MIDDLEMAN

The missing view in the Rails API stack

@bradgessler

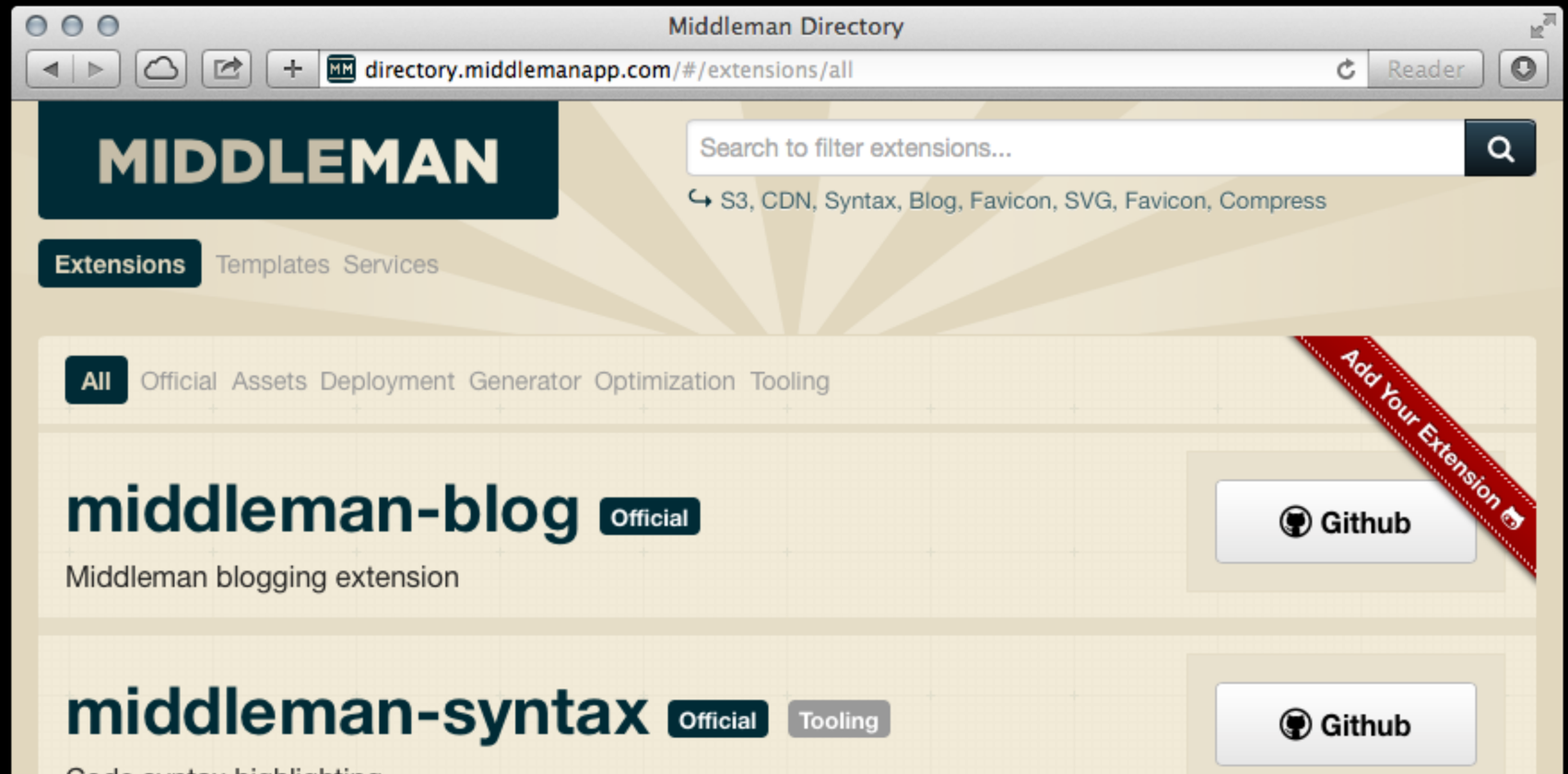
CTO & Cofounder of Poll Everywhere

~~Its like~~

~~**Jekyll**~~

Middleman is
extraordinarily well documented

Middleman is **modular** with a **rich ecosystem of extensions**



```
$ gem install middleman

$ middleman init my-app
       create my-app/.gitignore
       create my-app/config.rb
       create my-app/source/index.html.erb
       create my-app/source/layouts/layout.erb
       create my-app/source/stylesheets
       create my-app/source/javascripts

$ cd my-app ; middleman server
== The Middleman is standing watch at http://0.0.0.0:4567
```

Getting Started with Middleman

Has all of the same **front-end goodies from Rails**, so you'll feel right at home

```
gem 'haml'
gem 'sass'
gem 'sprockets'
gem 'coffeescript'
gem 'compass'
gem 'activesupport'
gem 'susy'
```

Middleman is **multi-environment aware**

```
# ./config.rb
# Local dev server settings
configure :development do
  activate :livereload
end

# Production settings
configure :build do
  activate :minify_css
  activate :minify_javascript
  activate :asset_hash
end
```

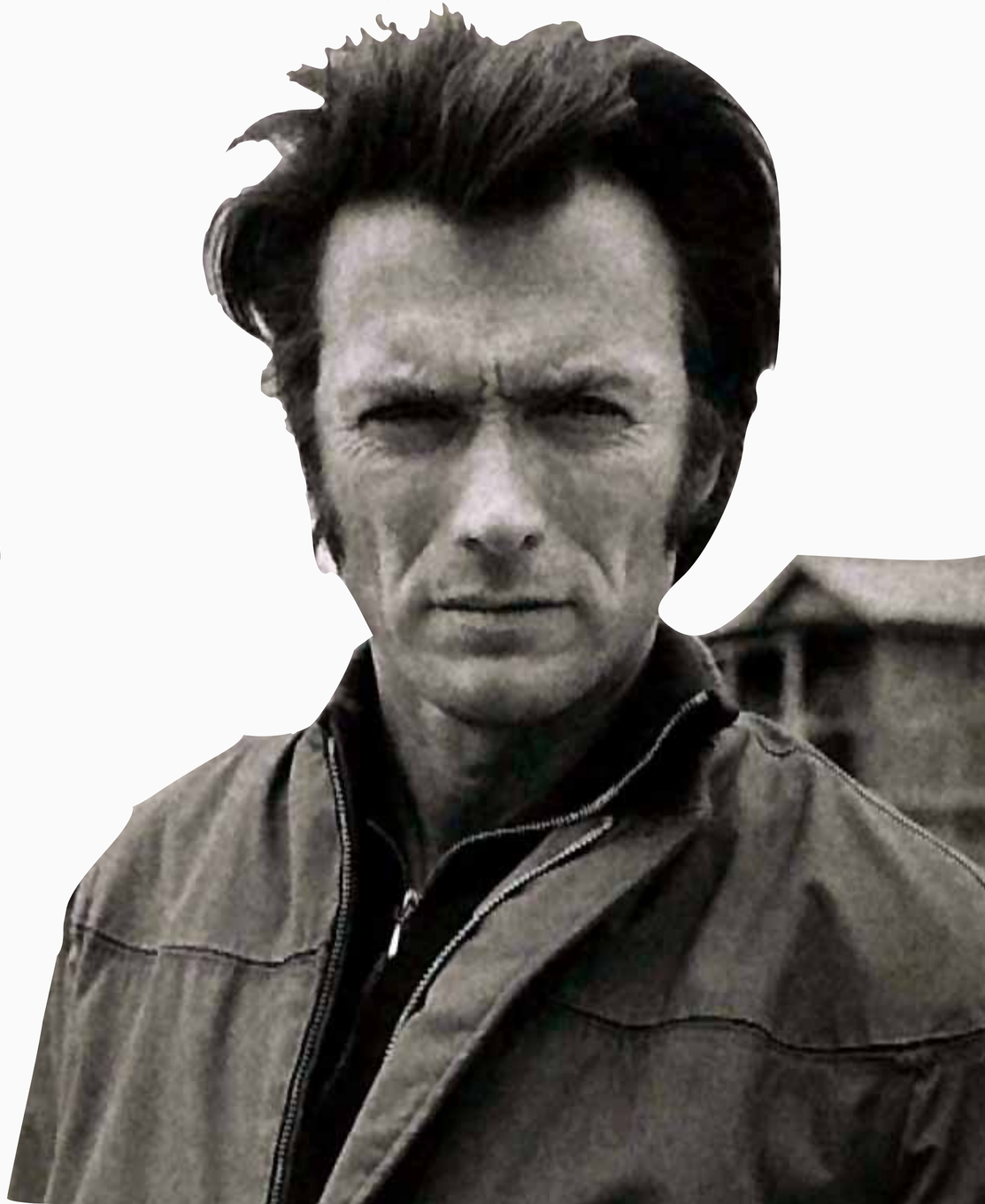
A simple **two step deploy** process

```
task :build do
  sh 'bundle exec middleman build'
end

task :upload do
  sh 'rsync -avz ./build/ deployer@site.com:/www/'
end

task deploy: %w[build upload]
```

