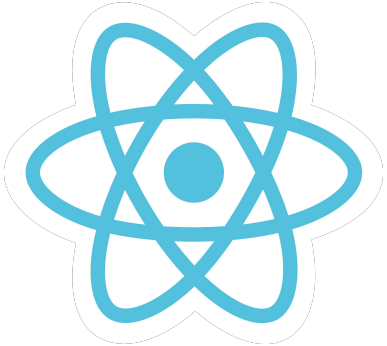


Frontend Frameworks

Introduction

Speakers

Lee Byron
React Core Team @ Facebook

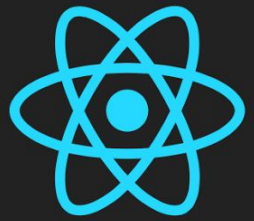
@leeb

Rob Wormald
Angular Core Team @ Google

@robwormald

Taras Mankovski
Ember Contrib @ EmberSherpa

@tarasm



React

A JAVASCRIPT LIBRARY FOR BUILDING USER INTERFACES

Started at Facebook in 2012

Created by Jordan Walke

Used for Comments, Chat, Notifications, and Instagram

React

Release Date: May 26, 2013

Contributors: 846

License: BSD

Ben Alman
@cowboy

Follow

Facebook: Rethink established best practices™

RETWEETS	LIKES
27	48

2:40 PM - 29 May 2013

27 48

Reply to @cowboy

Jaseem Abid @jaseemabid · 21 Jan 2015

React is different in many ways

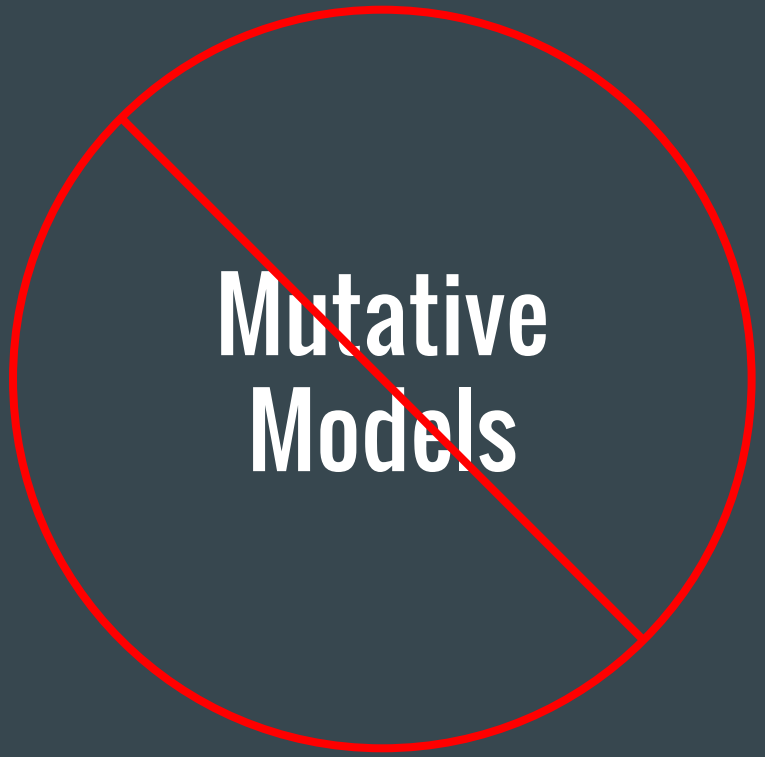
Three primary differences

1. React is not MVC

MVC MVVM MVW

Models

Mutative Models

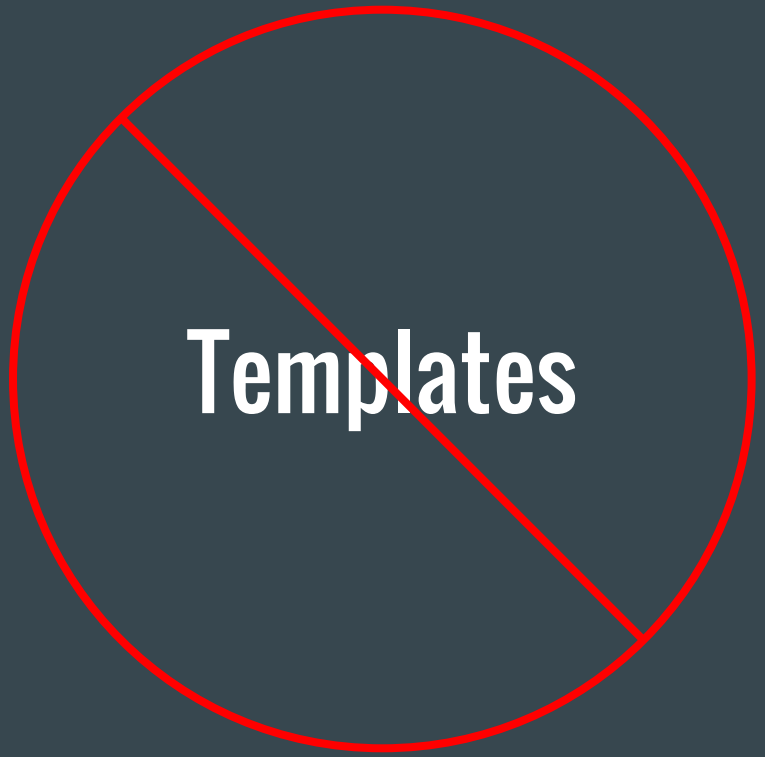

Mutative
Models

React is
The “V” in MVC?

React is
Component-based UI

```
function ExampleComponent(props) {
  return React.createElement(
    "div", { className: "fancy" },
    "Hello" + props.name
  )
}
```

Just JavaScript


Templates

```
function ExampleComponent(props) {
  return React.createElement(
    "div", { className: "fancy" },
    "Hello" + props.name
  )
}
```

```
function ExampleComponent(props) {
  return React.createElement(
    "div", { className: "fancy" },
    "Hello" + props.name
  )
}
```

Kinda miss those Templates...

JSX

A Syntax Extension to JavaScript

```
function ExampleComponent(props) {
  return React.createElement(
    "div", { className: "fancy" },
    "Hello" + props.name
  )
}
```

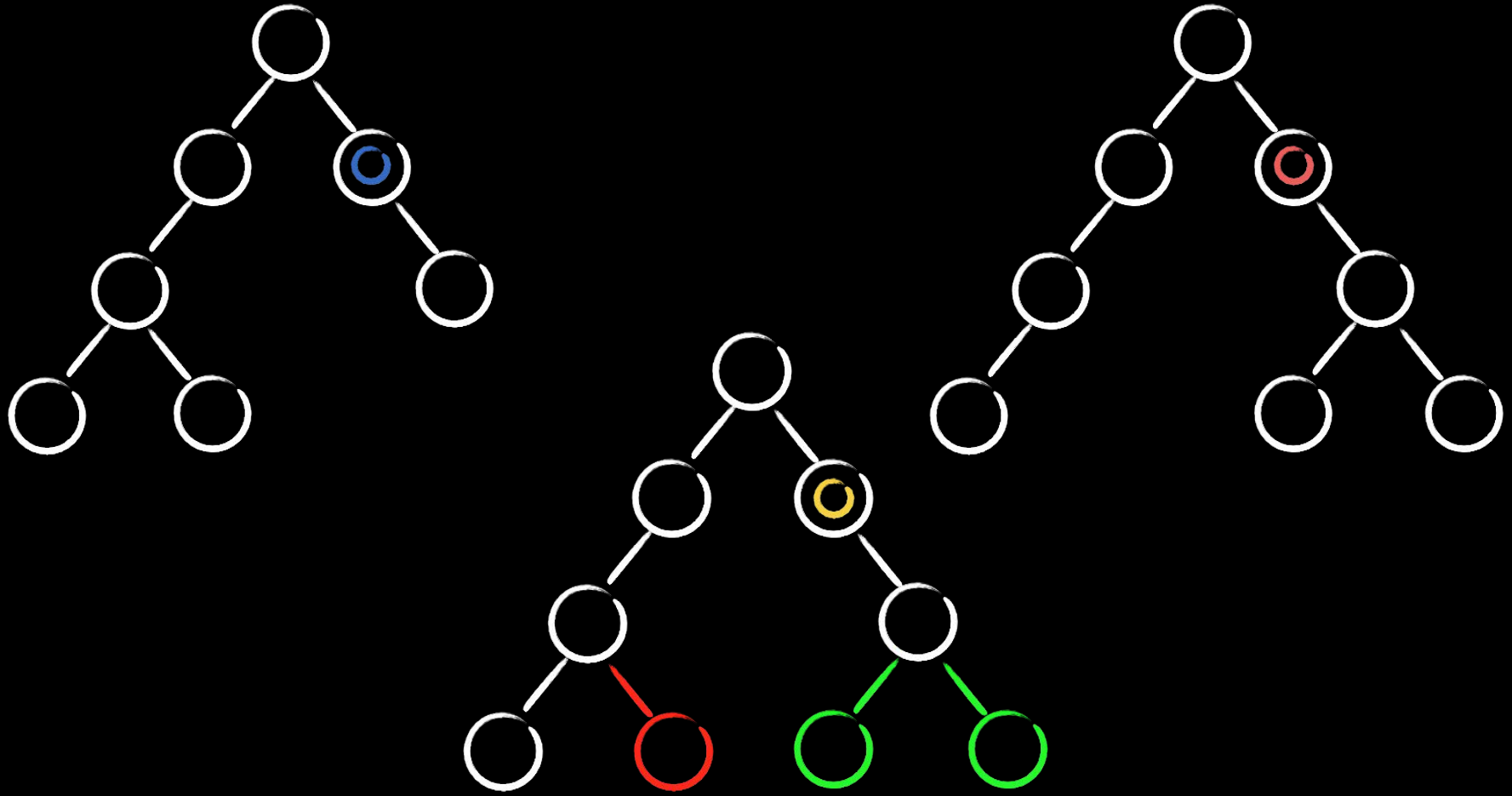
```
function ExampleComponent(props) {
  return (
    <div className="fancy">
      Hello {props.name}
    </div>
  )
}
```

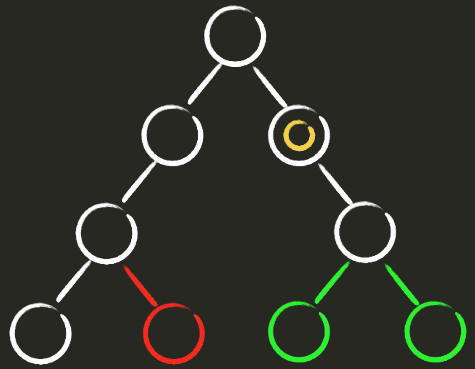
Something changed?
Re-render everything!

