

Front End Performance

Ralph von der Heyden

/me

- Ruby on Rails developer
- Studied computer science and business administration
- Working at XING (<http://xing.com>)

Front end performance matters...

Why?

For website operators

For website operators
Fast websites

For website operators

Happy customers

For website operators

More customers

For website operators

More money

For web consultants

For web consultants

Easy to learn

For web consultants

Big effect

For web consultants

Extended
portfolio

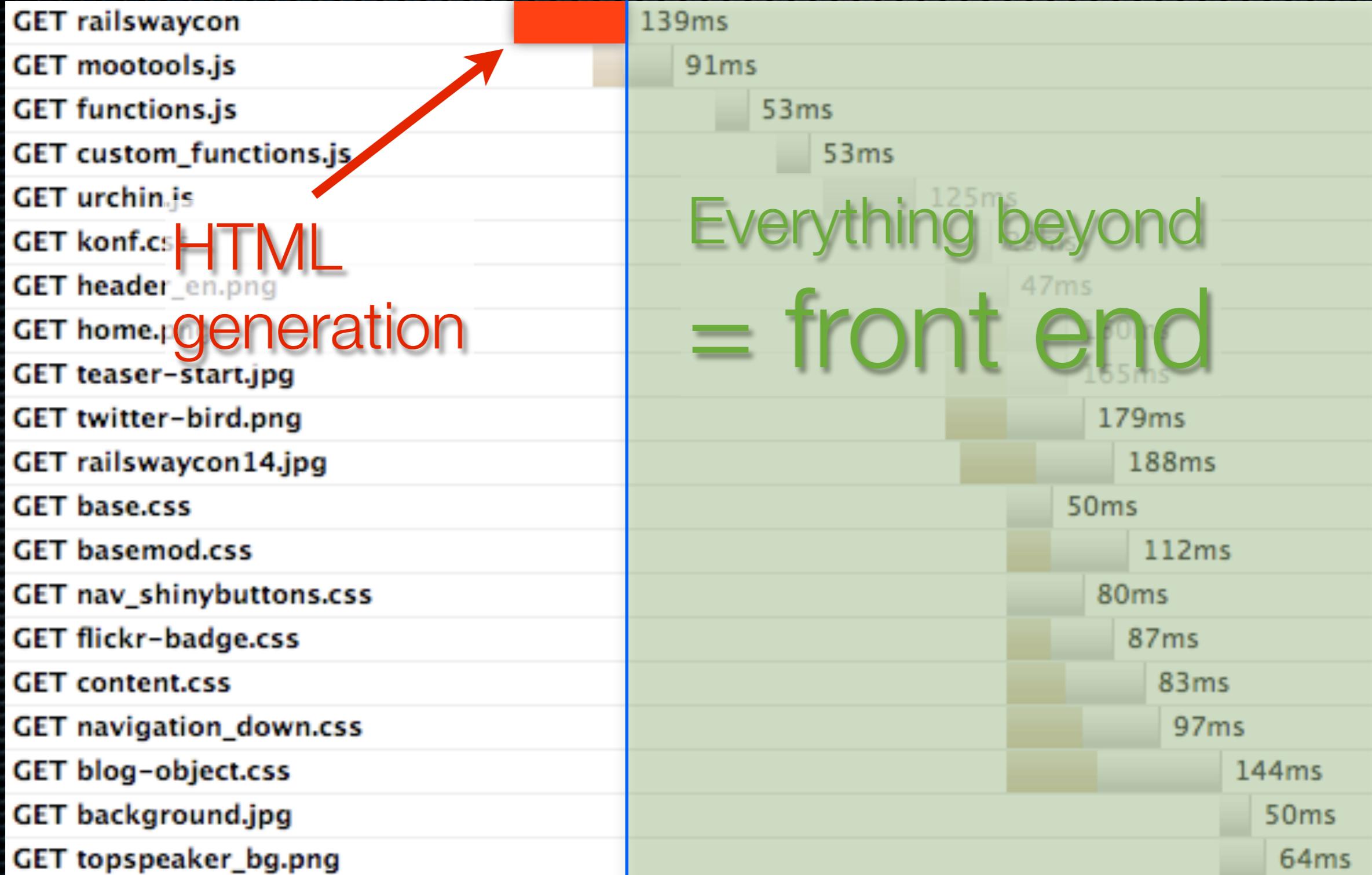
For web consultants

More consulting
jobs

Facts: The impact of user perceived performance

- Amazon: Sells 1% less stuff per 100ms more loading time
- Google: 20% less traffic on the search page after increasing the load time by 500ms

What is the front end?



But I can tune the back end!

- Back end: 10% of loading time
- Front end: 90% of loading time
- Assume you could double the speed of either one

But I can tune the back end!

- Back end: 10% of loading time **3s**
- Front end: 90% of loading time **3s**
- Assume you could double the speed of either one

But I can tune the back end!

- Back end: 10% of loading time **3s** → **2.85s**
- Front end: 90% of loading time **3s**
- Assume you could double the speed of either one

But I can tune the back end!

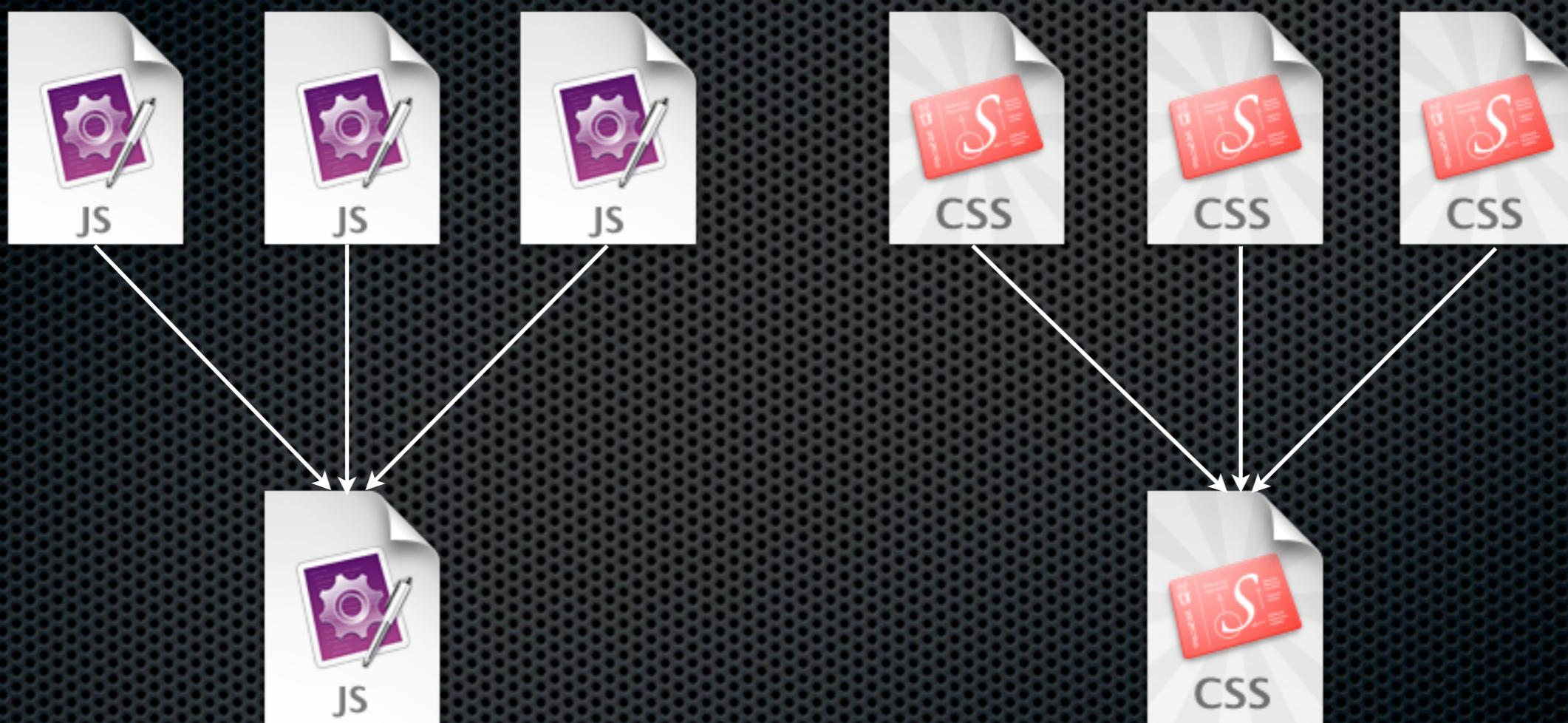
- Back end: 10% of loading time **3s** → **2.85s**
- Front end: 90% of loading time **3s** → **1.65s**
- Assume you could double the speed of either one

How to do it?

