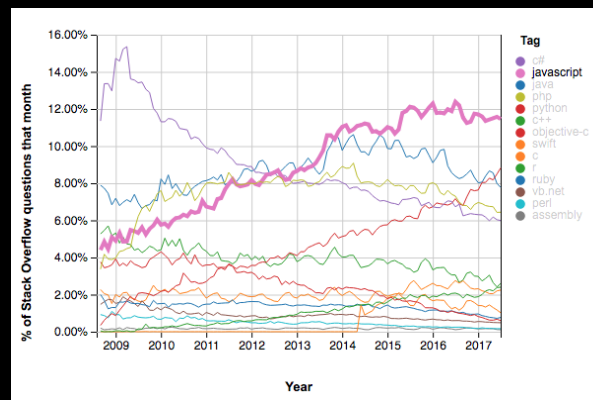


New trends in javascript development (2018)



By Peter Cosemans

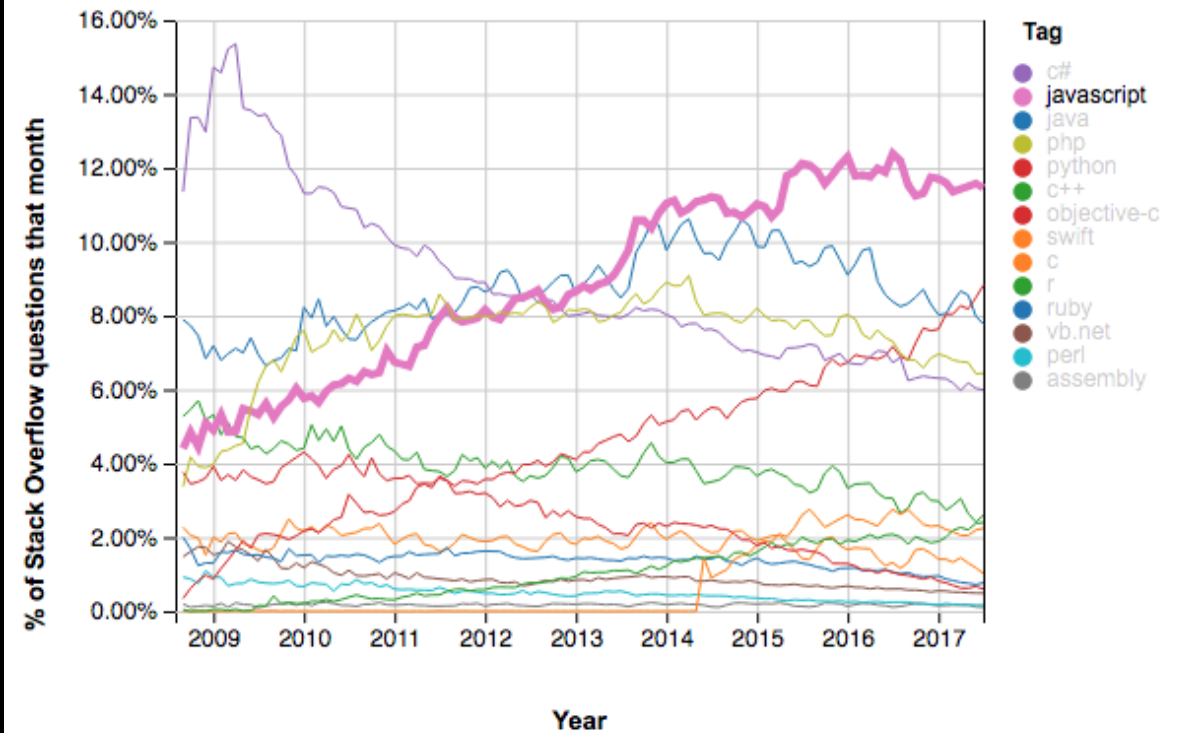
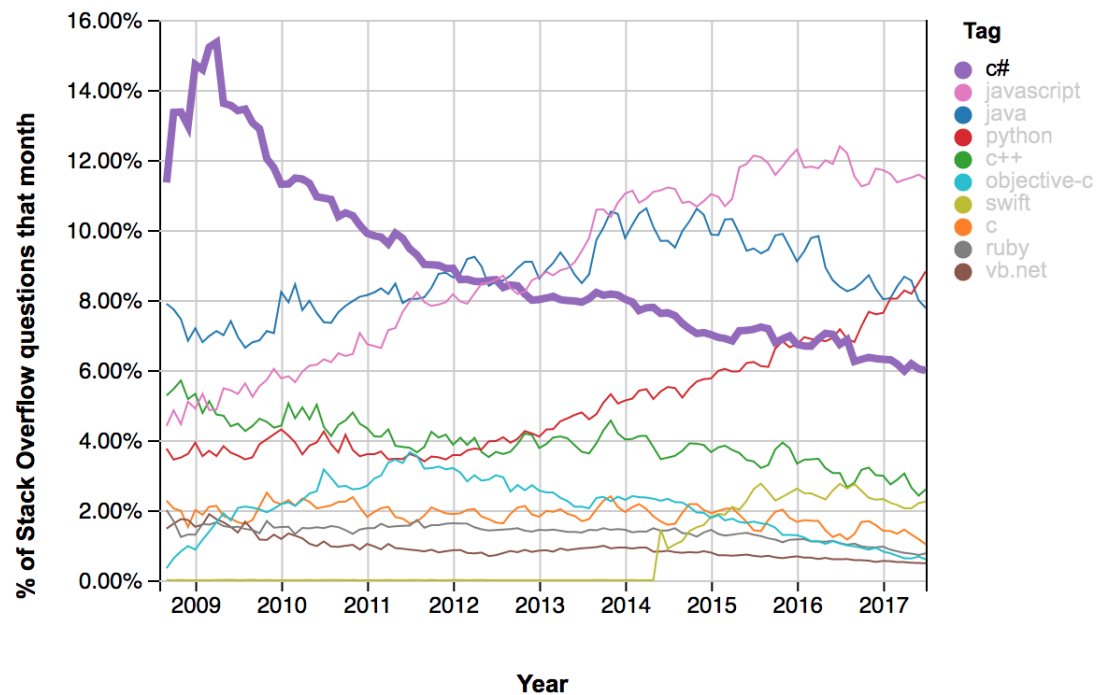
The Top Trends

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

Javascript the language

Still getting bigger every year

C# vs JavaScript

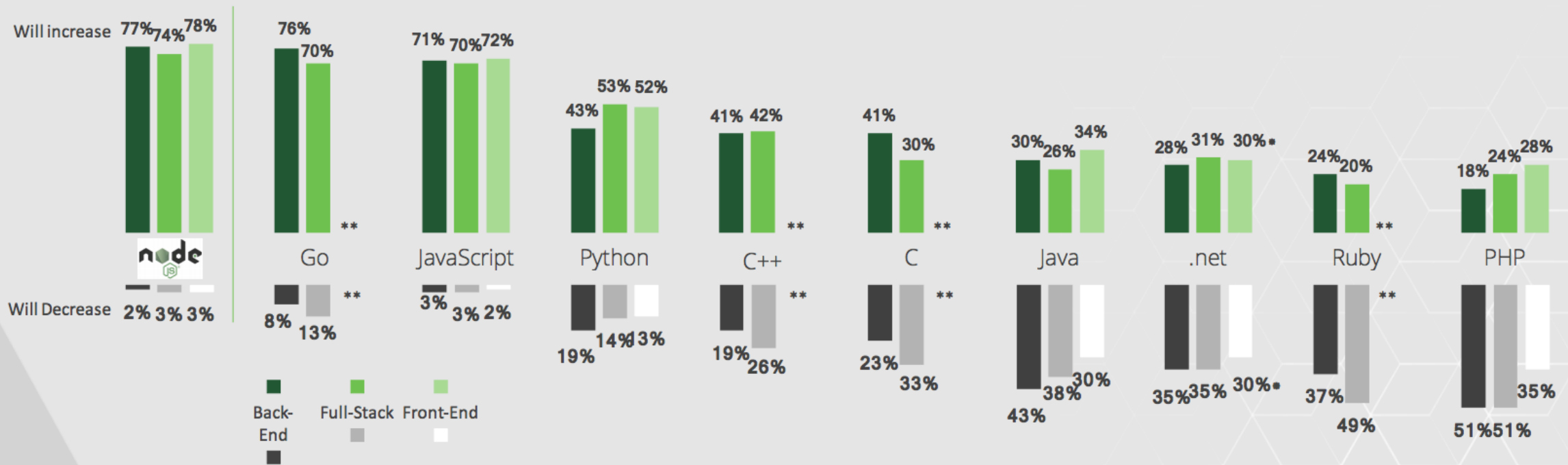


Statistics from StackOverflow

JavaScript Is Still One of the Fastest-growing Languages

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 JS near you

ES2018 - ES2019 - TS3.0

- Async/Await
- Promise finally
- Class fields
- Private fields/methods
- Dynamic import
- Numeric separators
- Others...

