Web components en 2018

On en est où ?

Horacio Gonzalez

@LostInBrittany

OVH
Innovation for Freedom

Who are we?

Introducing myself and introducing OVH

Horacio Gonzalez

OVH
Innovation for Freedom

@LostInBrittany

Spaniard lost in Brittany,
developer, dreamer and
all-around geek

OVH
Team DevRel

Breizh C@mp
Mix de technologies

Finist Devs

WARP 10
MEETUP

Google Developers Experts
2015
Web Technologies
GDE

OVH : Key Figures

OVH
Innovation for Freedom

1.3M Customers worldwide in 138 Countries
1.5 Billions euros investment over five years
30 Datacenters (growing)
350k Dedicated Servers
200k Private cloud VMs running
650k Public cloud Instances created in a month
15TB bandwidth capacity
35 Points of presence
4TB Anti DDoS capacity
Hosting capacity : 1.3M Physical Servers

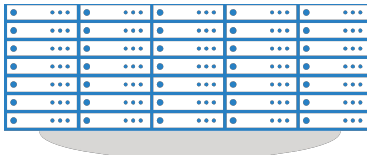
+ 2 500 Employees in 19 countries
18 Years of Innovation

OVH: A Global Leader on Cloud

OVH
Innovation for Freedom

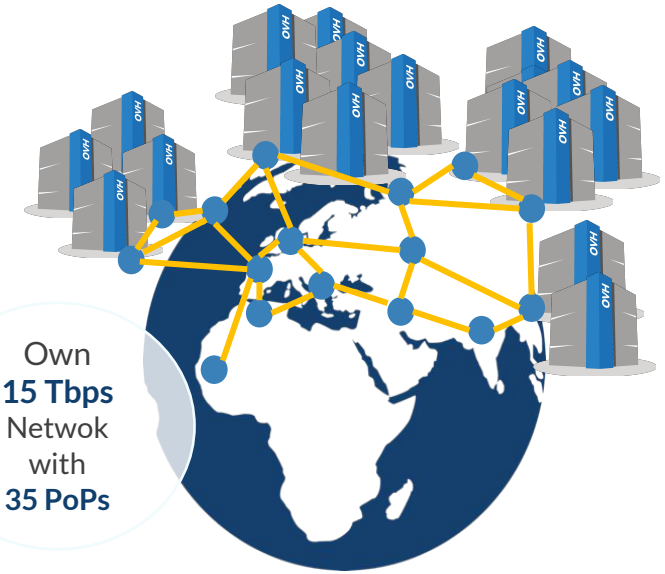
200k Private cloud VMs running

1 **Dedicated IaaS Europe**



Hosting capacity : **1.3M** Physical Servers

360k Servers already deployed



Own **15 Tbps** Netwok with **35 PoPs**

2018
27 Datacenters

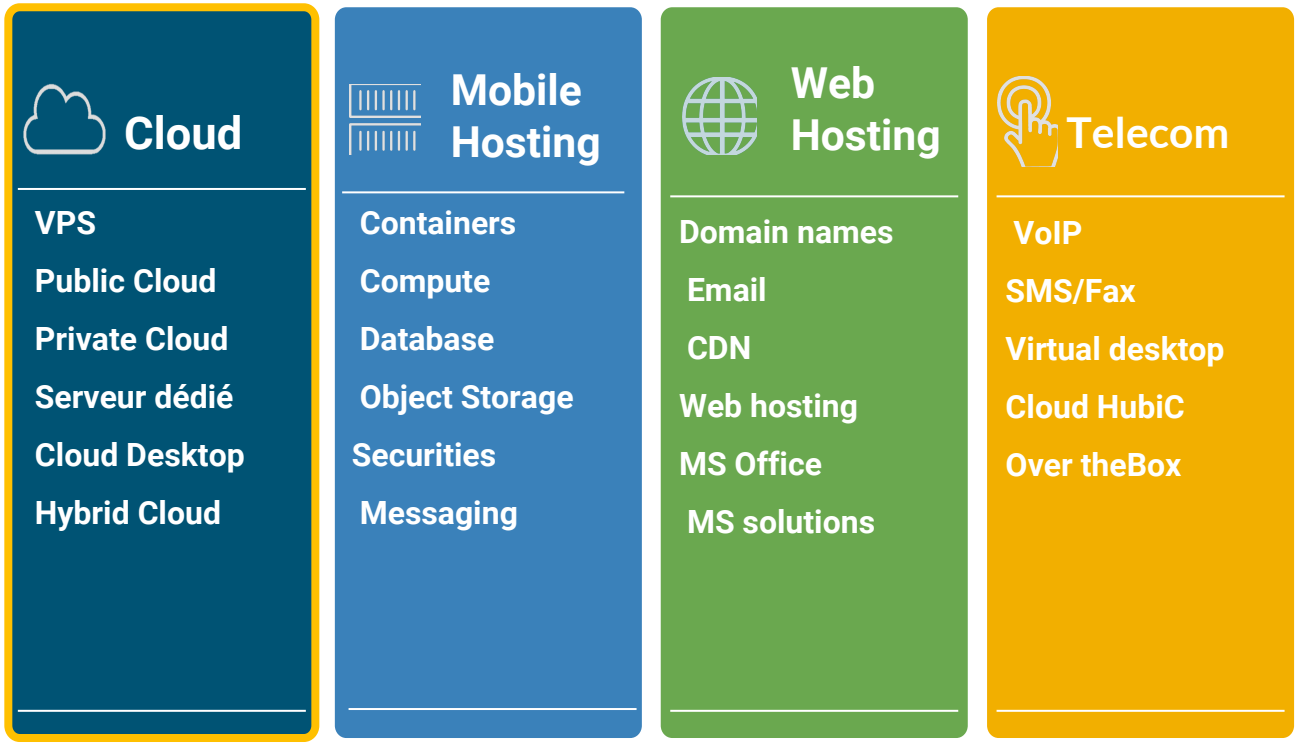
2020
50 Datacenters

> **1.3M** Customers in **138** Countries

OVH

OVH: Our solutions

OVH
Innovation for Freedom



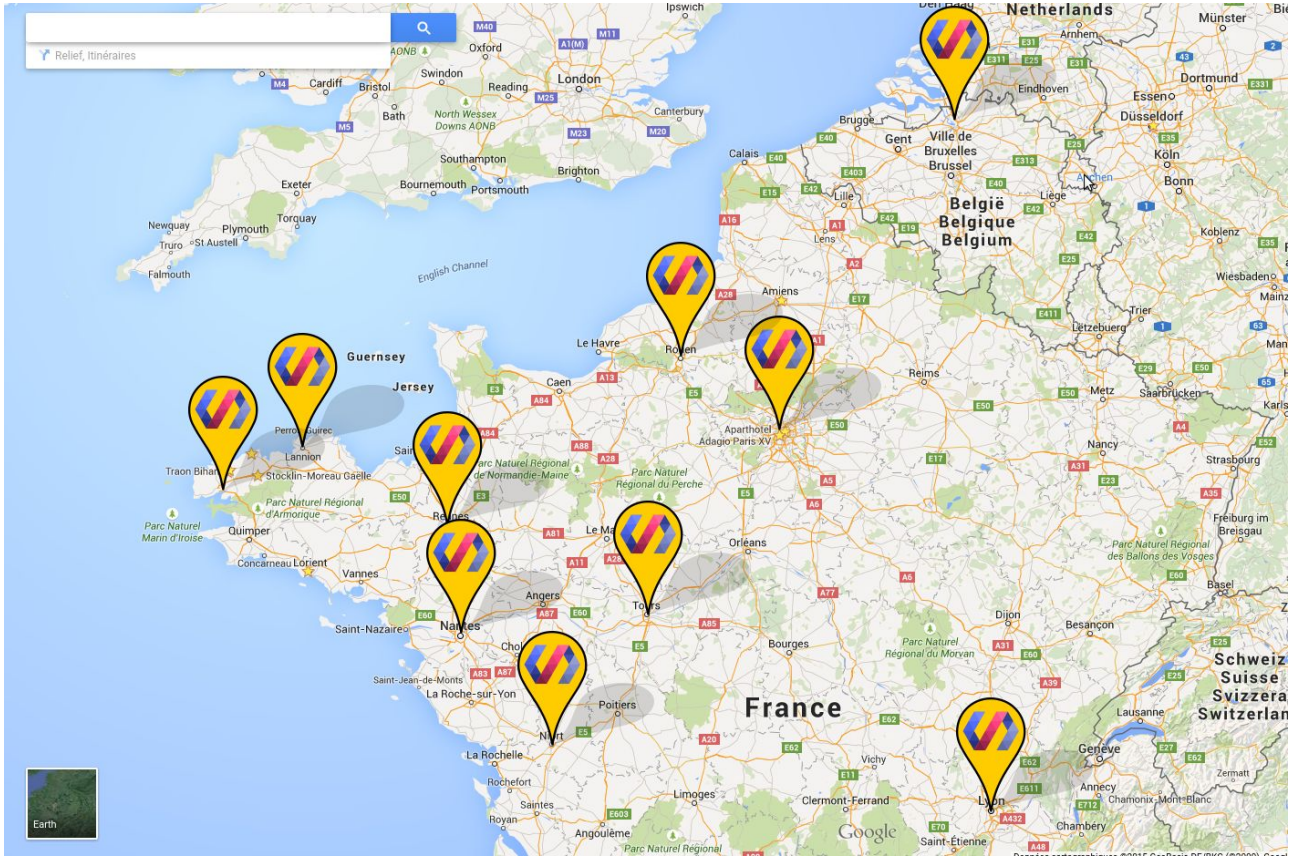
OVH
Innovation for Freedom

Sometimes I feel a bit grumpy

The stories of the grumpy old speaker...

