WEB
COMPONENTS
SF

Recorded talk: https://www.youtube.com/watch?v=iFp-P2UJT_Y

Web Components at Enterprise Scale

Why and How Salesforce has embraced Web Components

Diego Ferreiro Val
Principal Architect Salesforce
@diervo

FEEDBACK

LIBERTY CUP

SKATEBOARDING HERITAGE REIMAGINED FOR A NEW GENERATION

SHOP THE COLLECTION →

SHOP ALL SKATEBOARDING →

SKATEBOARDING

SHOES   APPAREL   BUSENITZ SHOES   3MC   ALL SKATEBOARDING

START YOUR FREE TRIAL

LOG IN

We're here to help.

Search for answers

HuluBot

An agent is on the way.

Log in for personalized help

LOG IN   SIGN UP

ACCOUNT & BILLING

USING HULU

WHAT'S ON HULU

Cancel Chat Request

Lightning   Home   Opportunities ▾   Leads ▾   Accounts ▾   Contacts ▾   Campaigns ▾   Contracts ▾   Quotes ▾   Orders ▾   Tasks ▾   Notes ▾   Calendar ▾   Forecasts   Products ▾   Price Books ▾   Files ▾   More ▾

Quarterly Performance

As of Jul 18, 2019 10:45:51 AM   ↻

CLOSED USD 0   OPEN (>70%) USD 1,941,309   GOAL -- ✎

Closed   Goal   Closed + Open (>70%)

News

Big companies commit to slash emissions ahead of U.N. climate...
General Business News
U.S. · 6h

UNITED NATIONS (Reuters) - Almost 90 big companies in sectors from food to cement to telecommunications are pledging to slash their greenhou...

Eyes on U.S. Prize, Primark Considers Central American Suppliers
General Business News
US News & World Report · 1h

US News is a recognized leader in college, grad school, hospital, mutual fund, and car rankings. Track elected officials, research health co...

Tilray Is Down 55% In 2019: What Do The Fundamentals Say?
General Business News
Yahoo · 1h

Central Bankers Hit Speaking Circuit Amid Splits: Economy Week
General Business News
Yahoo · 1h

Emirates Postpones Premi Economy Launch on Boein
General Business News
Msn · 1h

See More News   ‹ ›

Assistant

Opportunity is overdue   ✕
[Home Perf] Marc Schroeder Inc. - 90 Widgets
9/20/2019 · USD 83,014.00 · SFX_ UserRL10