Won’t that be slow?

Reconcile

Diff previous view and next view





```
// Imperative view operations
setProp(prev2, "class", "newValue")
removeElement(prev6)
addChild(prev4, next1)
addChild(prev4, next2)
```

Quite fast

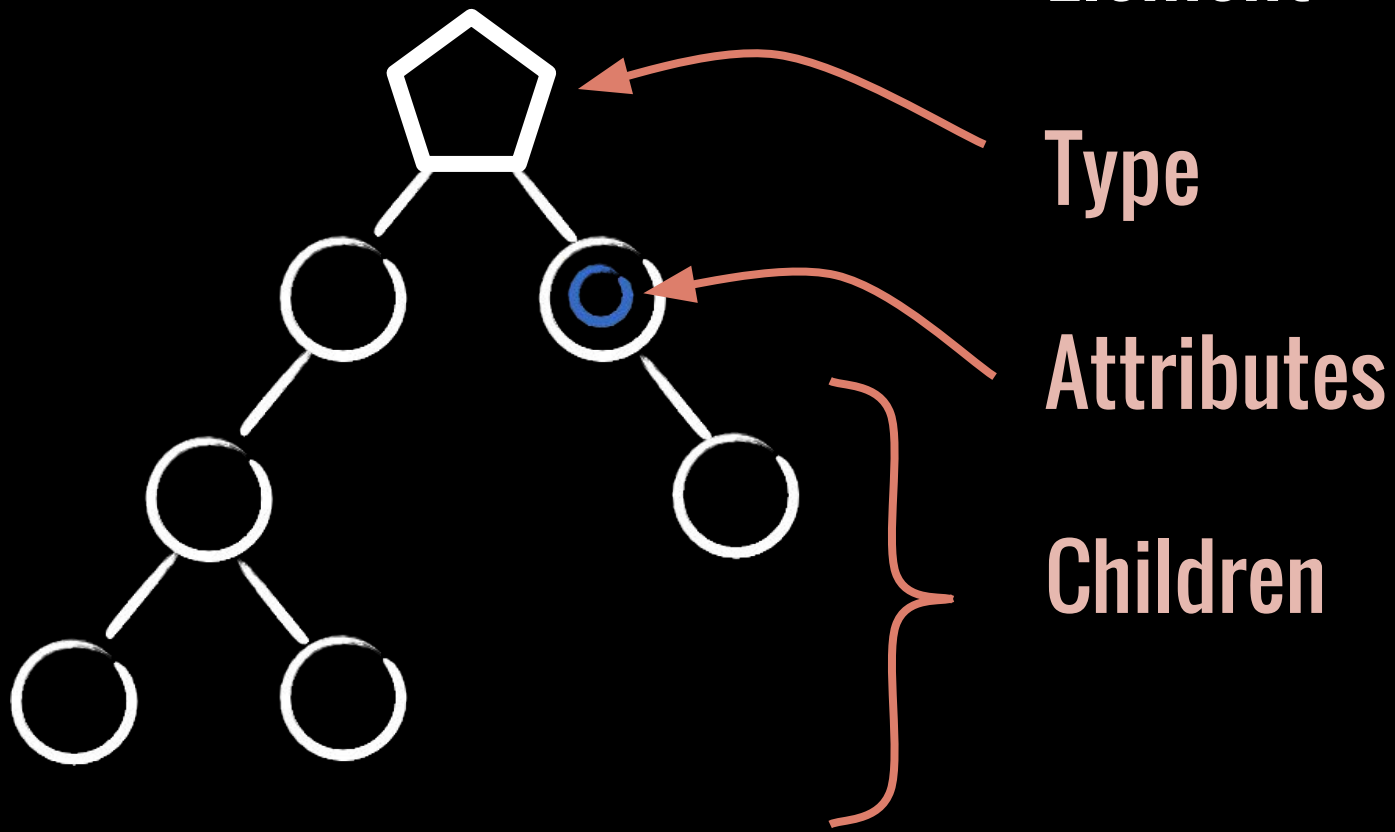
2. React is for UIs, not Web Apps

HTML

HTML

Component:

```
(Data) => Element
```


Element:
Type
Attributes
Children

UI	Elements
Browser	DOM
Server	HTML

UI	Elements
Browser	DOM
Server	HTML
iOS Apps	UIView
Android Apps	Android.View

UI	Elements
Browser	DOM
Server	HTML
iOS Apps	UIView
Android Apps	Android.View
Virtual Reality	3D Scene Graph

React	Elements
React DOM	DOM
	HTML
React Native	UIView
	Android.View
React VR	3D Scene Graph

3. React is adopted incrementally

Great for whole applications

Or small portions of the screen

https://www.facebook.com

Search Facebook

Lee Home

Lee Byron
Edit Profile

FAVORITES
News Feed
Ads Manager
Election 2016
Events 3
Messages
Boozers
Nuri 6
Facebook Design 20+
Critique 9
Ocracokers 2
Mobile UIE Open F… 20+
MUIE Team
UIE 20+
React Europe @ FB
GraphQL Core Co…
Shops
Saved
Buy and Sell Groups

PAGES
Mobile Mech LLC
Pages Feed 20+
Like Pages
Create Page
Create Ad

Photo/Video | Photo/Video Album

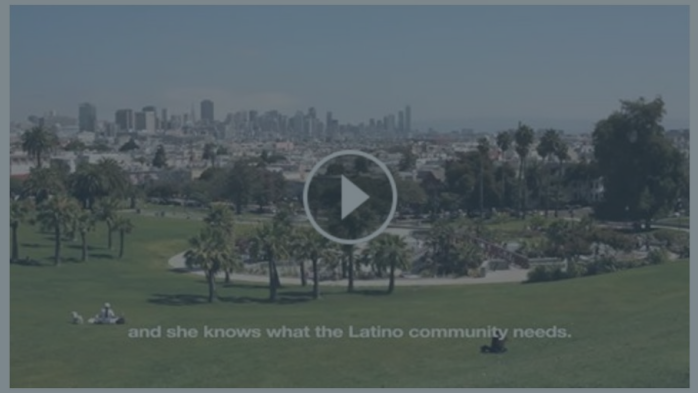
What's on your mind?

Friends | Post

Suggested Post

Affordable Livable San Francisco
Sponsored · Like Page

Help San Francisco fight to be Affordable and Livable for teachers. Vote Hillary Ronen for Supervisor.



Sophie Xie and 68 others 9 Comments 22 Shares 17K Views

Like Comment Share

Eric Zamore's faceversary is today
3 event invites
Jeff Baxendale and 2 others

TODAY'S GAMES

PIT	14	NO	41
BAL	21	SF	23
Final		Final	
IND	31	CAR	13
GB	26	LA	10
Final		Final	

TRENDING
Pixar
350K people talking about this
Mark Cuban
67K people talking about this
Google Search
180K people talking about this
See More

SPONSORED Create Ad

Dean Eckles and Octavio Raygoza are now friends.
Scott Raymond likes Ian Gordon's post.
Connie Yang replied to Mindy Zhang's comment.
Irene Tran likes Heather Hamel's post.
Arturo Bejar reacted to Monica Semergiu's post.
Elliot Lynde likes Joe Savona's post.
Sumeet Sham Vaidya likes Mary Kate's photo.
Greg Hinchliffe likes Brooke Lierman's photo.
Antonio Ono updated his cover photo.
Raquel DiSabatino likes Dan Rather's post.
Jerry Cain reacted to Mark Vogler's post.
Keito Uchiyama replied to Paul Chang's comment.
David Mortenson likes Venkat Venkataramani's photo.

Turn on chat to see who's

Introduce React “Bottom up”

Lifecycle Hooks:

`componentDidMount()`

`componentWillReceiveProps()`

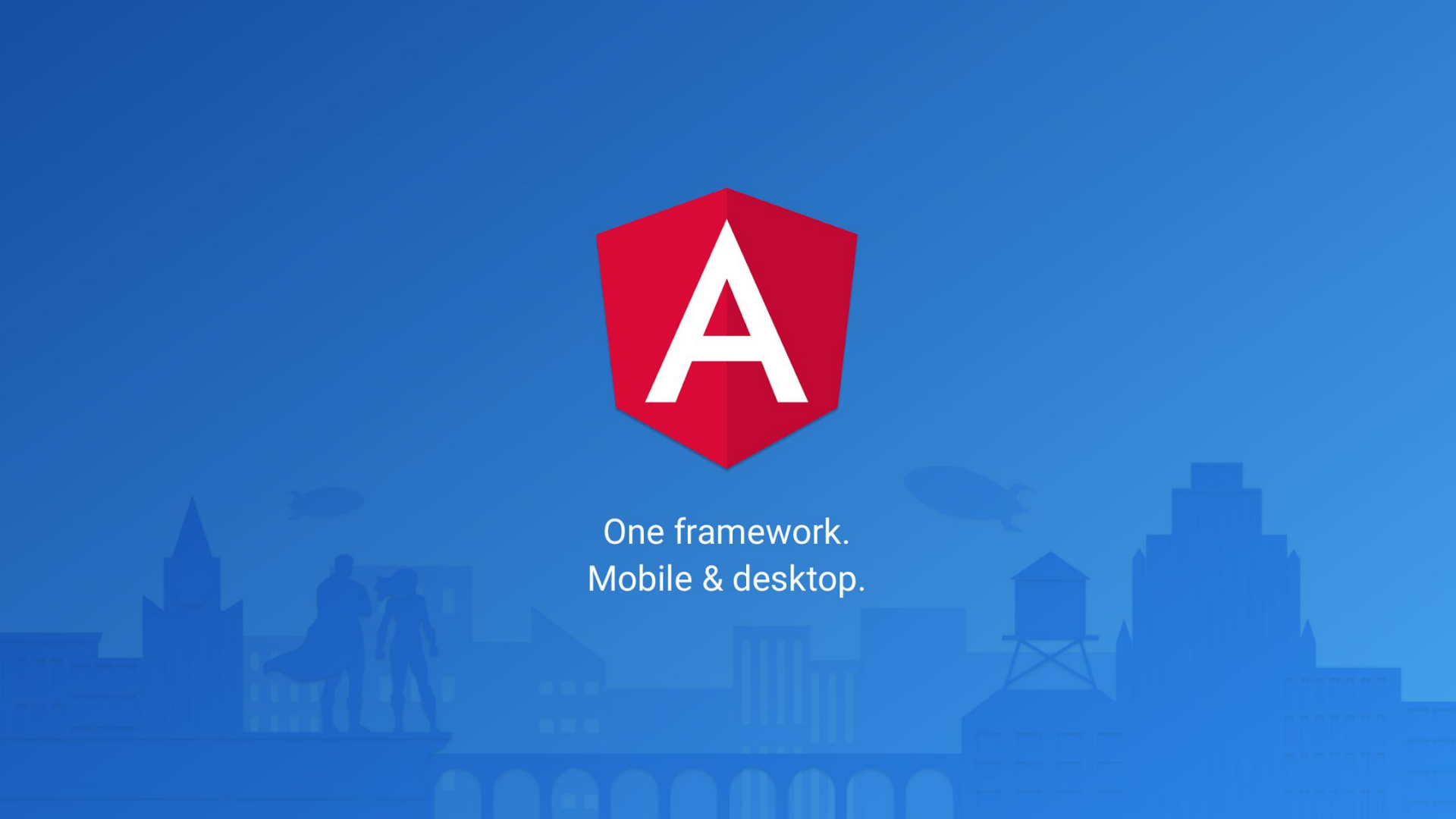
`componentWillUnmount()`

React's "Escape Hatch"

