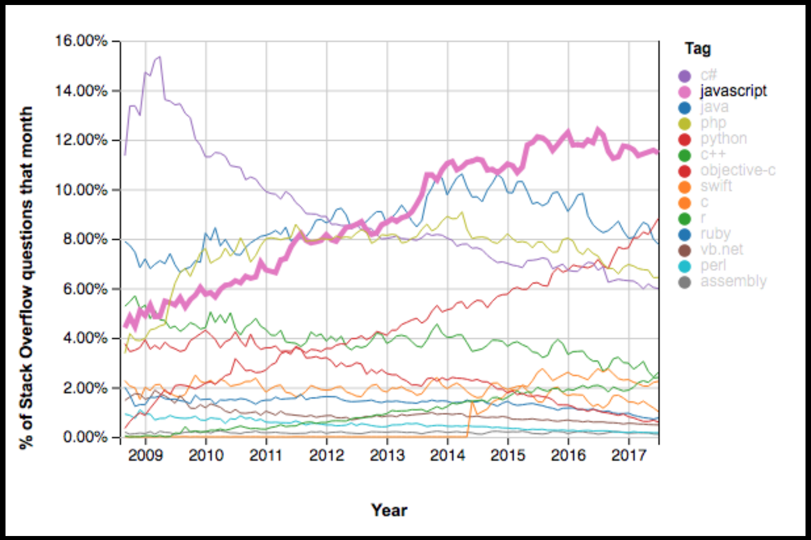
New trends in javascript development (2018)



By Peter Cosemans

The Top Trends

- Angular, React & Vue.js
- Javascript Is Still Fastest-growing
- The rize of framework CLI's
- Improve testability with Jest, Storybook & CypressJS
- GraphQL, your next API
- Client Side, Server Side and Pre-rendering
- Deployment to Netlify & Serverless

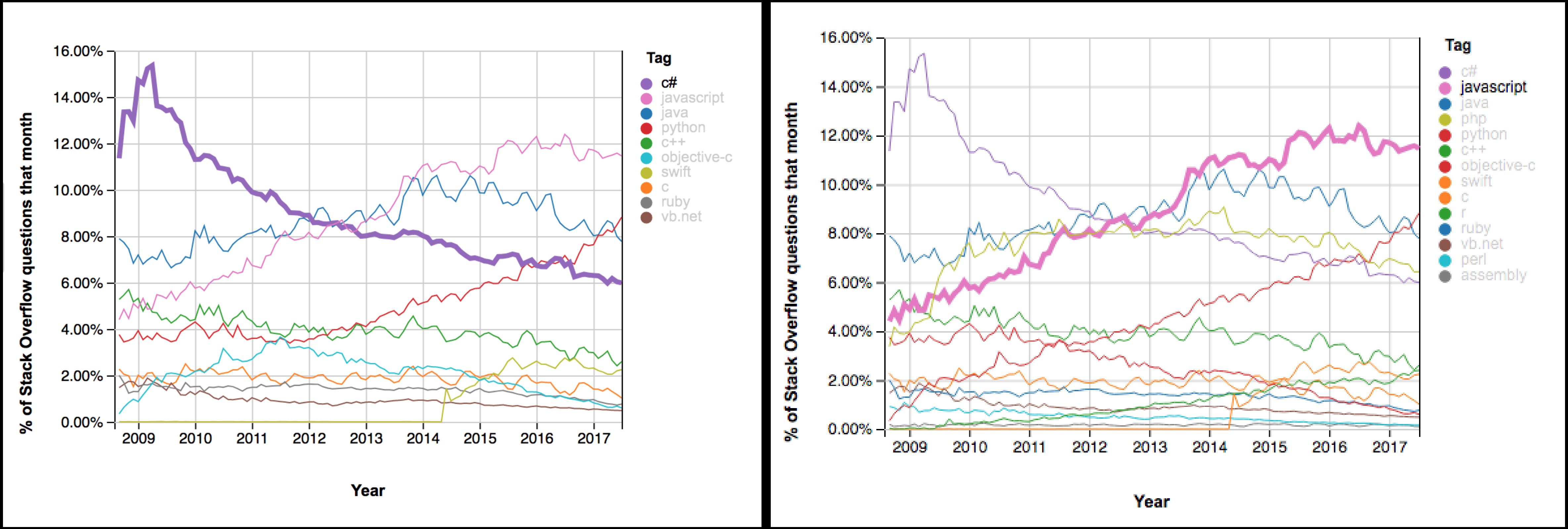
The Top Mobile Trends

- React Native & Flutter
- PWA
- Web Components
- Ionic 4 & Capacitor

Javascript the language

Still getting bigger every year

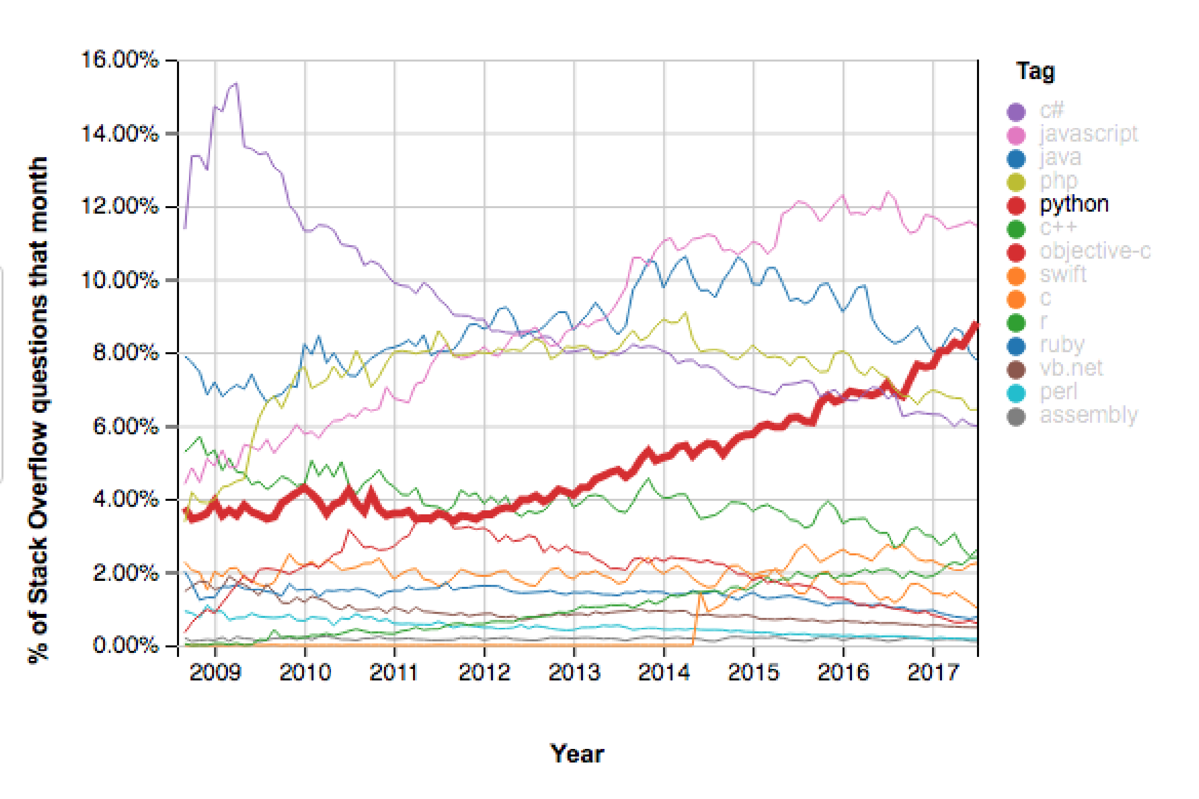
C# vs JavaScript



Statistics from StackOverflow

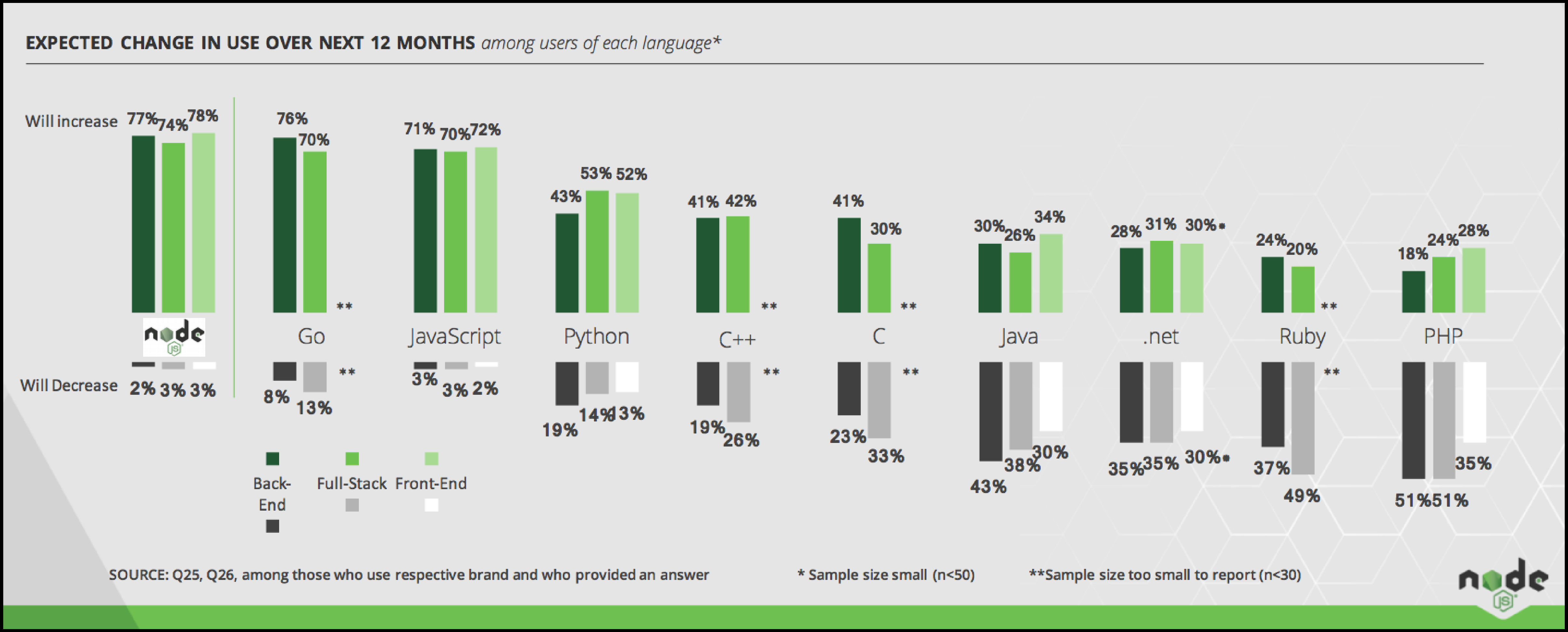
Javascript Is Still One of the Fastest-growing Languages

Python is Rising



Statistics from StackOverflow

Expected change in use



EXPECTED CHANGE IN USE OVER NEXT 12 MONTHS *among users of each language**

SOURCE: Q25, Q26, among those who use respective brand and who provided an answer

* Sample size small (n<50)

**Sample size too small to report (n<30)

Coming soon to JavaScript near you

- Async/Await
- ES Modules for NodeJS
- Class fields (stage-3, TS 1.x)
- Private fields/methods (stage-3, TS 2.5)
- Optional catch binding (stage-3, TS 2.5)
- Dynamic import (stage-3, TS2.4)
- Numeric separators (stage-2, TS2.7)
- Decorators (stage-2, TS1.5)
- Optional chaining (stage-1)

Async/Await

```
function getAllUsers() {
    return api.get('api/users')
        .then(res => {
            return res.data;
        })
}
```

```
function async getAllUsers() {
    const res = await api.get('api/users')
    return res.data;
}
```

Class fields

(ES stage-3, TS 1.x)

```
class MyClass {
    state = {
        counter: 0
    }
    static propTypes = {
        name: PropTypes.String
    }
}
```

Numeric separators

(ES stage-2, TS2.7)

```
const x = 123_234_242;
const y = 123234242;
x === y;    // true
```

Private fields/methods

(ES stage-3, TS ????)

```
class MyClass {
    #counter = 0;

    gimmTheCount() {
        this.#inc();
        return this.#coounter;
    }

    #inc() { this.#counter++; }
}
```

Optional chaining

(stage-1, TS ????)

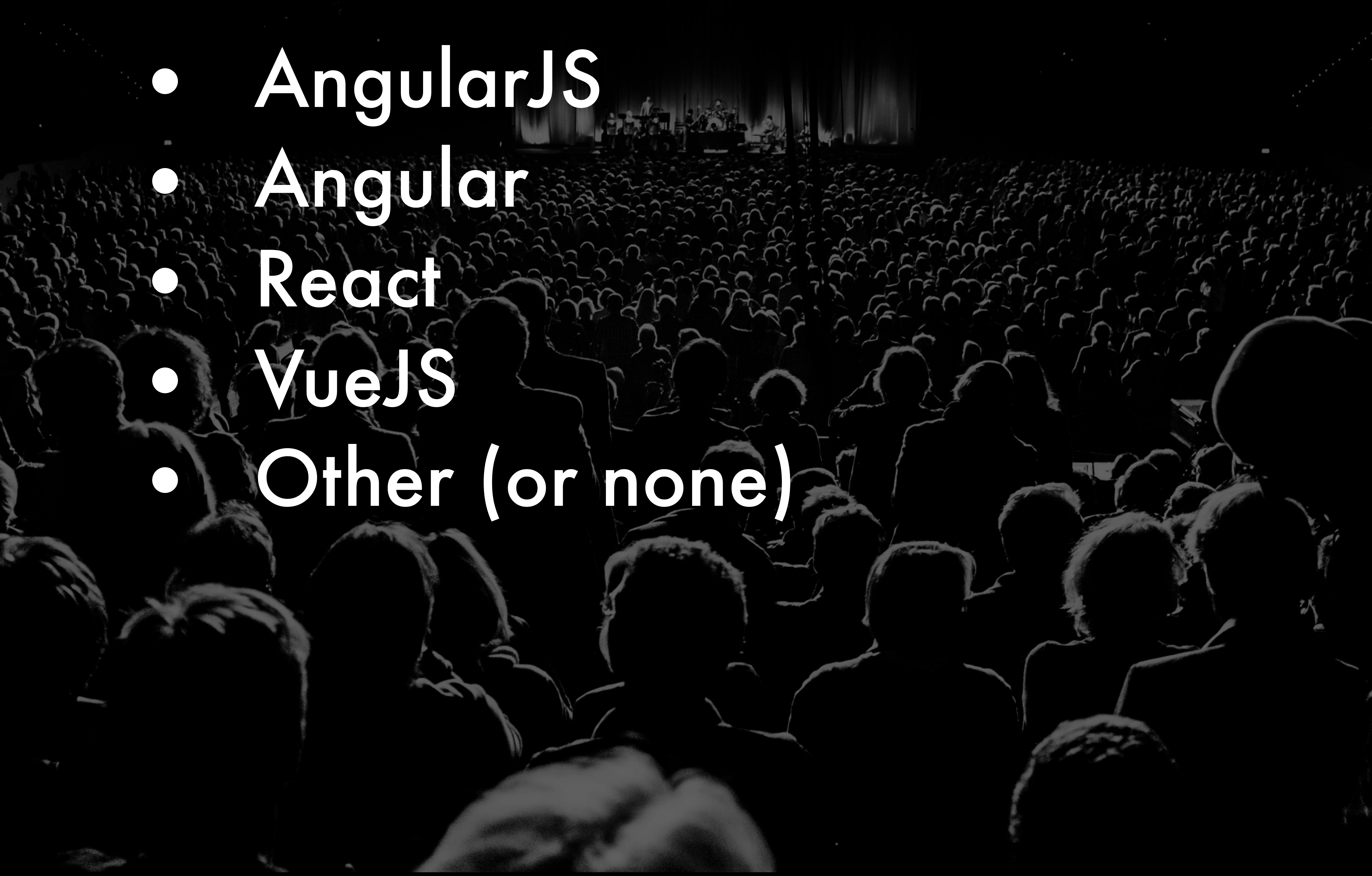
```
const x = (foo && foo.bar) ? foo.bar.x : undefined;

const x = foo?.bar?.y
```