Edit

Opportunity is overdue   ✎ ✕
[Home Perf] Marc Hu Inc. - 179 Widgets

Opportunity is overdue   ✎ ✕
[Home Perf] Elwood Warren Inc. - 398 Widgets

Opportunity is overdue   ✎ ✕
[Home Perf] Maryam Moreira Inc. - 442 Widgets

Opportunity is overdue   ✎ ✕
[Home Perf] Frank Chang Inc. - 202 Widgets

Opportunity is overdue   ✎ ✕
[Home Perf] Matilde Alves Inc. - 413 Widgets

30 days without any activity   ✕
[Home Perf] Garland Patel Inc. - 115 Widgets

30 days without any activity   ✕
[Home Perf] Mark Patel Inc. - 101 Widgets

30 days without any activity   ✕
[Home Perf] Kamala Michael Inc. - 260 Widgets

30 days without any activity   ✕
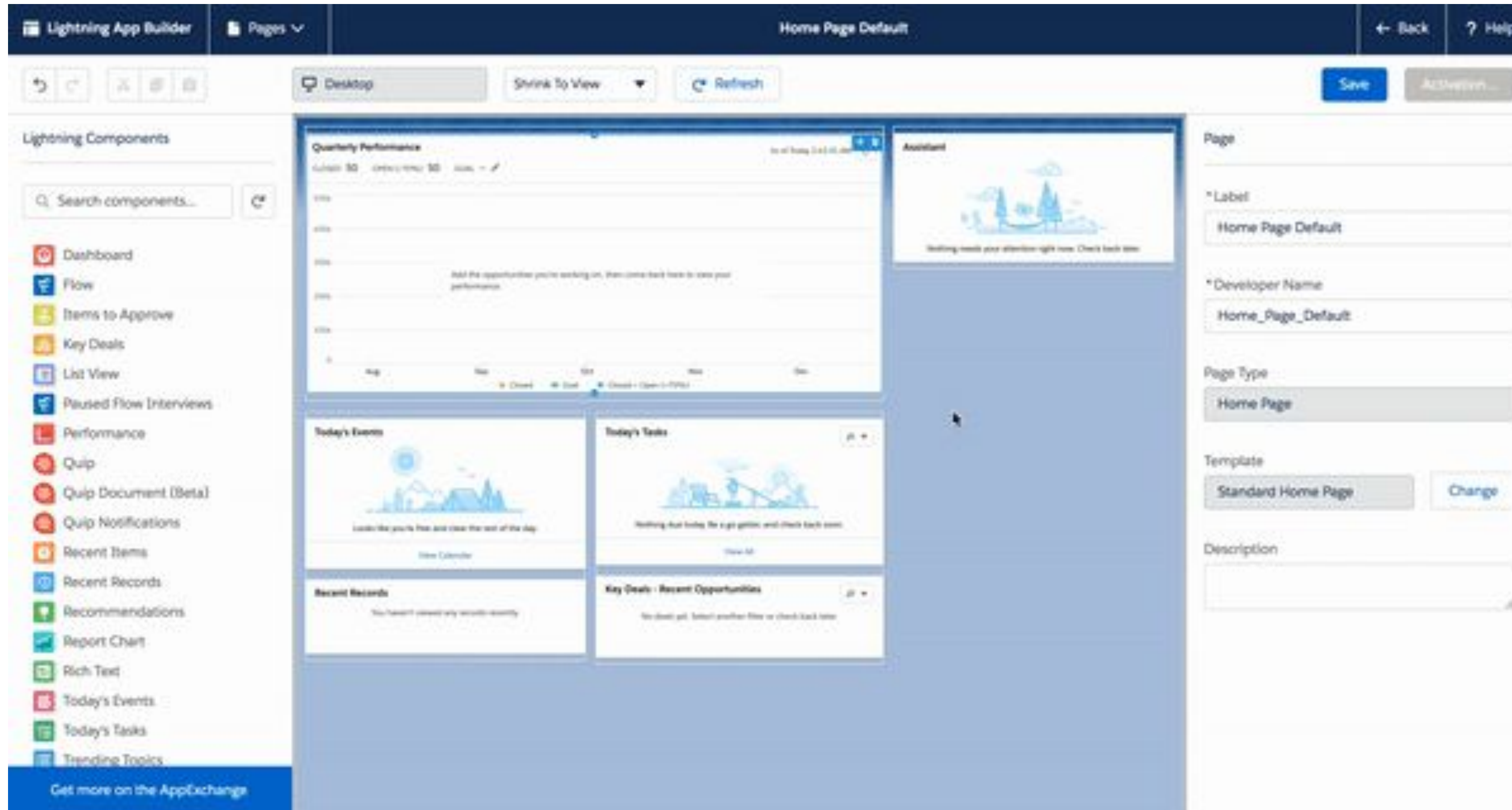[Home Perf] Matilde Thompson Inc. - 93 Widgets
10/6/2019 · USD 165,965.00 · SFX_ UserRL10

New Task   New Event

Today's Events

4:25 AM   sti_#42015-04-28T15:12:34_005x0000000chCTAAY_#5_Contact_ActivityTi...
9/22/2019 5:25 AM

12:55 PM   ffs_#52015-04-28T15:12:34_005x0000000chCTAAY_#0_Lead_ActivityTimeL...
9/22/2019 1:55 PM

Today's Tasks

☐ Call Anil Schroeder   Today
[Home Perf] Anil Schroeder Inc. - 405 Widgets

☐ S1Perf_Deferred2018-01-15T12:27:49-08   Today

IT'S NOT JUST AN APP, IT'S A PLATFORM!

PLATFORM BUSINESS REQUIREMENTS

What makes it really hard?

- **Multi-author, Multi-version**

- **Backwards compatibility** *indefinitely*

· Accessible

· Personalizable

· Localizable
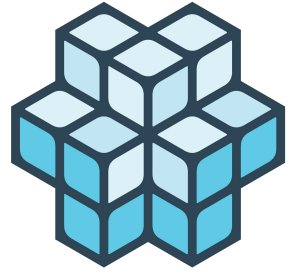
· Secure

· Performant (including old browsers)

DEVELOPER REQUIREMENTS

What do our customers want?

