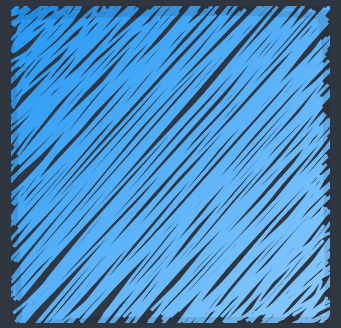
Web Components:

Natif

StencilJS



LitElement

Julien Renaux

GDE Web, Freelancer

@julienrenaux

https://julienrenaux.fr/

toptal™

Igor Laborie

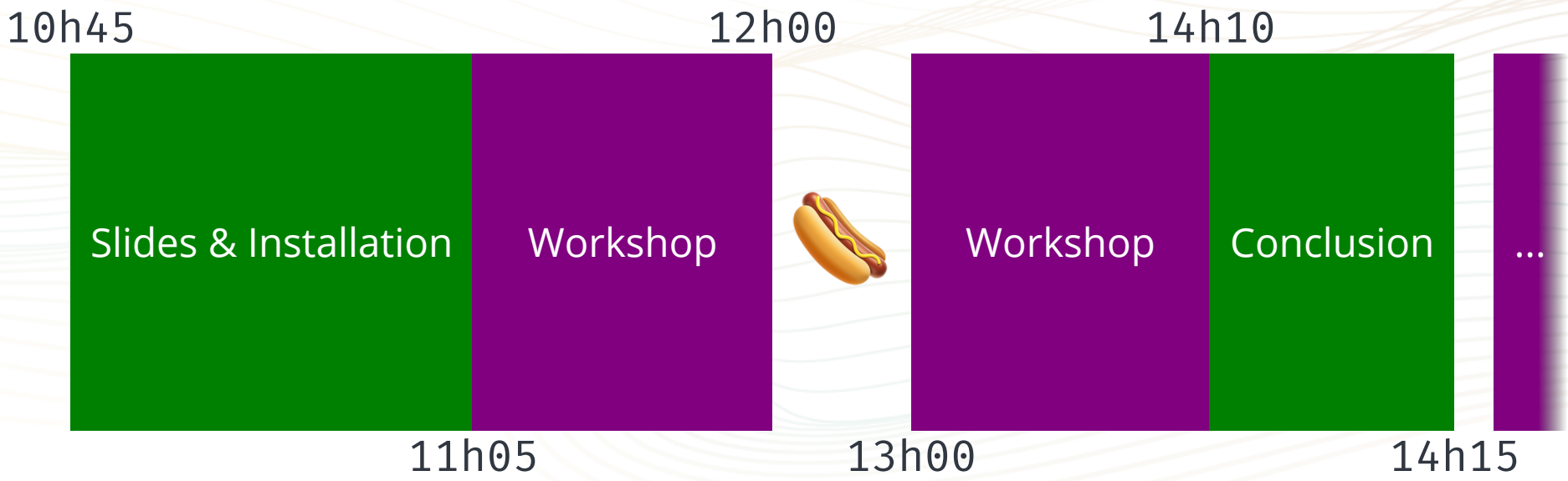
Expert Web & Java

@ilaborie

igor@monkeypatch.io

<Monkey Patch />

Roadmap

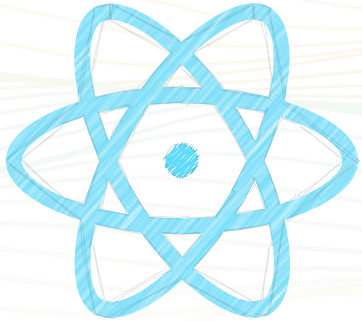

10h45
12h00
14h10
Slides & Installation
Workshop
Workshop
Conclusion
...
11h05
13h00
14h15

Instructions

TODO: Installation instructions, Wifi

Développons une application Web en 2019

Commençons par choisir un Framework



React

VueJs

Angular

Puis comment nous allons écrire le styles

- CSS
- Sass/Scss
- Less
- Stylus
- CSS-in-JS
- PostCSS
- NextCSS
- ...

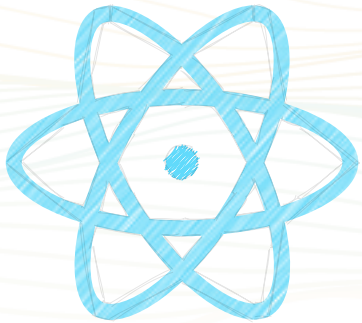
Puis construisons notre application

- Webpack
- ParcelJs
- RollupJs
- Bazel

Développer une applicatione en JS n'est plus simple...