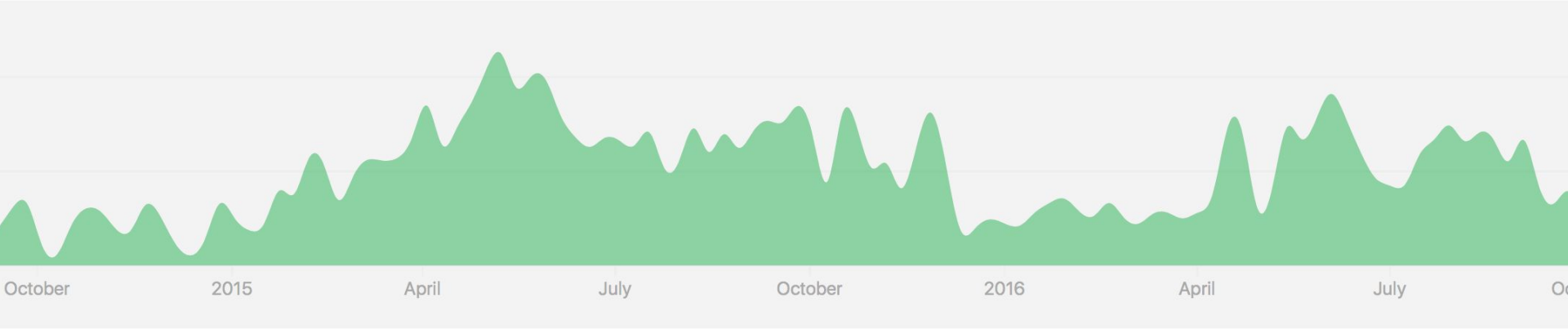
Use your existing frameworks and code with React

1. **React is not MVC**
   Component-based UI with functions instead of templates.
2. **React is for UIs, not Web Apps**
   Elements model web, native apps, and VR.
3. **React is adopted incrementally**
   Small areas bottom up, and escape hatch to use existing code.

1. **React is not MVC**
   Component-based UI with functions instead of templates.
2. **React is for UIs, not Web Apps**
   Elements model web, native apps, and VR.
3. **React is adopted incrementally**
   Small areas bottom up, and escape hatch to use existing code.
4. **React has a great community**
   Who provide education, tools, libraries, and services.


One framework.
Mobile & desktop.

