# Intro to Single Page Web Applications

Springfield Web Meetup Group - May 2015

# Dan Quinn

- twitter/github: @danielpquinn
- smashingideas.com
- earlybirdy.com

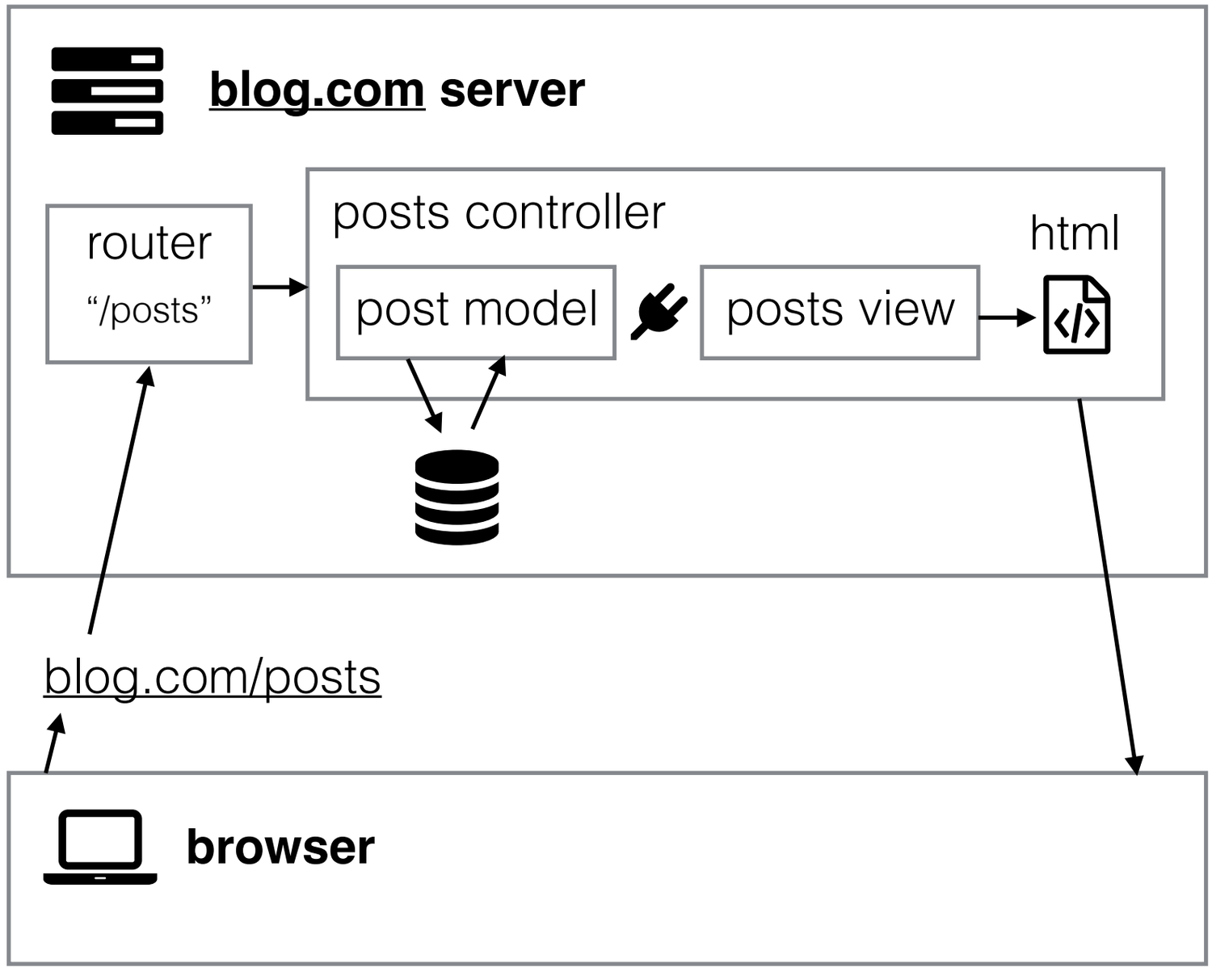
# Agenda

- Architectural overview
- Brief history of single page apps
- Why build single page apps? Why not?
- My experience with single page apps
- Check out some sample code
- QA / Open discussion

