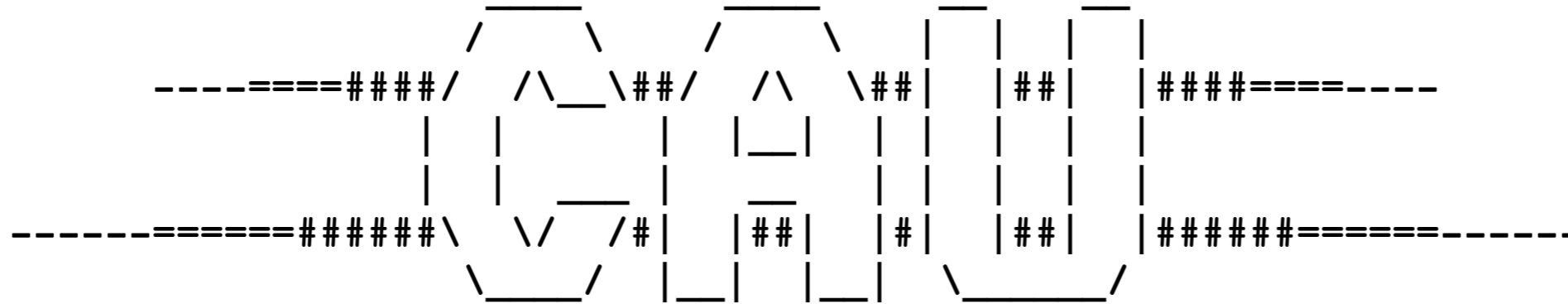
#### I. Description

The Domain Name System (DNS) is responsible for translating host names to IP addresses (and vice versa) and is critical for the normal operation of internet-con

<http://www.kb.cert.org/vuls/id/800113>



# There is Exploit Code



Computer Academic Underground  
<http://www.caughq.org>  
Exploit Code

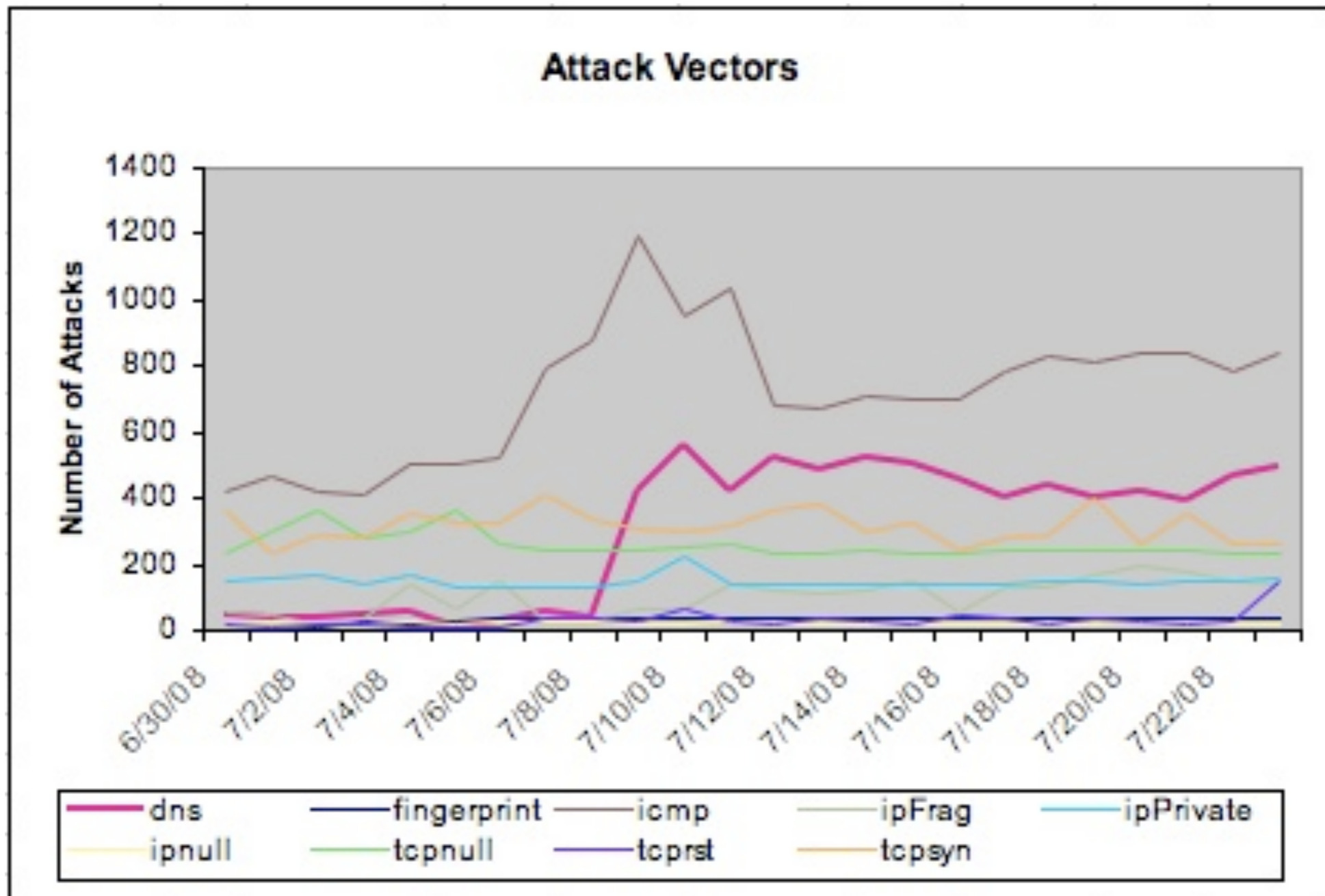
```
=====/  
Exploit ID: CAU-EX-2008-0002  
Release Date: 2008.07.23  
Title: bailiwicked_host.rb  
Description: Kaminsky DNS Cache Poisoning Flaw Exploit  
Tested: BIND 9.4.1-9.4.2  
Attributes: Remote, Poison, Resolver, Metasploit  
Exploit URL: http://www.caughq.org/exploits/CAU-EX-2008-0002.txt  
Author/Email: I)ruid <druid (@) caughq.org>  
H D Moore <hdm (@) metasploit.com>  
=====/  
=====
```

# And more exploit code

```
/*  
 * 2008+ Copyright (c) Evgeniy Polyakov <johnpol@2ka.mipt.ru>  
 * All rights reserved.  
 *  
 * This program is free software; you can redistribute it and/or modify  
 * it under the terms of the GNU General Public License as published by  
 * the Free Software Foundation; either version 2 of the License, or  
 * (at your option) any later version.  
 *  
 * This program is distributed in the hope that it will be useful,  
 * but WITHOUT ANY WARRANTY; without even the implied warranty of  
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the  
 * GNU General Public License for more details.  
 */
```


<http://tservice.net.ru/~s0mbre/archive/dns/>

# The networks are scanned



<http://asert.arbornetworks.com/2008/07/30-day-of-dns-attack-activity/>

# There have been successful attacks

 **Today's Internet Threat Level: GREEN**  
Handler on Duty: Jim Clausing GREEN

Diary Trends Reports About Presentations Top 10 Contact

Handler's Diary: Joomla user password reset vulnerability being actively exploited;Upcomi

## Diary

[previous](#) [next](#)

### DNS Cache Poisoning Issue Update

Published: 2008-07-30,  
Last Updated: 2008-07-30 21:20:49 UTC  
by David Goldsmith (Version: 1)

4 comment(s) [Digg](#) [submit](#)

Ok, we have a confirmed instance where the DNS cache poisoning vulnerability was used to compromise a DNS server belonging to AT&T. This PCWorld [article](#) covers the incident. The original article makes it sound as though the Metasploit site was 'owned' by this incident when really the issue was that the AT&T DNS server was compromised and was providing erroneous IP addresses to incoming queries. This updated PCWorld [article](#) clarifies the first one.

Additional details can be found in this Metasploit [blog post](#).

So we've moved from "the bad guys are out there" past "the invaders are at the gate" and on to "the bad guys are slipping inside". If your organization has not yet patched your DNS servers (see [here](#)) , please do so now.

We may be raising our InfoSec status to yellow soon to help raise attention to the serious nature of this issue.

<http://isc.sans.org/diary.html?storyid=4801>

Bits	50%	5%	Aka
16	10 s	1 s	Unpatched server, random ID
26	2.8 h	17 m	Patched, using only 1024 ports
34	28 days	2.8 days	unbound with defaults
44	28444 days	2844.4 days	unbound with 0x20 and source addresses configured

# 50%-5%-0.5%-0.05%

- There are literally millions of resolvers out there
- The calculations are based on certain assumptions
  - Scanning of Port ID and Query ID are independent: multiplication of chances?
- All steps in an arms run, do we count on the next quick fix or the solution that has been designed to cope with this?



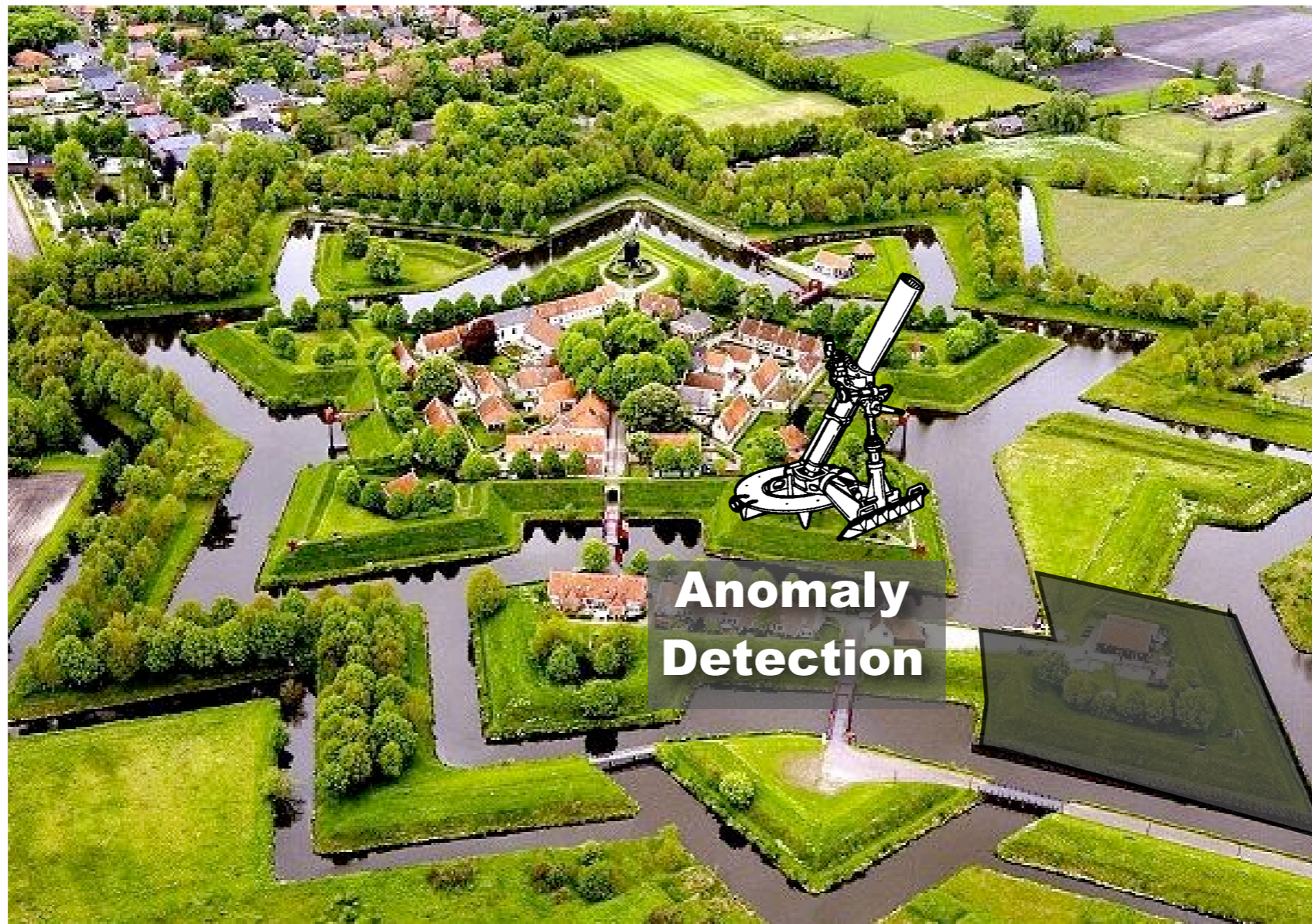
10101100101011011001011001100101110111  
0011010111111000111011010001111111  
111101010000111101010100100100111111  
00101001011100000110100001000000100000  
00001110110100111010010110110000  
10001010110010101000010001001000  
000111010010101011000111110101  
0010101101000100110001110101  
01011001001001000101101101  
10010100100001100000100101  
0010010100011111001010101  
110001011110011010101  
1011011011100101  
11001000010101  
1011110000101  
01101010101  
1101



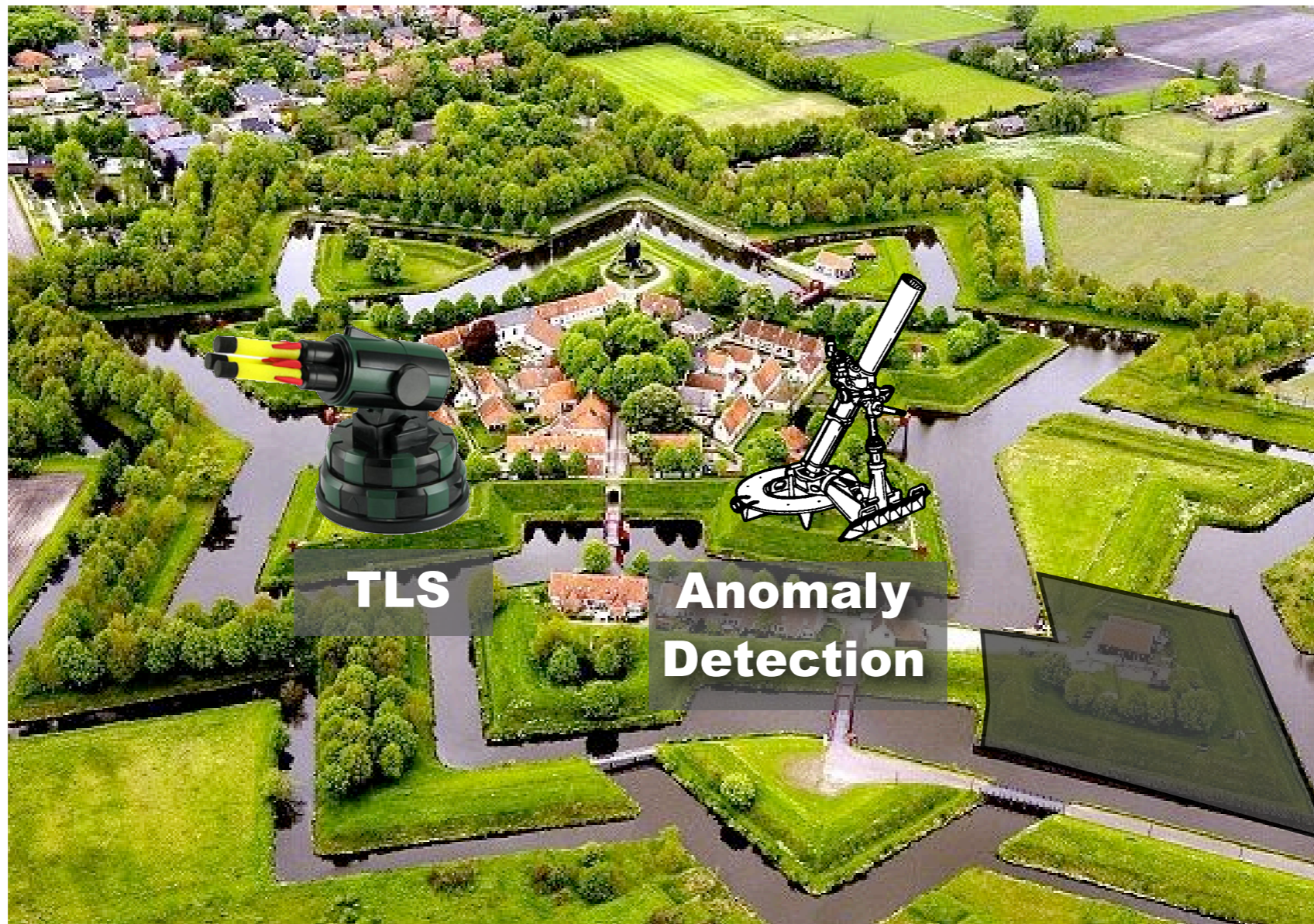




We lost DNS...  
How about the other defenses ?



We lost DNS...  
How about the other defenses ?



We lost DNS...  
How about the other defenses ?

# SSL?

- Current practices are sloppy
- Users connect to their banks
- Get redirected to unrelated domains
- User interfaces only show padlocks

# For example

## Mastercard



Press "Go to Your Bank" to authorise your credit card payment at your bank.

Amount **415.00 Euro**

Payment cluster ID **167102578**

Please deactivate your pop-up killer in your web browser, before proceeding with your payment.