JavaScript Frameworks

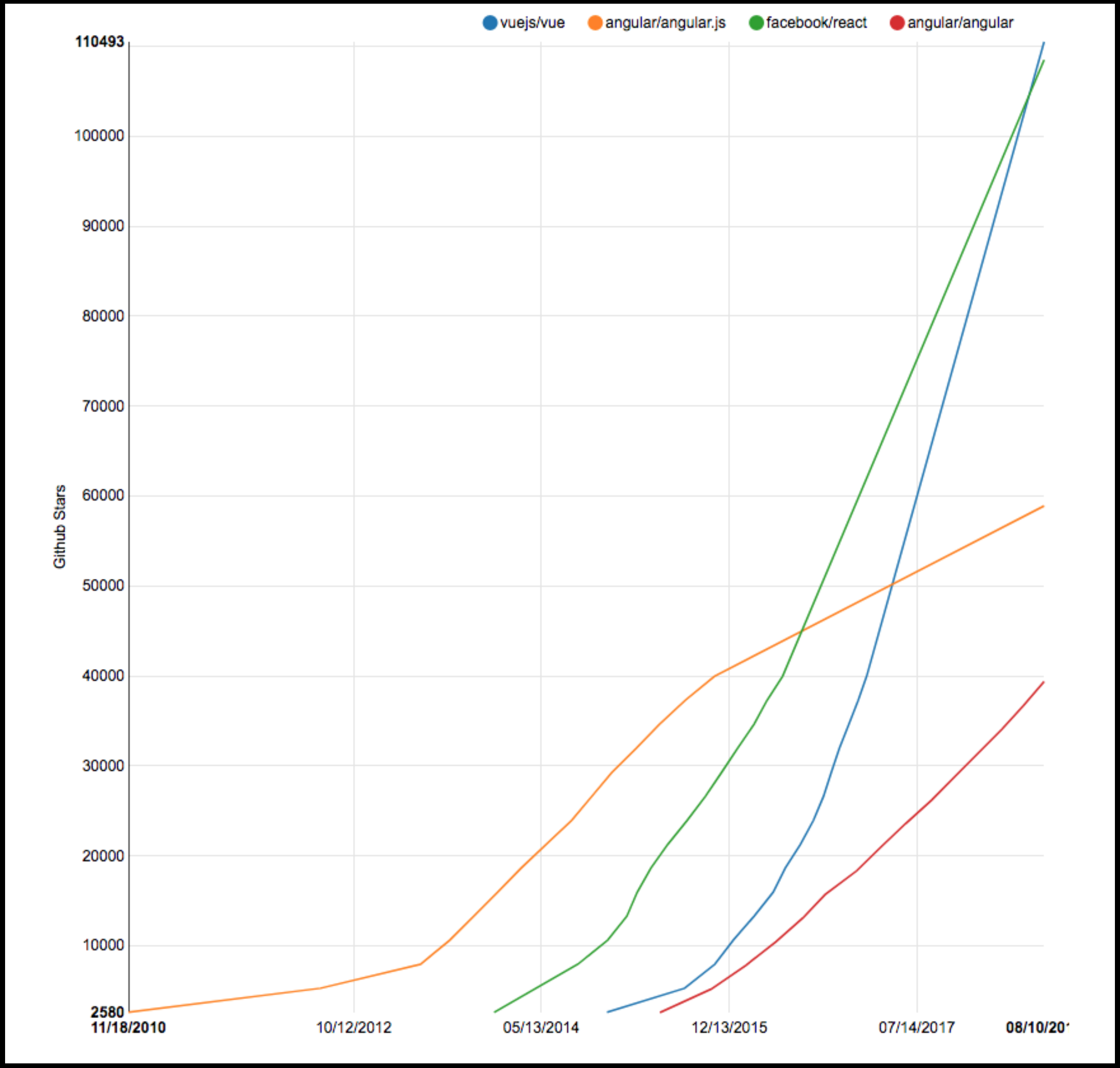
Battle of the Giants

What front-end framework are you using?

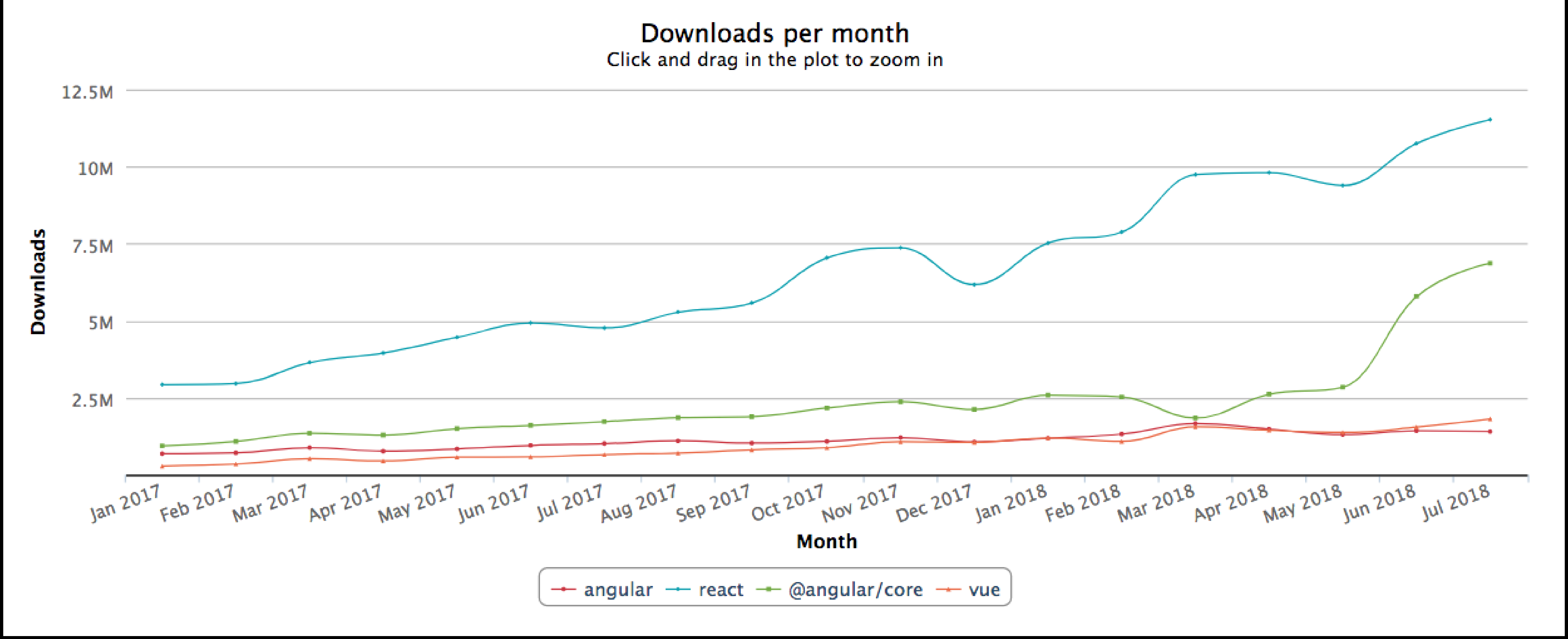
- AngularJS
- Angular
- React
- VueJS
- Other (or none)



