Building Web Apps with Rack and Sinatra

@blacktm

Objectives

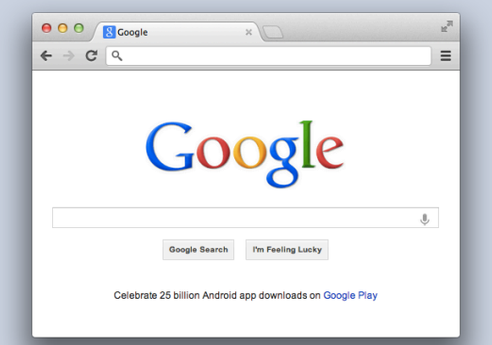
- ✔ Describe the HTTP request process
- ✔ Define Rack, describe its purpose
- ✔ Build a simple Rack web app
- ✔ Define, build, and use middleware
- ✔ Define DSL (compare with “framework”)
- ✔ Build a simple Sinatra app

Things You'll Need

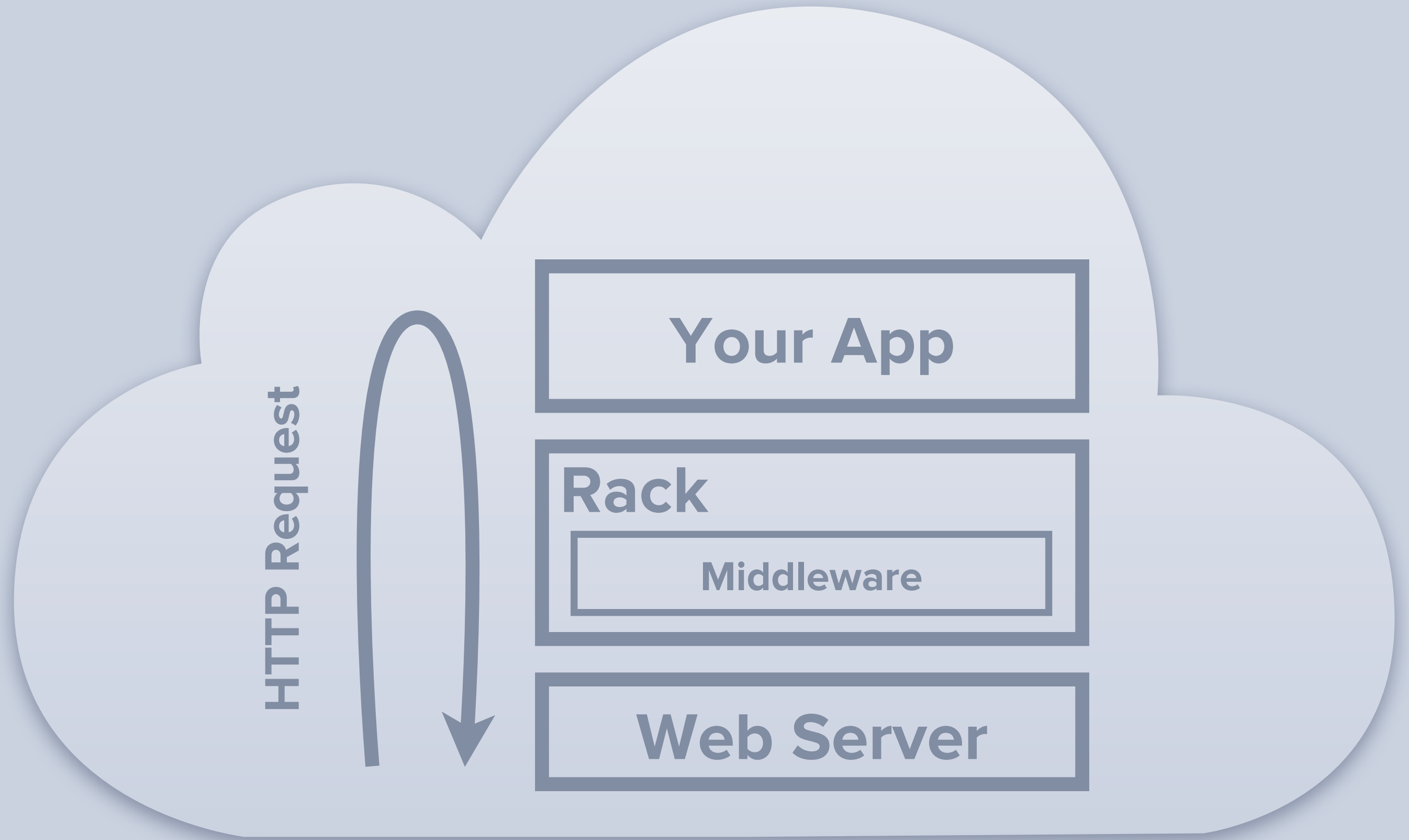
- ✔ gem install thin
- ✔ gem install sinatra
- ✔ gem install heroku
- ✔ Project files and notes at blacktm.com

Rails / Sinatra
Ruby

Deploy



HTML
CSS
JS


HTTP Request
Your App
Rack
Middleware
Web Server

HTTP Request

Your App

Rack

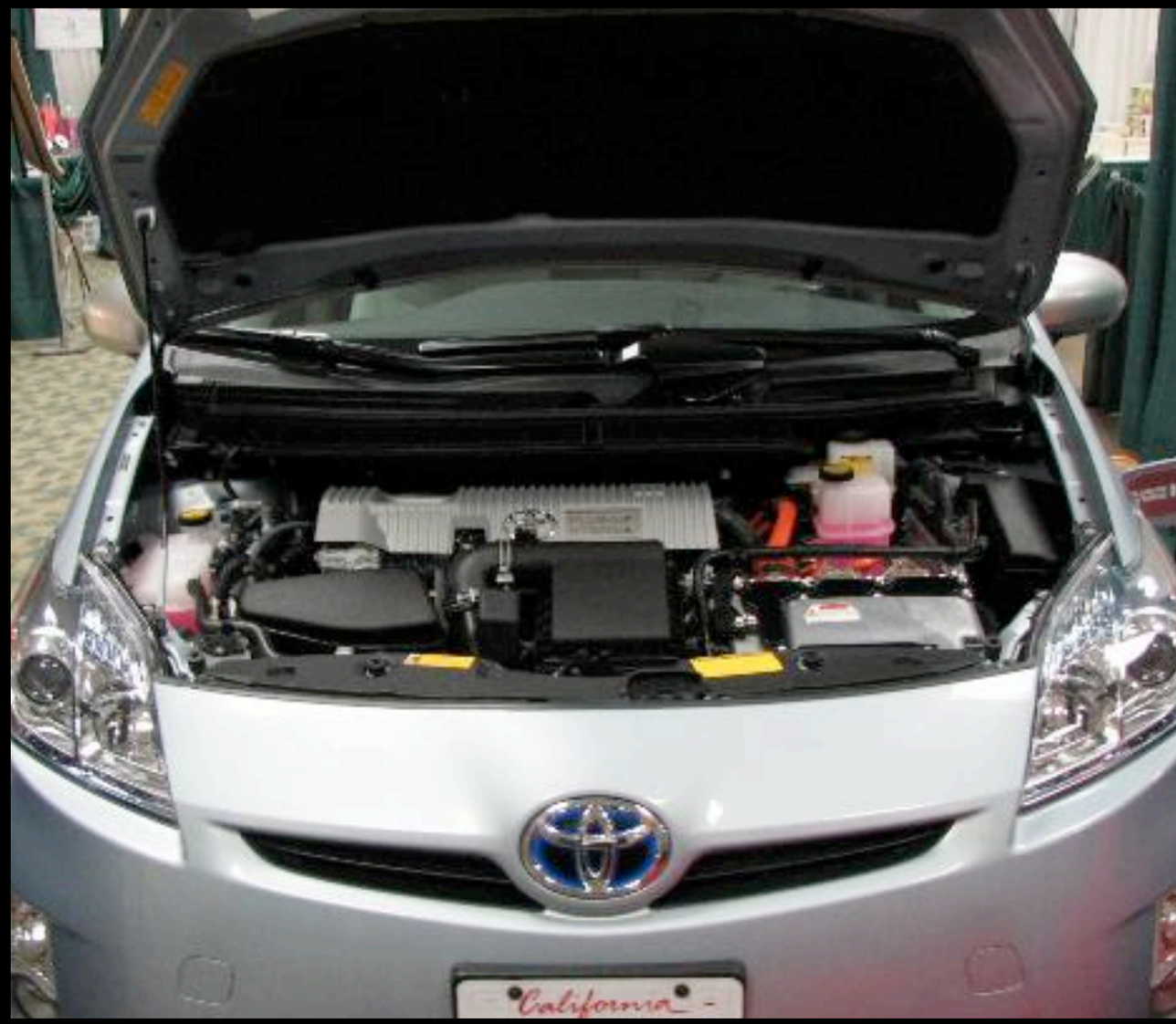
Middleware

Web Server

What is Rack?