ANGULAR



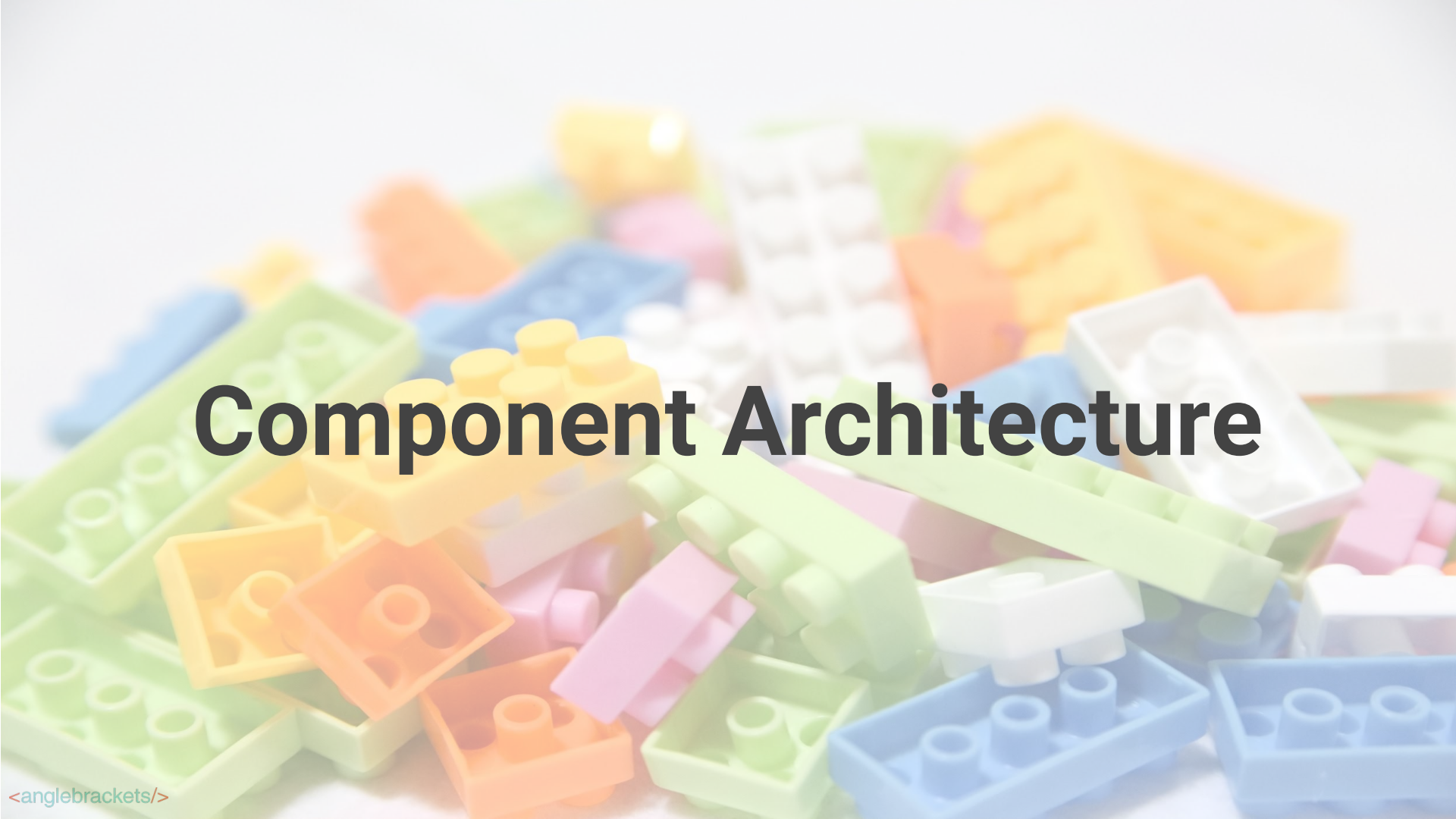
Release Date: September 14, 2014

Contributors: 357

License: MIT

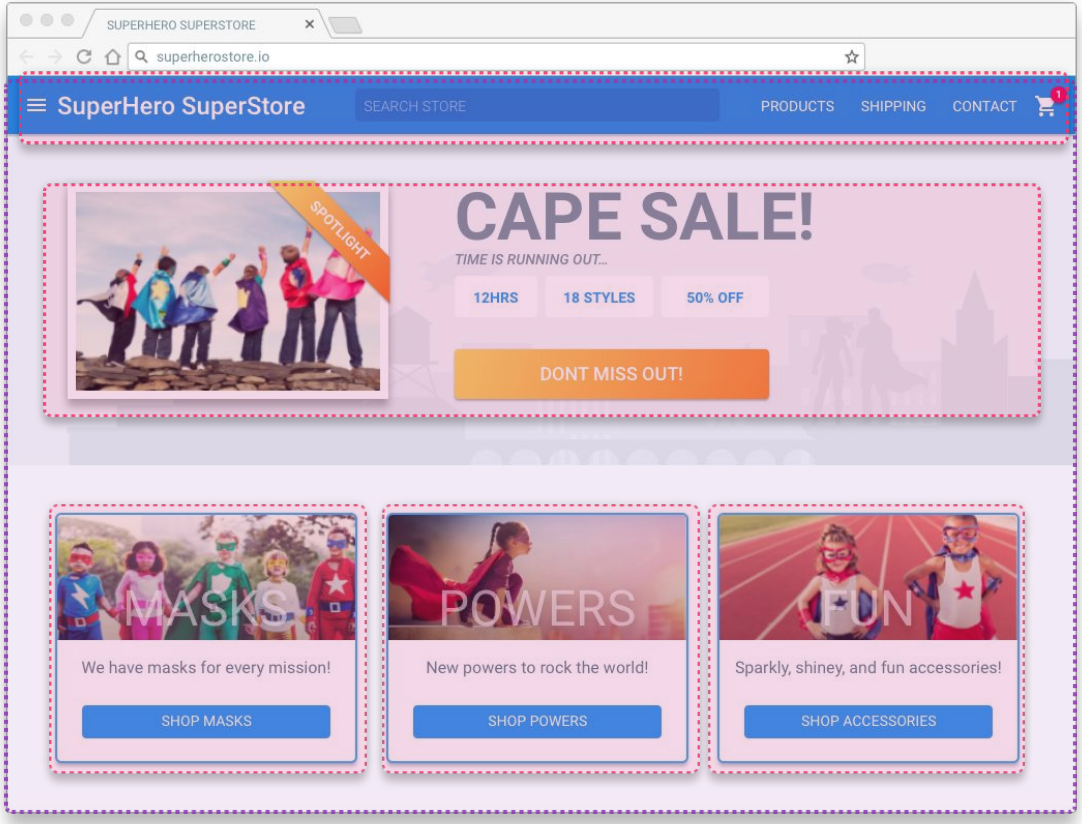
Angular 1

Angular 2



Component Architecture

Application Composition



NAV COMPONENT

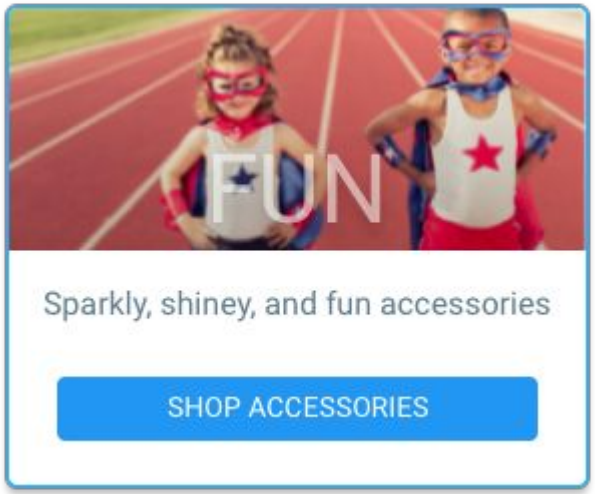
SALE COMPONENT

PRODUCT COMPONENTS

APP COMPONENT

Application Composition

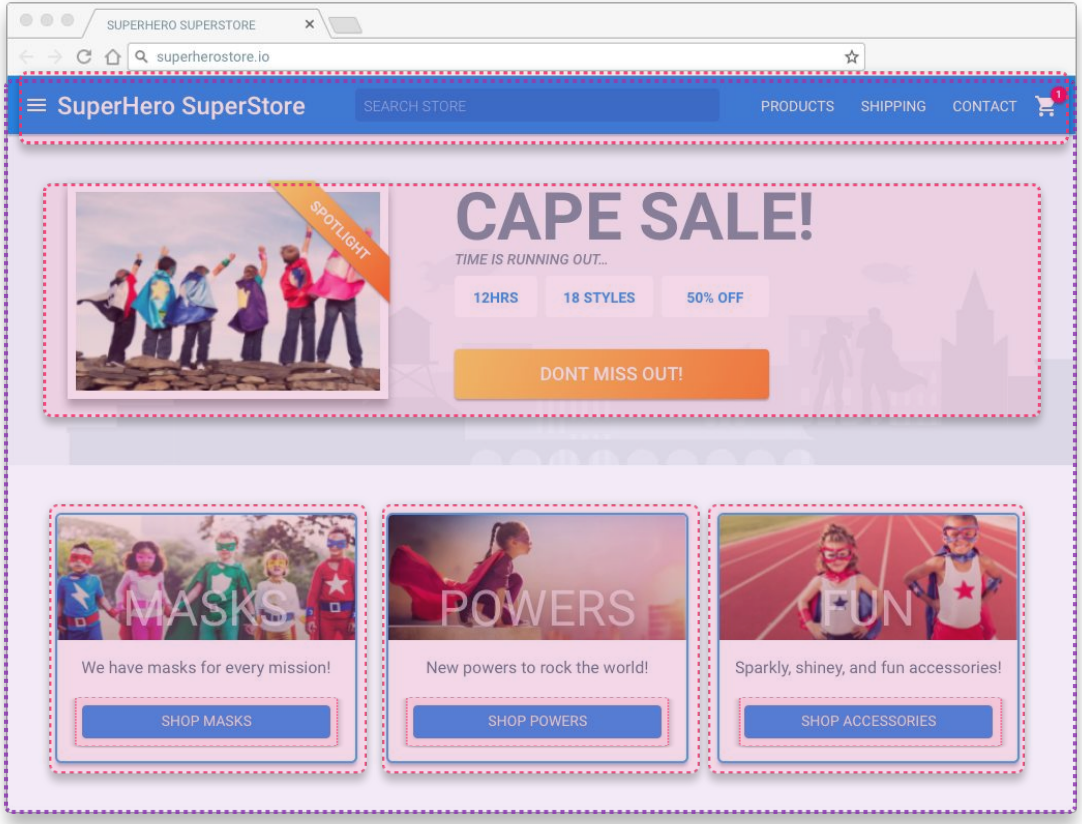
PRODUCT COMPONENT



BUTTON COMPONENT

SHOP ACCESSORIES

Application Composition



NAV COMPONENT

SALE COMPONENT

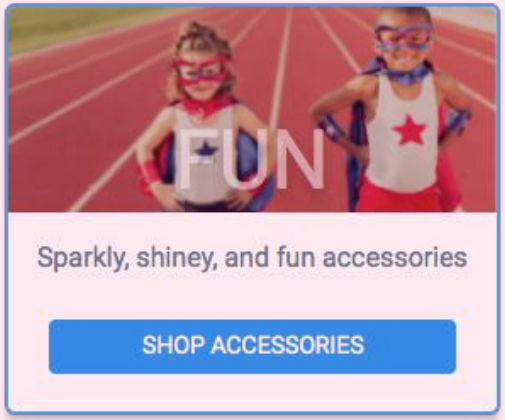
PRODUCT COMPONENTS

BUTTON COMPONENTS

APP COMPONENT

Components

PRODUCT COMPONENT



=

Metadata

HTML **< >** + **CSS** **[ ]** + **Class** **{ }**

HTML

Standard HTML

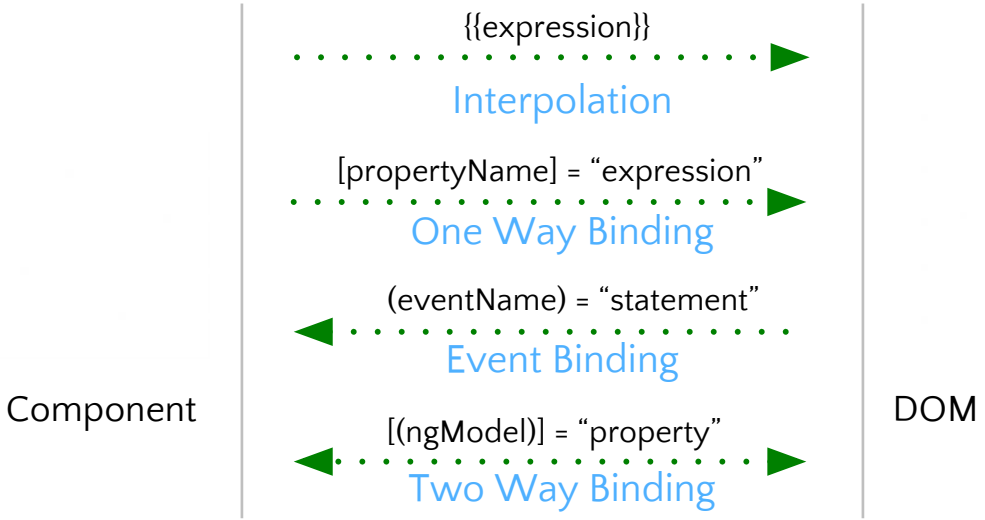
app/hero-list.component.html

```
<h2>Hero List</h2>

<p><em>Pick a hero from the list</em></p>
<ul>
 <li *ngFor="let hero of heroes" (click)="selectHero(hero)">
   {{hero.name}}
 </li>
</ul>

<hero-detail *ngIf="selectedHero" [hero]="selectedHero></hero-detail>
```

Angular 2 Templates: Will It Parse?

Templates

- HTML
- **Property & Event Binding**



app/hero-list.component.html

```
...
 <li *ngFor="let hero of heroes" (click)="selectHero(hero)">
   {{hero.name}}
 </li>
...
<hero-detail *ngIf="selectedHero" [hero]="selectedHero></hero-detail>
```

Templates

<anglebrackets/>

- HTML
- Property & Event Binding
- **Directives**

Structural

app/hero-list.component.html

```
...
 <li *ngFor="let hero of heroes" (click)="selectHero(hero)">
   {{hero.name}}
 </li>
...
<hero-detail *ngIf="selectedHero" [hero]="selectedHero></hero-detail>
```

Attribute

app/hero-detail.component.html

```
<input [(ngModel)]="hero.name">
```

Templates

- HTML
- Property & Event Binding
- Directives
- **Component**

app/hero-list.component.html

```
<h2>Hero List</h2>

<p><em>Pick a hero from the list</em></p>
<ul>
 <li *ngFor="let hero of heroes" (click)="selectHero(hero)">
   {{hero.name}}
 </li>
</ul>
```

Component Class

app/hero-list.component.ts

```
export class HeroListComponent implements OnInit {
 heroes: Hero[];
 selectedHero: Hero;

 constructor (private service: HeroService) { }
 ngOnInit() {
   this.heroes = this.service.getHeroes();
 }
 selectHero(hero: Hero) { this.selectedHero = hero; }
}
```

Service Composition

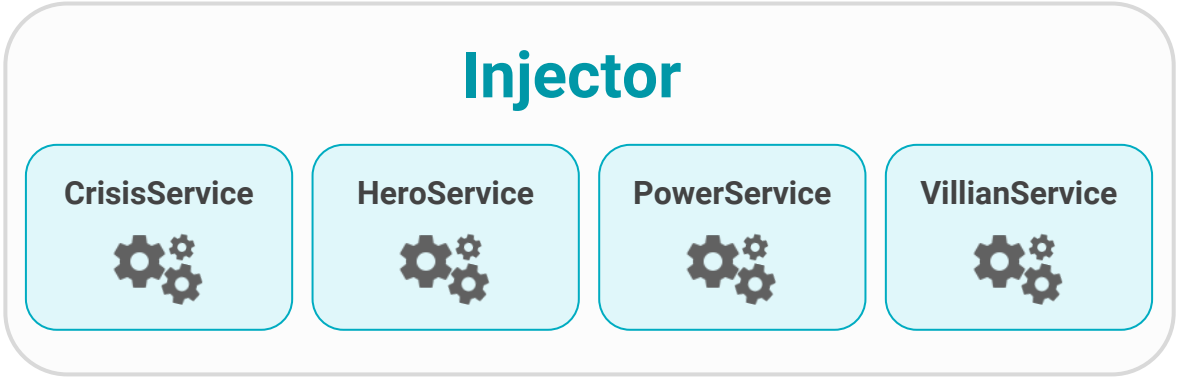
Just A Class

app/hero.service.ts

```
@Injectable()
export class HeroService {
 constructor (
   private http: Http,
   private errorHandler: HttpErrorHandler) { }
 getHeroes () {
   return this.http.get('app/heroes')
        .map(res => res.json().data)
        .catch(this.errorHandler.handle);
 }
}
```