- Less proprietary technology and more standards

- Common component model (interoperability)

- Transferable skills / easy-to-ramp-up developers

- Standard tooling

- Not being stuck with old technology (safe evolution)

- Not being stuck inside a walled garden (portability)

IF WE WERE TO START OVER, WHAT WOULD WE DO?

MONTHS OF RESEARCH

Learning from the best in the industry

- Frameworks

  React, Preact, Relay, Angular, Elm, Vue,

  Inferno, Ember, Aurelia, Polymer, Cycle.js

- Data flow

  Flux, Redux, GraphQL, MobX, RxJS,

  Falcor, ImmutableJS, Backbone.js

- Virtual DOM & reactivity model

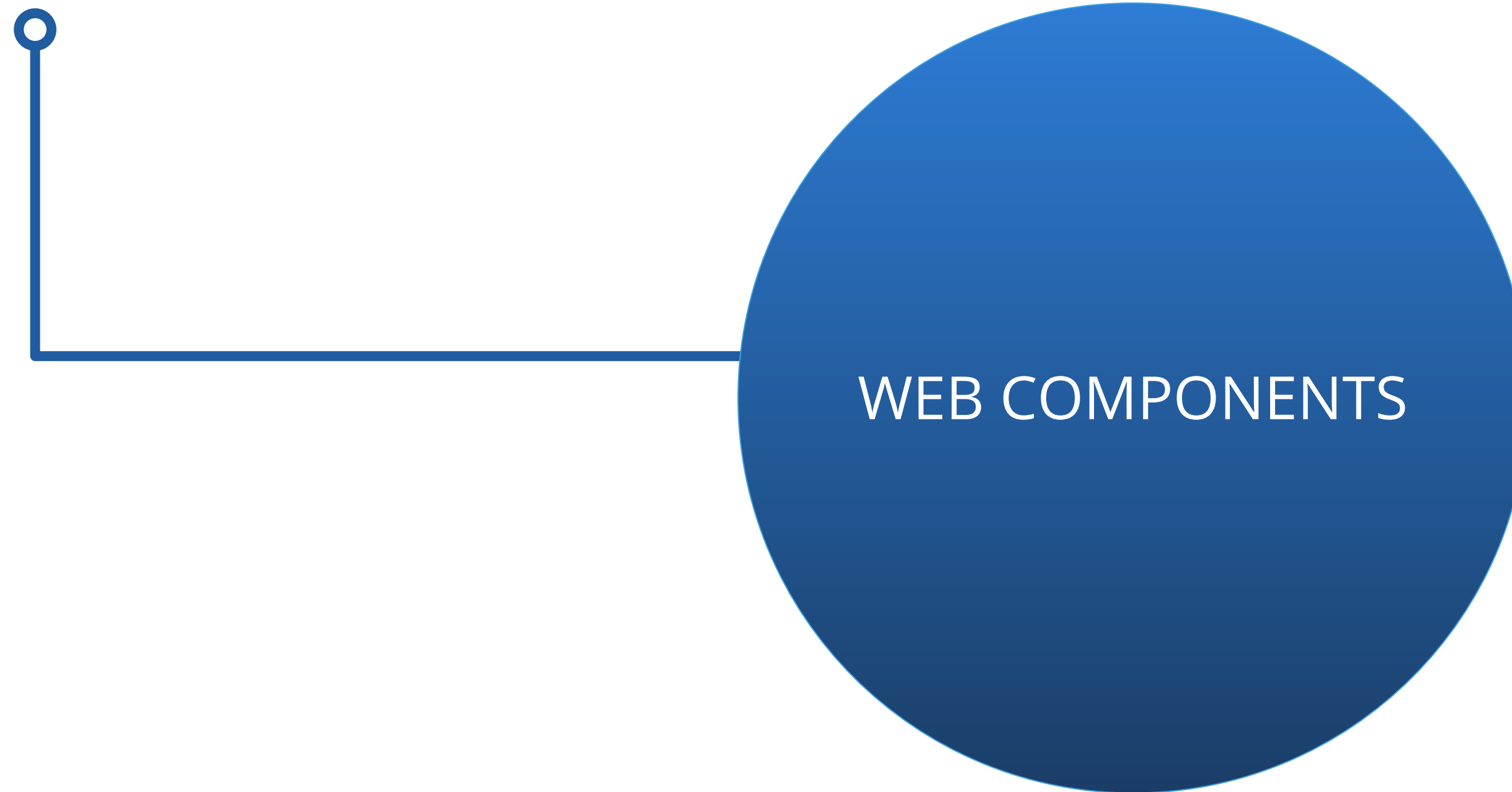 SnabbDom, ViDom, virtual-dom, vDom

  MobX, Svelte, Glimmer

The web platform has managed to evolve without breaking changes.