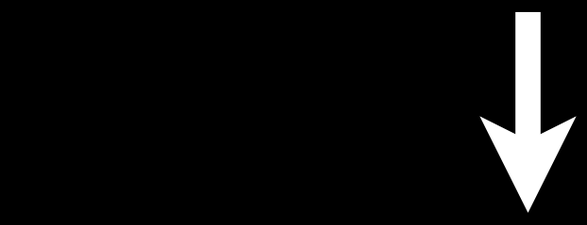
A modular web server interface.

(Connects your Ruby app to the web.)

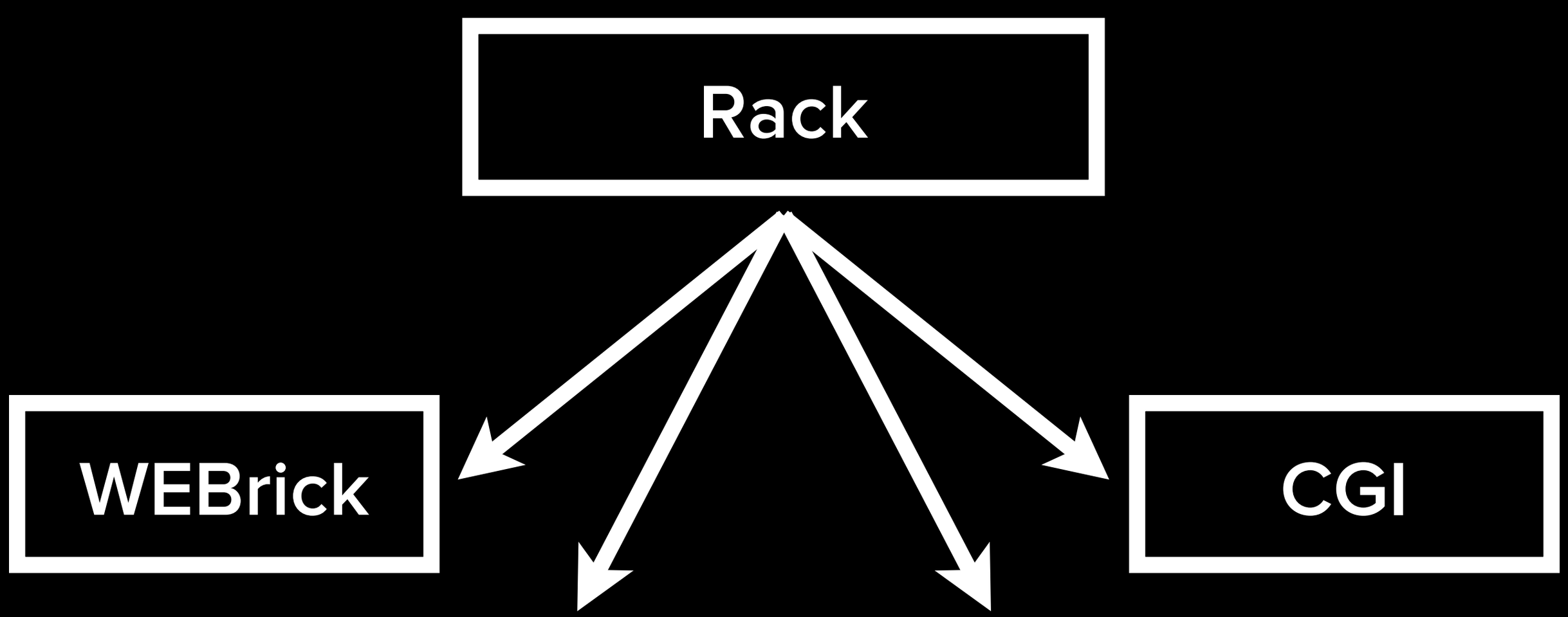

California

A common interface.

Your App



Rack



WEBrick

CGI

Mongrel

Thin

What is a Rack app?

A Ruby object that responds to a “call” method, taking a single hash parameter, and returning an array which contains the response status code, response headers, and response body as an array of strings.

Open “simple_rack” and edit the “config.ru” file.

```ruby
# config.ru - rackup file
require 'rack'

class MyApp
  def call(env)
    headers = { "Content-Type" => "text/html" }
    [200, headers, ["Hello world!"]]
  end
end

run MyApp.new
```

```
$ rackup config.ru
```

A very basic server.

```
http://0.0.0.0:9292

 #=> Hello world!
```

```
http://0.0.0.0:9292/there/isnt/a/route/here

 #=> Hello world!
```

Who cares?

- ✔ You just built a web app. Damn.
- ✔ Important in understanding (and building) web frameworks (e.g. Rails)
- ✔ Middleware!