Yeah, so what?
Why should I care
about Middleman?




Web apps fall somewhere within
a spectrum of dynacism

Realtime visualizations
Poll Ev graphs

Middleman
JS MVC Library

GUI-oriented applications
Google Spreadsheets

Rails
Less caching

Document-oriented web apps
Invoicing application

Blogging platform
Svbtle, Posthaven

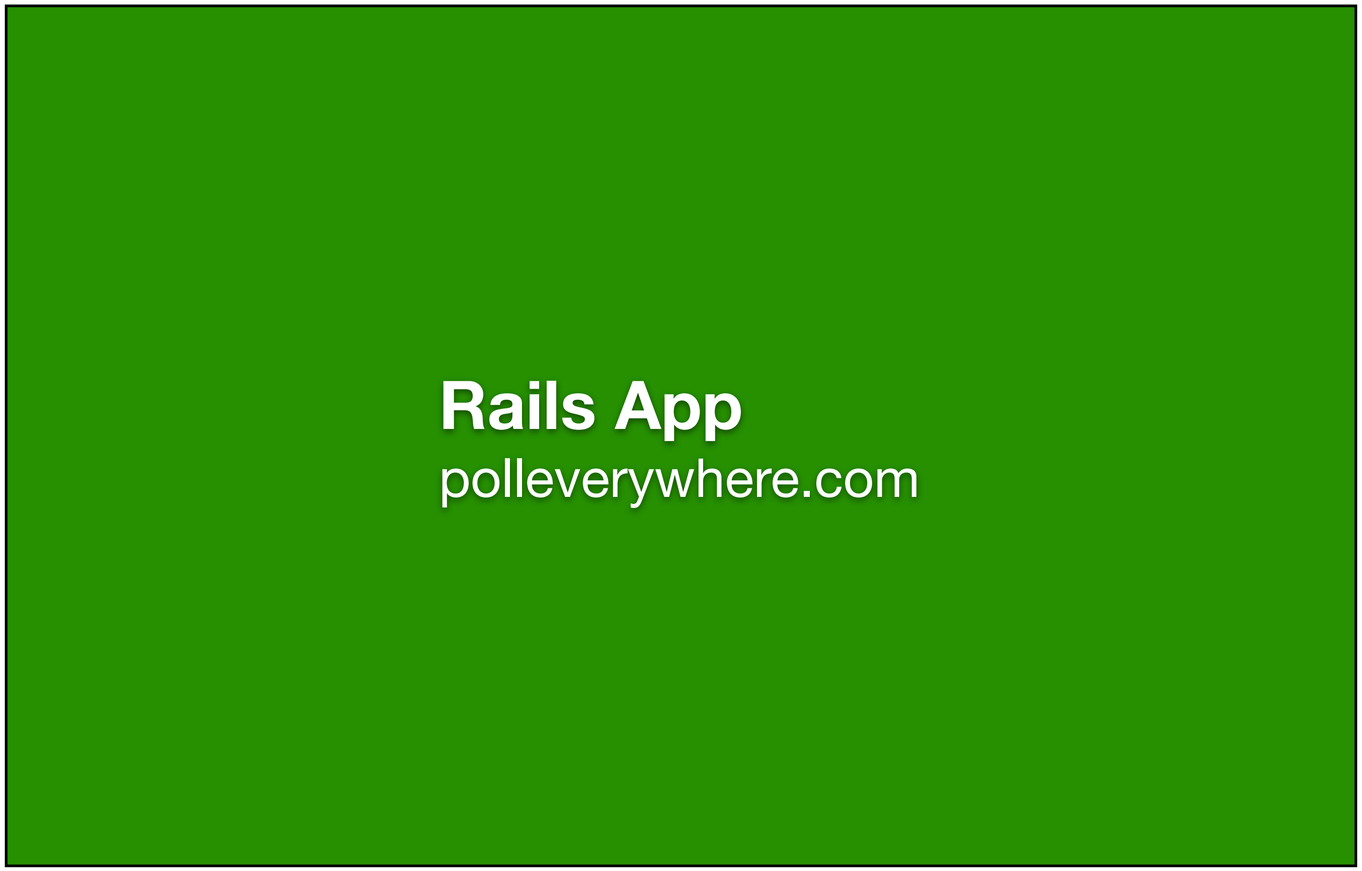
Rails
More caching

Personal blog
bradgessler.com

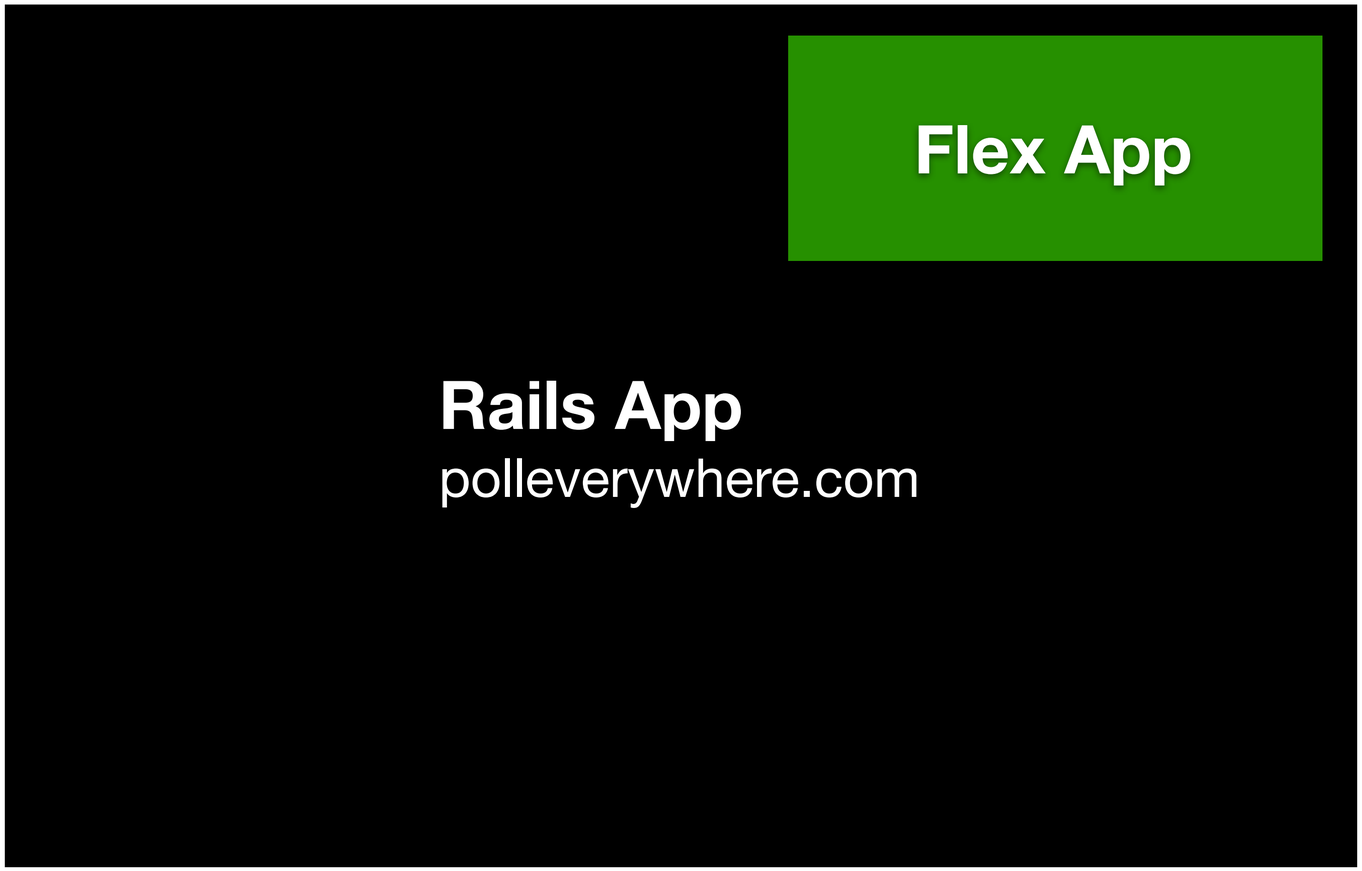
Informational website
Steve's Plumbin' Services

Middleman
HTML content

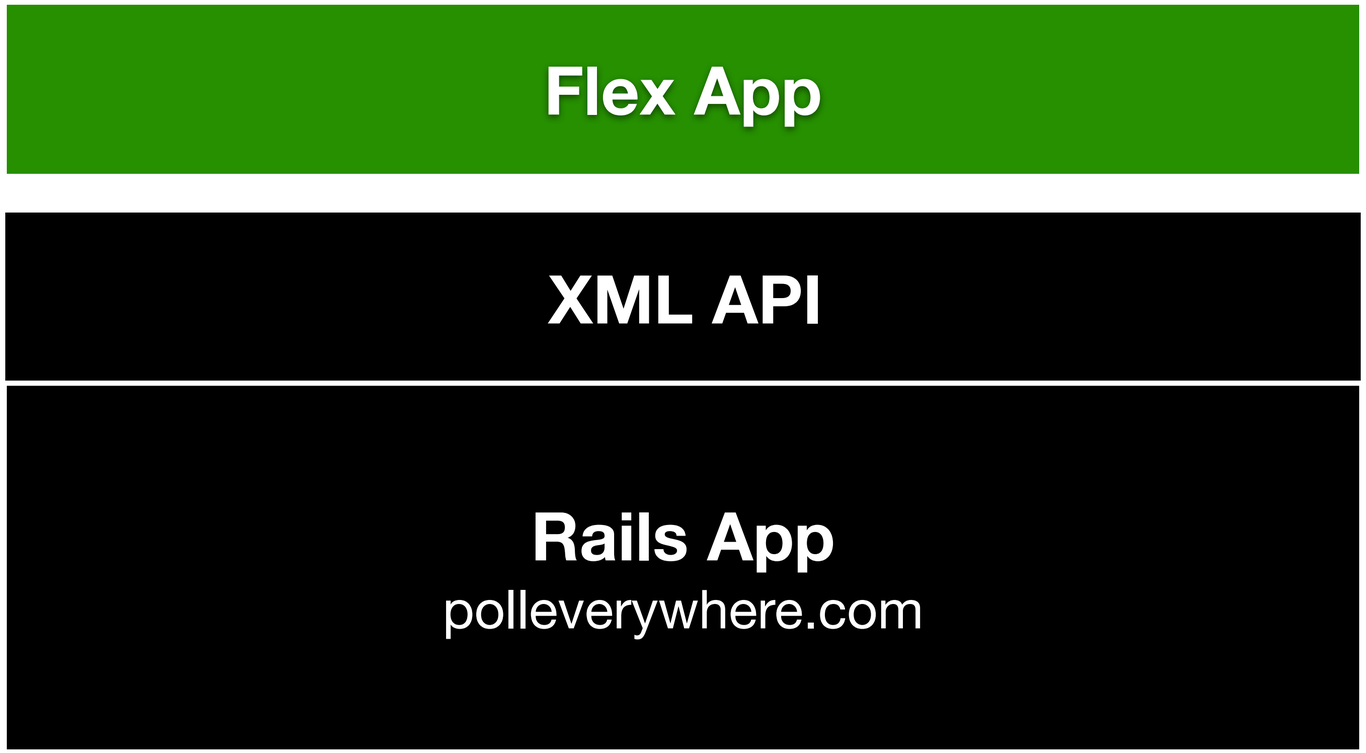
Say hello to the polleverywhere.com **Rails app** circa 2008