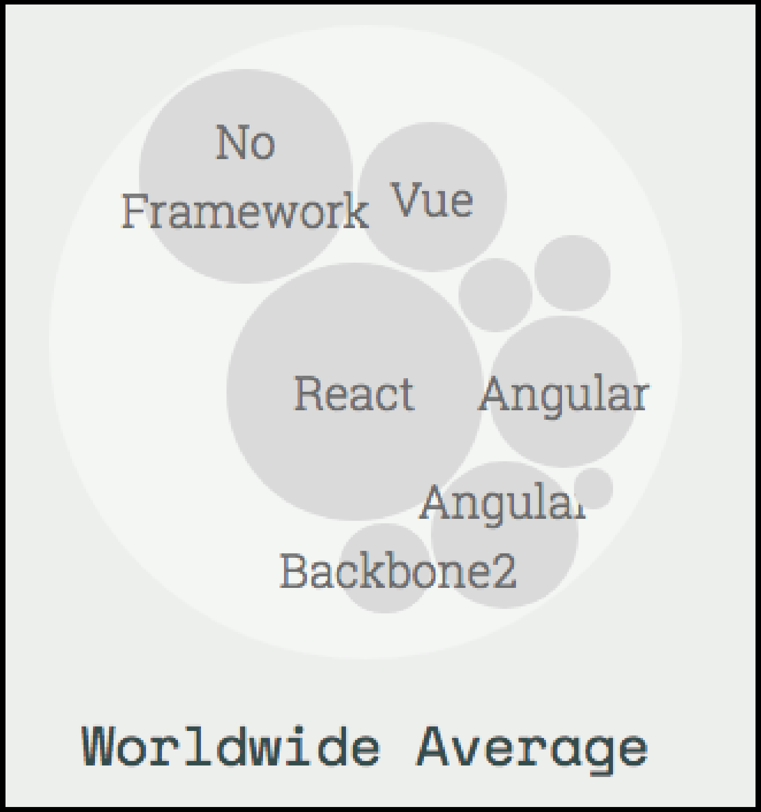
GitHub Stars

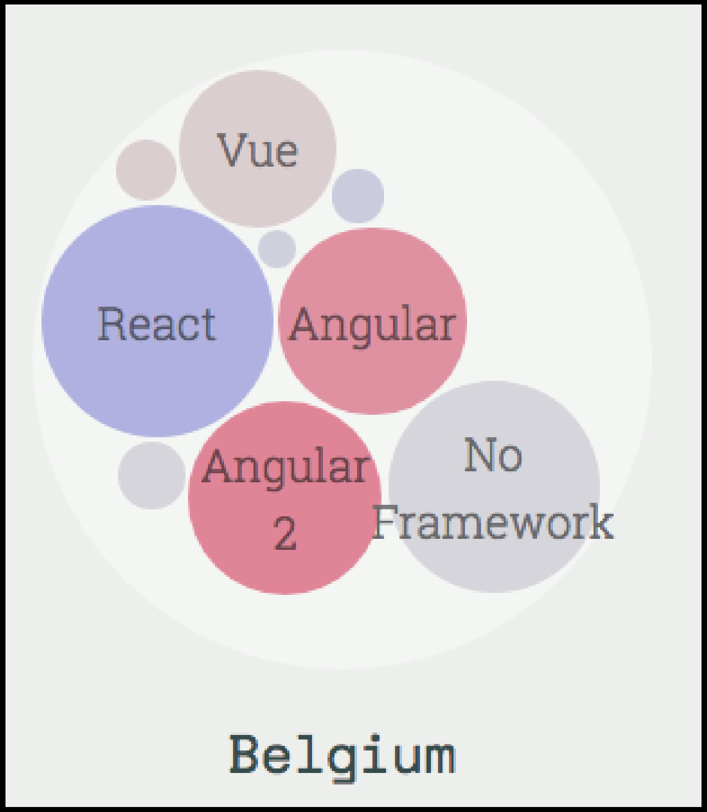


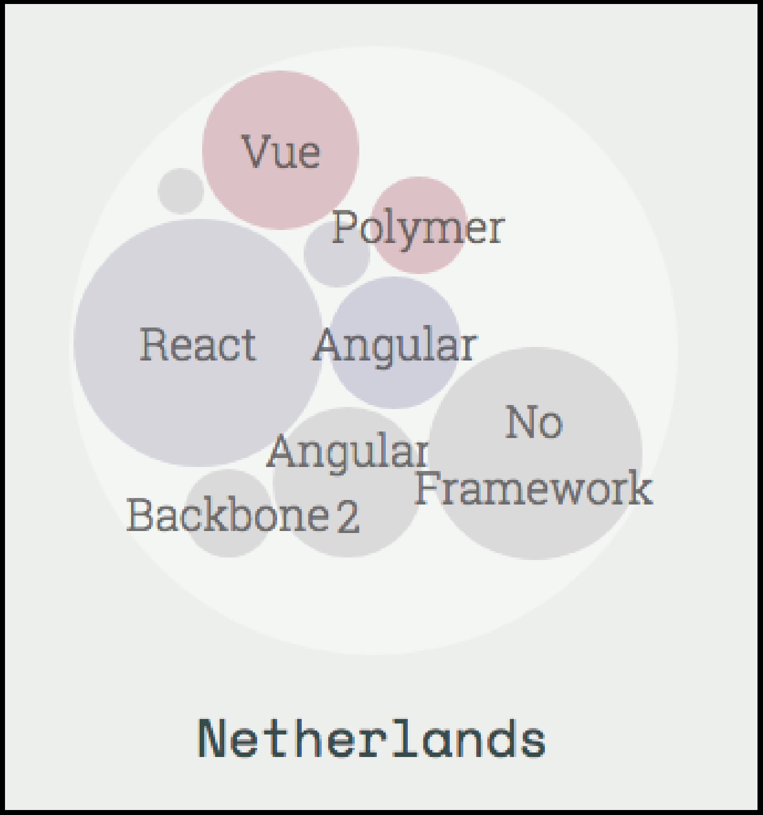
Npm Downloads



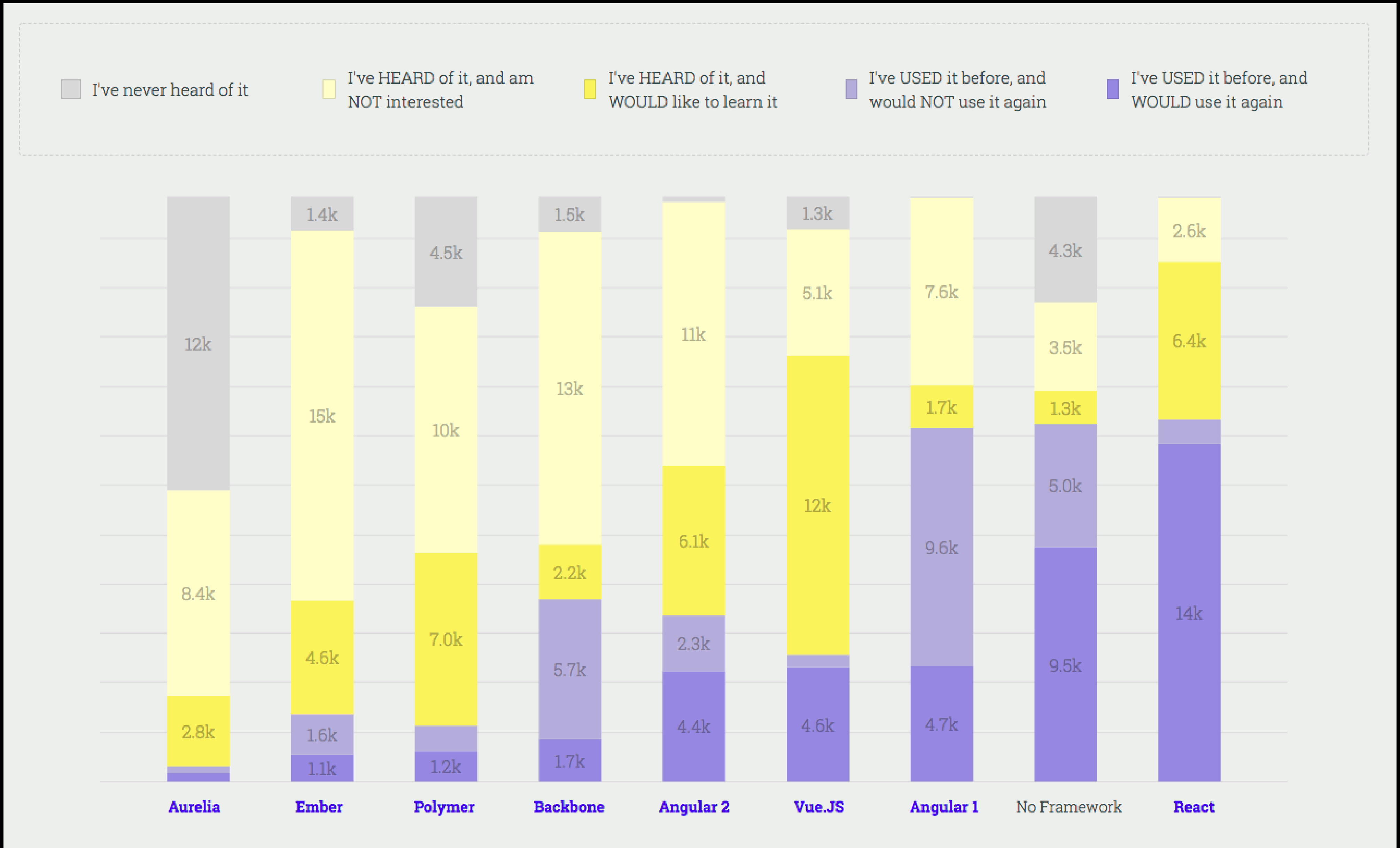
Frameworks Usage



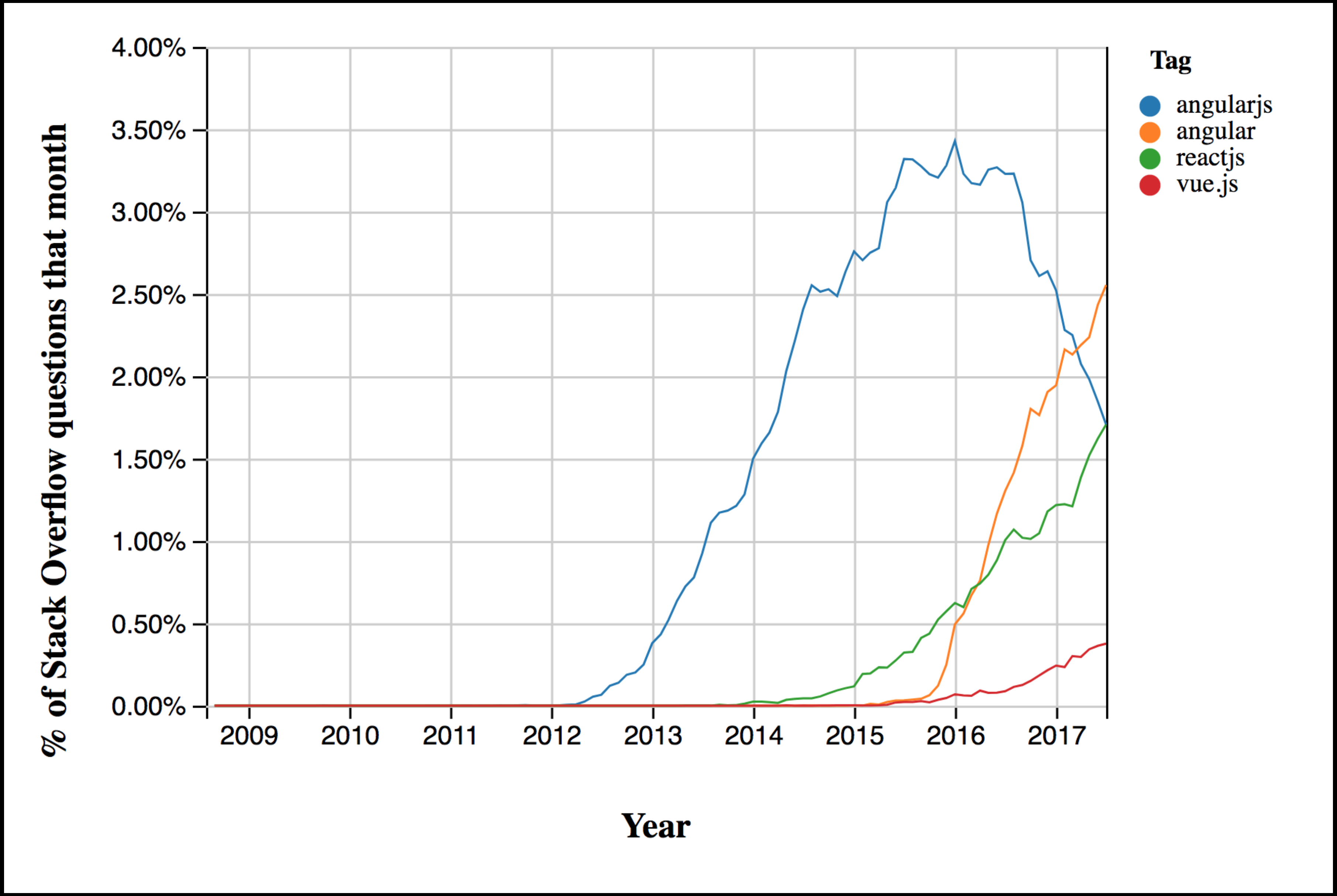




Use & Intrest


I've never heard of it
I've HEARD of it, and am NOT interested
I've HEARD of it, and WOULD like to learn it
I've USED it before, and would NOT use it again
I've USED it before, and WOULD use it again
12k
8.4k
2.8k
1.4k
15k
4.6k
1.6k
1.1k
4.5k
10k
7.0k
1.2k
1.5k
13k
2.2k
5.7k
1.7k
11k
6.1k
2.3k
4.4k
1.3k
5.1k
12k
4.6k
7.6k
1.7k
9.6k
4.7k
4.3k
3.5k
1.3k
5.0k
9.5k
2.6k
6.4k
14k
Aurelia
Ember
Polymer
Backbone
Angular 2
Vue.JS
Angular 1
No Framework
React

Stack Overflow

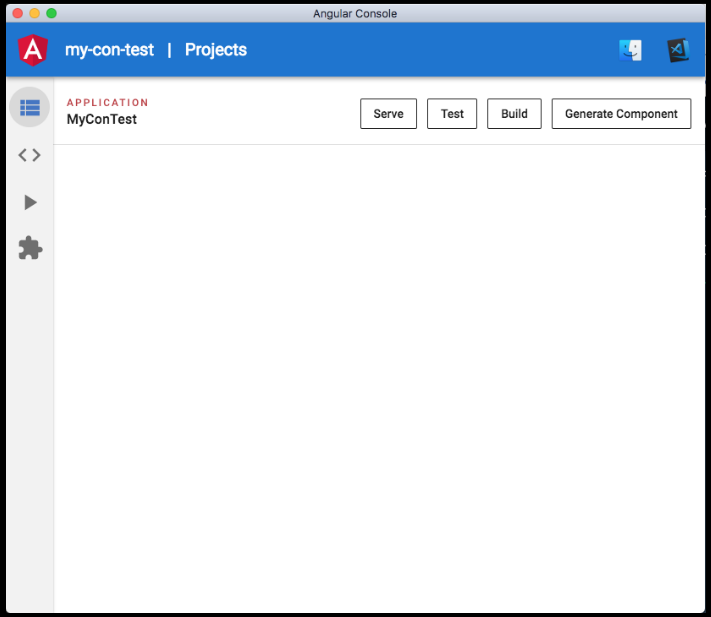


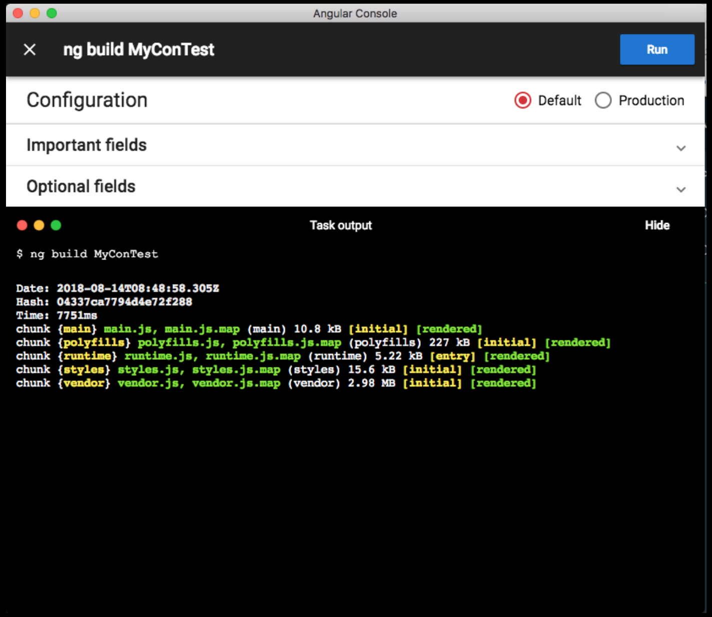
Angular

It's just Angular

Angular 6.0

- A new CLI (workspaces, library, schematics, webpack 4, ...)
- RxJS 6.0 (three-shaking)
- Tree-shakable providers
- Consistent versions (cli, material, router, ...)
- Ivy Renderer & Angular Elements (web components)
- Angular Console





```
$ ng build MyConTest

Date: 2018-08-14T08:48:58.305Z
Hash: 04337ca7794d4e72f288
Time: 7751ms
chunk {main} main.js, main.js.map (main) 10.8 kB [initial] [rendered]
chunk {polyfills} polyfills.js, polyfills.js.map (polyfills) 227 kB [initial] [rendered]
chunk {runtime} runtime.js, runtime.js.map (runtime) 5.22 kB [entry] [rendered]
chunk {styles} styles.js, styles.js.map (styles) 15.6 kB [initial] [rendered]
chunk {vendor} vendor.js, vendor.js.map (vendor) 2.98 MB [initial] [rendered]
```

Angular 6.0 - 7.0

😡😡😡

- Breaking & complex CLI
- Breaking changes in RxJS
- Angular Element != Web Components
- Ivy not yet
- Poor Angular Console
- v7: @angular/core split, @aiStore & @angular/mine, breaking compiler

Is Angular moving away from the JavaScript ecosystem?

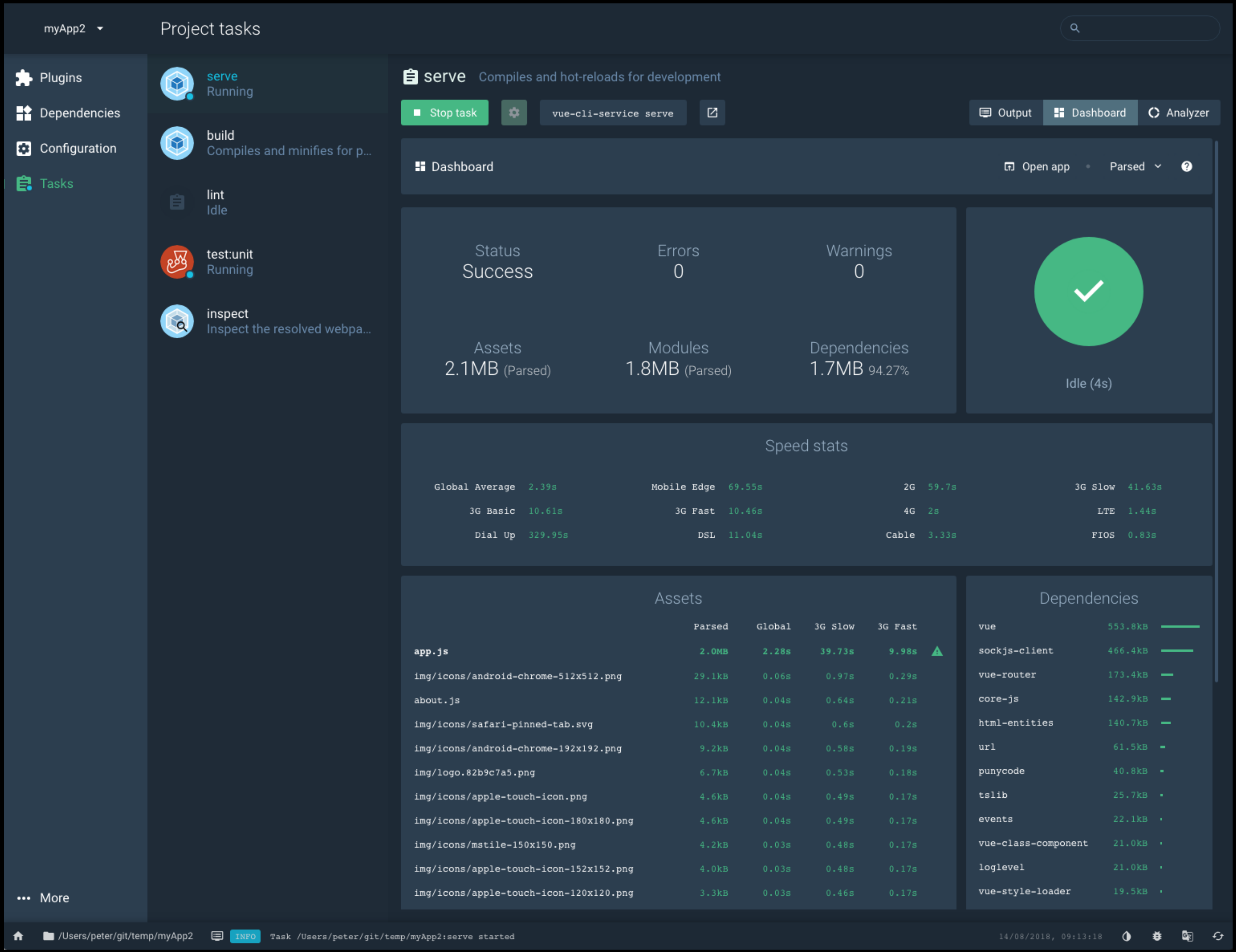
VueJS

Fast and simple

VueJS 2018 - Productivity

- Prettier Vue support
- eslint-plugin-vue (errors in templates)
- vue-test-utils
- Vue Devtools 5.0 (routing & perf tab, editable Vuex state)
- @vue/cli: v3.0 👏

Is makes you more productive as a VueJS developer



serve
Running

build
Compiles and minifies for p...

lint
Idle

test:unit
Running

inspect
Inspect the resolved webpa...

serve Compiles and hot-reloads for development

`vue-cli-service serve`

Dashboard

Status	Errors	Warnings
Success	0	0

Assets	Modules	Dependencies
2.1MB (Parsed)	1.8MB (Parsed)	1.7MB 94.27%

Idle (4s)

Speed stats

Global Average	2.39s	Mobile Edge	69.55s	2G	59.7s	3G Slow	41.63s
3G Basic	10.61s	3G Fast	10.46s	4G	2s	LTE	1.44s
Dial Up	329.95s	DSL	11.04s	Cable	3.33s	FIOS	0.83s

Assets

	Parsed	Global	3G Slow	3G Fast
app.js	**2.0MB**	**2.28s**	**39.73s**	**9.98s** ⚠
img/icons/android-chrome-512x512.png	29.1kB	0.06s	0.97s	0.29s
about.js	12.1kB	0.04s	0.64s	0.21s
img/icons/safari-pinned-tab.svg	10.4kB	0.04s	0.6s	0.2s
img/icons/android-chrome-192x192.png	9.2kB	0.04s	0.58s	0.19s
img/logo.82b9c7a5.png	6.7kB	0.04s	0.53s	0.18s
img/icons/apple-touch-icon.png	4.6kB	0.04s	0.49s	0.17s
img/icons/apple-touch-icon-180x180.png	4.6kB	0.04s	0.49s	0.17s
img/icons/mstile-150x150.png	4.2kB	0.03s	0.48s	0.17s
img/icons/apple-touch-icon-152x152.png	4.0kB	0.03s	0.48s	0.17s
img/icons/apple-touch-icon-120x120.png	3.3kB	0.03s	0.46s	0.17s

