



Web components en 2019, on en est où ?

Horacio Gonzalez

@LostInBrittany



Who are we?

Introducing myself and
introducing OVH



Horacio Gonzalez



@LostInBrittany

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



OVH : Key Figures



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



200k Private cloud
VMs running

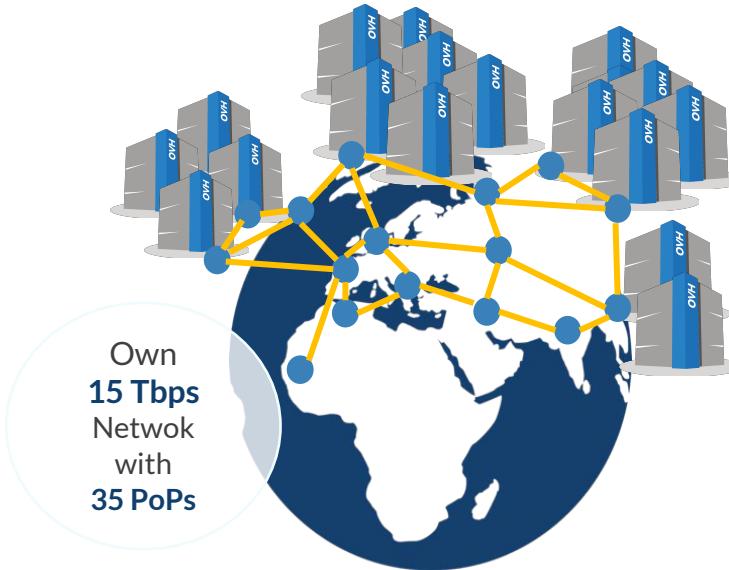


Dedicated IaaS
Europe



Hosting capacity :
1.3M Physical
Servers

360k
Servers already
deployed



2018
27 Datacenters



2020
50 Datacenters

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



OVH: Our solutions



Cloud

- VPS
- Public Cloud
- Private Cloud
- Serveur dédié
- Cloud Desktop
- Hybrid Cloud