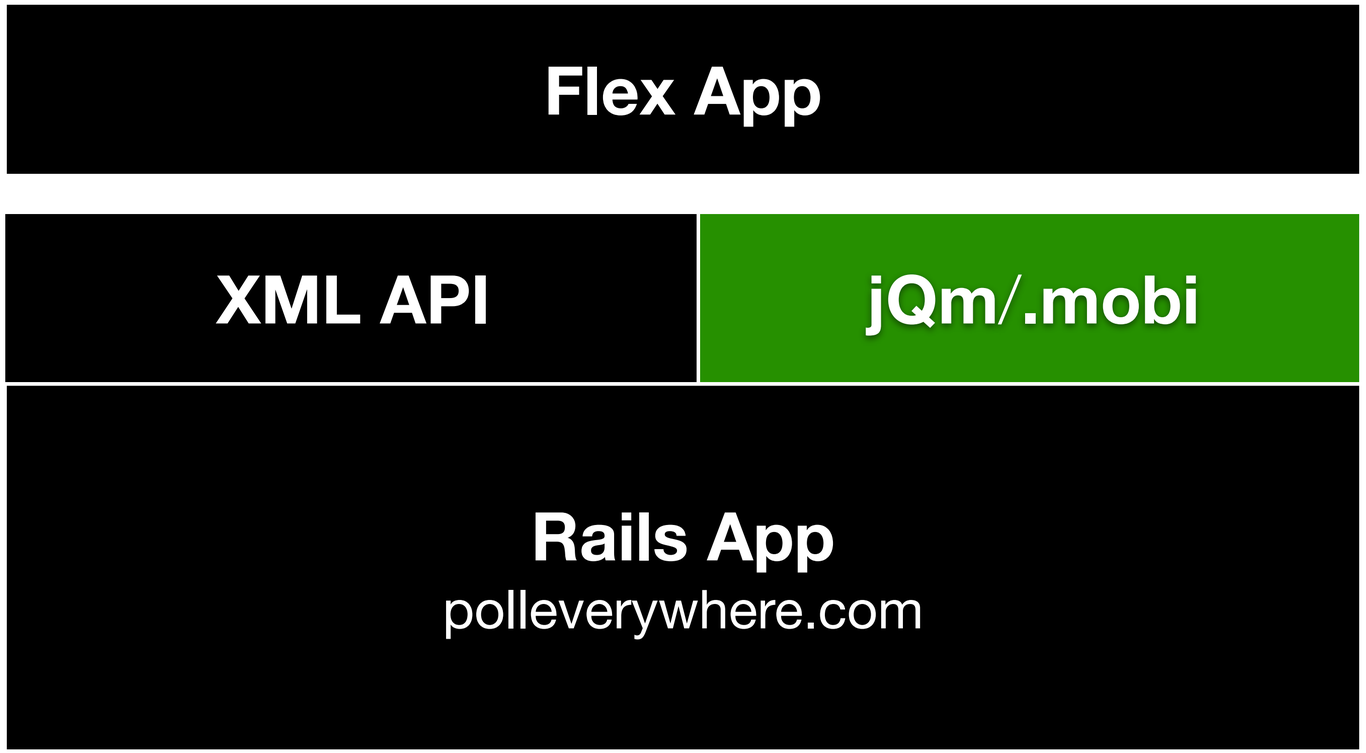
A **Flex app** was created **within the rails project** for PowerPoint visualizations



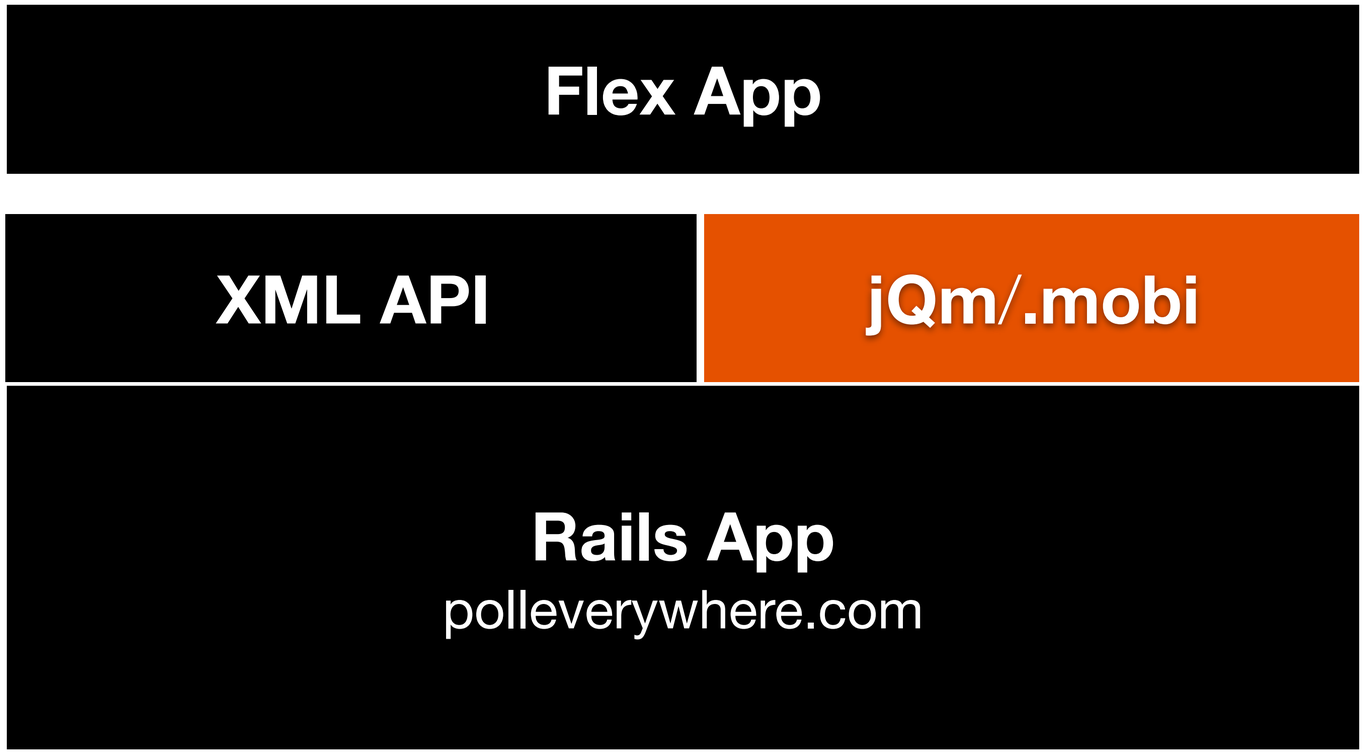
We broke the Flex app into its own project so that our Flex contractor could **deploy independently from the Rails app** team



Smart phones started taking off, so we built **.mobi extension views in Rails with jQuery mobile**