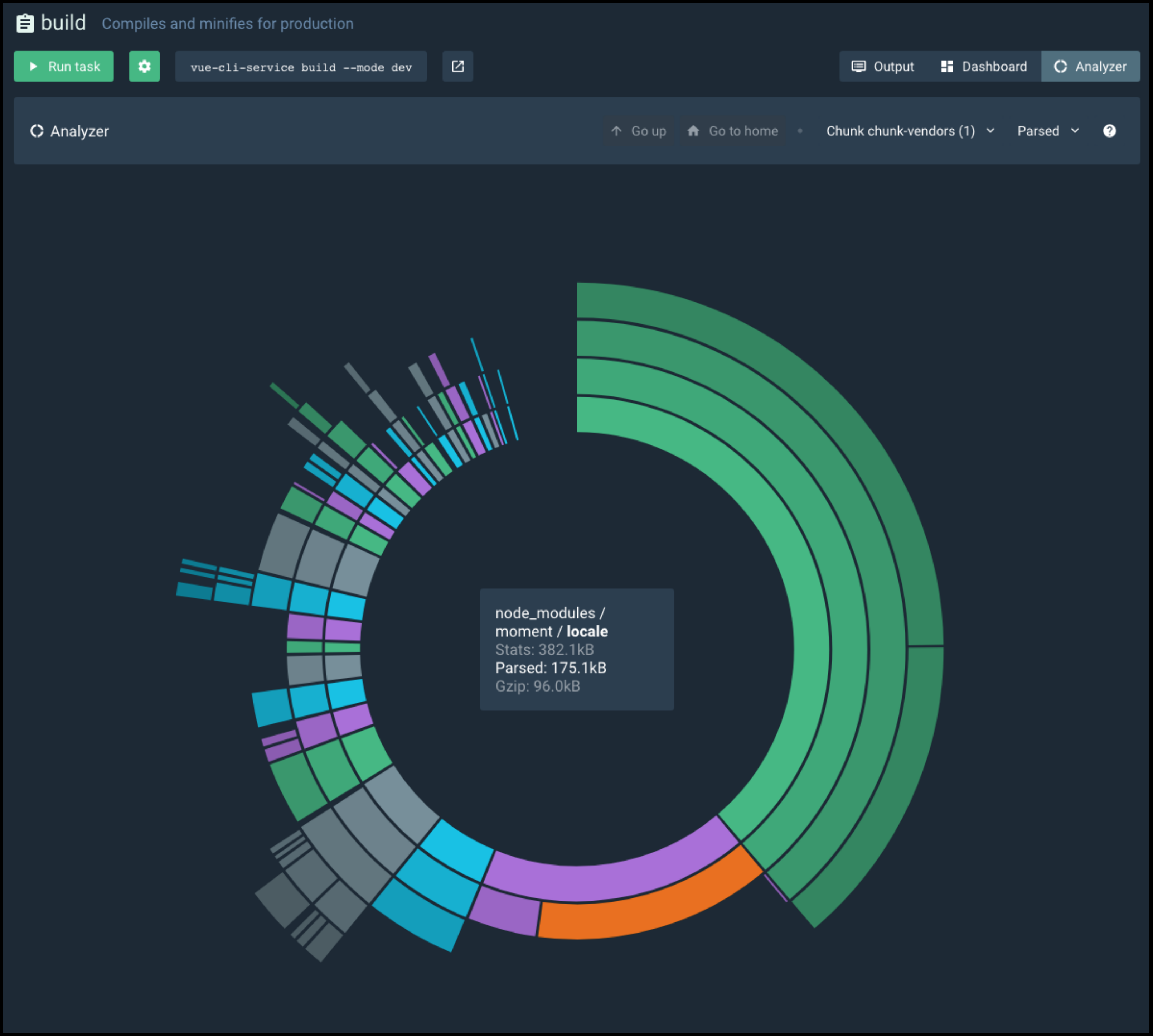
Dependencies

vue	553.8kB
sockjs-client	466.4kB
vue-router	173.4kB
core-js	142.9kB
html-entities	140.7kB
url	61.5kB
punycode	40.8kB
tslib	25.7kB
events	22.1kB
vue-class-component	21.0kB
loglevel	21.0kB
vue-style-loader	19.5kB

/Users/peter/git/temp/myApp2

```
INFO Task /Users/peter/git/temp/myApp2:serve started
```

14/08/2018, 09:13:18



```
vue-cli-service build --mode dev
```

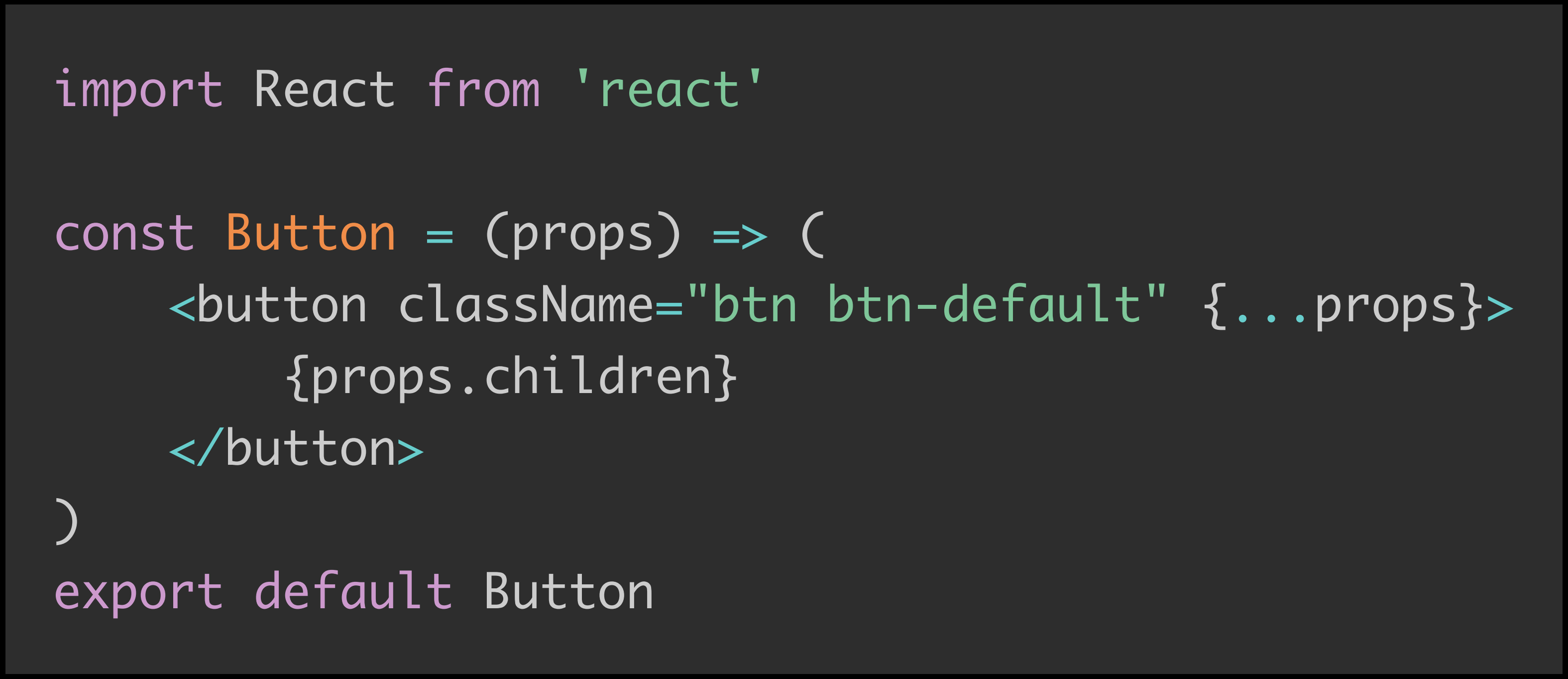
node_modules / moment / **locale**
Stats: 382.1kB
Parsed: 175.1kB
Gzip: 96.0kB

React

Gives you Wings

😍 I Love React 😍

Simple button component



```
import React from 'react'

const Button = (props) => (
    <button className="btn btn-default" {...props}>
        {props.children}
    </button>
)
export default Button
```

```
@Component({
    templateUrl: './button.component.html'
})
export class ButtonComponent {
    @Input() type: String;
    @Ouput() click = new EventEmitter();

    handleClick(event) {
        this.click.emit(event)
    }
}
```

```
// button.component.html
<button class="btn btn-default" [type]="type" (click)="handleClick($event)">
    <ng-content></ng-content>
<button>
```

```
// my.module.ts
import { NgModule } from '@angular/core'
import { ButtonComponent } from './components/button.component'

@NgModule({
    // ...
    declarations: [
```

😍 Styles Components 😍

```
import styled from 'styled-components';
const Button = styled.button`
  font-size: 1.5em;
  background: transparent;
  color: white;
  border: 2px solid #0099CC;
  border-radius: 6px;
  &:hover {
    color: red;
  }
`;
export default Button
```

vs Vue

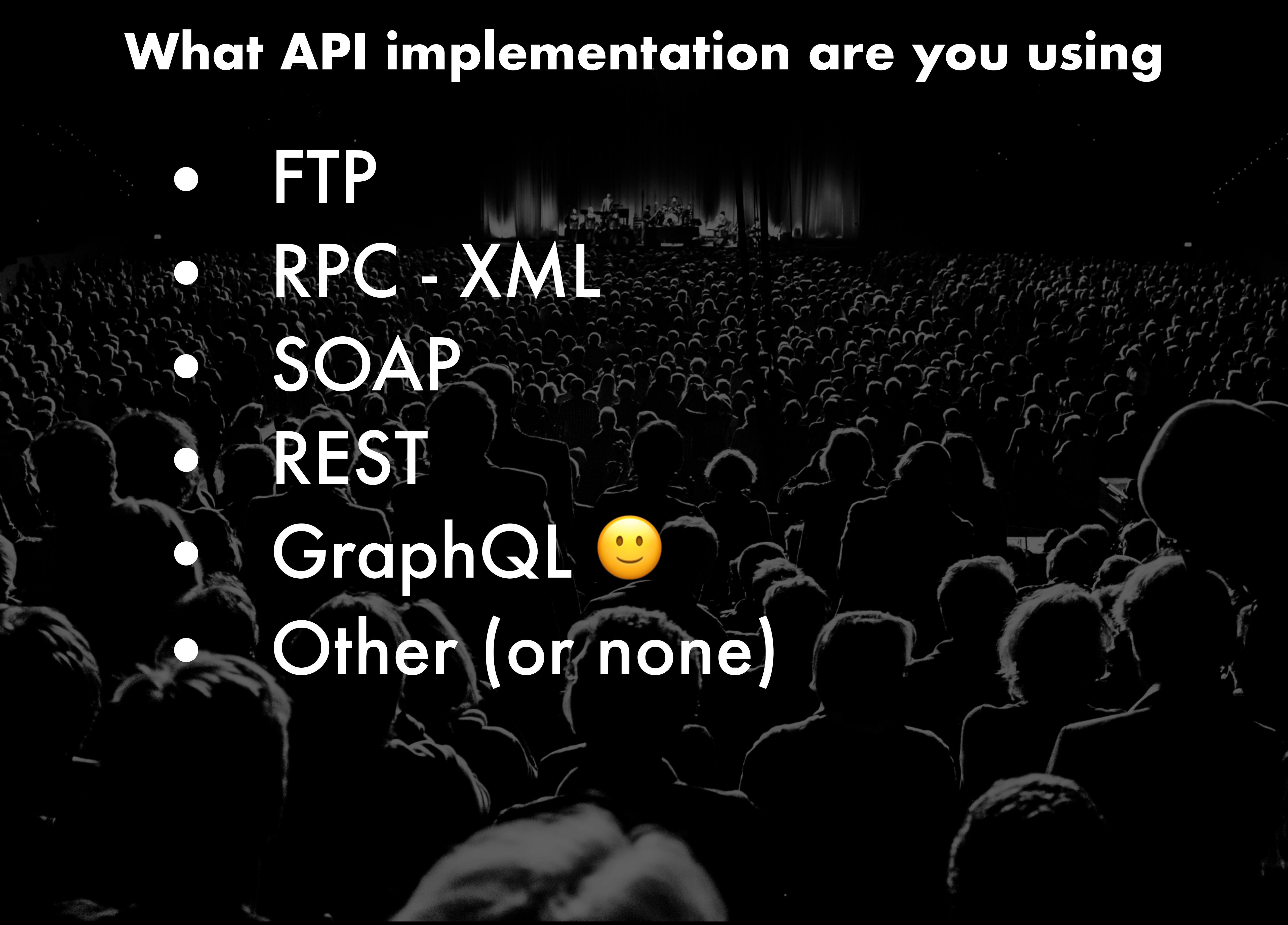
```
<template>
    <button class="btn btn-default" :type="type" @click="$emit($event)">
        <slot></slot>
    <button>
</template>
<script>
export default {
    props: ['type']
}
</script>
```

React 2018

- New core architecture: Fiber (100% backwards compatible)
- Faster Server Side Rendering & streaming
- Fragments & Portals
- Error bounderies
- Context API
- HOC vs render props

React 2019

- Async rendering & Suspense 😍
- https://build-mbfootjxoo.now.sh/
- https://www.youtube.com/watch?v=6g3g0Q_XVb4



What API implementation are you using

- FTP
- RPC - XML
- SOAP
- REST
- GraphQL 🙂
- Other (or none)

GraphQL

GraphQL: The next generation of API design

Usage of GraphQL

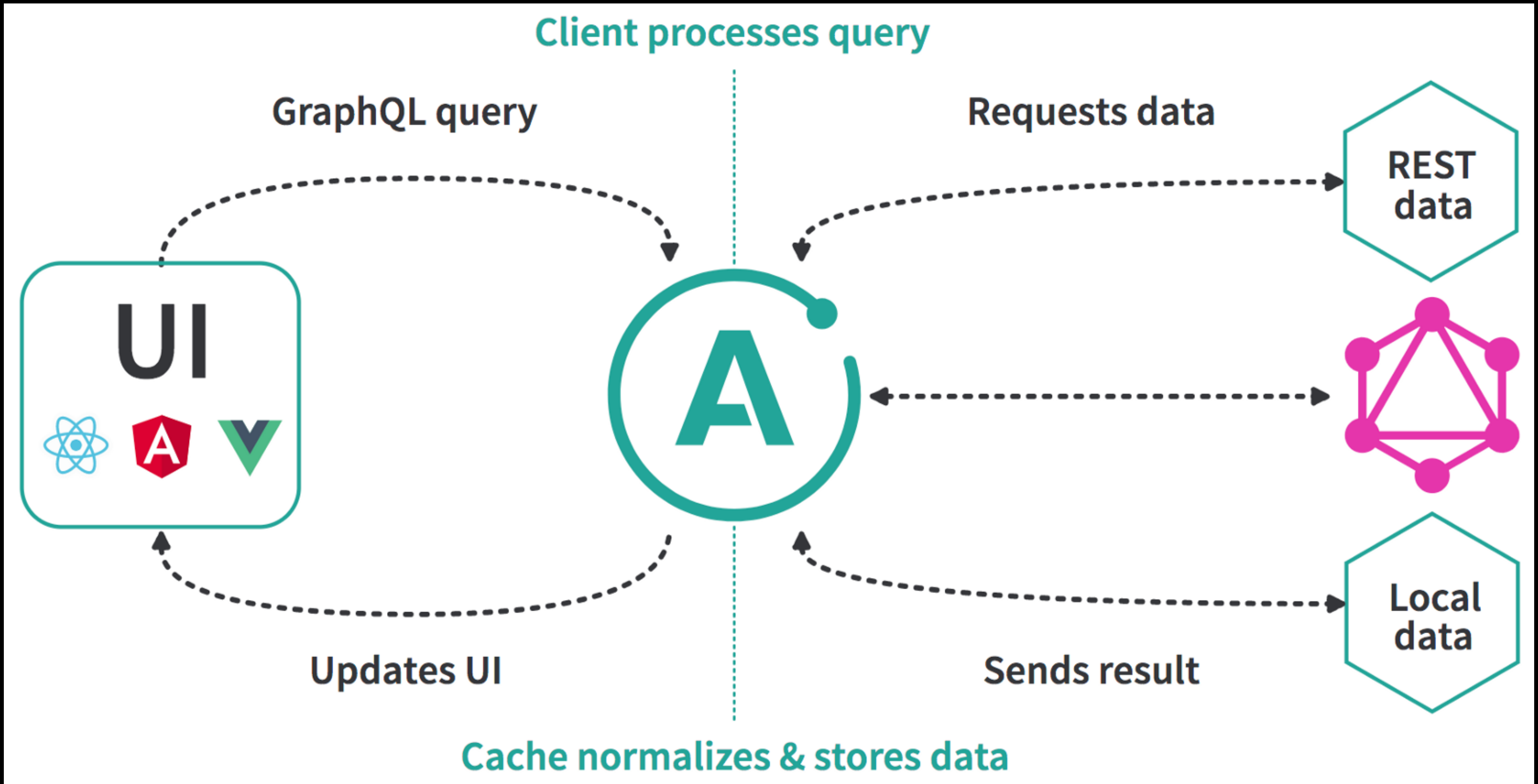
- Facebook
- Github, Amazon
- KLM
- PayPal
- AirBnb
- EggHead & Medium
- Pintrest
- IBM
- Walmart, Shopify & Starbucks
- American Express
- Sitecore, ContentFull, DatoCMS, WordPress, ...
- Microsoft (and not just because they bought Github)
- ...