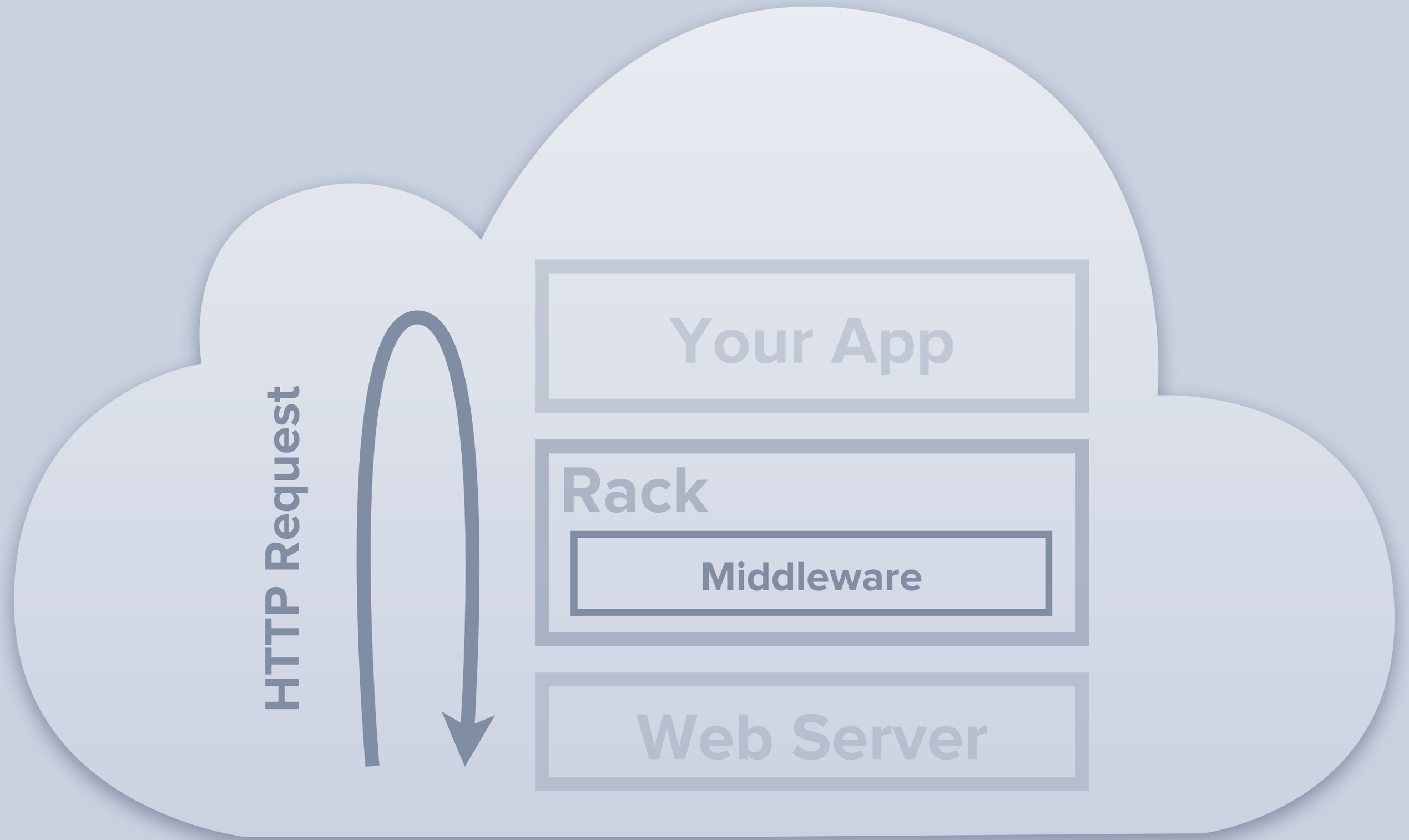
HTTP Request

Your App

Rack

Middleware

Web Server


HTTP Request
Your App
Rack
Middleware
Web Server

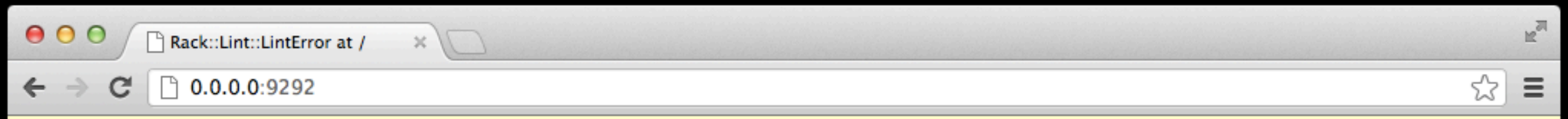
Let’s explore Middleware!

1. Open “rack_middleware”

2. Stop the server (ctrl-c)

3. Run: `$ rackup config.ru`

4. Open: http://0.0.0.0:9292
   or: http://localhost:9292



Rack::Lint::LintError at /

No Content-Type header found

Ruby /Users/Tom/.rvm/gems/ruby-1.9.3-p194/gems/rack-1.4.1/lib/rack/lint.rb: in assert, line 19
Web GET 0.0.0.0/

Jump to:

GET | POST | Cookies | ENV

Traceback (innermost first)

/Users/Tom/.rvm/gems/ruby-1.9.3-p194/gems/rack-1.4.1/lib/rack/lint.rb: in assert

```
19.           raise LintError, message                                                    ...
```

/Users/Tom/.rvm/gems/ruby-1.9.3-p194/gems/rack-1.4.1/lib/rack/lint.rb: in check_content_type

```
476.       assert("No Content-Type header found") {                                       ...
```

/Users/Tom/.rvm/gems/ruby-1.9.3-p194/gems/rack-1.4.1/lib/rack/lint.rb: in _call

```
54.       check_content_type status, headers                                              ...
```

/Users/Tom/.rvm/gems/ruby-1.9.3-p194/gems/rack-1.4.1/lib/rack/lint.rb: in call

```
36.       dup._call(env)                                                                 ...
```

/Users/Tom/.rvm/gems/ruby-1.9.3-p194/gems/rack-1.4.1/lib/rack/showexceptions.rb: in call

```
24.       @app.call(env)                                                                 ...
```

/Users/Tom/.rvm/gems/ruby-1.9.3-p194/gems/rack-1.4.1/lib/rack/commonlogger.rb: in call

```
20.       status, header, body = @app.call(env)                                          ...
```

/Users/Tom/.rvm/gems/ruby-1.9.3-p194/gems/rack-1.4.1/lib/rack/chunked.rb: in call

```
43.       status, headers, body = @app.call(env)                                         ...
```

/Users/Tom/.rvm/gems/ruby-1.9.3-p194/gems/rack-1.4.1/lib/rack/content_length.rb: in call

```
14.       status, headers, body = @app.call(env)                                         ...
```

/Users/Tom/.rvm/gems/ruby-1.9.3-p194/gems/thin-1.5.0/lib/thin/connection.rb: in block in pre_process

```
81.         response = @app.call(@request.env)                                           ...
```

/Users/Tom/.rvm/gems/ruby-1.9.3-p194/gems/thin-1.5.0/lib/thin/connection.rb: in catch

```
79.       catch(:async) do                                                               ...
```

Rack::Lint::LintError at /

0.0.0.0:9292

Rack::Lint::LintError at /

No Content-Type header found

Ruby /Users/Tom/.rvm/gems/ruby-1.9.3-p194/gems/rack-1.4.1/lib/rack/lint.rb: in assert, line 19
Web GET 0.0.0.0/

Jump to:

GET | POST | Cookies | ENV

Traceback (innermost first)

/Users/Tom/.rvm/gems/ruby-1.9.3-p194/gems/rack-1.4.1/lib/rack/lint.rb: in assert

```
19.           raise LintError, message                                    ...
```

/Users/Tom/.rvm/gems/ruby-1.9.3-p194/gems/rack-1.4.1/lib/rack/lint.rb: in check_content_type

```
476.       assert("No Content-Type header found") {                       ...
```

/Users/Tom/.rvm/gems/ruby-1.9.3-p194/gems/rack-1.4.1/lib/rack/lint.rb: in _call

```
54.        check_content_type status, headers                             ...
```

```
36.        dup._call(env)                                                 ...
```

/Users/Tom/.rvm/gems/ruby-1.9.3-p194/gems/rack-1.4.1/lib/rack/showexceptions.rb: in call

```
24.        @app.call(env)
```

/Users/Tom/.rvm/gems/ruby-1.9.3-p194/gems/rack-1.4.1/lib/rack/commonlogger.rb: in call

```
20.        status, header, body = @app.call(env)
```

/Users/Tom/.rvm/gems/ruby-1.9.3-p194/gems/rack-1.4.1/lib/rack/chunked.rb: in call

```
43.        status, headers, body = @app.call(env)
```

/Users/Tom/.rvm/gems/ruby-1.9.3-p194/gems/rack-1.4.1/lib/rack/content_length.rb: in call

```
14.        status, headers, body = @app.call(env)
```

/Users/Tom/.rvm/gems/ruby-1.9.3-p194/gems/thin-1.5.0/lib/thin/connection.rb: in block in pre_process

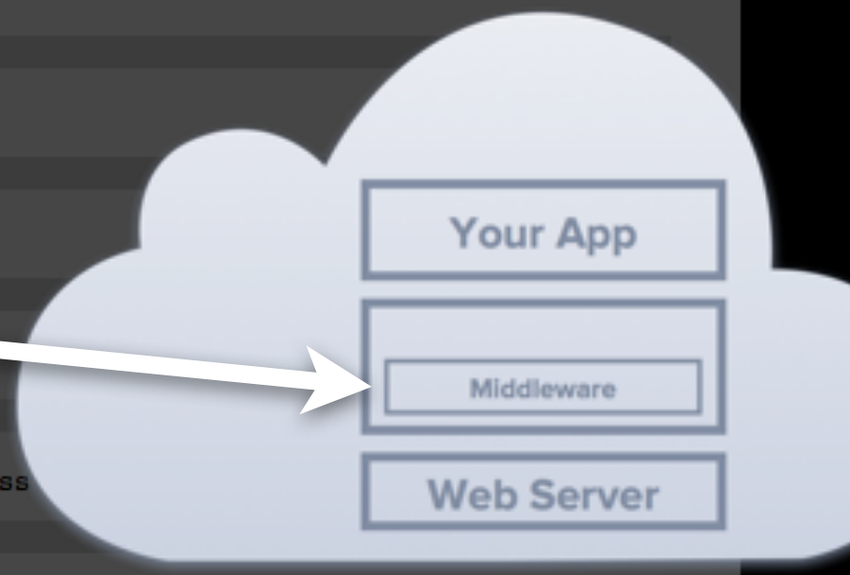
```
81.          response = @app.call(@request.env)
```

/Users/Tom/.rvm/gems/ruby-1.9.3-p194/gems/thin-1.5.0/lib/thin/connection.rb: in catch

```
79.        catch(:async) do                                               ...
```

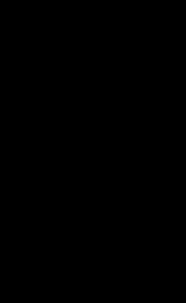
Rack's Default Middleware (in development mode):

```
Rack::CommonLogger
Rack::ShowExceptions
Rack::Lint
```



```ruby
# app.rb

class MyApp
  def call(env)
    puts "Hello from MyAPP"
    return [200, {}, ["Hello"]]
  end
end
```



No "Content-Type"

Solution?