On Polymer tour since 2014

OVH
Innovation for Freedom



Web components == Revolution

OVH
Innovation for Freedom

bu.edu

Building a world brick by brick

OVH
Innovation for Freedom

BitRebels Brickset

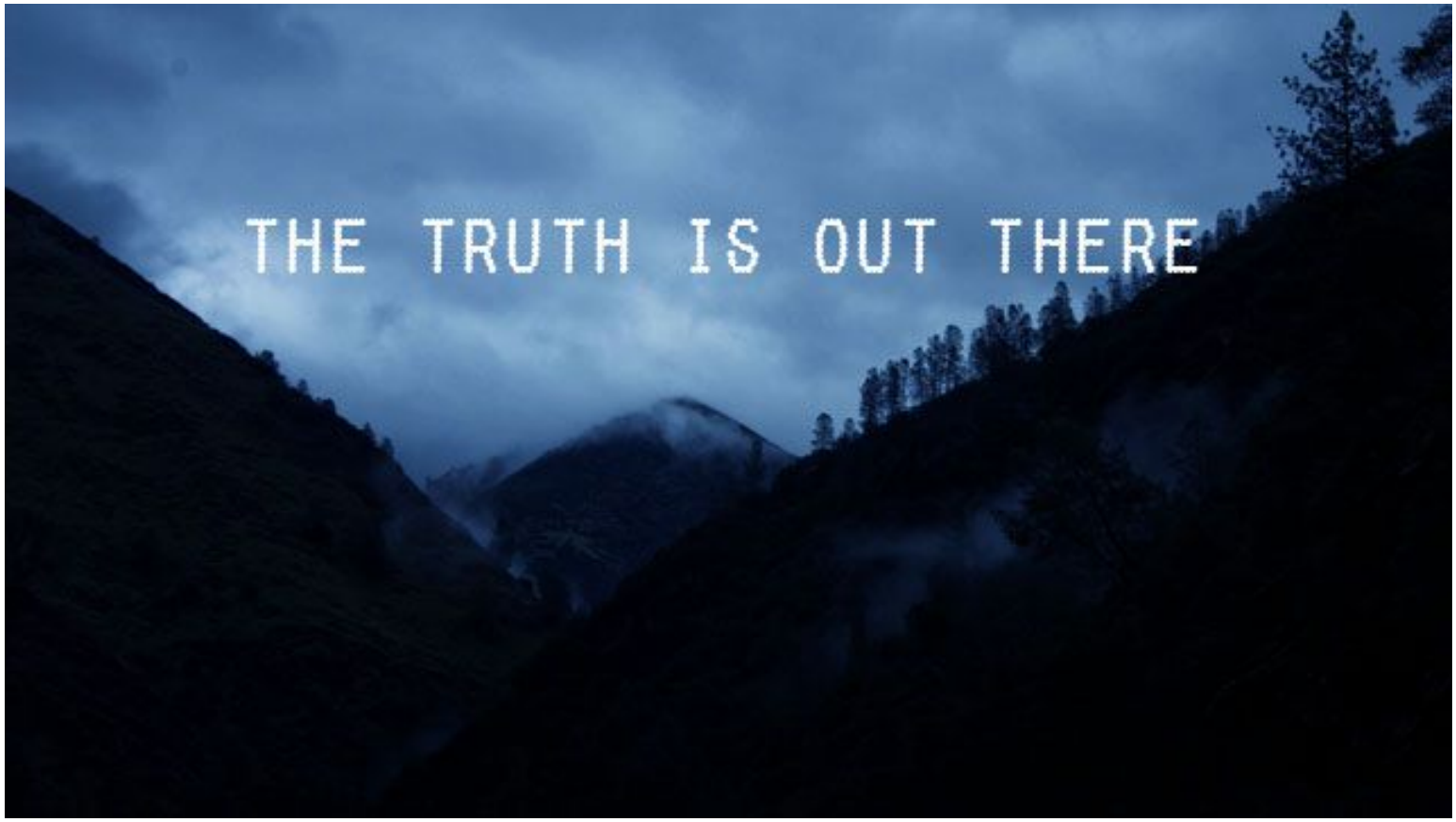
Is the promise unfulfilled?

OVH
Innovation for Freedom

It's almost 2019, where is your revolution, dude?

Is it a conspiracy?

OVH
Innovation for Freedom



Am I only a dreamer?

OVH
Innovation for Freedom

Well, revolution IS there

OVH
Innovation for Freedom

But it's a silent one...

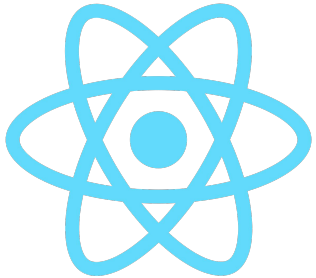
They are there, in everyday sites

OVH
Innovation for Freedom

More than you can imagine

The components architecture won

OVH
Innovation for Freedom



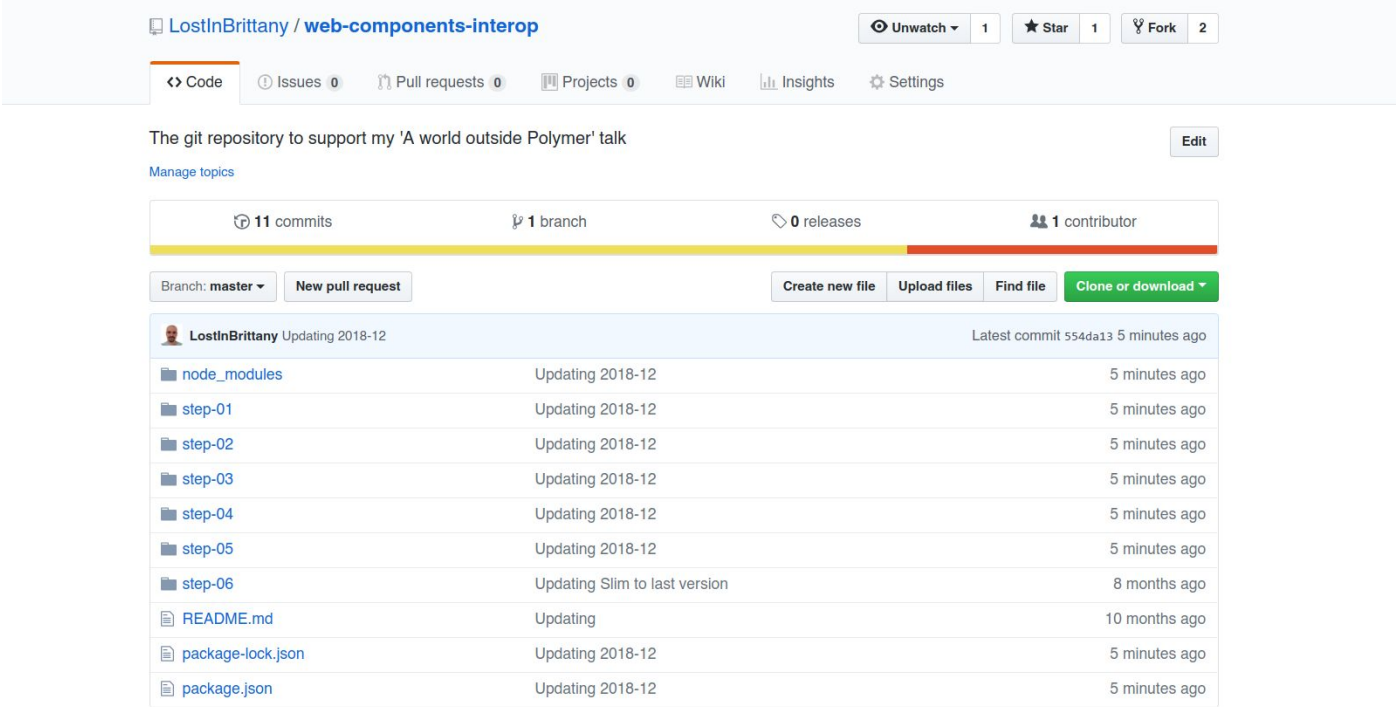
Components, components everywhere

OVH
Innovation for Freedom

Web Components

We want the code!