Component Architecture

- Service Composition
- **Dependency Injection**

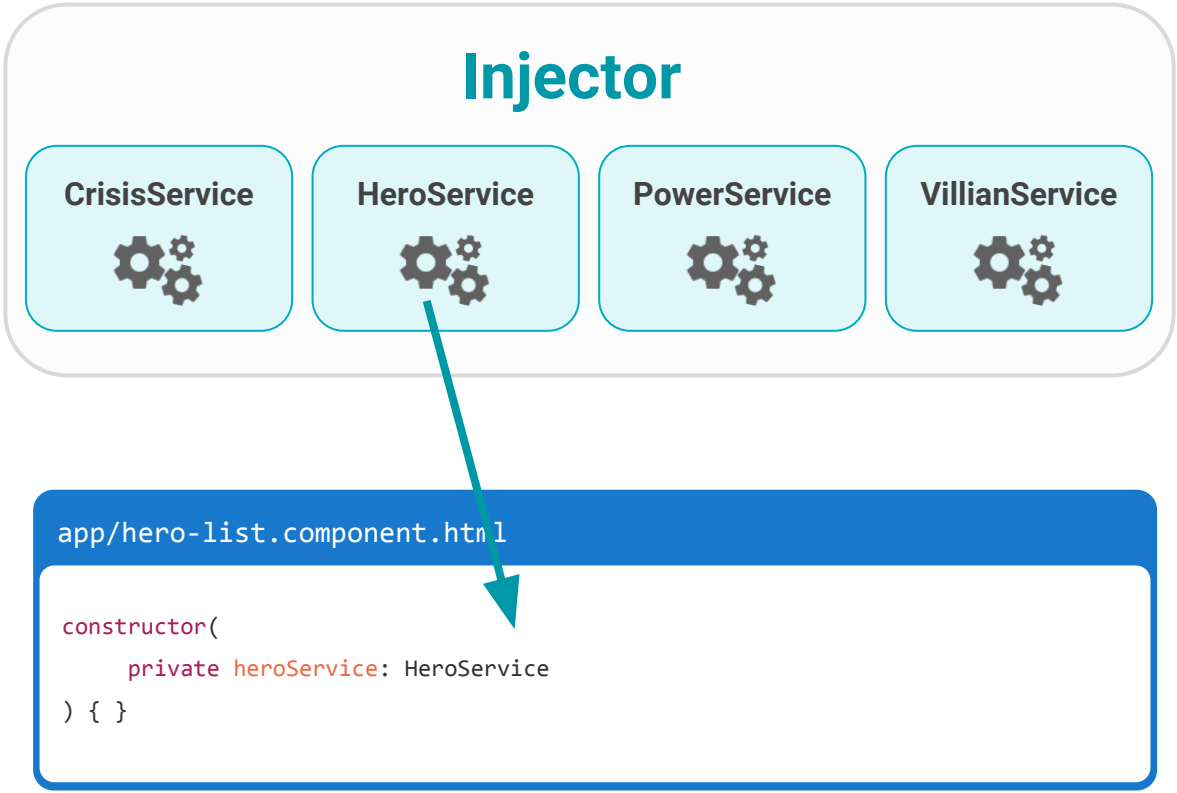


app/hero-list.component.html

```
constructor(
     private heroService: HeroService
) { }
```

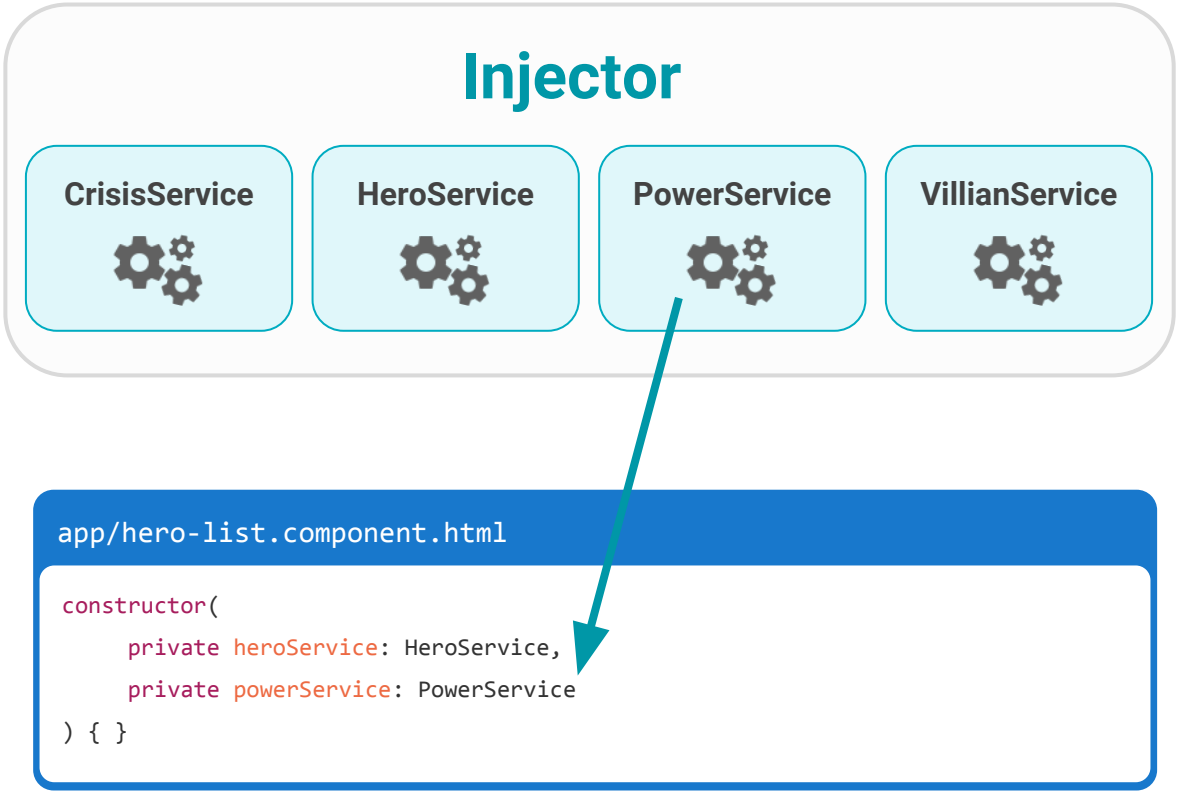
Component Architecture

- Service Composition
- **Dependency Injection**



Component Architecture

- Service Composition
- **Dependency Injection**



Component Architecture

- Service Composition
- Dependency Injection
- **Routing**

app/app.module.ts

```
@NgModule({
 imports: [
   RouterModule.forRoot([
     { path: 'crisis-center', component: CrisisListComponent },
     { path: 'heroes', component: HeroListComponent }
   ])
   ...
 ],
 ...
};
```

Component Architecture

- Service Composition
- Dependency Injection
- **Routing**

app/app.component.ts

```
import { Component } from '@angular/core';

@Component({
 selector: 'my-app',
 template: `
  <h1>Angular Router</h1>
  <nav>
   <a routerLink="/crisis-center">Crisis Center</a>
   <a routerLink="/heroes">Heroes</a>
  </nav>
  <router-outlet></router-outlet>
  `
})
export class AppComponent {}
```

