



BEWD My First Rails App

Matt Heath
Tech Lead, Platform
Hailo

AGENDA

- » Intro to Rails
- » Rails & my browser
- » Structure of a Rails app
- » Getting ready with Rails 4
- » Scaffold application
- » Lab time

RAILS

A BIT ABOUT RAILS

- » Rails is an open source web framework written in Ruby
- » Created by David Heinemeier Hansson (@dhh) in 2004 as an extraction of the reusable components of Basecamp



RAILS

FRAMEWORK DEFINED

- » Rails is a framework for building web apps
- » It solves the most common problems faced when building websites or web apps

RAILS

WEBSITES BUILT ON RAILS

GitHub

GOCARDLESS



Basecamp[®]



shopify



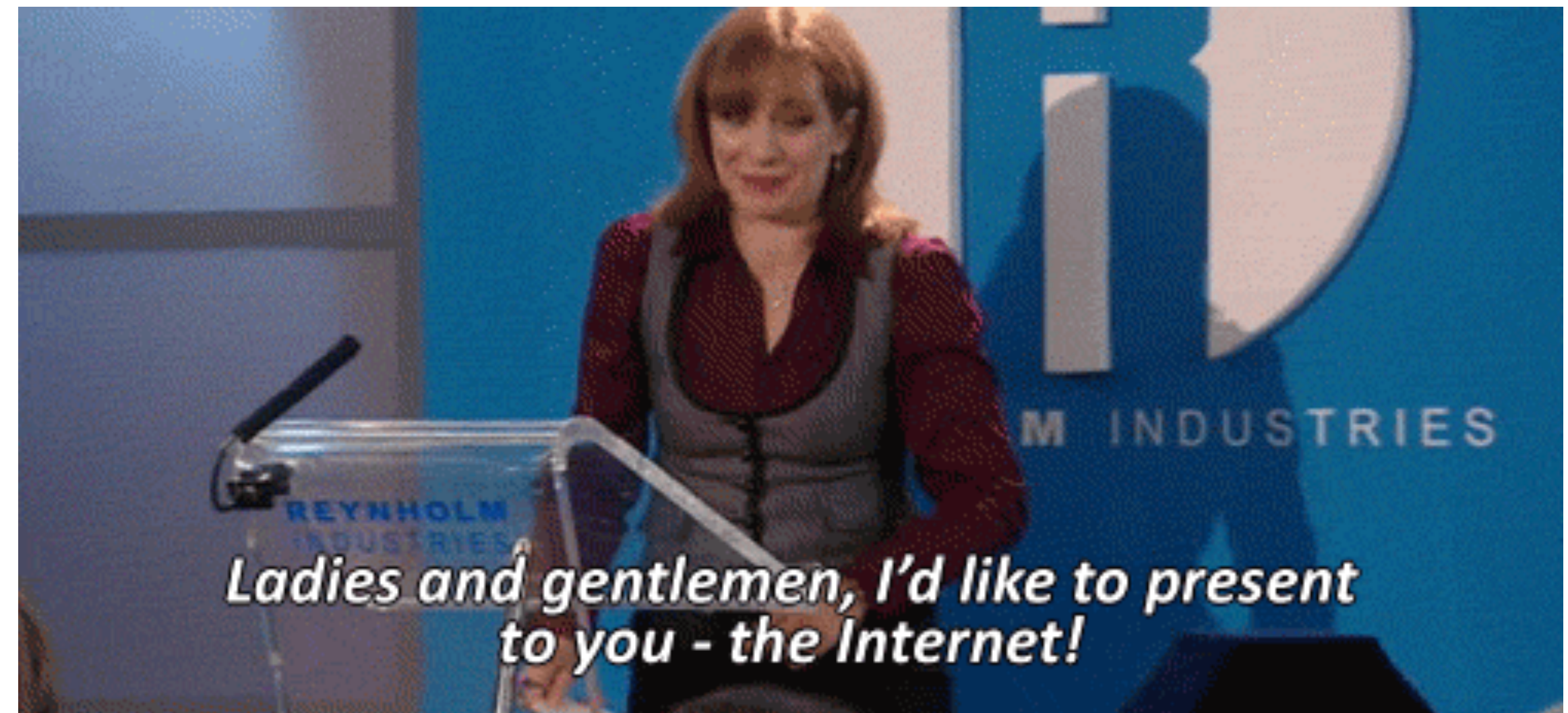
hulu



RAILS

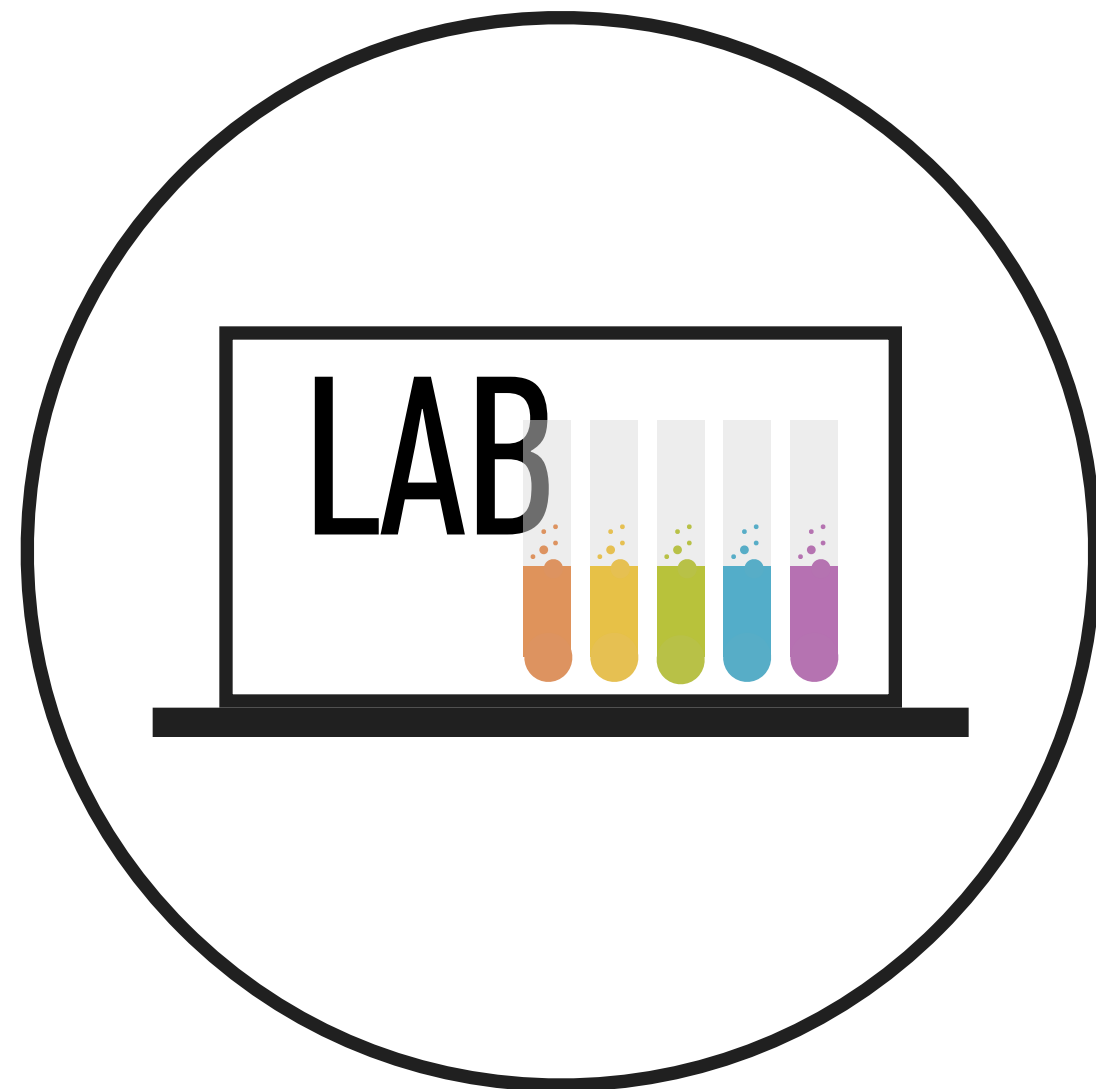
THE INTERNET

» To understand Rails we need to understand the Internet