Middleware!

Open “config.ru”

```
class ContentType
  def initialize(app)
    @app = app
  end

  def call(env)
    status, headers, body = @app.call(env)
    puts "Hello from ContentType"

    # TODO: Add "Content-Type" to the HTTP
    #       headers and return the response.

  end
end
```

Add the “Content-Type” to the headers.

```
# TODO: Add "Content-Type" to the HTTP
#       headers and return the response.
headers.merge!( "Content-Type" => "text/html" )
return [status, headers, body]
```

```ruby
class FinishSentence
  def initialize(app)
    @app = app
  end

  def call(env)
    status, headers, body = @app.call(env)
    puts "Hello from FinishSentence"

    # TODO: Add " world!" to the HTTP
    #       body and return the response.

  end
end
```

Add the “ world!” to the body.

```
# TODO: Add " world!" to the HTTP
#       body and return the response.
body.push(" world!")
return [status, headers, body]
```

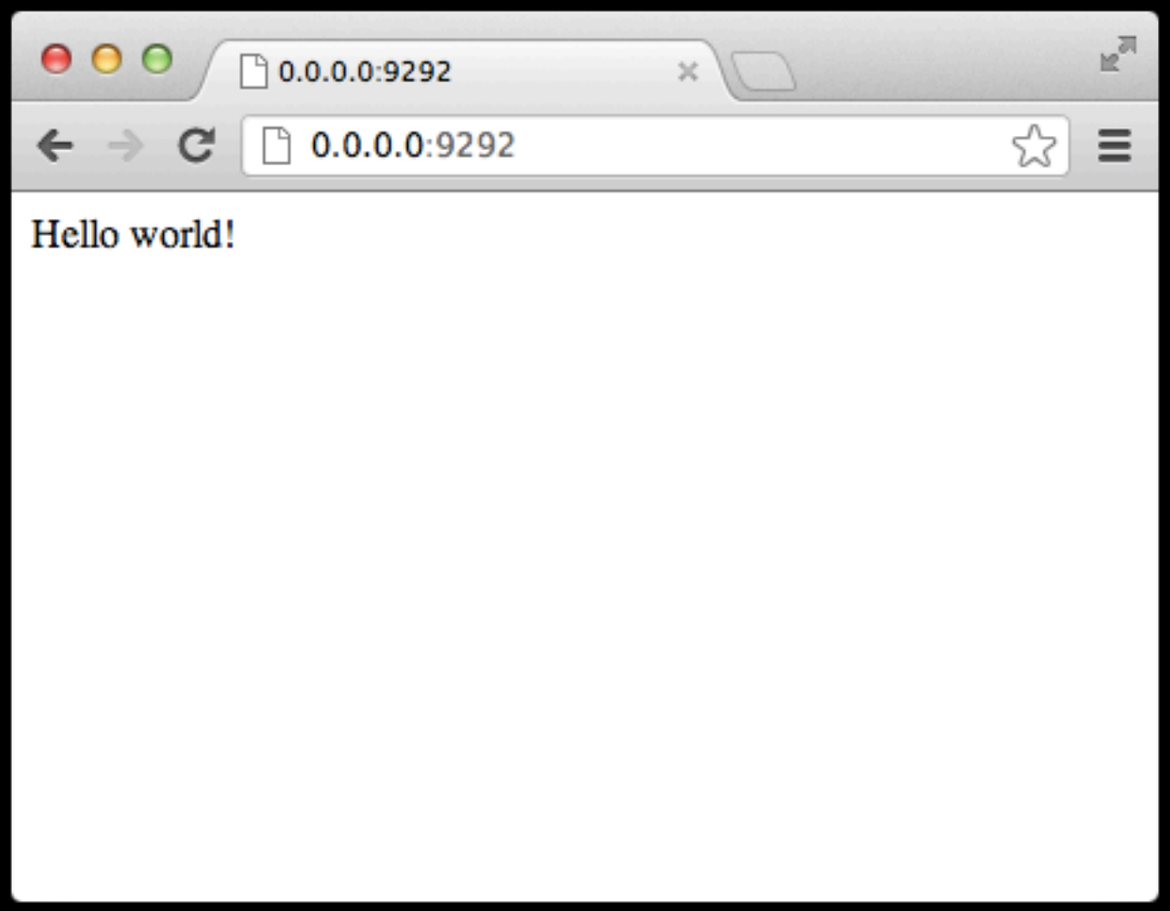
Use the middleware.

```
# TODO: Tell the app to use the "ContentType"
#       and "FinishSentence" middleware.
use ContentType
use FinishSentence
```

Run it!

```
$ rackup config.ru

http://0.0.0.0:9292
```




```
127.0.0.1 - - [13/Oct/2012 21:18:58] "GET / HTTP/1.1" 200 12 0.0009
```

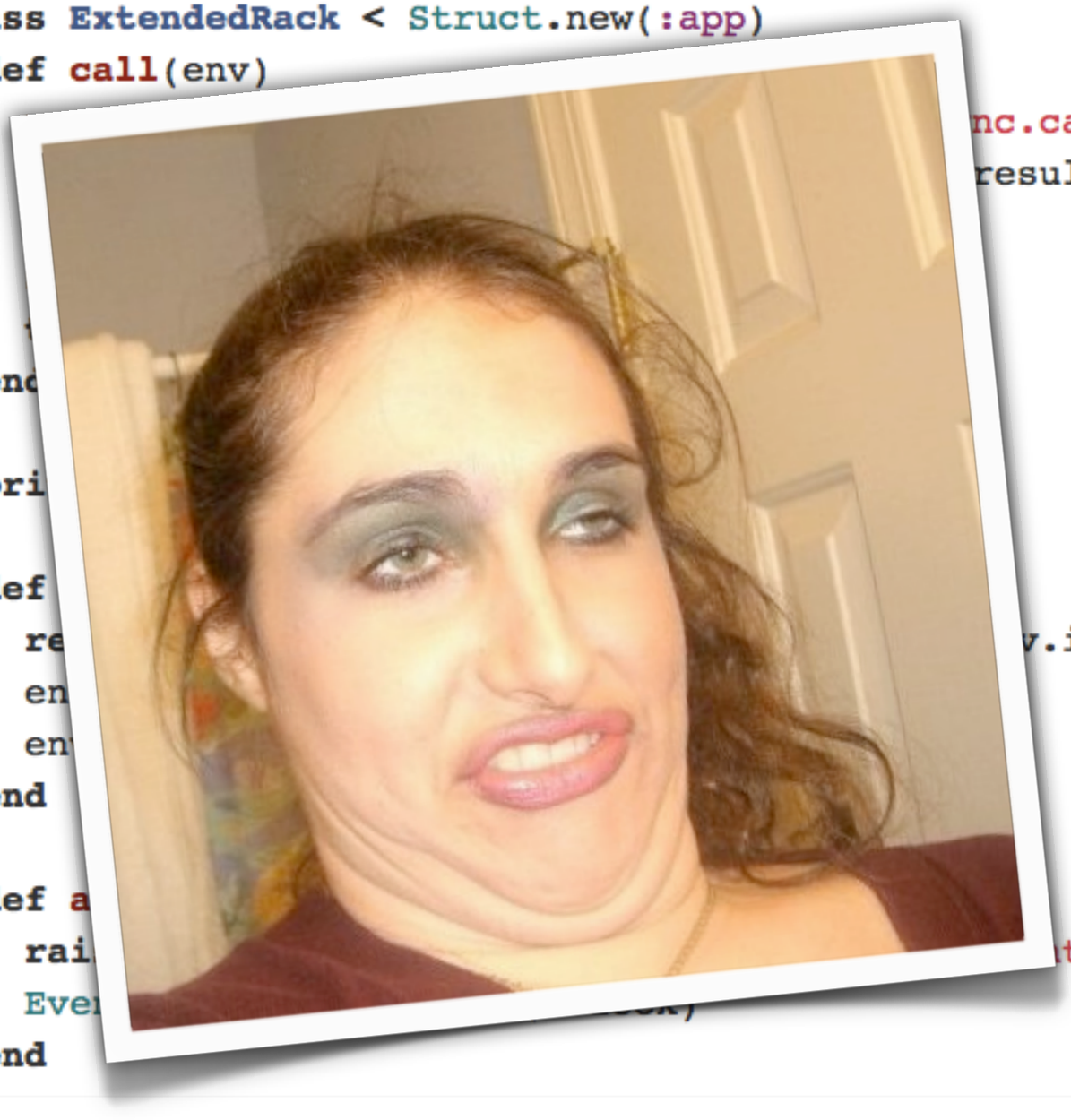
```
rack_middleware:$ rackup config.ru
>> Thin web server (v1.5.0 codename Knife)
>> Maximum connections set to 1024
>> Listening on 0.0.0.0:9292, CTRL+C to stop
{ Hello from MyAPP
{ Hello from FinishSentence
{ Hello from ContentType
  127.0.0.1 - - [14/Nov/2012 17:01:07] "GET / HT
```

