



Web Components & Angular Elements

Alex Thalhammer

Long Time  
Maintain-  
ability

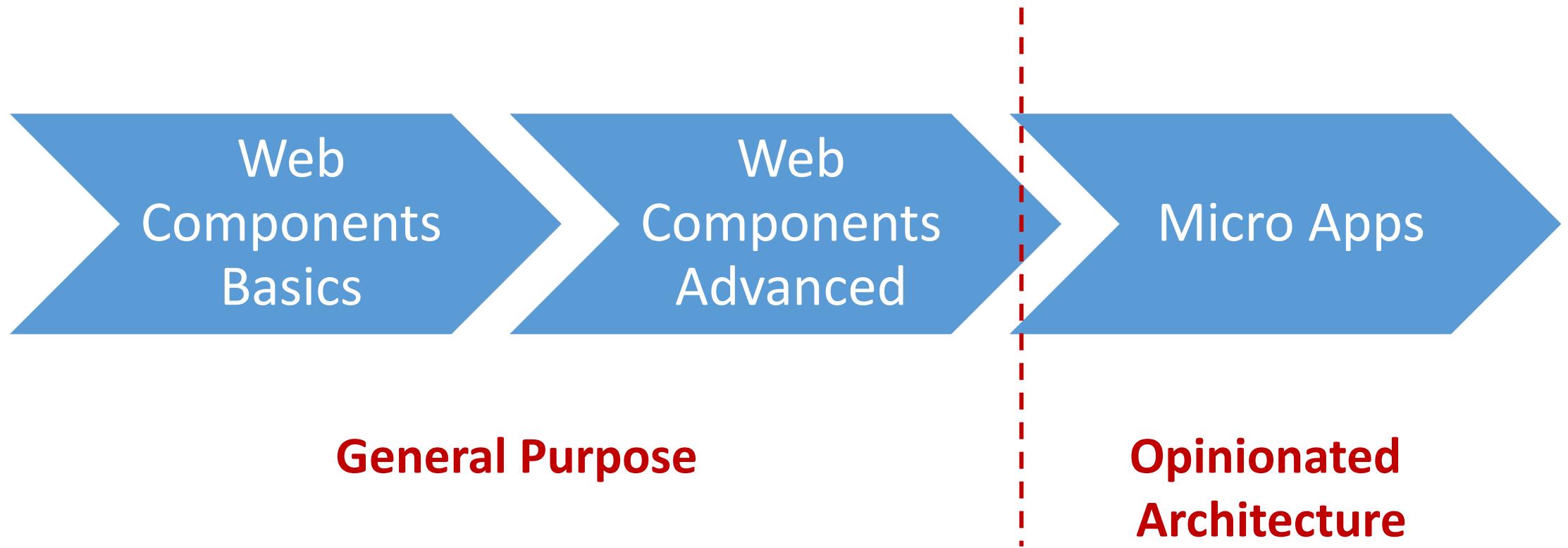


Maintain + Extend for 10 +/- yrs

# Contents

- Web Components
- Using Web Components in Angular
- Web Components with Angular Elements
- Building Web Components
- Micro Apps