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;  
}
```

Promise Finally

(ES stage-4, TS2.7)

```
promise  
  .then(result => {...})  
  .catch(error => {...})  
  .finally(() => {...});
```


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

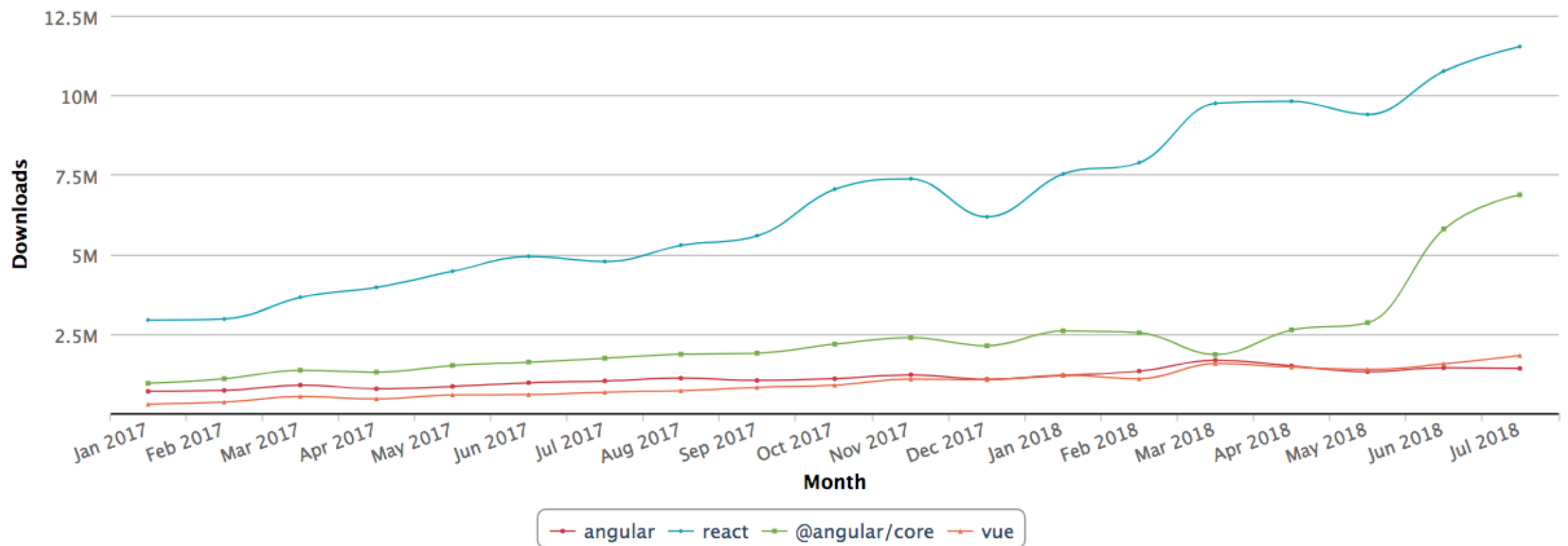
What front-end framework are you using?

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

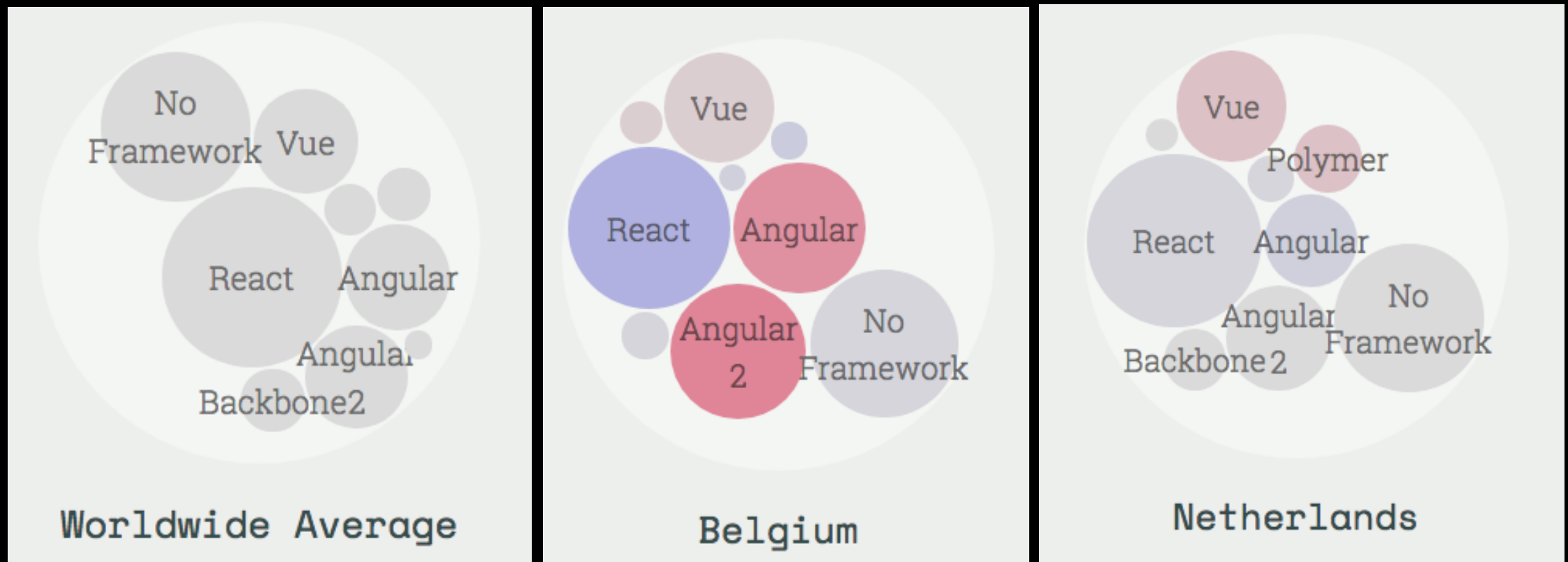
Npm Downloads

Downloads per month

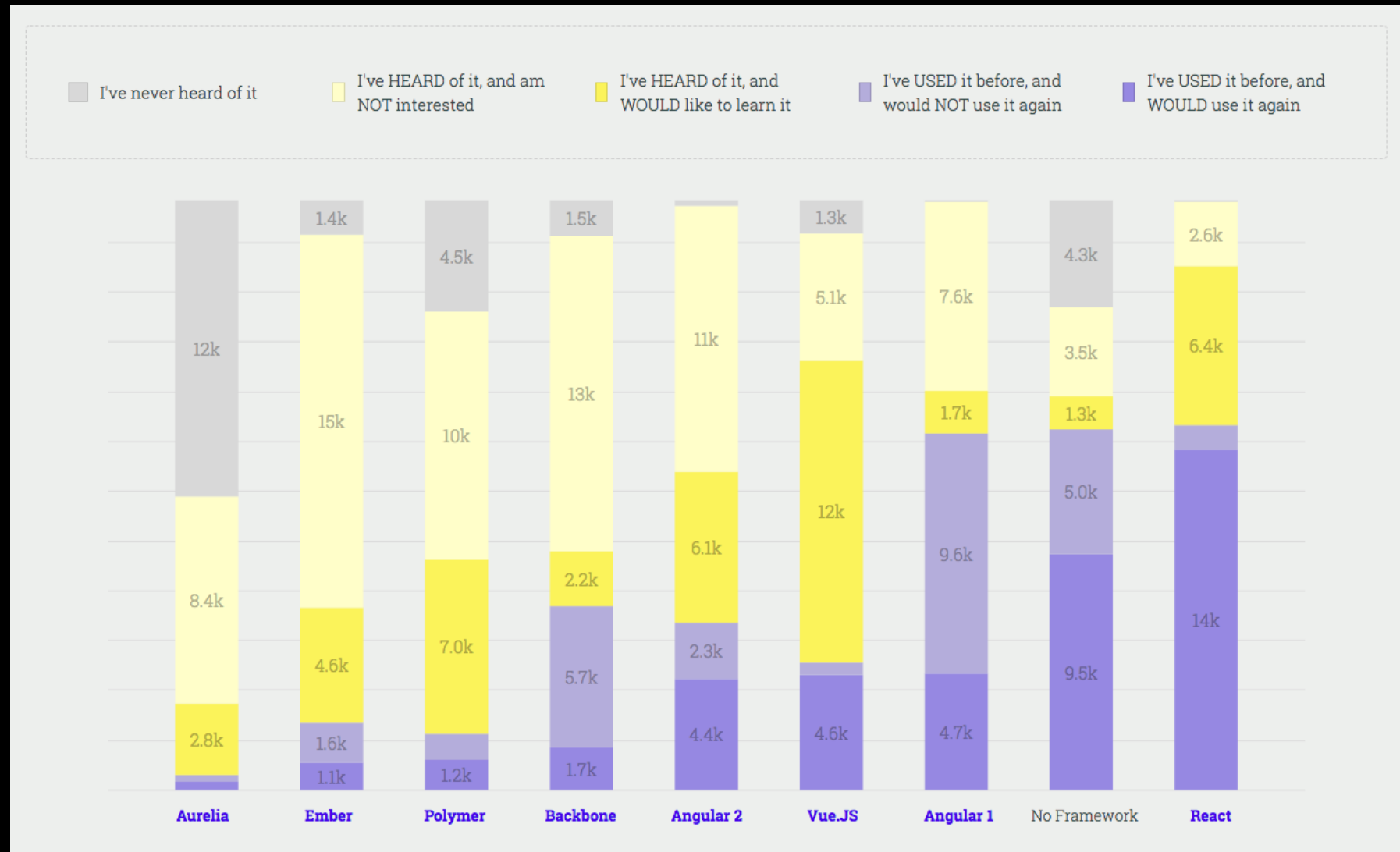
Click and drag in the plot to zoom in



Frameworks Usage - 2017Q4



Use & Intrest



- High Intrest 🤔 - VueJS
- Not Use Again 😞 - AngularJS, Backbone & (Angular2)
- Use Again 😊 - React

Poll by Todd Motto



Todd Motto™ @toddmotto · Sep 13

Which framework do you prefer?



30% Angular

51% React ✓

19% Vue

7,701 votes • 3 hours left



64



107



123



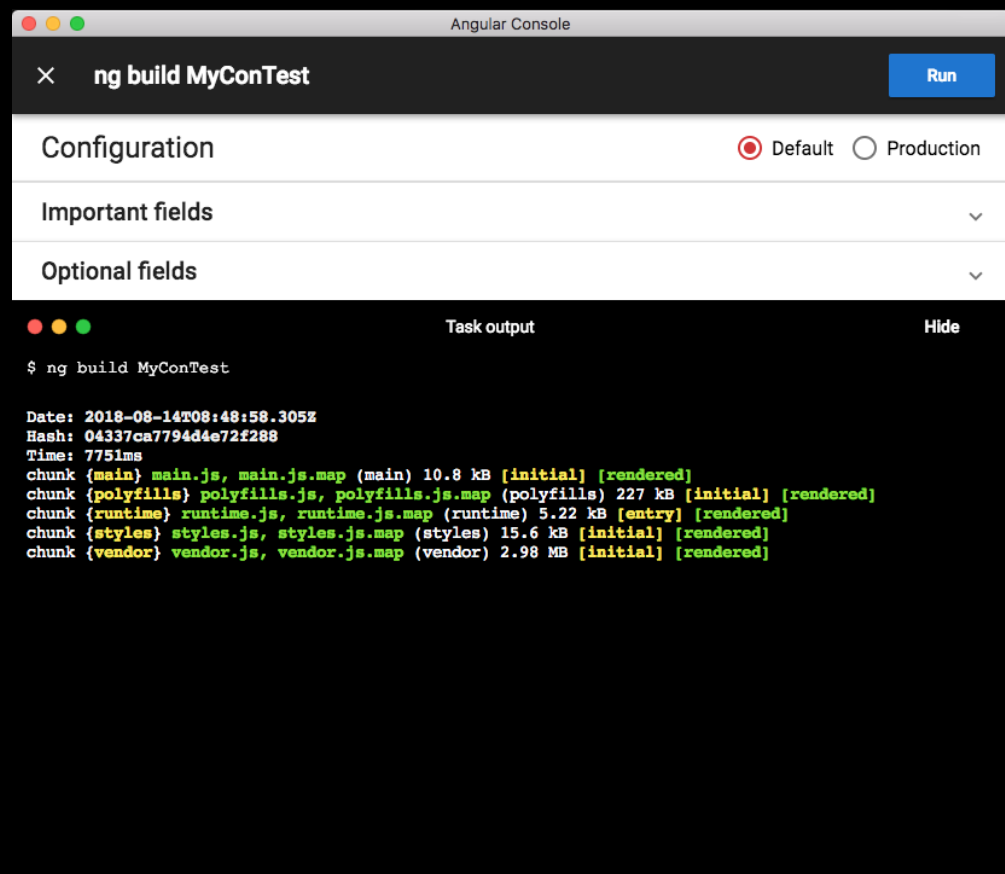
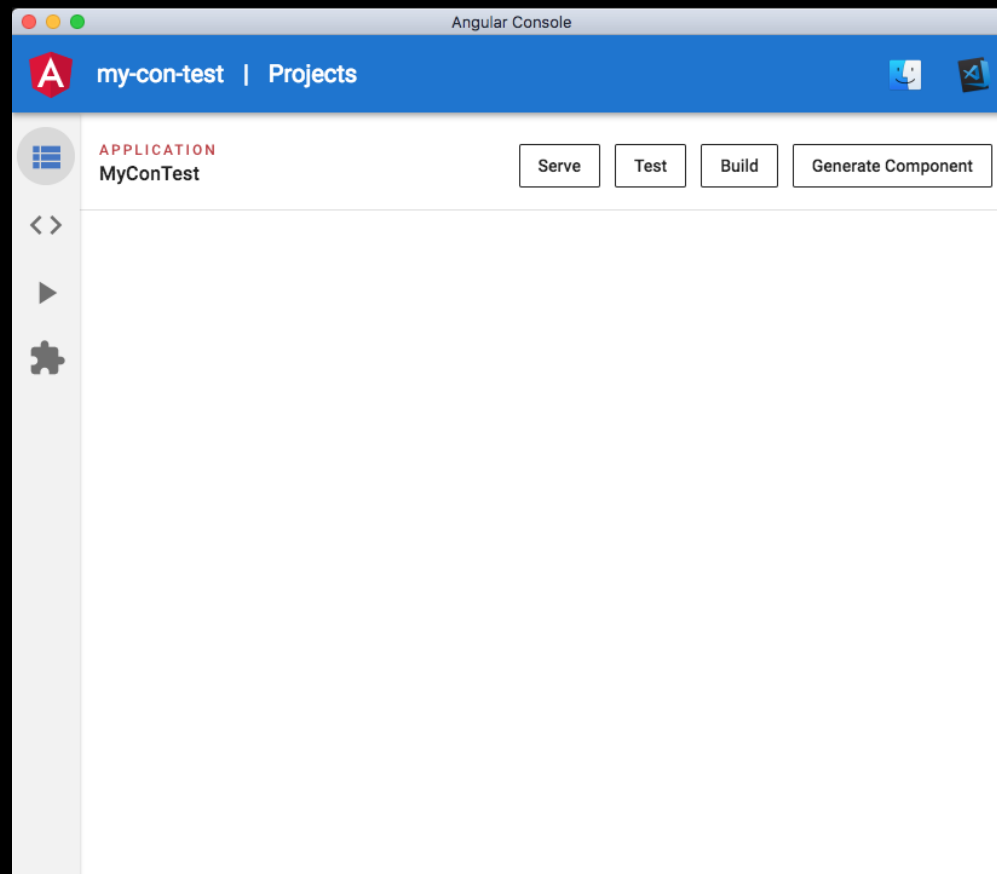
Angular

It's just Angular

Angular 6.0

- A new CLI (workspaces, library, schematics, webpack 4, ...)
- Smaller bundle size
 - Tree-shakable RxJS 6.0 & providers
 - Ivy Renderer
- Consistent versions (cli, material, router, ...)
- Angular Elements (web components)
- Soon: Angular 7 - Very small bundle sizes (95-99% reduction)

Angular Console