What's wrong with this?

```
97    class ExtendedRack < Struct.new(:app)
98      def call(env)
99        result, callback = app.call(env), env['async.callback']
100       return result unless callback and async?(*result)
101       after_response { callback.call result }
102       setup_close(env, *result)
103       throw :async
104     end
105
106     private
107
108     def setup_close(env, status, header, body)
109       return unless body.respond_to? :close and env.include? 'as
110       env['async.close'].callback { body.close }
111       env['async.close'].errback { body.close }
112     end
113
114     def after_response(&block)
115       raise NotImplementedError, "only supports EventMachine at
116       EventMachine.next_tick(&block)
117     end
```

What's wrong with this?

```
97    class ExtendedRack < Struct.new(:app)
98      def call(env)
99                                                    nc.callback']
100                                                   result)
101
102
103
104     end
105
106     pri
107
108     def
109       re                                          v.include? 'as
110       en
111       en
112     end
113
114     def a
115       rai                                         tMachine at
116       Ever                                  ck)
117     end
```



Fork me on GitHub

Sinatra

Put this in your pipe

```
require 'sinatra'

get '/hi' do
  "Hello World!"
end
```

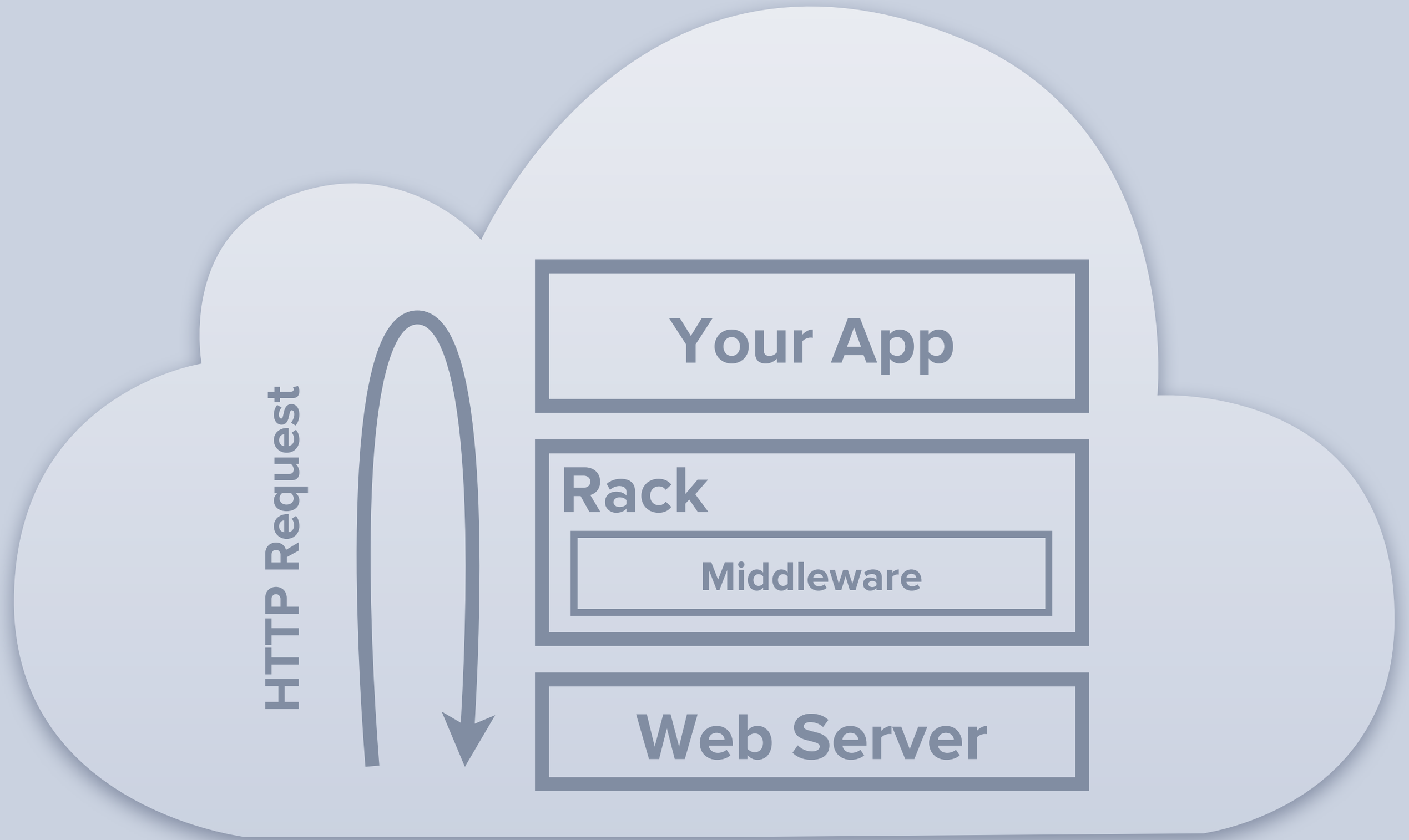
and smoke it

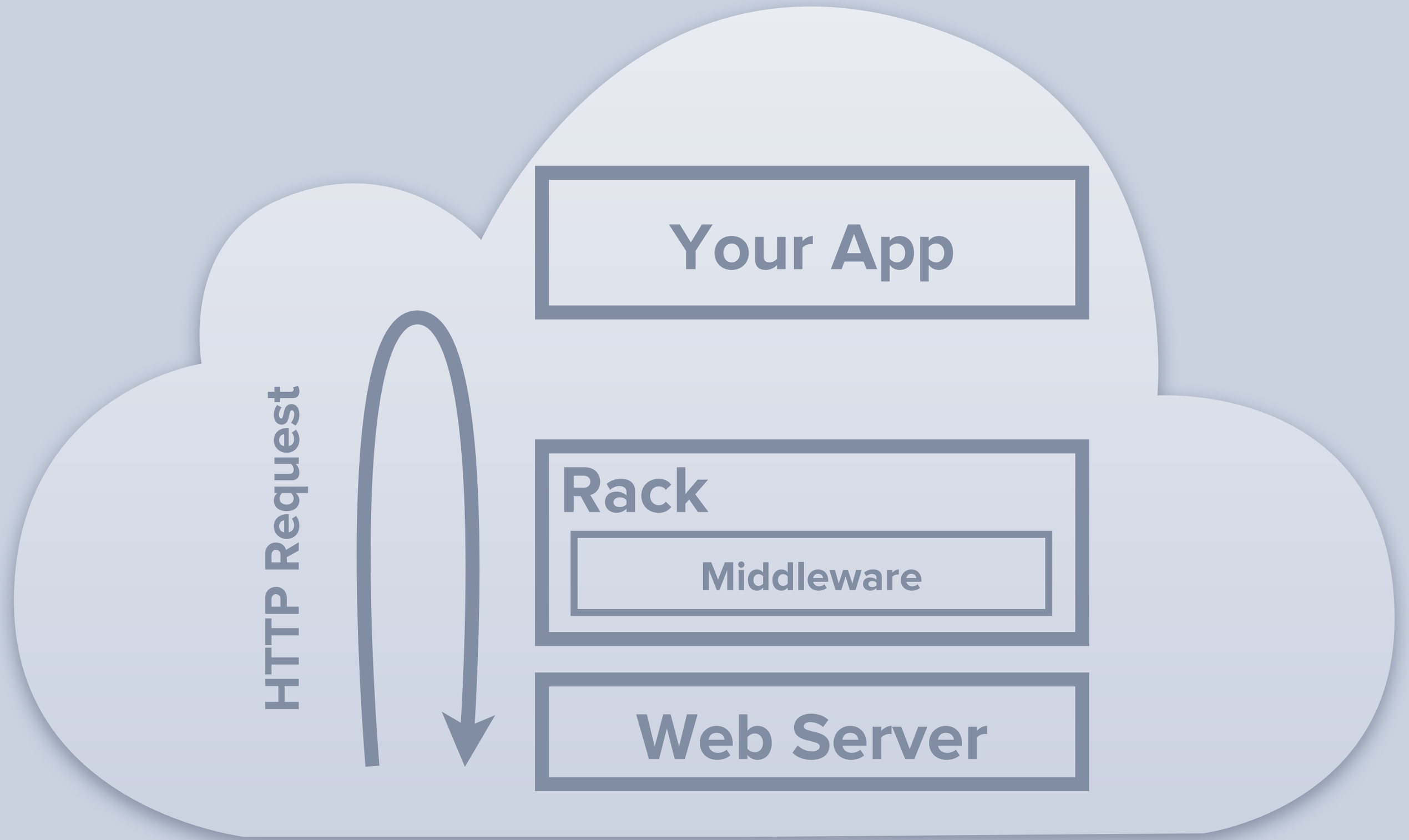
```
$ gem install sinatra
$ ruby hi.rb
== Sinatra has taken the stage ..
>> Listening on 0.0.0.0:4567
```