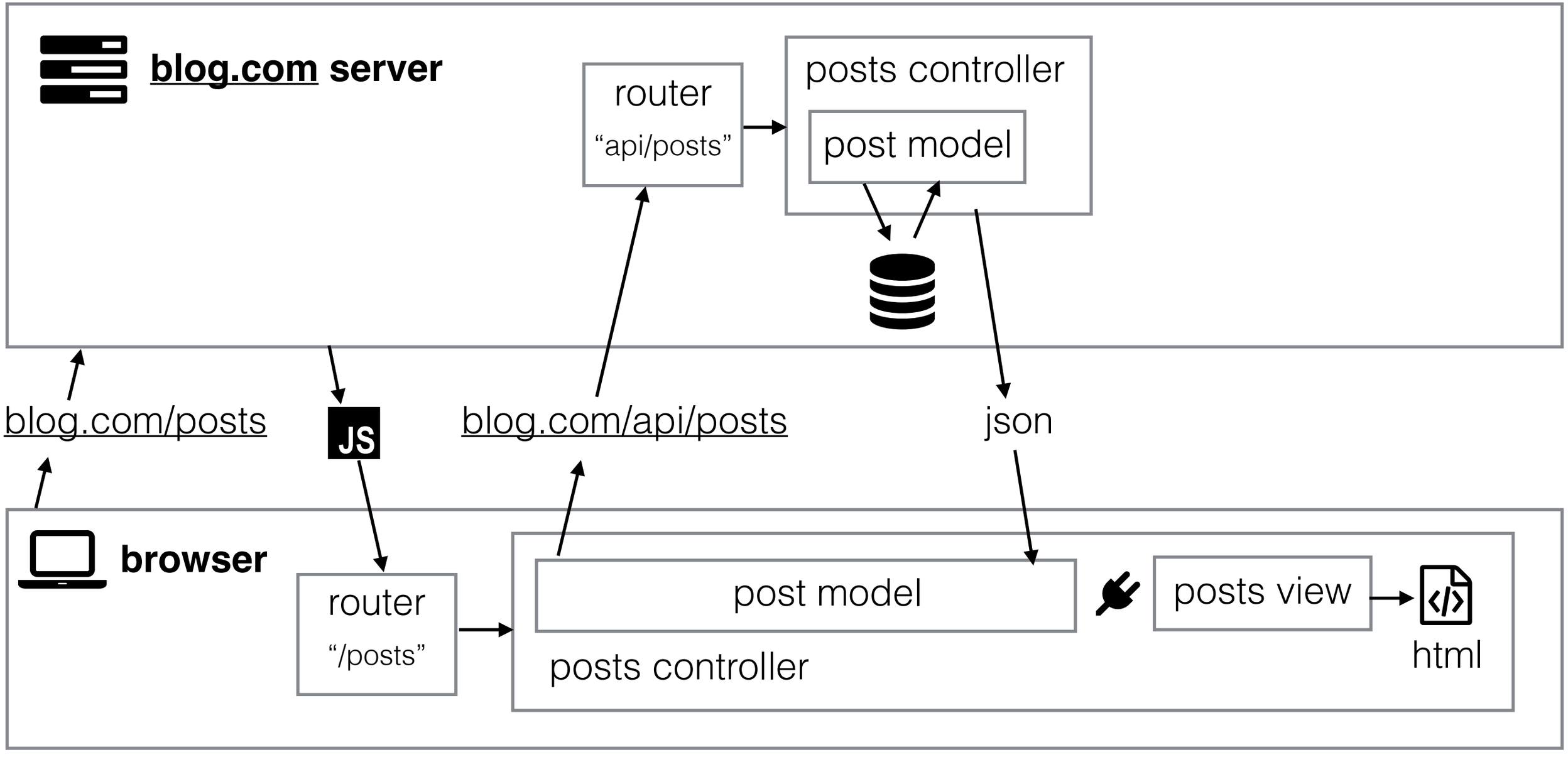
# What is a SPA?