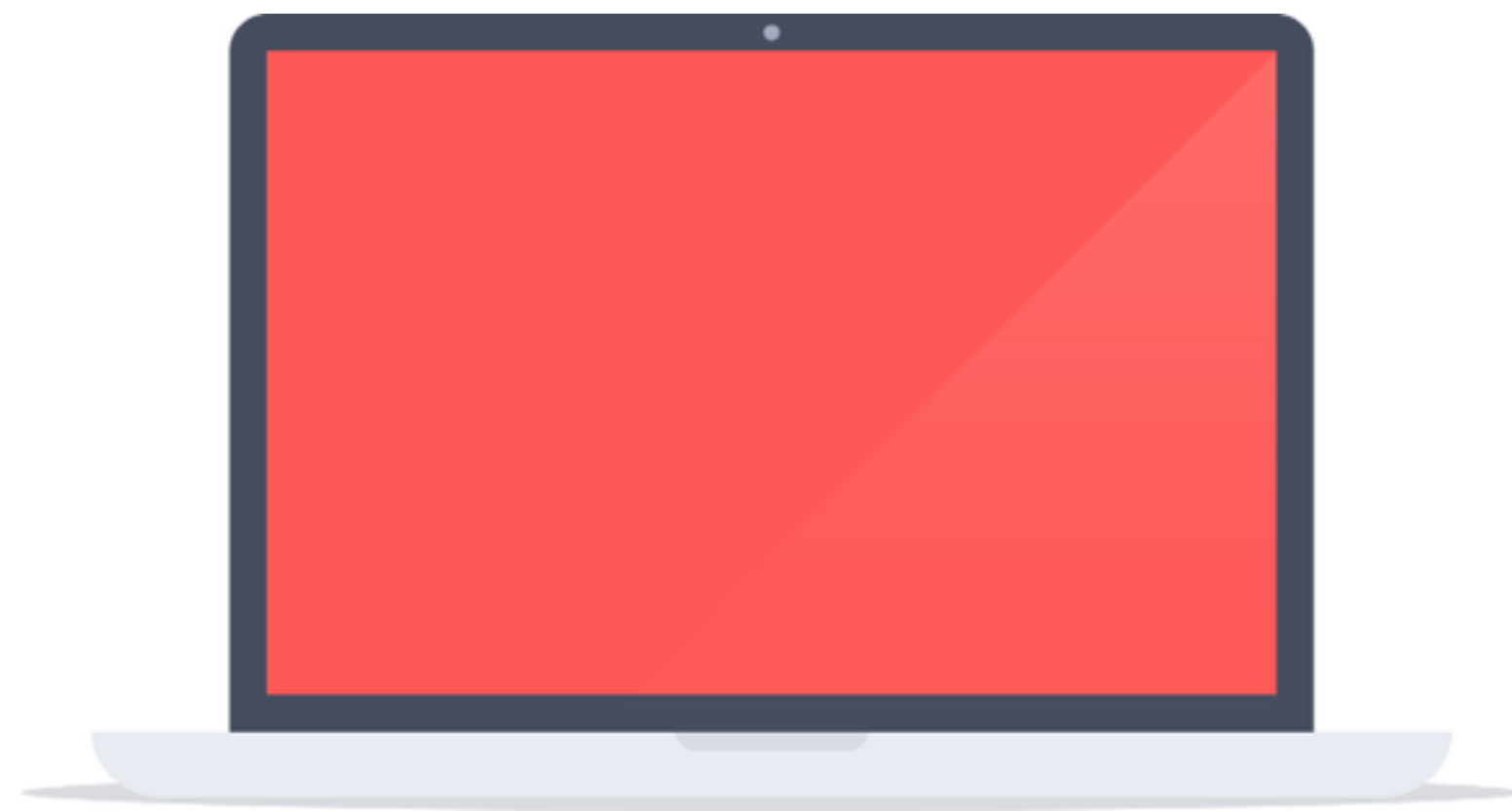
RAILS

DRAW THE INTERNET

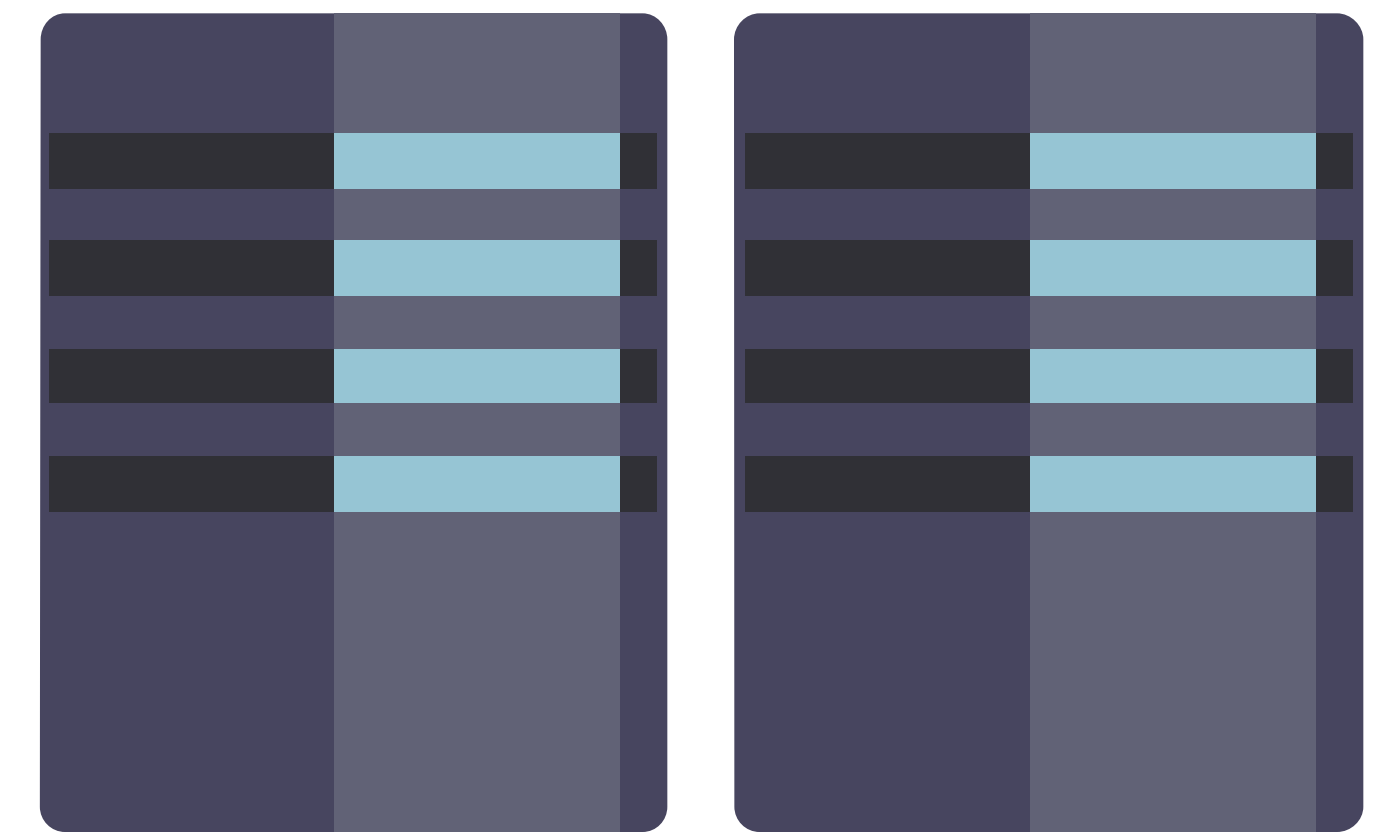


-
- » Get into groups of 2-3.
 - » Draw a diagram that represents your understanding of how the Internet works.
 - » Here are a couple of questions to consider as you draw:
 - » What happens when you hit enter on your address bar?
 - » Where does a website live?