- Fewer Requests
- Smaller Requests
- Speed up Requests

Fewer Requests

Fewer Requests: Join CSS and JavaScripts



Fewer Requests: Join CSS and JavaScripts

- `<%= stylesheet_link_tag ['layout', 'pagination'], :media => "screen, projection", :cache => "style" %>`
- `<%= javascript_include_tag(['prototype', 'application'], :cache => true) %>`
- Only 1 GET request in production
- RailsWayCon Startpage: 5 JS files, 8 CSS files

| |
|---|
|  js (5) |
|  css (8) |

Fewer Requests: Join CSS and JavaScripts

- `<%= stylesheet_link_tag ['layout', 'pagination'], :media => "screen, projection", :cache => "style" %>`
- `<%= javascript_include_tag(['prototype', 'application'], :cache => true) %>`
- Only 1 GET request in production
- RailsWayCon Startpage: ~~JS~~ JS files, ~~CSS~~ CSS files

↓ ↓
2 1

| | |
|--|----------------|
| | js (5) |
| | css (8) |

Fewer Requests: Join CSS and JavaScripts

- ```
<%= stylesheet_link_tag ['layout',
'pagination'], :media => "screen",
projection", :cache => true) %>
```
- ```
<%= javascript_include_tag 'application',  
:cache => true) %>
```
- ```
st_in_production do
```
- RateWayCon Startpage: ~~JS files, CSS files~~

10



2



1

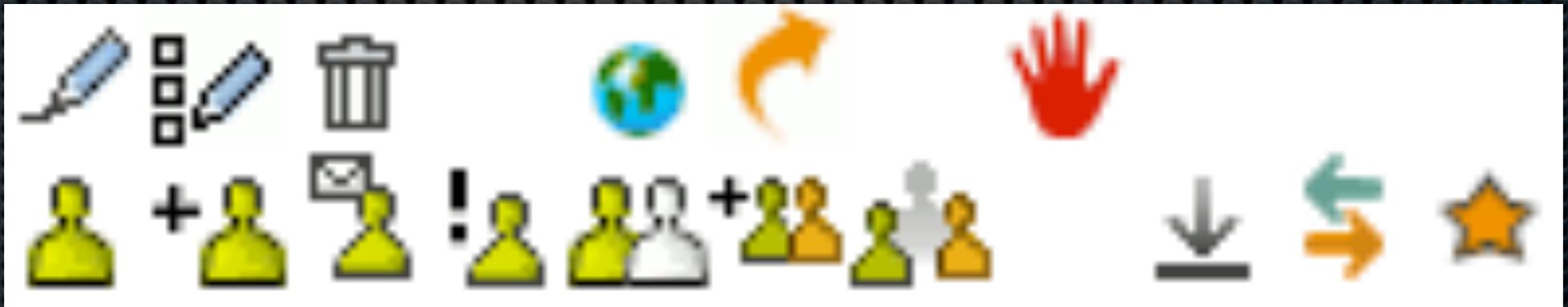
|  |         |
|--|---------|
|  | js (5)  |
|  | css (8) |

# Fewer Requests: Sprite your images

- Background image
- Position
- Size
- RailsWayCon: more than 30 images < 5K

# Fewer Requests: Sprite your images

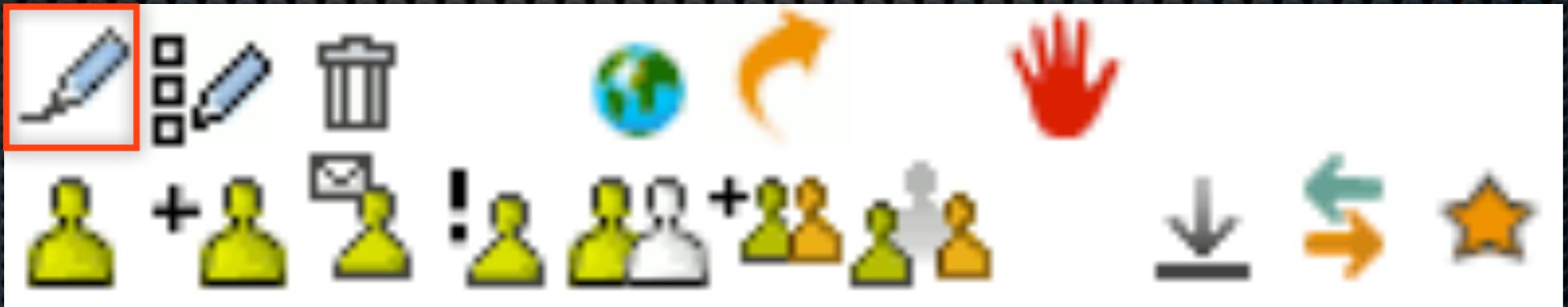
- Background image
- Position
- Size



- RailsWayCon: more than 30 images < 5K

# Fewer Requests: Sprite your images

- Background image
- Position
- Size



- RailsWayCon: more than 30 images < 5K

# Fewer Requests: Sprite your images

- Background image
- Position
- Size



- RailsWayCon: more than 30 images < 5K

# Fewer Requests: Sprite your images

- Background image
- Position
- Size



- RailsWayCon: more than 30 images < 5K

|     |              |
|-----|--------------|
| [+] | cssimage (5) |
| [+] | image (53)   |

# Fewer Requests: Sprite your images

- Background image
- Position
- Size



1 Sprite

- RailsWayCon: more than ~~30 images~~ < 5K

|     |              |
|-----|--------------|
| [+] | cssimage (5) |
| [+] | image (53)   |

# Fewer Requests: Sprite your images

- Background image
- Position

- Size

30 requests saved!



1 Sprite

- RailsWayCon: more than ~~30 images < 5K~~

|                   |                              |
|-------------------|------------------------------|
| <a href="#">+</a> | <a href="#">cssimage (5)</a> |
| <a href="#">+</a> | <a href="#">image (53)</a>   |

# Fewer Requests: Far future expires header

- Static files (usually) don't change

# Fewer Requests: Far future expires header

- Static files (usually) don't change

C

# Fewer Requests: Far future expires header

- Static files (usually) don't change

C

S

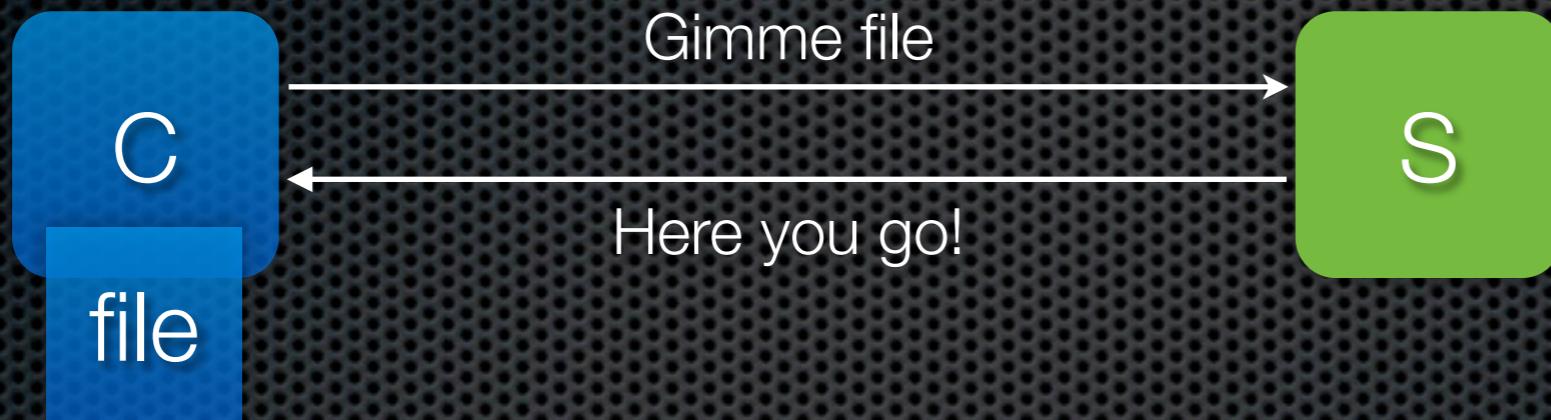
# Fewer Requests: Far future expires header