Mobile Hosting

- Containers
- Compute
- Database
- Object Storage
- Securities
- Messaging



Web Hosting

- Domain names
- Email
- CDN
- Web hosting
- MS Office
- MS solutions



Telecom

- VoIP
- SMS/Fax
- Virtual desktop
- Cloud HubiC
- Over theBox



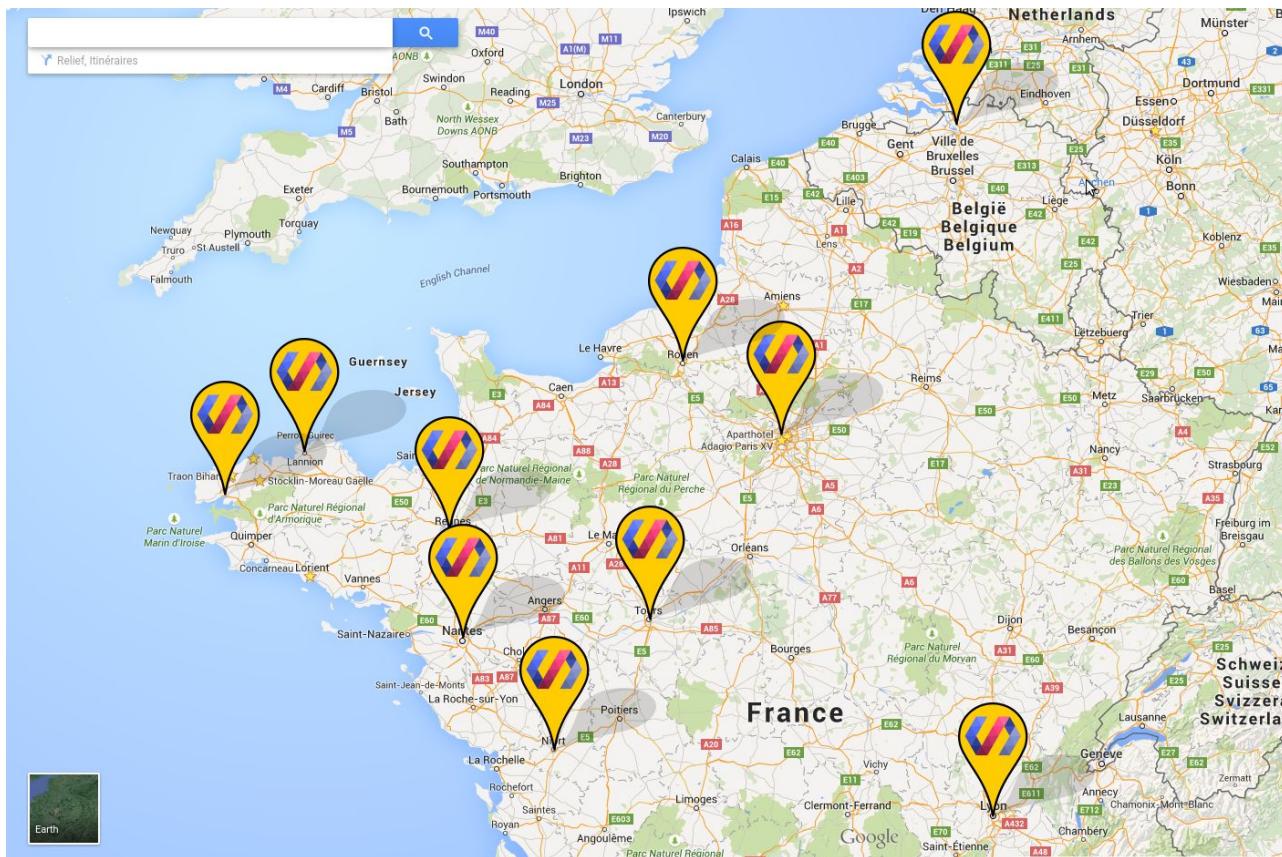
Sometimes I feel a bit grumpy



The stories of the grumpy old speaker...



On Polymer tour since 2014



Web components == Revolution



bu.edu



Building a world brick by brick



[BitRebels](#) | [Brickset](#)



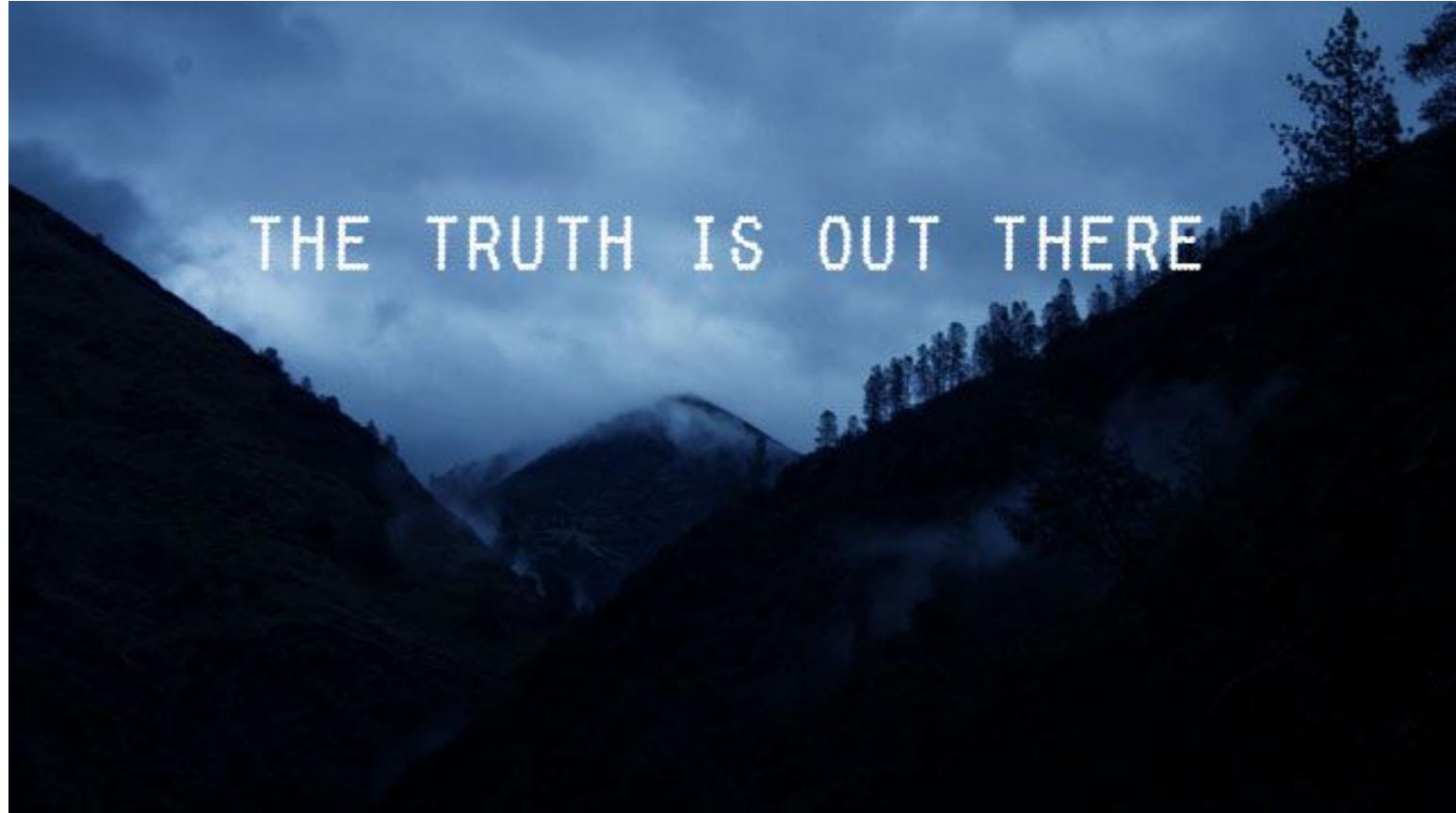
Is the promise unfulfilled?