single page web application

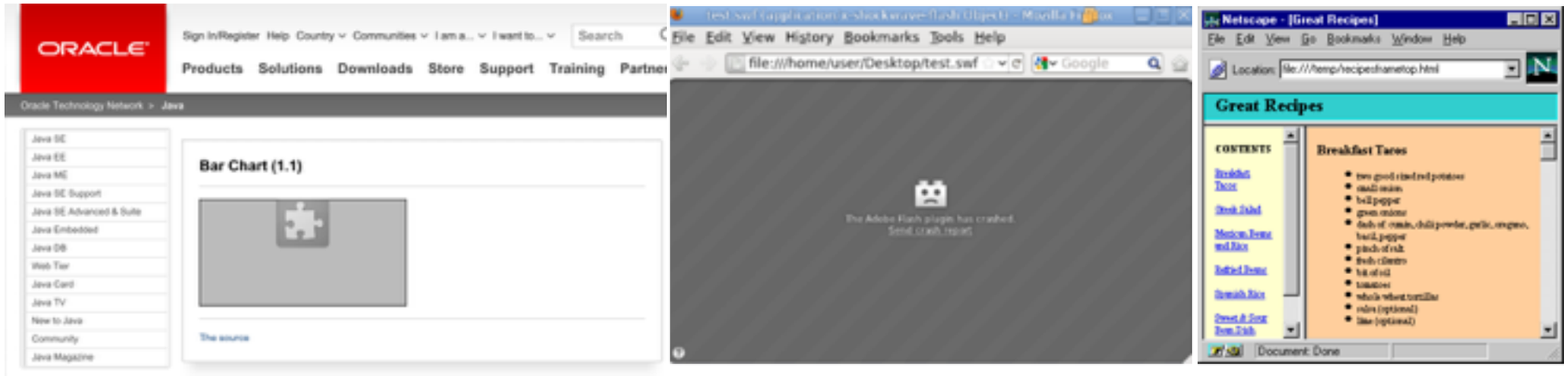
# Traditional Web Application Architecture