- Static files (usually) don't change



# Fewer Requests: Far future expires header

- Static files (usually) don't change



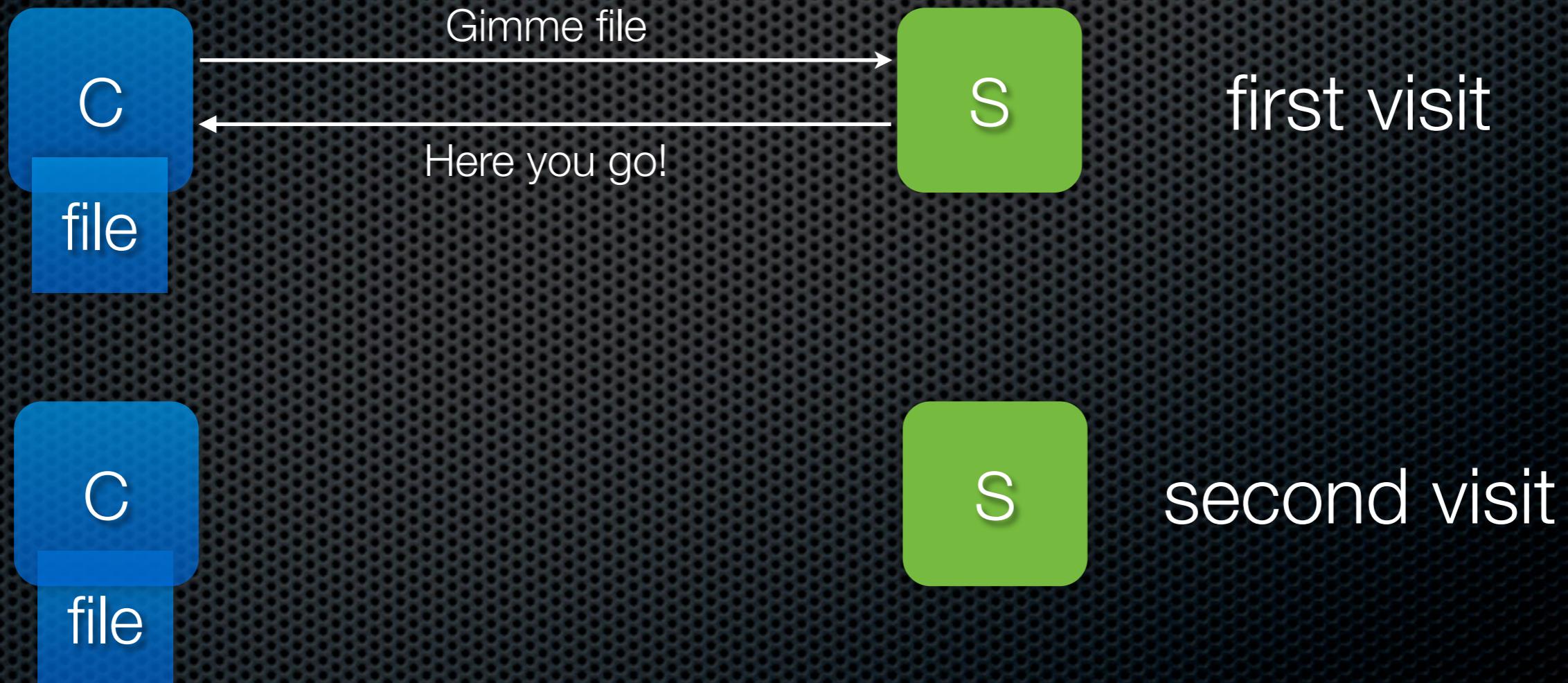
# Fewer Requests: Far future expires header

- Static files (usually) don't change



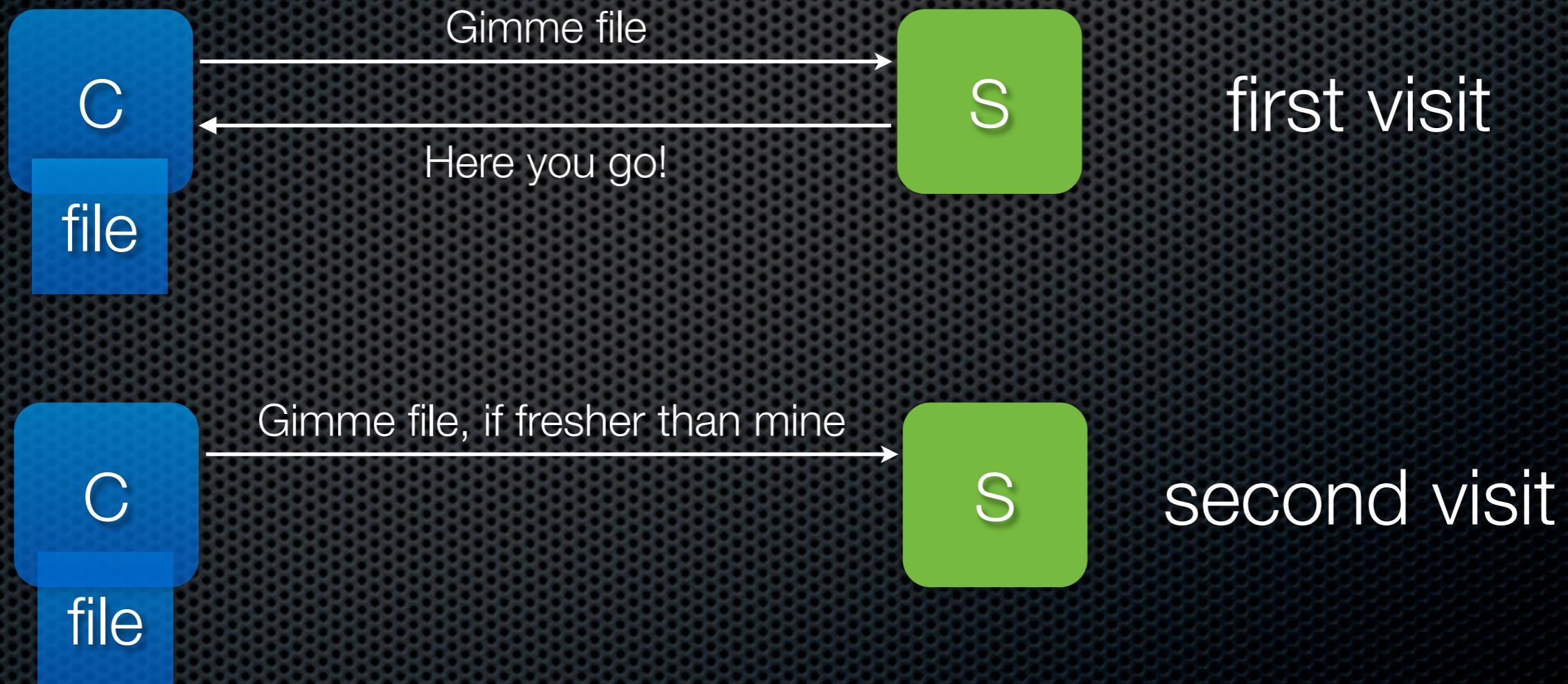
# Fewer Requests: Far future expires header