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



Is it a conspiracy?



Am I only a dreamer?



Well, revolution IS there



But it's a silent one...



They are there, in everyday sites



Max Lynch @maxlynch

Following

Looks like Twitter tweet embeds are now shipped as Web Components? This is how WC's will take over: as plumbing most won't even realize they are using

```
<div id="posts" class="entry-content section-paragraph post">
  <p>...</p>
  <div style="margin: auto">
    <twitter-widget class="twitter-tweet twitter-tweet-rendered" id="twitter-tweet-1072527151092678657" data-tweet-id="1072527151092678657"> = $0
      <#shadow-root (open)>
        <style type="text/css">.SandboxRoot { display: none; max-height: 100%; width: 100%; }</style>
        <div data-twitter-event-id="0" class="SandboxRoot env-bp-350" style="border: 1px solid #ccc; border-radius: 4px; padding: 10px; width: fit-content; margin: auto">
          <div class="EmbeddedTweet EmbeddedTweet--cta js-clickToOpenTarget" target="https://twitter.com/maxlynch/status/1072527151092678657" data-tweet-id="1072527151092678657" data-twitter-widget-id="0" lang="en" data-twitter-event-id="4">
            <div class="EmbeddedTweet-tweetContainer">
```

9:22 PM - 11 Dec 2018

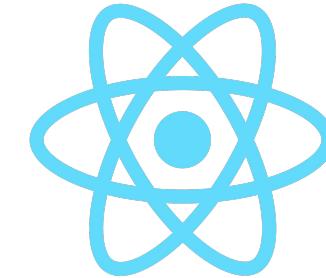
48 Retweets 163 Likes

9 48 163

More than you can imagine



The components architecture won



Components, components everywhere



Web components ARE platform



↻ Yann Bertrand and 2 others Retweeted

Justin Fagnani @justinfagnani · 23 Oct 2018

🎉🎊 Firefox shipped Web Components!!! 🎉🎊

Mozilla Hacks @mozhacks

Shadow DOM, Custom Elements, a handy (variable) font editor panel in @FirefoxDevTools, reduced-motion CSS media queries — treat yourself to a spook-tacular brand new @Firefox 63, available today! 🎃
hacks.mozilla.org/2018/10/firefo...

4

76

211



Truly part of the platform...



The 3 minutes context

What the heck are web component?



Web Components



CUSTOM
ELEMENTS



SHADOW
DOM



TEMPLATES



Custom Element



To define your own HTML tag

```
<body>
  ...
<script>
  window.customElements.define('my-element',
    class extends HTMLElement {...});
</script>
<my-element></my-element>
</body>
```



Shadow DOM



To encapsulate subtree and style in an element

Hello, world!



