# System Administration & Security

COMP 175 | Fall 2021 | University of the Pacific | Jeff Shafer

## Lab 6 Discussion
## HTTPS, DNS

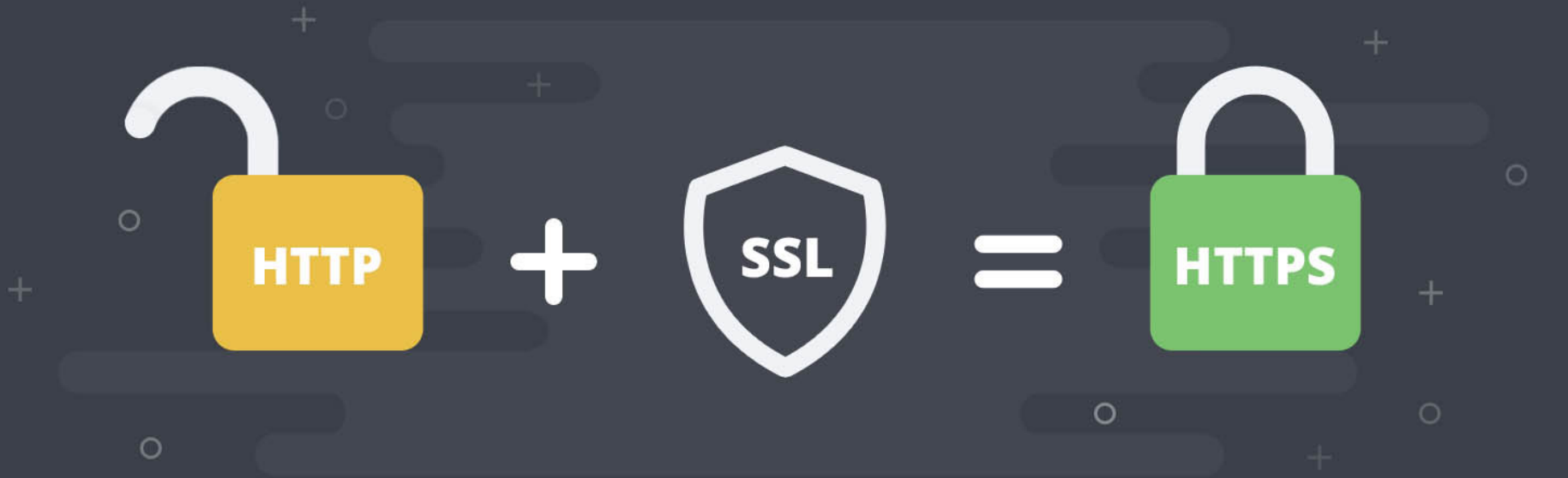
# Lab 6 – Web Server (Part 3)

## Objectives

- Assign DNS name to load balancer
- Obtain HTTPS certificate for encrypted access

## Discussion

- HTTPS
  - Certificates
  - Let's Encrypt
  - CertBot
- DNS




# HTTPS

# HOW HTTPS WORKS


**How HTTPS works ...in a comic! 🌈🎉🍕**

https://howhttps.works/



https://howhttps.works/

# HTTPS Motivations

- Confidentiality
  - No one can eavesdrop on your communication
- Integrity
  - No one can tamper with your communication (without being detected)
- Identification
  - No one can pretend to be the site you think you are visiting
- **All over an untrusted network**. *How?*

# Encryption!

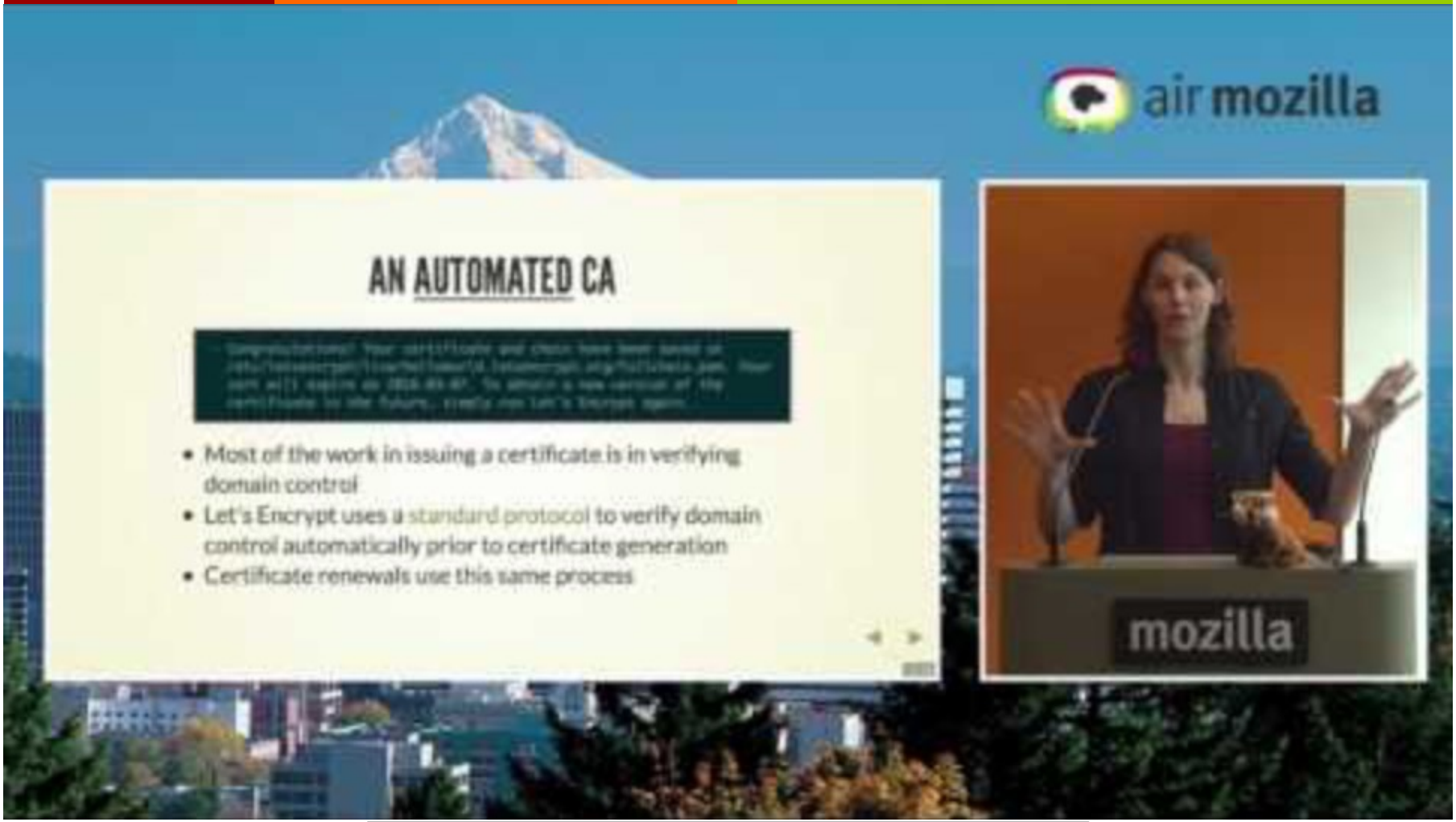
***Asymmetric encryption with public & private keys***

# Certificate Authority

- Third party organization that
  - Issues certificates
  - Confirms identity of certificate owner
  - Provides proof that certificate is valid
- **Let's Encrypt** is a certificate authority run by the Internet Security Research Group
  - Issues certificates for **free** (instead of for $$)
  - Goal: **Encrypt entire web** (even your blog!)