Angular 6.0 - 7.0



- Complex & Closed CLI (where is my eject?)
- Breaking changes (RxJS, CLI & ng-compiler)
- Angular Element != Web Components
- Ivy (preview)
- Angular Console (no added value)
- Angular 7: Bazel, 418 modules split... @aiStore & @angular/mine

VueJS

Fast and simple

VueJS 2018

- Full Prettier support (for vue files)
- eslint-plugin-vue (errors in templates)
- vue-test-utils
- Vue Devtools 5.0 (routing & perf tab, editable Vuex state)
- @vue/cli: v3.0 🙌 (webpack 4, UI, app/lib, 100% customizable)

@vue/cli - ui (console)

myApp2 ▾

Project tasks

🔍

⚙️ Plugins

📦 Dependencies

⚙️ Configuration

📋 Tasks

serve
Running

build
Compiles and minifies for p...

lint
Idle

test:unit
Running

inspect
Inspect the resolved webpa...

⋮ More

serve Compiles and hot-reloads for development

■ Stop task

⚙️

vue-cli-service serve

🔗

🗨️ Output

📊 Dashboard

🔄 Analyzer

📊 Dashboard

🔗 Open app

⌵

Parsed ▾

?

Status
Success

Errors
0

Warnings
0

Assets
2.1MB (Parsed)

Modules
1.8MB (Parsed)

Dependencies
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

🔍

🔄

▶ Run task



vue-cli-service build --mode dev



Output

Dashboard

Analyzer

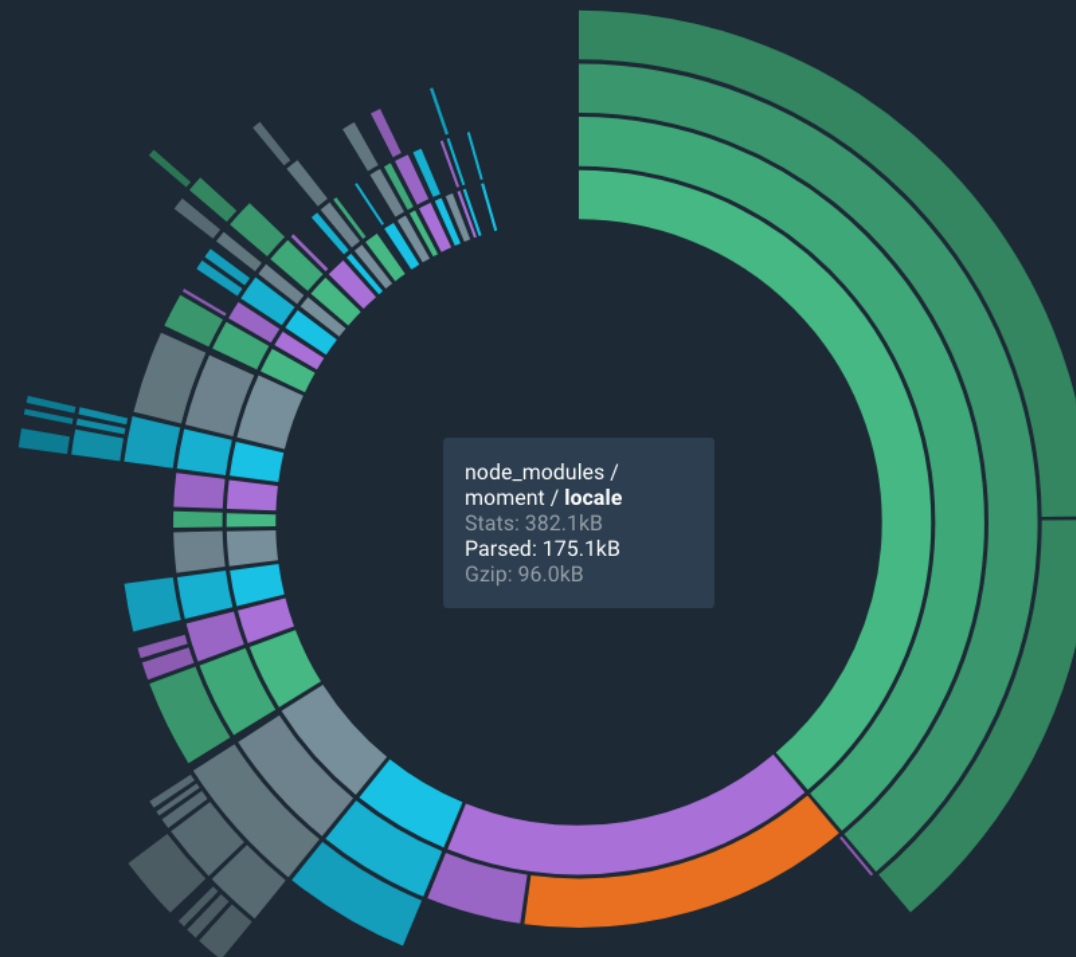
Analyzer

↑ Go up

🏠 Go to home

• Chunk chunk-vendors (1) ▾

Parsed ▾



VueJS 2018 - Productivity