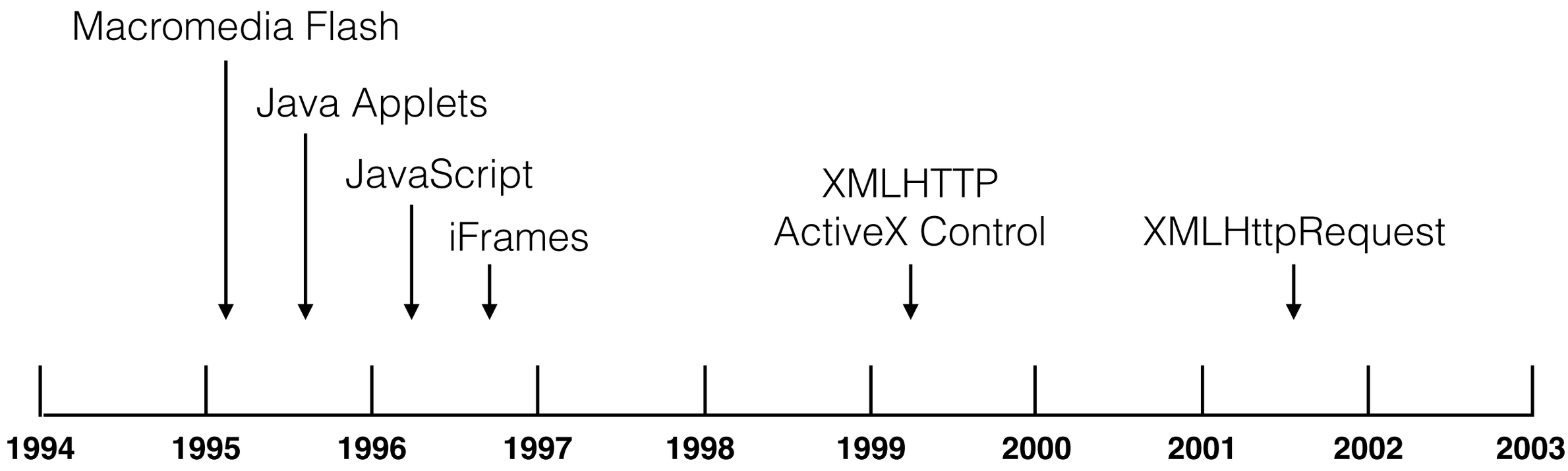


# Single Page Web Application Architecture

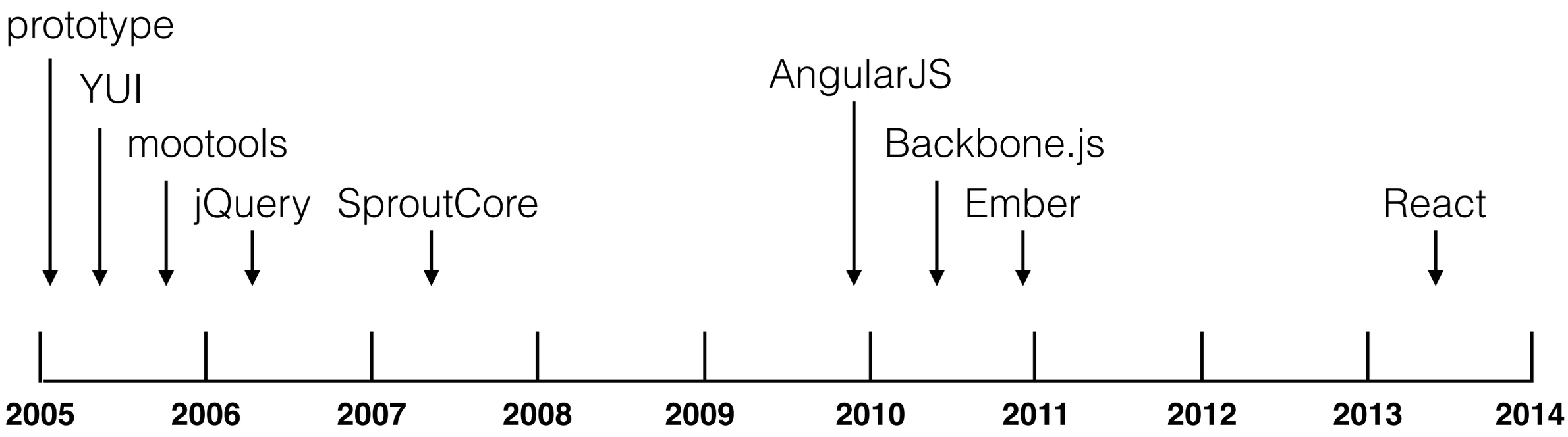


# How Did We Get Here?



# How Did We Get Here?

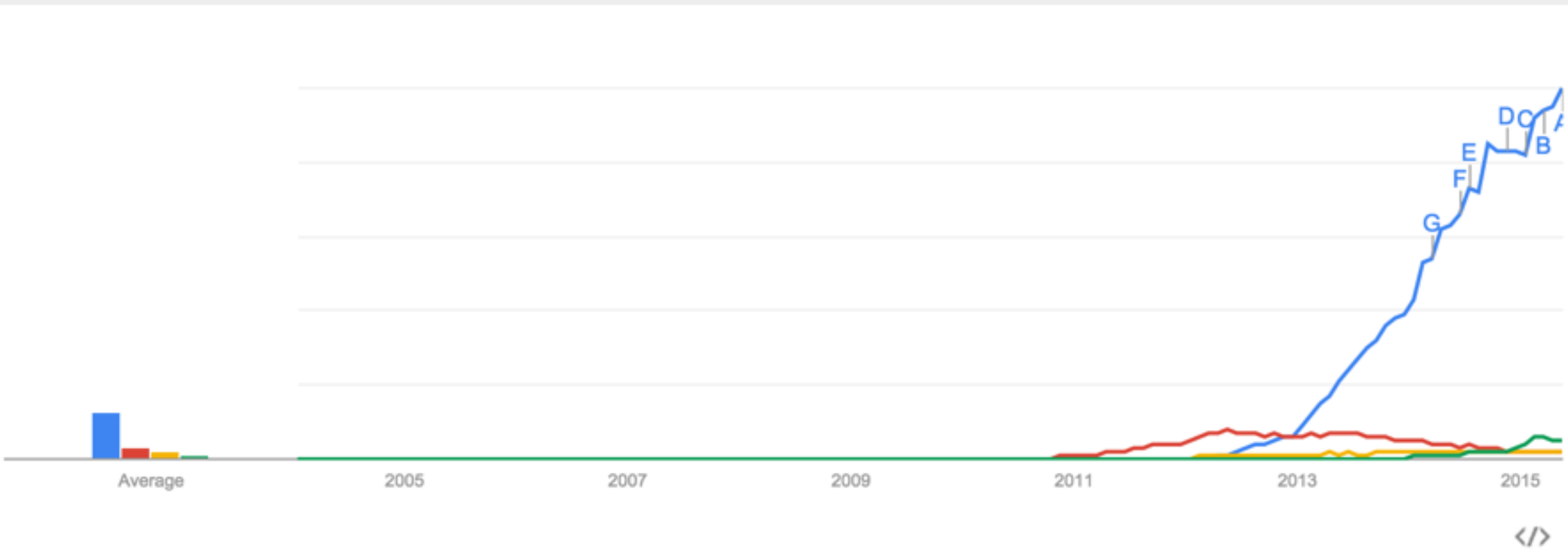


## Compare Search terms

| AngularJS | Backbone.js | EmberJS | ReactJS | + Add term |
|---|---|---|---|---|
| Search term | Search term | Search term | Search term | |