- Static files (usually) don't change



# Fewer Requests: Far future expires header

- Static files (usually) don't change



# Fewer Requests: Far future expires header

- Static files (usually) don't change



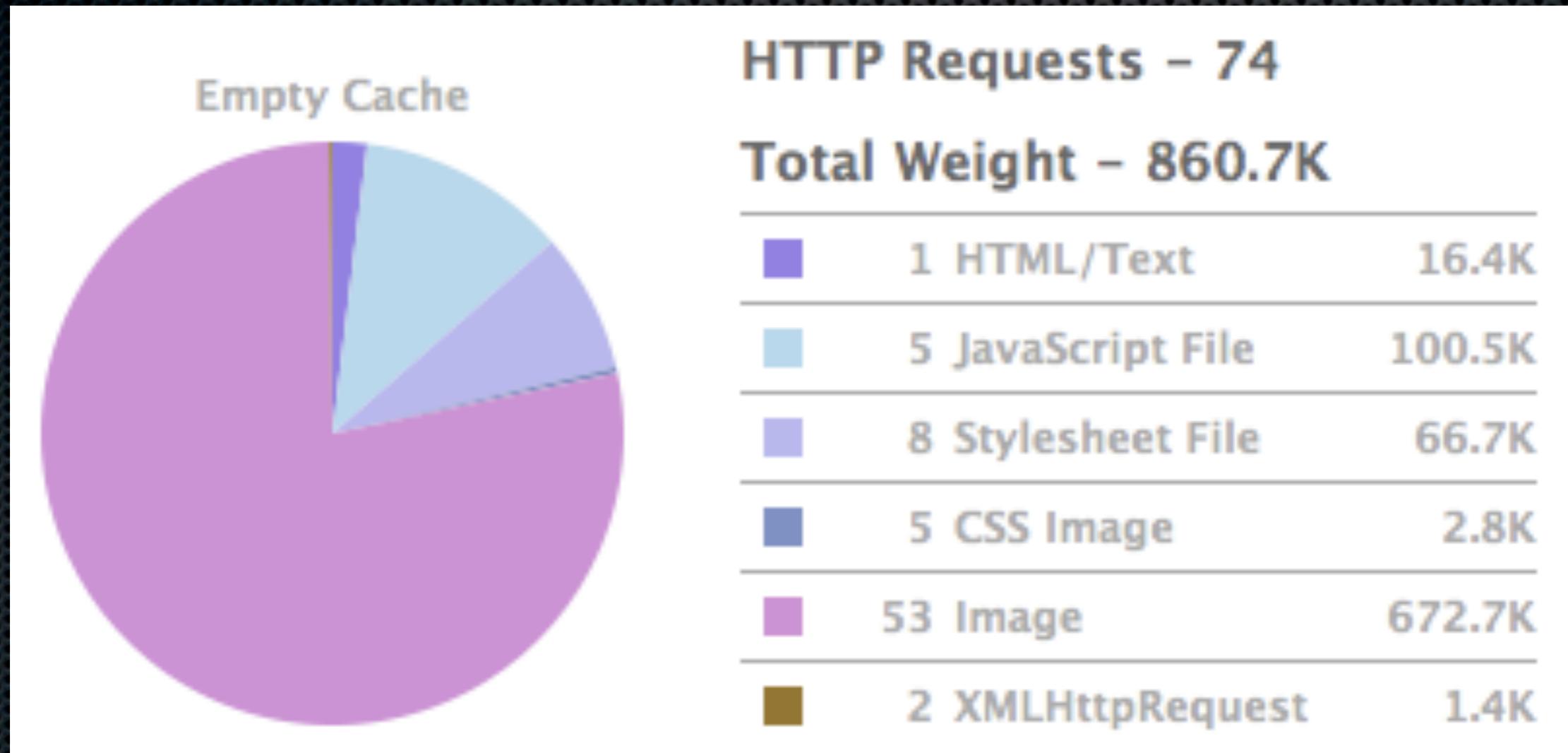
# Fewer Requests: Far future expires header

- Conditional GETs are plugging up the queue
- Avoid Conditional GETs altogether
  - Expires: Thu, 27 May 2010 20:00:00 GMT
  - Cache-Control: max-age=31536000
  - <filesmatch "\.(jpg|gif|png|css|js)\$">  
  ExpiresActive on  
  ExpiresDefault "access plus 1 year"  
</filesmatch>

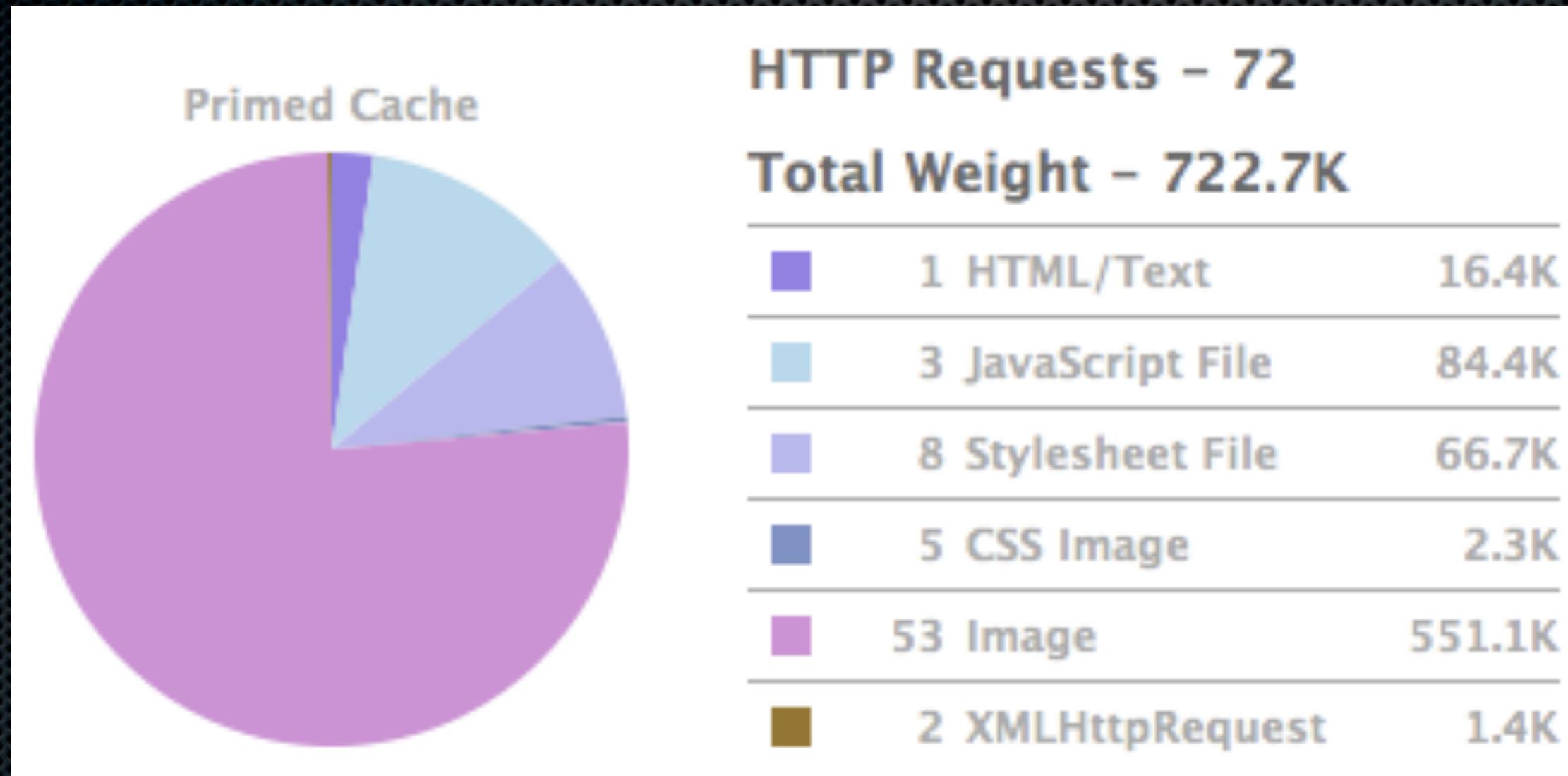
# Fewer Requests: Rewving filenames

- Reflect change in the URL
- `File.mtime("my.js").to_i_to_s` ⇒ `/javascripts/my.js?1242765640`
- Also:
  - Be careful with CSS images
  - Disable ETags for static files!