```
# create a component  
echo '<template><h1>My View App</h1></template>' > hello.vue
```

```
# fast prototype  
vue serve hello.vue
```

```
# create library  
vue build --target lib hello.vue
```

VueJS 2019 (v3.0)

- Double runtime performance
- Half the size
- Optional features tree-shakable
- Change detection coverage via JS Proxies (no IE 11 support)

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;
  @Output()
  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({
  // ...
```

😍 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 (v16)

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

React 2019 (v17)

- 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

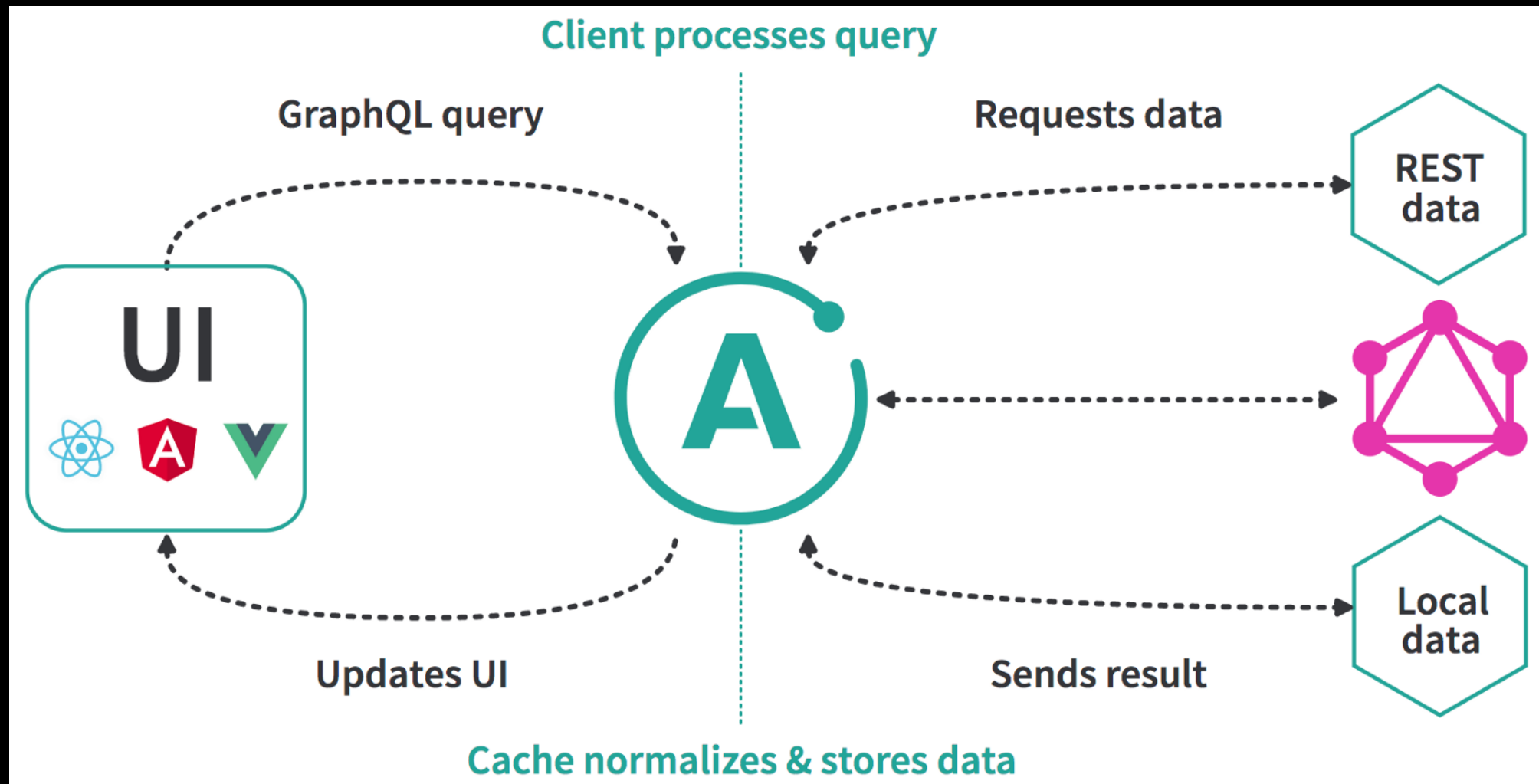
- Facebook
- Github, Amazon
- KLM
- PayPal
- AirBnb
- EggHead & Medium
- Pintrest
- IBM
- Walmart, Shopify & Starbucks
- American Express
- Sitecore, ContentFull, DatoCMS, WordPress, ...
- Microsoft ??? (next to Github)
- ...

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 kkempe merged 5 commits into master from break-down-match-data-requests on Apr 4

Conversation 1 Commits 5 Files changed 46

peggyrayzis commented on Apr 4

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

kkempe and others added some commits on Mar 31

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

+1,187 -4,950

Assignees: No one—assign yourself

Labels: None yet

Projects: None yet

Milestone

Apollo Client Sample

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

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

```
@graphql(GET_DOGS_QUERY)
```