THE INTERNET

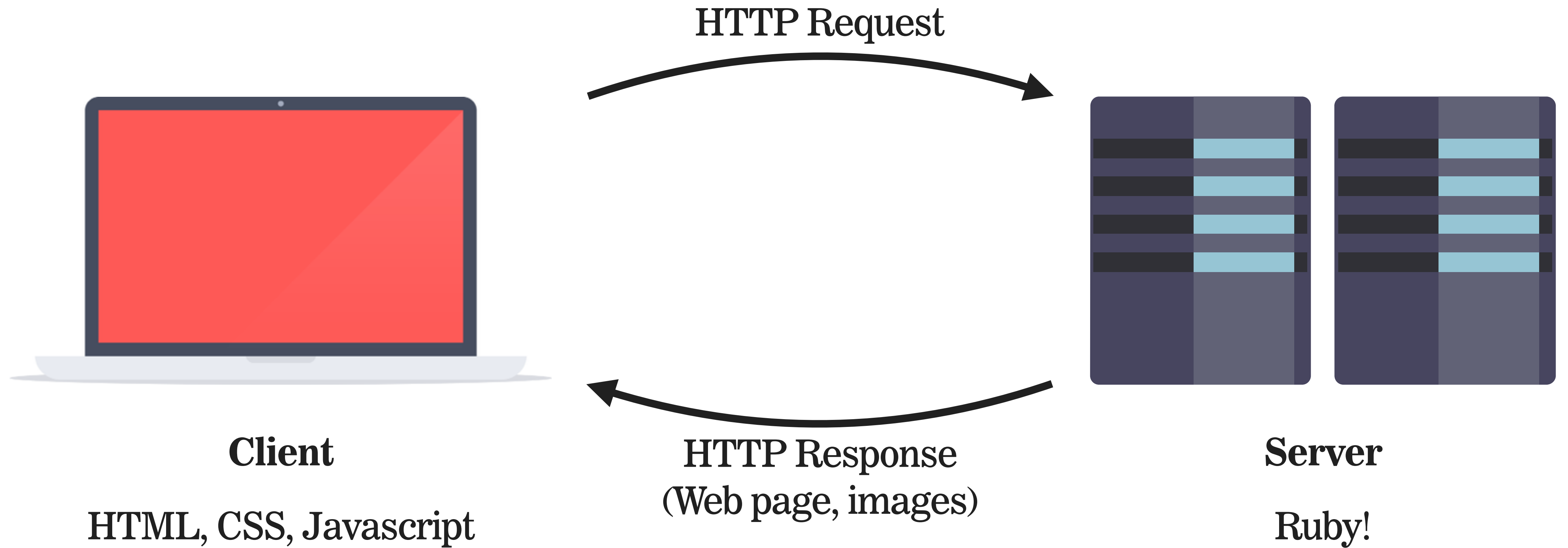


Client

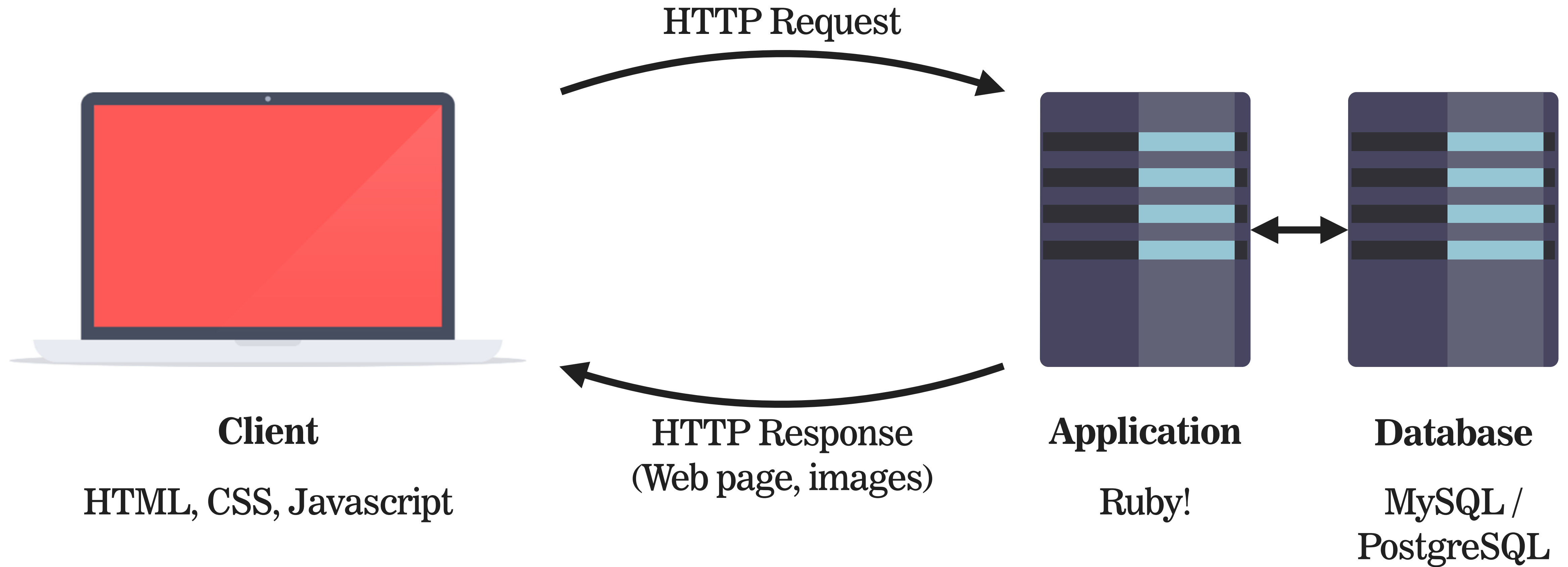


Server

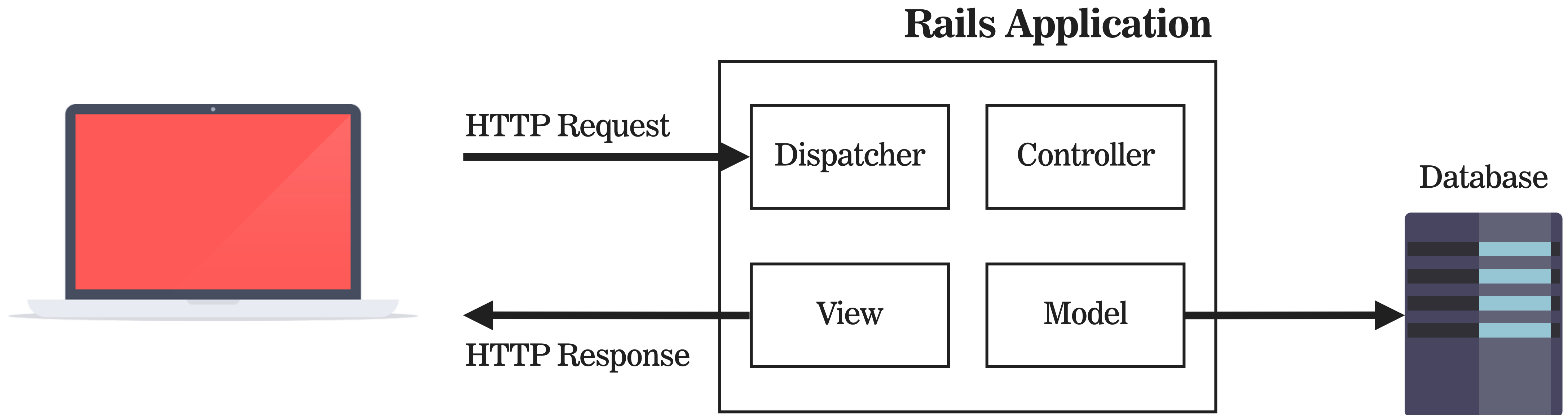