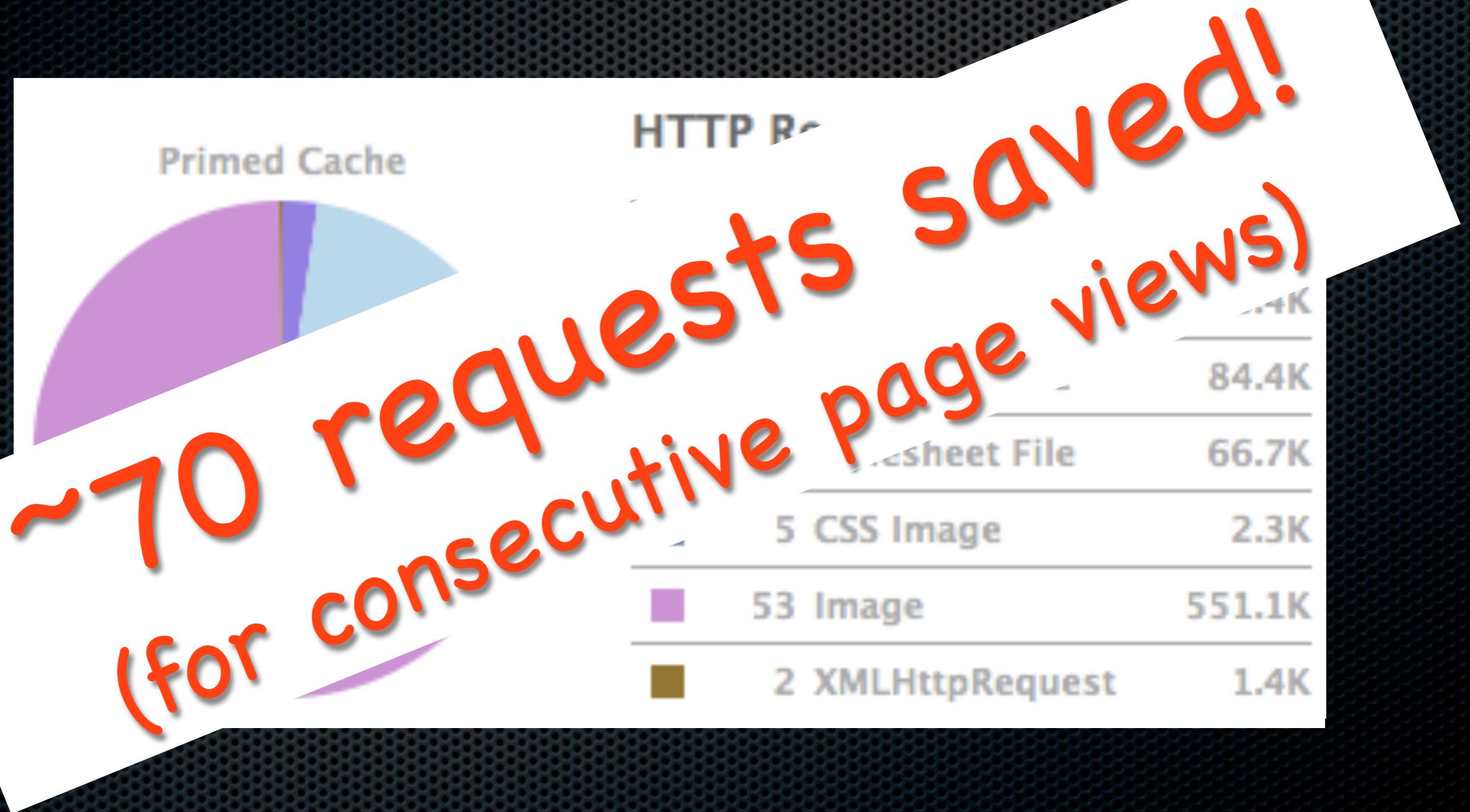
# Fewer Requests: Far future expires header



# Fewer Requests: Far future expires header



# Fewer Requests: Far future expires header



# Smaller Requests

# Smaller Requests: gzip

- Enable gzipping for all text files
- Apache: mod\_deflate
- RailsWayCon: without gzip 169K, with gzip 54K

**F Compress components with gzip**

# Smaller Requests: gzip

- Enable gzipping for all text files
- Apache: mod\_deflate
- RailsWebkit

**68% savings  
for text files**

**express components with gzip**

# Smaller Requests: Minify your JS and CSS

- Strip comments, whitespace etc.
- Rails: asset\_packager plugin
- RailsWayCon: 151K -> 108K

# Smaller Requests: Minify your JS and CSS

- Strip comments, whitespace
- Rails: asset\_packer
- RailsWax

28.5% savings  
for CSS and JS

# Smaller Requests: Know your images

- PNG is usually smaller than GIF when optimized\*
- Strip JPG meta data and thumbnails\*
- Use smush.it (<http://smush.it>) to show optimization potential

\*Mac users: PNGpong, ImageOptim

# Smaller Requests: Know your images

Smushed **20.66%** or **58.27 KB** from the size of your image(s). How did we do it? See the table below for more details.

- Total page weight: 331 KB
- Images: 286 KB

# Smaller Requests: Know your images

Smushed 20.66% or 58.27 KB  
your image(s). How did we do?  
for more details.