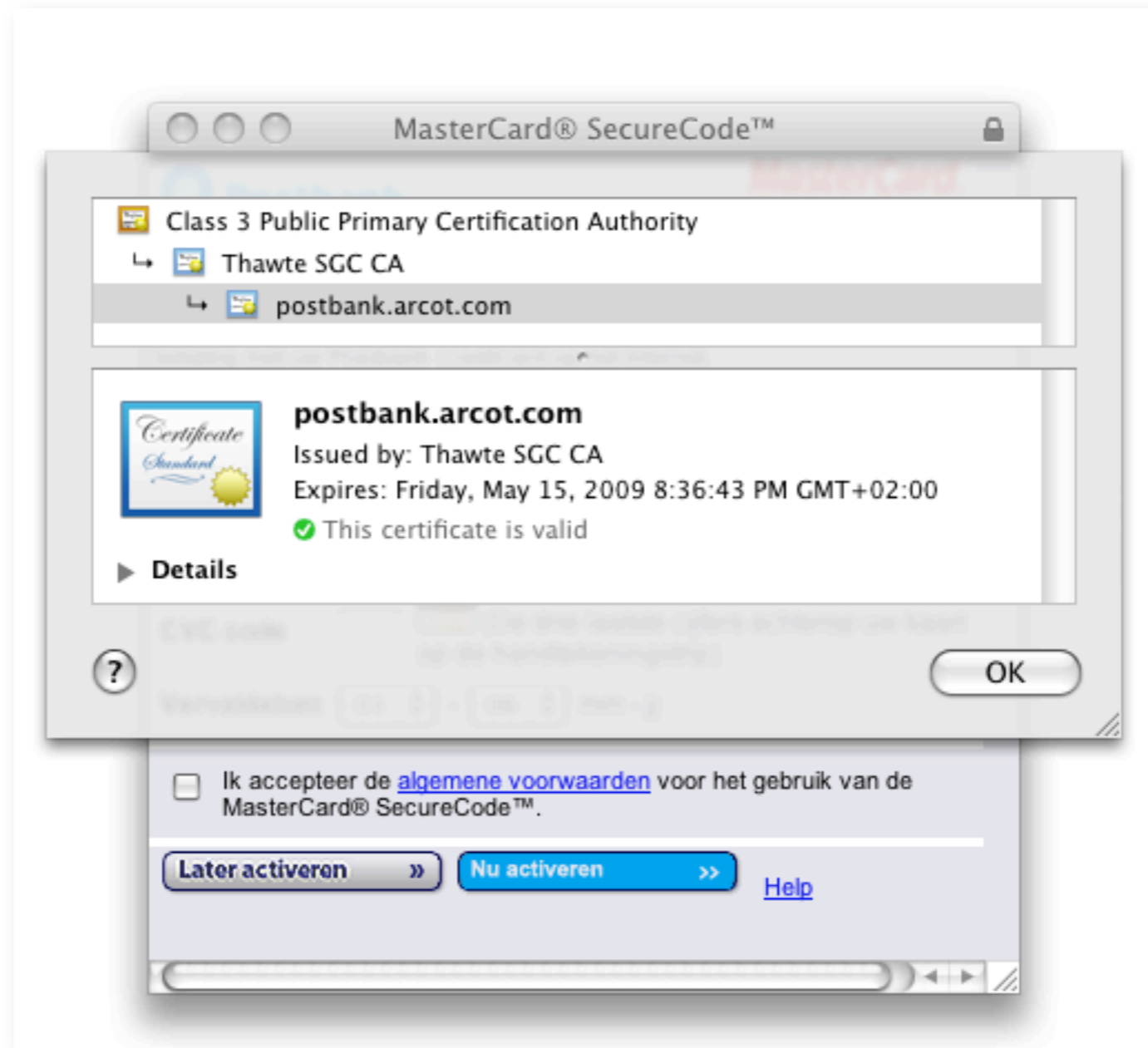
[more info](#)

To Your Bank

# For example

The screenshot shows a web browser window titled "MasterCard® SecureCode™". The page features the Postbank logo on the left and the MasterCard SecureCode logo on the right. A blue banner at the top reads "Met Postbank veiliger winkelen op het internet!". Below this, a paragraph explains that Postbank offers this service for free and provides instructions on how to activate it. A section titled "Stap 1: Controle persoonlijke gegevens" contains a form with the following fields: "Naam" (Name) with a text input box; "CVC code" with a text input box, a calendar icon, and a note "(De drie laatste cijfers achterop uw kaart op de handtekeningstrip)"; and "Vervaldatum" (Expiration date) with two dropdown menus showing "01" and "06" followed by "mm - jj". At the bottom of the form, there is a checkbox labeled "Ik accepteer de algemene voorwaarden voor het gebruik van de MasterCard® SecureCode™." and two buttons: "Later activeren >>" and "Nu activeren >>". A "Help" link is also present.

# For example



# Exploit

- Attacker poisons DNS for www.postbank.nl
- Fake www.postbank.nl redirects to [postbank.webbanksecurity.com](http://postbank.webbanksecurity.com)
- Obtaining the domain name and certificate is trivial for organized criminals
- Users are used to these sort of redirections and the domainname looks trustworthy

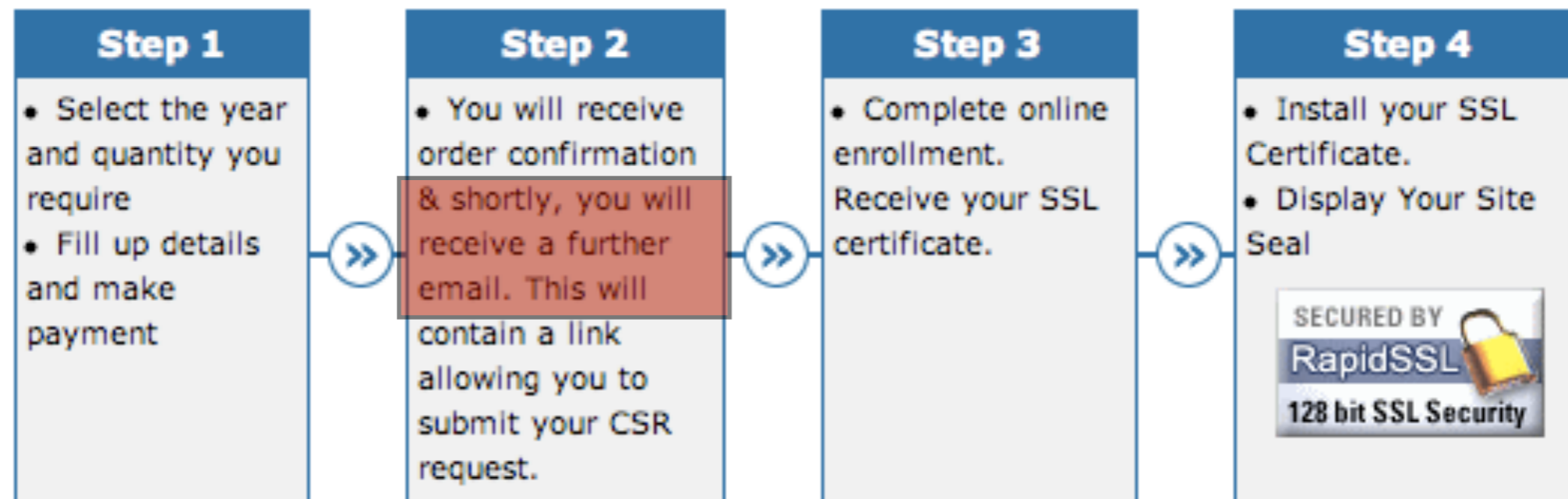


# Things get worse

- Fake `www.postbank.nl` redirects to fake `https://www.postbank.nl`
- SSL protects against that?
- Not if the attacker has a signed certificate
  - How would an attacker do that?

## How SSL purchase works?

Ordering SSL from rapidsslonline.com online store is easy, fast and secure!  
You need to go through 4 simple steps to complete your SSL order



\*\*\* As part of GeoTrust's ongoing commitment to prevent fraud, some orders are randomly flagged for an additional security review. Please note that this order will not be fulfilled until GeoTrust completes this manual security review. Usually such orders are processed within 24 hours but sometimes may take longer than 24 hours. Please contact us via Email or Live Chat for Support in such cases.

<http://www.rapidsslonline.com/index.php>

# Don't rely on DNS for the Security review

- Don't get the contact details out of the WHOIS, getting to WHOIS is DNS based
- Don't send confirmation e-mails to typical addresses in the domain
- Mail uses the DNS
- Don't try to see if domain already has a SSL certificate installed. That uses the DNS

# Lower hanging fruit: email

- Just attack e-mail
- Eavesdropping on e-mail
- Modifying text
- Inserting malicious content

enterprise

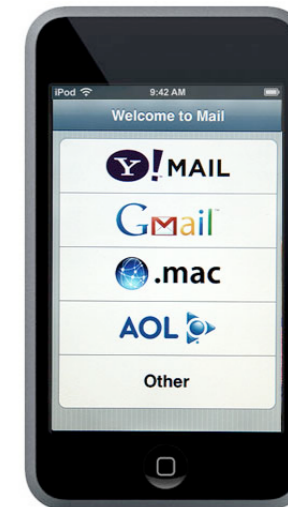
# Recursive DNS



# Mail server



Internet



# Mail server

# DNSSEC

# NLnet Labs

© 2006-2008 NLnet Labs

enterprise

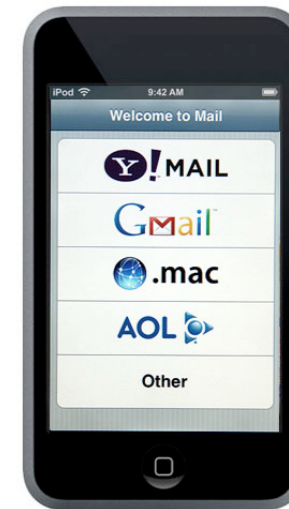
# Recursive DNS



**DNSSEC**



**Mail server**



**Mail server**

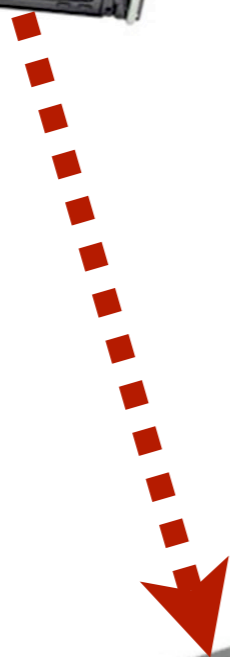
Internet



**Labs**

enterprise

# Recursive DNS



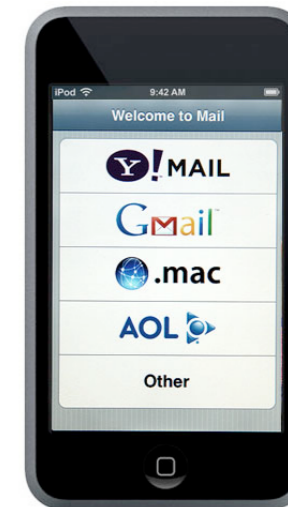
## Mail server



DNSSEC



Internet



## Mail server



Labs

enterprise

# Recursive DNS



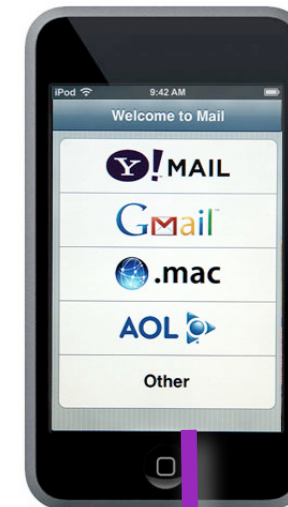
DNSSEC



## Mail server



internet



## Mail server



# Labs



enterprise

# Recursive DNS



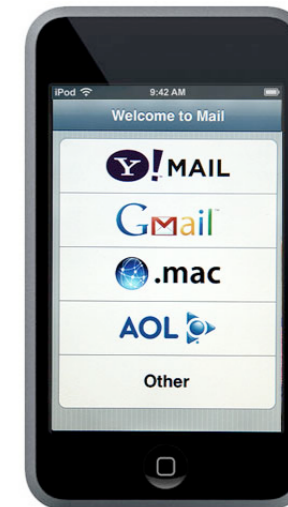
**DNSSEC**



**Mail server**



internet



**Mail server**



**Labs**

# Technique to notice these attacks

- SPF protocol for spam recognition
  - Based on... DNS
- SSL based connections and certification
  - In practice only used for encryption of the channel
  - Often misconfigured, or with fallback in place
  - And remember the possibilities wrt SSL

**Introducing**

**Introducing**

**DNSSSEC**



101011100101011101100101100111001011110111  
0011101011111110001111011010001111110111  
1111101010000111101010100100100111110111  
001010010111000001110100001000000100000  
00001110111010011101001011101100001111  
100010110111001011010000100011001000111  
000111101001101101100011111101111110111  
00101011101000110011100011110111110111  
010111001001001000101101101111111111111  
100101001100001110000010011001111111111  
001001010001111100101010111111111111111  
1110001011110011101011111111111111111111  
101101101110111101111101111111111111111  
0001011001010010100111111111111111111111  
1000111001001001111111111111111111111111  
1110110111001111111111111111111111111111  
1100110000011111111111111111111111111111  
1011111000011111111111111111111111111111  
0110101111111111111111111111111111111111  
0101111111111111111111111111111111111111  
11  
11