Solutions of GraphQL

- Headless CMS (GraphCMS, wpgraphql, DataCMS, SiteCore, Mozaik, ...)
- Client libraries (Apollo Client, AWS Amplify, urql, ...)
- Server libraries (Apollo Server, Yoga, Prisma, ...)
- Managed Services (GraphCool, AWS AppSync, Apollo Engine)
- Platforms (JS, ruby, java, elixir, dotNet, php, python)
- New (Subscriptions, stitching, code generation)

GraphQL Stack

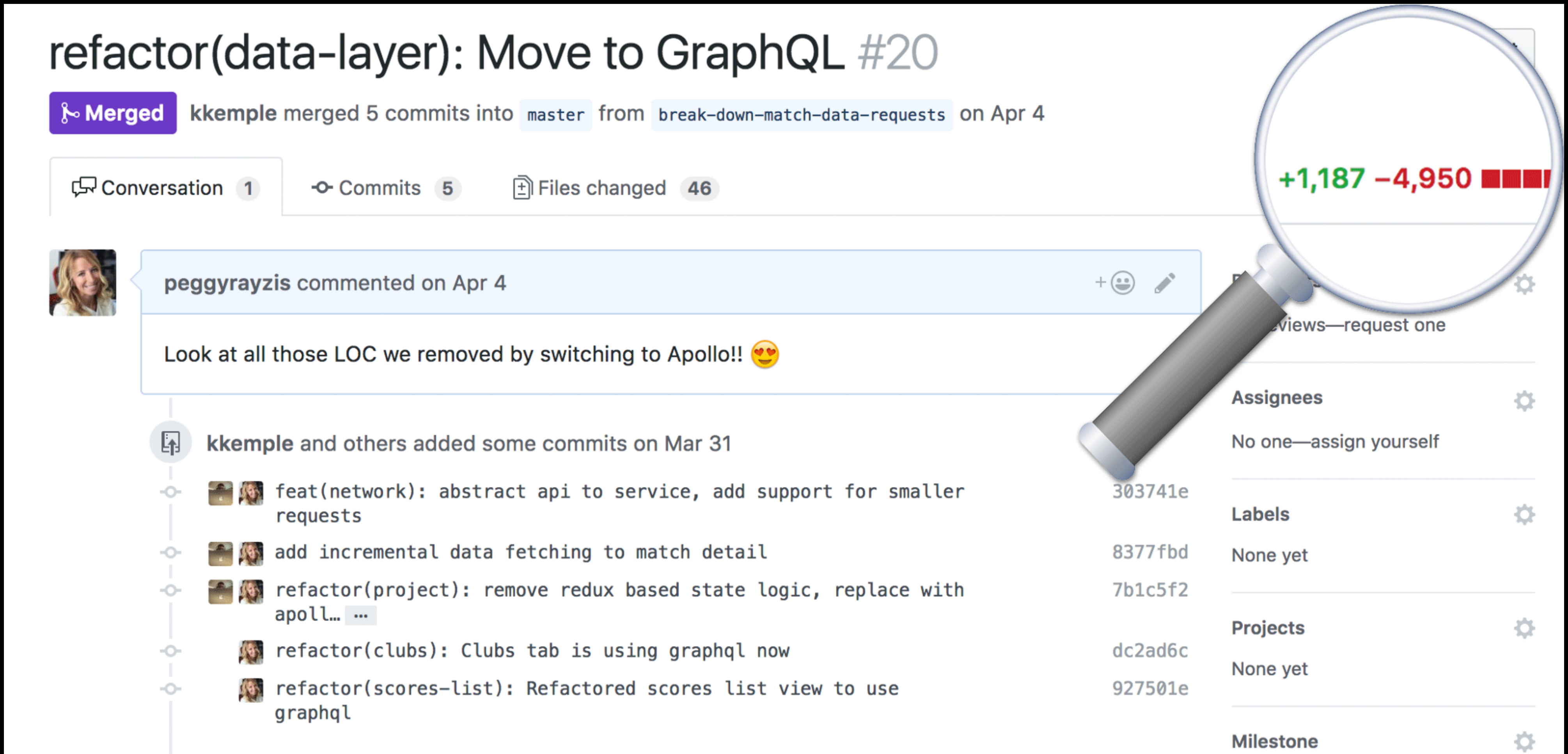
Apollo Client



When using Apollo & GraphQL, in 90% of the cases, you don't need Redux, MobX, ngrx, Observables or RxJS

More productive with GraphQL

Reducing our Redux code with React Apollo



refactor(data-layer): Move to GraphQL #20

Merged kkemple merged 5 commits into `master` from `break-down-match-data-requests` on Apr 4

+1,187 −4,950

peggyrayzis commented on Apr 4

Look at all those LOC we removed by switching to Apollo!! 😍

kkemple and others added some commits on Mar 31

```
feat(network): abstract api to service, add support for smaller    303741e
requests
add incremental data fetching to match detail                     8377fbd
refactor(project): remove redux based state logic, replace with   7b1c5f2
apoll… …
refactor(clubs): Clubs tab is using graphql now                   dc2ad6c
refactor(scores-list): Refactored scores list view to use         927501e
graphql
```

views—request one

Assignees

No one—assign yourself

Labels

None yet

Projects

None yet

Milestone

```
// Apollo Client
import gql from 'graphql-tag';
import { graphql } from 'react-apollo';

const GET_DOGS_QUERY = gql`{
    dogs {
        id
        breed
    }
}`;

@graphql(GET_DOGS_QUERY)
const Dogs = ({ onDogSelected, data: { loading, dogs, error } }) => {
  if (loading) return 'Loading...';
  if (error) return `Error! ${error.message}`;
  return (
    <ul>{dogs.map(dog => <li key={dog.id}>{dog.breed}</li>)}</ul>
  );
};
```

Learn Graphqh

https://www.howtographql.com/

What is your web app target?

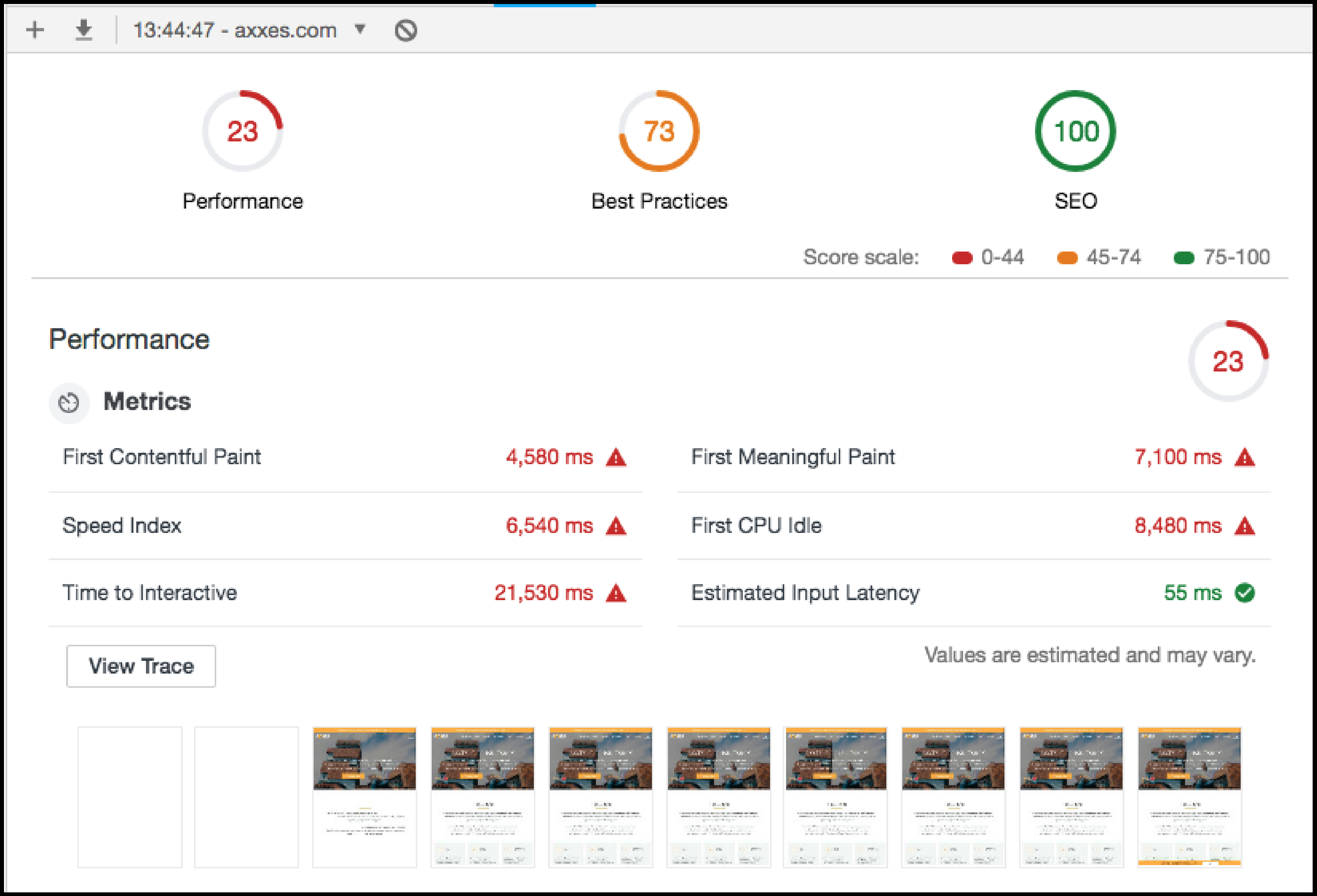
- Private Corporate Application
- Customer Facing Site/Apps
- Mobile Apps
- Other (or none)

Performance matters

For any web application

Lets test …

ordina.be



23 Performance

73 Best Practices

100 SEO

Score scale: 🟥 0-44 🟧 45-74 🟩 75-100

Performance

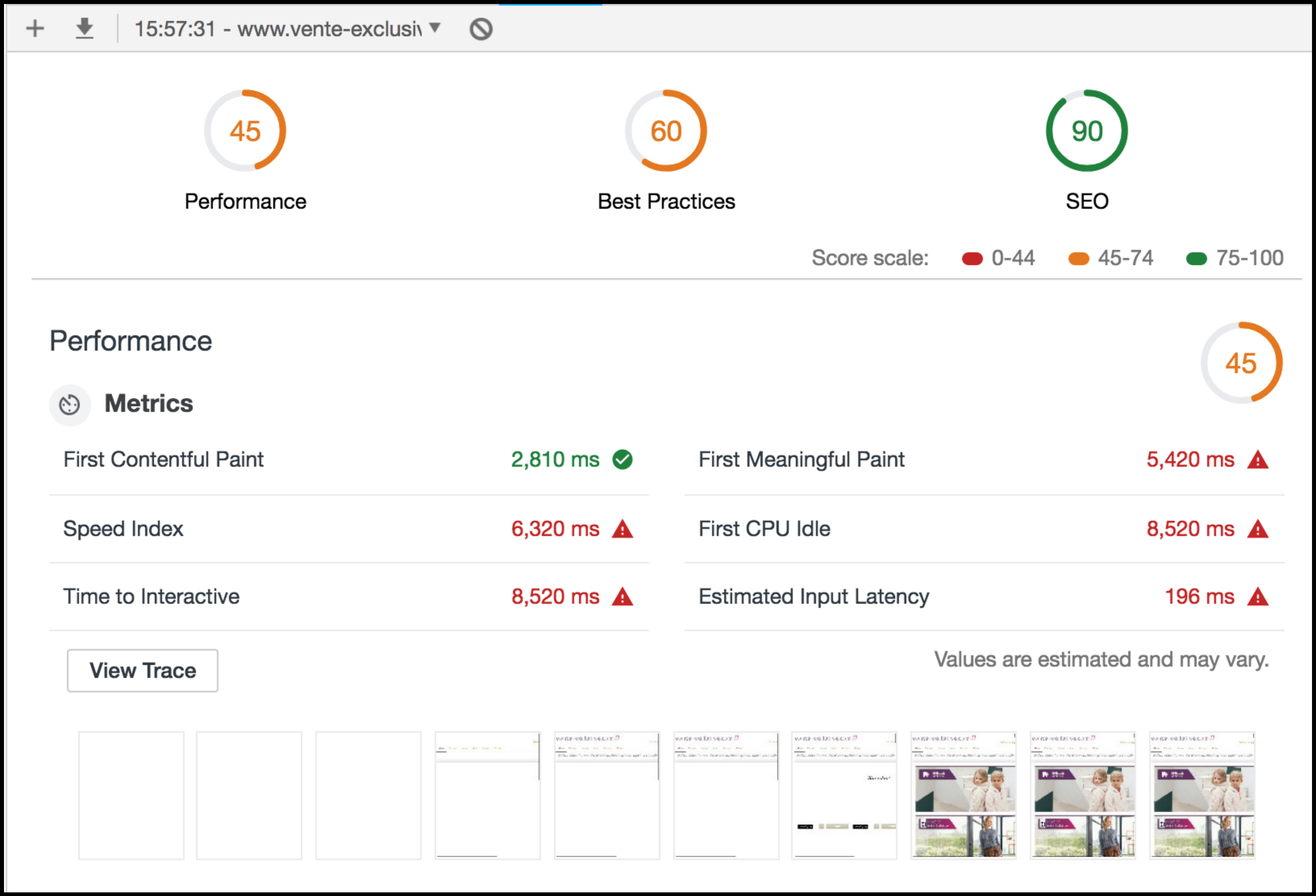
23

Metrics

First Contentful Paint	4,580 ms ▲	First Meaningful Paint	7,100 ms ▲
Speed Index	6,540 ms ▲	First CPU Idle	8,480 ms ▲
Time to Interactive	21,530 ms ▲	Estimated Input Latency	55 ms ✓

Values are estimated and may vary.