OVH
Innovation for Freedom



https://github.com/LostInBrittany/web-components-interop

A very basic web component

OVH
Innovation for Freedom

```
class MyElement extends HTMLElement {

 // This gets called when the HTML parser sees your tag
 constructor() {
   super(); // always call super() first in the ctor.
   this.msg = 'Hello, Bordeaux JUG!';
 }

 // Called when your element is inserted in the DOM or
 // immediately after the constructor if it's already in the DOM
 connectedCallback() {
   this.innerHTML = `<p>${this.msg}</p>`;
 }

}

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


Custom Elements:

OVH
Innovation for Freedom

- Let you define your own HTML tag with bundled JS behavior
- Trigger lifecycle callbacks
- Automatically “upgrade” your tag when inserted in the document

Custom Elements don't:

OVH
Innovation for Freedom

- Scope CSS styles
  - Shadow DOM
- Scope JavaScript
  - ES2015
- "Reproject" children into <slot> elements
  - Shadow DOM

Adding ShadowDOM

OVH
Innovation for Freedom

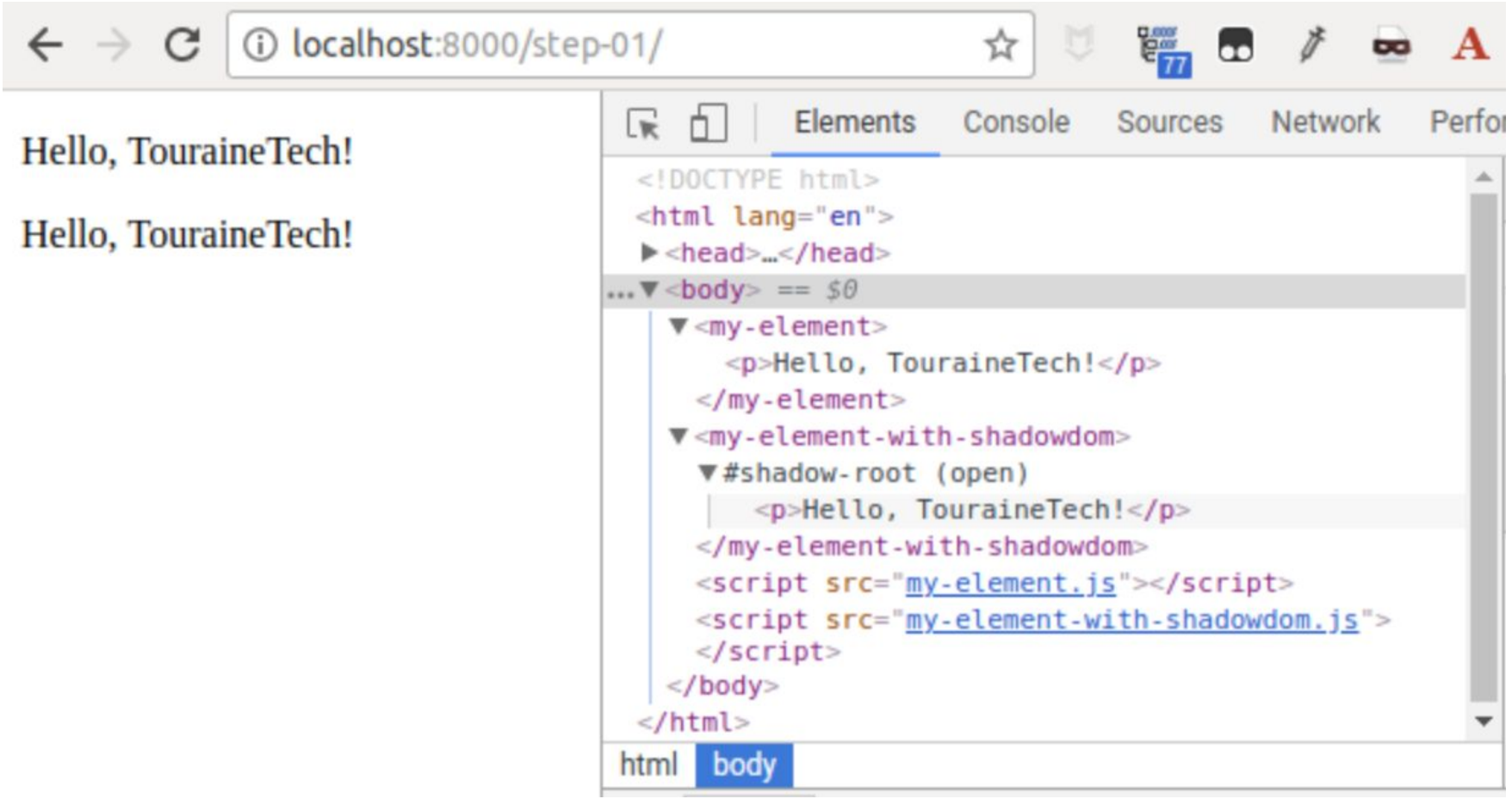
```
class MyElementWithShadowDom extends HTMLElement {

 // This gets called when the HTML parser sees your tag
 constructor() {
   super(); // always call super() first in the ctor.
   this.msg = 'Hello, Bordeaux JUG!';
   this.attachShadow({ mode: 'open' });
 }
 // Called when your element is inserted in the DOM or
 // immediately after the constructor if it's already in the DOM
 connectedCallback() {
   this.shadowRoot.innerHTML = `<p>${this.msg}</p>`;
 }
}

customElements.define('my-element-with-shadowdom', MyElementWithShadowDom);
```


Adding ShadowDOM

OVH
Innovation for Freedom



Lifecycle callbacks

OVH
Innovation for Freedom