THE INTERNET

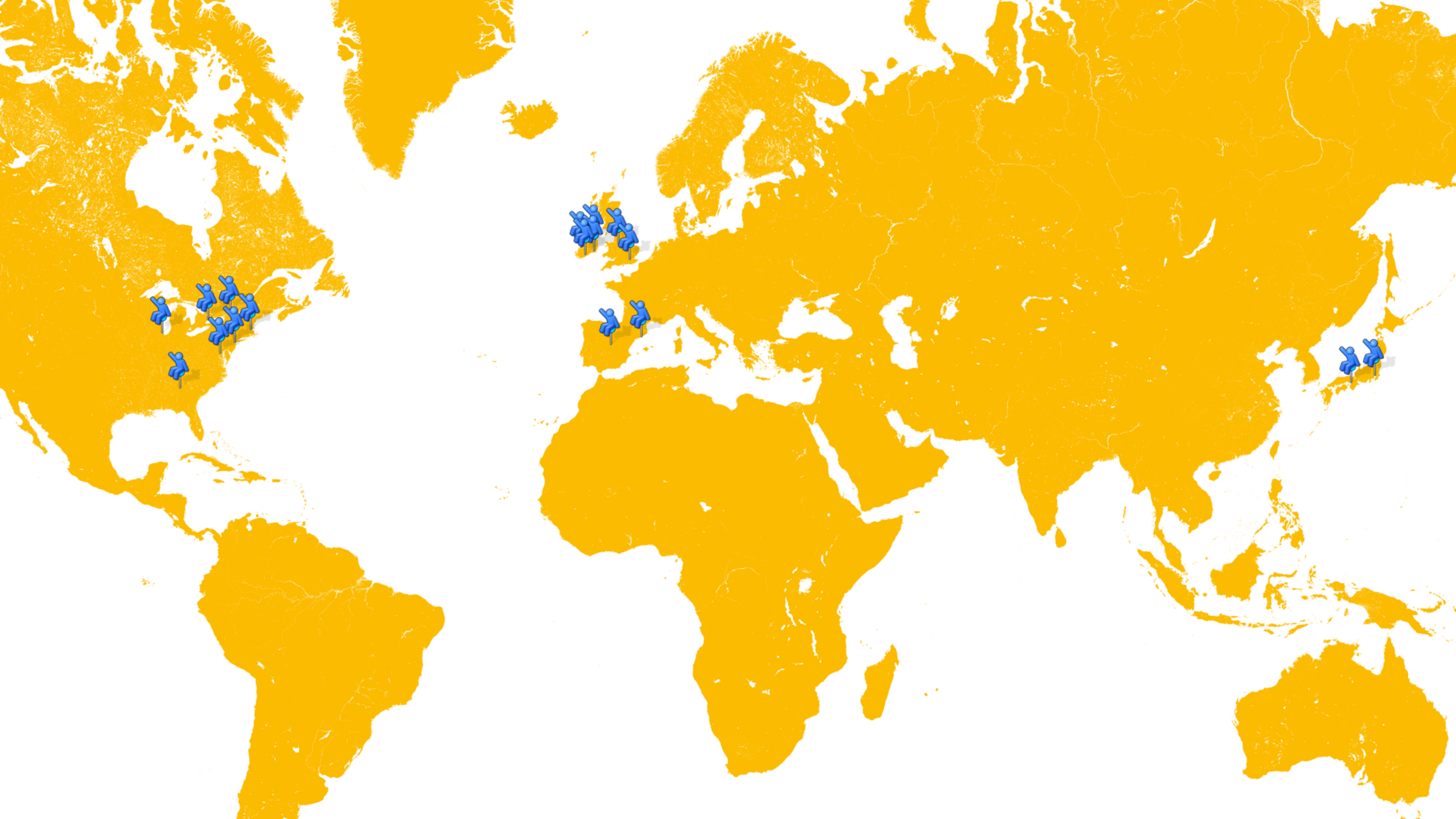


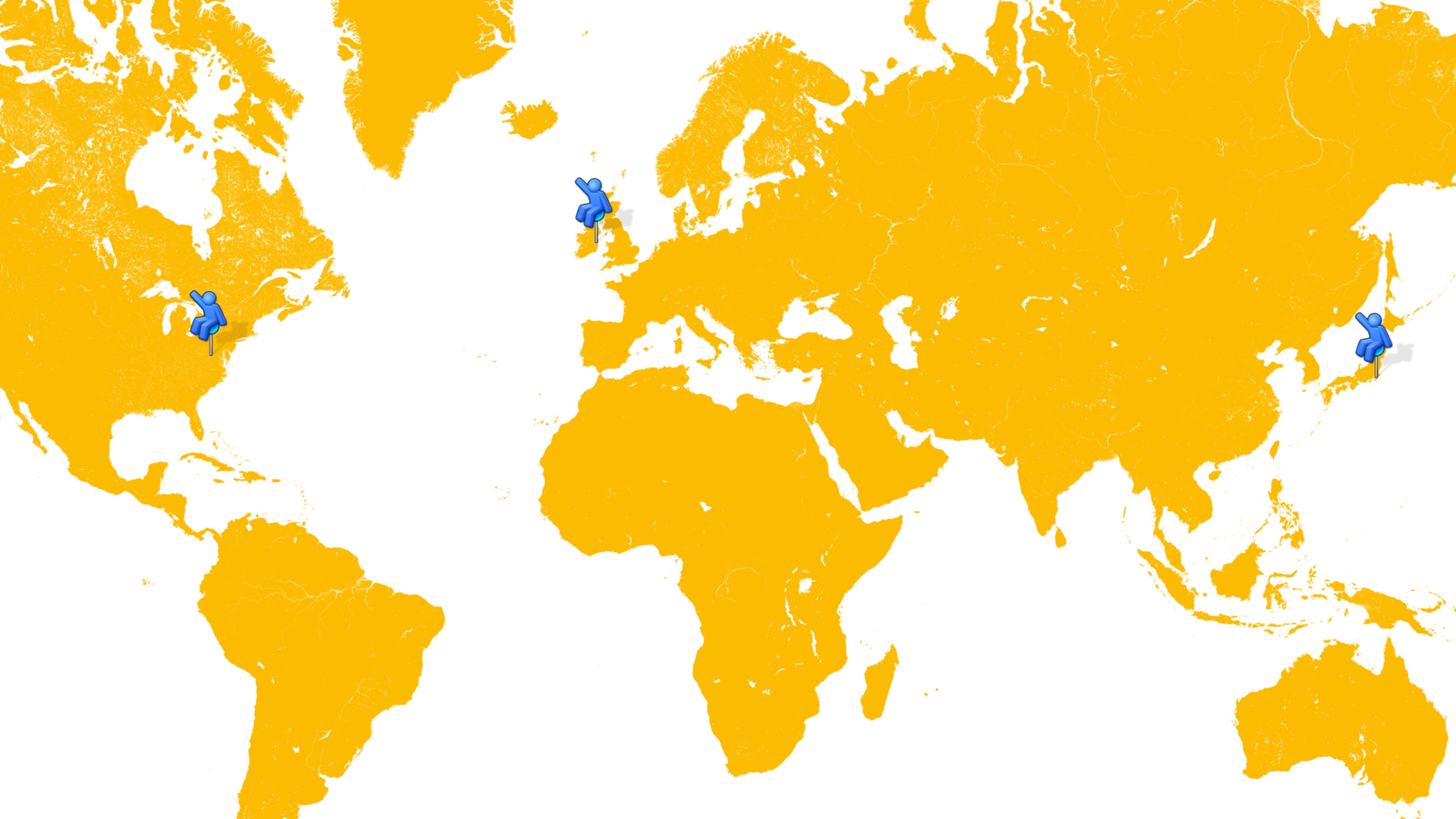
THE INTERNET



RAILS