# DNSSEC

## What does it protect?

# DNSSEC

## What does it protect?

Let us have another look at the DNS architecture

# Data flow through the DNS

**Registrars  
& Registrants**



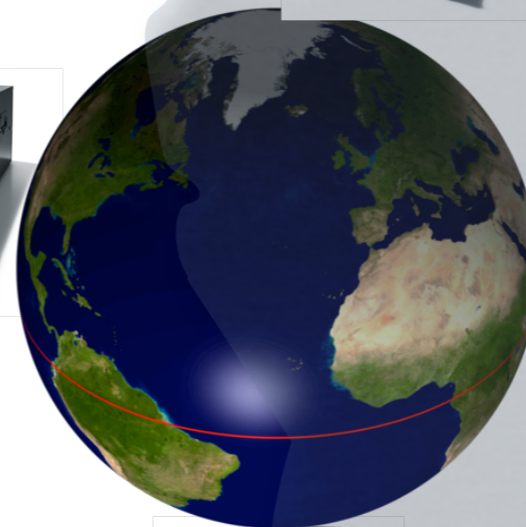
**Secondary  
DNS**



**primary  
DNS**



**Registry**



**Secondary  
DNS**

# Data flow through the DNS

Publication 

**Registrars  
& Registrants**



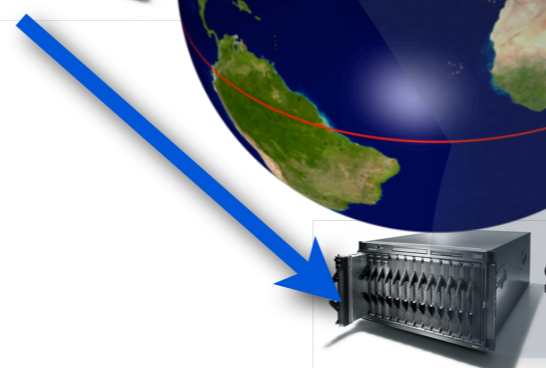
**Registry**



**primary  
DNS**



**Secondary  
DNS**



**Secondary  
DNS**





# Data flow through the DNS

Publication   
Look up 

**Registrars  
& Registrants**



**Registry**



**primary  
DNS**



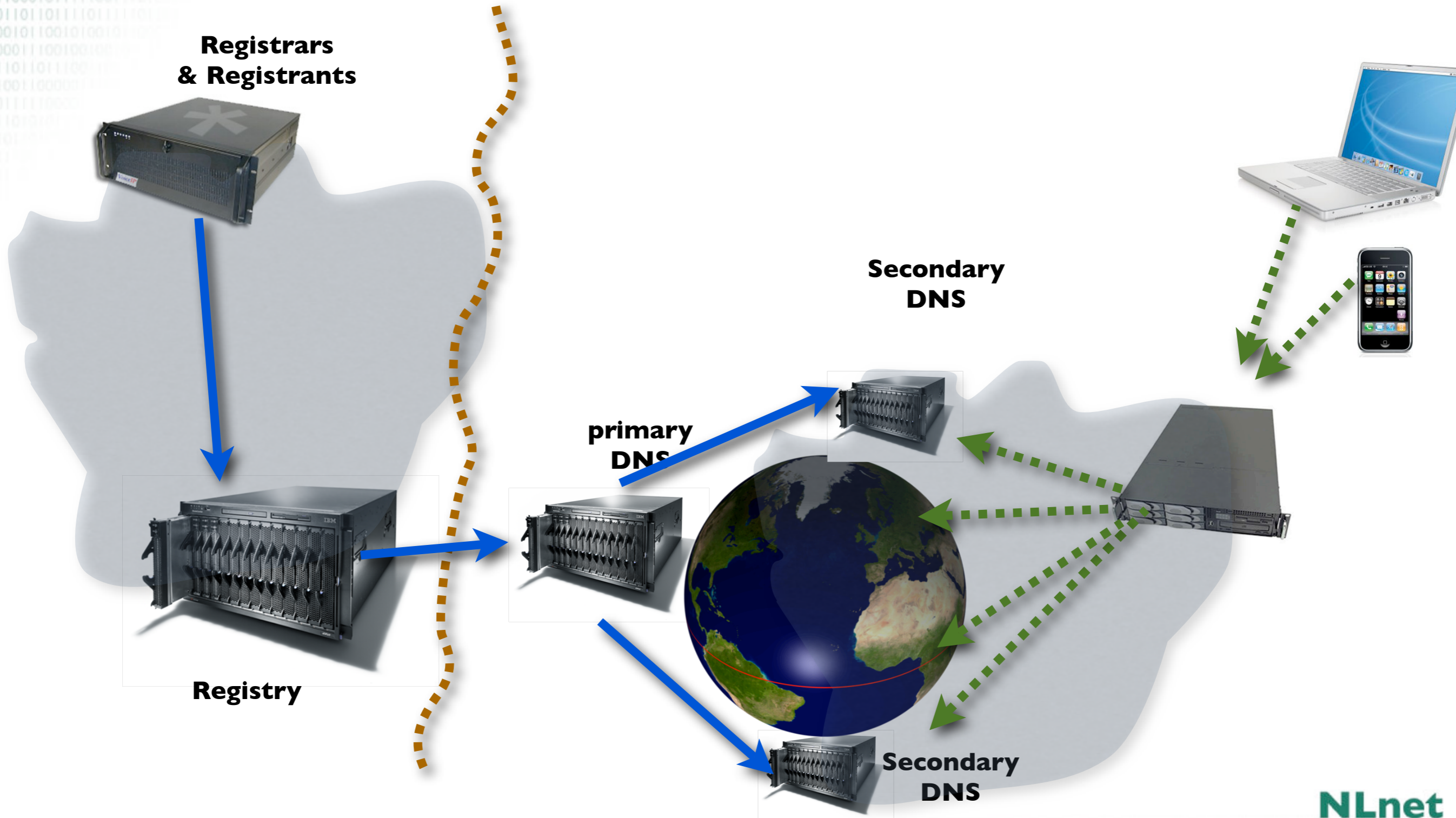
**Secondary  
DNS**



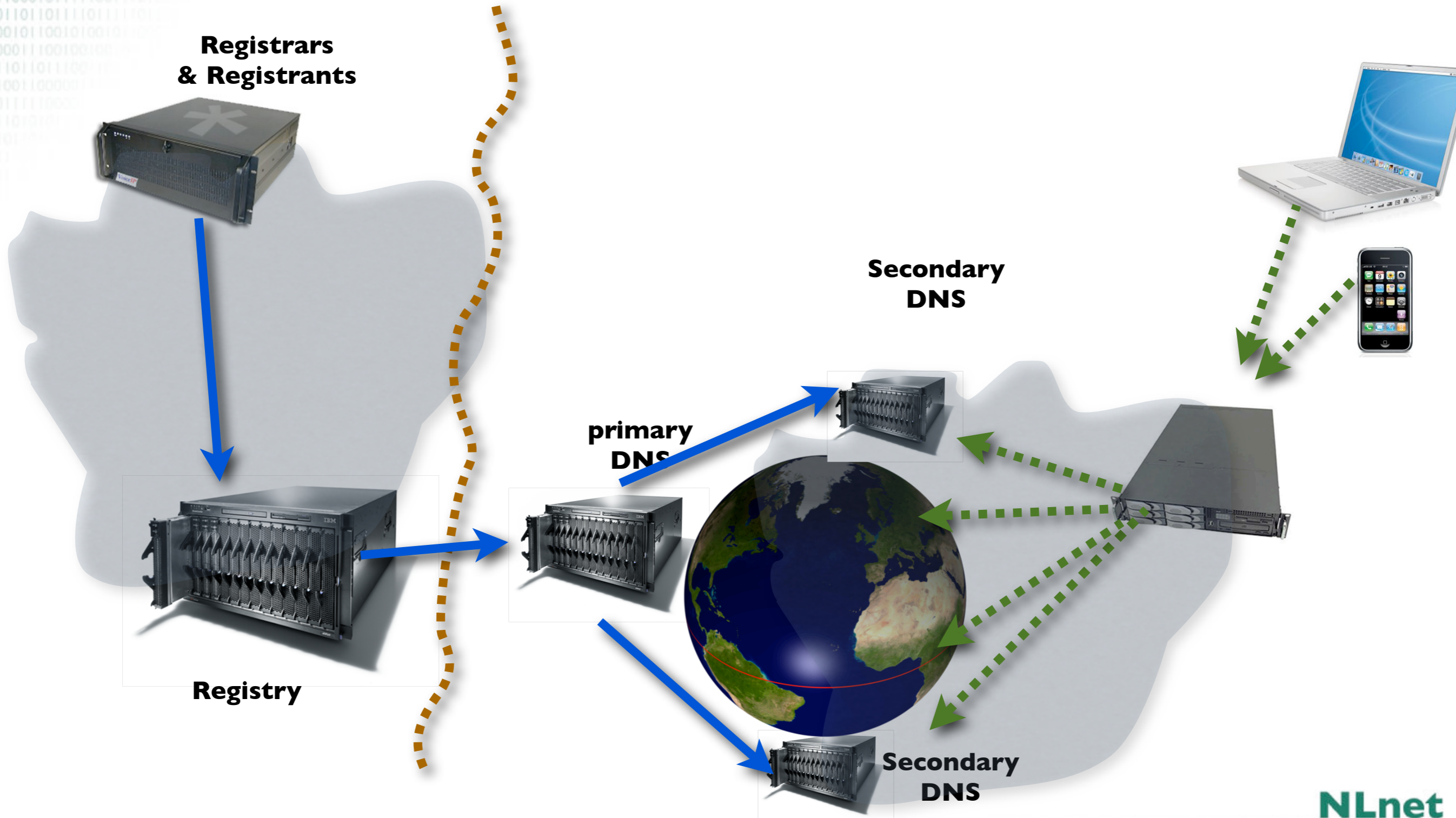
**Secondary  
DNS**



# Data flow through the DNS

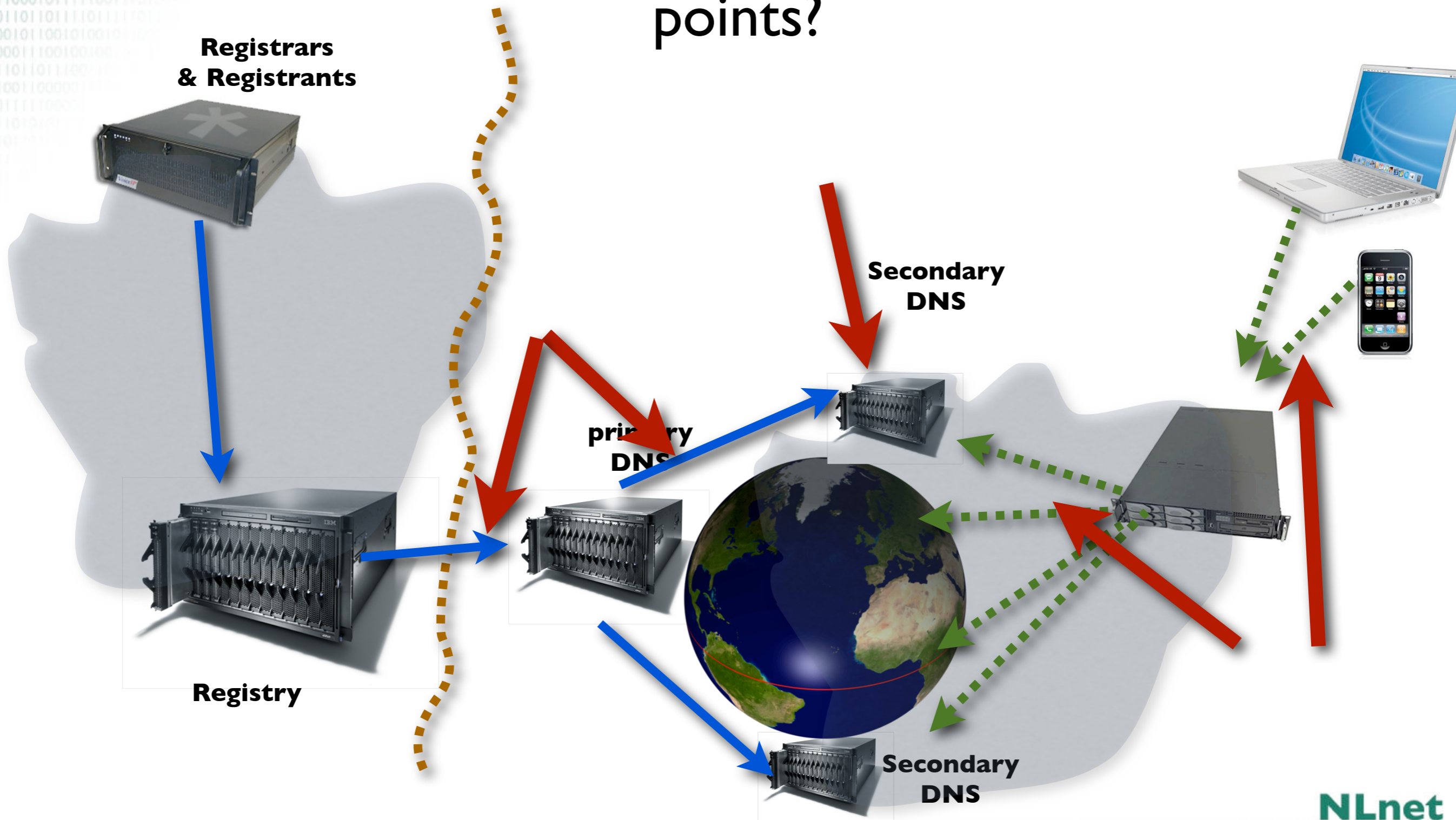


# Data flow through the DNS



# Data flow through the DNS

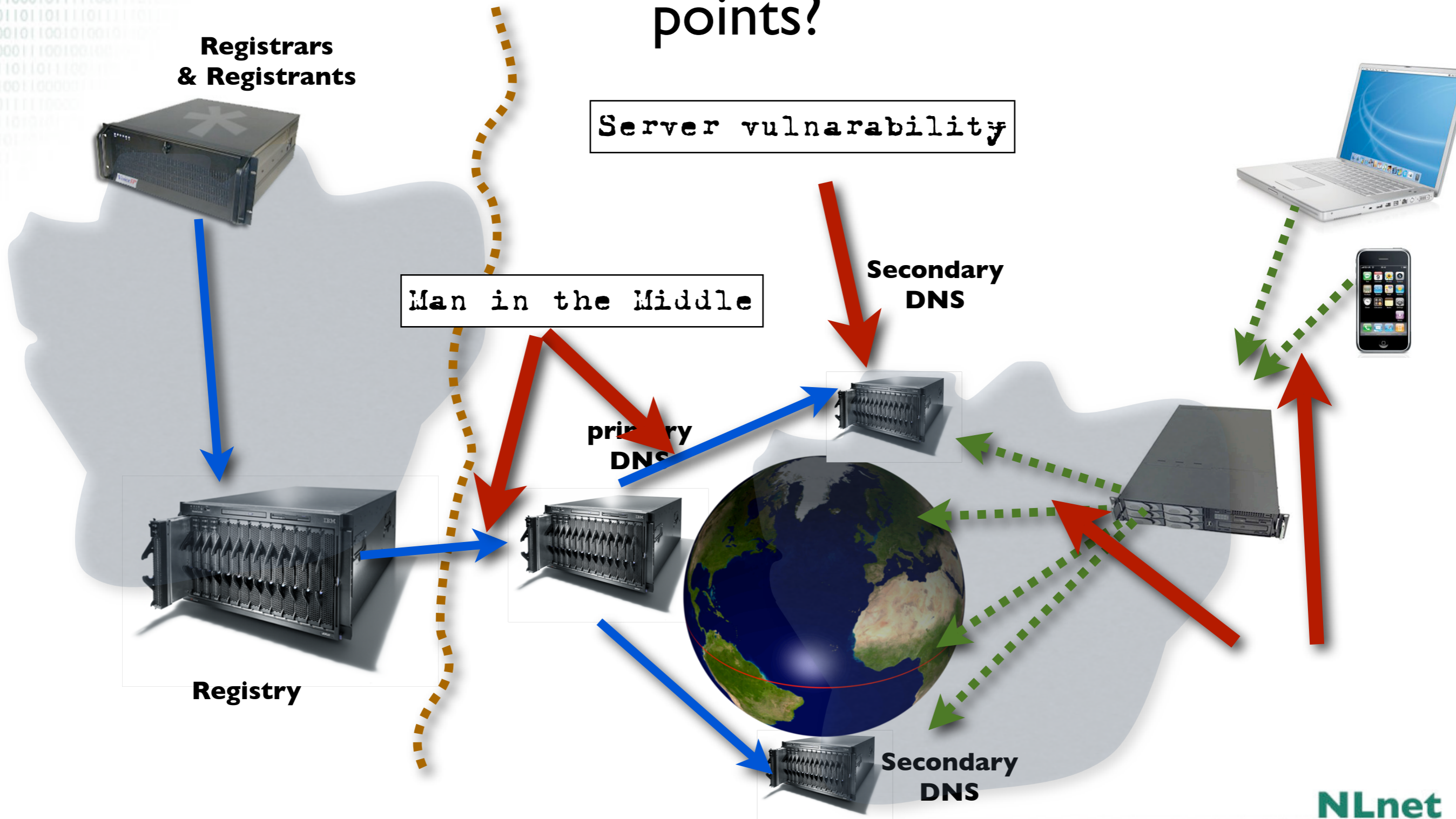
## Where are the vulnerable points?