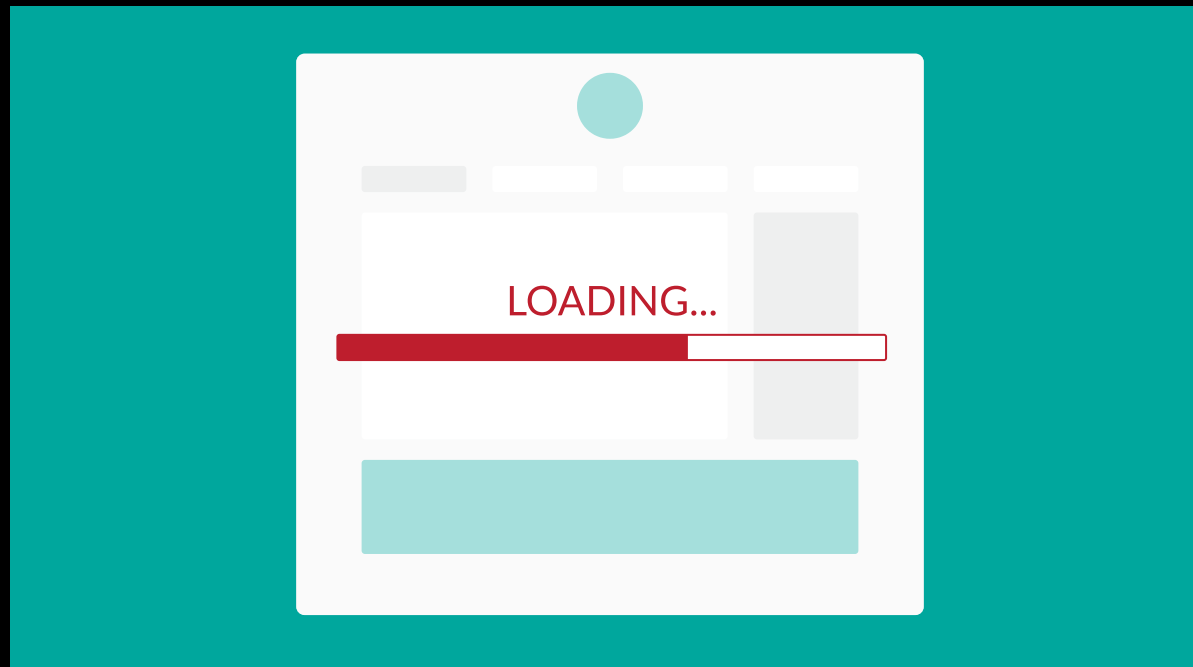
```
const Dogs = ({ onDogSelected, data: { loading, dogs, error } }) => {
  if (loading) return 'Loading...';
  if (error) return `Error! ${error.message}`;
  return (
    <ul>{dogs.map(dog => <li key={dog.id}>{dog.breed}</li>)}</ul>
  );
};
```

What is your web app target?

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

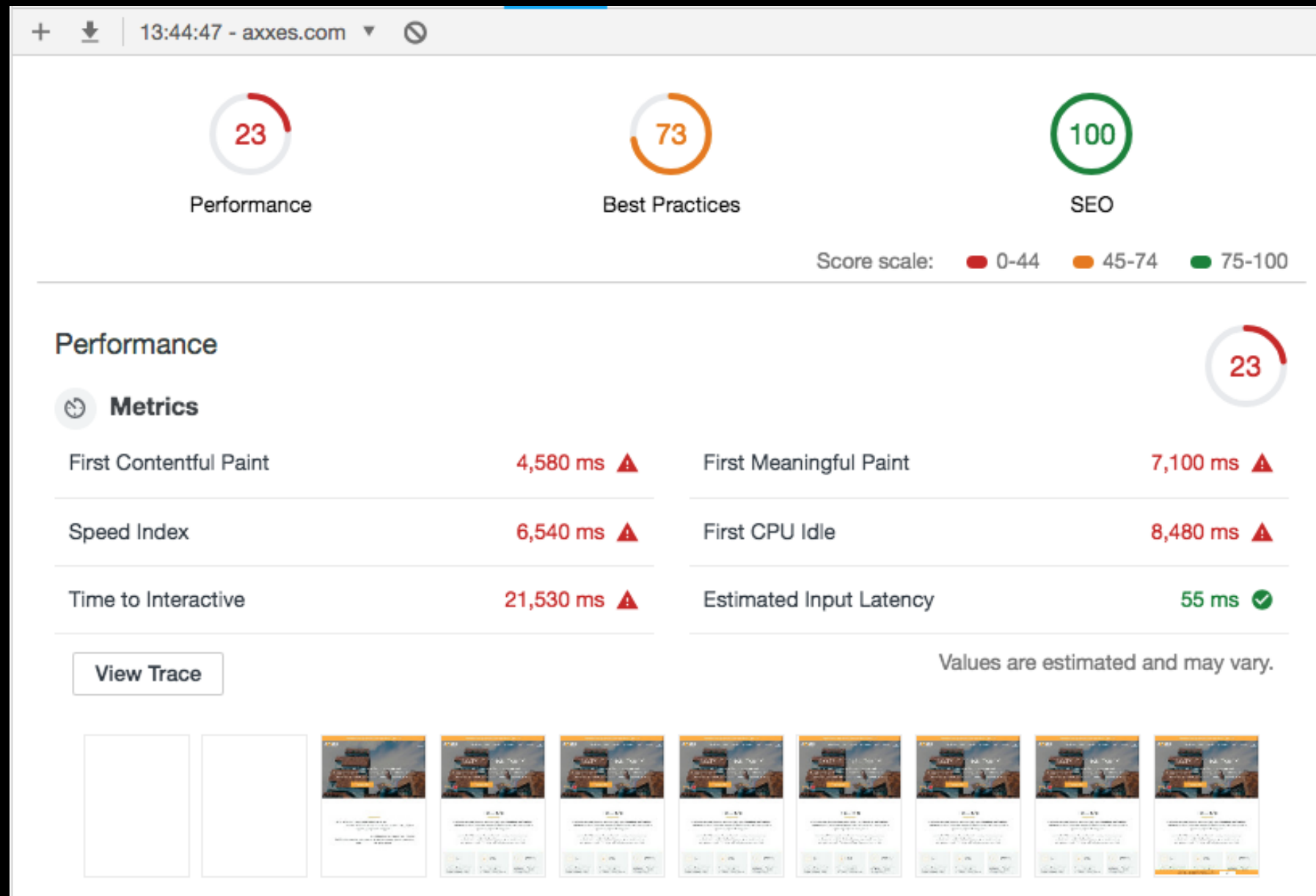
Performance matters

For any web application

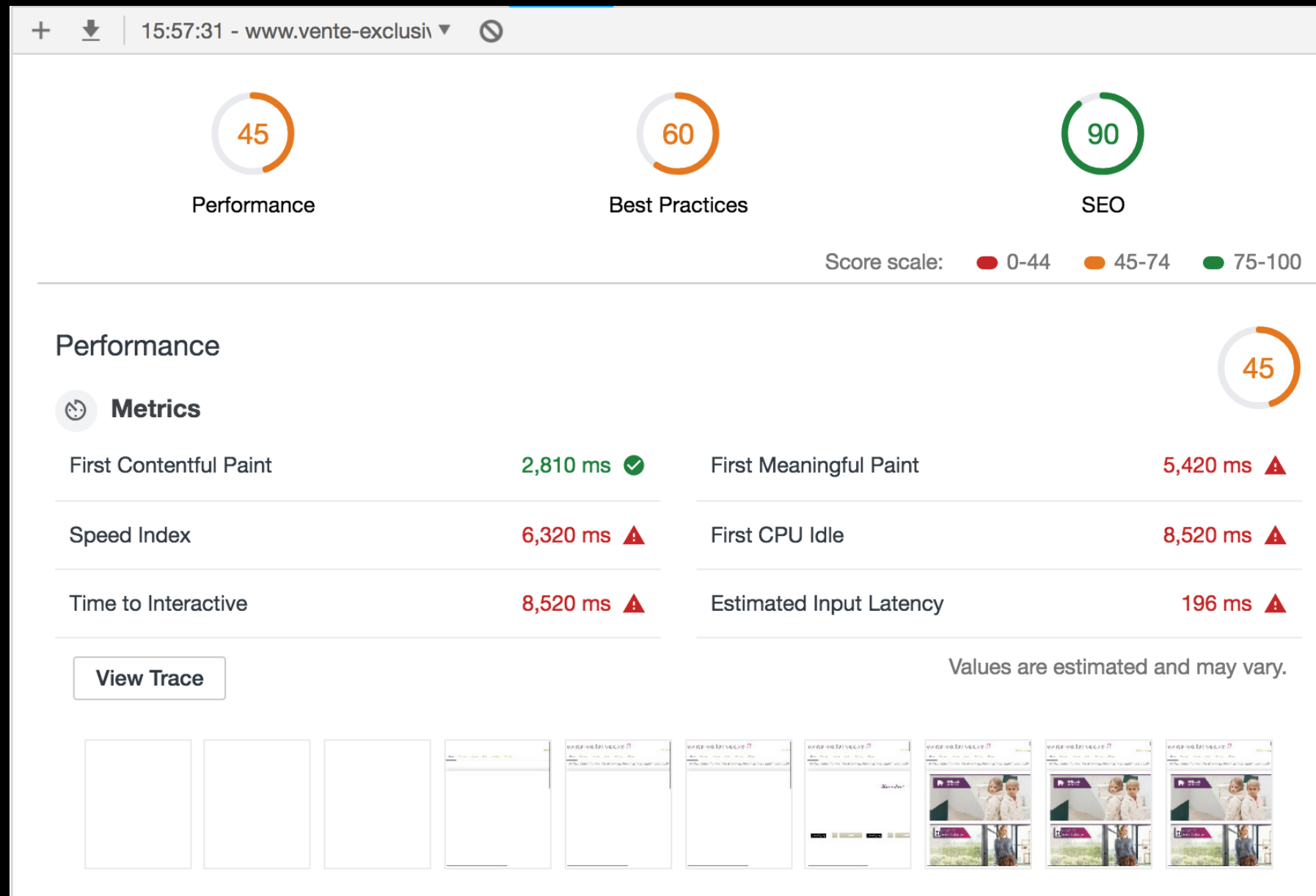


Lets test ...