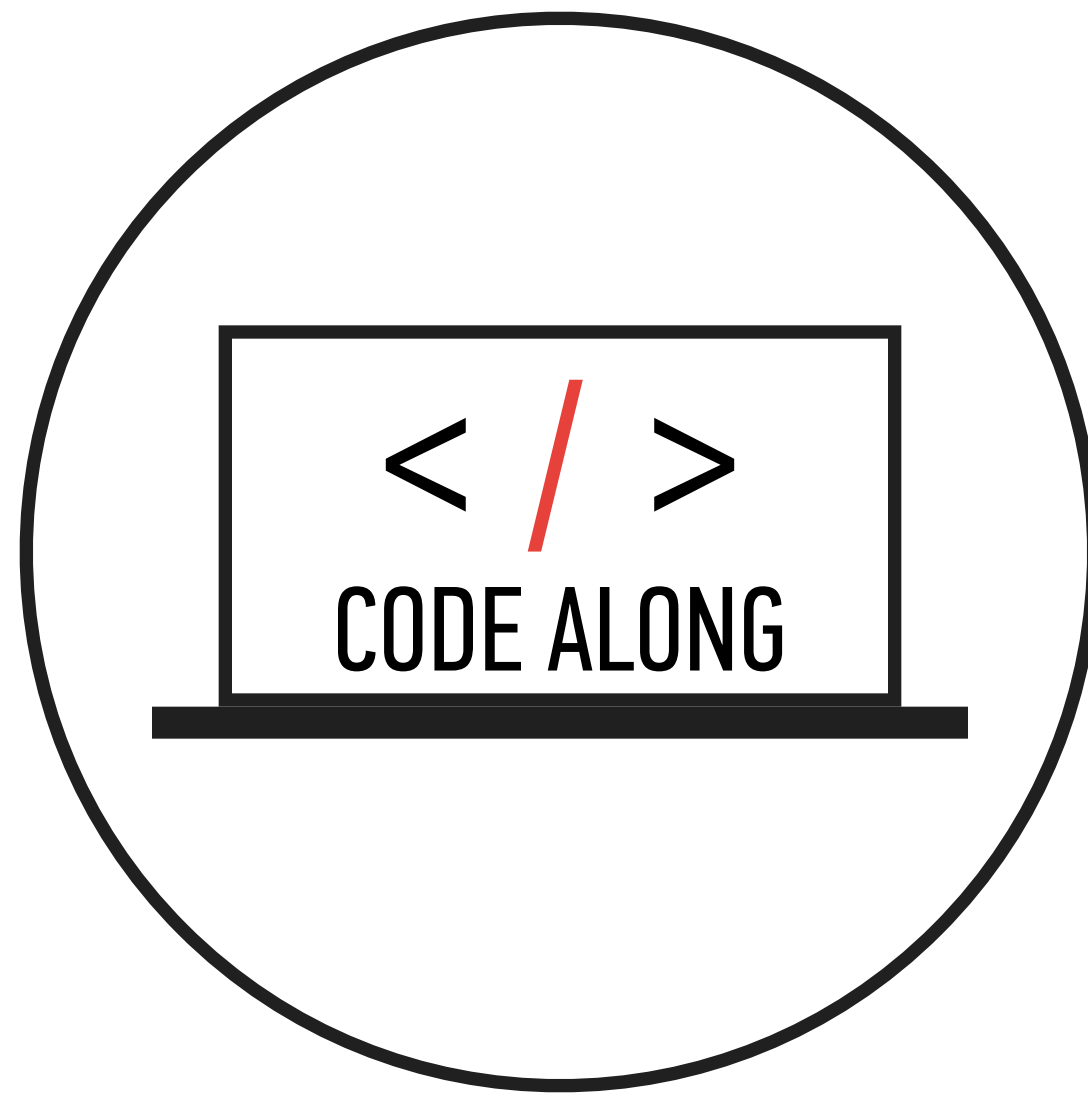
RAILS 4

LET'S GO!

```
$ gem install rails --version 4.1.4 --no-ri --no-rdoc
```

```
$ rbenv rehash
```

```
$ rails -v
```