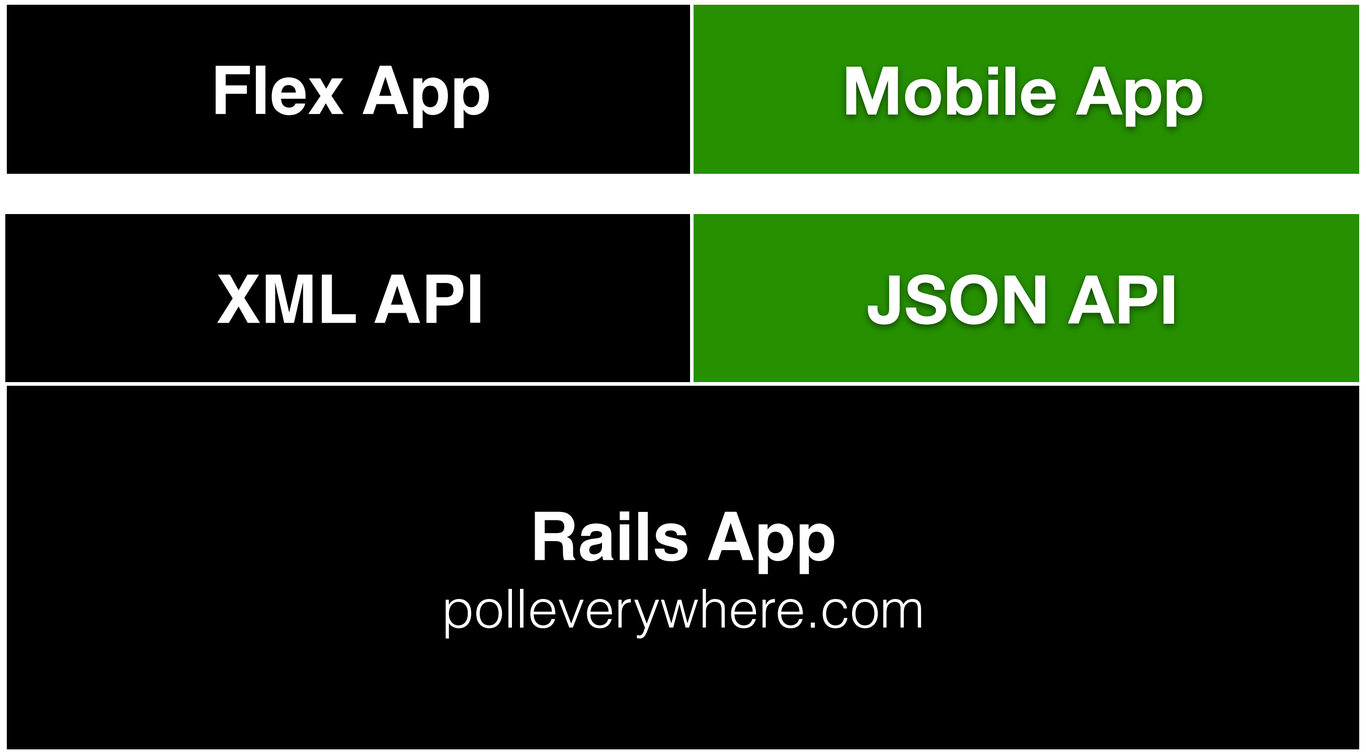
jQuery mobile and .mobi **was a big disaster**. Framework got us 80% there, the other 20% was painful and error prone



Our team made the decision to **hand pick libraries** to have **exactly what we need, when we need it.**

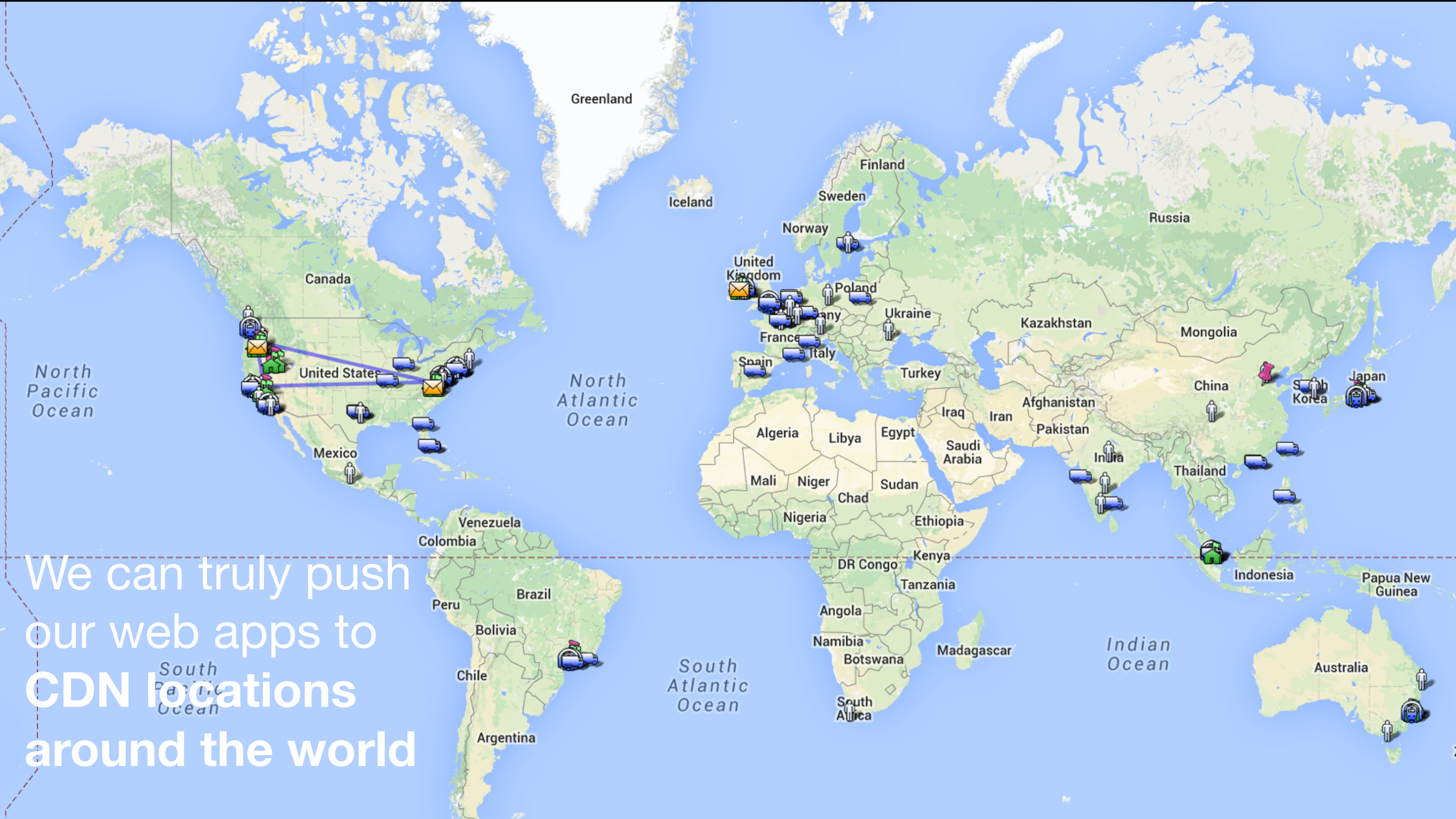
Framework agnostic Middleman would be there to manage all of the assets