Component Architecture

- Service Composition
- Dependency Injection
- **Routing**

Angular Everywhere

Scenarios

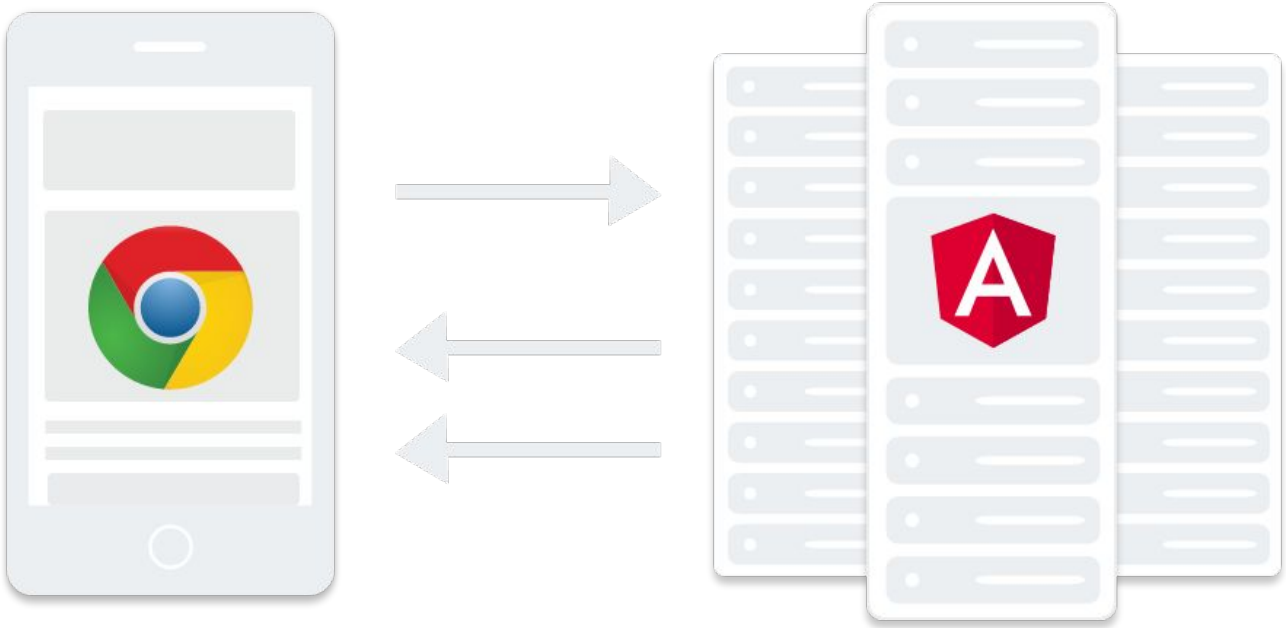
PWA

Progressive Web Apps

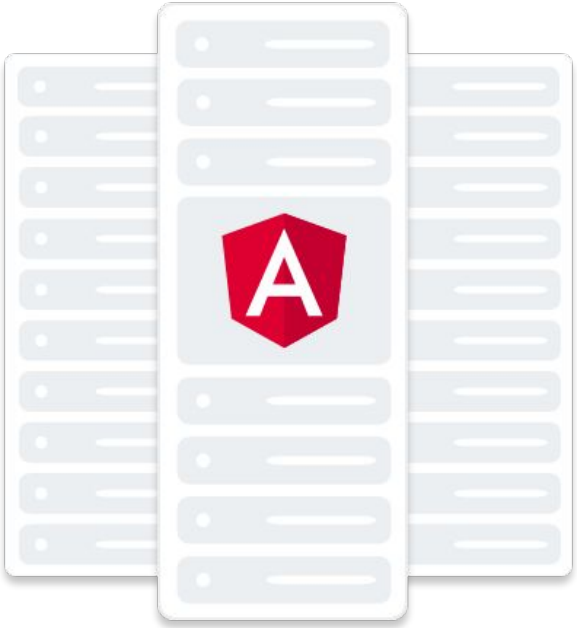
Mobile

Desktop

Server Scenarios



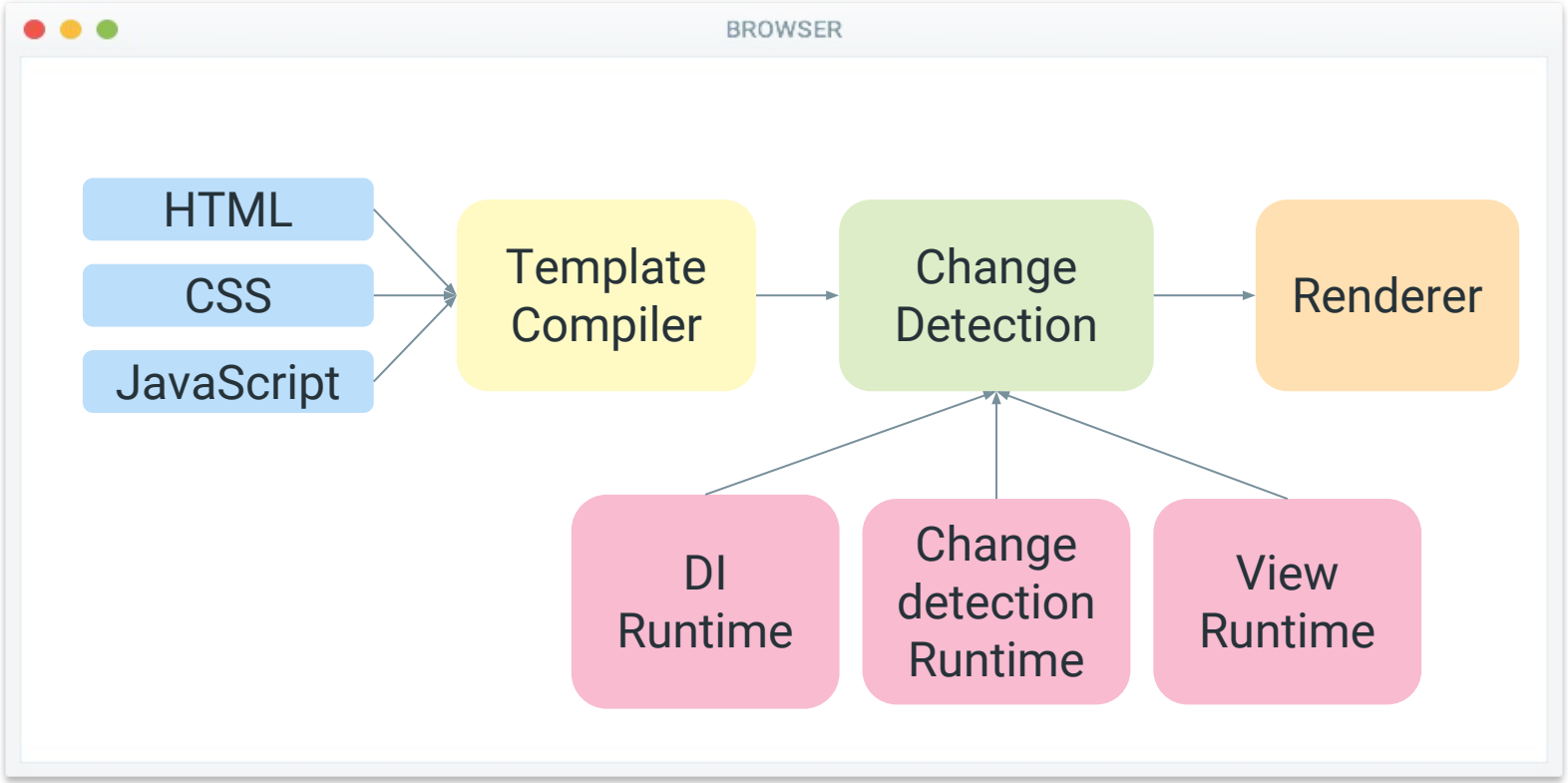
Server Scenarios



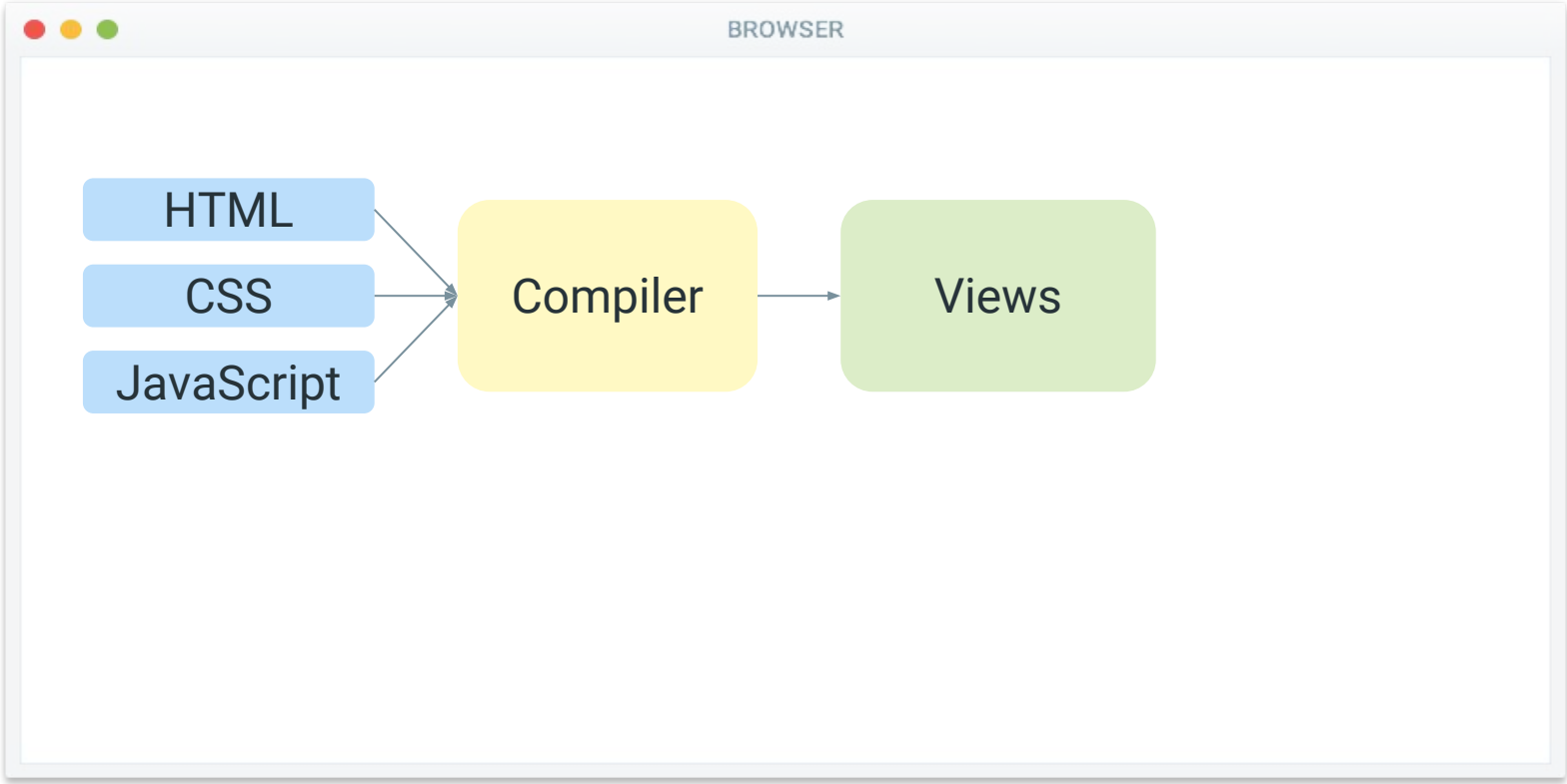
TypeScript

- Superset of Javascript