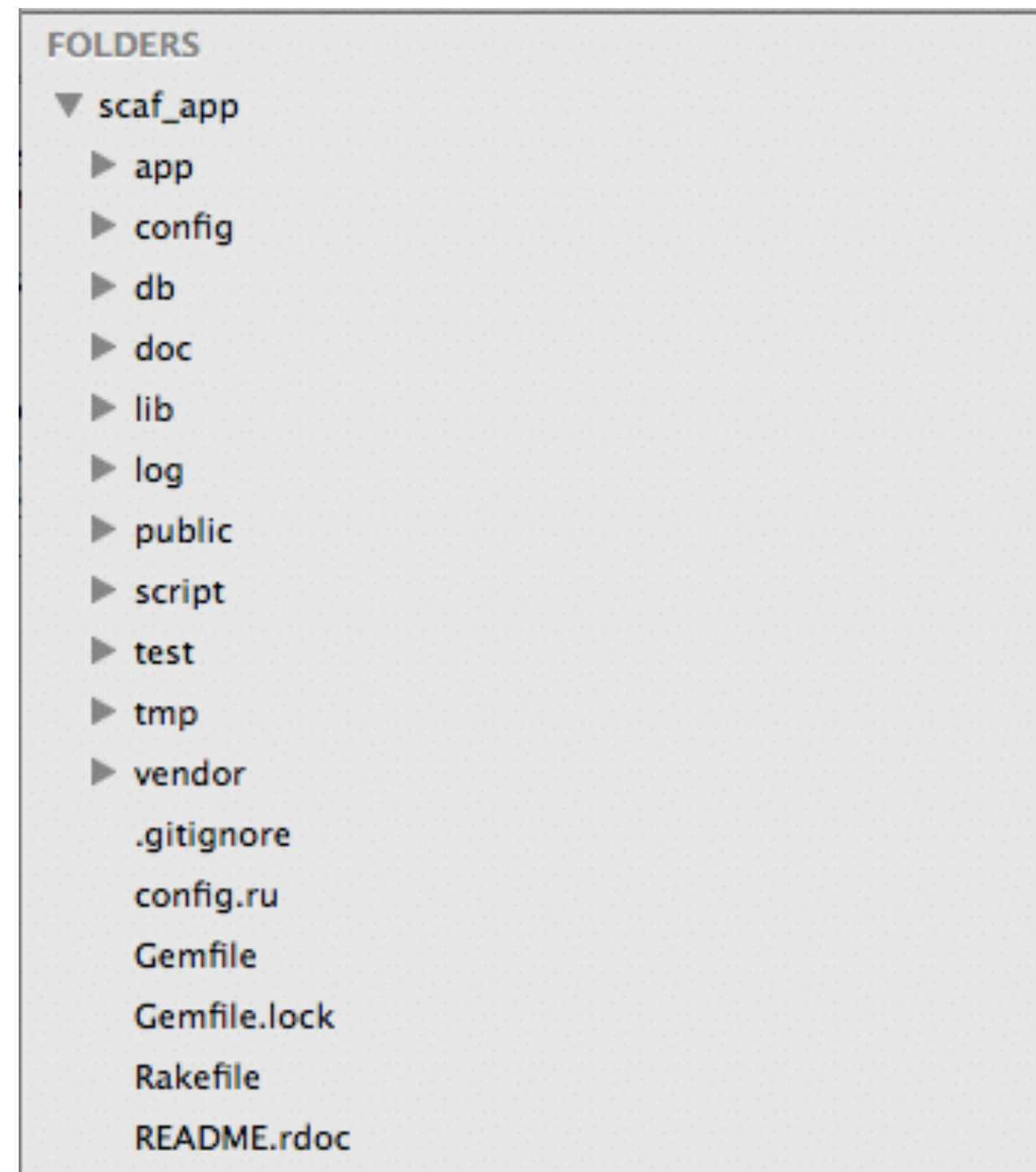


Scaffolding

STRUCTURE

STRUCTURE OF A RAILS APP

- » Most of your application logic belongs in **app/** (including css and javascript files)
- » **public/** contains static files that are directly available



STRUCTURE

THE APP FOLDER

- » Most of the code we write belongs somewhere in the ``app`` folder.
- » **app/assets:** CSS, JavaScript and images used in templates.
- » **app/controllers:** Classes which gather data and render responses for specific requests.
- » **app/helpers:** Utility modules which define methods for templates to use
- » **app/mailers:** Classes that define various outbound emails in your app
- » **app/models:** Classes defining your data models (e.g. User, Comment, Post)
- » **app/views:** Templates called to render HTML.
- » **app/views/layouts:** Master templates to be used throughout your website.

RECAP

MAKE AN APP

```
$ rails new my_app_name  
$ ls  
$ cd my_app_name
```

RECAP

CREATE THE TABLE(S)

```
$ rake db:migrate
```

RECAP

START THE SERVER

```
$ rails s
```

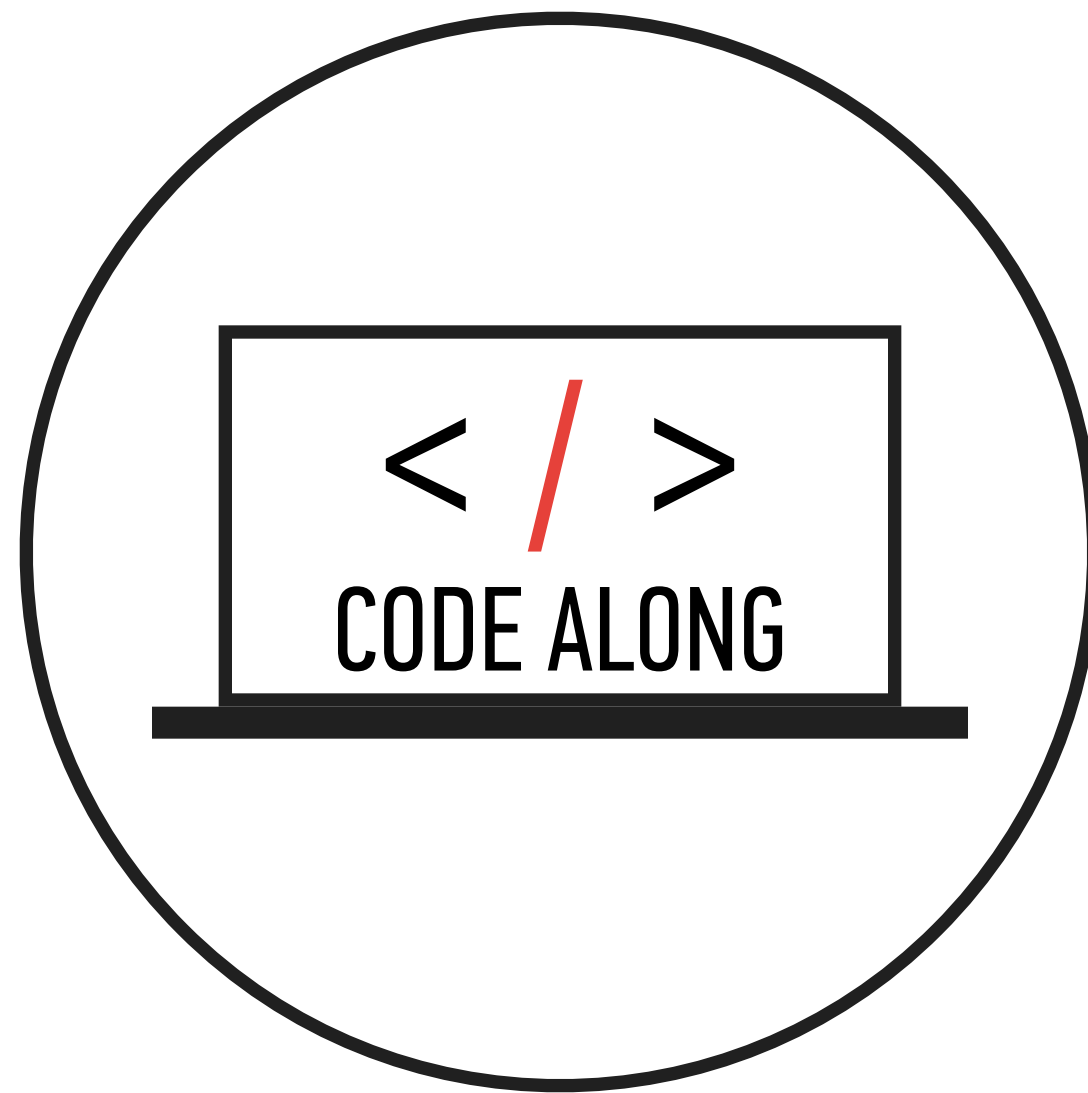
» Go to `localhost:3000/books` in your browser