ordina.be

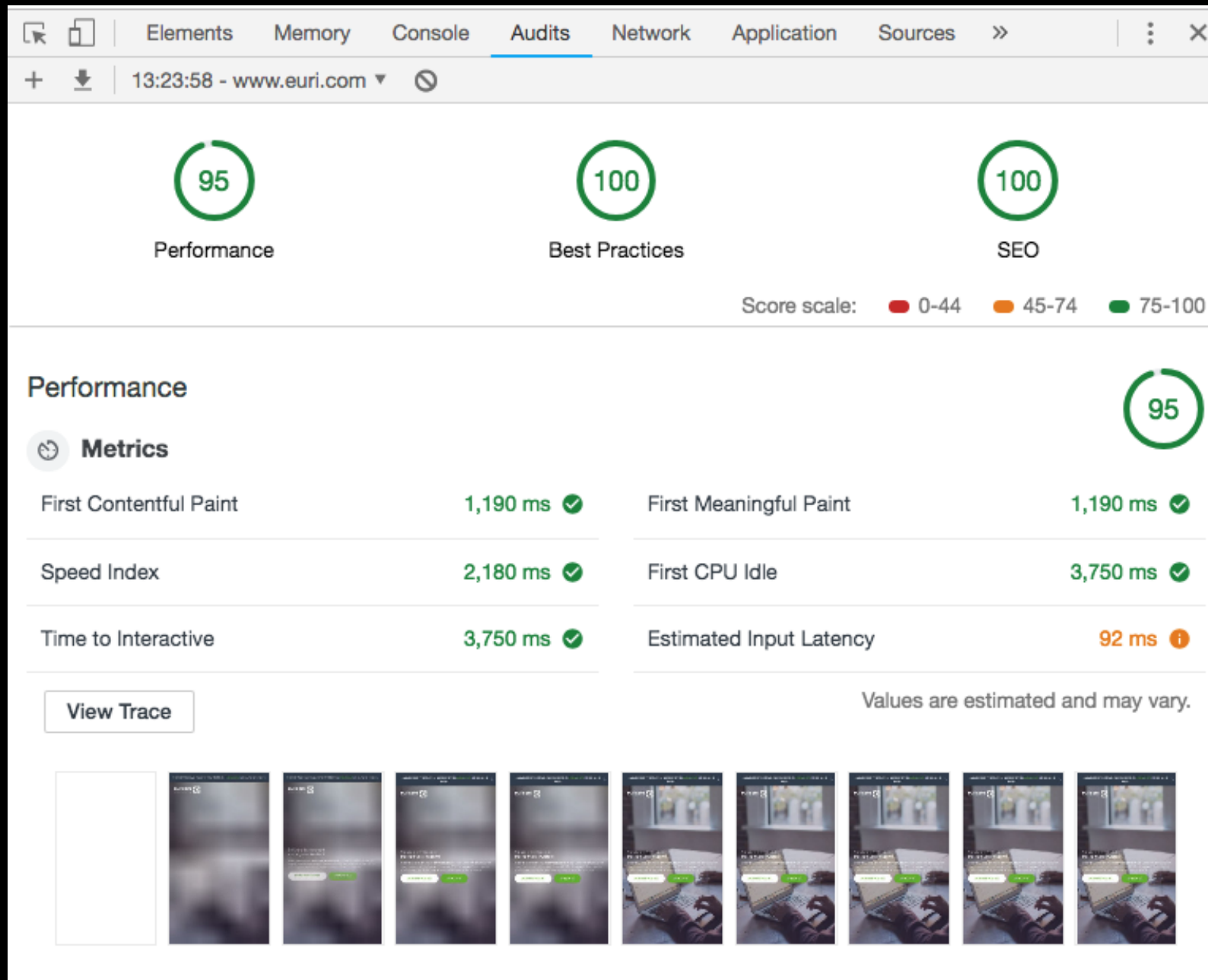


www.vente-exclusive.com



huri.com ?