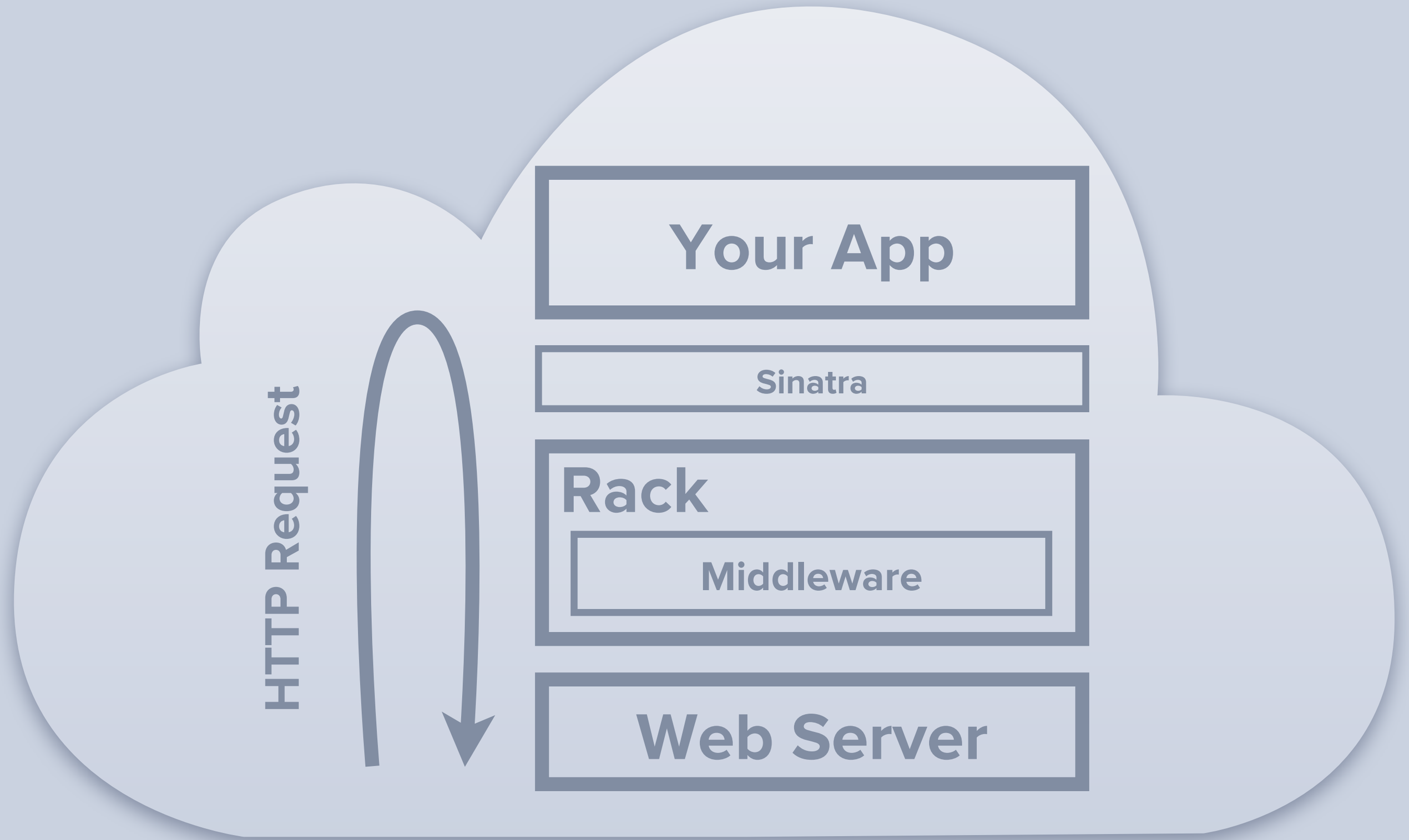
"Domain Specific Language"

"Sinatra is a DSL for quickly creating web applications in Ruby with minimal effort."

A "human interface" to Rack.


HTTP Request
Your App
Rack
Middleware
Web Server


Your App
HTTP Request
Rack
Middleware
Web Server


HTTP Request
Your App
Sinatra
Rack
Middleware
Web Server

The Showdown

Plain 'ol Rack

```
class MyApp
  def call(env)
    headers = { "Content-Type" => "text/html" }
    [200, headers, ["Hello world!"]]
  end
end
```

Sinatra

```
get '/' do
  "Hello world!"
end
```

Open “simple_sinatra” and edit the “config.ru” file.

```
# config.ru
require 'sinatra'

get '/' do
  "Hello Sinatra!"
end

run Sinatra::Application
```

```
$ rackup config.ru
```

What else can  do?

```ruby
# config.ru
require 'sinatra'

get '/' do
  "Hello Sinatra!"
end

get '/hello/:name' do |n|
  "Hello #{n}!"
end

get '/wildcard/*' do
  request.inspect
end

error 404 do
  "OMG, 404!"
end

get '/breakit' do
  500
end

run Sinatra::Application
```


ERMAHGERD
Potato Buds
RUBY ON RAILS
memegenerator.net

Rails

"A full-stack web application framework."

email

models

database

views

ajax

routes

sessions

orm

tasks

controllers

csrf

logging

tests

Your App

Sinatra

Rack

Middleware

Web Server

Your App

Rails

Rack

Middleware

Web Server

Your App

Your App

RAILS

Sinatra

Rack

Middleware

Rack

Middleware

Web Server

Web Server

Vacation rentals, private roo

https://www.airbnb.com/rooms/13979

airbnb

Where are you going?