こんにちは、影の世界!

```
<button>Hello, world!</button>
<script>
var host = document.querySelector('button');
const shadowRoot = host.attachShadow({mode: 'open'});
shadowRoot.textContent = 'こんにちは、影の世界!';
</script>
```



Template



To have clonable document template

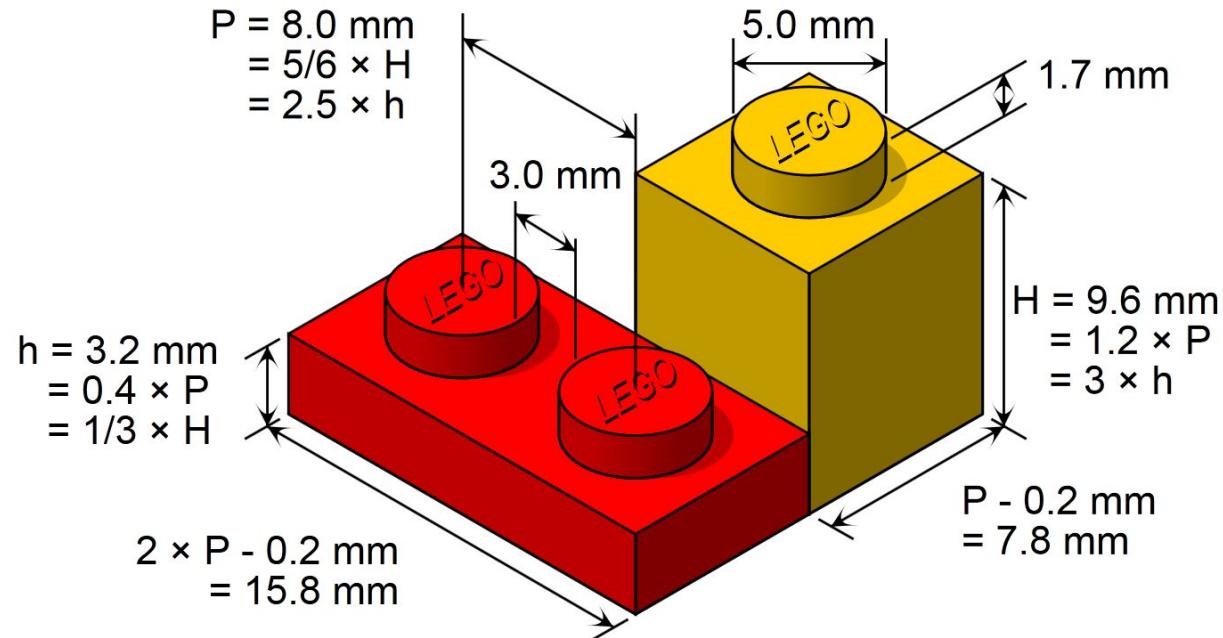
```
<template id="mytemplate">
  <img src="" alt="great image">
  <div class="comment"></div>
</template>
```

```
var t = document.querySelector('#mytemplate');
// Populate the src at runtime.
t.content.querySelector('img').src = 'logo.png';
var clone = document.importNode(t.content, true);
document.body.appendChild(clone);
```



But in fact, it's just an element...

- Attributes
- Properties
- Methods
- Events



Why those libs?

Why people don't use vanilla?



Web component standard is low level



At it should be!



Standard == basic bricks

Standard exposes an API to:

- Define elements
- Encapsulate DOM



Libraries are helpers



They give you higher-level primitives



Different high-level primitives



Each one tailored to a use



Sharing the same base



High-performant, low-level, in-the-platform
web components standard



Libraries aren't a failure of standard



They happen by design



Let's talk libraries

What's new?



Lot's of players, lot of evolutions



hybrids



SkateJS



slim.js



LitElement



snuggsi ツ



It's JavaScript after all...





Powering the new Ionic 4!



Not another library



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



Not a beta anymore

ionic 🎉 @Ionicframework · Jan 23

Ionic Framework 4.0 is here! 🎉

- ⚙️ Web Components = Stability + Compatibility
- ⚡ Better performance & startup times
- 💻 Made for developers & their preferred tooling
- 🎨 Powerful theming & CSS variables
- 🛠 Future-proof development, built in
- >New 4.0 docs



Introducing Ionic 4: Ionic for Everyone

Today I am thrilled to announce the 4.0 release of Ionic Framework, lovingly known as "Ionic for Everyone." 🎉 Ionic 4 represents the culmination of blog.ionicframework.com

Ionic 4 released, powered by Stencil!