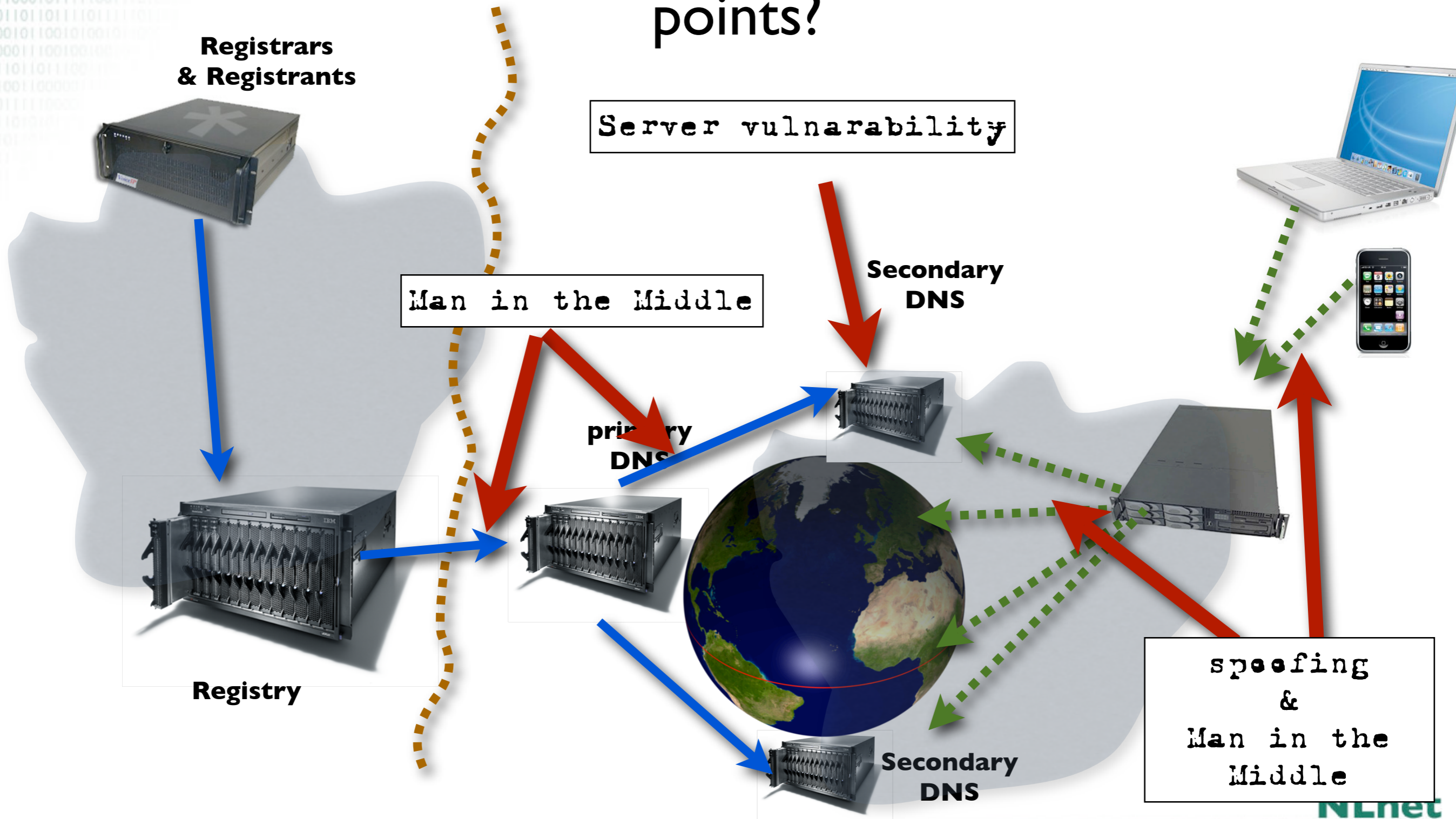
# Data flow through the DNS

## Where are the vulnerable points?



# Data flow through the DNS

## Where are the vulnerable points?



# DNSSEC protects all these end-to-end

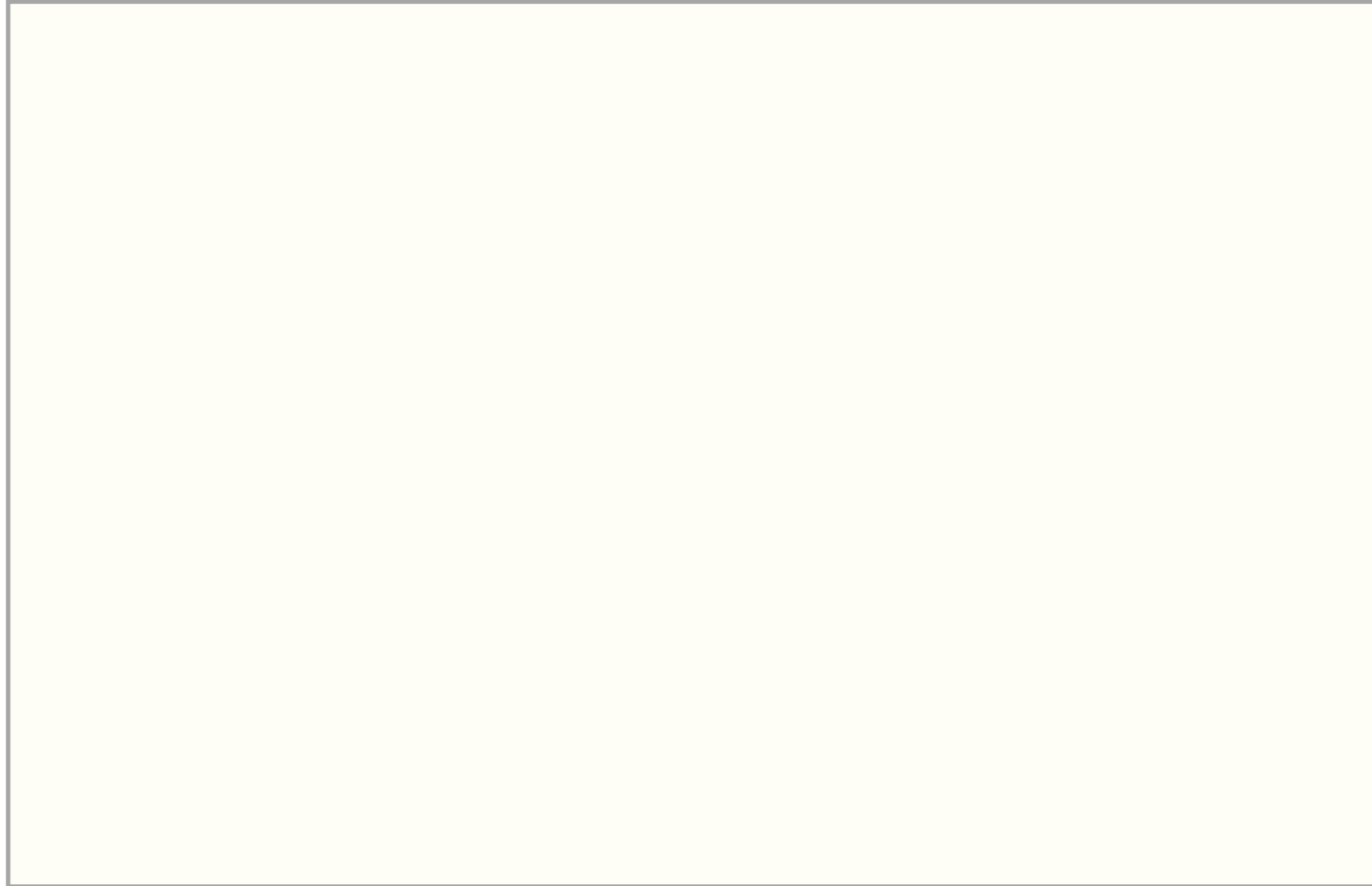
- As an aside:  
There is a protection mechanism against the man in the middle: TSIG
  - Provides hop-by-hop security
  - TSIG is operationally deployed today
  - Based on shared secret: not scalable



# What does DNSSEC provide

- provides message authentication and integrity verification through cryptographic signatures
  - You know who provided the signature
  - No modifications between signing and validation
- It does not provide authorization
- It does not provide confidentiality
- It does not provide protection against DDOS

# Metaphor



# Metaphor

```
www.secret-wg.org A 213.154.48
```

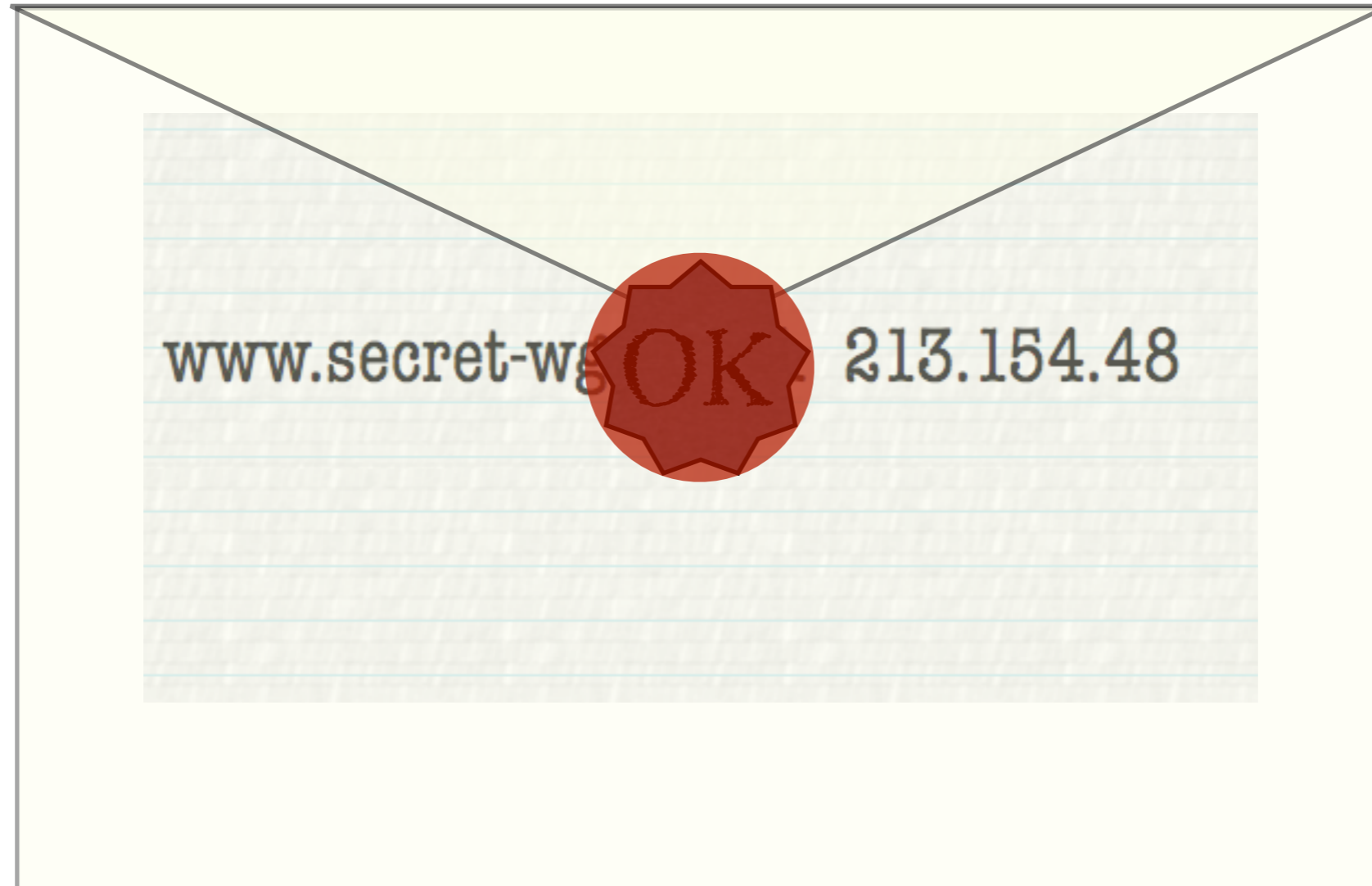
# Metaphor



# Metaphor



# Metaphor



# Metaphor



# Metaphor

- Envelope sealed when data is published in the DNS system





# Metaphor

- Envelope sealed when data is published in the DNS system
- Does not provide confidentiality



# Metaphor



- Envelope sealed when data is published in the DNS system
- Does not provide confidentiality
- The seal protects the delivery process

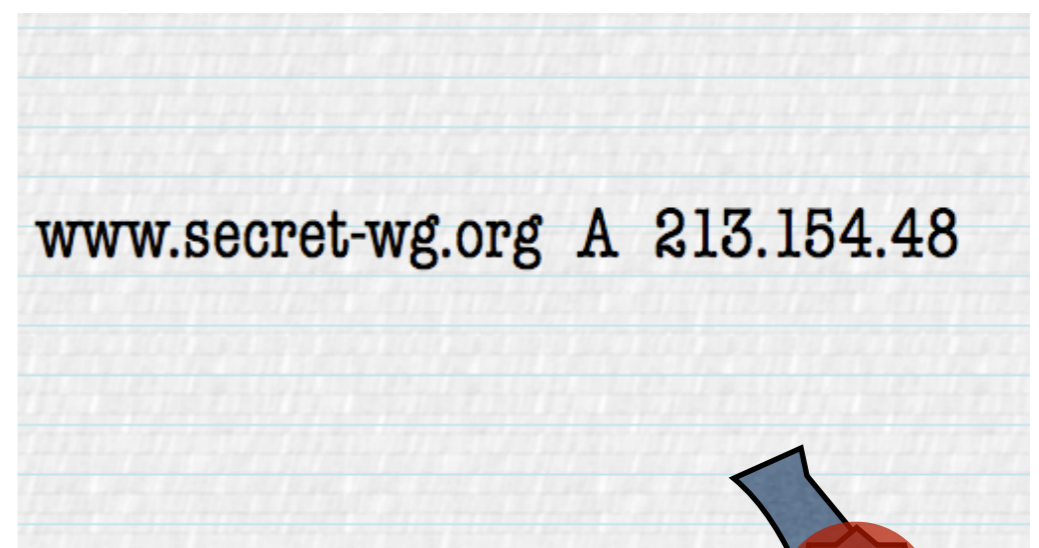
# Metaphor



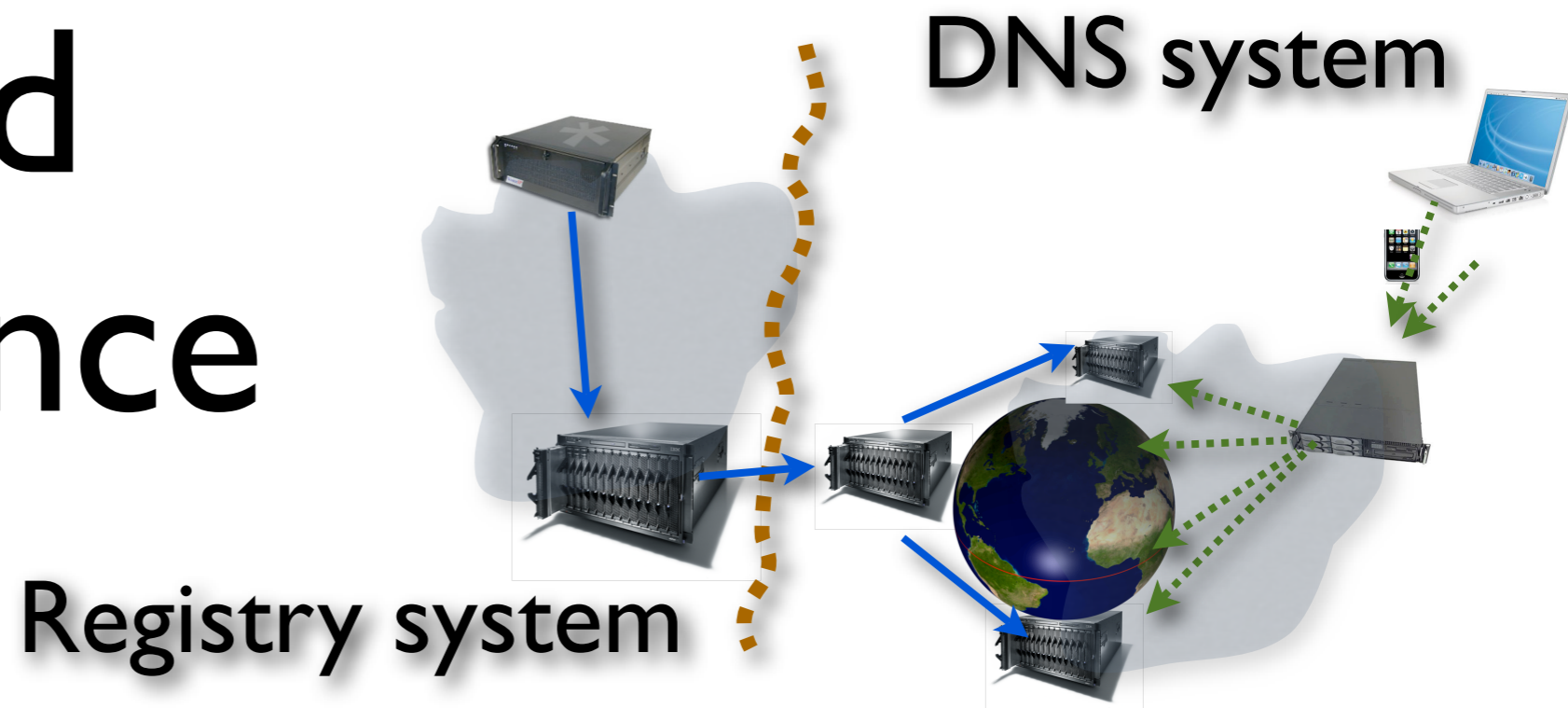
- Envelope sealed when data is published in the DNS system
- Does not provide confidentiality
- The seal protects the delivery process
- No assertion about the message

# Metaphor

- Envelope sealed when data is published in the DNS system
- Does not provide confidentiality
- The seal protects the delivery process
- No assertion about the message



# Trust and Confidence



- DNSSEC enables confidence in the DNS
- It does not change the trust we put in the Registry/Registrar procedures
- Although introduction of DNSSEC may improve some of the procedures

# The mechanism used

- Using public key cryptographic algorithms signatures are applied over the DNS data
- By comparing the signatures with public keys the integrity and authenticity of the data can be established.

# Public key cryptography in a nutshell

- Two large numbers and an encryption and decryption algorithm
- If one of the numbers (the private key) and a message are used for encryption
- The other number (public key) and the decryption algorithm can be used to retrieve the original message





Message

Private Key

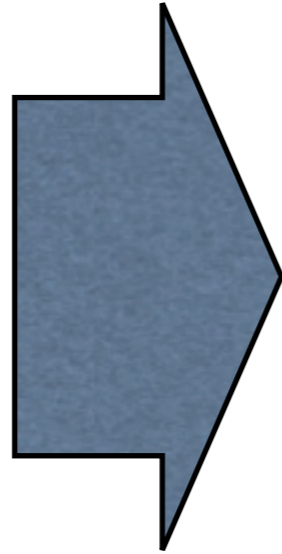
Message

Private Key



Message

Private Key



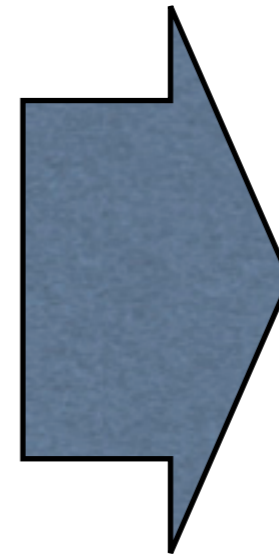
3ncrypt3d

Public Key

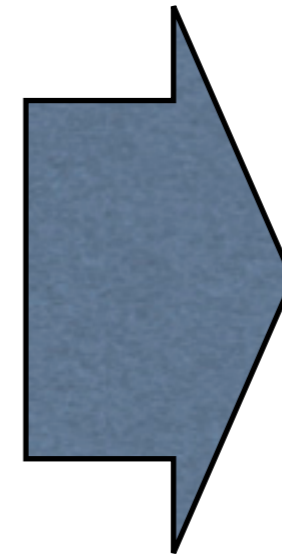
`3ncrypt3d`

Public Key

3ncrypt3d



Public Key  
JncryptJd



Message

Decryption only with matching key:  
If you can decrypt with a public key you  
may assert the message was signed with  
corresponding private key