huri.com 🙌

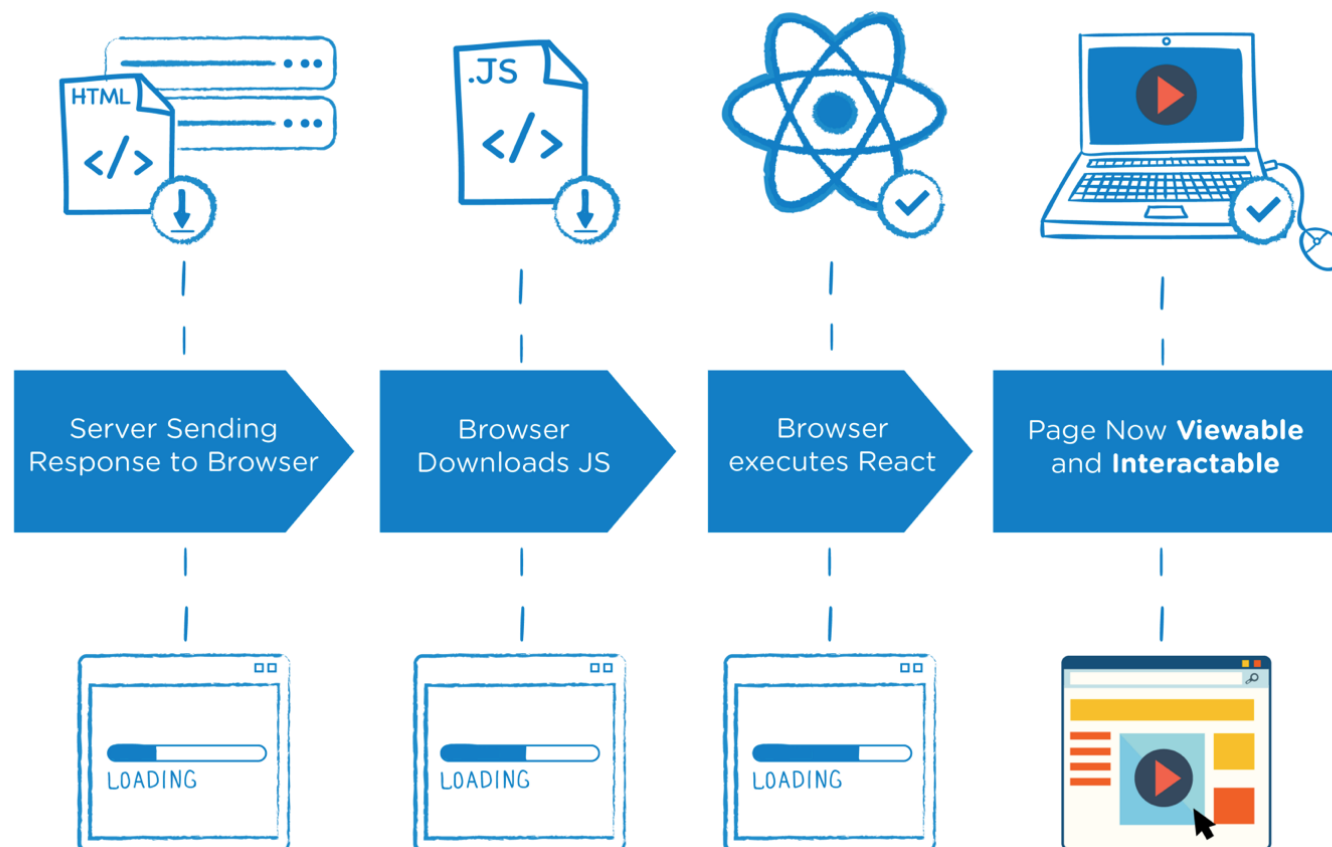


Client Side, Server Side and Pre-rendering

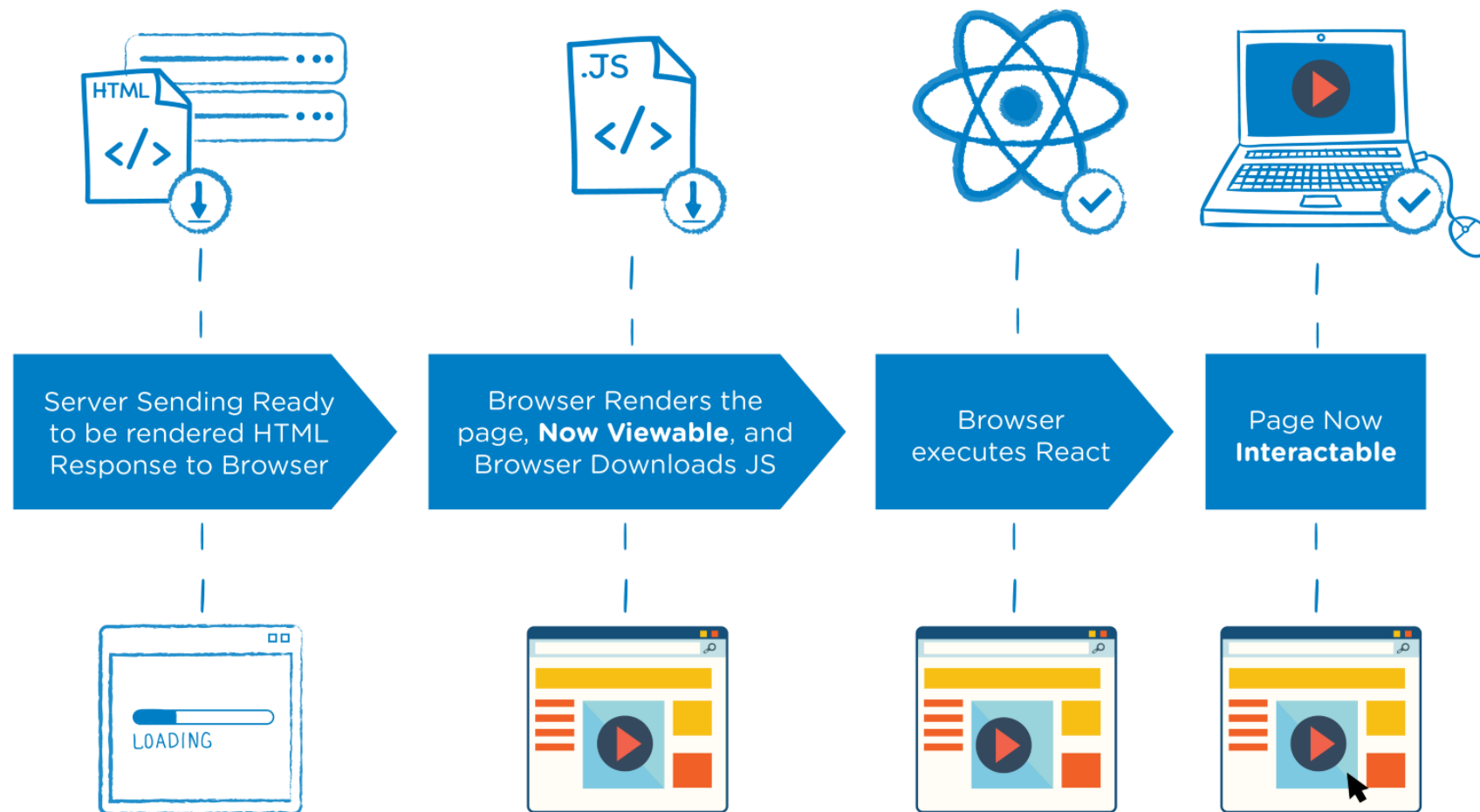
Render what, where, ...?

CLIENT SIDE RENDERING

CSR



SSR



Improve SEO and noticeable performance.

Server Side Render (SSR)

- Angular Universal 🥹
- Next.js 7 - React SSR Done Right
- Nuxt 2 - Universal Vue.js Apps

Pre-rendering

App is rendered 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

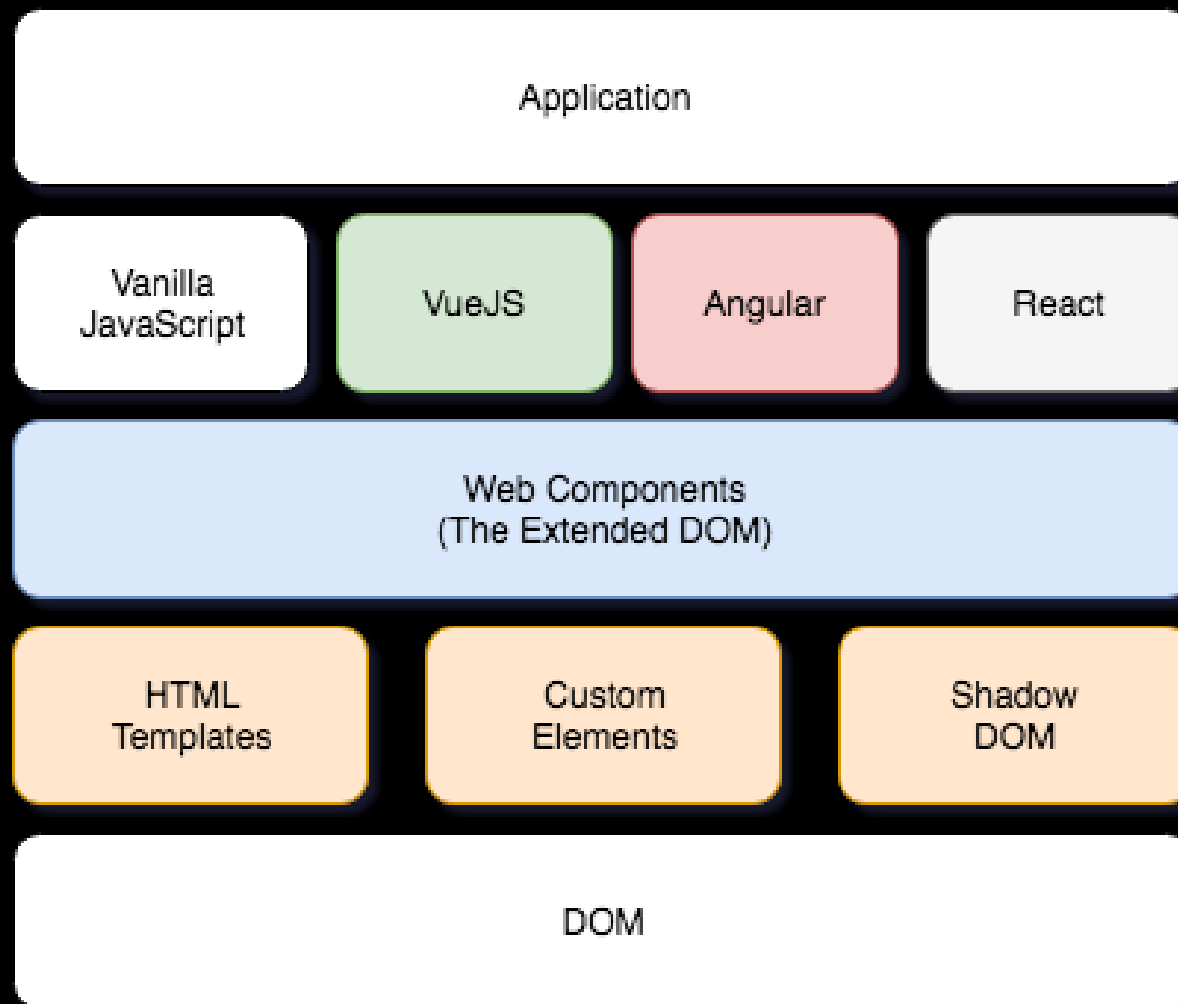
Static Web Host

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

Web-Components

The building blocks of the future

Web-Components



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',  
  styleUrls: '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