The best way to future proof our technology is to align with standards.

- interoperability
- encapsulation
- future proof
- backwards compatibility

WEB COMPONENTS

```
class HelloWorld extends HTMLElement {
    constructor() {
      super();

      const t = document.getElementById('my-template');
      this.attachShadow({ mode: 'open' }).appendChild(t.cloneNode(true));
      …
    }

    connectedCallback() { … }
    attributeChangeCallback(value) { … }
    disconectedCallback() { … }
  }
);

document.customElements.define('hello-world', HelloWorld);
```
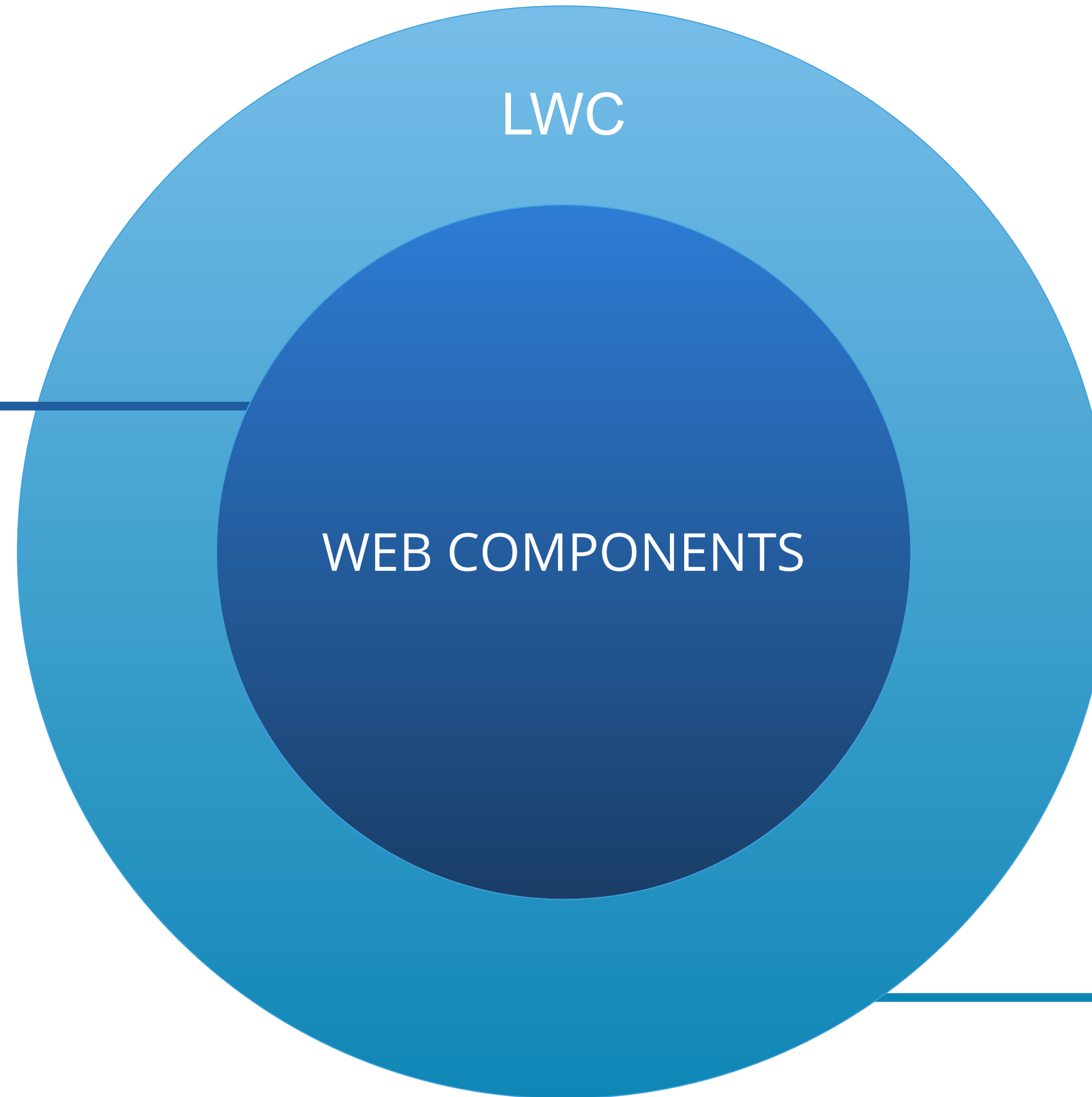
- interoperability
- encapsulation
- future proof
- backwards compatibility