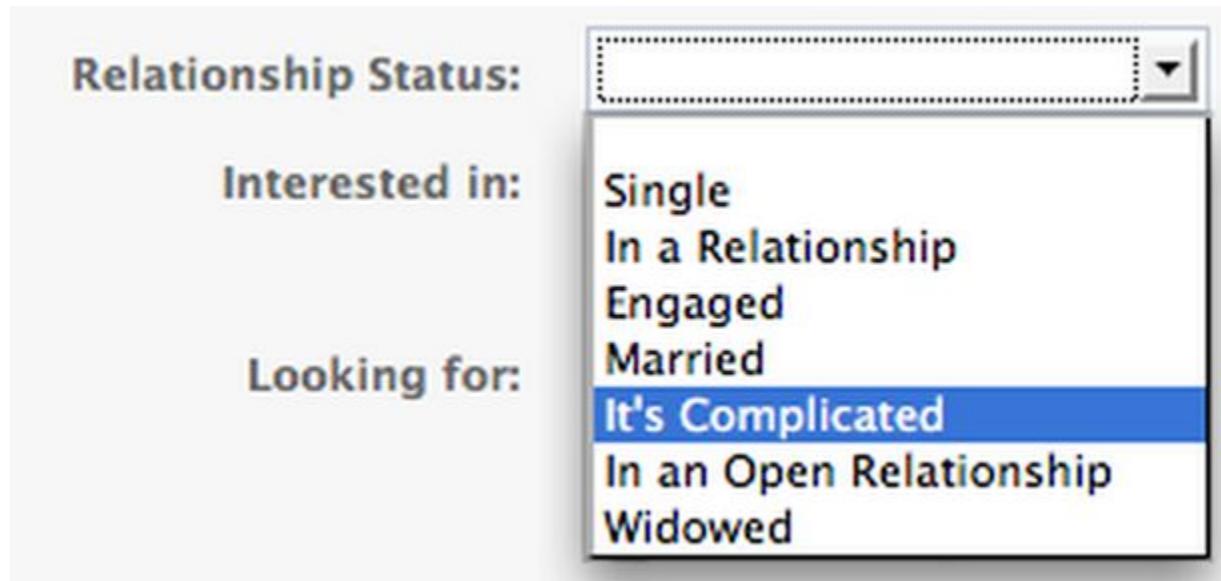


Polymer

Is the old player still alive?



What's Polymer status today?



Well, how could I say... it's complicated



It seems it's going to be deprecated...

Polymer Library

The Polymer library is our original web components library. Version 3.0 of the Polymer library brings web components into the mainstream, embracing JavaScript modules and npm.

Update your apps and elements for seamless interop with popular frameworks and tools and easy access to the full JavaScript ecosystem.

[WHAT'S NEW](#)[UPGRADE GUIDE](#)

Welcome to the future...

Starting a new project? Try our next-gen products, headed for release in the coming months:

[LitElement](#)

An ultra-light, highly performant custom element base class with a simple but expressive API.

[PWA Starter Kit](#)

Your one-stop shop for building Progressive Web Apps that make the most of the modern web.

[Material Web Components](#)

Pixel-perfect realizations of Google's Material Design spec that work anywhere on the web.

Technically yes... and that means good news!



Let's try to see clearer



Let's dive into Polymer history...