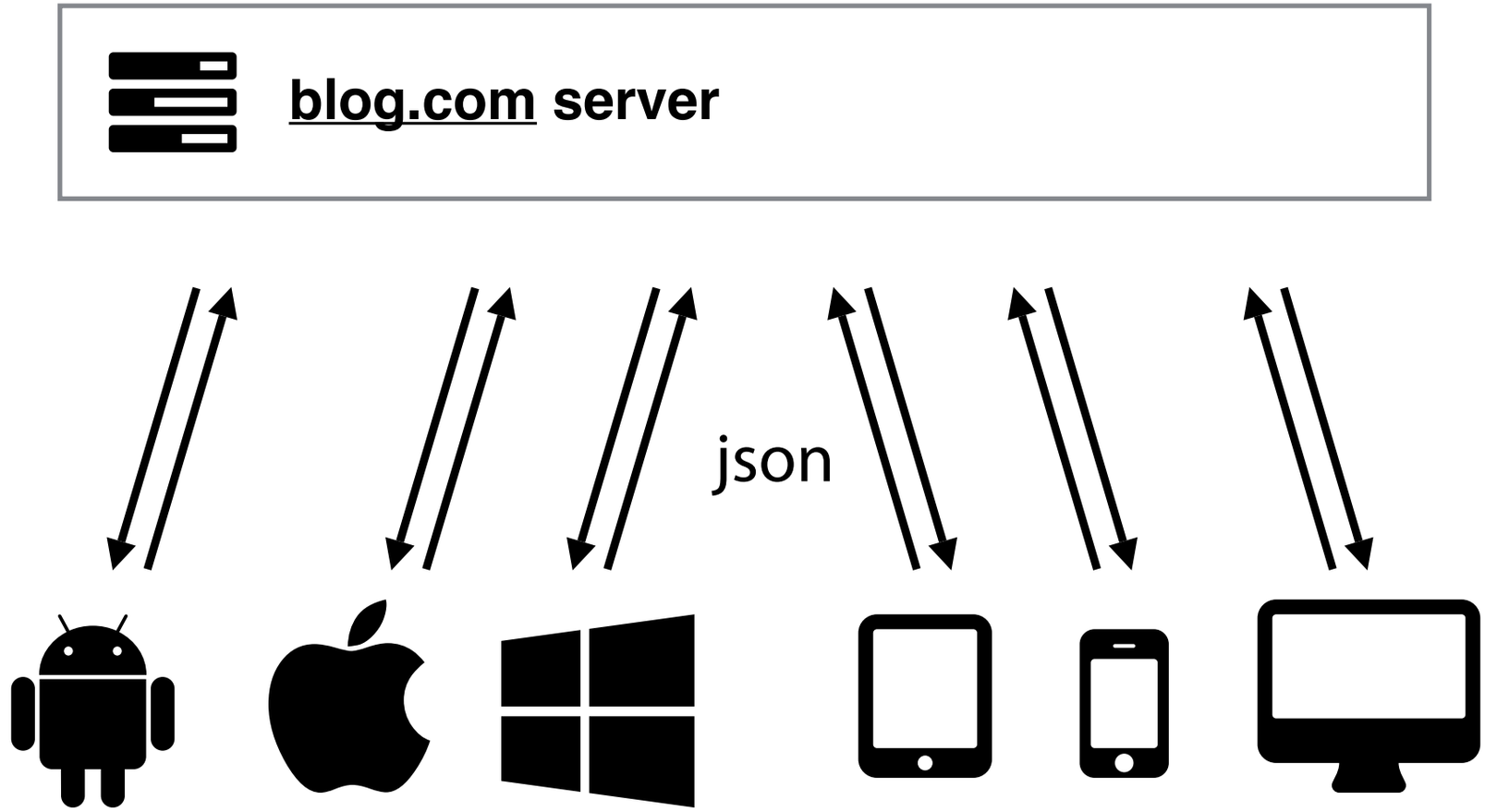
## Interest over time

News