LWC

WEB COMPONENTS

- better ergonomics
- accessibility (fill gaps)
- performance (fill gaps)
- browser compatibility

WEB COMPONENTS SF

```
<template>
  <p>Counter: {count}</p>
  <button onclick={increaseCounter}>Add</button>
</template>
```
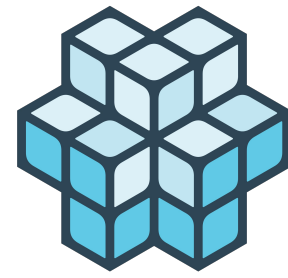
```
import { LightningElement, track } from "lwc";

export default class MyCounter extends LightningElement {
  @track count = 0;

  increaseCounter() {
    this.count += 1;
  }
}
```
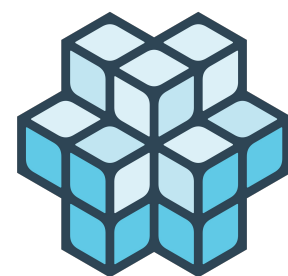
CHALLENGES ADOPTING WEB COMPONENTS

Not everything was beautiful...

CHALLENGES ADOPTING WEB COMPONENTS

The two main offenders

- Shadow DOM encapsulation
    - Testing
    - Styling & theming

- Compatibility mode (IE11) 🙍

SUPPORTING WEB COMPONENTS IN OLD BROWSERS

Down the IE11 rabbit hole

SUPPORTING WEB COMPONENTS IN OLD BROWSERS

Enterprise software requires it

Javascript: ES7 down to ES5

- Polyfills (a lot of bugs were found)

- Babel transpilation (ad-hoc plugins for perf)

- Proxy support (first in industry!)

```javascript
var __callKey2 = Proxy.callKey2;
var __setKey = Proxy.setKey;
var __getKey = Proxy.getKey;

var MyCounter =
/*#__PURE__*/
function (_LightningElement) {
  _inherits(MyCounter, _LightningElement);

  function MyCounter() {
    _classCallCheck(this, MyCounter);

    return _possibleConstructorReturn(this, __callKey2(_getPrototypeOf(MyCounter), "apply", this,
arguments));
  }

  _createClass(MyCounter, [{
    key: "handleClick",
    value: function handleClick() {
      __setKey(this, "counter", __getKey(this, "counter") + 1);
    }
  }]);

  return MyCounter;
}(LightningElement);
```

SUPPORTING WEB COMPONENTS IN OLD BROWSERS

Enterprise software requires it

Javascript: ES7 down to ES5

- Polyfills (a lot of bugs were found)

- Babel transpilation (ad-hoc plugins for perf)

- Proxy support (first in industry!)

Shadow DOM (synthetic shadow)

- CSS encapsulation (compile time)

- CSS variables (compile time)

- Event retargeting, focus, tabs...

```css
:host {
    color: var(--fancy-color);
    padding: 10px;
}
```

```javascript
function stylesheet(hostSelector, shadowSelector, varResolver) {

  return [

      hostSelector,

      " { color: " + varResolver("--fancy-color"),

      ";padding: 10px; }"

  ].join('');
}
```

SUPPORTING WEB COMPONENTS IN OLD BROWSERS

Enterprise software requires it

Javascript: ES7 down to ES5

- Polyfills (a lot of bugs were found)

- Babel transpilation (ad-hoc plugins for perf)

- Proxy support (first in industry!)