```
class MyElementLifecycle extends HTMLElement {
 constructor() {
   // Called when an instance of the element is created or upgraded
   super(); // always call super() first in the ctor.
 }
 // Tells the element which attributes to observer for changes
 // This is a feature added by Custom Elements
 static get observedAttributes() {
   return [];
 }
 connectedCallback() {
   // Called every time the element is inserted into the DOM
 }
 disconnectedCallback() {
   // Called every time the element is removed from the DOM.
 }
 attributeChangedCallback(attrName, oldVal, newVal) {
   // Called when an attribute was added, removed, or updated
 }
 adoptedCallback() {
   // Called if the element has been moved into a new document
 }
}
```


my-counter custom element

OVH Innovation for Freedom

```
class MyCounter extends HTMLElement {

 constructor() {
   super();
   this._counter = 0;
   this.attachShadow({ mode: 'open' });
 }

 connectedCallback() { this.render() }
  static get observedAttributes() { return [ 'counter' ] }

 attributeChangedCallback(attr, oldVal, newVal) {
   if (oldVal !== newVal) {
     this[attr] = newVal;
   }
 }
```


my-counter custom element

OVH
Innovation for Freedom

```
get counter() { return this._counter; }
set counter(value) {
  if (value != this._counter) {
    this._counter = Number.parseInt(value);
    this.setAttribute('counter', value);
    this.display();
  }
}

increment() {
  this.counter = this.counter + 1;
  this.dispatchEvent(new CustomEvent(
       'increased', {detail: {counter: this.counter}}));
}
```


my-counter custom element

OVH Innovation for Freedom

```
render() {
  let button = document.createElement('button');
  button.innerHTML = '+';
  button.addEventListener('click', this.increment.bind(this));
  this.shadowRoot.appendChild(button);

  this.output = document.createElement('span');
  this.shadowRoot.appendChild(this.output);

  this.style.display = 'block';
  this.style.fontSize = '5rem';
  button.style.fontSize = '5rem';
  button.style.borderRadius = '1rem';
  button.style.padding = '0.5rem 2rem';
  this.output.style.marginLeft = '2rem';
}

display() {
  this.output.innerHTML = `${this.counter}`;
}
```


`my-counter` custom element

OVH
Innovation for Freedom

OVH
Innovation for Freedom

Polymer

Adding syntactic sugar to the standard

Polymer has evolved in 2018

OVH
Innovation for Freedom

Image: © Nintendo

Polymer 3 is here!

Classes, JavaScript modules...

OVH
Innovation for Freedom

```
import {html, PolymerElement} from
    '/node_modules/@polymer/polymer/polymer-element.js';

class MyPolymerCounter extends PolymerElement {
 static get template() {
    <style>
       :host {
         font-size: 5rem;
       }
       button {
         font-size: 5rem; border-radius: 1rem; padding: 0.5rem 2rem;
       }
     </style>
     <button on-click="increment">+</button>
     <span>[[counter]]</span>
   `;
 }
```


But it's still mostly syntactic sugar

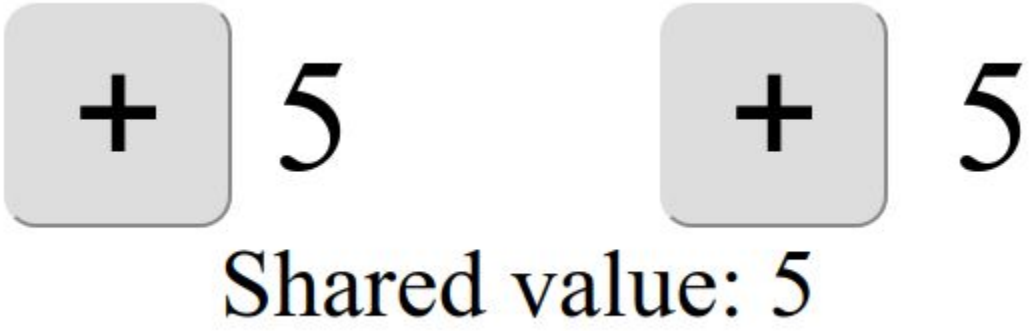
OVH
Innovation for Freedom

```
 static get properties() {
   return {
     counter: { type: Number, reflectToAttribute:true, value: 0 }
   };
 }
 increment() {
   this.counter = Number.parseInt(this.counter) + 1;
   this.dispatchEvent(new CustomEvent('increased', {detail: {counter: this.counter}}));
 }
}

window.customElements.define('my-polymer-counter', MyPolymerCounter);
```


And they are still custom elements

OVH Innovation for Freedom



100% interoperable

Interoperation pattern

OVH
Innovation for Freedom

```
<div class="container">
  <my-polymer-counter
      counter="[[value]]"
      on-increased="_onCounterChanged"></my-polymer-counter>
  <my-counter
      counter="[[value]]"
      on-increased="_onCounterChanged"></my-counter>
</div>
<div class="container">
  <div class="value">Shared value: [[value]]</div>
</div>
```

Attributes for data in

Events for data out

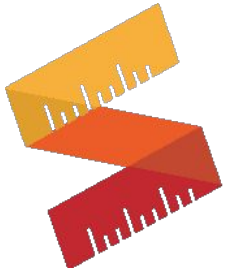
OVH
Innovation for Freedom

To infinity and beyond!

There is a world outside Polymer

Lots of web components libraries

OVH
Innovation for Freedom



slim.js

STENCIL

For different need and sensibilities

Lots of web components libraries

OVH
Innovation for Freedom

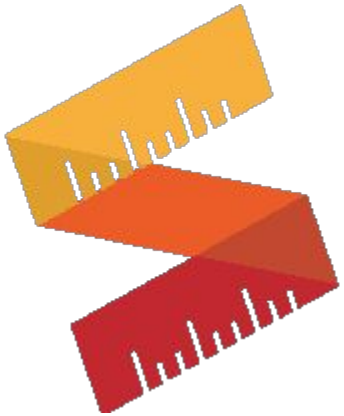
Angular Elements

Vue Web Component Wrapper

And the frameworks work on it too!

OVH
Innovation for Freedom

Slim.js



Slim.js

OVH
Innovation for Freedom

Slim.js Documentation | Submit an issue | Submit a documentation issue

- Getting started
- Creating an element
  - Lifecycle
  - Accessing children
  - Conditional rendering
  - Shadow-DOM
- Data binding
- Repeating elements
- Plugins
- Extending Slim & Directives

Getting started with Slim.js

Slim.js is a lightweight web component authoring library that provides extended capabilities for components (such as data binding) using es6 native class inheritance. This library is focused on providing developers the ability to write robust web components quickly and without the hassle of unnecessary dependencies or an overhead of a framework.

Installation