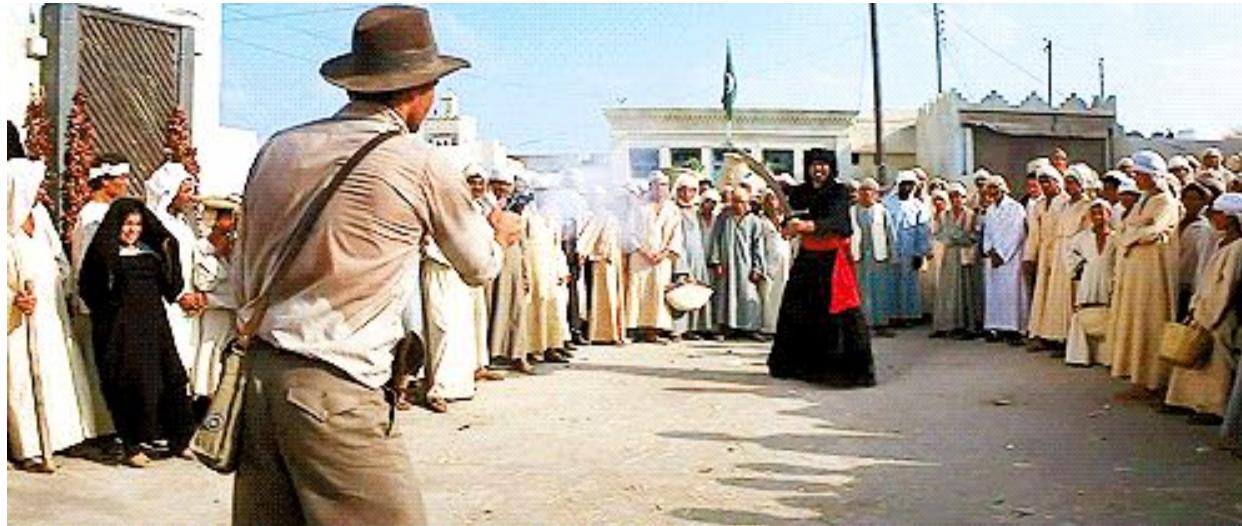
A tool built for another paradigm



No web component support on browsers
No React, Angular or Vue innovations



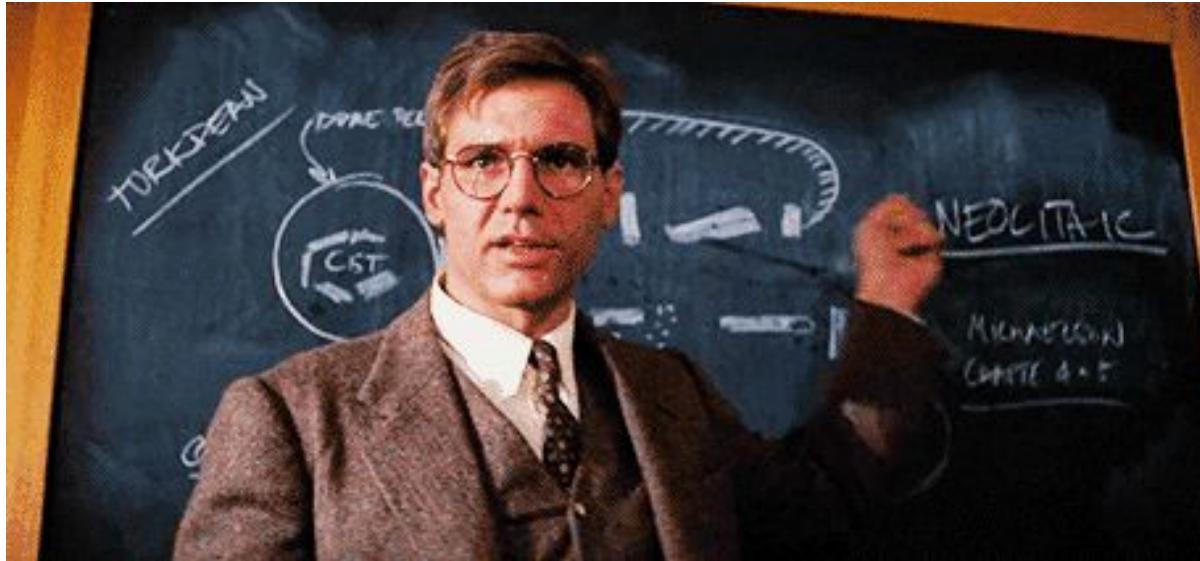
No so well suited for the current one



The current platform is way more powerful
The state of art has evolved



Let's learn from its lessons



The current platform is way more powerful
The state of art has evolved



And let it rest...



There will have no Polymer 4...



So Polymer as we know it is dead...



But the Polymer Project is indeed alive!



But I have invested so much on it!



What to do?



That's why web components are top



You can keep using all your Polymer components and create the new ones with a new library... And it simply works!



And without metaphors?



About the Polymer Project

As front-end engineers in the Chrome team,
our mission is to make the web better.

We work on
libraries & tools

to help developers unlock the web's full potential,
taking advantage of cutting-edge features like Web
Components, Service Workers and HTTP/2.