*From December 2017, but excellent motivations for existence of Let's Encrypt*

# Let's Encrypt

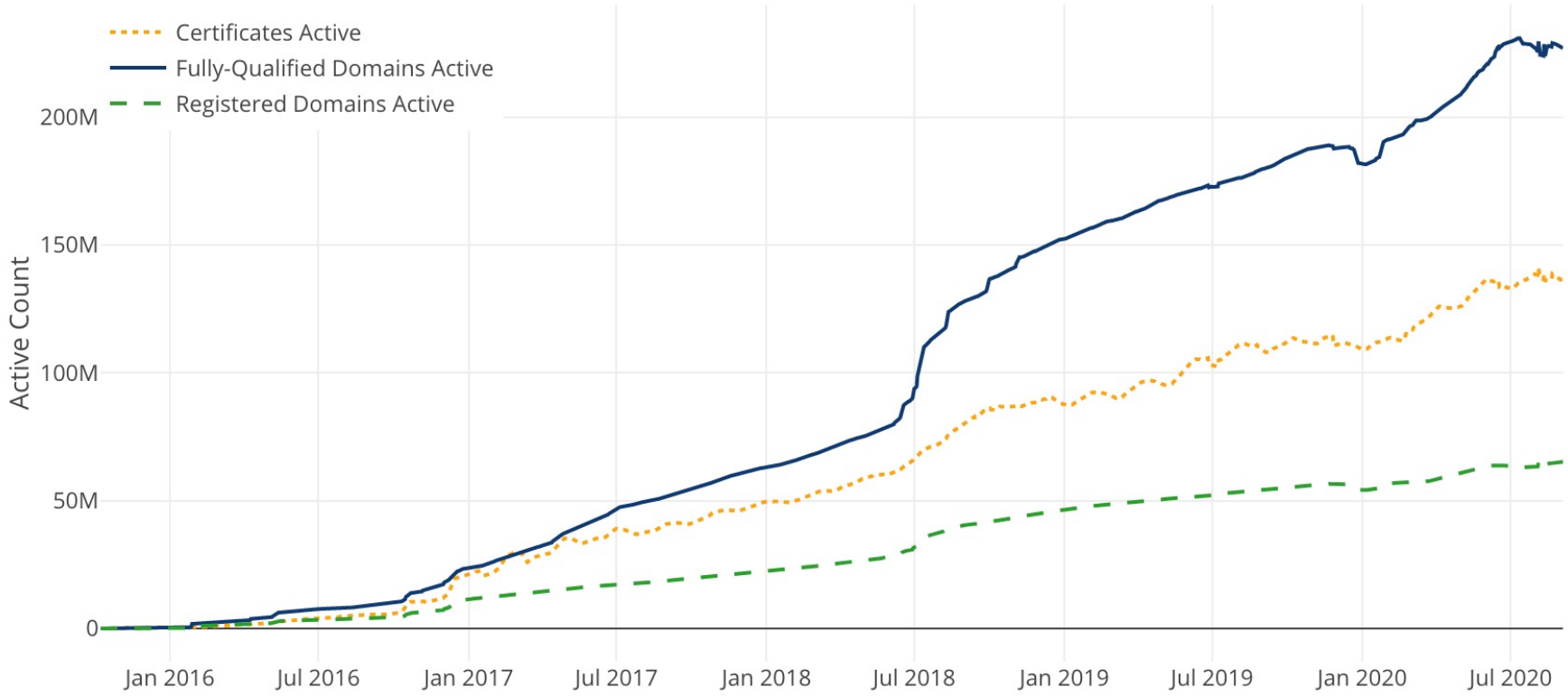





https://www.youtube.com/watch?v=W_OBpJmrKOc

# Growth of Let's Encrypt

## Let's Encrypt Growth




https://letsencrypt.org/stats/

# ACME

- Unlike other certificate authorities (that may or may not validate your *identity*), Let's Encrypt doesn't care about you as a person
  - All it requires is that you have **control over the domain**
- Automatic Certificate Management Environment (ACME) protocol
  - Rather than *humans* renewing certificate manually (every year?), have *machines* renew certificates automatically every 2-3 months
  - https://tools.ietf.org/html/rfc8555

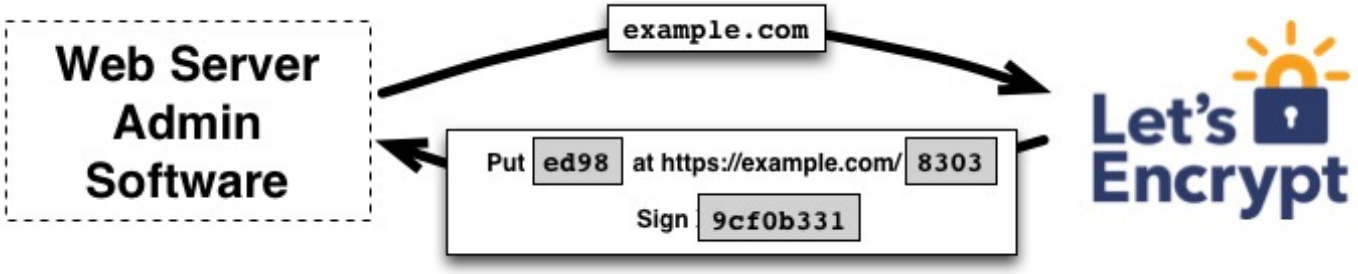
# CertBot

- Open-source tool created by Electronic Frontier Foundation
  - Requires an HTTP website already online with port 80 open
    - *Other methods, like DNS validation, also exist*
  - Speaks ACME to automate certificate renewal every 60 days
  - Understands Apache and NGINX configuration files
    - Can update config to automatically forward HTTP visitors to HTTP**S** version of your site