Shadow DOM (synthetic shadow)

- CSS encapsulation (compile time)

- CSS variables (compile time)

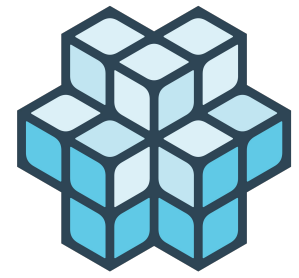- Event retargeting

- Focus & Tabs

Bugs on spec, browsers and tooling

- Selenium (maintainers of IE11 driver)

- JSDOM (implemented WC missing capabilities)

- Issues in other browsers too (not only IE11)

SUPPORTING WEB COMPONENTS IN OLD BROWSERS

SHADOW DOM ENCAPSULATION

Testing and Styling: Easier said than done

SHADOW DOM ENCAPSULATION

Changing the way we do Testing and Styling

```html
<div class="context">
    <h2>My Context</h2>          ← matches
    <lightning-foo>
        #shadow-root
            <div class="my-component">
                <h2>My Component</h2>     ← doesn't match
            </div>
    </lightning-foo>
</div>
```

```css
h2 { ... }
```

```css
h2 { ... }
.context h2 { ... }
.my-component h2 { ... }
```

```js
document.querySelector(".my-component"); // returns null
```

SHADOW DOM ENCAPSULATION

State of the art for testing before Shadow DOM

```
/html/body/div[1]/section/div/div[2]/div/form/div[2]/input[3]
```

```
document.querySelector("x-container").shadowRoot.querySelector("x-item")
```

SHADOW DOM ENCAPSULATION

Rewriting thousands of tests to be Shadow DOM compliant was painful...

```json
{
    "root": "one-app-nav-bar", //root or default locator
    "isShadowHost": true,
    "elements": [
    {
    "name":"item", //relative to root, expands shadow root
    "css":"one-app-nav-bar-item-root",
    "isShadowHost":true,
    "hasIndex":true,
     …
}
```

```java
public WebElement expandShadowIfPresent(WebElement element) {
    String checkShadowRoot =
            "var shadowRoot = arguments[0].shadowRoot;"
            + "if (shadowRoot && shadowRoot.nodeType === 11 && !!shadowRoot.host) {"
            + "    return true;"
            + "}"
            + "return false;";

    Boolean shadowRoot = (Boolean)((JavascriptExecutor)getCurrentWebDriver())
        .executeScript(checkShadowRoot, element);
    boolean hasShadowRoot = shadowRoot.booleanValue();

    return hasShadowRoot ? new ShadowRootWebElement(element) : element;
}
```
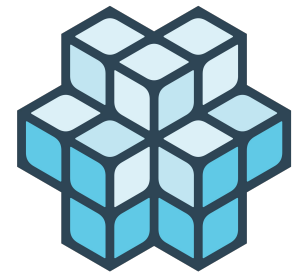
SHADOW DOM ENCAPSULATION

The good parts: Clean boundaries helped us scale

Our **components** and **tests** are more reliable, resilient and scalable.

- We have already fixed Selenium for you :)

- A lot of testing frameworks already implement some shadow selector abstraction (ex. webdriverio)

- Selenium WebDriver soon to have a [shadow piercing primitive](#).

SHADOW DOM ENCAPSULATION

Customizing styles with Shadow boundaries

SHADOW DOM ENCAPSULATION

Before the "shadow", there was nothing preventing evil…

```css
.my_container .lightning-button .icon > div > span {
  border-radius: 5px;
}
```

SHADOW DOM ENCAPSULATION

Changing the design system to comply with Shadow DOM semantics

Instead of having a monolithic Design System with a lot of cascading rules:

```css
.slds-box .slds-button { margin-left: .25rem; }
```