Ça va trop vite ...

...Interoperabilité ne vient plus gratuitement...

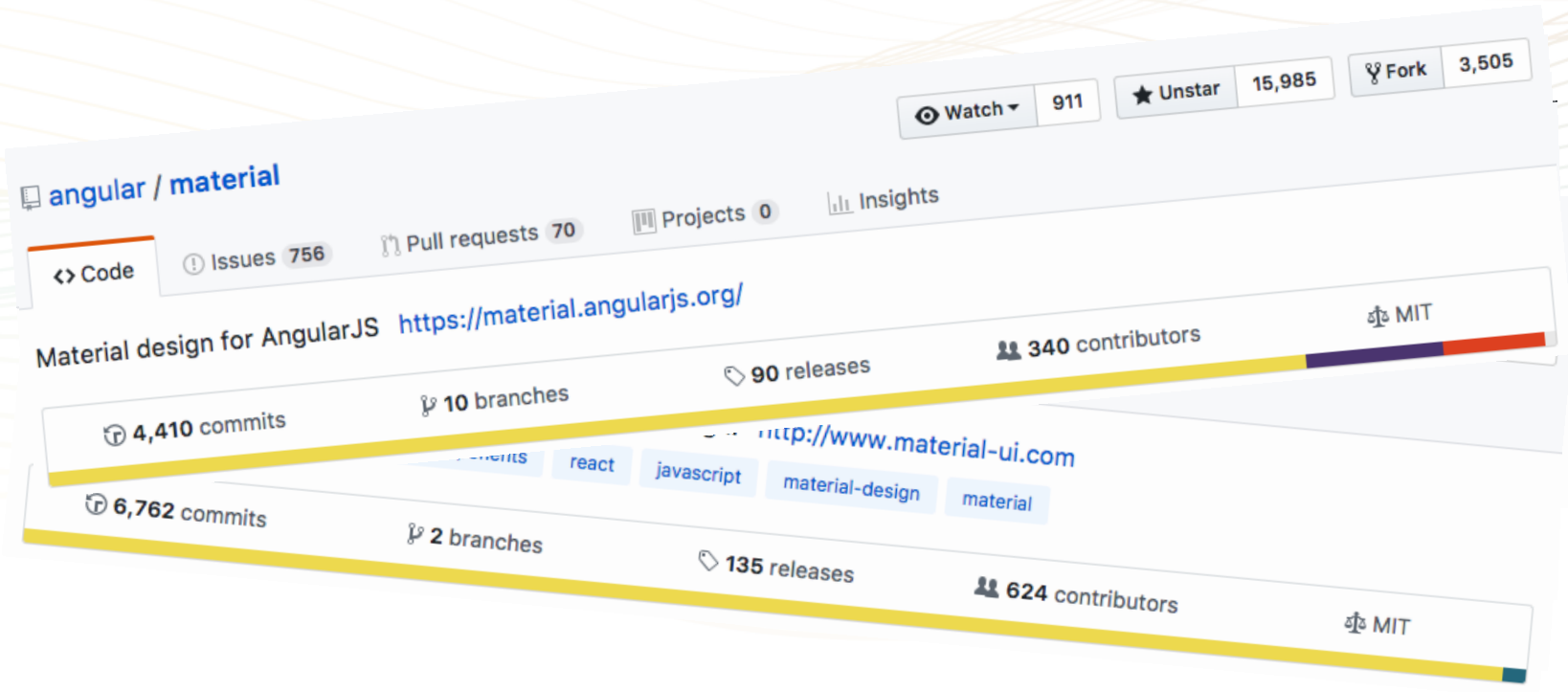


React

VueJs

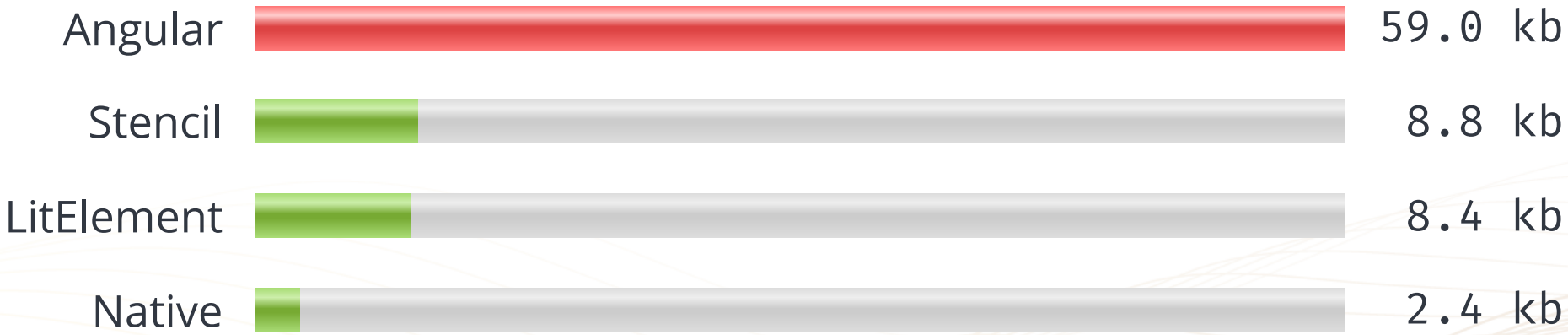
Angular

...et on réinvente sans arrêt la roue !



TODO: Meme Luci Do It: add another one in WebComponent LitElement...

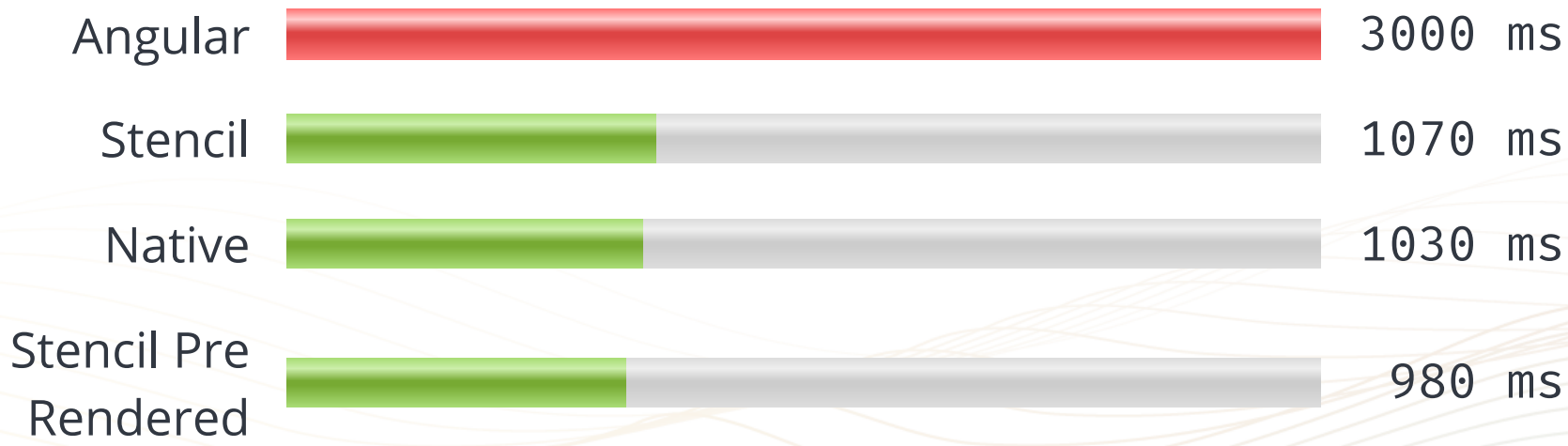
Toute cette complexité doit s'arréter avec les Web Components



Size matters (Gzipped)

😨 Stencil et litElement sont 5 fois plus petit qu'Angular

😱 Native est 23 fois plus petit qu'Angular



Le temps ça compte (FMP 3G 📱 en ms)

😨 Native et Stencil sont 3 fois plus rapide qu'Angular

Web Components

Spécifier par le **World Wide Web Consortium** (W3C)

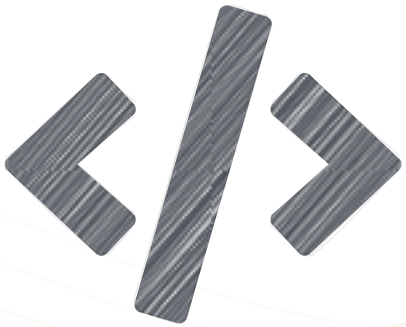
Débuté en **2012**

Custom Elements

Shadow DOM

HTML templates

HTML Imports



Custom Elements

> *Les Custom Elements sont la capacité de créer une propes balise HTML avec ses propres attributes et méthodes*