https://certbot.eff.org/

# Domain Validation

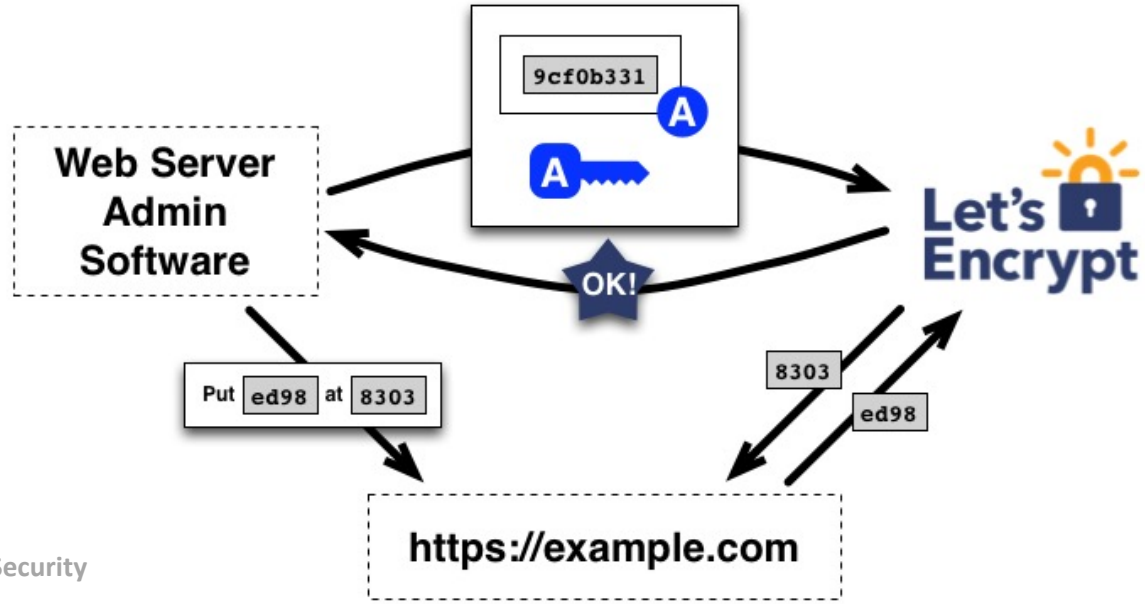
- Agent (CertBot) – *Please create certificate for me!*
  - Here is requested domain (example.com) and my public key
- Challenge from CA – *How can I trust you?*
  - Provision a HTTP resource under a well-known URI on http://example.com/
  - Sign a *nonce* (unique number) with your private key




https://letsencrypt.org/how-it-works/

# Domain Validation

- Agent (CertBot) – Signs and returns nonce + makes file available on website
- CA – Verifies results, issues certificate, sends authorized key pair to agent




**DNS**
Domain Name System

# DNS

# Motivation

- IP addresses are hard to remember
  - `138.9.110.12`? Or was it `.21`?
- Human-friendly names are much better
  - `engineering.pacific.edu`
- How can we translate between the two?

# Domain Name System (DNS)

- **Distributed database** implemented in hierarchy of many **name servers**
- **Application-layer protocol**
  - Hosts, routers, and name servers communicate to resolve names (address/name translation)
  - Core Internet function, implemented as application-layer protocol
  - Complexity at network's "edge"

# DNS is Decentralized

- No single point of failure
- No distant centralized database
- Easier maintenance
  - Take one or a dozen servers offline without issue
- Support high traffic volume
- *** **Scalability** ***

# What's in a Name?

- `engineering.pacific.edu`
  - .edu is top-level domain
  - "pacific" belongs to .edu
  - "engineering" belongs to "pacific"
  - Hierarchical!  Read from right to left
- Limits?
  - Up to 127 levels of hierarchy
  - Each label can have up to 63 characters
  - Full domain name cannot exceed 253 characters

# DNS: Services

- Hostname to IP address translation
  - *"www.pacific.edu" is 138.9.110.12*
- Hostname aliasing
  - Canonical, alias names
- Hostname load distribution
  - Replicated servers – Multiple IP addresses available for one name
  - *"google.com" is 74.125.239.128 or 74.125.239.135 or … or …. or … or ….*

# DNS: Services

- Mail server aliasing
  - What are the **multiple** host names that receive mail for this domain?
  - 1st priority, then 2nd backup, then 3rd backup, etc…
  - Allows you to use 3rd party email services (e.g. Google Apps)
  - *Mail to "pacific.edu" is directed to "*pacific-edu.mail.protection.outlook.com*" (SPAM filtering)*
- Other / Misc
  - SPF entries for email (Anti-spam)
  - DNSSEC (security/encryption)
  - Many other attributes…

# DNS: Record Types (Distributed Database)

Resource Record (RR) format: **`(name, value, type, ttl)`**

- Type=**A**
  - *name* is **hostname**
  - *value* is **IP address**
- Type=**NS**
  - *name* is domain (e.g. foo.com)
  - *value* is **hostname** of **authoritative name server** for this domain
- Type=**CNAME**
  - *name* is alias name for some "canonical" (real) name
  - value is canonical name
- Type=**MX**
  - *value* is name of **mailserver** associated with name
- Type=**TXT**
  - *value* is machine readable text (arbitrary)

# DNS: Example

```
$ dig pacific.edu any

; <<>> DiG 9.8.3-P1 <<>> pacific.edu any
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 5270
;; flags: qr rd ra; QUERY: 1, ANSWER: 9, AUTHORITY: 0, ADDITIONAL: 0

;; QUESTION SECTION:
;pacific.edu.                   IN   ANY

;; ANSWER SECTION:
pacific.edu. 59 IN A 52.38.242.166
pacific.edu. 59 IN A 34.210.252.224
pacific.edu. 899 IN NS ns-110.awsdns-13.com.
pacific.edu. 899 IN NS ns-1289.awsdns-33.org.
pacific.edu. 899 IN NS ns-2044.awsdns-63.co.uk.
pacific.edu. 899 IN NS ns-705.awsdns-24.net.
pacific.edu. 899 IN SOA ns-110.awsdns-13.com. awsdns-
hostmaster.amazon.com. 1 7200 900 1209600 86400
pacific.edu. 59 IN MX 0 pacific-edu.mail.protection.outlook.com.
pacific.edu. 59 IN TXT "status-page-domain-verification=tnw7vhhyh60c"
pacific.edu. 59 IN TXT "v=spf1 ip4:138.9.110.0/25 ip4:208.117.48.237
ip4:176.31.145.254 include:spf.protection.outlook.com
include:_spf.qualtrics.com include:spf.mandrillapp.com include:stspg-
```

Resource Record Type

Resource Record Value

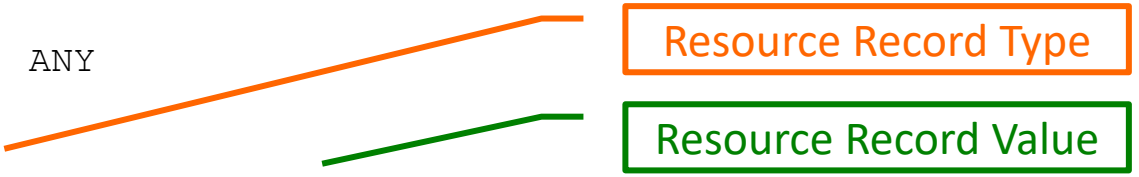
# DNS: Example

```
$ dig www.pacific.edu any

; <<>> DiG 9.8.3-P1 <<>> pacific.edu any
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 5270
;; flags: qr rd ra; QUERY: 1, ANSWER: 9, AUTHORITY: 0, ADDITIONAL: 0

;; QUESTION SECTION:
;pacific.edu.                   IN   ANY

;; ANSWER SECTION:
www.pacific.edu. 59 IN A 23.185.0.4
www.pacific.edu. 59 IN AAAA 2620:12a:8000::4
www.pacific.edu. 59 IN AAAA 2620:12a:8001::4
www.pacific.edu. 60 IN TXT "google-site-
verification=t3PZMb1DhGWjZb0EUyfhnd_zoAMN7yOkDMXyMxSHAh4"
```



# Hosted zones Info

## How hosted zones work

A hosted zone contains records that define how internet traffic is routed for a domain and its subdomains. For example, in the example.com hosted zone, you can create records for example.com and www.example.com that route traffic to a web server running on an EC2 instance or to an S3 bucket.