Small blueprints that contain decoupled OOM like customizable rules.

```css
.lwc-button[disabled], .lwc-button:disabled {
  background-color:var(--disabled-button-color-background, transparent);
  color:var(--disabled-button-color, #dddbda);
  cursor:default;
}
```
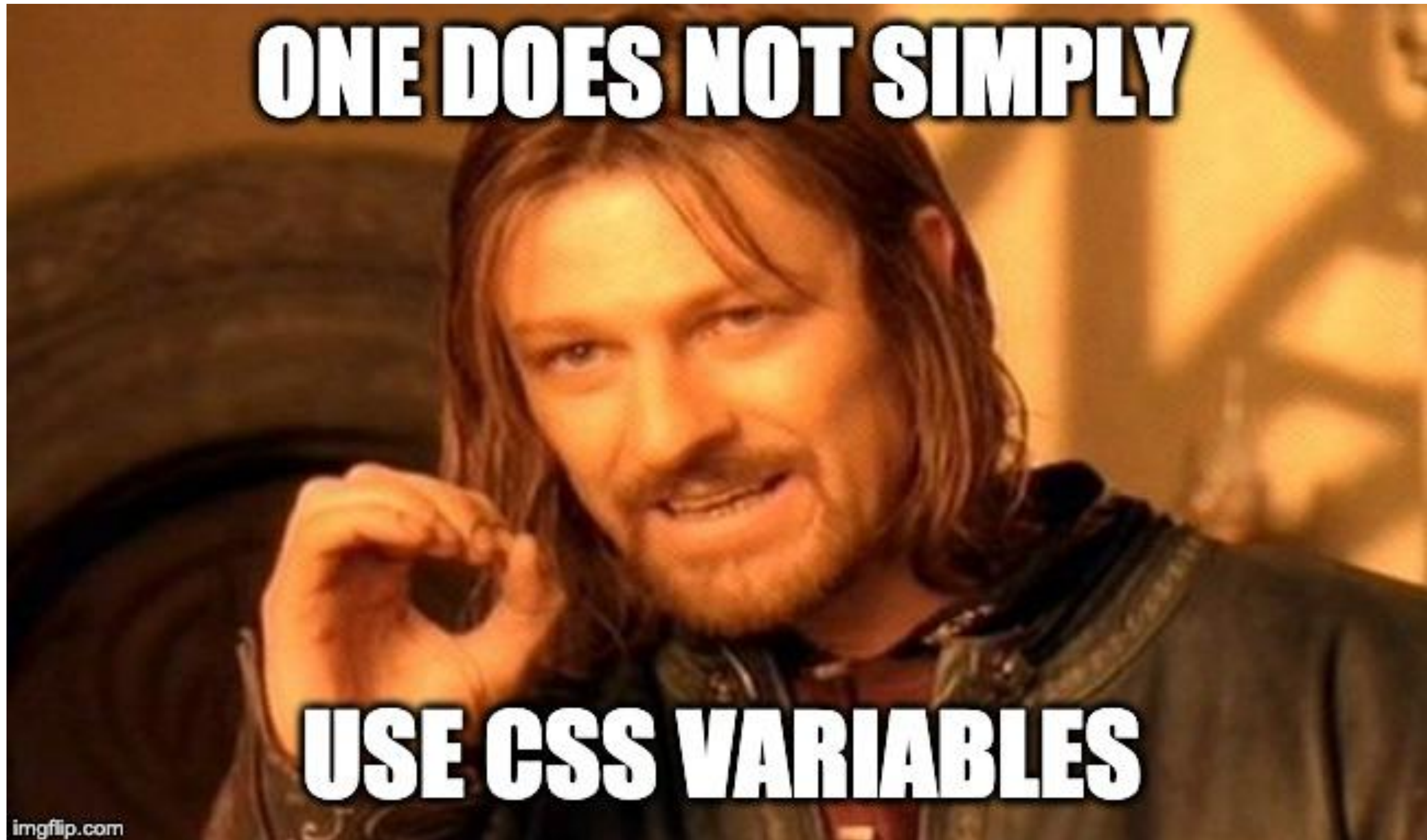
SHADOW DOM ENCAPSULATION - STYLING

How to build fully customizable components?

- We believe custom properties + ::part + :state() will solve 90% of the use cases.

- For pixel perfect customization we are exploring Constructible Stylesheets where you can take over small component blueprints in a controlled way.

Our **styles** have become modular, predictable and API driven (no side-effects).
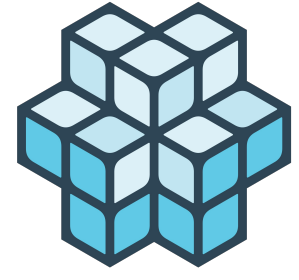
ADOPTION TAKEAWAYS

ADOPTION TAKEAWAYS

- Although the path to production was hard, Web Component is helping us scale.

- Building on top of standards is making our platform future proof.

- We get interoperability between different applications for "free".

- As the platform matures we can leverage more native APIs which turns into better performance.

- Our developers can be productive from day one.