- Because Types!!
- Readability
- Familiarity
- Tooling

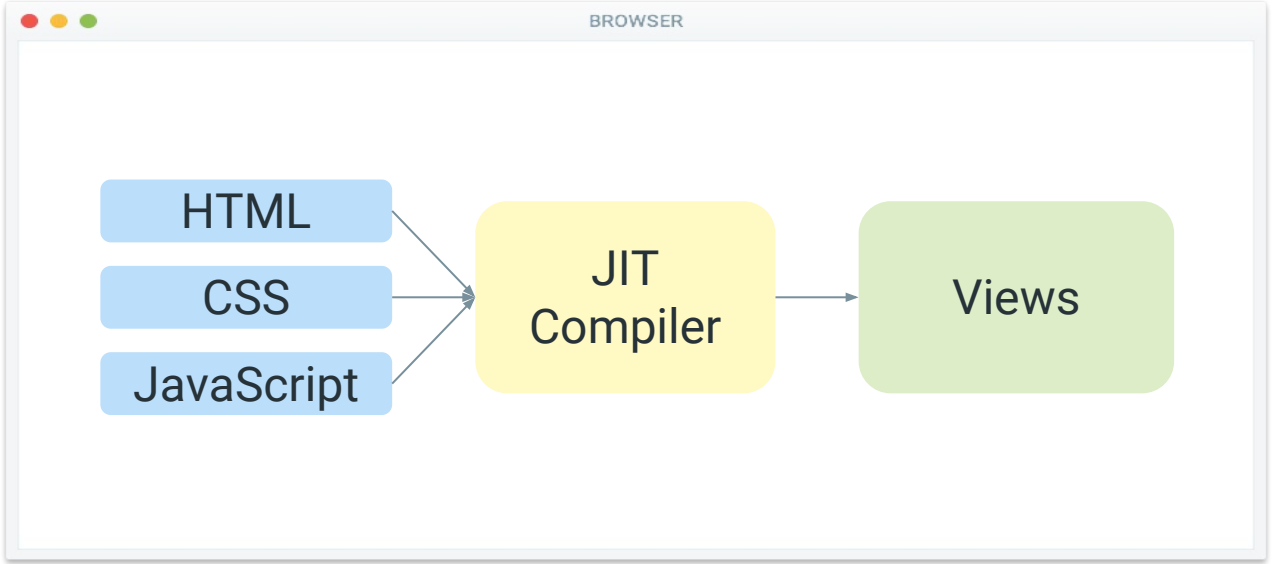
Angular 1: Interpreted Templates



Angular 2: Compiler Generates Code



Just In Time (JIT) Compilation

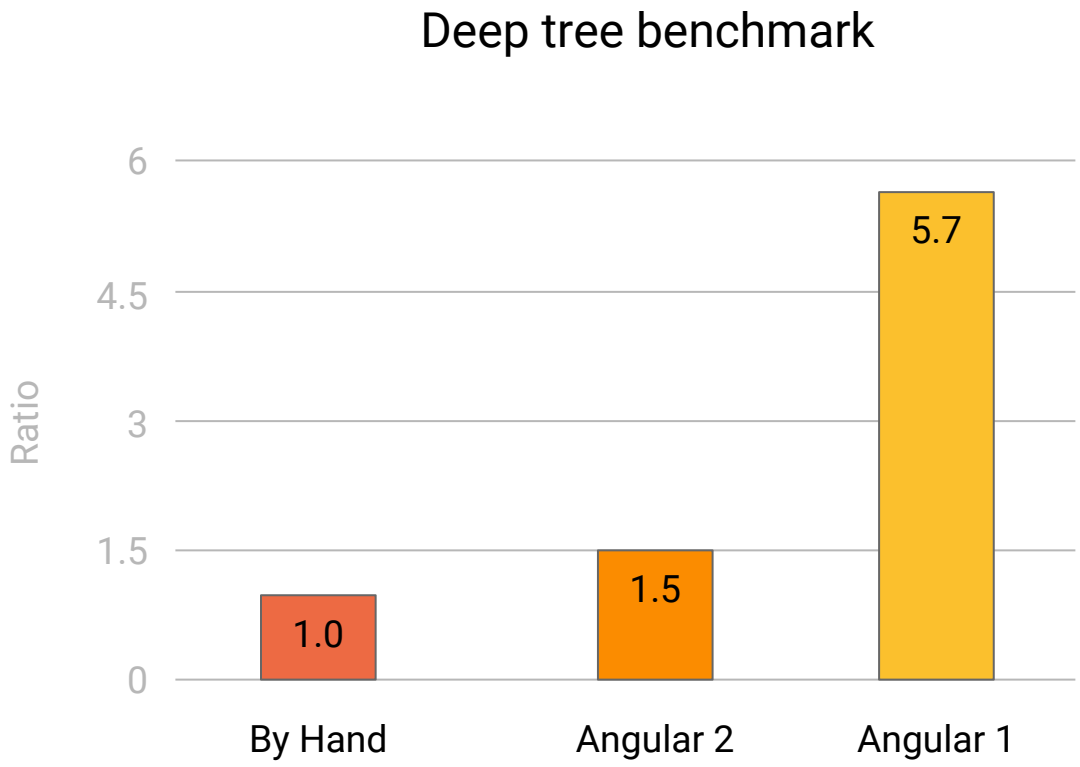


Just In Time (JIT) Compilation

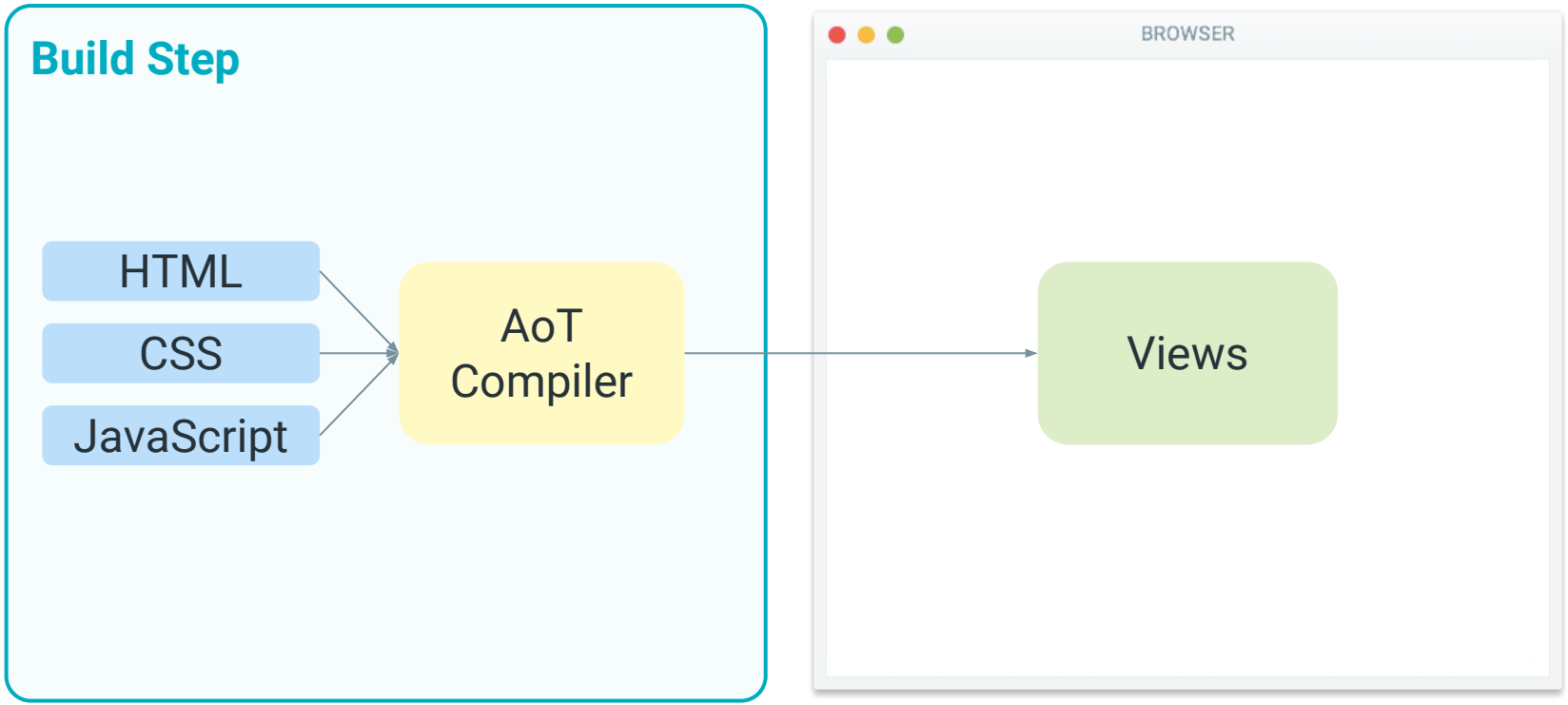
Optimized for JavaScript Virtual Machines

Change detection
10x faster!