Our first **single page HTML application** was built in Middleman at PollEv.com



CORS

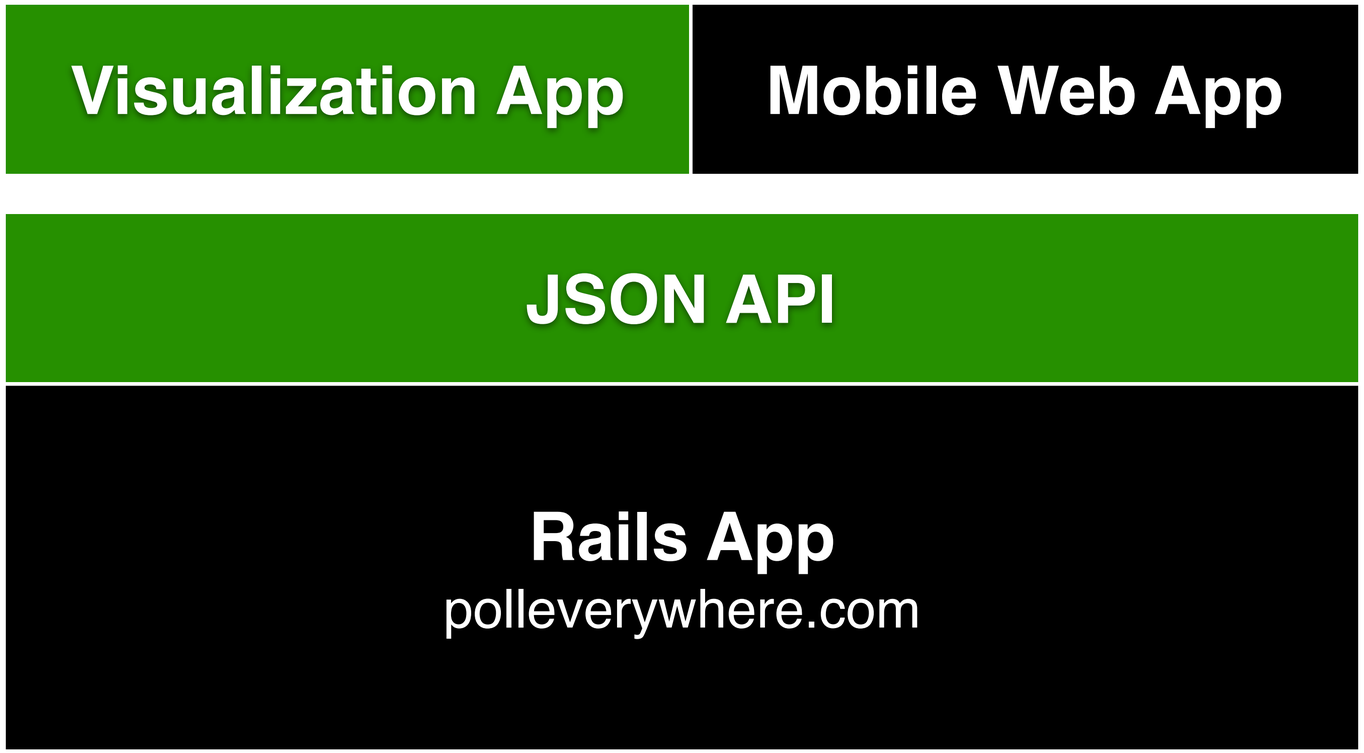
Cross-origin resource sharing



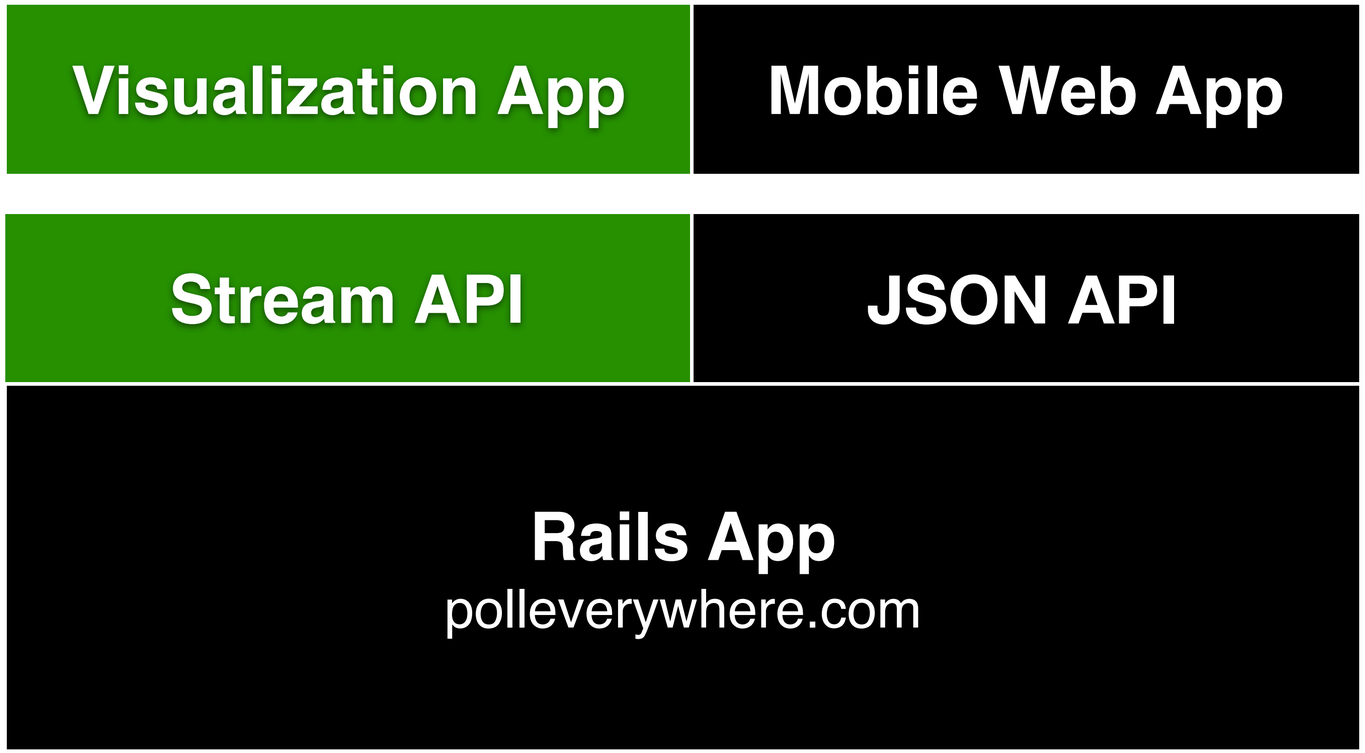
We can truly push our web apps to **CDN locations around the world**

… or to **floppy disks** and Cordova apps

Flash was on its deathbed, so we set out to replace the Flex app with **another single page Middleman app**