# Use that for signatures

# Use that for signatures

Message



# Use that for signatures

Message



# Use that for signatures

Message



Message  
Digest

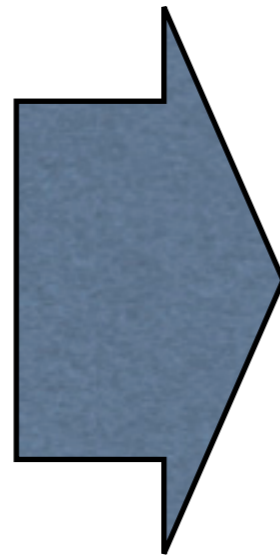
# Use that for signatures

Message



Message  
Digest

Private Key



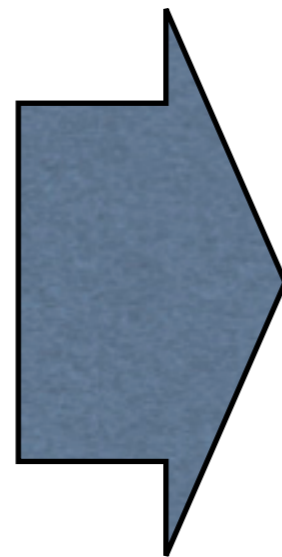
# Use that for signatures

Message



Message  
Digest

Private Key

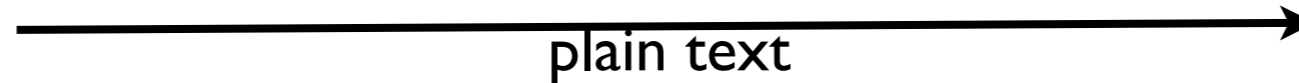


`S19nature`

# Use that for signatures

Message

Message



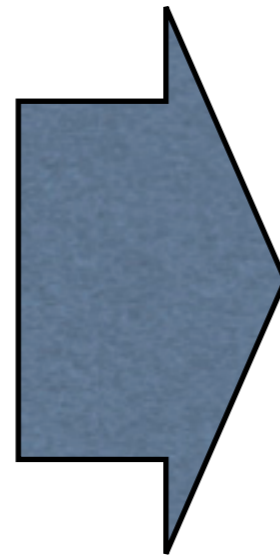
plain text

**S19nature**



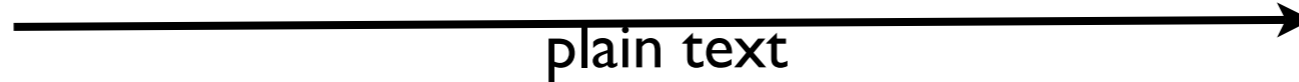
Message  
Digest

Private Key



# Use that for signatures

Message



Message

plain text

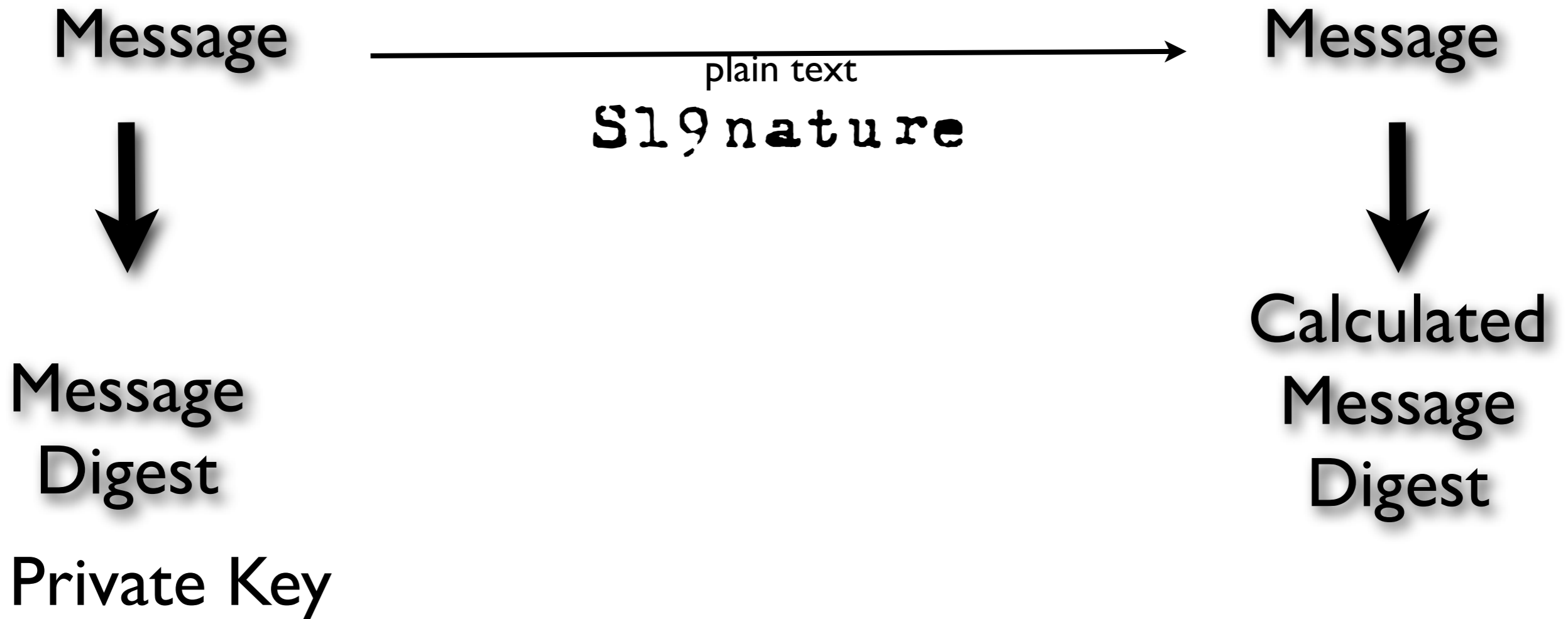
**S19nature**



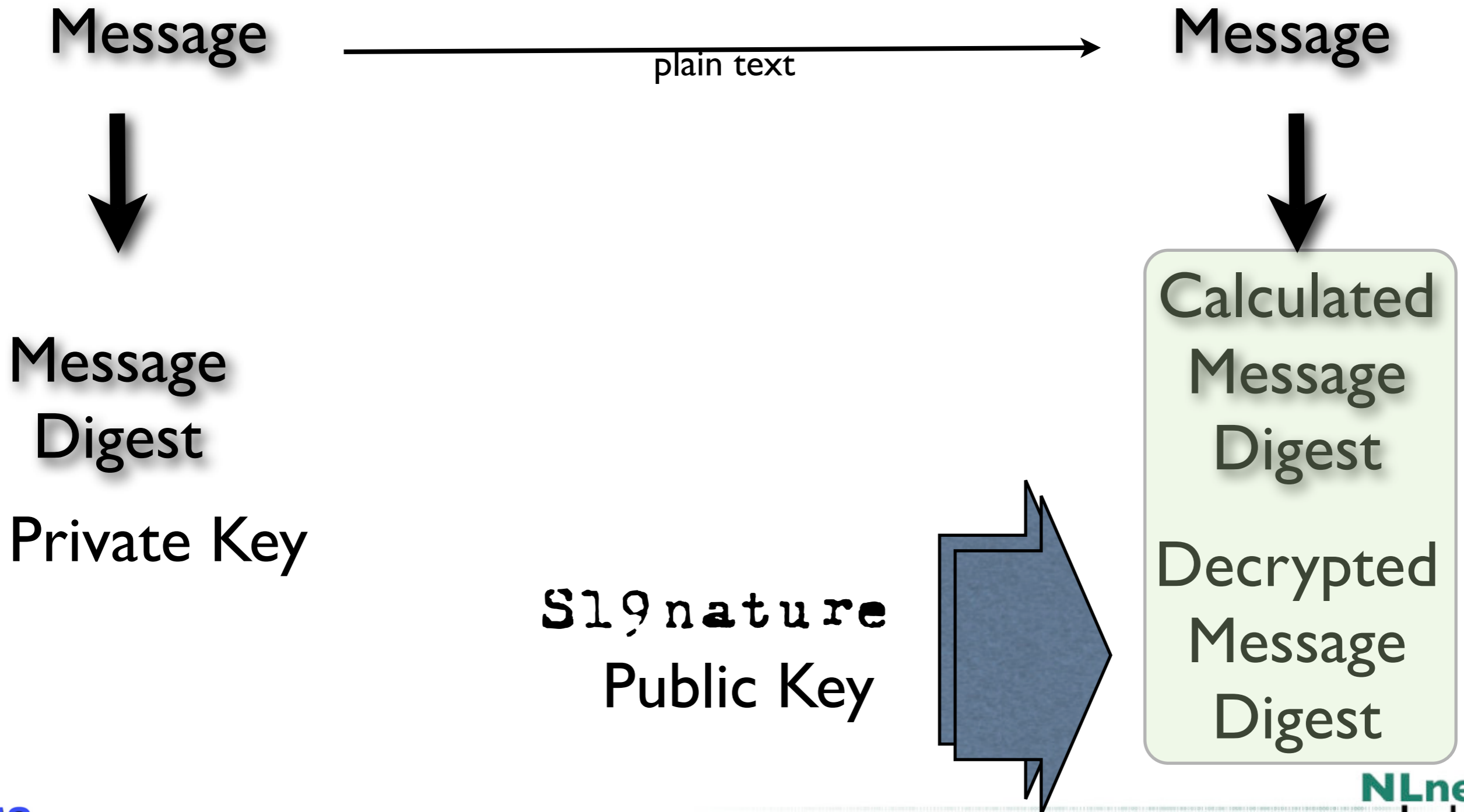
Message  
Digest

Private Key

# Use that for signatures



# Use that for signatures





# Validate Public Keys

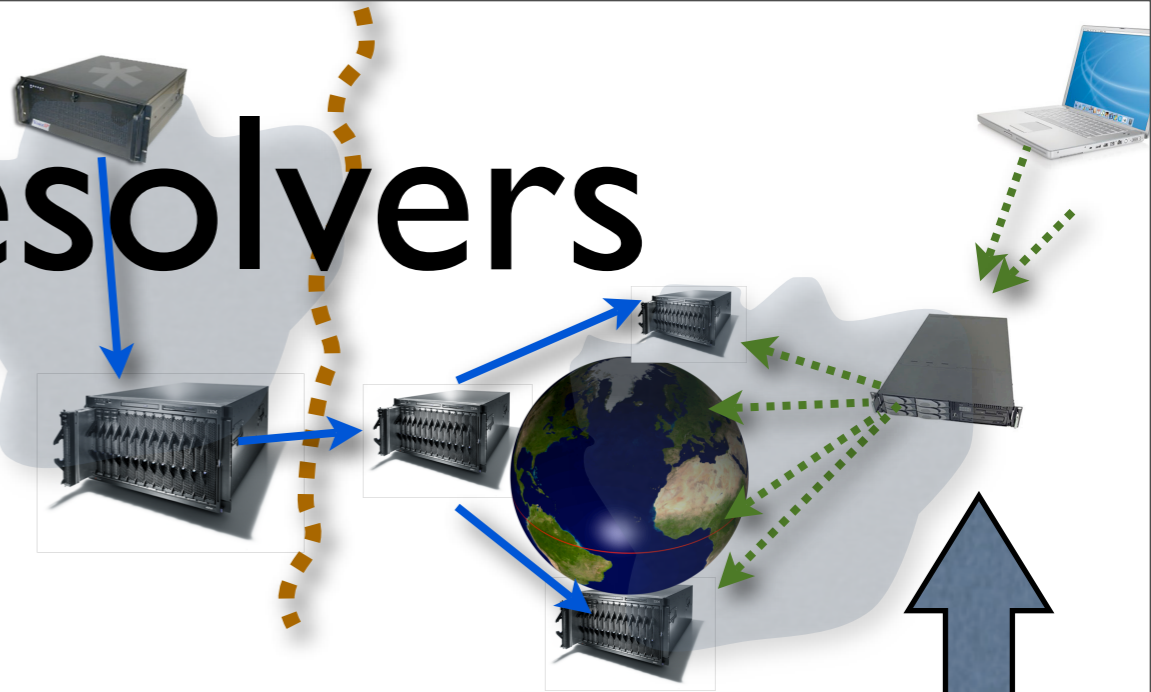
- Make sure you get them from the appropriate entity
- If you validate against the wrong public key there is a problem again
- For DNSSEC: key distribution through the DNS
- Ideally only one key needed: that of the root of the DNS hierarchy (more on that later)

# Steps towards Secure DNS

# Various Players

- Suppose we want confident mapping for [www.bank.in](http://www.bank.in)
- What needst to be done

# From the resolvers view



- The resolver will need to verify the signature over `www.bank.in` is valid
- Two tasks:
  - implement a verifying recursive nameserver
  - configure the appropriate public key
  - maintain the configured public keys

# DNSSEC on a Recursive Nameserver

- Install the appropriate piece of software
- Latest BIND or Unbound
  - Both run on commodity hardware
  - Both are open-source freely available
- Perform the appropriate amount of testing to understand the failure modes

# Configuring Public Keys

- Public keys are configured in the files (manual)
- Make sure public keys are rolled
- Make sure you know the policies of the signing entities
- Not rolling turns into severe failure mode
- Use tools  
(RFC 5011 implementation will be available from NLnet Labs shortly)

# The costs involved

- Commodity hardware
- No need to upgrade Routers and IP equipment
- Though Firewalls may cause issues
- Free software
- The rest is a knowledge exercise

# Public Keys

- Suppose you want to verify the data from all your banks:





# Public Keys

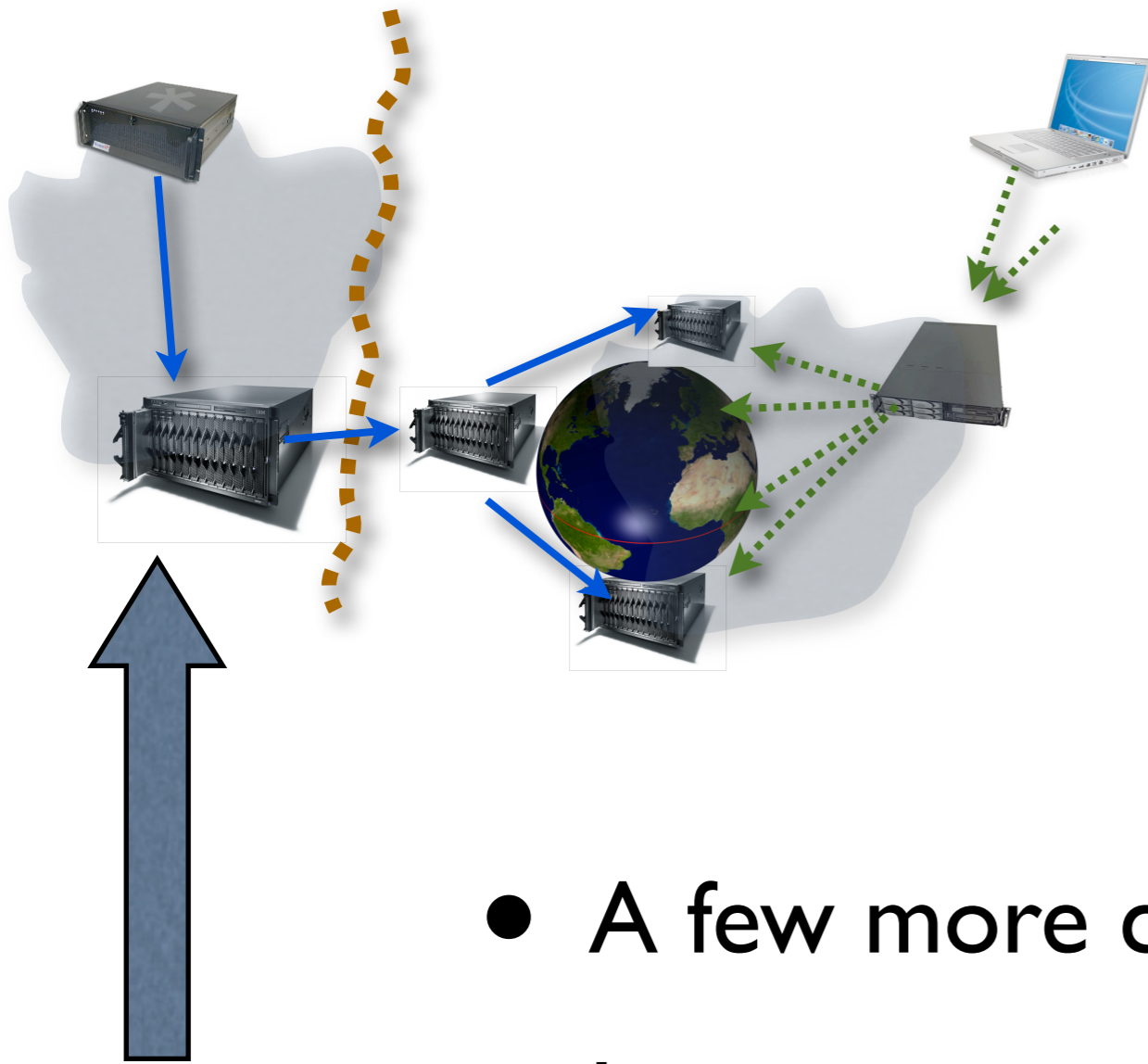
- Suppose you want to verify the data from all your banks:



# Use the DNS for public key distribution

- Publish the public key of bank.in in .in
- Have .IN signed
- Reduces the key-maintenance issues greatly!
- Signing .IN facilitates DNSSEC for all parties involved

# From the Authoritative end



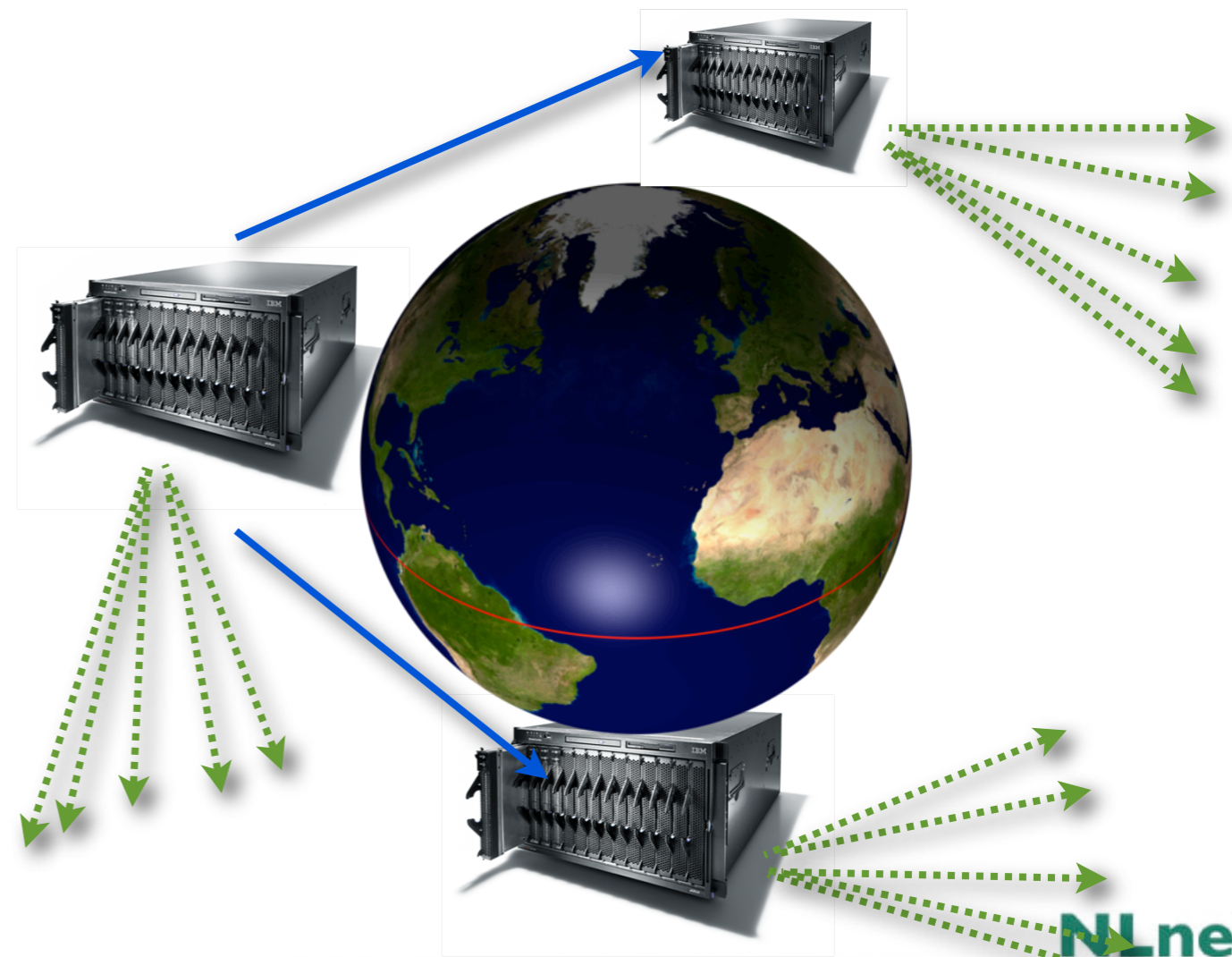
- A few more components are involved
- Let us zoom in on the several components
  - The generic case
  - For a zone with delegations (like .IN)