```
npm install slim-js
yarn add slim-js
bower install slimjs
```

Web-Components Spec. & Polyfills

Slim.js is based on the browser's native DOM API's web-components spec. For browsers that do not support this natively, a polyfill is required.

You can load the polyfill directly from the html file

```
<script src="https://cdnjs.cloudflare.com/ajax/libs/webcomponentsjs/1.0.17/webcomponents-lite.js"></script>
```


Slim.js

OVH
Innovation for Freedom

- Lightweight web component library
- Extended capabilities for components
  - data binding
- Using es6 native classes and modules
- Without Shadow DOM by default

Like a lighter and lesser-featured Polymer

Slim.js

OVH
Innovation for Freedom

```
import {Slim} from '/node_modules/slim-js/Slim.js'

let template = `
   <style>
     ...
   </style>
   <div class="container">
       <div class="button"  slim-id="button" click='increment'>
           <img src="./img/slim.png">
       </div>
       <div class="value" bind> {{toString(counter)}} </div>
   </div>`;

Slim.tag('my-slim-counter', template,

class extends Slim {
...
});
```


Slim.js

OVH
Innovation for Freedom

```
class extends Slim {
   // native API
   static get observedAttributes () {
       return ['counter'];
     }
   // bind attributes to properties
   get autoBoundAttributes() {
       return ['counter'];
    }
   increment() {
     this.counter = this.counter + 1;
     this.dispatchEvent(new CustomEvent('increased', {detail: {counter: this.counter}}));
   }
   toString(value) {
     return value.toString();
   }
});
```


Skatejs

Skatejs

OVH
Innovation for Freedom

Skatejs

OVH
Innovation for Freedom

- Lightweight web component library
- Abstracts away attribute / property semantics
- Very very fast
- Can use many renderers
  - Basic innerHTML (default)
  - preact
  - lit-html

Skatejs

OVH Innovation for Freedom

```
import { props, withComponent } from '/node_modules/skatejs/dist/es/index.js';
import withLitHtml from '/node_modules/@skatejs/renderer-lit-html/dist/es/index.js';
import { html } from '/node_modules/lit-html/lit-html.js';

class MySkateCounter extends withComponent(withLitHtml()){

 static get props () {
   return {
     // By declaring the property an attribute, we can now pass an initial value
     // for the count as part of the HTML.
     counter: props.number({ attribute: true })
   };
 }
```


Skatejs

OVH
Innovation for Freedom