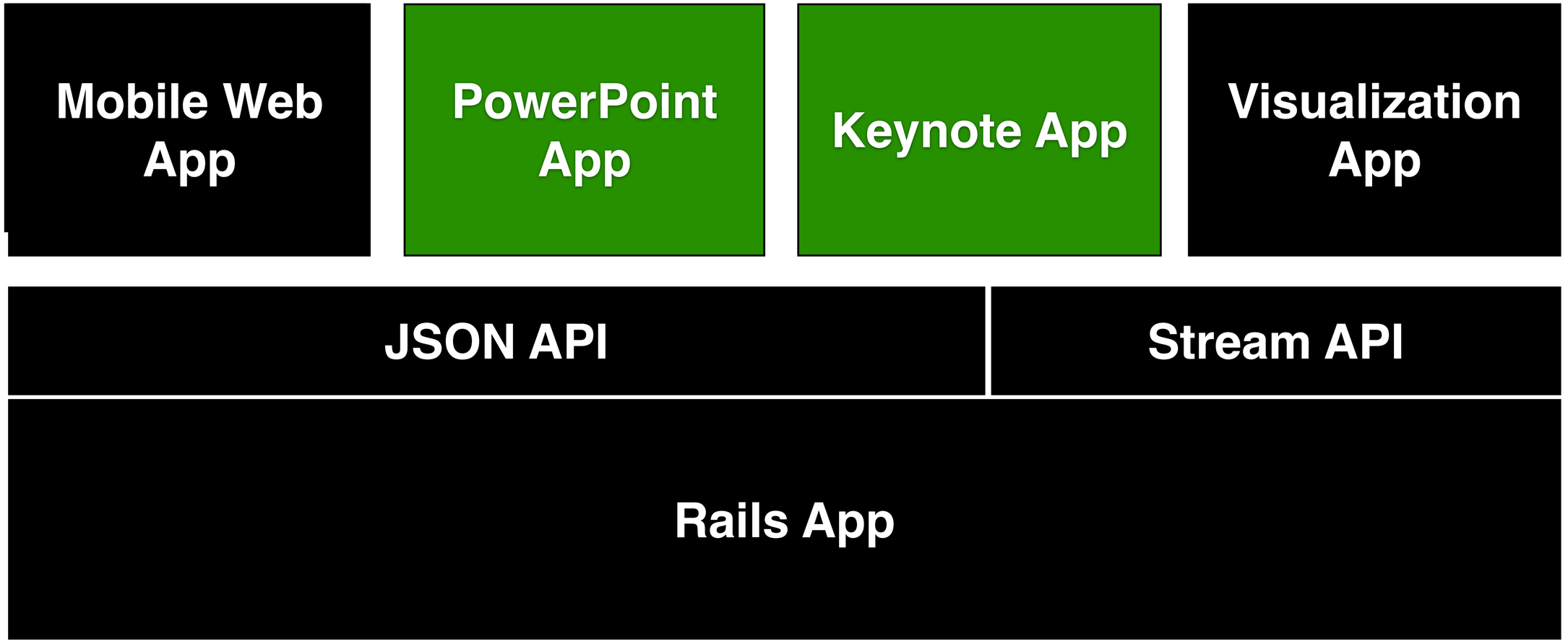


To reduce latency, a **Stream API** was deployed to production

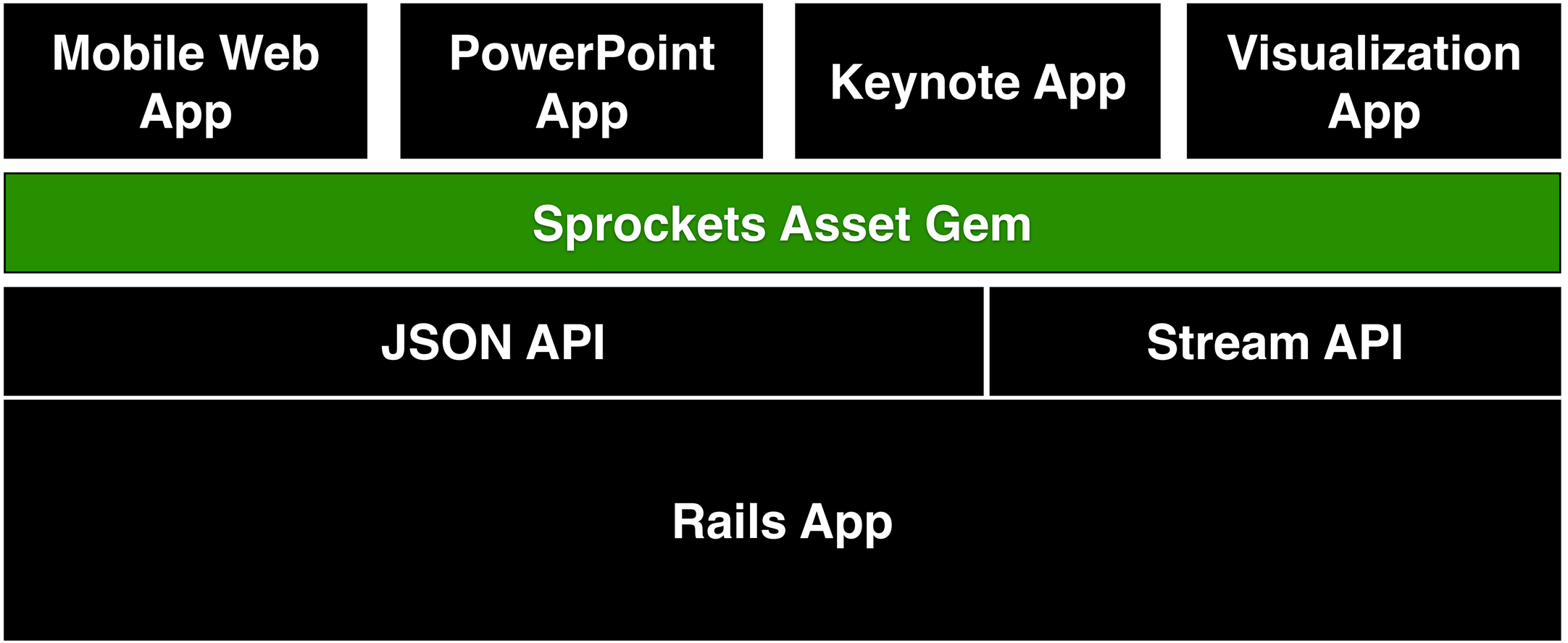


Stream API was isolated to its own host and server fleet for isolation and stability purposes. **Client-side SOA** allowed us to start using the Stream API, stabilize in production, gain confidence, and ensure a smooth rollout.

More apps emerge on the scene, **how do we keep it all DRY**?