# Classic DNS



# Classic DNS

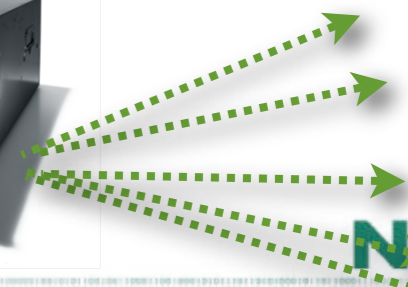
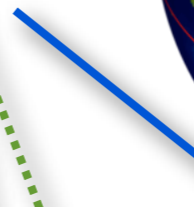
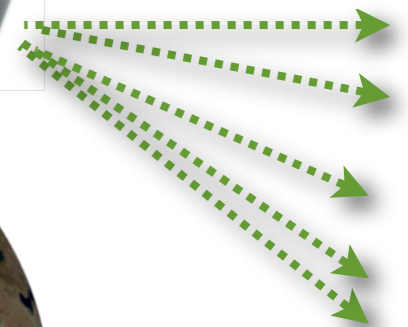
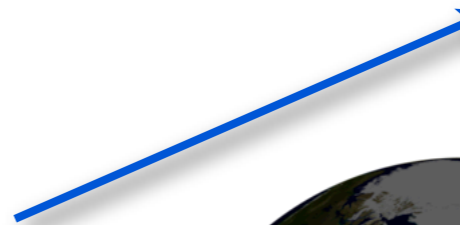
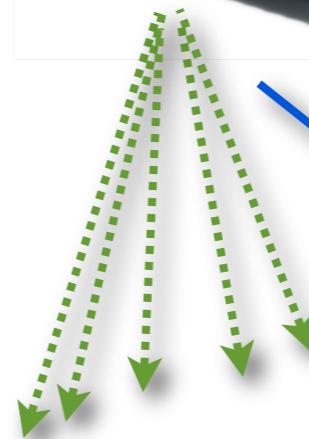
Serving



# Classic DNS

Publishing

Serving



# Classic DNS

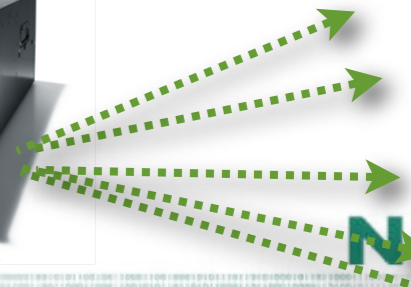
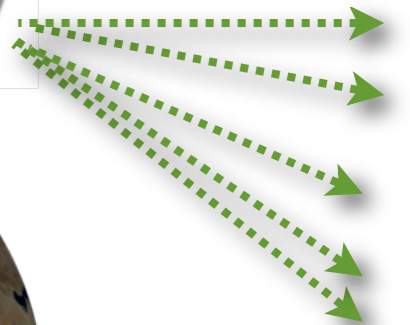
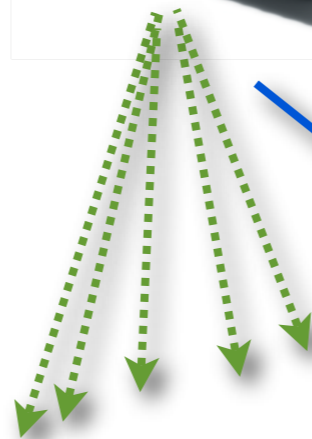
Provisioning



Publishing



Serving

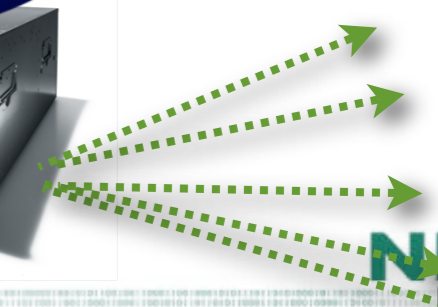
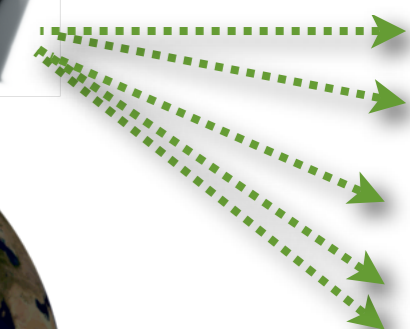
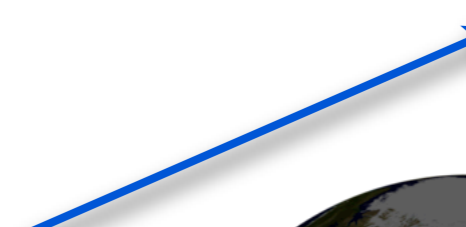
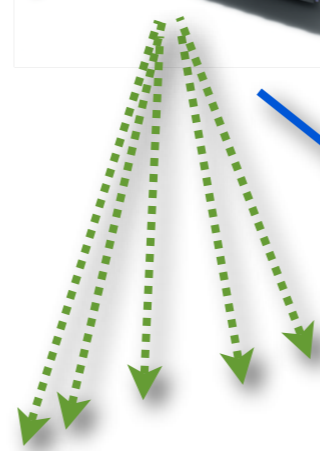


# Introducing DNSSEC

Provisioning

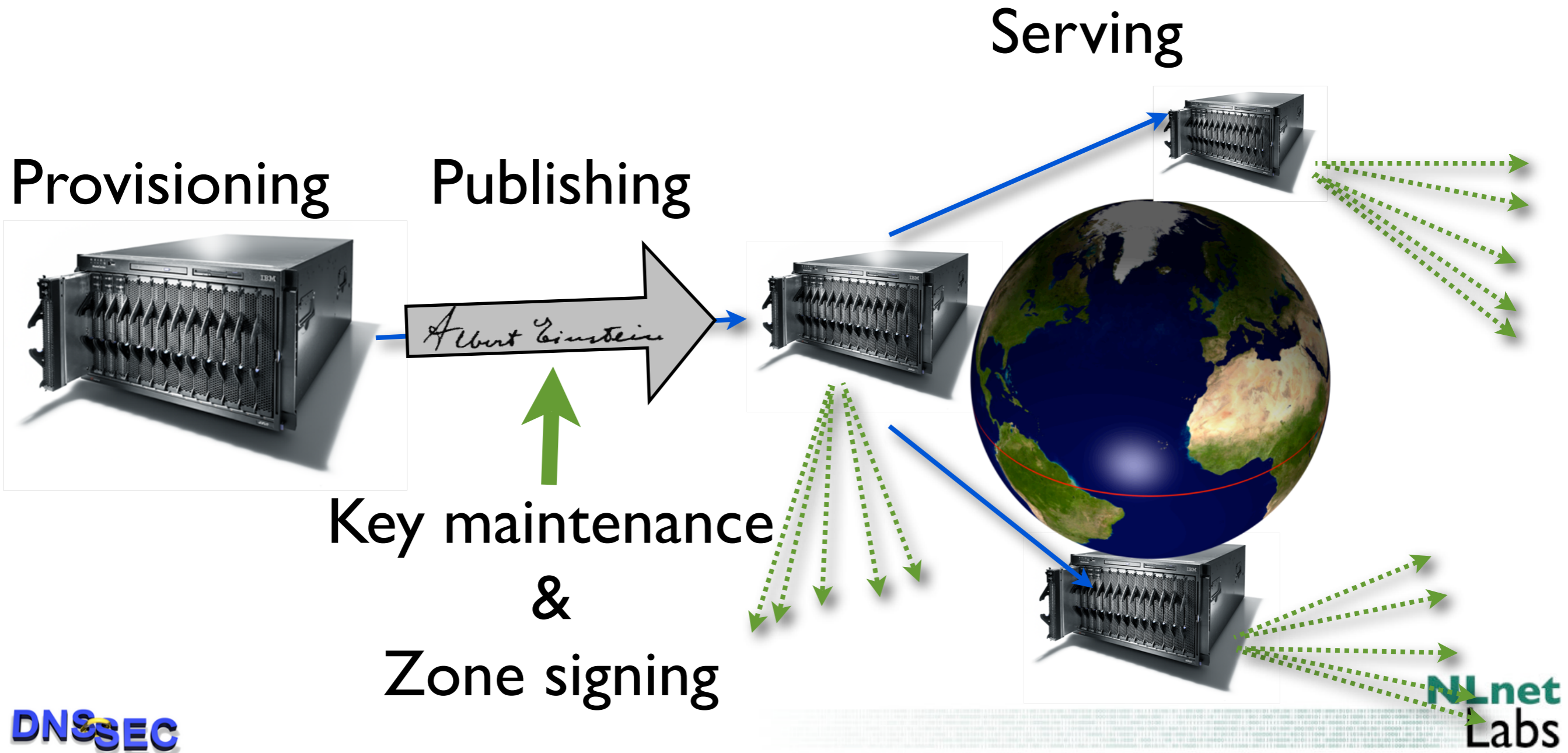
Publishing

Serving





# Introducing DNSSEC

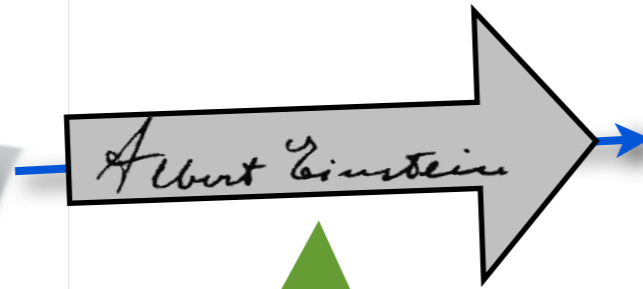


# Introducing DNSSEC

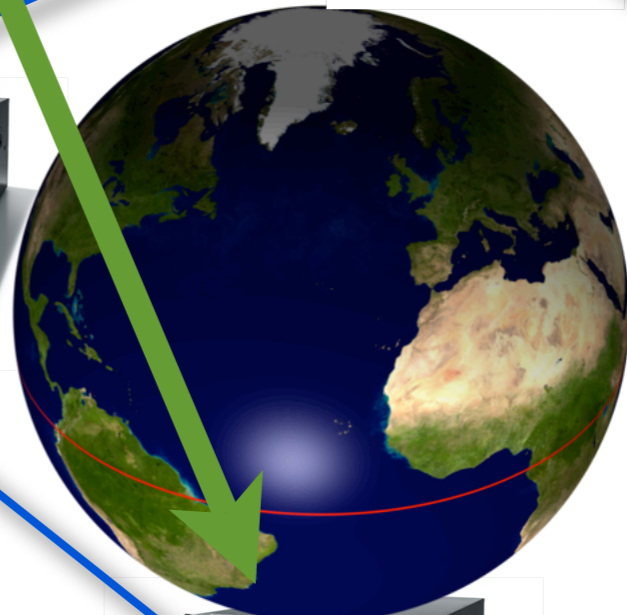
DNSSEC aware name server Software  
Serving

Provisioning

Publishing



Key maintenance  
&  
Zone signing



# DNSSEC Aware Nameservers

- Software Exercise
- BIND and Unbound free and open source
- Some hardware requirements:
  - memory requirements increase
  - RIPE 352 or measure

# Key maintenance

## Private Keys

- Determine a policy and implement it
- Think about risks and operations

Risk	On or offline	System consideration
high	on	HSM (FIPS Level 4)
high	off	Reviewed procedures, Physical Safe
medium	on	HSM (FIPS level 2) or shielded system
medium	off	Reviewed Procedures
low	on	Connected or Local system
low	off	System

# Key maintenance

## Public Keys

- Your users may configure your public key as a trust anchor
- Consider how your users will fetch the key: Out of band validation
- Document your procedures
- Upload to your parent

# Key Maintenance Rollovers

- Document rollover procedures
  - Take into account the timing sequences
  - Understand, train, and automate

# Signing

- Use the BIND/Unbound tools
- Depending on your requirements build or buy machinery that allows secure key storage
- Open source tools and proprietary solutions

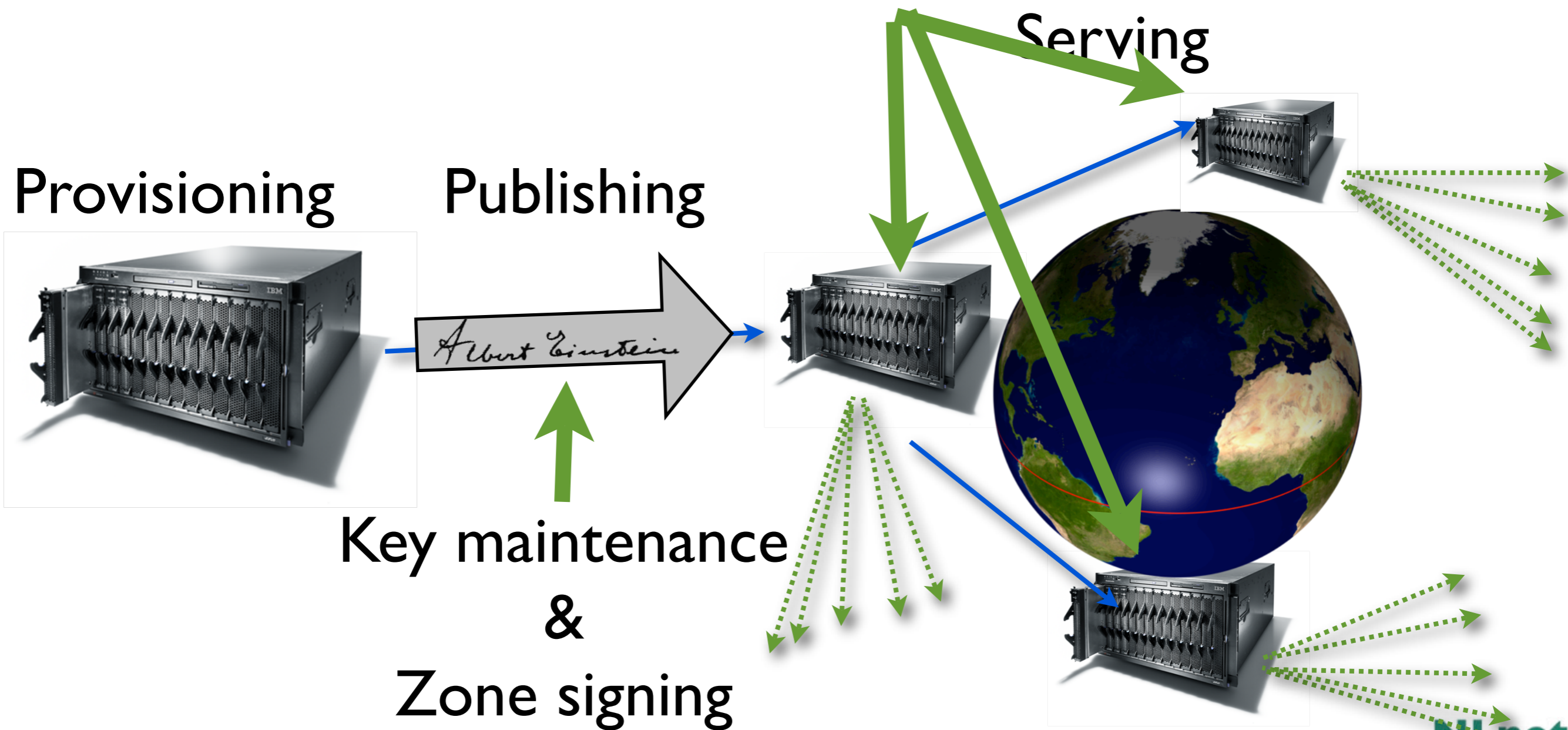
# Note about costs

- Again: Knowledge Exercise
- Understanding the issues about publication, maintenance and rolling of the keys
- Draw up requirements
- Implement or buy solutions
- On the server side: Simple upgrade of software