Angular 2 Compiler Perf Improvement



Ahead of Time (AoT) Compilation



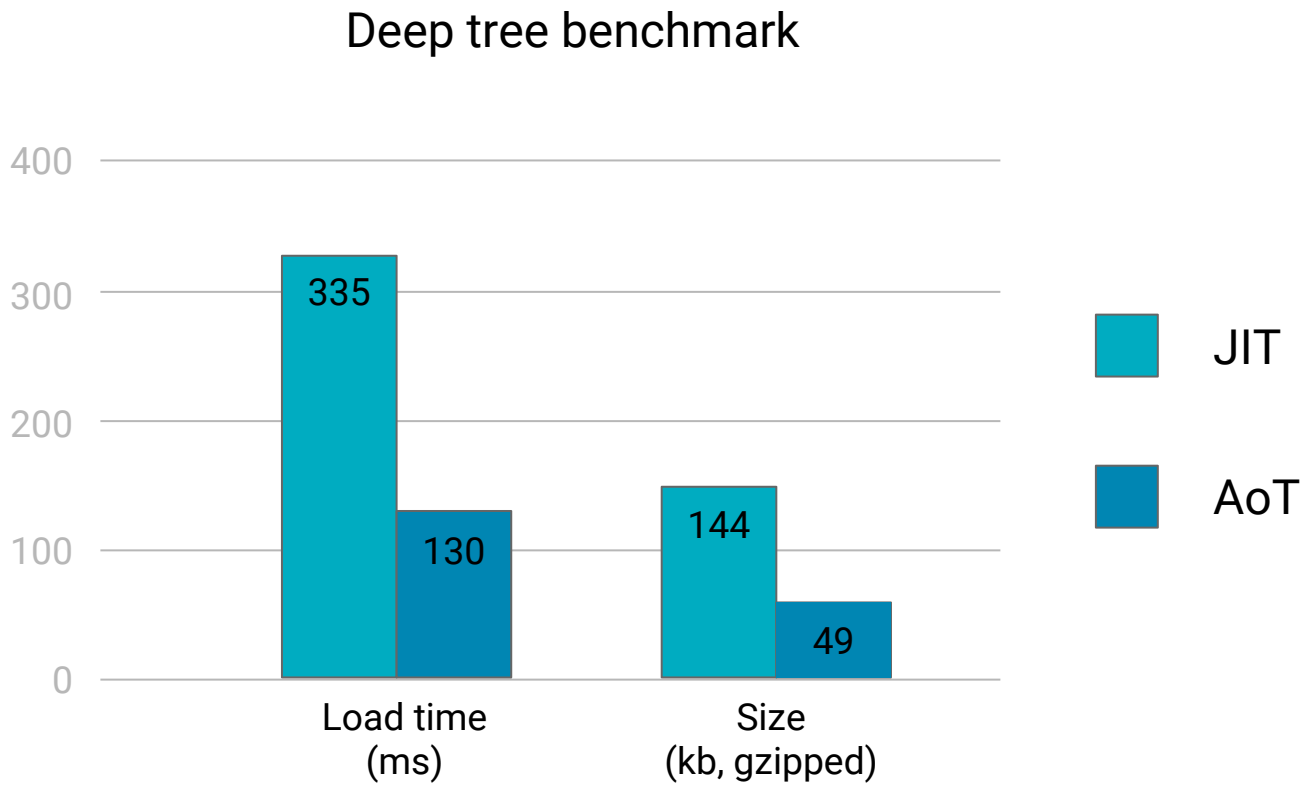
Ahead of Time (AoT) Compilation

Optimized for JavaScript Virtual Machines

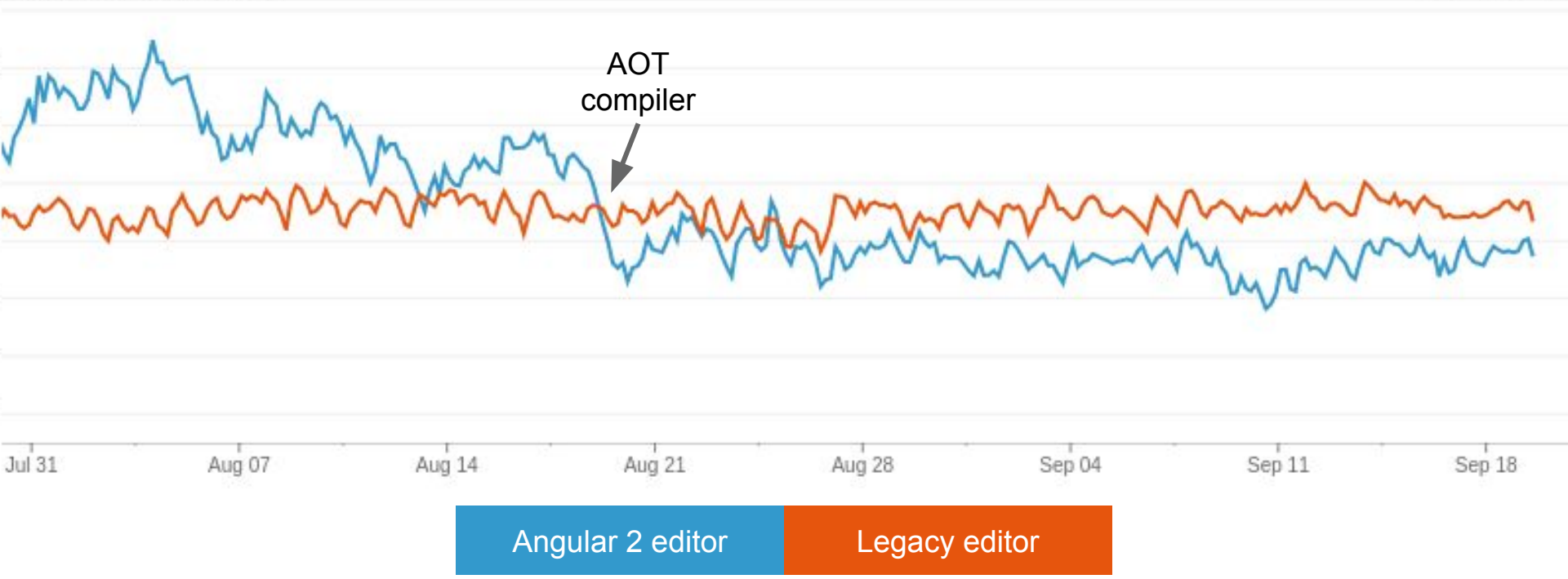
No in-browser parsing or compilation

Ship less code

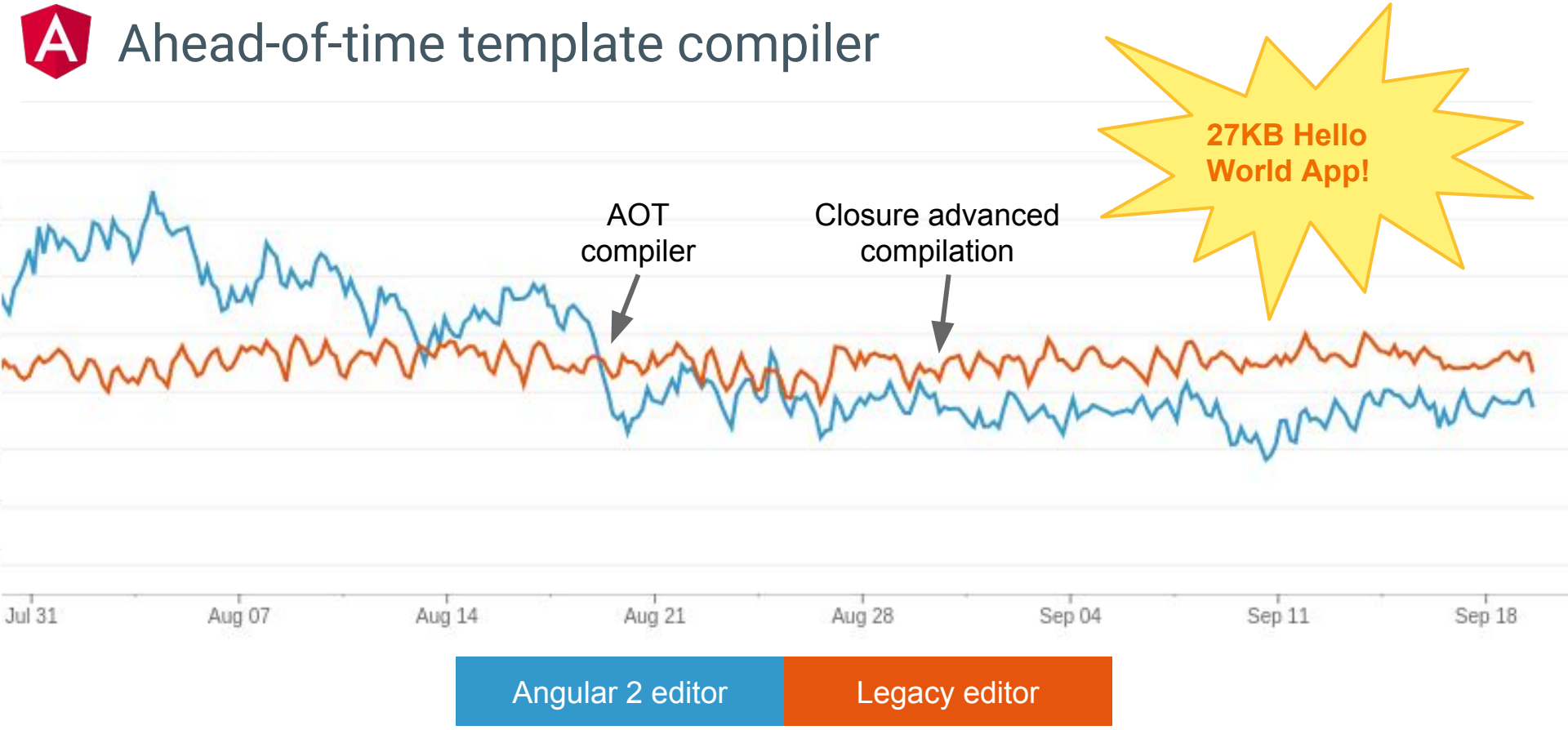
JIT vs AOT Compilation - Page Load Performance



Ahead-of-time template compiler

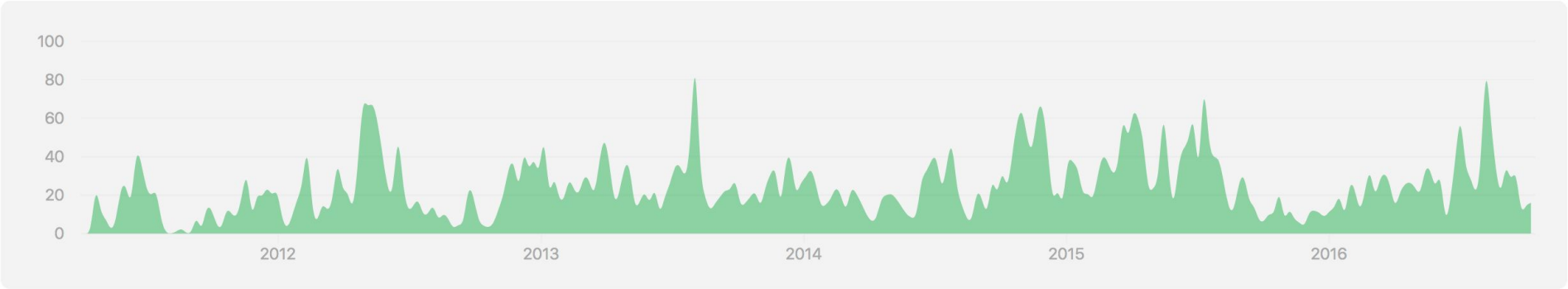


Ahead-of-time template compiler



ember

ember



Release Date: December 8, 2011

Contributors: 620

License: MIT

Created by Yehuda Katz and Tom Dale

Based on SproutCore

A framework for building ambitious web applications.

ambitious web applications

- created by many people
- maintained for a long time
- upgraded not rewritten

Features

- ✓ Routing
- ✓ Component Architecture
- ✓ DOM Diffing
- ✓ One Way & Two Way Property binding
- ✓ HTML Templating
- ✓ Service Composition

3 biggest strengths

1. Framework for large applications

Contributing to web standards

- Participate in TC39
- Polyfill new features
- Provide feedback

Contributing and adopting good ideas

- Introduced 1st class URL based routing
- Introduced CLI development environment
- Adopted unidirectional data flow and re-render everything from React

Predictable Upgrade Path

- Strict adherence to Semantic Versioning
- 6 week releases for early adopters
- LTS releases for slower adopters
- Guide upgrades with deprecation warnings

2. Mature Ecosystem

Supported by many companies

3000+ addons

- Ease integration
- Complete solutions
- Primitives
- Experimental & Deprecated features

3. Comprehensive Tooling

Ember CLI

- Standard development environment
- Codifies conventions
- Common build process

Ember FastBoot

- Server side rendering
- Speed up initial render
- Node.js server

Ember Engines

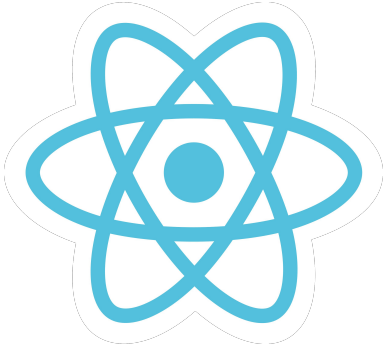
- Big app = many small apps
- Apps are addons
- Lazy load support

ThoughtWorks®

ADOPT

If you are faced with building a single-page application (SPA) and trying to choose a framework to build with, Ember.js has emerged as a leading choice.... The Ember CLI build tooling is a haven in the storm of JavaScript build tools, and the Ember core team and community are highly active and responsive.

https://www.thoughtworks.com/radar/languages-and-frameworks/ember-js



Frontend Frameworks

Panel Discussion