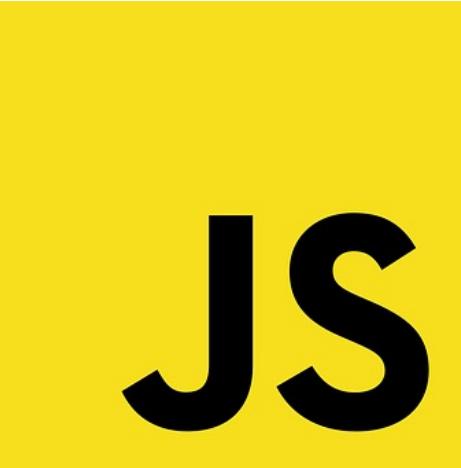
# From Web Components to Micro Apps





# Web Components

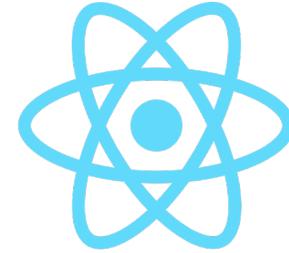
---



JS



V

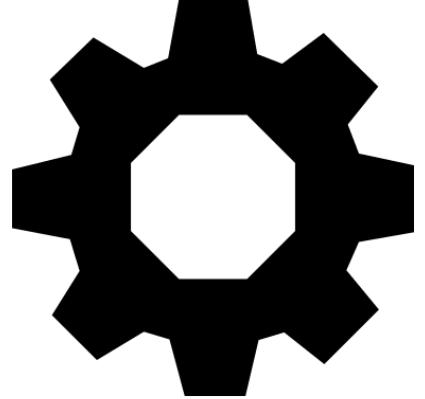


Framework independent  
Components

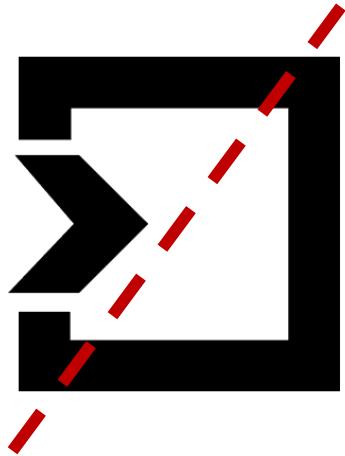




Umbrella Term  
for several  
Standards



Templates



HTML Imports



Custom Elements



Shadow DOM

Standards

# Templates

---

Markup that is rendered but not displayed

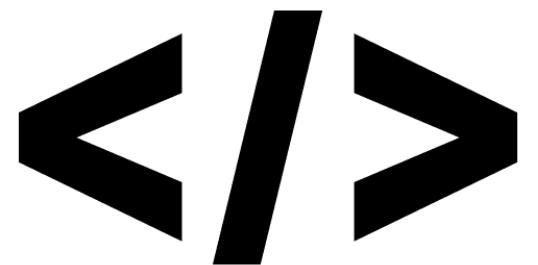
---

Can be displayed when needed, e. g. within components

---

```
<template><span class="number"></span> Records found</template>
```

---



# Custom Elements

---

Custom HTML-Element, e. g. <date-picker> (dash!)

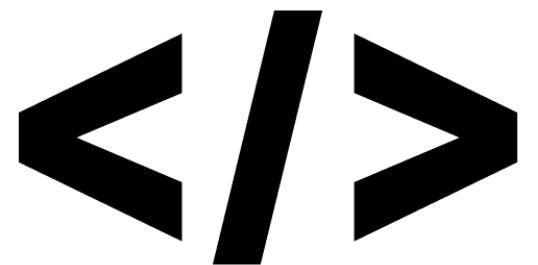
---

Behaves like build-in ones

---

Write once; consume with any framework

---



# Examples

```
<sli-der min="0" max="100"> ...
```

```
<customer-selector selected="selected()"> ...
```

```
<my-micro-app state="sharedState" message="handle(...)">...
```



# Possibilities

Design  
Systems

Enriching  
existing Apps

Migrating  
Legacy Apps

UI Integration/  
Composition

Micro Apps

# Shadow DOM

---

Isolate CSS

---

Parent cannot influence Component's layout

---

Component cannot influence Parent or Siblings

---



All this can be  
polyfilled  
down to IE11  
(more or less)



[Nonagons Please Me](#)[npm Enterprise](#) [Features](#) [Pricing](#) [Docs](#) [Support](#) Search packages[Search](#)[log in or sign up](#)

Unleash awesomeness. Private packages, team management tools, and powerful integrations. [Get started with npm Orgs »](#)

## @webcomponents/webcomponentsjs

2.2.3 • [Public](#) • Published a day ago

[Readme](#)[0 Dependencies](#)[404 Dependents](#)[49 Versions](#)[build](#) passing[install](#)

# webcomponents.js (v1 spec polyfills) Polyfill

**Note.** For polyfills that work with the older Custom Elements and [Shadow DOM v0](#) specs, see the [v0 branch](#).

[weekly downloads](#)

90.189



# Selected Polyfills

- `webcomponents-loader.js`
  - Loads Polyfills on demand
  - Custom Elements
  - Shadow DOM
  - Templates
- `native-shim.js`
  - Needed for browsers that DO support polyfills
  - When downlevel to ES5

# API

```
class HelloElement extends HTMLElement {  
    [...]  
}
```

# API

```
class HelloElement extends HTMLElement {  
  
    constructor() {  
        super();  
        this.attachShadow({mode: 'open'});  
  
        this.shadowRoot.innerHTML = '<div>Hello World</div>';  
    }  
  
    [...]  
}
```

# API

```
class HelloElement extends HTMLElement {  
  
    constructor() {  
        super();  
        this.attachShadow({mode: 'open'});  
  
        this.shadowRoot.innerHTML = '<div>Hello World</div>';  
    }  
  
    [...]  
}  
  
customElements.define('hello-world', HelloElement);
```

# Calling a Web Component

```
<hello-world></hello-world>
```

# Using Properties

```
class HelloElement extends HTMLElement {  
  
    get name() { [...] }  
    set name(value: string) { ... }  
  
    constructor() {  
        [...]  
    }  
  
    [...]  
}
```

# Using Properties

```
<!-- VanillaJS -->
var elm = document.getElementById(...)
elm['name'] = max;
```

```
<!-- Property Binding with Angular -->
<hello-world [name]="firstName"></hello-world>
<hello-world [name]="'Max'"></hello-world>
```