LWC BY THE NUMBERS (as of October 2019)

1M components created

73% of devs use LWC

95% of devs feel that WC is the right direction

🔄 Retwitteaste

Web Components .dev
@webcomp_dev

🚨New release!🚨

- Support for #LightningWebComponent (LWC) by @SalesforceEng
- Improved support for @vuejs ✌️
- Improved support for @angular
- a few surprises 😉

You can try right now on webcomponents.dev

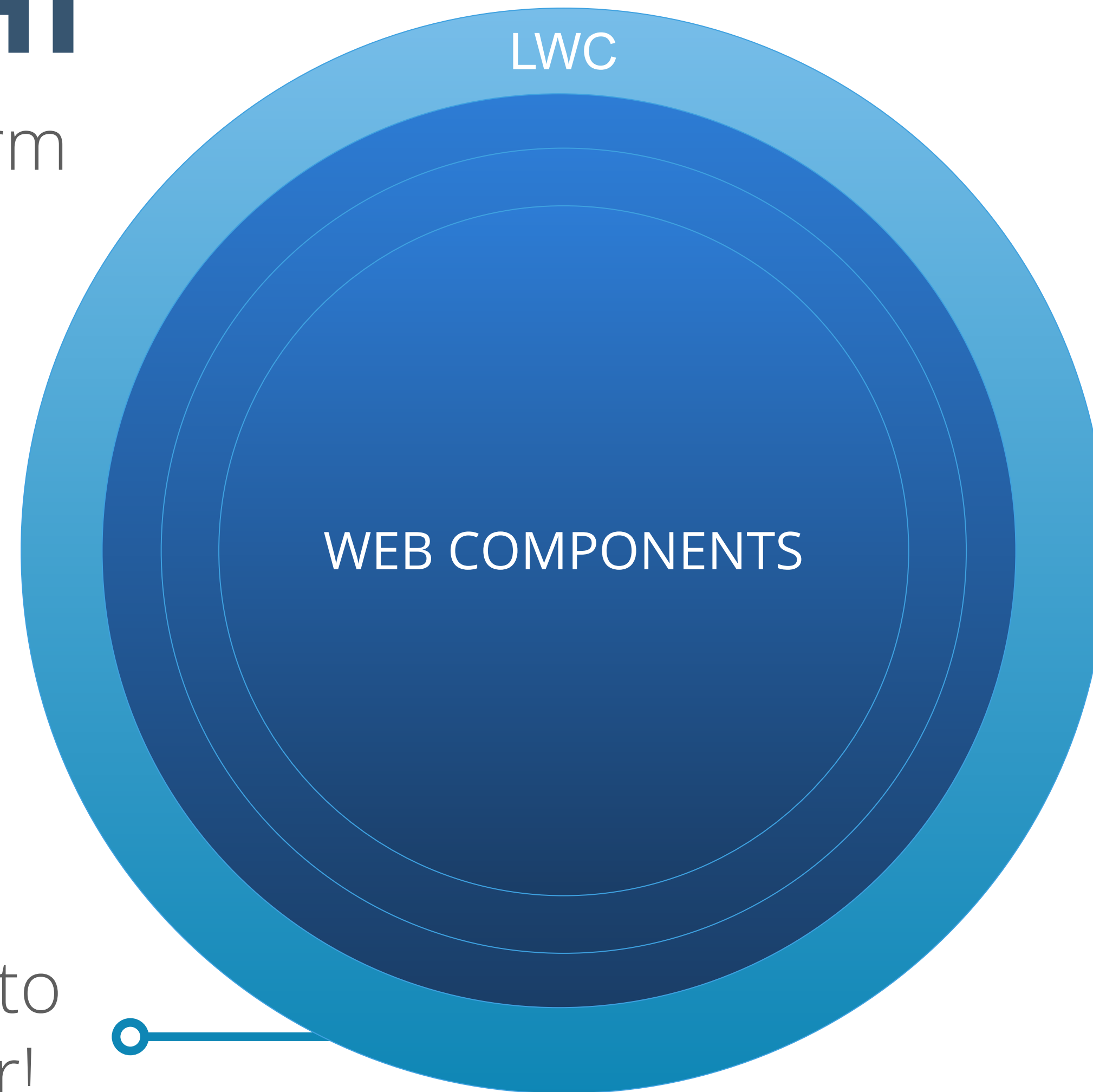#webcomponents

WEB COMPONENTS SF

THE FUTURE IS BRIGHT

Let us keep pushing the platform and specifications forward!

We will keep representing the enterprise use cases in W3C and TC39

Our goal is for LWC to eventually disappear!

LWC

WEB COMPONENTS

If the enterprise can deploy web components so can you!

Docs: **https://lwc.dev**

Git: **https://github.com/salesforce/lwc**

RFCs: **https://github.com/salesforce/lwc-rfcs**

WEB COMPONENTS SF

Thank You!

Diego Ferreiro Val
@diervo

WEB
COMPONENTS
SF