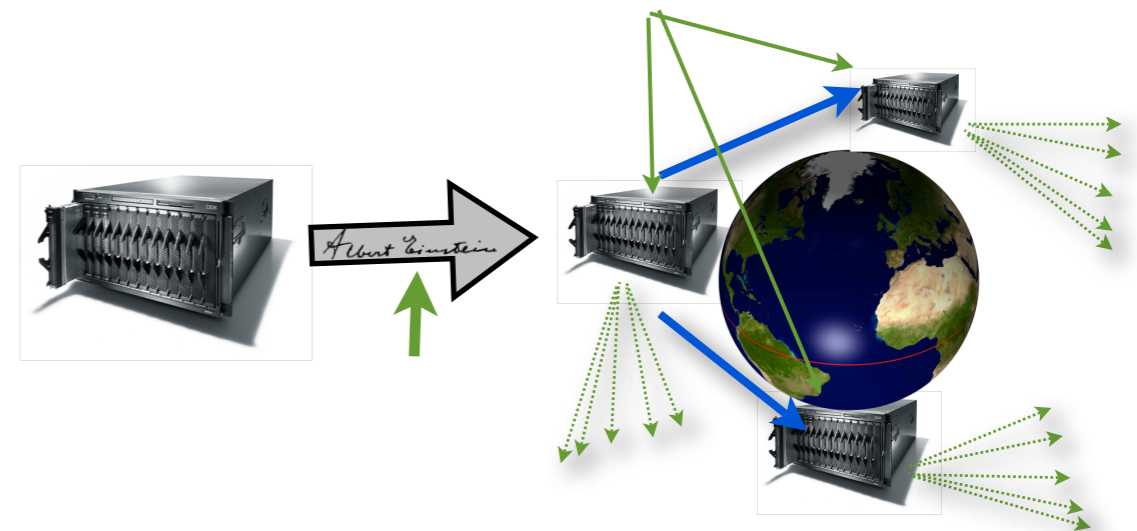


# For the Registry

DNSSEC aware name server Software



# For the Registry

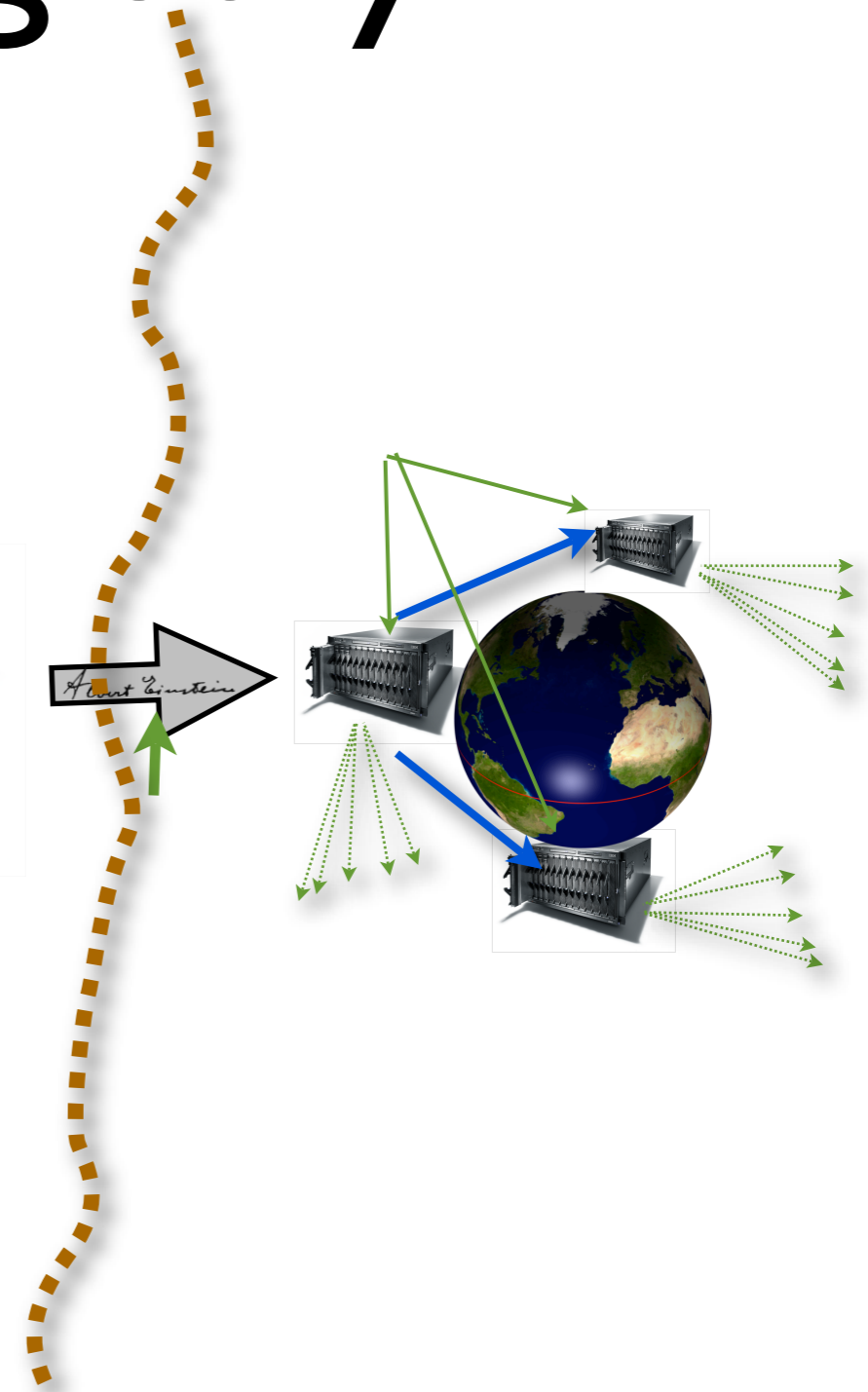
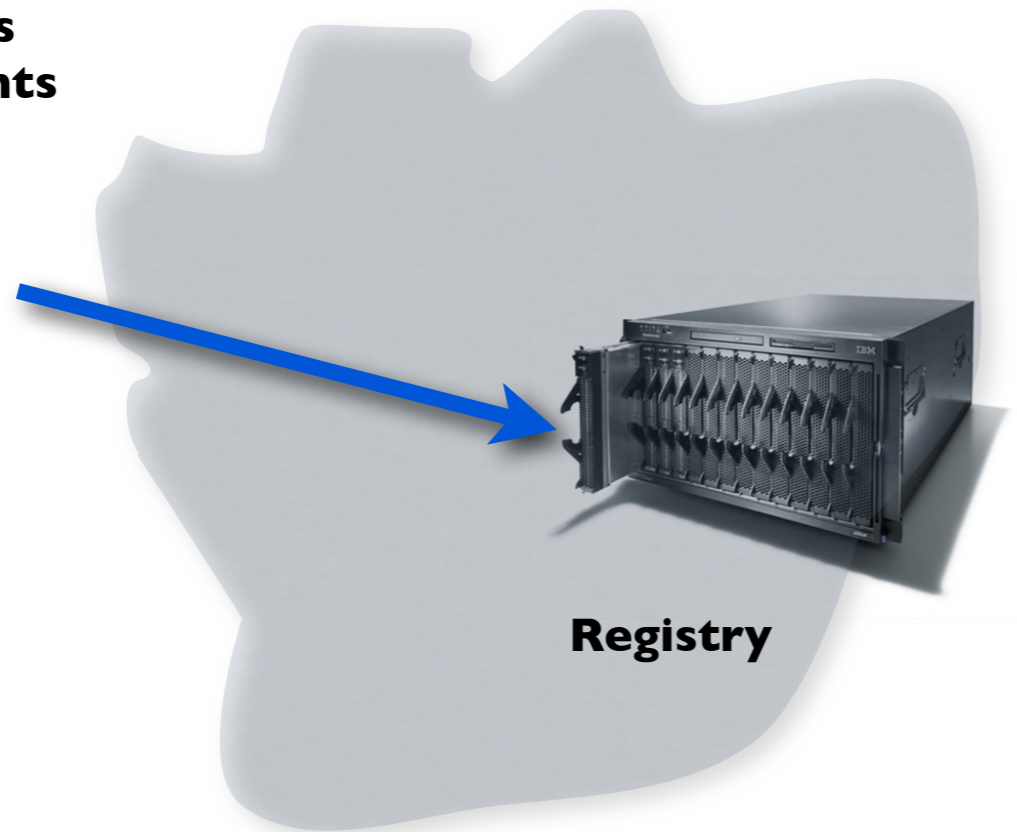


# For the Registry



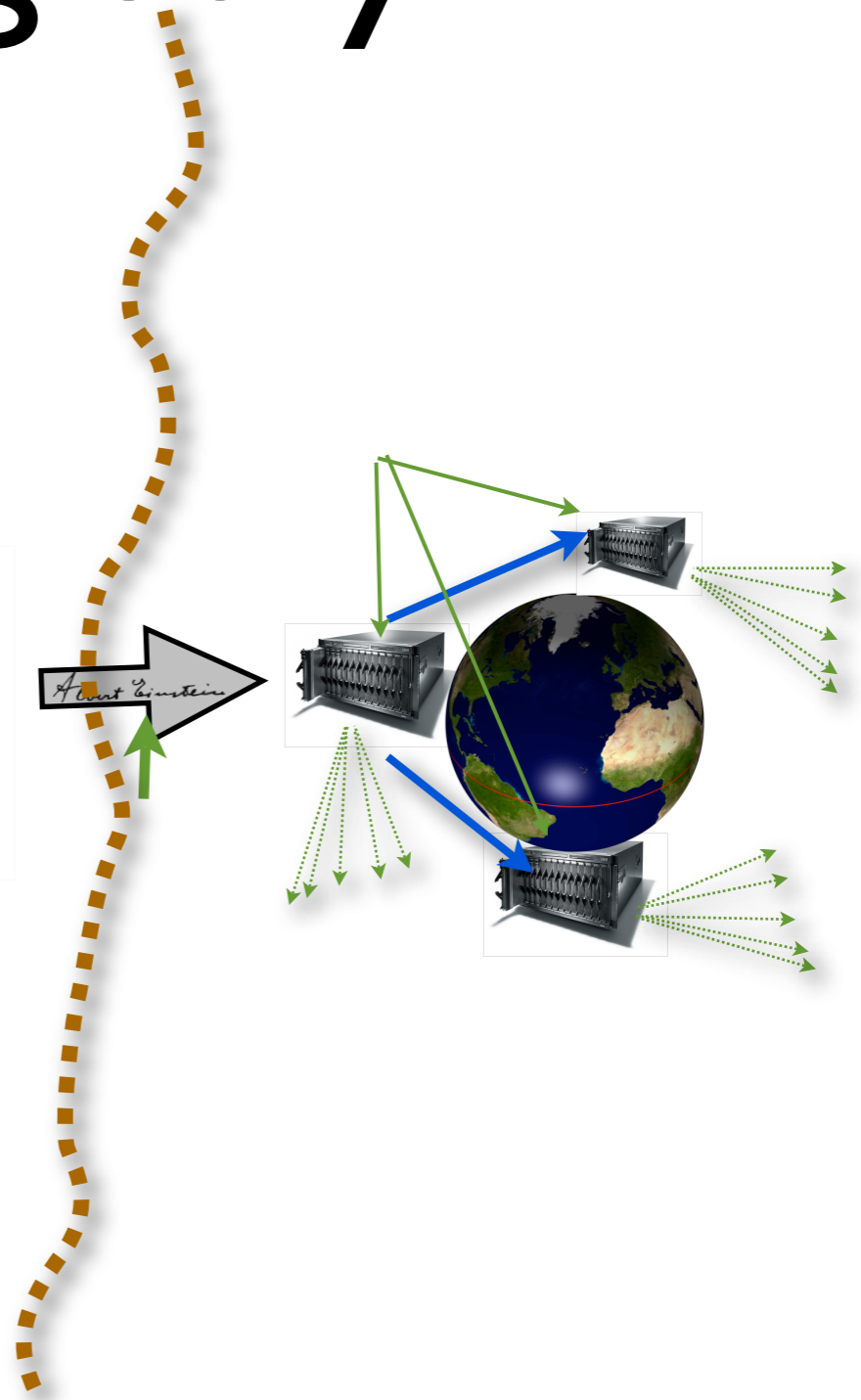
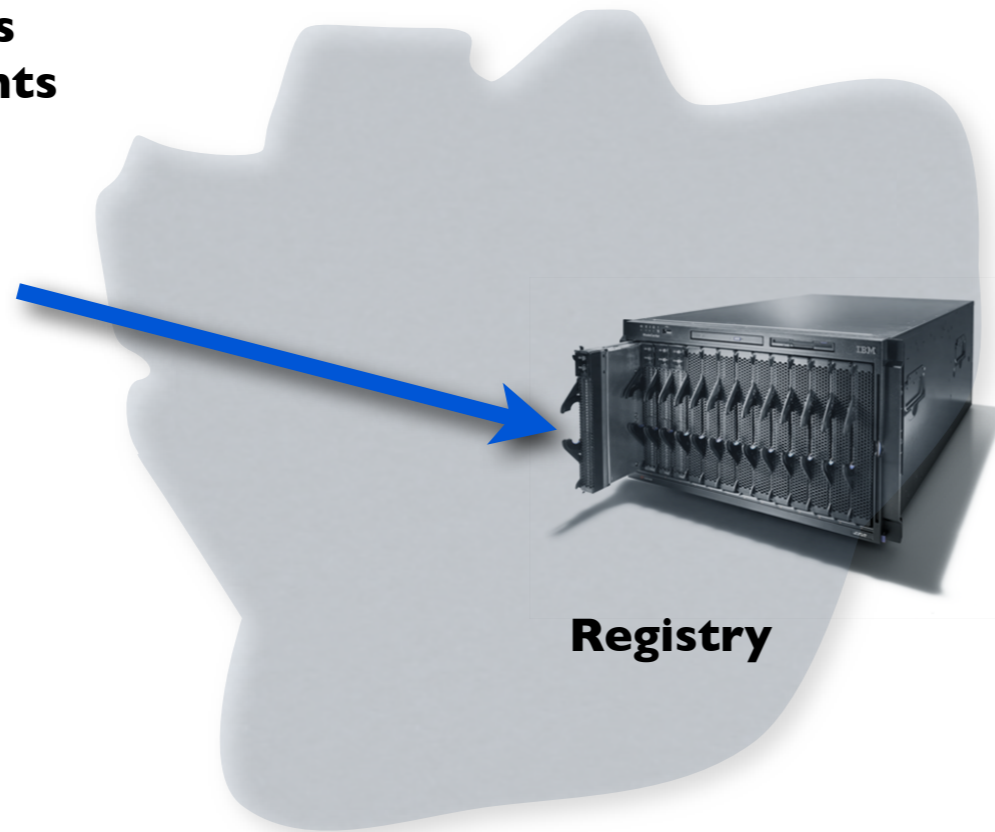
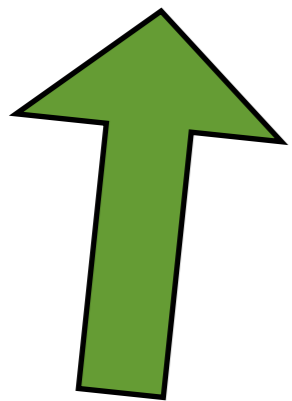
# For the Registry