www.vente-exclusive.com



15:57:31 - www.vente-exclusi

45 Performance

60 Best Practices

90 SEO

Score scale: 🔴 0-44 🟠 45-74 🟢 75-100

Performance

45

Metrics

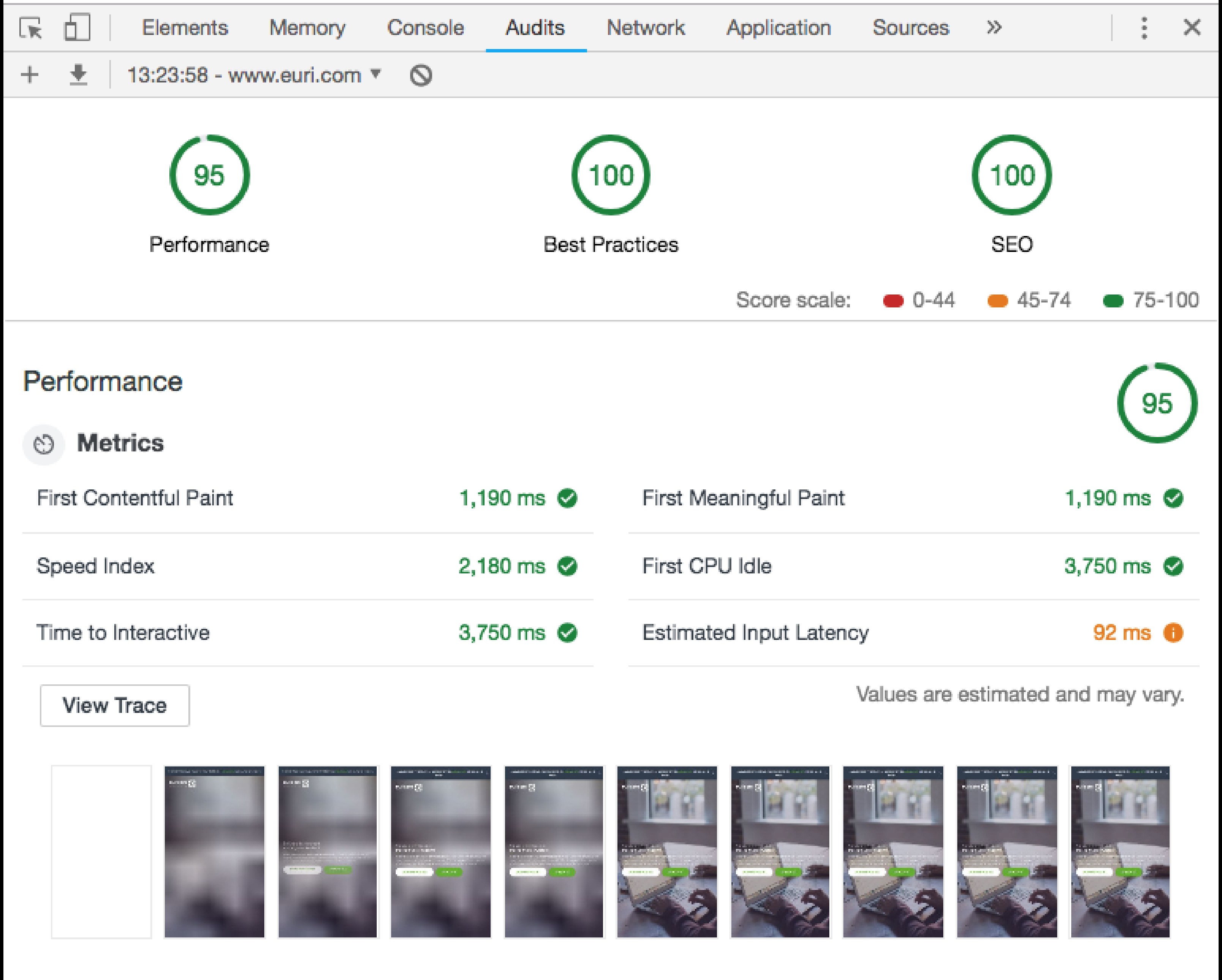
First Contentful Paint	2,810 ms ✓	First Meaningful Paint	5,420 ms ⚠
Speed Index	6,320 ms ⚠	First CPU Idle	8,520 ms ⚠
Time to Interactive	8,520 ms ⚠	Estimated Input Latency	196 ms ⚠

View Trace

Values are estimated and may vary.

euri.com ?

euri.com 👏



95 Performance

100 Best Practices

100 SEO

Score scale: 🟥 0-44 🟧 45-74 🟩 75-100

Performance

95

Metrics

First Contentful Paint	1,190 ms ✓	First Meaningful Paint	1,190 ms ✓
Speed Index	2,180 ms ✓	First CPU Idle	3,750 ms ✓
Time to Interactive	3,750 ms ✓	Estimated Input Latency	92 ms ⓘ

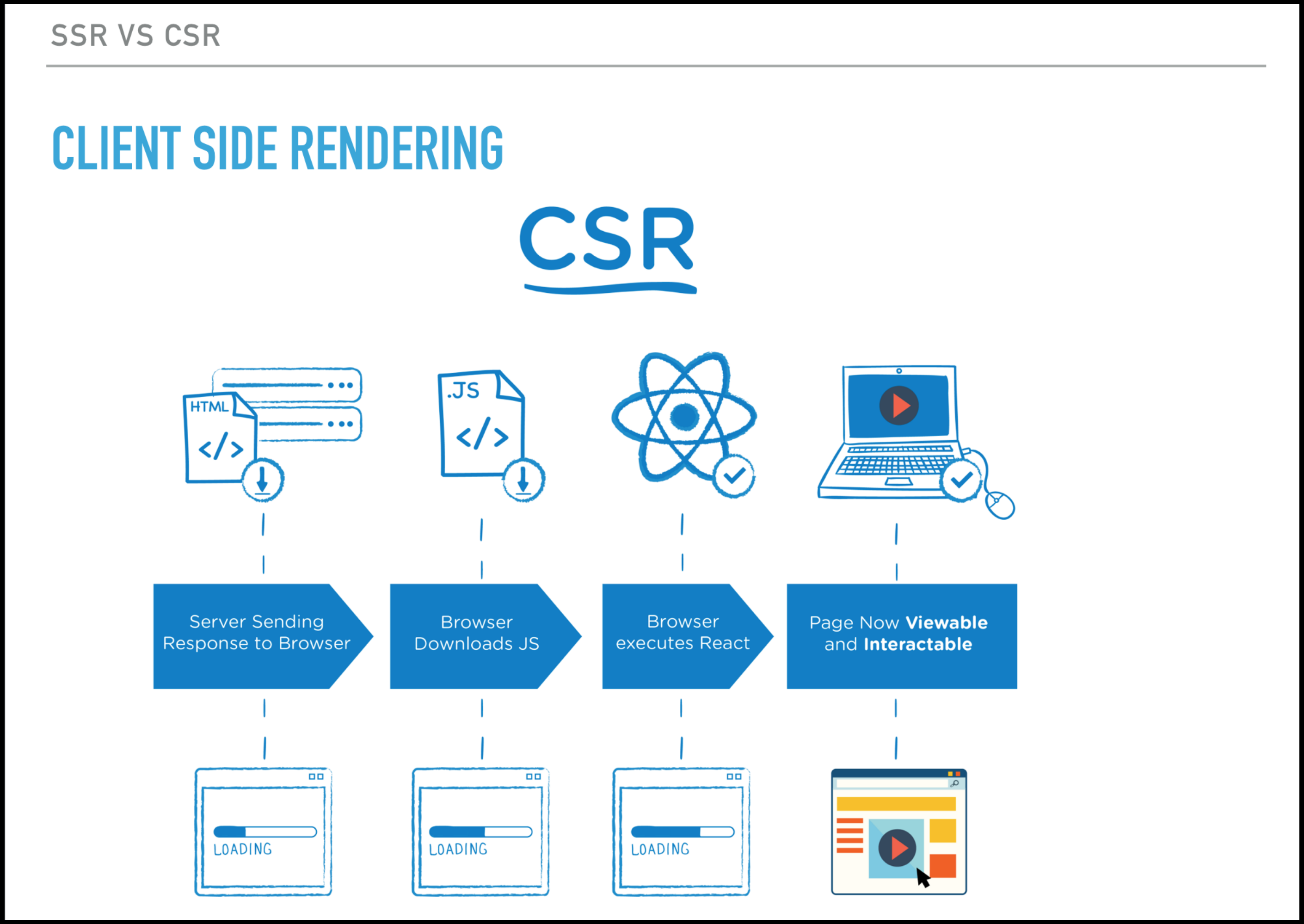
View Trace

Values are estimated and may vary.

Client Side, Server Side and Pre-rendering

Render what, where, ...?

Client Side Render (CSR)



SSR VS CSR

CLIENT SIDE RENDERING

CSR

Server Sending Response to Browser

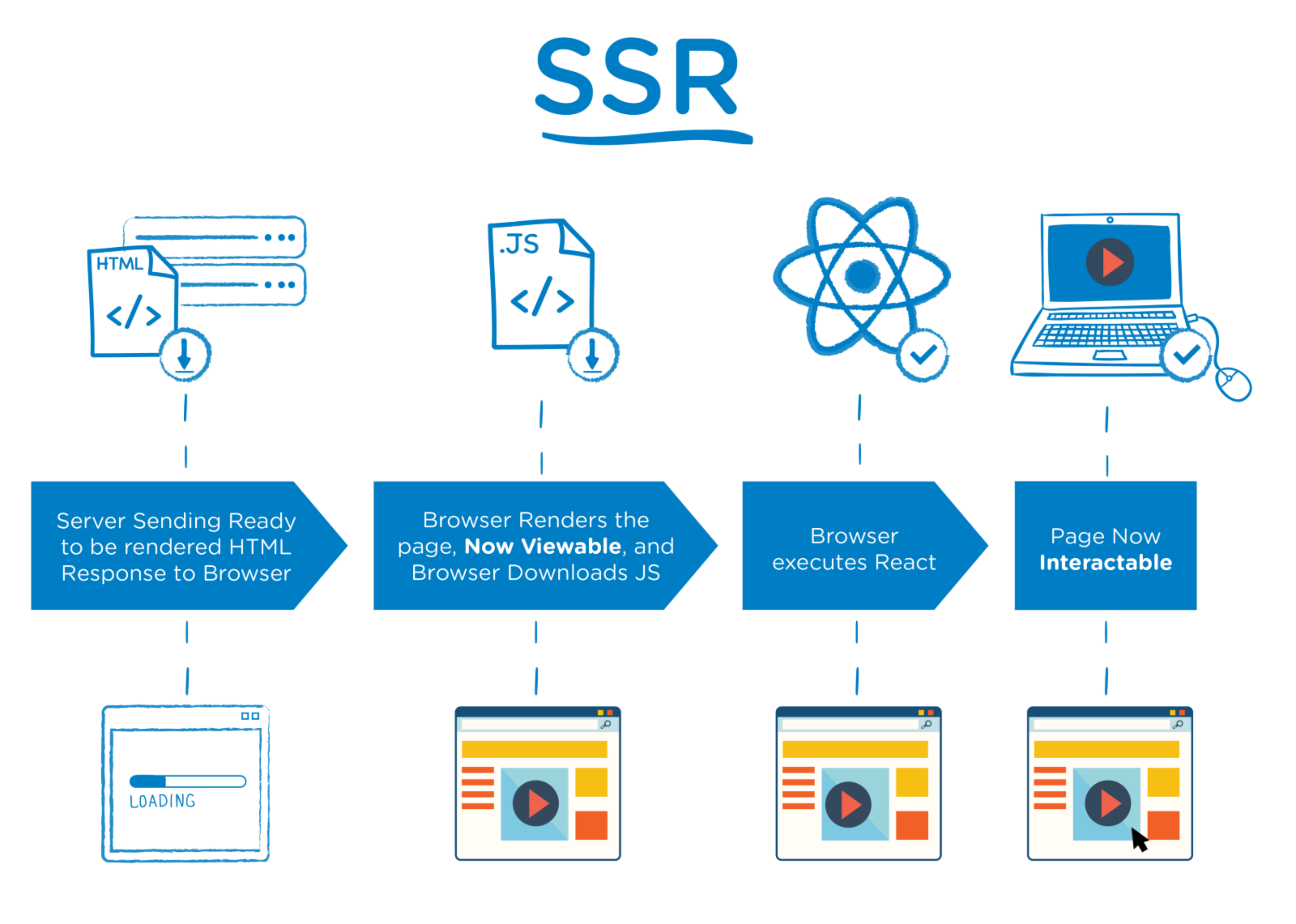
Browser Downloads JS

Browser executes React

Page Now **Viewable** and **Interactable**

You standard SPA application.

Server Side Render (SSR)


SSR
HTML
.JS
Server Sending Ready to be rendered HTML Response to Browser
Browser Renders the page, **Now Viewable**, and Browser Downloads JS
Browser executes React
Page Now **Interactable**
LOADING

Improve SEO and noticeable performance.

Server Side Render (SSR)

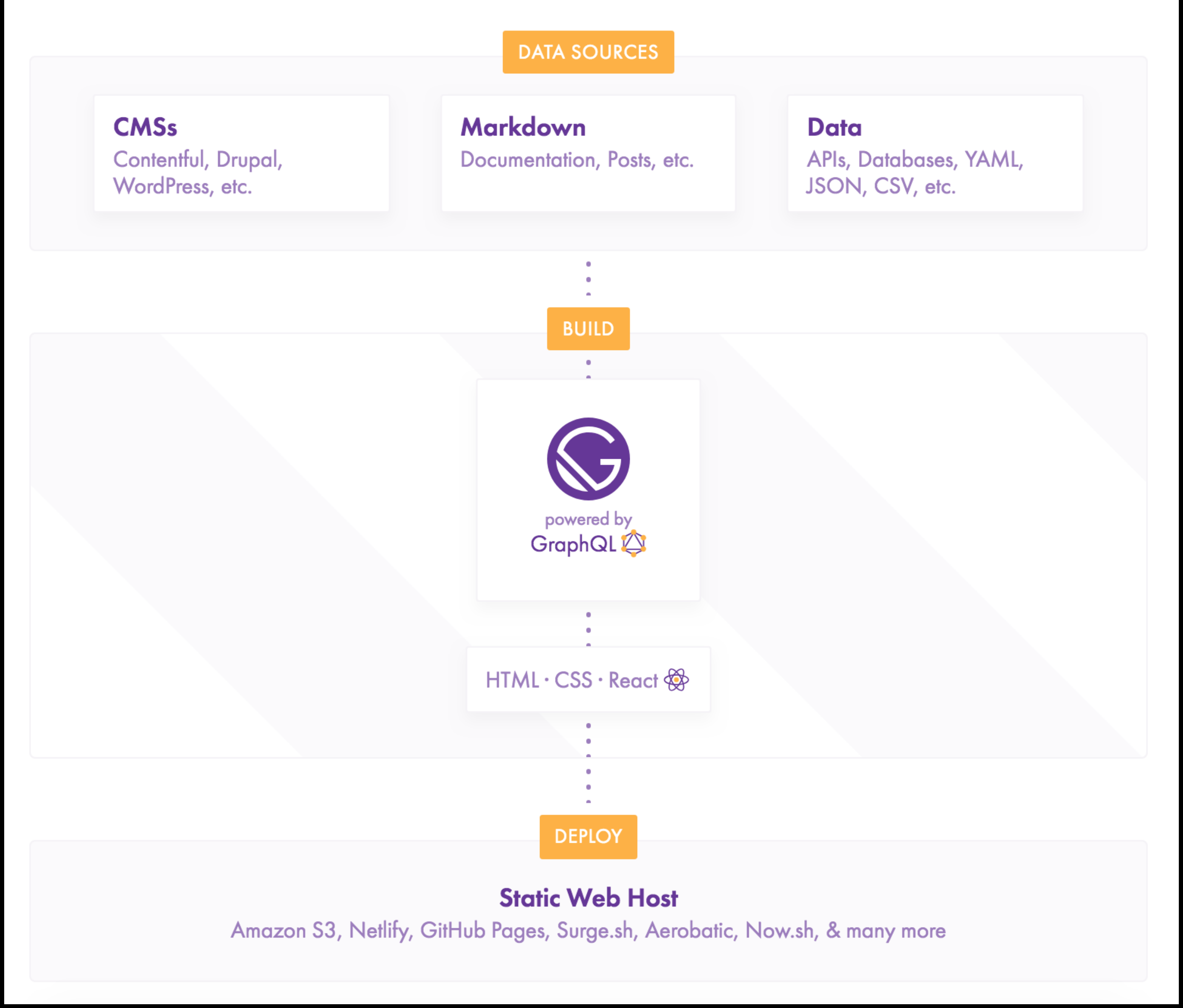
- Angular Universal 😥
- Next 6.x - React SSR Done Right
- Nuxt 1.x - Universal Vue.js Apps

Pre-rendering

Render the complete site at build time

- Jekyll & Hugo
- Gatsby - 🚀 Blazing fast site generator
- VuePress - Vue Static Site Generator

Gatsby


DATA SOURCES

CMSs
Contentful, Drupal, WordPress, etc.