Extract common Coffeescript, Sass, assets into a **sprockets asset gem**



Create a **sprockets asset gem** to share assets between apps

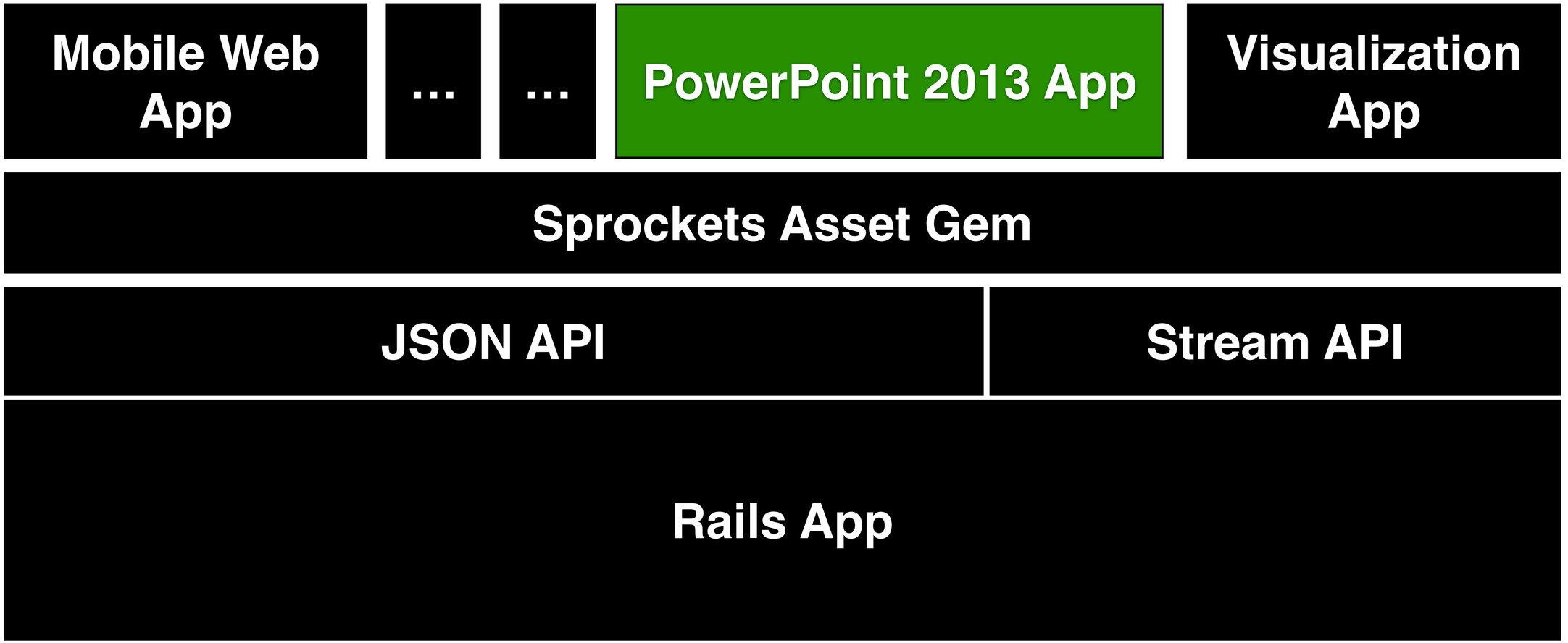
```
# The ./lib/pollev_assets.rb file
require 'pollev_assets/version'

if defined? Sprockets
  Sprockets.env.append_path 'lib/assets/javascripts'
  Sprockets.env.append_path 'lib/assets/stylesheets'
  Sprockets.env.append_path 'vendor/assets/javascripts'
end
```

```
# App ./Gemfile
if path = ENV['ASSETS_GEM_PATH']
  gem "pollev_assets", path: path
else
  gem "pollev_assets", branch: 'new-feature'
end
```

Manage asset gem dependencies and branches **with Bundler**

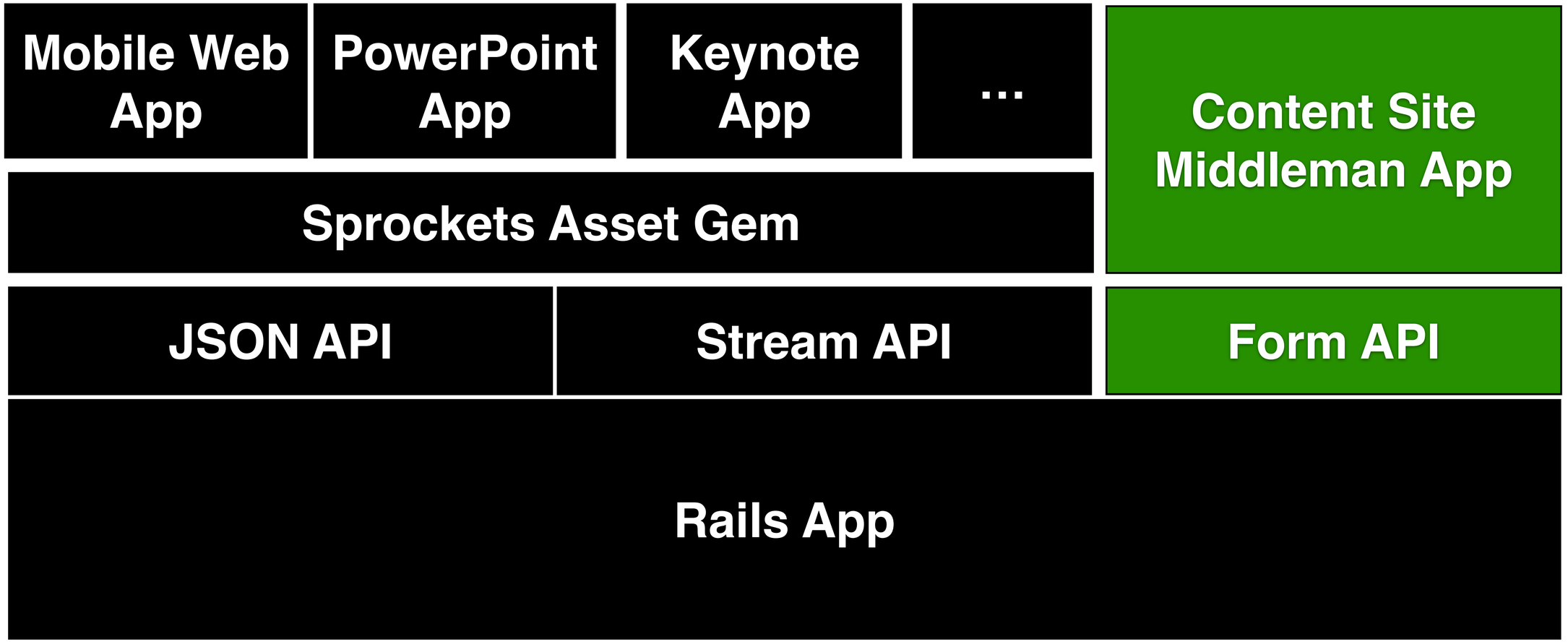
We **reused** mobile app **assets** to build PowerPoint 2013 app in 2 weeks



No out-of-box solution exist for organizing JS MVC apps in Middleman

What about **static websites**?

Extract content from Rails app so marketing team can move faster



Some **dynamic components** are still **required**

Render login state via JavaScript snippet

Integration between Middleman content site and Rails app should have well defined and tested integration points

Stripe.js, GA, Optimizely are all dynamic tools that can be integrated into a static website

Why bother with a static content website?

Don’t let “TechCrunched” or HN’ed or whatever the cool kids call it these days bring down your site

Introduce completely different workflows into content production without a heavy CMS

Easier deployments, can you upload a file?

…a few things I didn’t cover

How to not use hash bang URLs with pushState

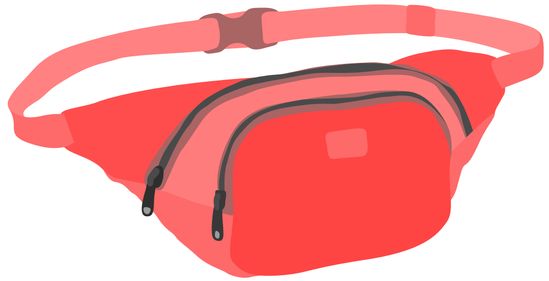
12-factor middleman apps

Validate test, staging, and any other environment with acceptance tests

The state of exception monitoring in production environments

Building an API side-by-side with client code deployments

MIDDLEMAN



getfannypack.com

Join our team!

polleverywhere.com/jobs

@bradgessler

for slides, links, and code