headlines  Forecast

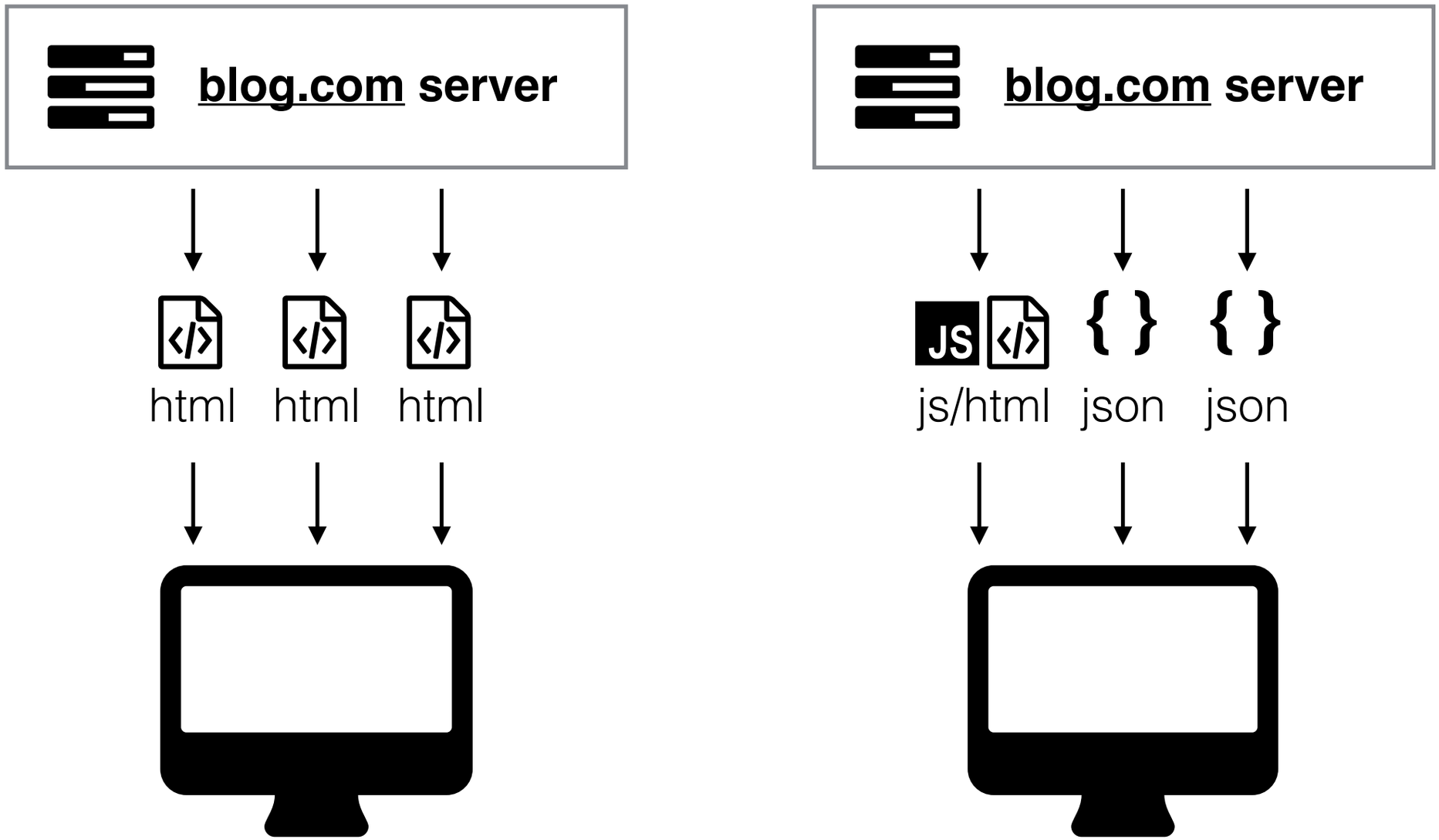


Why?