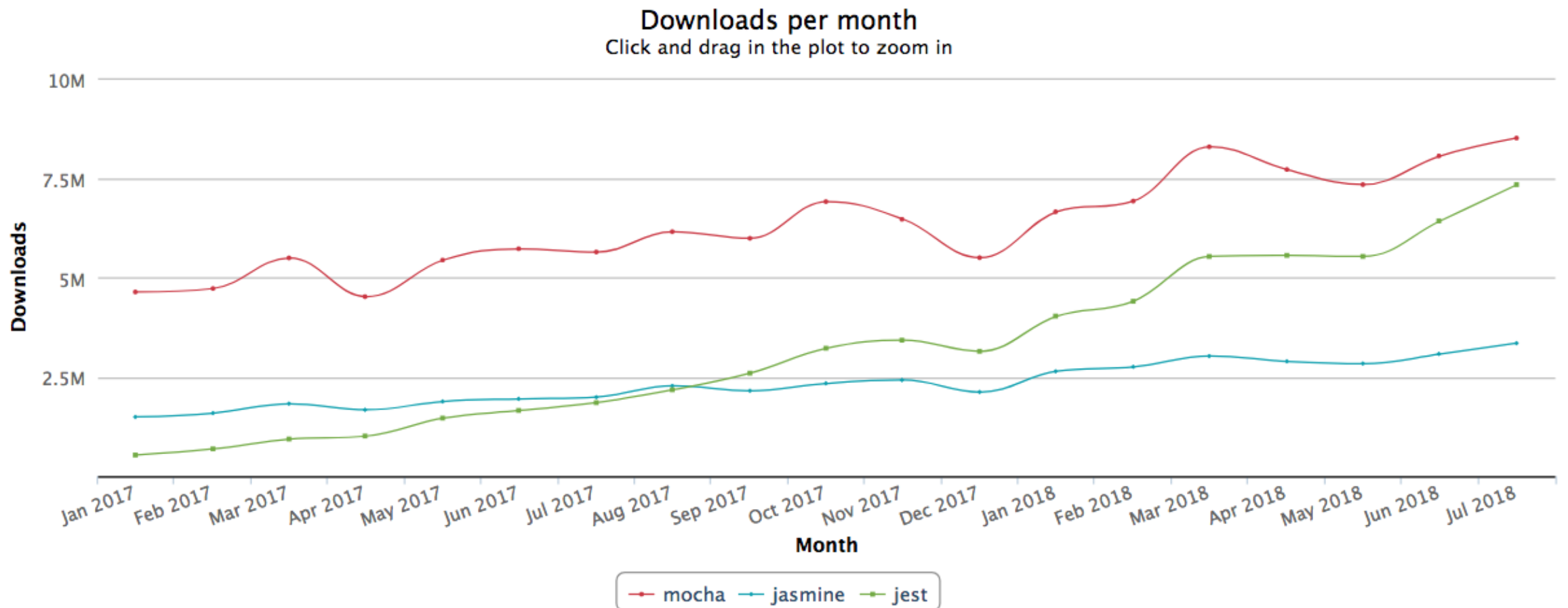
Use

```
<html>
  <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

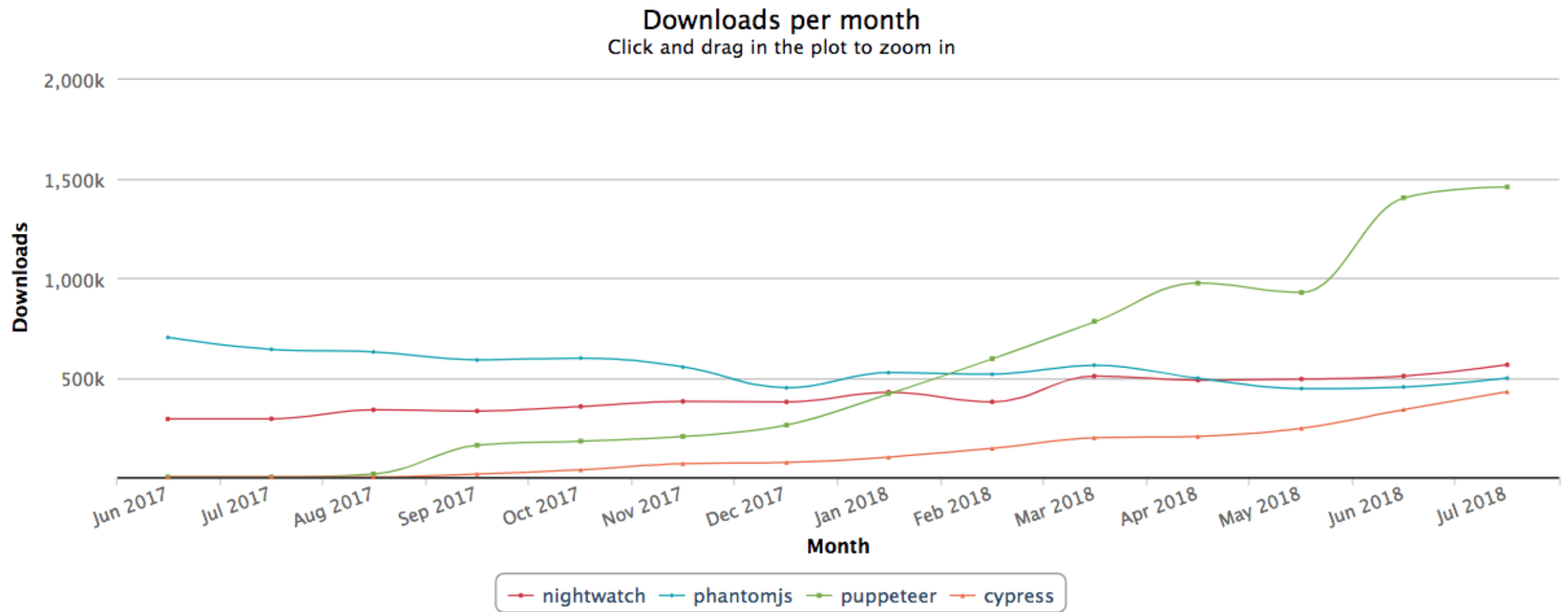
- Component Driven Development
- Component Library
- Visual Component Development & Testing
- Documentation
- Available for: Angular, React, VueJS, Polymer, HTML/CSS

[Live Sample - React](#)[Live Sample - Angular](#)

What e2e test framework are you using?

- None
- Protractor (angular)
- Nightwatch
- Puppeteer
- TestCafe
- Cypress 😊

Rizing stars: Puppeteer & Cypress





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

Cypress

Cypress at Twitter

↻ Cypress.io Retweeted



Kurt Kemple @kurtiskemple · Sep 24

I have never had actual fun writing e2e tests before, but I'll be damned if @Cypress_io hasn't finally done it for me. Writing these tests is so easy I'm worried I'll over-test just to write 'em. 😂



1



4



33



↻ Cypress.io Retweeted



Matt Pocock @mpocock1 · Sep 22

.@Cypress_io is the real fucking deal. At my last job I was the solo frontend guy on a mid-size application and wrote a ton of end-to-end tests with it. Incredible bit of kit.



3



9



↻ Cypress.io Retweeted



Blair Brooks @developerblair · Sep 21

Just found @Cypress_io for integration and end to end testing and I'm very impressed. Their docs are so much better than anything else I've found out there.



2



11

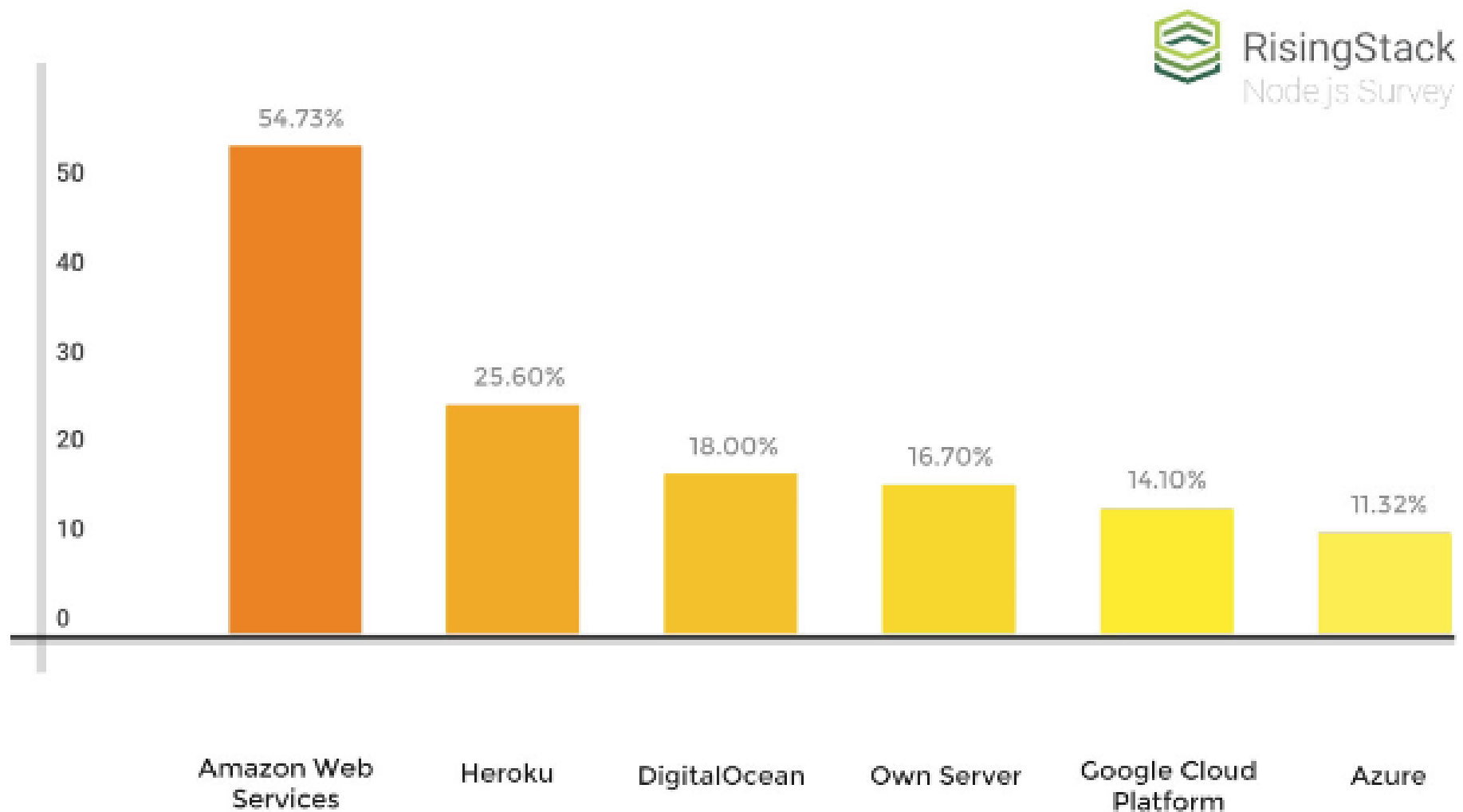


Where do you deploy your NodeJS app

- Virtual Machine (AWS, Azure, Google)
- Docker
- App Engine (Heroku, Azure, AWS, Google)
- Other
- Zeit Now 😊 (Quick and easy)
- Serverless 😊

NodeJS Survey

Where do you deploy Node.js?



Now – Global Serverless Deployments



```
bash

▲ ~/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 or Surge.sh
- Netlify 😊



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

[Netlify](https://www.netlify.com)

Honorable Mentions

- DateFns (a modern date library)
- Babel 7.0
- PWA
- WebPack 4.x & TypeScript 3.0
- Capacitor (replaces Cordova)
- React Native (still strong)
- Flutter (vs React Native)

Thank You 🖐️

<https://mjr-javascript-trends-2018-ftjsgtwyhg.now.sh>

Credits

Built with MDX Deck

Deployed on now.sh