BROWSE

SIGN UP

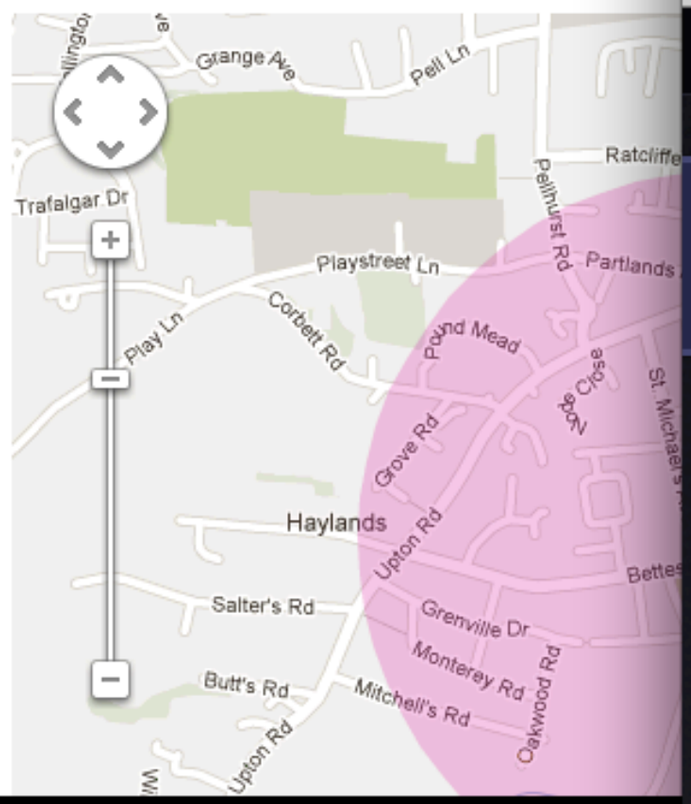
LOG I

Tweet Pin it +1 Like 3

Airstream B&B by Vintage Vacations

Camper/RV - Entire home/apt · Ashey Rd, Ryde, PO33 2UT, United Kingdom

Photos | Maps | Street View



Feedback

Heroku Status

https://status.heroku.com

Subscribe to Notifications

heroku status

current status and incident report

Production

99.92% October uptime

99.95% August - October

Development

98.47% October uptime

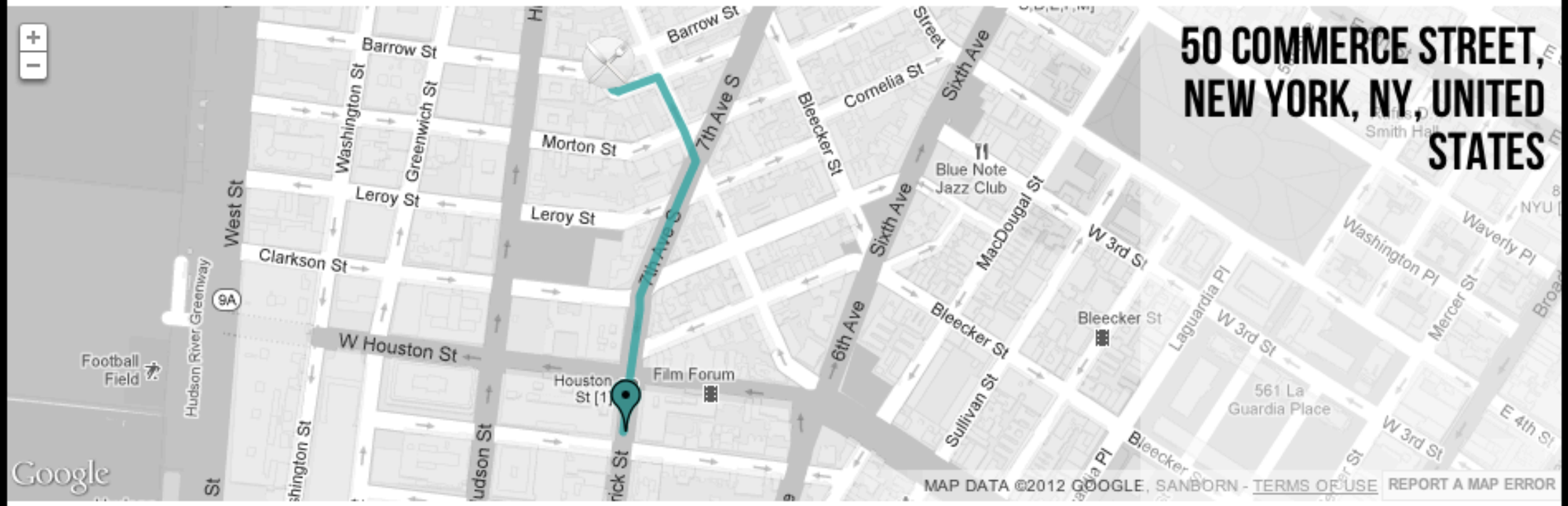
99.13% August - October

Production Apps Operational	Now	Development Apps & Tools Operational

WHERE THE FUCK SHOULD I

wherethefuckshouldigotoeat.com

WHY DON'T YOU FUCKING GO TO COMMERCE



MY LOCATION IS FUCKING WRONG

NO, THAT PLACE LOOKS LIKE SHIT

ACTUALLY, I'M FUCKING THIRSTY