## Hosted zones (1)

View details | Edit | Delete | Create hosted zone

Filter hosted zones by property or value

1

| Domain name | Type | Created by | Record count | Description | Hosted zone ID |
|---|---|---|---|---|---|
| tigerenterprises.org | Public | Route 53 | 4 | HostedZone created by Route53 Registrar | Z05671742MAUBSH0KZXLY |

# tigerenterprises.org Info

Delete | Test record | Configure query logging

## Hosted zone details

Edit

Records (4) | Hosted zone tags (0)

### Records (4) Info

The following table lists the existing records in tigerenterprises.org. You can't delete the SOA record or the NS record named tigerenterprises.org.

Edit | Delete | Import zone file | Create record

Filter records by property or value

1

| Record name | Type | Routing policy | Differentiator | Alias | Value/Route traffic to |
|---|---|---|---|---|---|
| tigerenterprises.org | NS | Simple | - | No | ns-27.awsdns-03.com.<br>ns-538.awsdns-03.net.<br>ns-1211.awsdns-23.org.<br>ns-1729.awsdns-24.co.uk. |
| tigerenterprises.org | SOA | Simple | - | No | ns-27.awsdns-03.com. awsdns-hostmaster.amazon.com. 1 7200 900 1209600 86400 |
| [redacted].tigerenterprises.org | NS | Simple | - | No | ns-1804.awsdns-33.co.uk.<br>ns-103.awsdns-12.com.<br>ns-666.awsdns-19.net.<br>ns-1080.awsdns-07.org. |
| [redacted]tigerenterprises.org | NS | Simple | - | No | ns-651.awsdns-17.net.<br>ns-1534.awsdns-63.org.<br>ns-1718.awsdns-22.co.uk.<br>ns-88.awsdns-11.com. |

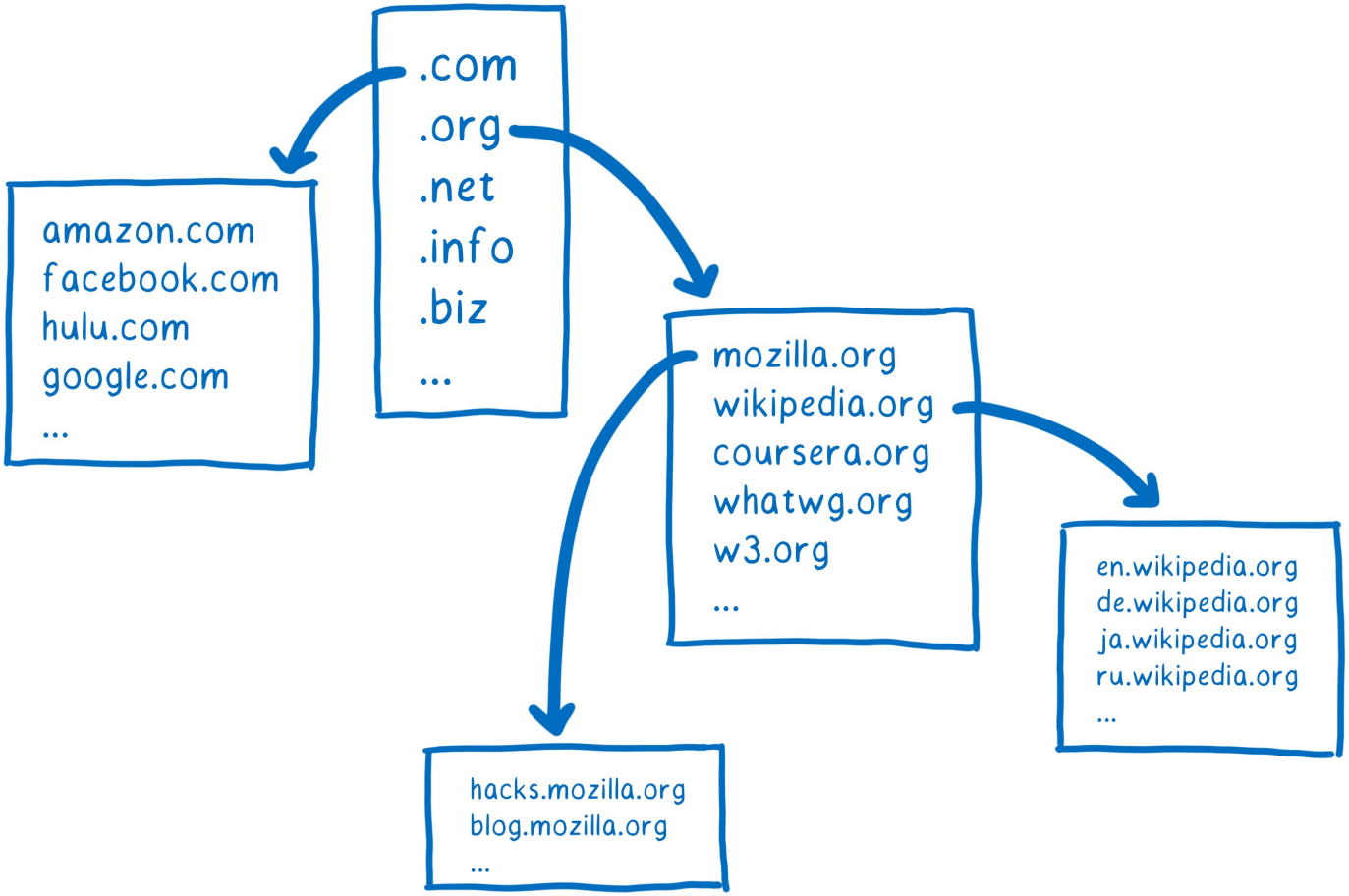
NS records for STUDENT-1, STUDENT-2, ...