**Registrars  
& Registrants**



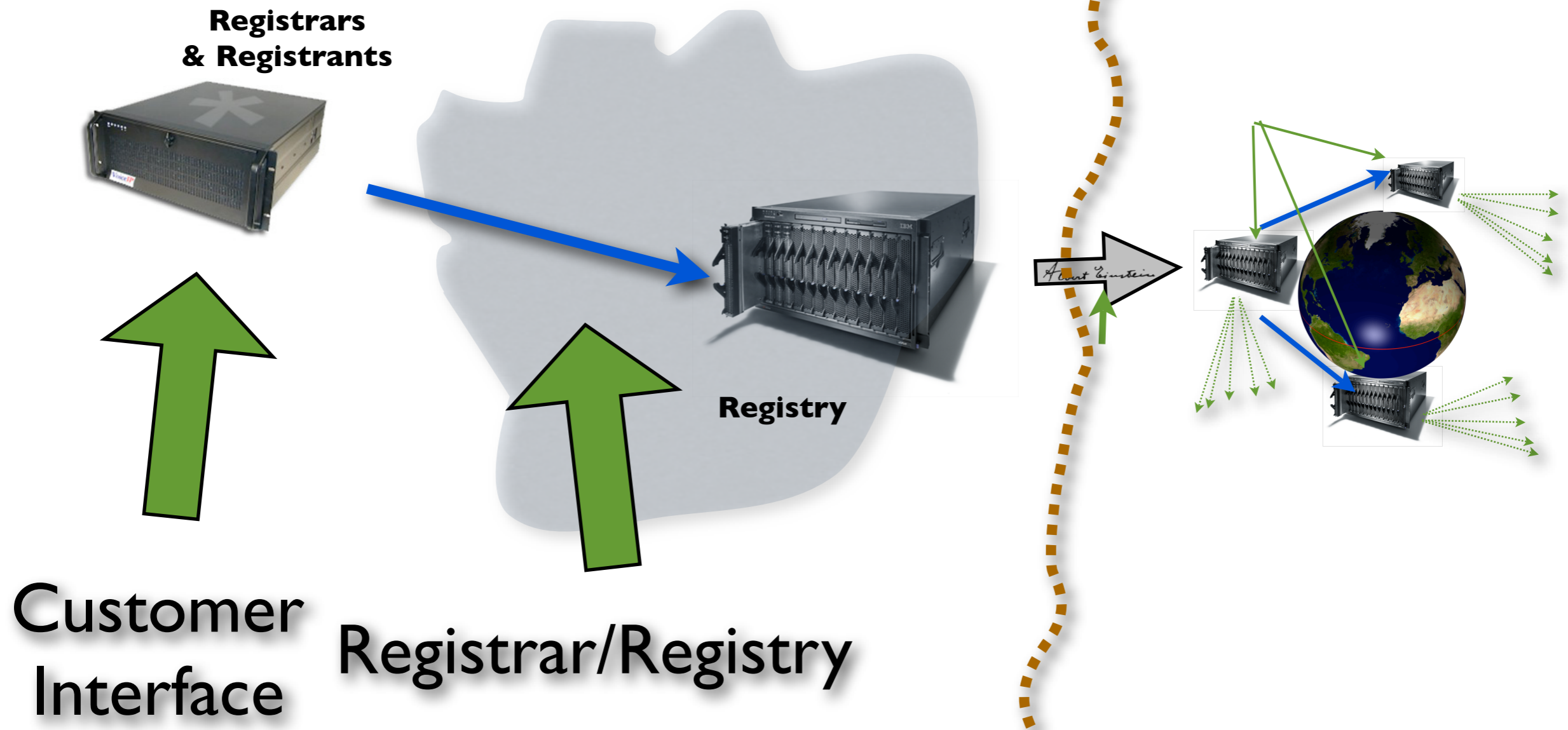
# For the Registry

Registrars  
& Registrants



Customer  
Interface

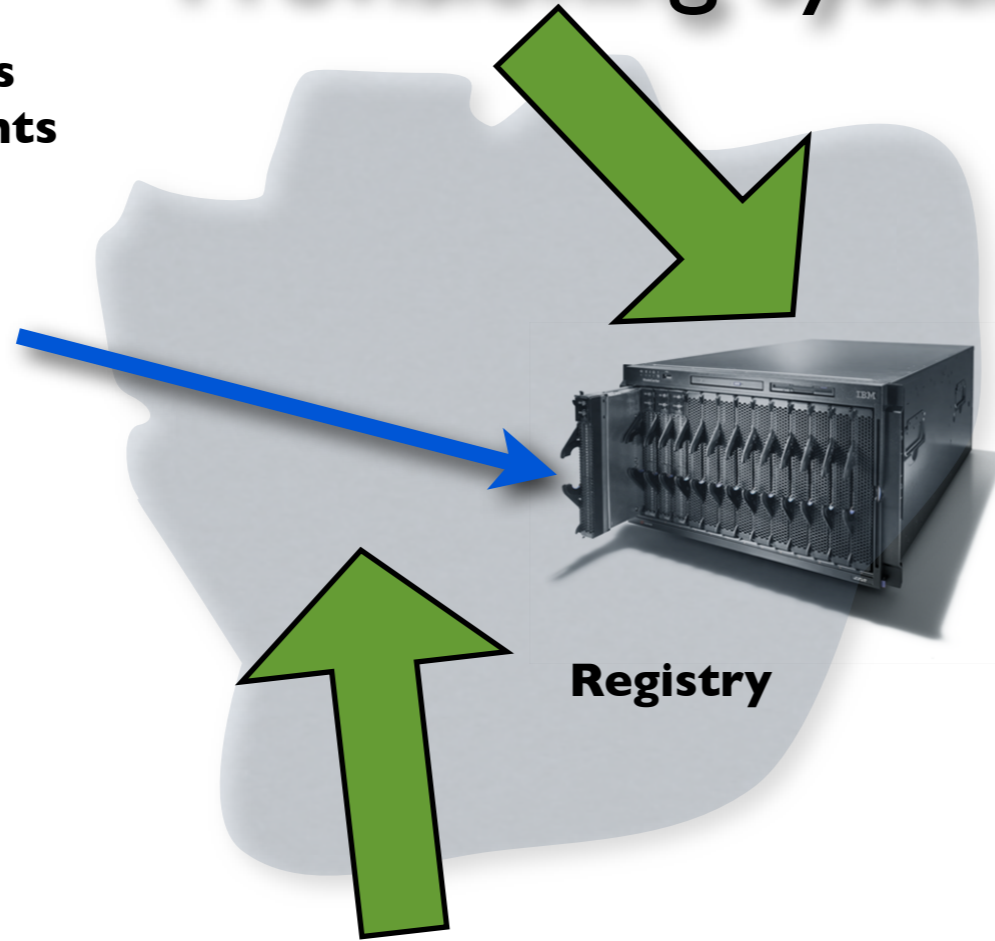
# For the Registry



# For the Registry

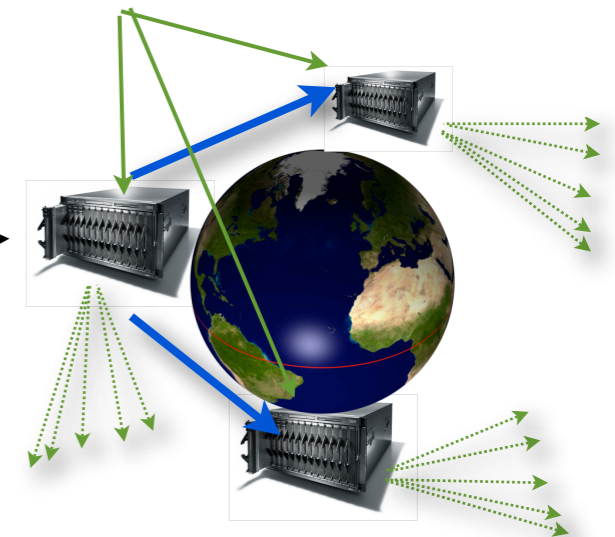
## Provisioning System

Registrars  
& Registrants



Registry

*Agent Einstein*



Customer  
Interface

Registrar/Registry

# Questions to address Registrar

- Customer interaction
  - How much checking of your customer setup (value add)
  - How do you validate the public key
    - Is this any different than how you validate a change in the NS?



# Questions to address Registry

- What will you store DNSKEY or DS
- Consider DS hash algorithm agility: Will you ask all your customers to provide new keys?
- How will you get the DNSKEYs from your Registrars?
- How is that different from how you get the NS records?

# Questions to address Registry II

- What are your operational constraints?
- Will you allow direct Registrant interaction
  - e.g. when a registrants key went broken at 2 am

# Follow the NS

- From a registration perspective the NS and the DS data have very similar properties

# State of DNSSEC Deployment

# The Numbers

(A sad state of affairs)

- <http://secspider.cs.ucla.edu/> reports a little over 10.000 zones signed, only little under 1000 are production zones
- Reverse zones in the RIPE region
- .se, .pr, .br and .bg are signed top level domains
- .uk, .arpa, .org, and a few enum trees have voiced some form of commitment
- There is a testbed for the root

# Chicken and Egg

- Little deployment means little experience and few tools.
- Little experience and few tools increase the cost of deployment
- Little signing infrastructure to justify cost of validation
- Little validators to justify the signing infrastructure
- No short term benefits, only long term

# Breaking the Egg

- Deployment by the custodians of the DNS infrastructure (TLDs and the Root) allows others to hook in
- Resolver side deployment to immediately benefit

Domain	Country	Registrations	The most frequently registered ccTLDs today are an reflective of the world's most populous countries — at least in part.
.de	Germany	11,120,000	China has the fastest growing ccTLD and is no pace to overtake Germany by 2018. Overall, the 10 most popular ccTLDs account for nearly 70% of all ccTLD registrations.
.cn	China	6,035,000	
.uk	United Kingdom	6,010,000	
.nl	Netherlands	2,545,000	
.it	Italy	1,428,000	
.us	United States	1,300,000	
.ar	Argentina	1,200,000	
.br	Brazil	1,139,000	
.ru	Russia	1,000,000	
.ch	Switzerland	1,000,000	
.au	Australia	953,142	Please note that not all ccTLDs are equally easy (and cheap) to register, which is one reason some countries have had stronger growth than others.
.jp	Japan	905,000	

Source: Country Registry, Data-Land Research

## Country Codes of the World

At the end of every URL, and most address is a top-level domain (TLD). Although rare in the world's most popular TLD, it is the most common. There are more than 200 TLDs in use around the world, most of which are country code top-level domains (ccTLDs).

ccTLDs are two-digit codes assigned to countries and territories.

Of the 193 out there, TLDs that have been registered, more than 21 million are ccTLDs.

Some ccTLDs are easy to remember, such as .us for Australia.

But many codes are not so obvious, such as .lk for Sri Lanka or .za for South Africa.

That's where this map fits in.

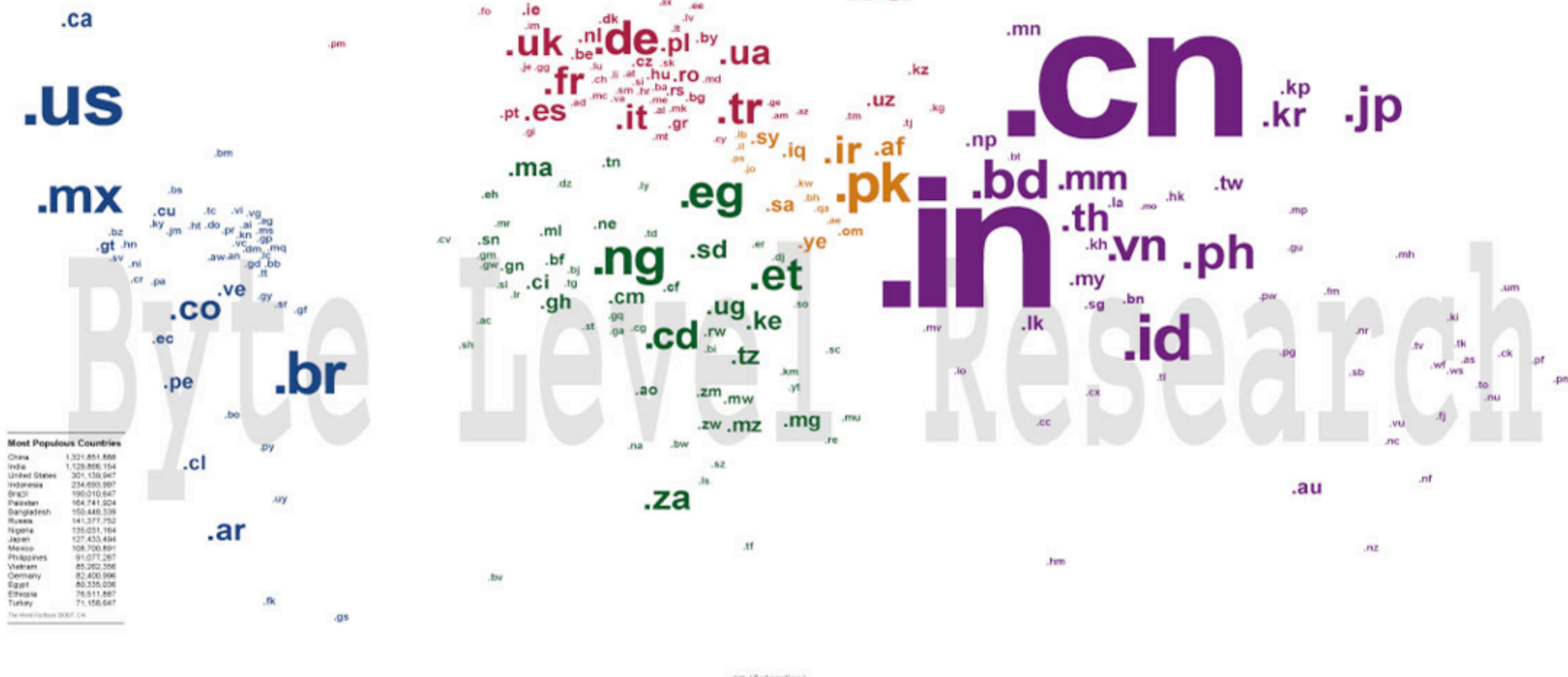
This map includes 197 ccTLDs, each aligned with the country or territory it represents.

Each ccTLD is sized relative to the population of the country or territory, with the exception of China and India, which were reduced by 20% to accommodate the legend. At the other end of the spectrum, the smallest type size used reflects their smaller

with 10 million or fewer residents.

Each geographic region is color-coded in the legend below for easy reference.

For more information about ccTLDs, visit [www.icann.org](http://www.icann.org).



Country	Population
China	1,321,851,888
India	1,129,866,154
United States	301,139,947
Indonesia	234,693,997
Brazil	190,010,647
Pakistan	164,741,004
Bangladesh	150,448,339
Russia	141,377,752
Nigeria	135,051,164
Japan	127,433,494
Mexico	108,750,891
Philippines	91,077,287
Vietnam	85,282,356
Germany	82,400,996
Egypt	80,335,036
Ethiopia	76,511,867
Turkey	71,156,647

The World Factbook 2007, CIA

**Americas** .ag Antigua and Barbuda | .ai Anguilla | .an Netherlands Antilles | .ar Argentina | .aw Aruba | .bb Barbados | .bm Bermuda | .bo Bolivia | .br Brazil | .bs Bahamas | .bz Belize | .ca Canada | .cl Chile | .co Colombia | .cr Costa Rica | .cu Cuba | .dm Dominica | .do Dominican Republic | .ec Ecuador | .fk Falkland Islands (Malvinas) | .gd Grenada | .gf French Guiana | .gp Guadeloupe | .gs South Georgia and the South Sandwich Islands | .gt Guatemala | .gy Guyana | .hm Heard and McDonald Islands | .hn Honduras | .hr Croatia | .hu Hungary | .id Indonesia | .ie Ireland | .il Israel | .im Isle of Man | .in India | .io British Indian Ocean Territory | .it Italy | .je Jersey | .kg Kyrgyzstan | .kz Kazakhstan | .lk Sri Lanka | .lb Lebanon | .li Liechtenstein | .lt Lithuania | .lu Luxembourg | .lv Latvia | .mc Monaco | .md Moldova | .me Montenegro | .mk Macedonia | .ml Malta | .nl Netherlands | .no Norway | .pl Poland | .pm Saint Pierre and Miquelon | .pt Portugal | .ro Romania | .rs Serbia | .ru Russian Federation | .se Sweden | .si Slovenia | .sj Svalbard and Jan Mayen Islands | .sk Slovak Republic | .sm San Marino | .tj Tajikistan | .tm Turkmenistan | .tr Turkey | .ua Ukraine | .uk United Kingdom | .uz Uzbekistan | .va Holy See (Vatican City State) **Africa** .ac Ascension Island | .ag Angola | .bf Burkina Faso | .bi Burundi | .bj Benin | .bv Bouvet Island | .bw Botswana | .cd Congo, The Democratic Republic of the | .cf Central African Republic | .cg Congo, Republic of | .ci Côte d'Ivoire | .cm Cameroon | .cv Cape Verde | .dj Djibouti | .dz Algeria | .eg Egypt | .eh Western Sahara | .er Eritrea | .et Ethiopia | .ga Gabon | .gh Ghana | .gm Gambia | .gn Guinea | .gq Equatorial Guinea | .gw Guinea-Bissau | .ke Kenya | .km Comoros | .lr Liberia | .ls Lesotho | .ly Libya | .ma Morocco | .mg Madagascar | .ml Mali | .mr Mauritania | .mu Mauritius | .mw Malawi | .mz Mozambique | .na Namibia | .ne Niger | .ng Nigeria | .re Reunion Island | .rw Rwanda | .sc Seychelles | .sd Sudan | .sh Saint Helena | .sl Sierra Leone | .sn Senegal | .so Somalia | .st Sao Tome and Principe | .sz Swaziland | .td Chad | .tf French Southern Territories | .tg Togo | .tn Tunisia | .tz Tanzania | .ug Uganda | .yt Mayotte | .za South Africa | .zm Zambia | .zw Zimbabwe **Middle East** .ae United Arab Emirates | .af Afghanistan | .bh Bahrain | .il Israel | .iq Iraq | .ir Iran, Islamic Republic of | .jo Jordan | .kw Kuwait | .lb Lebanon | .om Oman | .pk Pakistan | .ps Palestinian Territory | .qa Qatar | .sa Saudi Arabia | .sy Syrian Arab Republic | .ye Yemen **Asia-Pacific** .as American Samoa | .au Australia | .bd Bangladesh | .bn Brunei | .bt Bhutan | .cc Cocos (Keeling) Islands | .ck Cook Islands | .cn China | .cx Christmas Island | .fj Fiji | .fm Micronesia, Federated States of | .gu Guam | .hk Hong Kong | .hm Heard and McDonald Islands | .id Indonesia | .in India | .io British Indian Ocean Territory | .jp Japan | .kh Cambodia | .ki Kiribati | .kp Korea, Democratic People's Republic | .kr Korea, Republic of | .la Lao People's Democratic Republic | .lk Sri Lanka | .mh Marshall Islands | .mm Myanmar | .mn Mongolia | .mo Macao | .mp Northern Mariana Islands | .mv Maldives | .my Malaysia | .nc New Caledonia | .nf Norfolk Island | .np Nepal | .nr Nauru | .nu Niue | .nz New Zealand | .pf French Polynesia | .pg Papua New Guinea | .ph Philippines | .pn Pitcairn Island | .pw Palau | .sb Solomon Islands | .sg Singapore | .th Thailand | .tk Tokelau | .tl Timor-Leste | .to Tonga | .tv Tokelau | .um United States Minor Outlying Islands | .vn Vietnam | .vu Vanuatu | .wf Wallis and Futuna Islands | .ws Samoa | .aq (Antarctica)

Copyright and Integrity: All Rights Reserved. This page is a registered trademark of Byte Level Research LLC. © 2007 Byte Level Research LLC - [www.bytelevel.com](http://www.bytelevel.com)

[bytelevel.com/map/ccTLD.html](http://bytelevel.com/map/ccTLD.html)



© 2006-2008 NLnet Labs



# How .IN sets a global example

- Simultaneous action
- Registry and Registrar implementation
- Validation turned on by mayor ISPs and Enterprises
- Signing by key-stakeholders (banks)
- Sharing of experience and tools!

# How ccTLDs set a global example

- Simultaneous action
- Registry and Registrar implementation
- Validation turned on by mayor ISPs and Enterprises
- Signing by key-stakeholders (banks)
- Sharing of experience and tools!