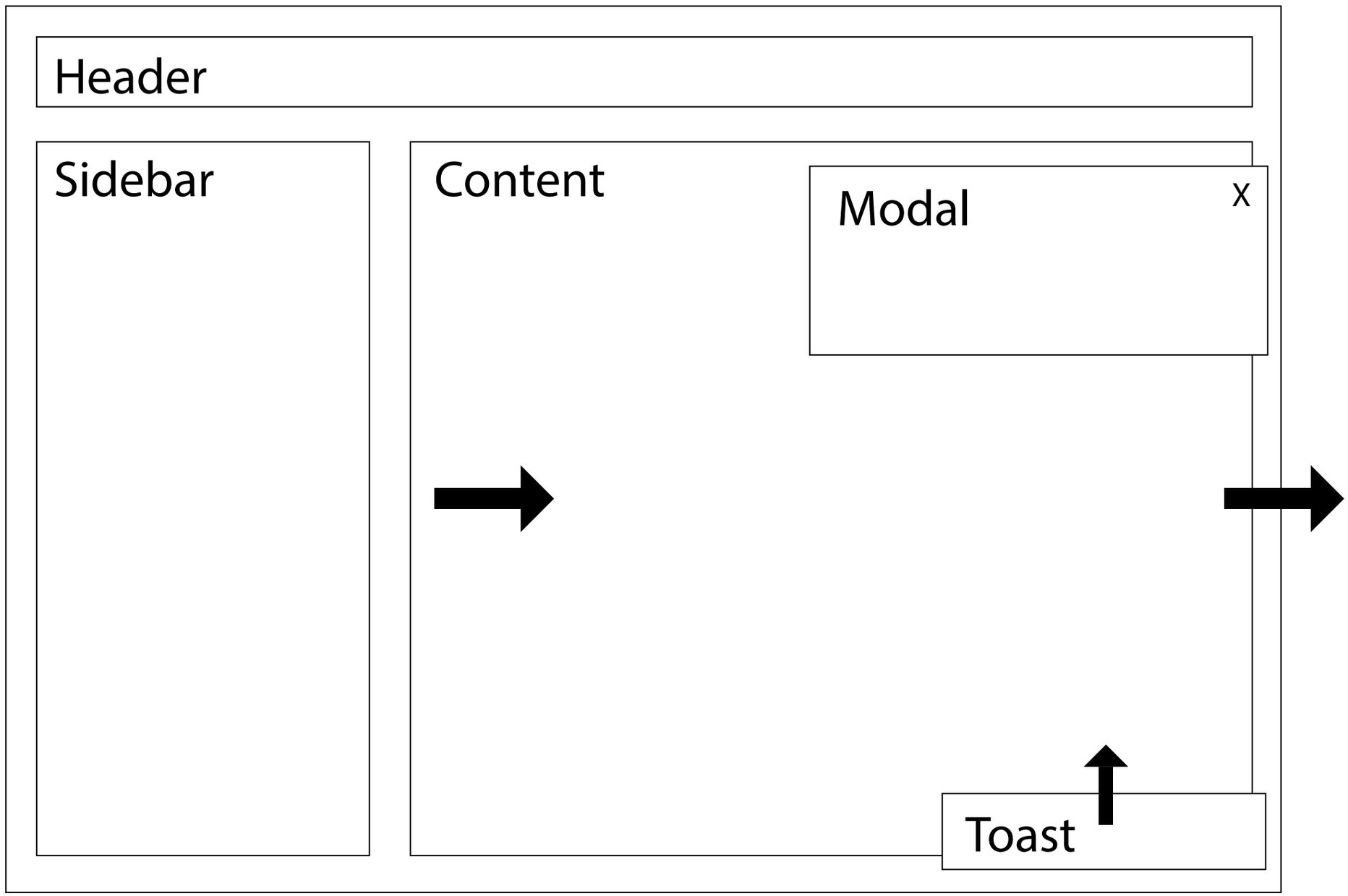
# Separation of Concerns



# Less data (maybe)



# User Experience

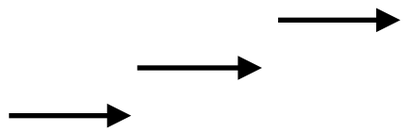


# Let’s look at code

http://github.com/danielpquinn/springfield-web-spa

# main.js

- Entry point into application
- Create a new instance of App after DOMContentLoaded
- Create Post model
- Create header, sidebar and content View instances
- Create a router, define routes
- Execute current route (home)

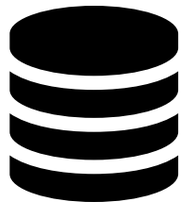


# router.js

- Stores internal list of route definitions

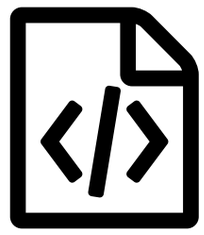
- **addRoute** takes route, controller function

- **navigate** takes path, matches route, pulls out params and executes controller function using those params. Finally updates url using history API's pushState



# model.js

- Subclassed and assigned static config properties
- **list** returns a list of items (/api/posts)
- **fetch** returns a single item (/api/posts/:id)
- **save** saves current state (put /api/posts/:id)
- **create** creates a new item (post /api/posts)
- **remove** removes item (delete /api/posts/:id)



# view.js

- Subclassed, instantiated with template container id template function

- **render** takes data as argument and plugs it into string template, inserts result into DOM

# Thanks!

Links to slides and code on meetup.com/springfieldweb