- Total page

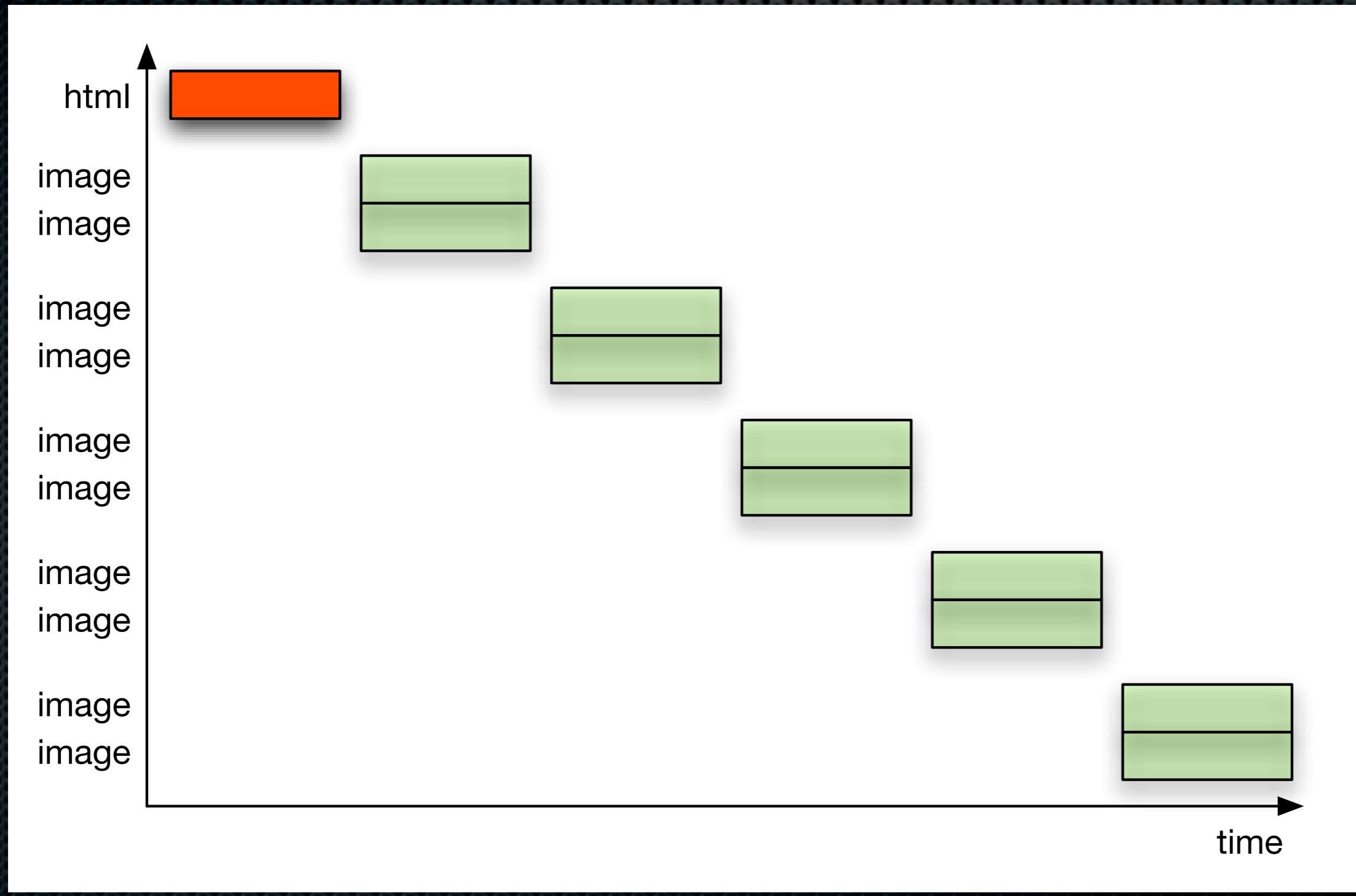
17.6% savings  
for total page

# Speed up Requests

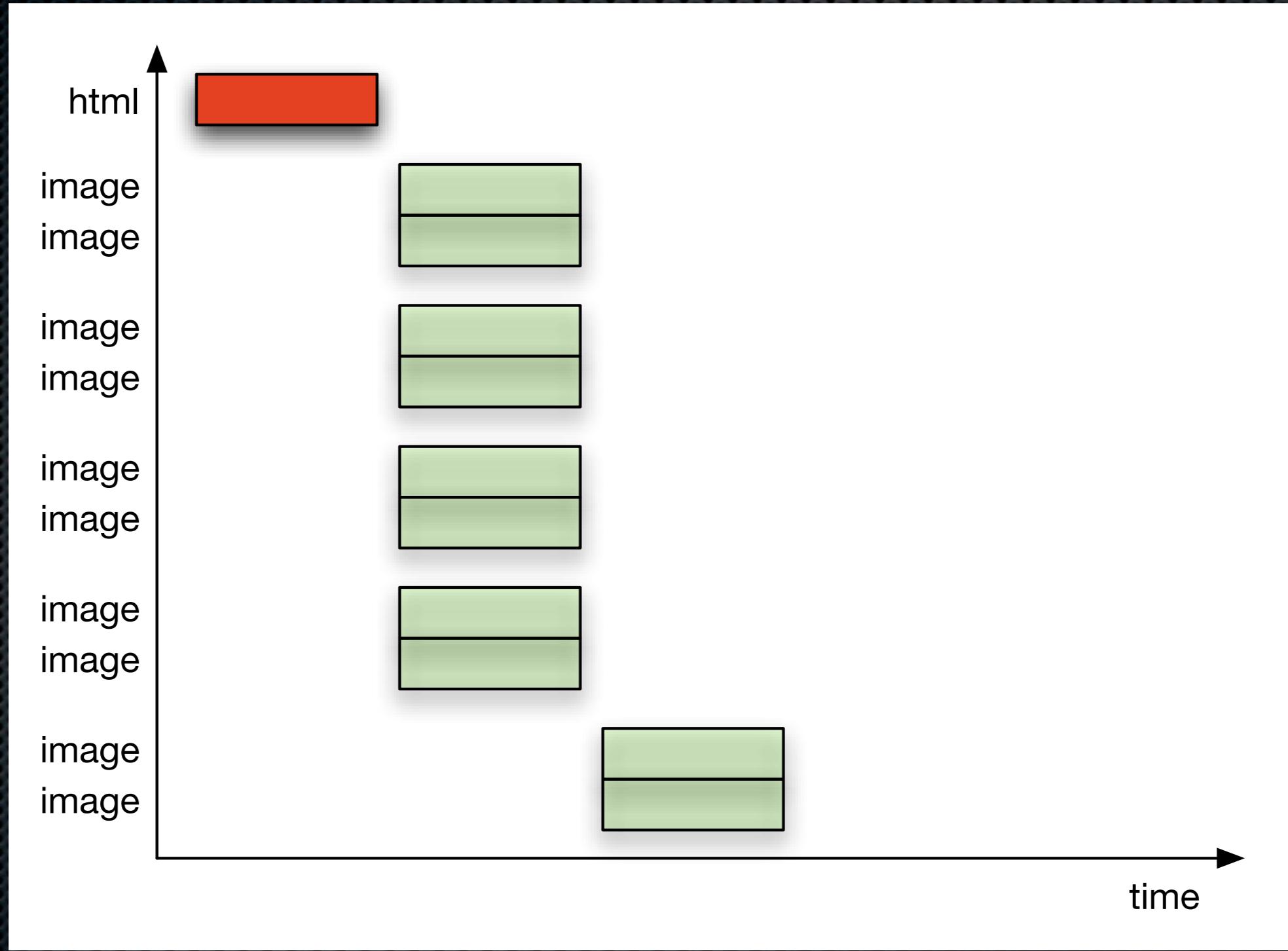
# Speed up requests: Domain sharding

- Most browsers load 2 files per host
- Multiple domains for static content (CNAMEs)
- Rails: Set 4 asset hosts ⇒ 8 parallel downloads
- `config.action_controller.asset_host`  
`= "http://assets%d.example.com"`