```
render({ counter }) {
  return html`
  <style>
    …
  <div class="container">
    <button class="button" @click="${() => this.increment()}">
      <img src="./img/skate.png" alt="Polymer">
    </button>
    <div class="value">${counter}</div>
  </div>`;
}
increment() {
  console.log('sdgws', this.counter)
  this.counter = Number.parseInt(this.counter) + 1;
  this.dispatchEvent(new CustomEvent('increased', {detail: {counter: this.counter}}));
}
}
window.customElements.define('my-skate-counter', MySkateCounter);
```


OVH
Innovation for Freedom

STENCIL

A new breed of Web Components

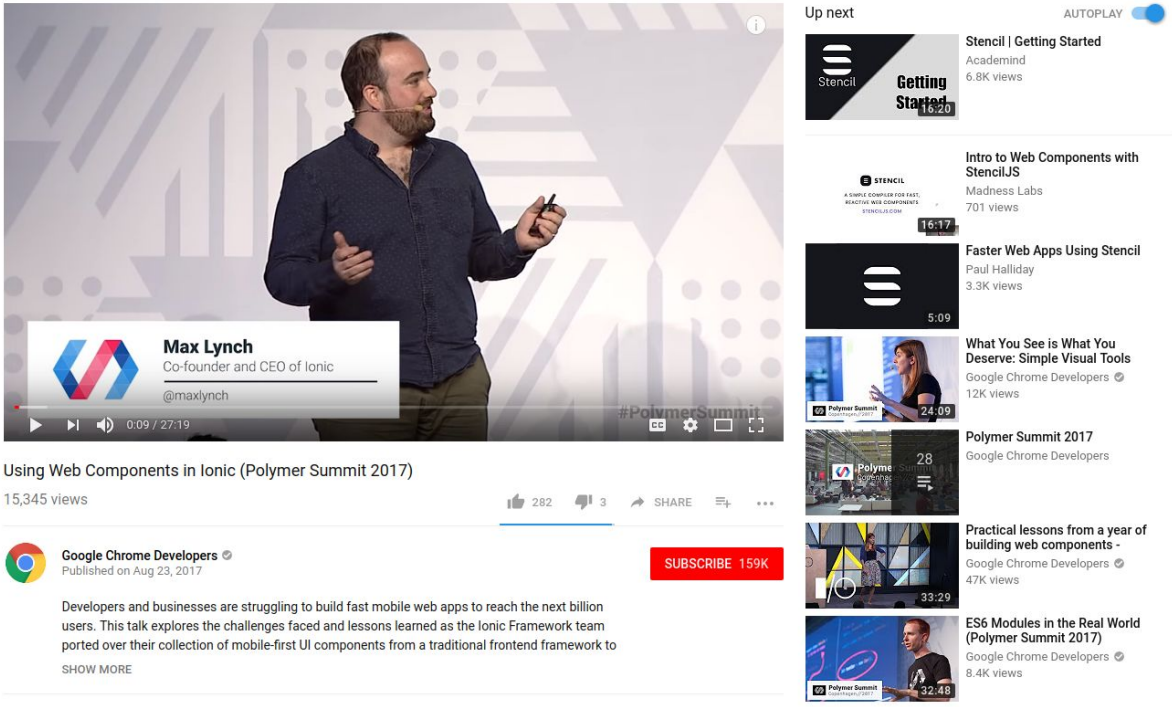
Next generation Ionic

OVH
Innovation for Freedom

Ionic 4 will be fully based on web components
using a new toolkit: Stencil

New kid on the block

OVH
Innovation for Freedom



Announced during Polymer Summit

Not another library

OVH
Innovation for Freedom

The magical, reusable web component compiler

Simple

With intentionally small tooling, a tiny API, zero configuration, and TypeScript support, you're set.

Performant

6kb min+gzip runtime, server side rendering, and the raw power of native Web Components.

Future proof

Build versatile apps and components based 100% on web standards. Break free of Framework Churn.

A Web Component compiler

A build time tool

OVH
Innovation for Freedom

To generate standard web components

Fully featured

OVH
Innovation for Freedom

- Virtual DOM
- Async rendering
- Reactive data-binding
- TypeScript
- JSX

And the cherry on the cake

OVH
Innovation for Freedom

Server-Side Rendering

Hands on Stencil

OVH
Innovation for Freedom

Clone the starter project

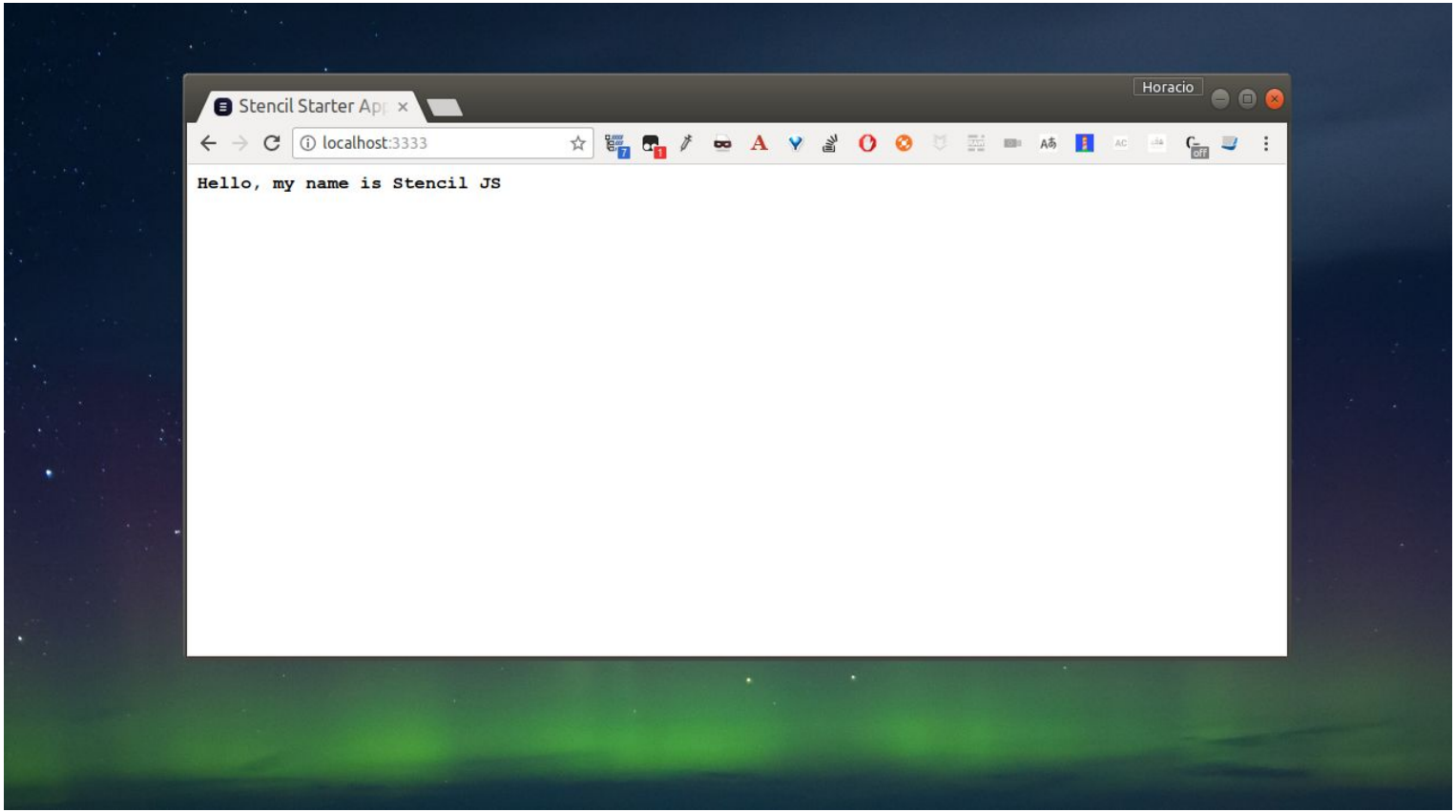
```
git clone https://github.com/ionic-team/stencil-app-starter my-app
cd my-app
git remote rm origin
npm install
```

Start a live-reload server