Markdown
Documentation, Posts, etc.

Data
APIs, Databases, YAML, JSON, CSV, etc.

BUILD

powered by GraphQL

HTML · CSS · React

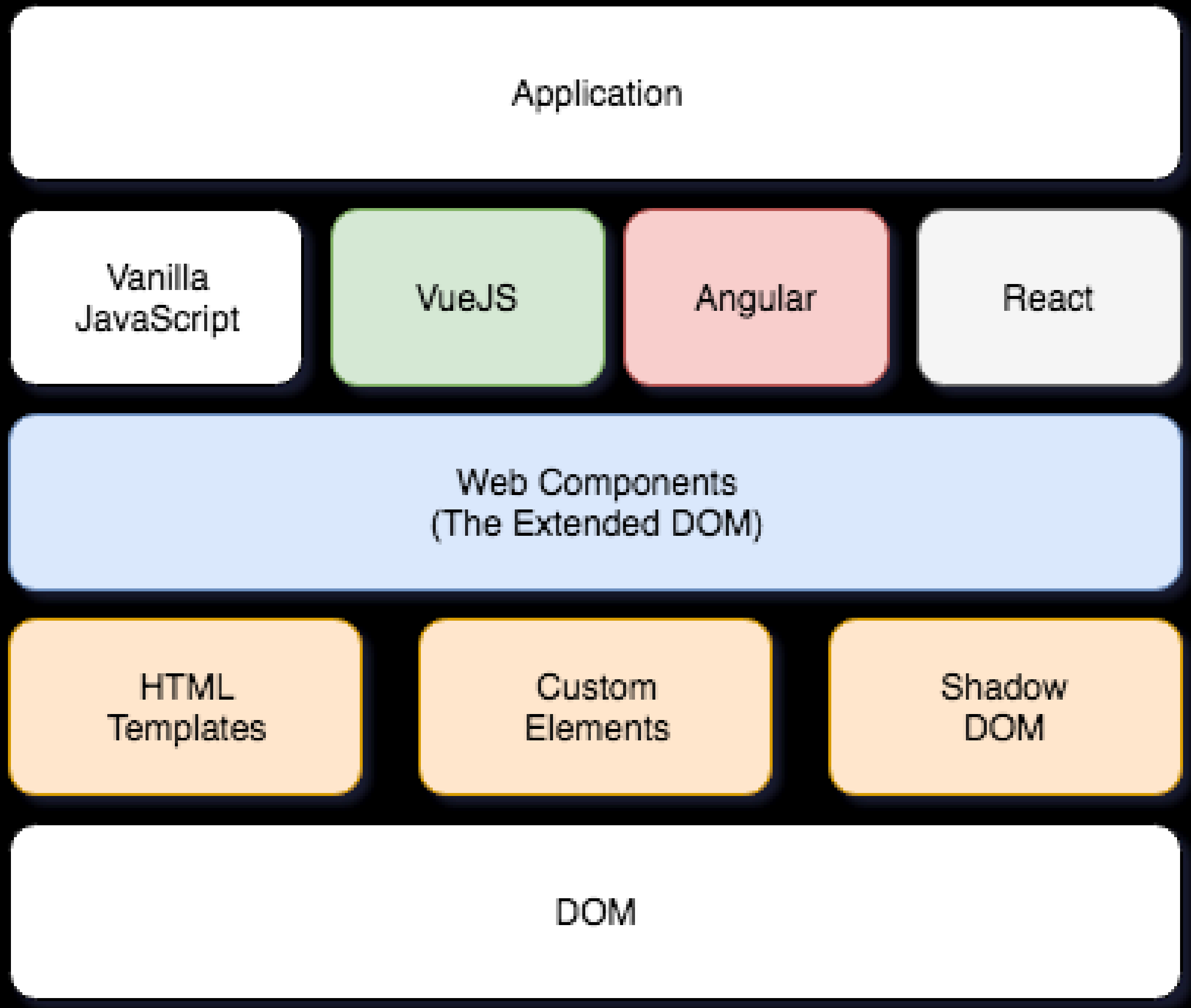
DEPLOY

Static Web Host
Amazon S3, Netlify, GitHub Pages, Surge.sh, Aerobatic, Now.sh, & many more

Web-Components

The building blocks of the future

Web-Components



Application

Vanilla JavaScript

VueJS

Angular

React

Web Components
(The Extended DOM)

HTML Templates

Custom Elements

Shadow DOM

DOM

Creating Web Components

- (NOT YET) Angular Elements
- Polymer Library v3
- Ionic StencilJS
- VueJS

StencilJS

```
import { Component, Prop } from '@stencil/core';

@Component({
  tag: 'my-first-component',
  styleUrl: 'my-first-component.scss'
})
export class MyComponent {
  @Prop() name: string;
  render() {
    return (
      <p>
        My name is {this.name}
      </p>
    );
  }
}
```

VueJS

```
<template>
   <p>prop value: {{myProp}}</p>
</template>
<script>
export default {
  props: ['myProp'],
};
</script>
```

Any vue component can be exported as web-component

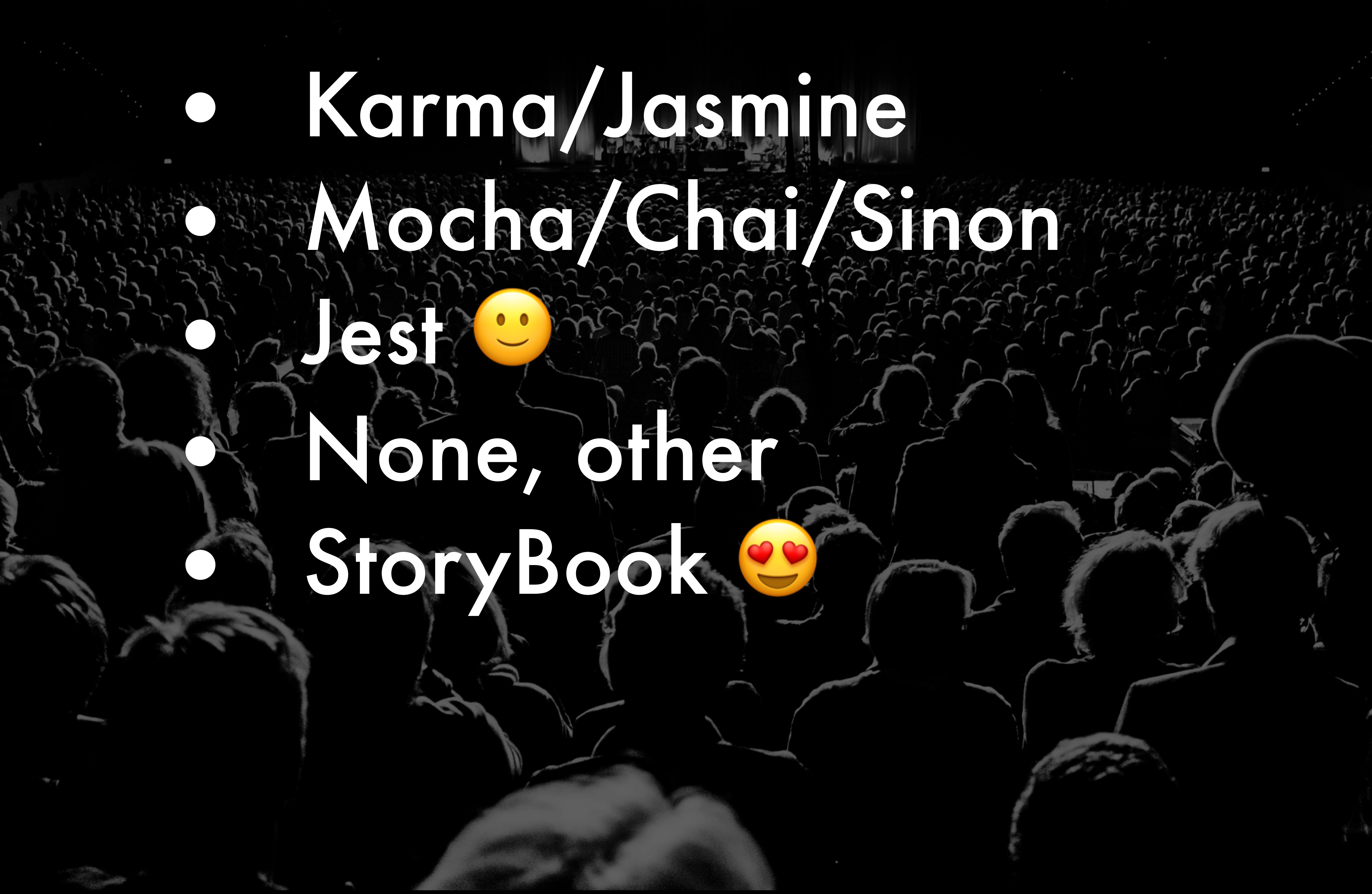
```
# create web-component
vue build ./src/components/Sample.vue --target wc --name my-sample
```

VueJS

Use

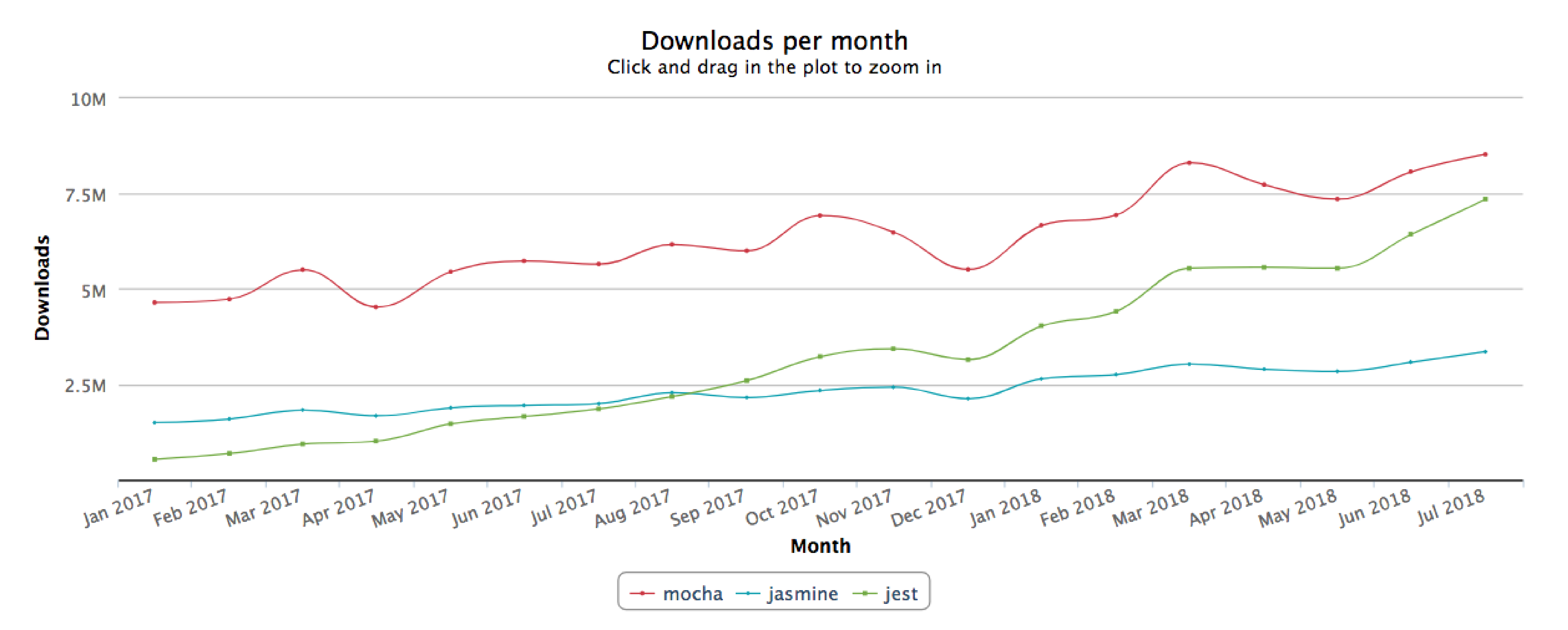
```html
<html>
    <head>
        <title>my-sample demo</title>
        <script src="https://unpkg.com/vue"></script>
        <script src="./my-sample.js"></script>
    </head>
    <body>
        <my-sample prop="hello"></my-sample>
    </body>
</html>
```

What testing framework are you using?



- Karma/Jasmine
- Mocha/Chai/Sinon
- Jest 🙂
- None, other
- StoryBook 😍

Jest is the rising star



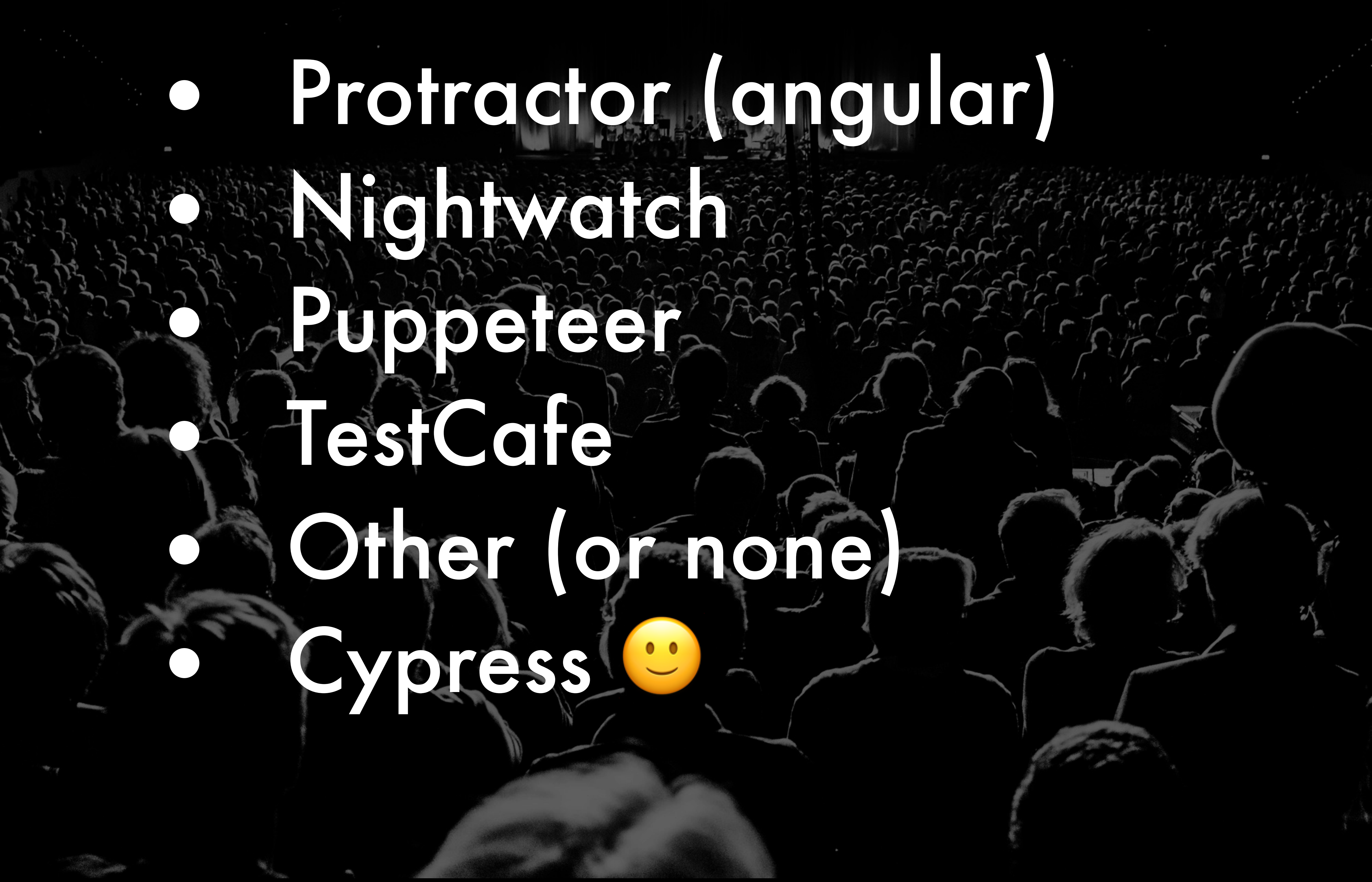
- Default on React project
- Preferred on VueJS projects
- Snapshot testing is awesome