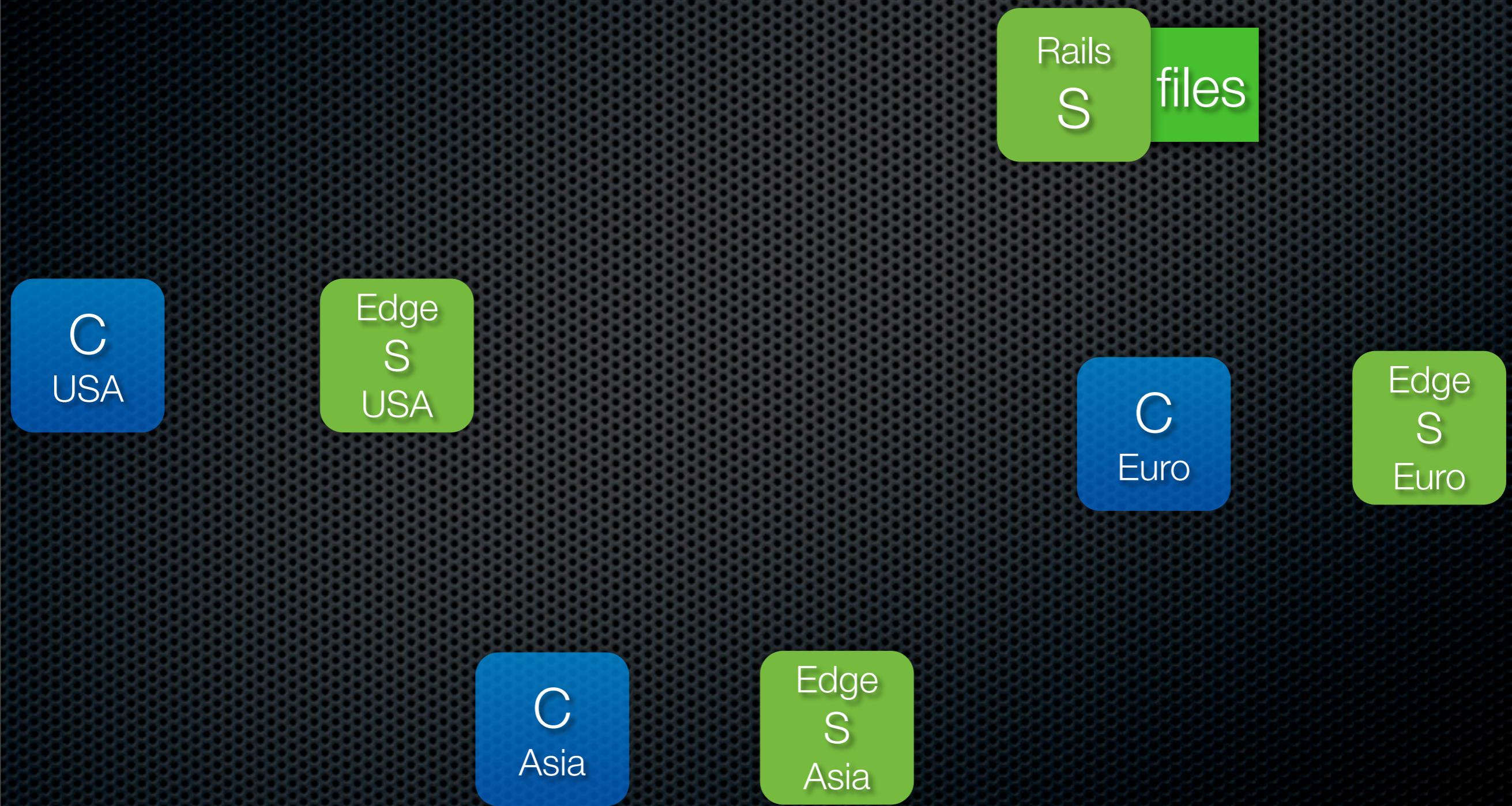
# Speed up requests: Domain sharding



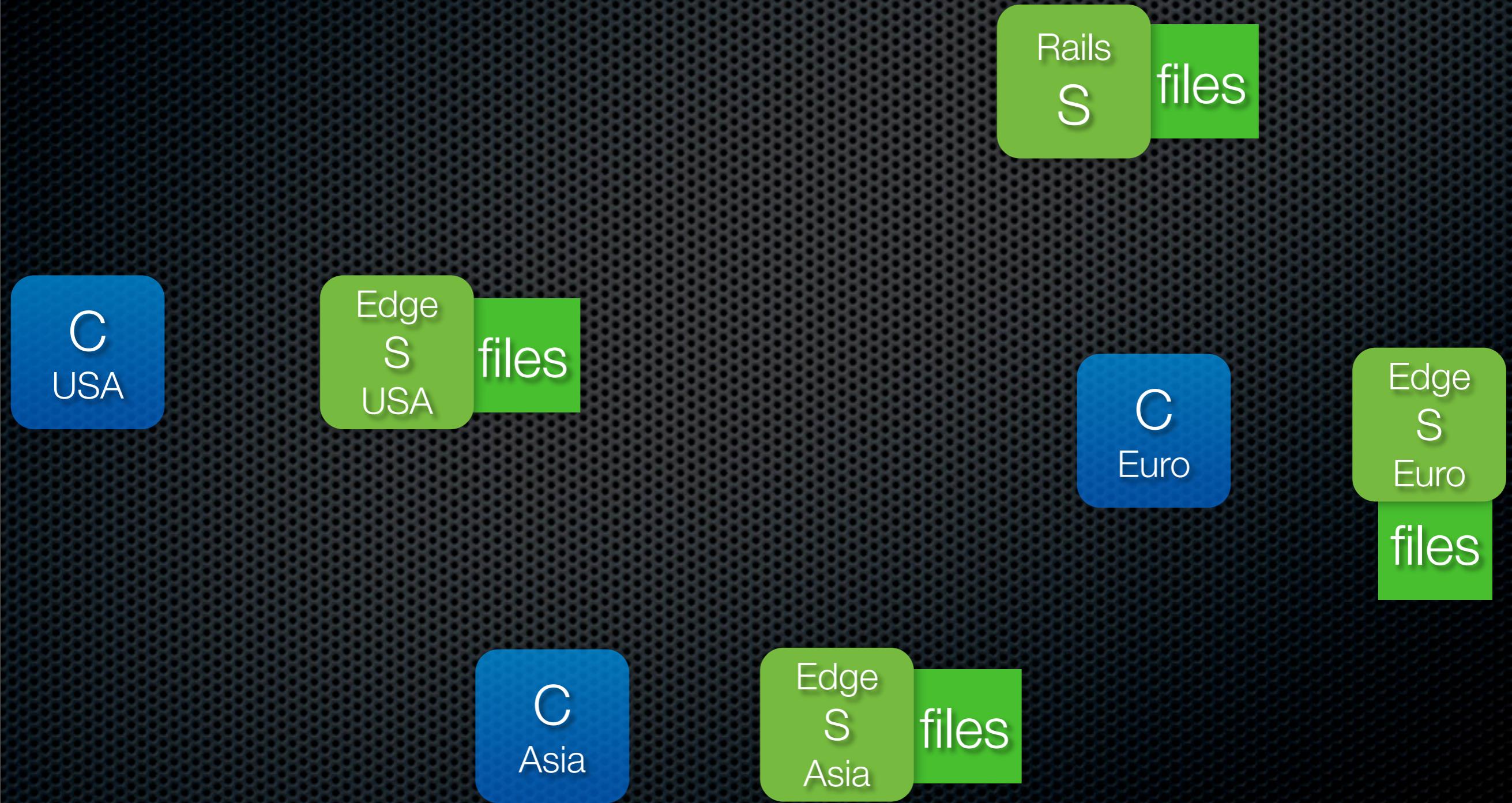
# Speed up requests: Domain sharding



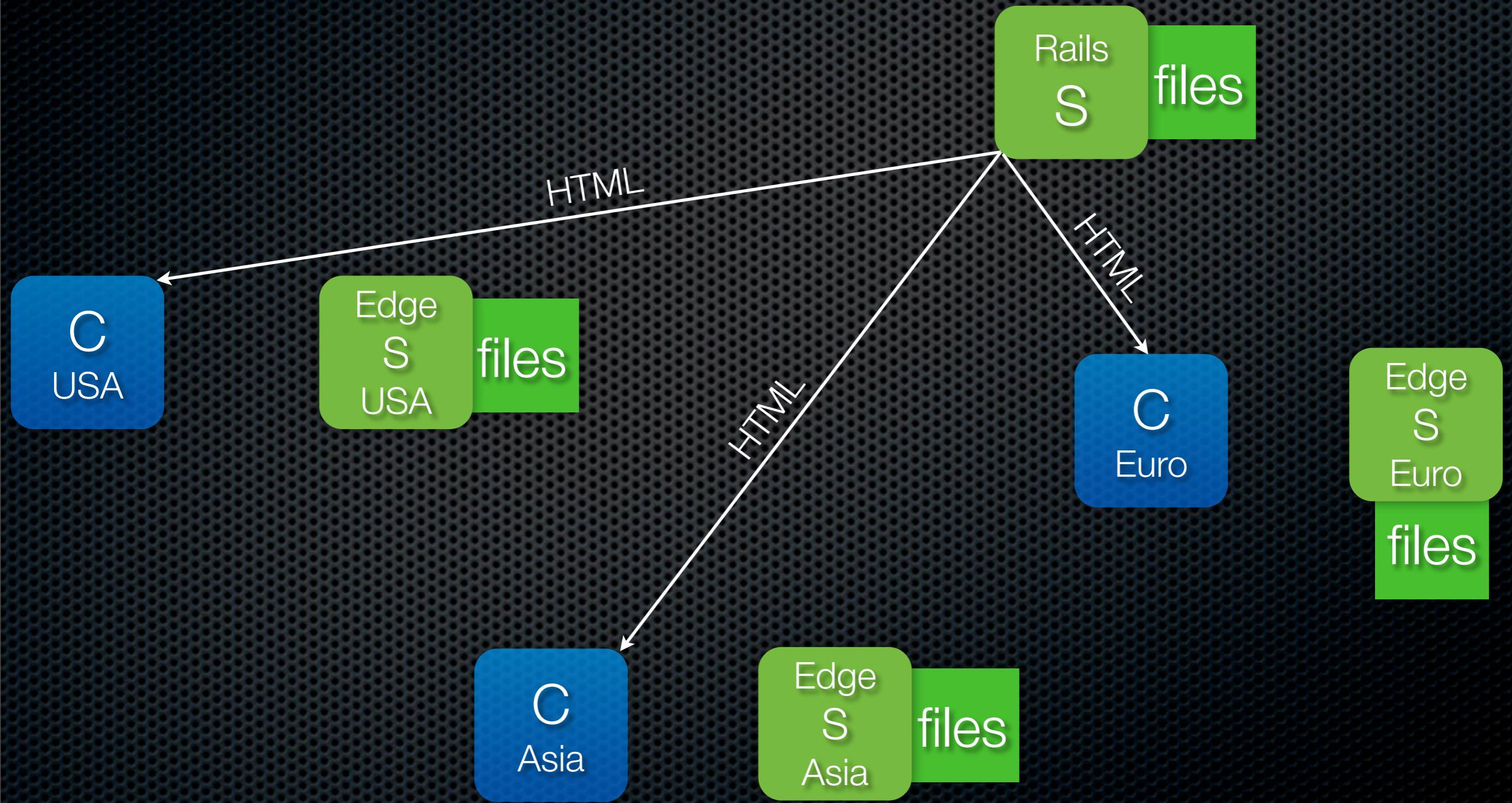
# Speed up requests: Content Delivery Network