Nameservers in **student** Route53 zones

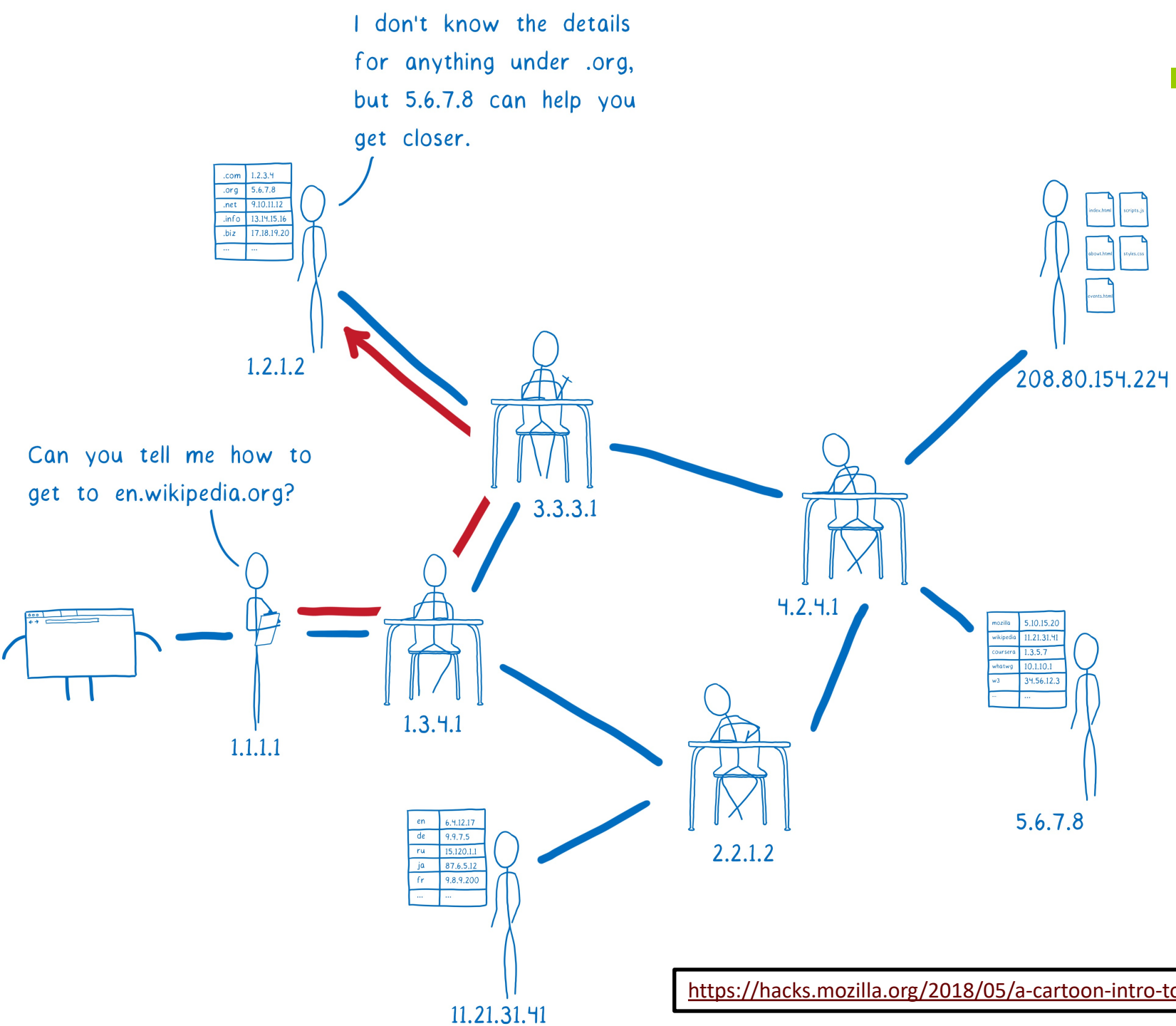
# DNS: Name Resolution

- Two types
- **Recursive**
  - The server you contact provides the final answer
  - *Behind the scenes, it may make several consecutive requests*
- **Iterative**
  - The server you contact directs you to a different server to get (closer to) the final answer