RECAP

SCAFFOLDING

```
$ rails generate scaffold Book author:string  
title:string abstract:text
```

- » Scaffolds provide the bare necessities for creating, editing, deleting, and viewing a resource. Otherwise known as CRUD.
- » Scaffolds are great when getting started with Rails or a new app, but are generally not used in production websites.
- » Scaffolding is not considered best practice, and should generally only be used initially as a means of learning (the primary drawback of using them is that you generate lots of code that you don't need, and it's difficult to undo).

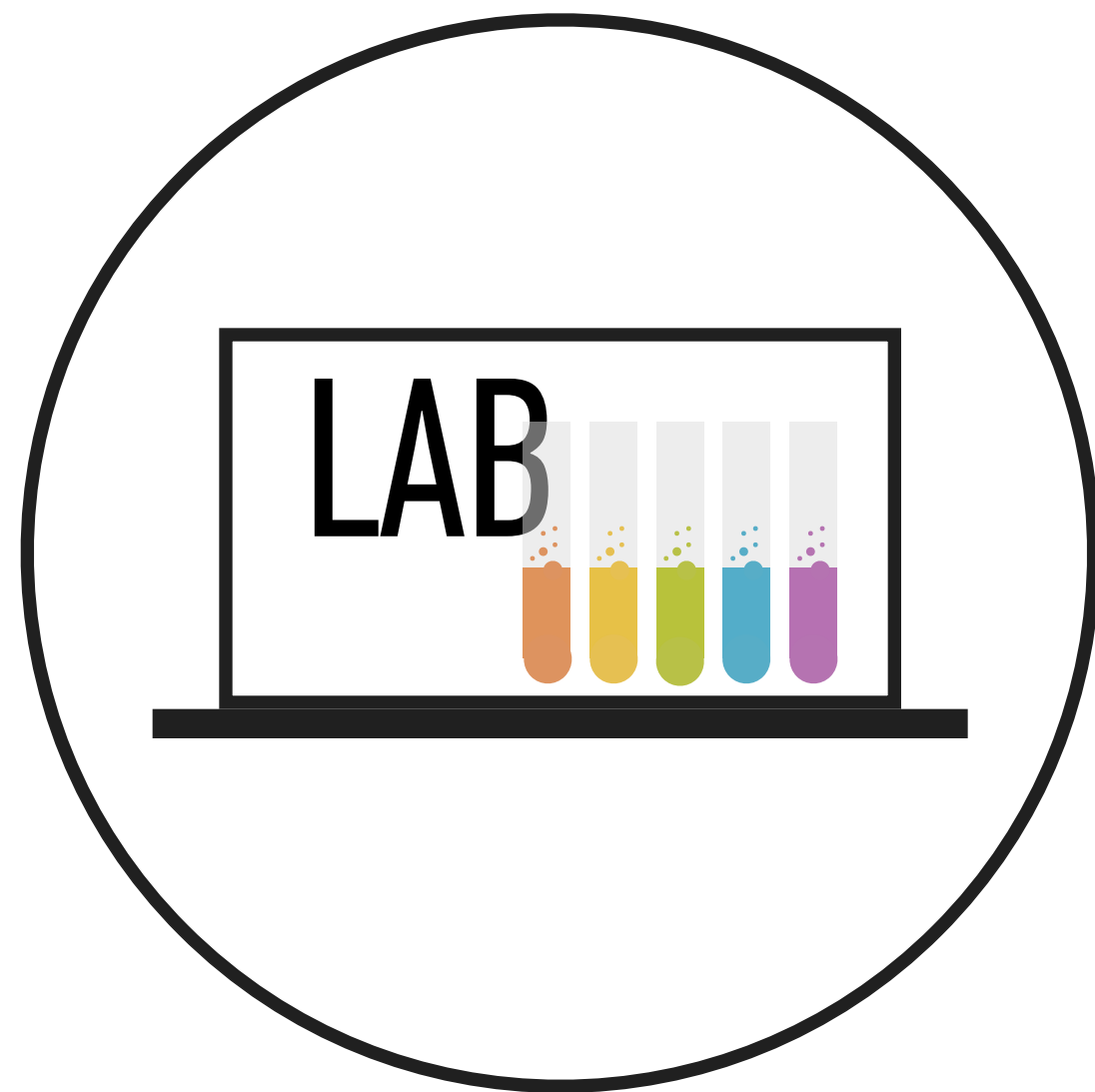


Hosting Public Files

RECAP

HOSTING PUBLIC FILES

- » Rails can be a simple file server too!
- » Add a static file to the 'public' folder :
 - » Create the file `public/hello.txt`
 - » Start your server with `$ rails s`
 - » Visit `http://localhost:3000/hello.txt`



-
- » Create a new application called **User_Scaffold**
 - » The resource should be **User** and it should have the attributes:
 - » First Name
 - » Last Name
 - » Age
 - » Email
 - » **BONUS!** Remove the ability to edit after a user has been created
 - » *Hint: app/views*

RESOURCES

Article by How Stuff Works:

<http://computer.howstuffworks.com/internet/basics/internet.htm>

Five minute video on how the internet works:

http://www.youtube.com/watch?v=7_LPdttKXPc

In depth look at how the internet works from Stanford University:

<http://www.stanford.edu/class/msande91si/www-spr04/readings/week1/InternetWhitepaper.htm>