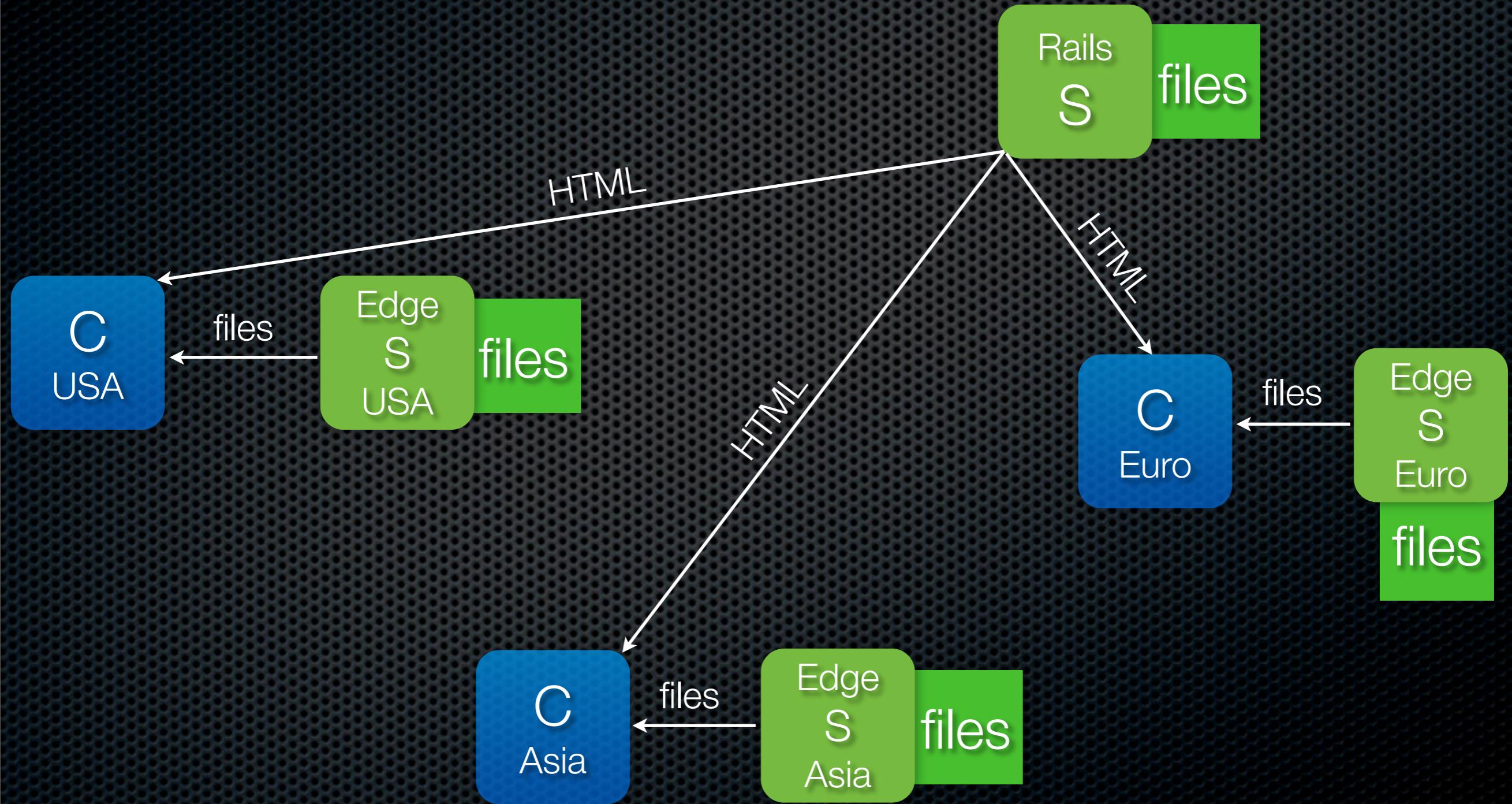
# Speed up requests: Content Delivery Network



# Speed up requests: Content Delivery Network



# Speed up requests: Content Delivery Network



Weapon of  
choice

# Weapon of choice: Firebug and YSlow

- Performance Grade
- Shows your weak spots
- Hints how to improve
- Other performance related tools

Grade **D**

Overall performance score 62

Ruleset applied: YSlow(V2)

# Weapon of choice: Firebug and YSlow

The screenshot shows a web browser window with the following details:

- Address Bar:** Startpage | RailswayCon – The conference on Ruby, Rails and more...  
http://it-republik.de/conferences/railswaycon/
- Page Content:** RailswayCon logo, event date May 25 - 27, 2009, location Berliner Congress Center, Alexanderplatz, Berlin.
- Navigation Bar:** Home, Conference, Program, Registration, Partner, Contact.
- Social Sharing:** Twitter, Facebook, LinkedIn, etc.
- Call-to-Action:** Register now!
- Performance Analysis (Firebug/YSlow):**
  - Grade:** D (Overall performance score 62)
  - Ruleset applied:** YSlow(V2)
  - URL:** http://it-republik.de/conferences/railswaycon/
  - Filters:** ALL (22), FILTER BY: CONTENT (6), COOKIE (2), CSS (6), IMAGES (2), JAVASCRIPT (4), SERVER (5)
  - Grade D on Make fewer HTTP requests:**
    - This page has 5 external Javascript scripts. Try combining them into one.
    - This page has 8 external stylesheets. Try combining them into one.
    - Decreasing the number of components on a page reduces the number of HTTP requests required to render the page, resulting in faster page loads. Some ways to reduce the number of components include: combine files, combine multiple scripts into one script, combine multiple CSS files into one style sheet, and use CSS Sprites and image maps.

Where to  
start?

# Where to start: Priorities

- Start with optimizations that...
  - Avoid requests
  - Make requests smaller
  - Speed up requests
- the **most!**
- Trade off between ease of deployment and efficiency

# Where to start: Priorities

- Start with optimizations that...
  - Avoid requests **Joining, Sprites, Expires Header**
  - Make requests smaller
  - Speed up requests
- the **most!**
- Trade off between ease of deployment and efficiency

# Where to start: Priorities

- Start with optimizations that...
  - Avoid requests **Joining, Sprites, Expires Header**
  - Make requests smaller **gzip, image compression**
  - Speed up requests
- the **most!**
- Trade off between ease of deployment and efficiency

# Where to start: Priorities

- Start with optimizations that...
  - Avoid requests **Joining, Sprites, Expires Header**
  - Make requests smaller **gzip, image compression**
  - Speed up requests **domain sharding, cdn**
- the **most!**
- Trade off between ease of deployment and efficiency

SSL & Caching

ETags

Redirects

# Questions?

gzipping HARD!

Script loading

Proxies

Twitter @ralph  
<http://rvdh.de>

Thank you for your  
kind attention!

# Fewer Requests: SSL and https

- SSL: Loading http assets from a https page: **Warning!**
- DHH suggests (Google for „37signals mixed content“):
  - config.action\_controller.asset\_host = Proc.new do |source, request|non\_ssl\_host = "http://asset%d.example.com"ssl\_host = "https://asset1.example.com"if request.ssl?casewhen source =~ /\.js\$/ssl\_hostwhen request.headers["USER\_AGENT"] =~ /(Safari)/non\_ssl\_hostwhen request.headers["USER\_AGENT"] =~ /Firefox/ && source =~ /^\/images/non\_ssl\_hostelsessl\_hostendelsenon\_ssl\_hostendend

# Fewer Requests: Avoid redirects

- Redirect to asset: breaks caching
- Redirect to html: Delays loading of assets
- Always add Trailing-Slash to URLs