Available on the App Store

@WTFSIGTE

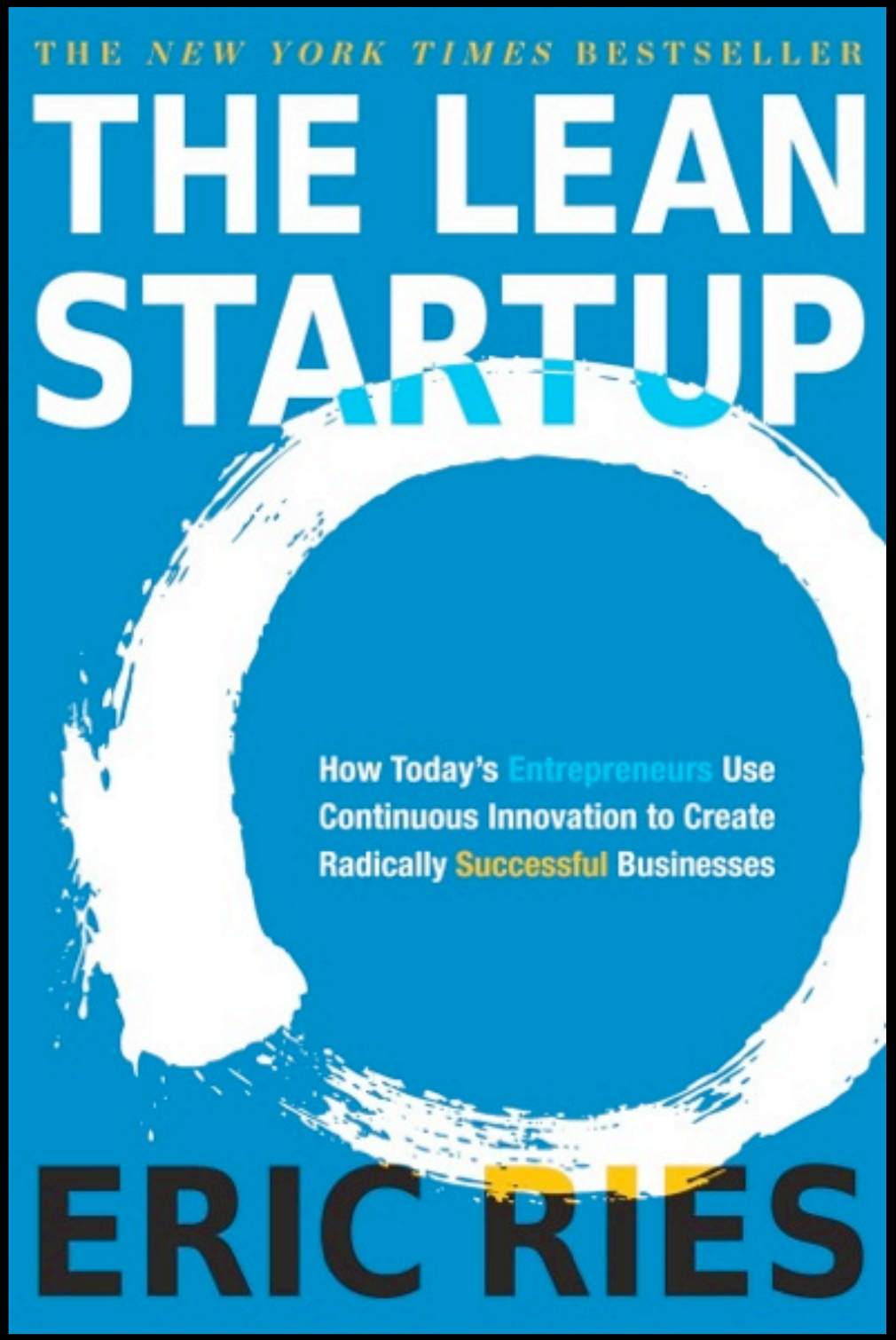
CREATED BY

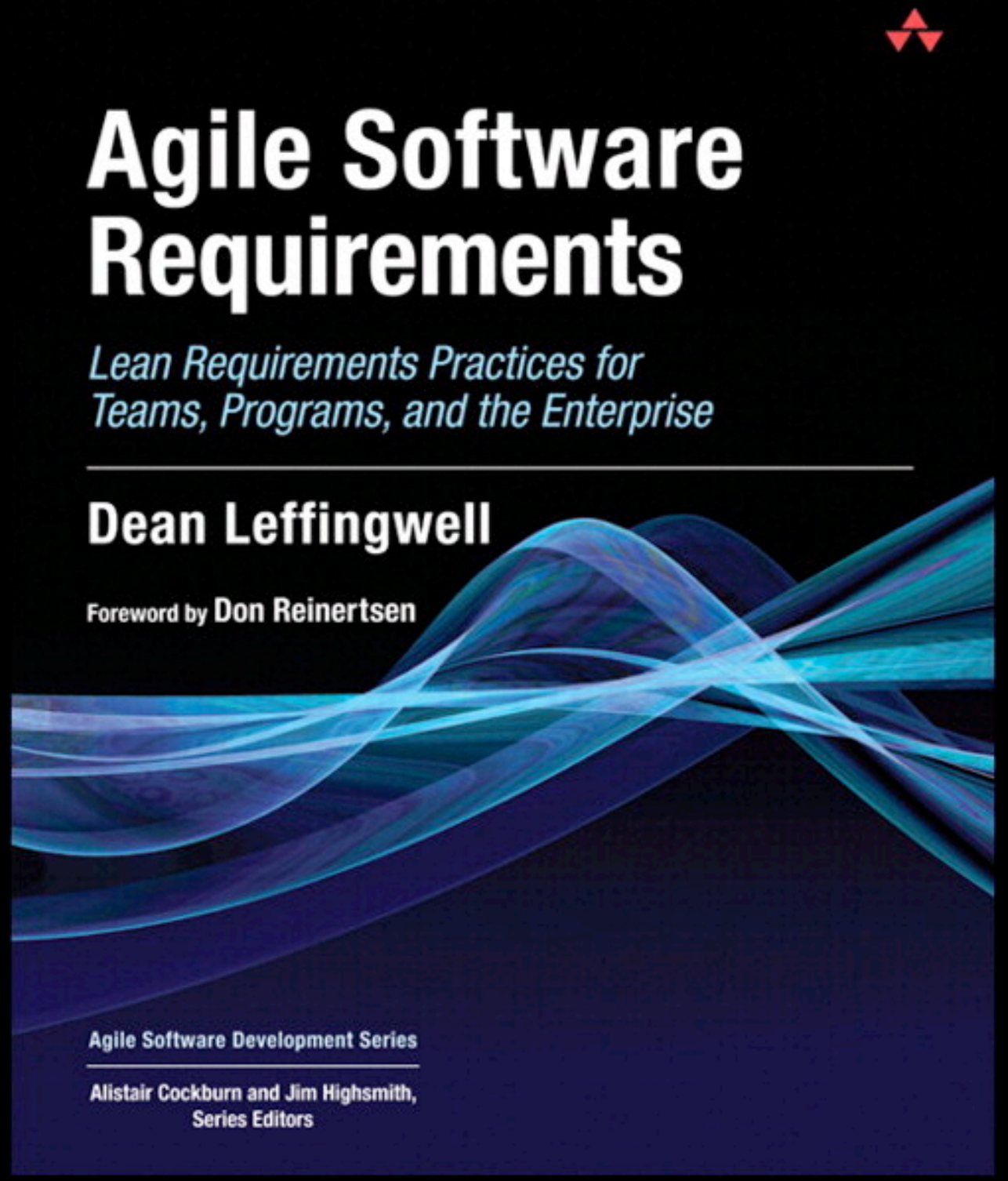
Like 56k

Tweet 2,254

+1 522

me


THE NEW YORK TIMES BESTSELLER
THE LEAN STARTUP
How Today's Entrepreneurs Use Continuous Innovation to Create Radically Successful Businesses
ERIC RIES


Agile Software Requirements
Lean Requirements Practices for Teams, Programs, and the Enterprise
Dean Leffingwell
Foreword by Don Reinertsen
Agile Software Development Series
Alistair Cockburn and Jim Highsmith, Series Editors

OVERBILT F-650

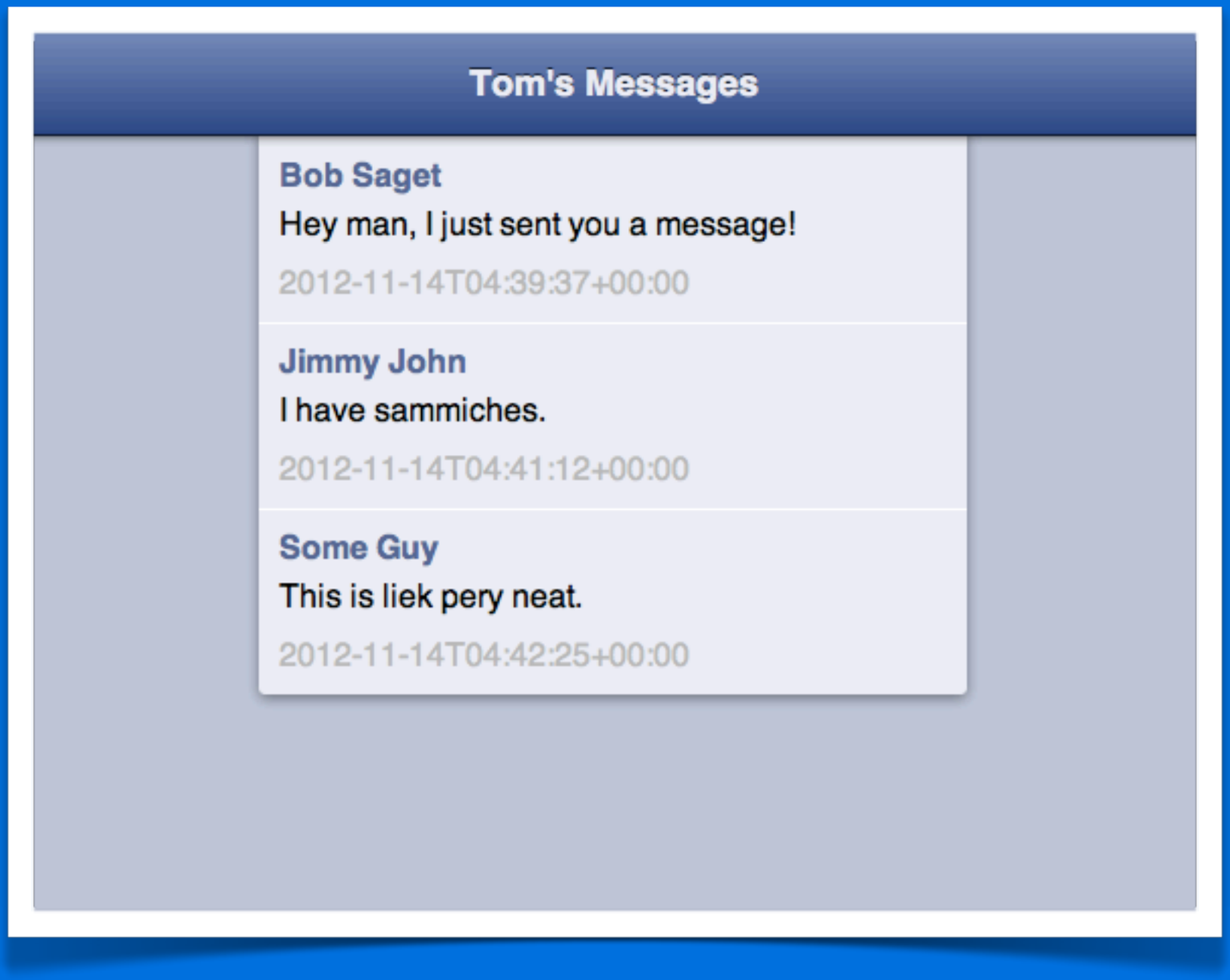
DIESELPOWER

THE VOICE OF THE TURBODIESEL ENTHUSIAST

dieselpowermag.com

Messages App

Rack + Sinatra + DataMapper + Heroku



Thanks!

@blacktm