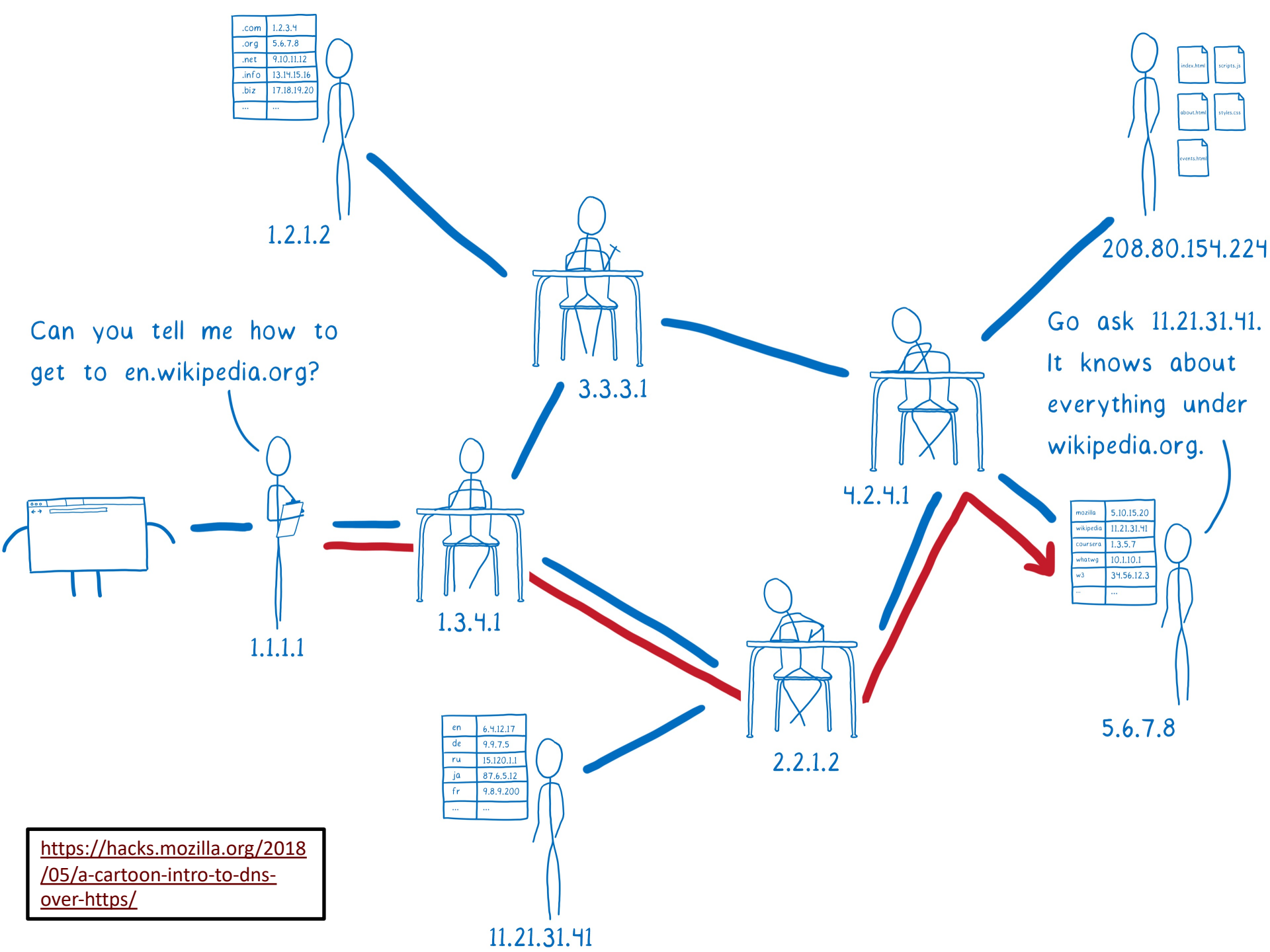
# en.wikipedia.org = 208.80.154.224



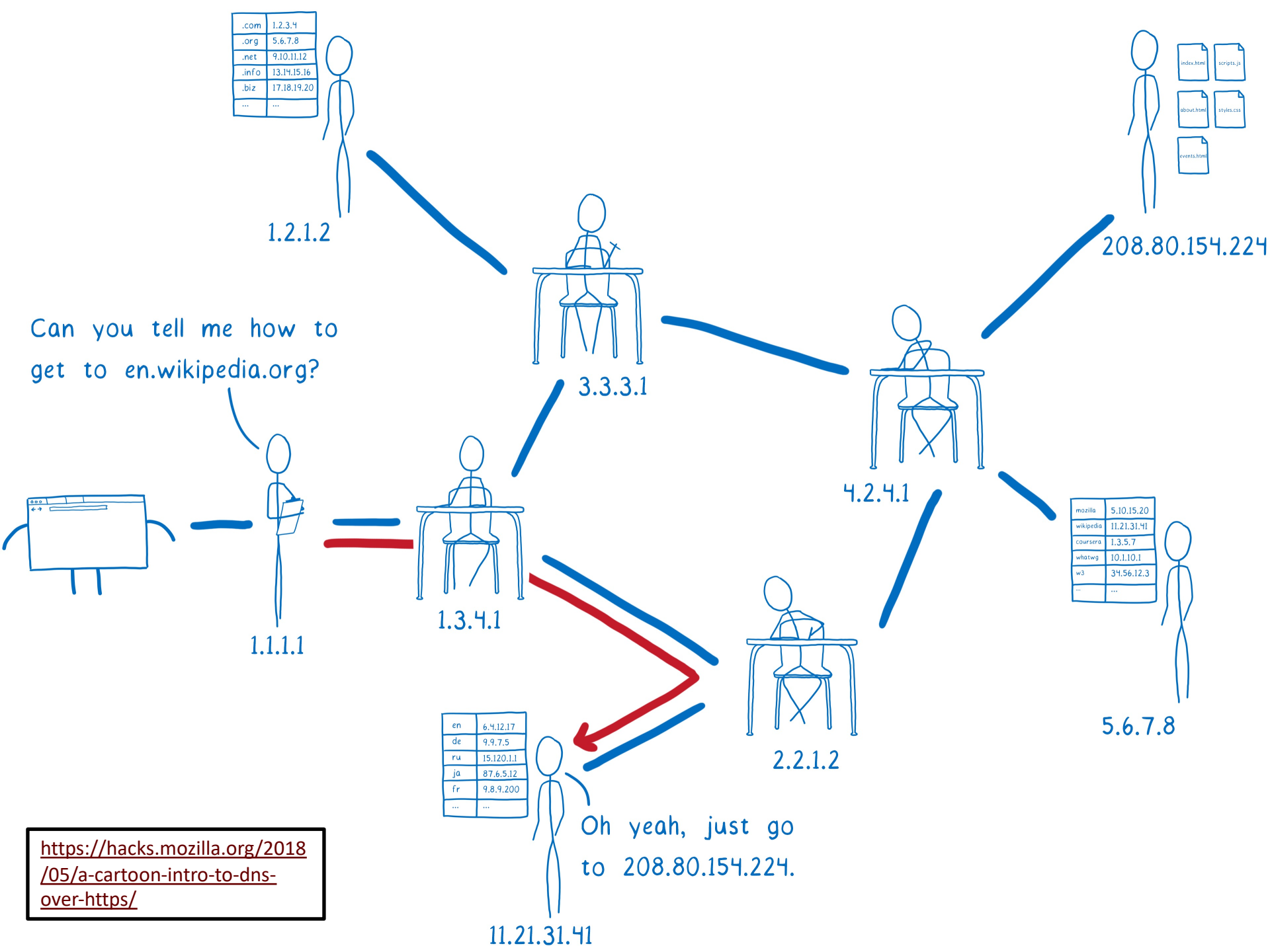

https://hacks.mozilla.org/2018/05/a-cartoon-intro-to-dns-over-https/




https://hacks.mozilla.org/2018/05/a-cartoon-intro-to-dns-over-https/




https://hacks.mozilla.org/2018/05/a-cartoon-intro-to-dns-over-https/




https://hacks.mozilla.org/2018/05/a-cartoon-intro-to-dns-over-https/

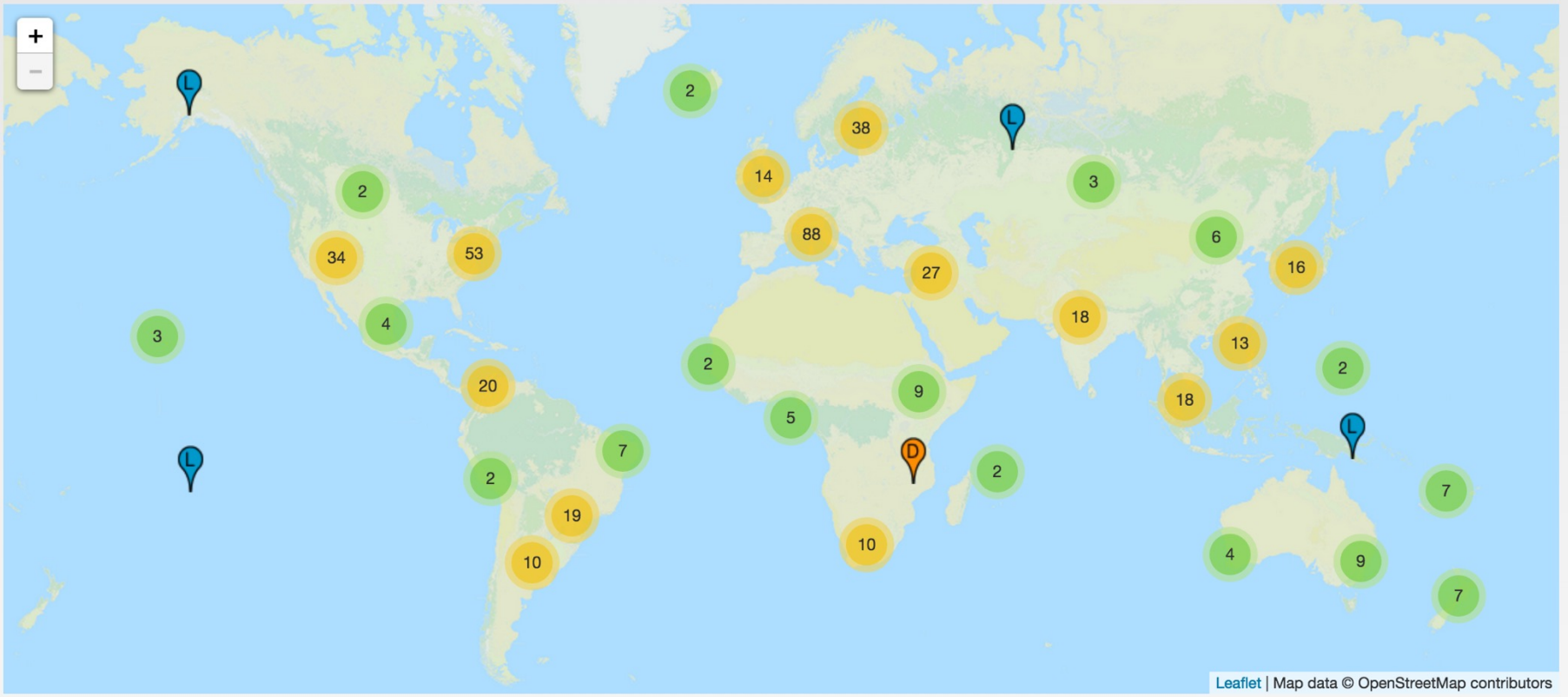