Shadow DOM

> *Le Shadow DOM fournit l'encapsulation du DOM et du CSS*

HTML templates

> *Définit un bloc d'HTML réutilisable au moment de l'exécution*

```js
class PopUpInfo extends HTMLElement {
    constructor() {
        super();
        // ...
    }
    // ...
}

customElements.define('popup-info', PopUpInfo);
```

Support des navigateurs


11


18


65


72


12

Custom Elements (V1)	😫	😫	😀	😀	🙂
Shadow DOM (V1)	😫	😫	😀	😀	🙂
HTML templates	😫	😀	😀	😀	😀

Supported

Partial Support

Not Supported

https://caniuse.com

Support avec le polyfill



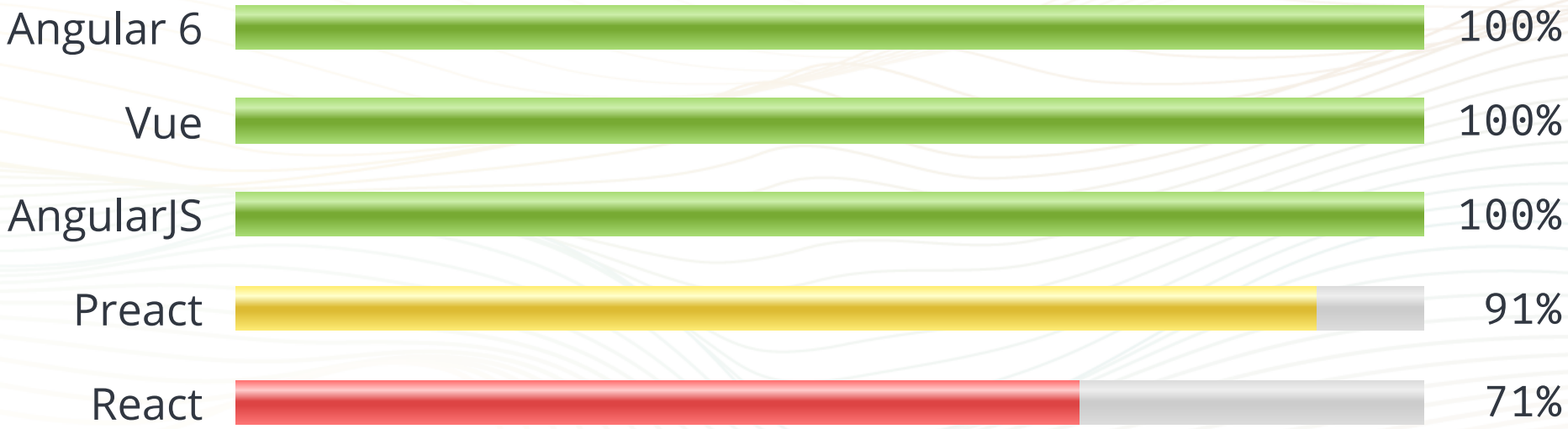


11+ 9+

Custom Elements (V1)	✔	✔	✔	✔	✔
Shadow DOM (V1)	✔	✔	✔	✔	✔
HTML templates	✔	✔	✔	✔	✔