Storybook is your new friend

- Component Driven Development
- Component Library
- Visual Component Development & Testing
- Documentation
- Available for: Angular, React, VueJS, Polymer, HTML/CSS

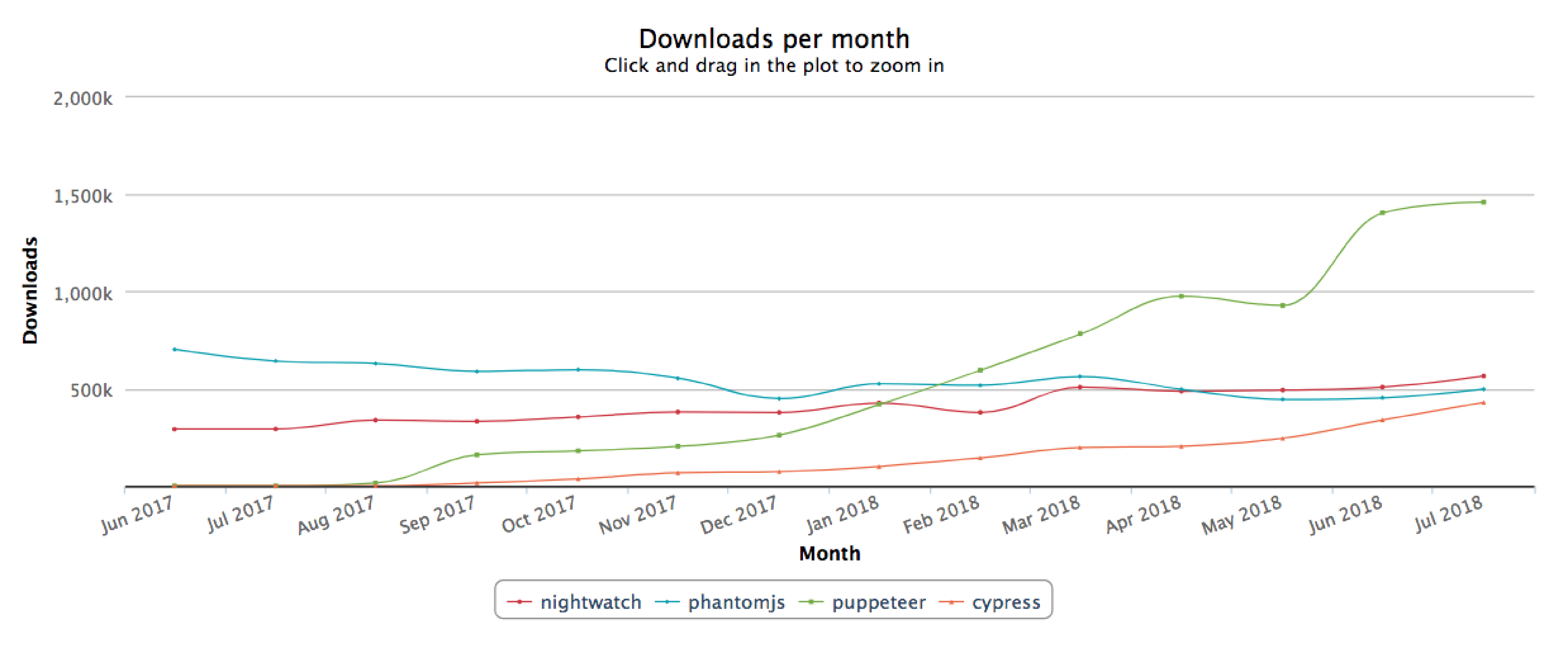
Live Sample - ReactLive Sample - Angular

What e2e test framework are you using?



- Protractor (angular)
- Nightwatch
- Puppeteer
- TestCafe
- Other (or none)
- Cypress 🙂

Rizing stars: Puppeteer & Cypress

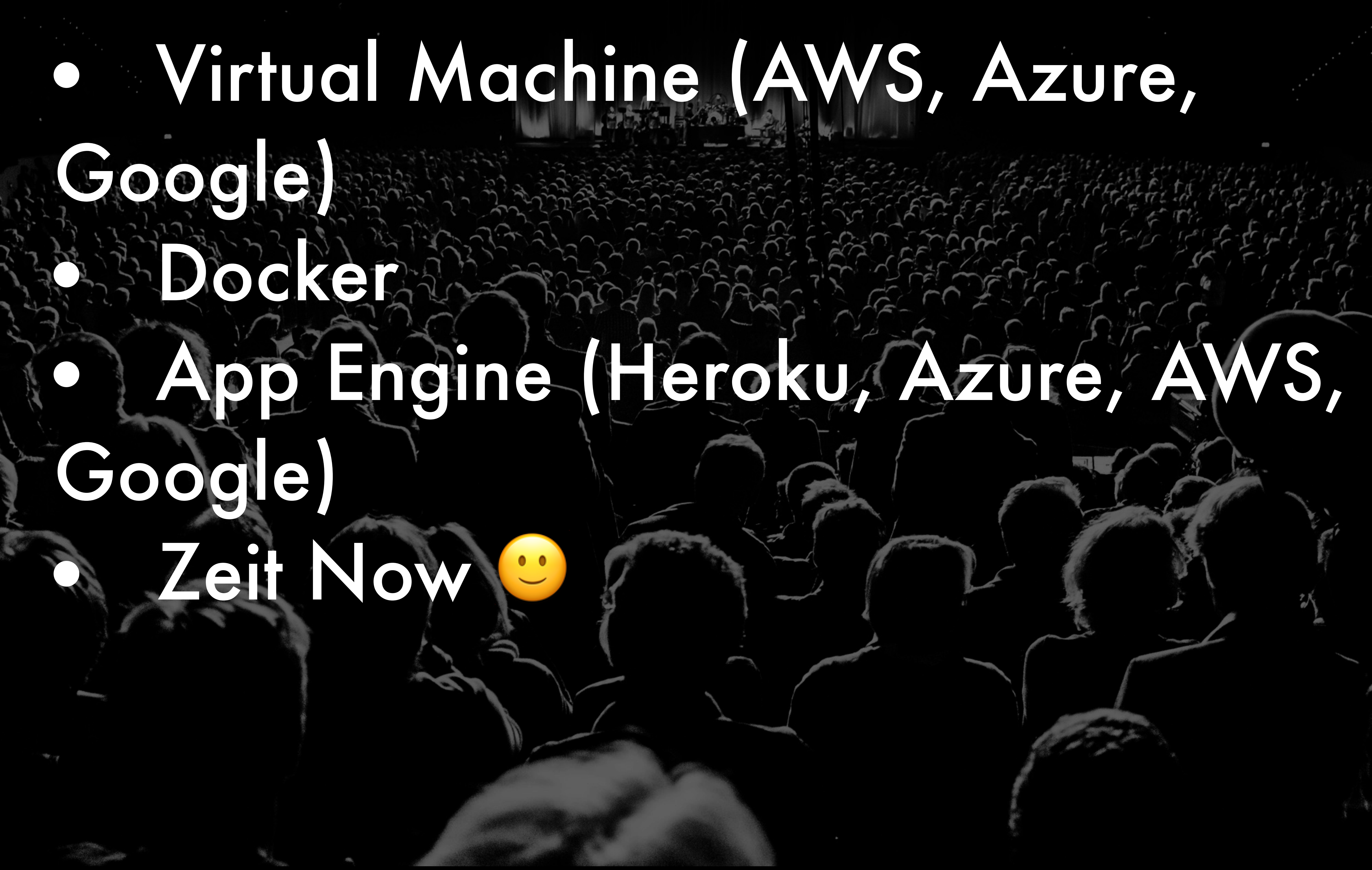


cypress.io

- Fast, easy and reliable testing
- Watch and Auto reload
- Time travel
- For anything that runs in a browser

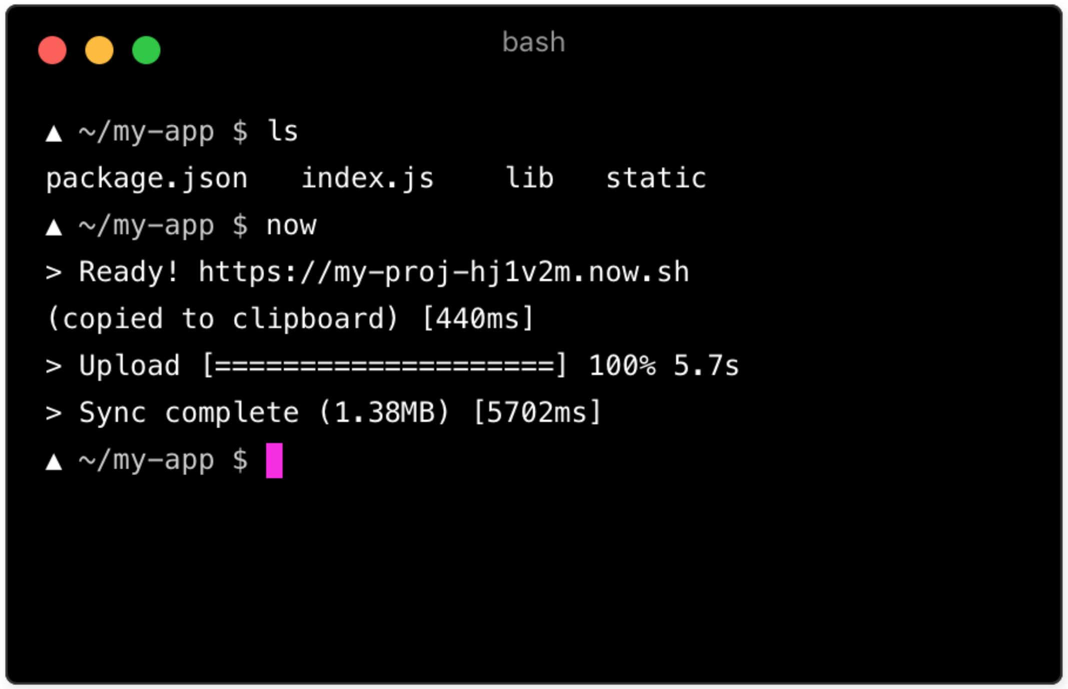
Cypress

Where do you deploy your NodeJS app



- Virtual Machine (AWS, Azure, Google)
- Docker
- App Engine (Heroku, Azure, AWS, Google)
- Zeit Now 🙂

Now – Global Serverless Deployments



```
▲ ~/my-app $ ls
package.json   index.js    lib   static
▲ ~/my-app $ now
> Ready! https://my-proj-hj1v2m.now.sh
(copied to clipboard) [440ms]
> Upload [====================] 100% 5.7s
> Sync complete (1.38MB) [5702ms]
▲ ~/my-app $
```

☑ **Docker**

```
$ my-app/ ls
Dockerfile server.go
$ my-app/ now
```

☑ **Node.js**

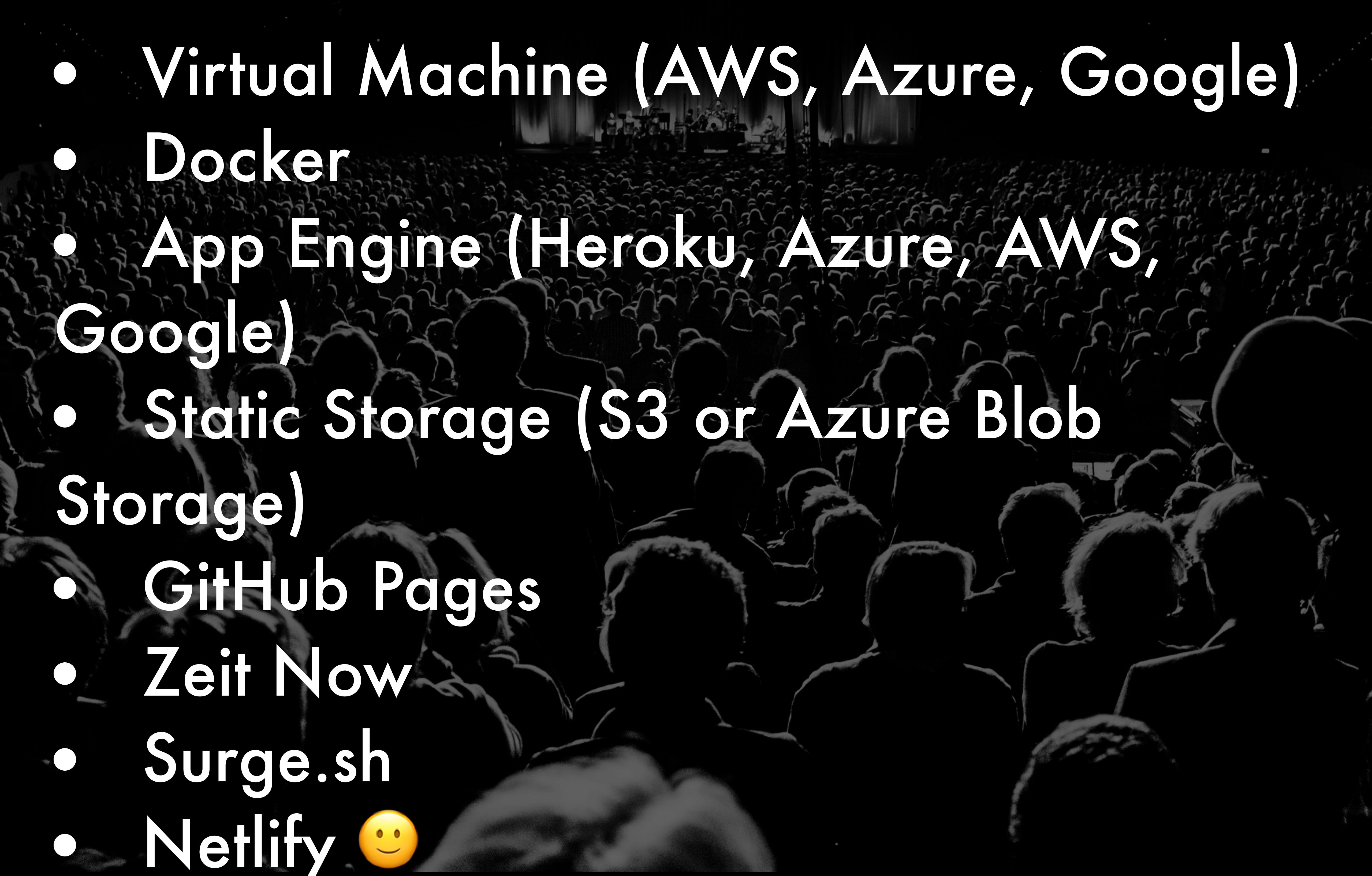
```
$ my-api/ ls
package.json index.js
$ my-api/ now
```

☑ **Static Websites**

```
$ my-site/ ls
index.html logo.png
$ my-site/ now
```

Now

Where do you deploy your static app's



- Virtual Machine (AWS, Azure, Google)
- Docker
- App Engine (Heroku, Azure, AWS, Google)
- Static Storage (S3 or Azure Blob Storage)
- GitHub Pages
- Zeit Now
- Surge.sh
- Netlify 🙂

netlify

- CDN Hosting
- HTTPS is automatic
- Full cache control
- Automate build & deployment
- Identity, Functions, Forms
- Low pricing strategy

Netlify

Honorable Mentions

- DateFns (a modern date library)
- Babel 7.0
- TypeScript 3.0
- LogLevel (universal logging)
- Capacitor (replaces Cordova)
- React Native (still strong)
- Flutter (vs React Native)

Thank You 👋

https://mjr-javascript-trends-2018.now.sh/

Credits

```
Built with MDX Deck
```

```
Deployed on now.sh
```