# DNS: Root Name Servers

- Contacted by local name server that can not resolve top-level domain
- Root name server:
  - Contacts authoritative name server for TLD if name mapping not known
  - Gets mapping
  - Returns mapping to local name server

**13 root name "servers" worldwide labeled `a` - `m`**

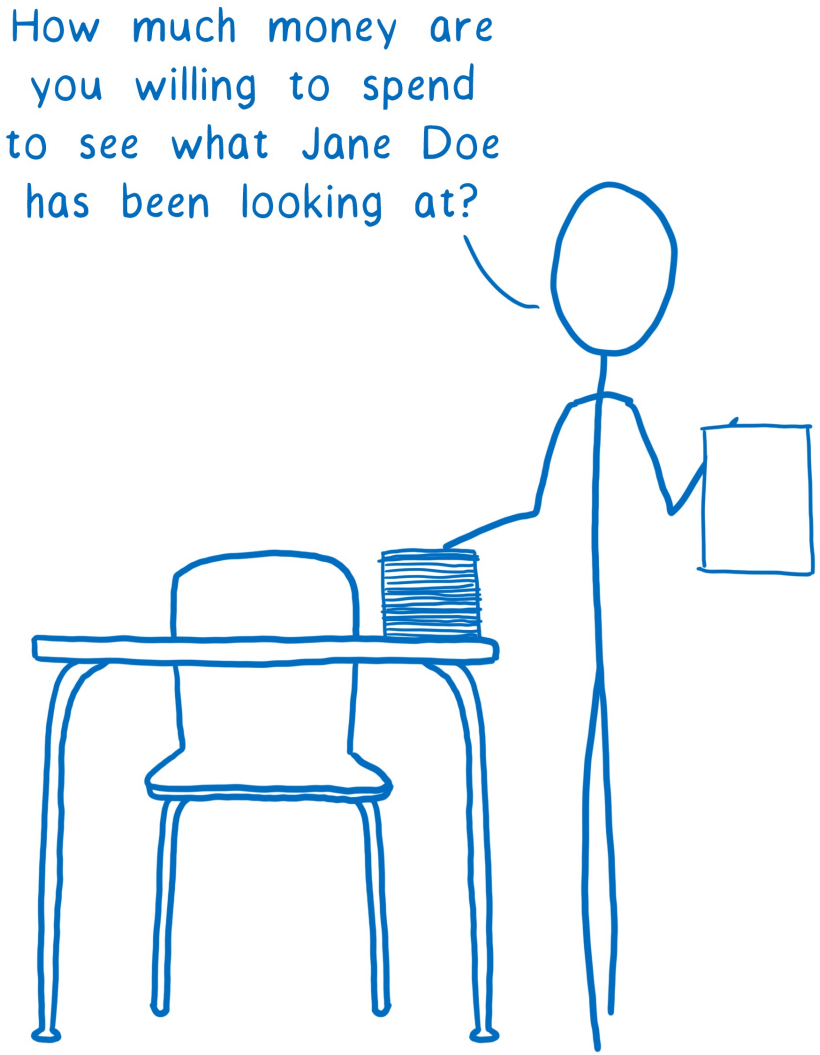
- Each "server" is really a cluster
- Some clusters are geographically distributed
- 1094 total in Spring 2020