https://github.com/webcomponents/webcomponentsjs

Interopérabilité des Web Component



https://custom-elements-everywhere.com/

StencilJS

https://stenciljs.com/

Projet **Open Source**, ➳ MIT License

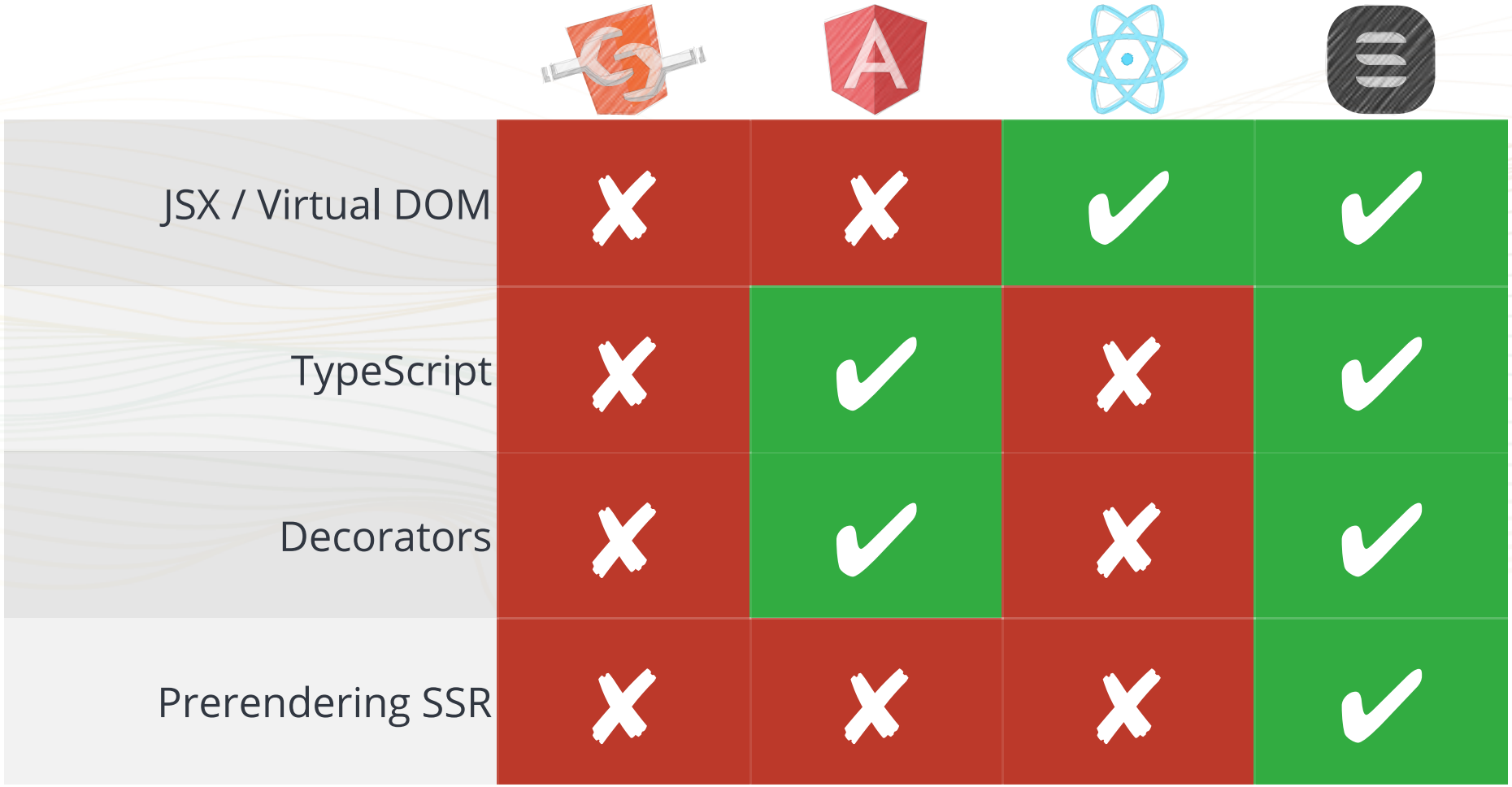
Créer par l'équipe d'**Ionic** en 2017

5.1k ⭐ sur github

StencilJS n'est pas un autre framework

StencilJS c'est un compileur qui génère des web components

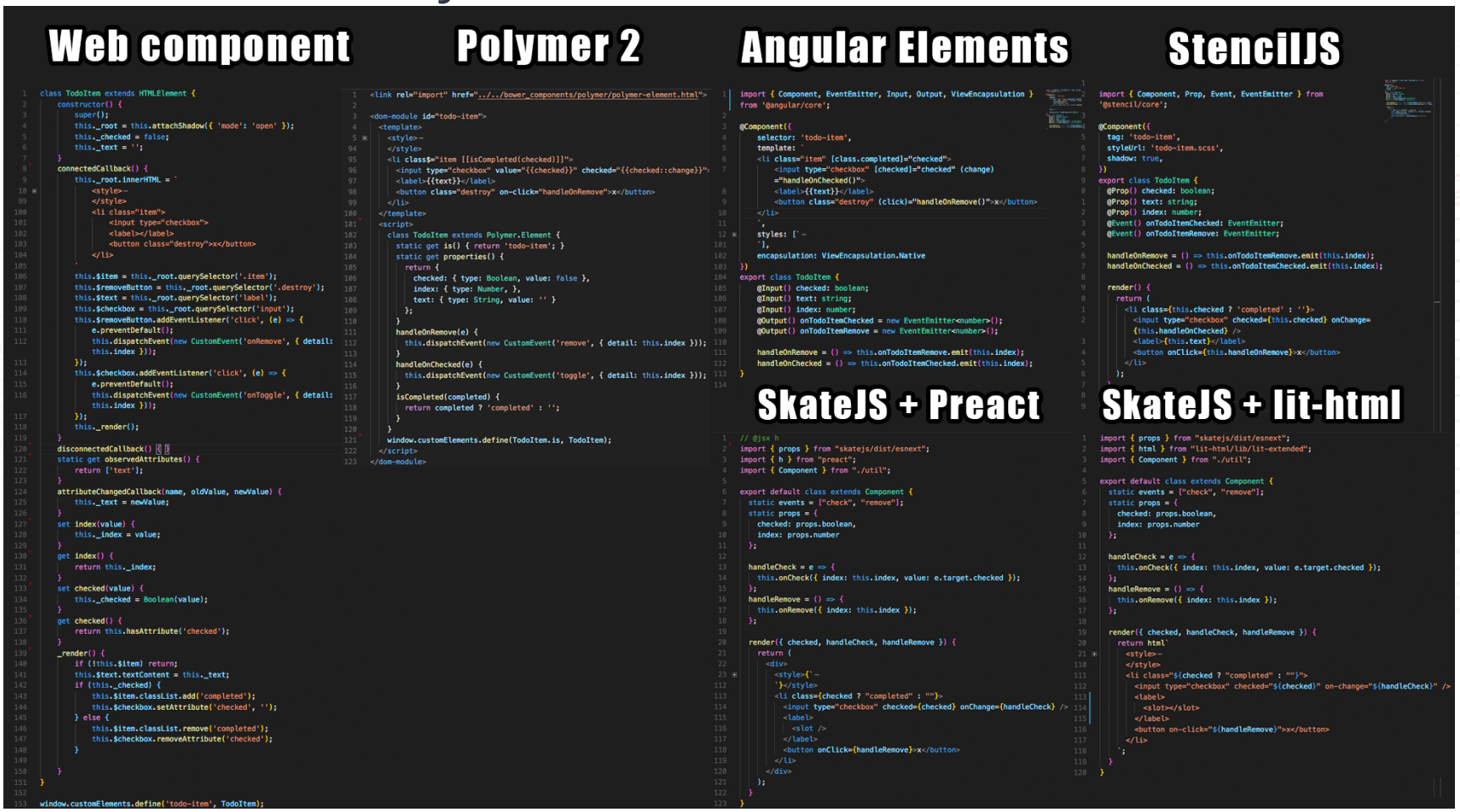
StencilJS c'est un ensemble de bons outils



StencilJS marche partout

Il charge les polyfills à la demande

La syntax de Stencil est concise



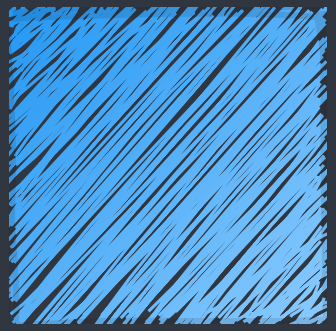
Pour démarrer

```
$ npm init stencil
```

```
? Pick a starter › - Use arrow-keys. Return to submit.
❯  ionic-pwa     Everything you need to build fast, production ready PWAs
   app           Minimal starter for building a Stencil app or website
   component     Collection of web components that can be used anywhere
```

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

@Component({
    tag: 'my-first-component',
    styleUrl: 'my-first-component.scss'
})
export class MyComponent {
    // Indicate that name should be
    // a public property on the component
    @Prop() name: string;

    render() {
        // JSX
        return (<p>My name is {this.name}</p>);
    }
}
```

Lit-Elements



https://lit-element.polymer-project.org/

Projet **Open Source**, ➳ BSD 3-Clause License

Créer par l'équipe **Polymer Team** en 2017

1.7k ⭐ sur github

Utilise la bibliothèque de template lit-html

https://lit-html.polymer-project.org/

Basé sur les **templates HTML**

Avec les ➳ *Template literals* de ES2015