```
npm start
```

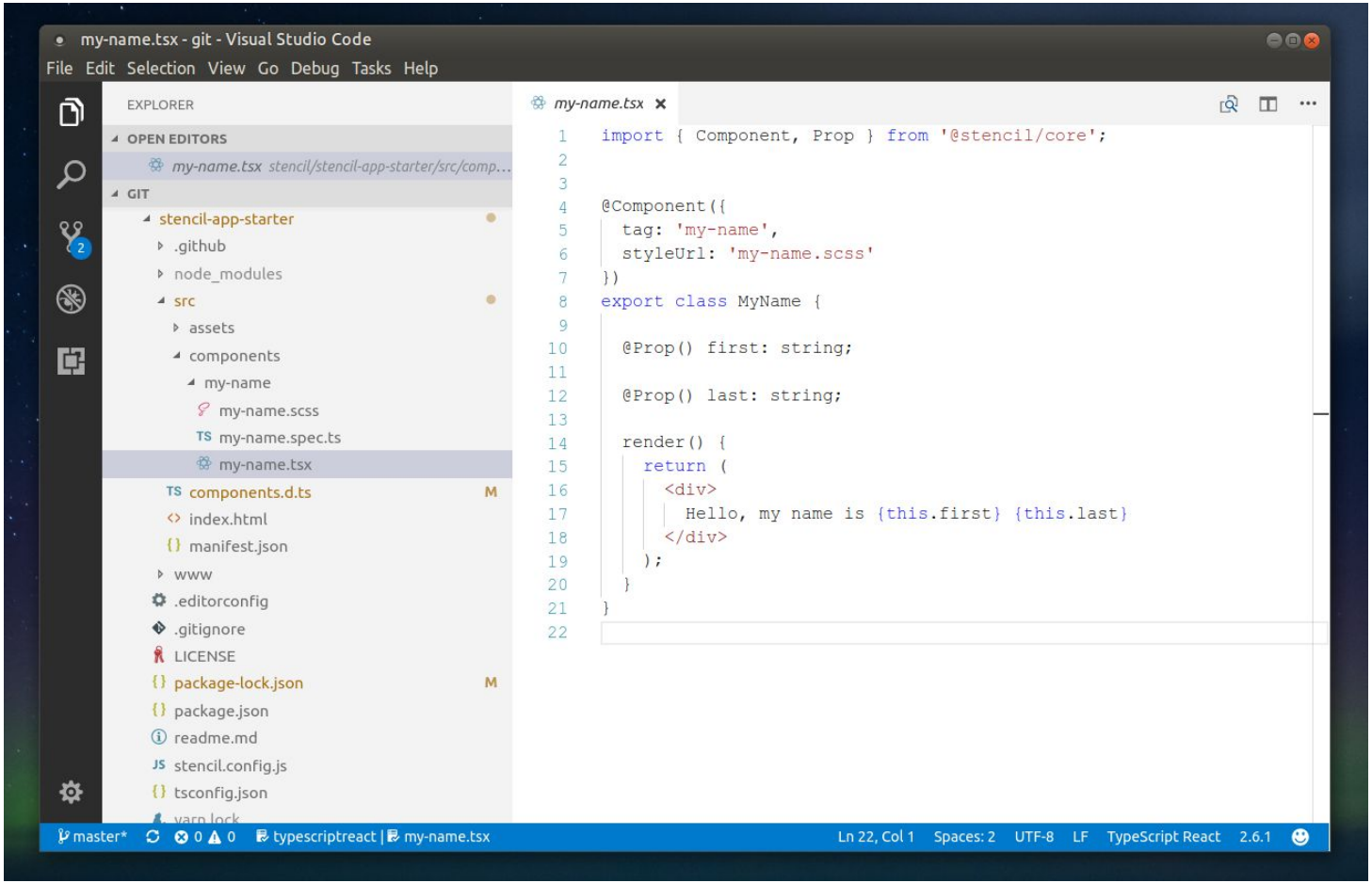

Hands on Stencil

OVH
Innovation for Freedom



Hands on Stencil

OVH
Innovation for Freedom



Some concepts

OVH
Innovation for Freedom

```
render() {
  return (
    <div>Hello {this.name}</div>
  )
}
```

```
render() {
  return (
    <div>{this.name ? <p>Hello {this.name}</p> : <p>Hello World</p>}</div>
  );
}
```

JSX declarative template syntax

Some concepts

OVH
Innovation for Freedom

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

@Component({
  tag: 'todo-list',
  styleUrl: 'todo-list.scss'
})
export class TodoList {
  @Prop() color: string;
  @Prop() favoriteNumber: number;
  @Prop() isSelected: boolean;
  @Prop() myHttpService: MyHttpService;
}
```

Decorators

Some concepts

OVH
Innovation for Freedom

```
import { Event, EventEmitter } from '@stencil/core';

...
export class TodoList {

  @Event() todoCompleted: EventEmitter;

  someAction(todo: Todo) {
    this.todoCompleted.emit(todo);
  }

  @Listen('todoCompleted')
  todoCompletedHandler(event: CustomEvent) {
    console.log('Received the custom todoCompleted event: ', event.detail);
  }
}
```

Events

Some concepts

OVH
Innovation for Freedom

```
@Component({
  tag: 'shadow-component',
  styleUrl: 'shadow-component.scss',
  shadow: true
})
export class ShadowComponent {

}
```

Optional Shadow DOM

Some concepts

OVH
Innovation for Freedom

stencil.config.js

```
exports.config = {
  namespace: 'myname',
  generateDistribution: true,
  generateWWW: false,
  ...
};
```

Generate distribution

Stencil

OVH
Innovation for Freedom

```
import { Component, Prop, PropWillChange, State, Event, EventEmitter } from '@stencil/core';

@Component({
 tag: 'stencil-counter',
 styleUrl: 'stencil-counter.scss',
 shadow: true
})
export class StencilCounter {
 @Prop() counter: number;
 @State() currentCount: number;
 @Event() currentCountChanged: EventEmitter;

 @Watch('counter')
 counterChanged(newValue: number) {
   this.currentCount = newValue;
 }

 componentWillLoad() {
   this.currentCount = this.counter;
 }

 increase() {
   this.currentCount++;
   this.currentCountChanged.emit({ counter: this.currentCount });
 }

 render() {
   return (
     <div class="container">
       <div class="button" onClick={() => this.increase()}> <img src="./img/stencil.png" /> </div>
       <div class="value" > {this.currentCount} </div>
     </div>
   );
 }
}
```


OVH
Innovation for Freedom

Conclusion

That's all folks!