# DNS: Root Name Servers



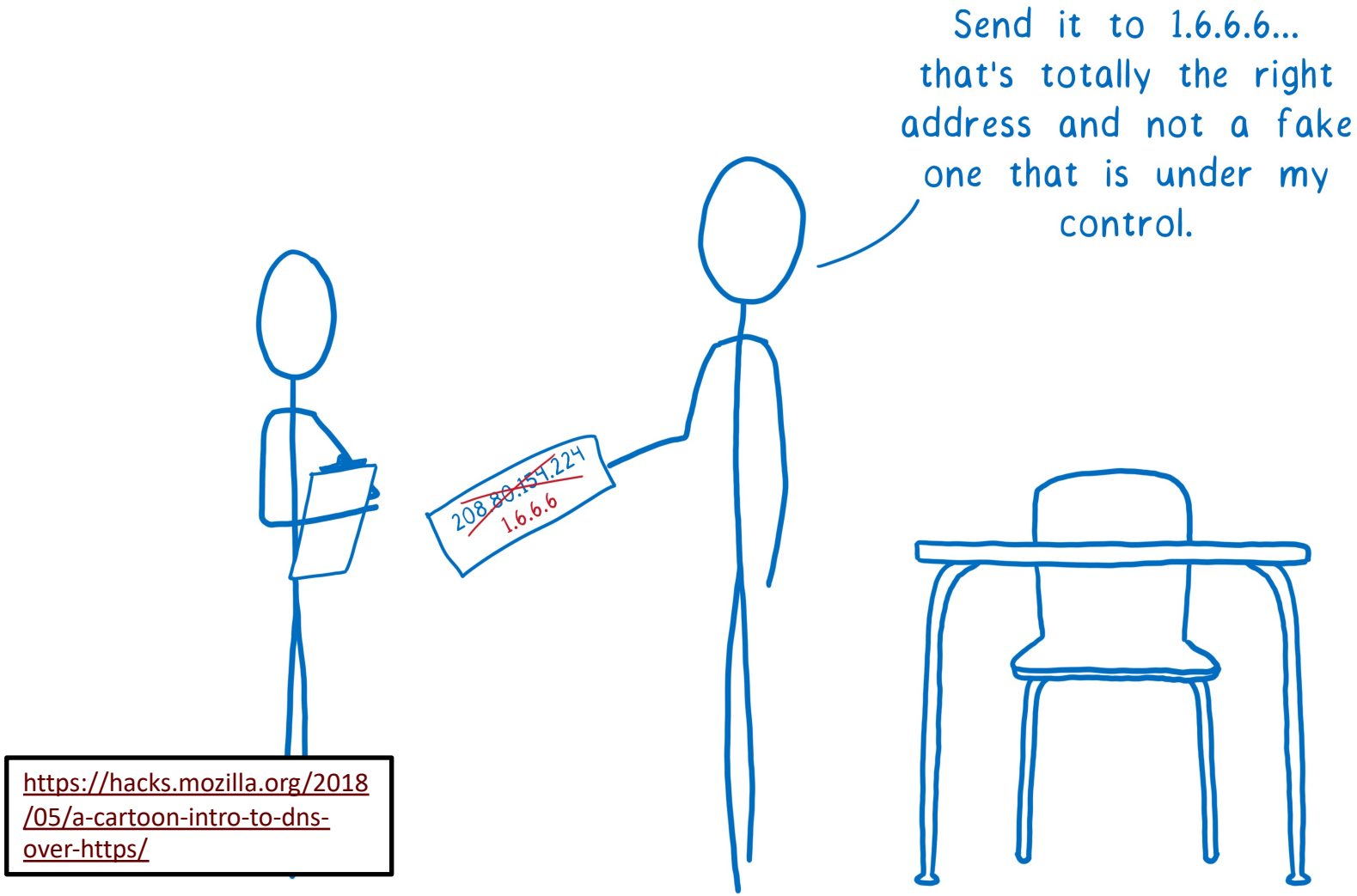

http://www.root-servers.org/

# DNS and Security



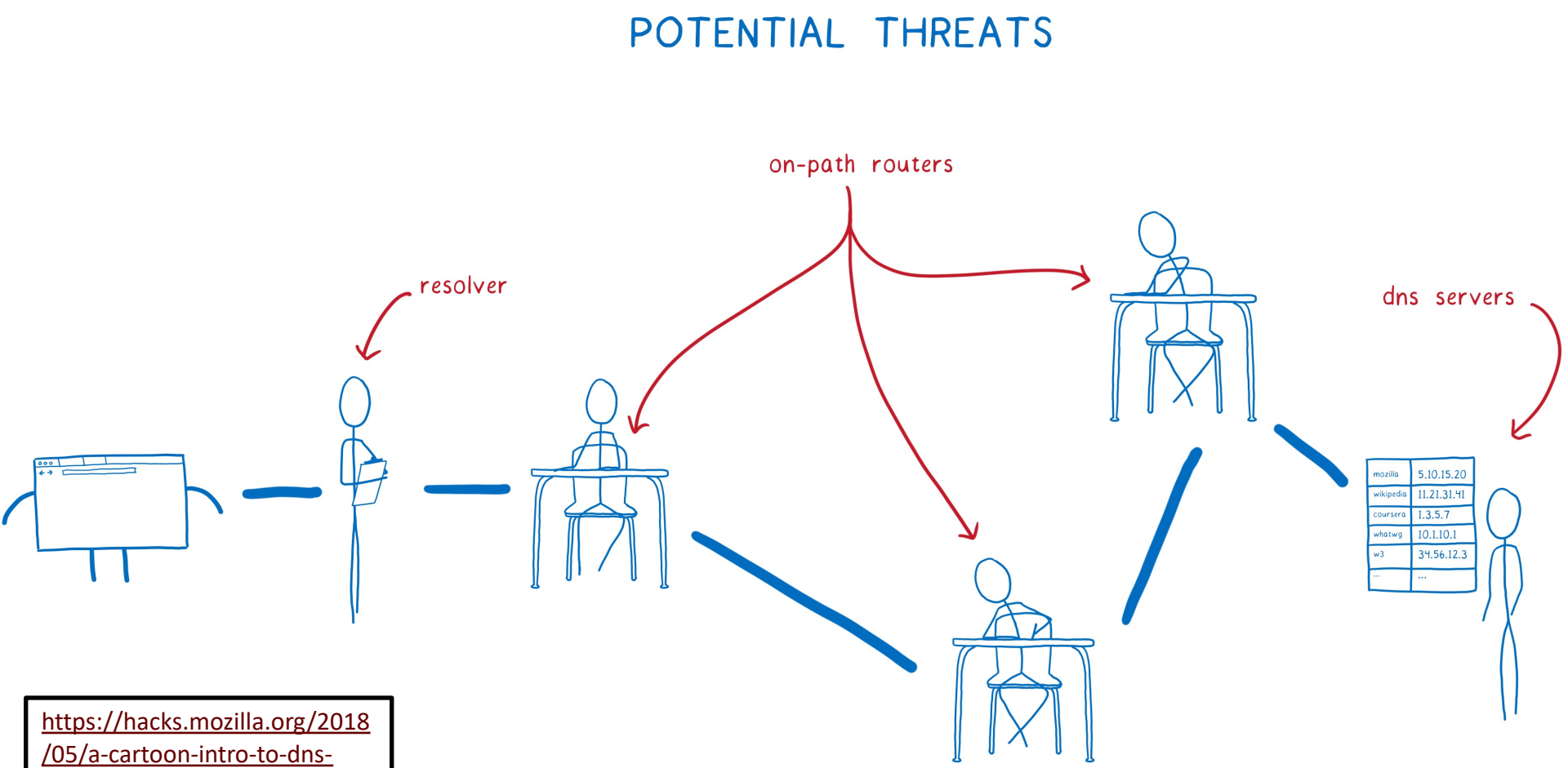

https://hacks.mozilla.org/2018/05/a-cartoon-intro-to-dns-over-https/

# DNS and Security




https://hacks.mozilla.org/2018/05/a-cartoon-intro-to-dns-over-https/

# DNS and Security




https://hacks.mozilla.org/2018/05/a-cartoon-intro-to-dns-over-https/

# DNS and Security

- **Confidentiality**
  - Traditional DNS request and reply (over UDP) is plaintext
    - ISP spies on your Internet usage for profit?
    - NSA spies on your Internet usage for control?
    - DNS is *not just for names*
  - Solutions: **DNS over HTTPS**, **DNS over TLS**
- **Integrity**
  - Traditional DNS request and reply (over UDP) is unsigned
  - ISP tampers with reply message? (NXDOMAIN replaced with ad-laden site)
  - Governments tamper with reply message? (Domain blocked by court order)
  - Hackers tamper with reply message? (Redirect to malware site)
  - Solutions: **DNSSEC** (and DNS over HTTP/TLS)
- **Availability**
  - *Addressed by DNS distributed database design*

https://dnsprivacy.org

# Wrap-Up

- Questions?
- Concerns?

- **Today**
  - **Lab 5** – Web Server (Part 2)
  - **Lab 6** – Web Server (Part 3)