```js
import {html, render} from 'lit-html';

// A lit-html template uses
// the `html` template tag:
const sayHello = (name) =>
    html`<h1>Hello ${name}</h1>`;

// It's rendered with the `render()` function:
render(sayHello('World'), document.body);

// And re-renders only update the
// data that changed, without VDOM diffing!
render(sayHello('Everyone'), document.body);
```

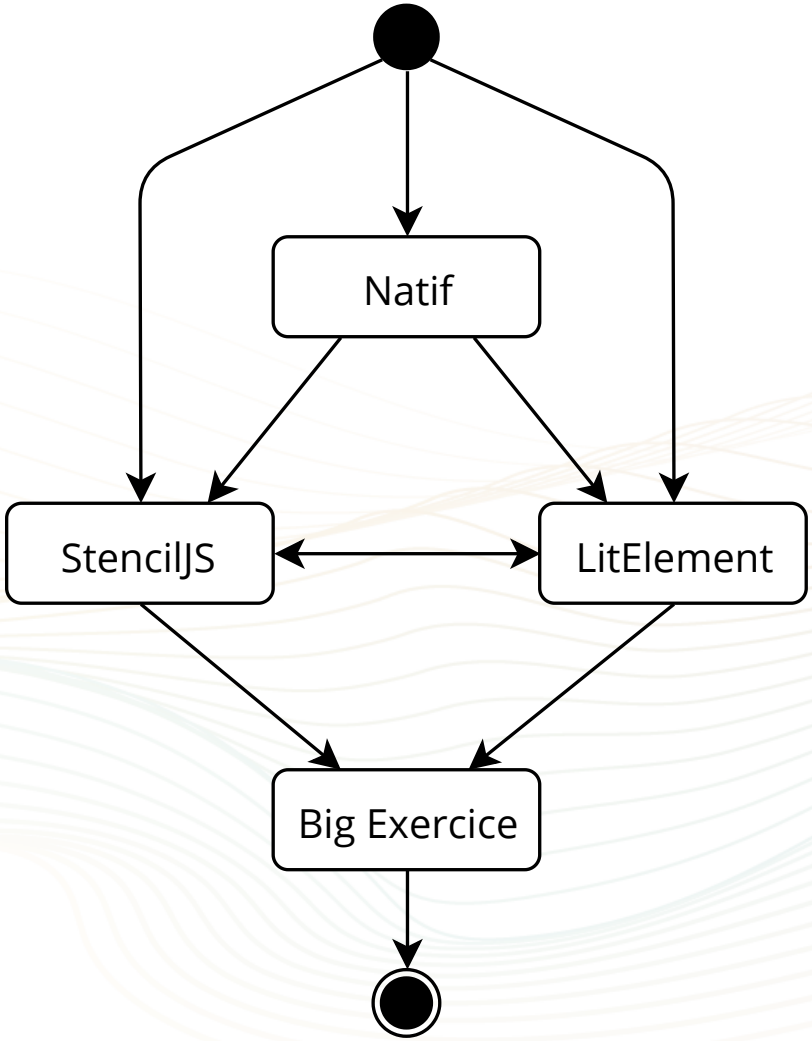
```js
import { LitElement, html, css } from 'https://unpkg.com/lit-element/l

class MyElement extends LitElement {
    static get properties() {
        return {
            mood: { type: String }
        }
    }
    static get styles() {
        return css`.mood { color: green; }`;
    }
    render() {
        return html `Web Components are
            <span class="mood">${this.mood}</span>!`;
    }
}

customElements.define('my-element', MyElement);
```

```ts
import { /* ... */ } from 'lit-element';

@customElement('my-element')
class MyElement extends LitElement {

    @property({ type: String }) mood;

    static get styles() {
        return css`.mood { color: green; }`;
    }

    render() {
        return html`Web Components are
            <span class="mood">${this.mood}</span>!`;
    }
}
```

✅ Chrome, Safari, Opera, Firefox

💊polyfills pour Edge et IE 11

Workshop


Natif
StencilJS
LitElement
Big Exercice

Conclusion

Construire une Application

Utiliser un gestionnaire d'état externe

Style avec un thème

Utilisé les *custom properties* CSS

Support des navigateurs

Utiliser le polyfill ou Electron

Alternatives modernes

- SkateJS
- Svelte
- Slim.js

...

Fin

Merci

Pensez à nous faire des retours (votez !)