We experiment with
new patterns

for building faster and smaller web applications.

We advocate for
standards

helping ensure that web developers have a strong
voice in the process.

Polymer Project != Polymer library
Polymer Project well alive
Polymer library was only one library





LitElement

New kid on the block



Born from the Polymer team



For the new web paradigm



Modern lightweight web components



LitElement

A simple base class for creating fast, lightweight web components

→ GET STARTED

About

Fast, lightweight web components

LitElement is a simple base class for creating fast, lightweight web components that work in any web page with any framework.

Using lit-html

For rendering, LitElement uses [lit-html](#)—a fast HTML templating library. To build an app out of LitElement components, check out [PWA Starter Kit](#).

Who are we?

LitElement is brought to you by developers on the Google Chrome team with the input of web developers at organizations big and small around the world.

For the new web paradigm



Based on lit-html



Next-generation HTML Templates in JavaScript

lit-html lets you write HTML templates in JavaScript, then efficiently render and *re-render* those templates together with data to create and update DOM:

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

// A lit-html template uses the `html` template tag:
let 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 differencing!
render(sayHello('Everyone'), document.body);
```

An efficient, expressive, extensible
HTML templating library for JavaScript



Do you know tagged templates?



```
function uppercaseExpression(strings, ...expressionValues) {  
  var finalString = ''  
  for ( let i = 0; i < strings.length; i++ ) {  
    if (i > 0) {  
      finalString += expressionValues[i - 1].toUpperCase()  
    }  
    finalString += strings[i]  
  }  
  return finalString  
}  
  
const expressions = [ 'Tours', 'Touraine Tech', 'Thank you'];  
  
console.log(  
  `uppercaseExpression`  
  ` I am so happy to be in ${expressions[0]} for ${expressions[1]} again!  
  ${expressions[2]}, ${expressions[1]}!`  
);
```

Little known functionality of template literals



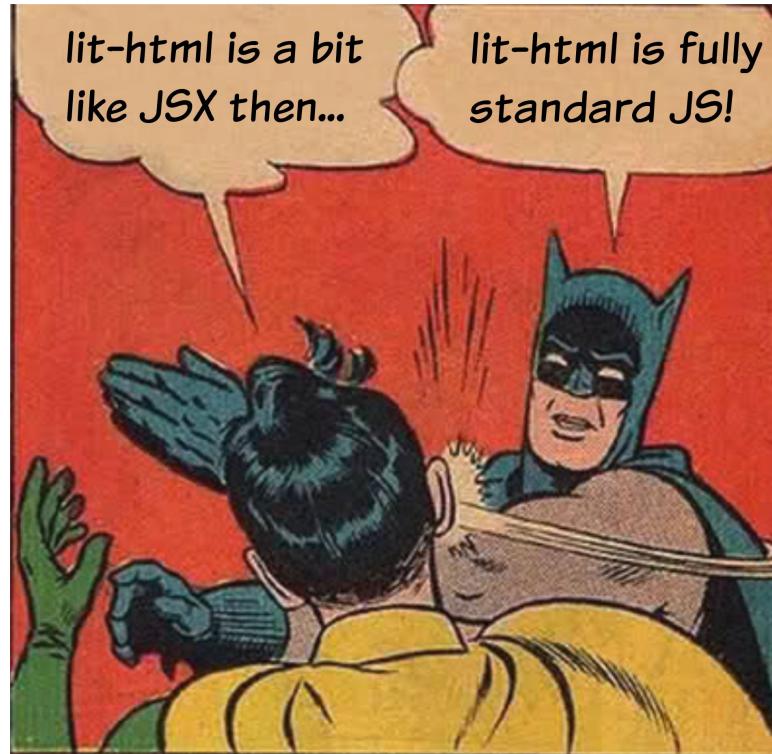
lit-html Templates

```
let myTemplate = (data) => html`  
  <h1>${data.title}</h1>  
  <p>${data.body}</p>  
`;
```

Lazily rendered
Generates a TemplateResult



It's a bit like JSX, isn't it?



The good sides of JSX... but in the standard!



LitElement



```
import { LitElement, html } from 'lit-element';

// Create your custom component
class CustomGreeting extends LitElement {
  // Declare properties
  static get properties() {
    return {
      name: { type: String }
    };
  }
  // Initialize properties
  constructor() {
    super();
    this.name = 'World';
  }
  // Define a template
  render() {
    return html`<p>Hello, ${this.name}!</p>`;
  }
}
// Register the element with the browser
customElements.define('custom-greeting', CustomGreeting);
```

Lightweight web-components using lit-html



One more thing...*



*Let's copy from the master



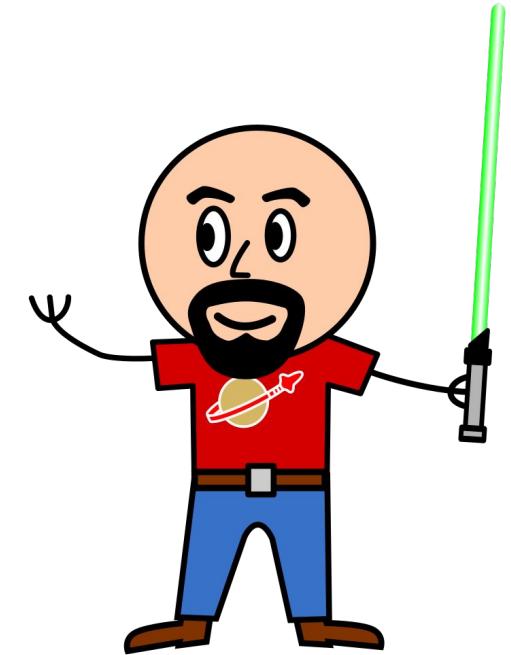
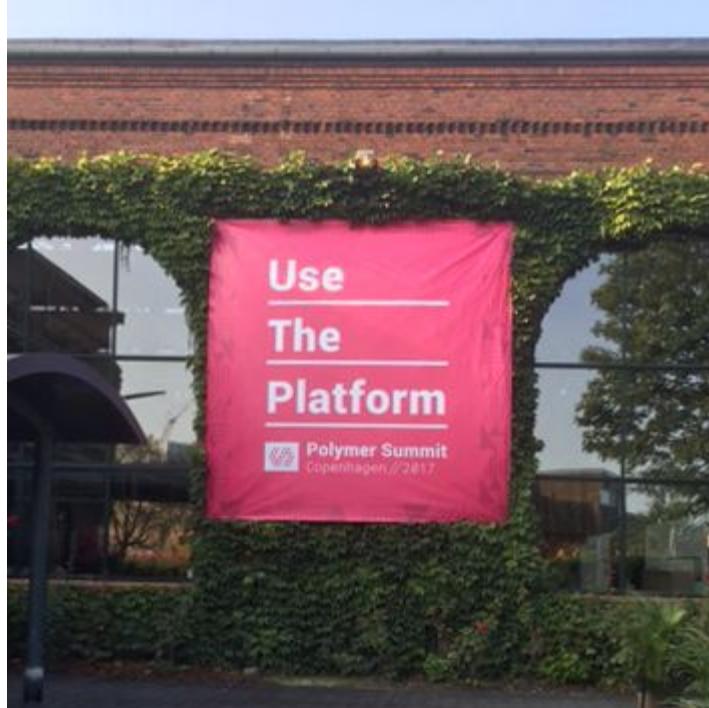
Polymer is not important



WebComponents ARE



Use the Platform, Luke...



WebComponents ARE native



Do you love your framework?



Oh yeah, we all do



Would you marry your framework?



Like until death...



How much does cost the divorce?



Do you remember when you dropped AngularJS for Angular?



Why recode everything again?



Reuse the bricks in your new framework



Lots of web components libraries



hybrids



SkateJS



slim.js



smart
HTML ELEMENTS

LitElement



snuggsi ツ

stencil

For different need and sensibilities



And some good news



Angular Elements



Vue Web Component
Wrapper

Frameworks begin to understand it



So for your next app



Choose a framework, no problem...

But please, help your future self

Use Web Components!



Thank you!



A large, colorful word cloud centered around the words "thank you" in various languages. The word "thank" is in red, "you" is in yellow, and "thank you" together is in red. The background is white with a subtle grid pattern. The word cloud includes many other words related to gratitude and thanks in different languages, such as "merci" (French), "gracias" (Spanish), "mochchakkeram" (Malay), and "mernis" (Swedish). The text is in a variety of fonts and colors, including red, blue, green, yellow, and purple.