# Using Attributes

```
<hello-world name="Max"></hello-world>
```

**Not a property!!**

# Component Callbacks & Co.

- attributeChangedCallback(name, oldValue, newValue)
  - static get observedAttributes
- connectedCallback
- disconnectedCallback
- adoptedCallback



# Angular Elements

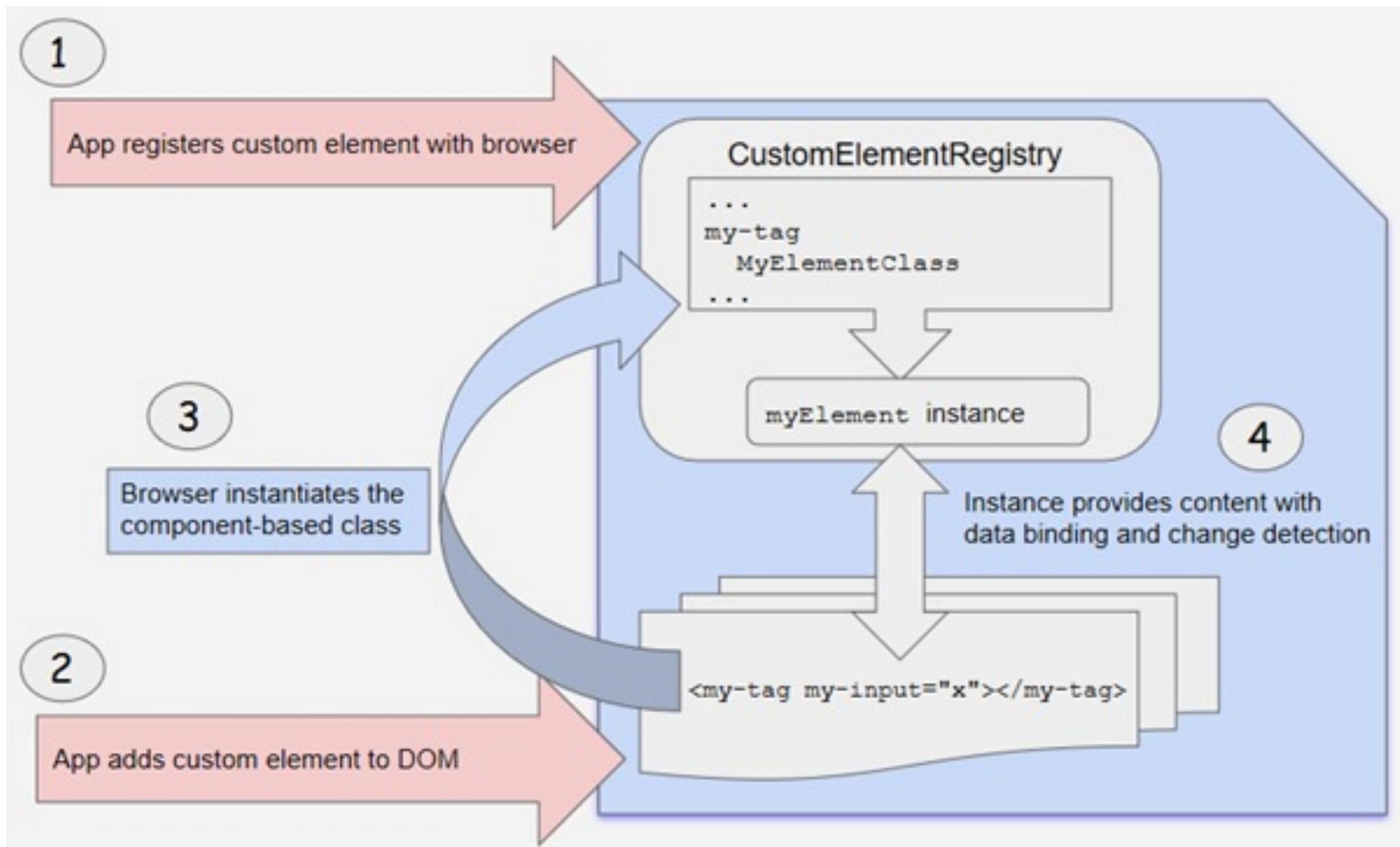
# Angular Elements

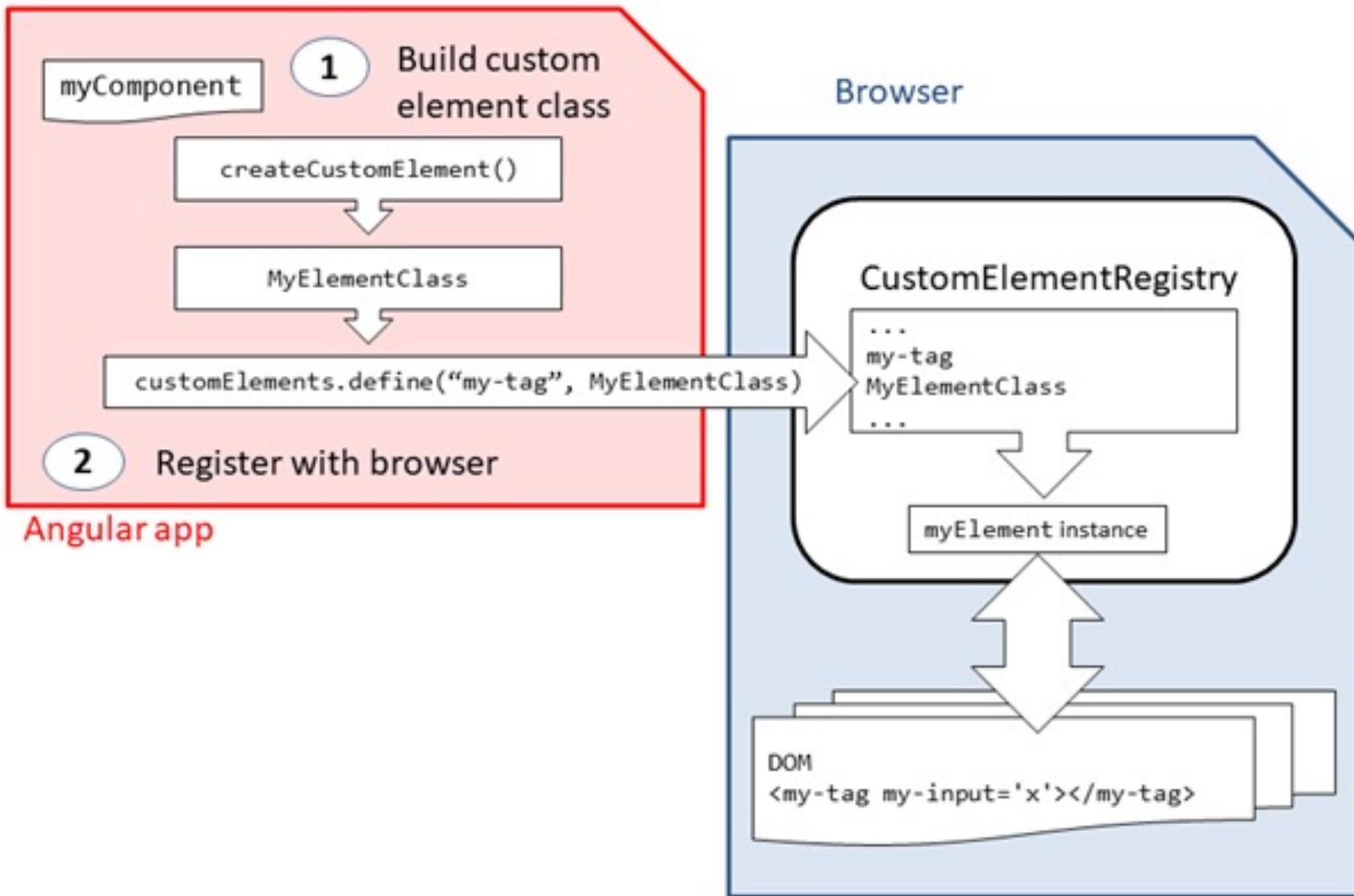


Wrap Angular Components



Expose Custom Elements





# Angular Elements

```
@NgModule({
  imports: [BrowserModule],
  declarations: [DateComponent],
  entryComponents: [DateComponent]
})
export class AppModule { }
```

# Angular Elements

```
@NgModule({
  imports: [BrowserModule],
  declarations: [DateComponent],
  entryComponents: [DateComponent]
})
export class AppModule { }
```

# Angular Elements

```
@NgModule({
  imports: [BrowserModule],
  declarations: [DateComponent],
  entryComponents: [DateComponent]
})
export class AppModule {

  constructor(private injector: Injector) { }

}
```

# Angular Elements

```
@NgModule({
  imports: [BrowserModule],
  declarations: [DateComponent],
  bootstrap: [],
  entryComponents: [DateComponent]
})
export class AppModule {

  constructor(private injector: Injector) {

    const DateElement = createCustomElement(
      DateComponent, { injector: this.injector });

    customElements.define('date-picker', DateElement);

  }
}
```

# Angular (Elements) & Shadow DOM

- Baked-in
- Default: Emulation
  - `@Component({ encapsulation: ViewEncapsulation.Emulated })`
- Shadow DOM (v0)
  - `@Component({ encapsulation: ViewEncapsulation.Native })`
- Shadow DOM (v1)
  - `@Component({ encapsulation: ViewEncapsulation.ShadowDom })`
- Can be turned off:
  - `@Component({ encapsulation: ViewEncapsulation.None })`



Lab



Micro Apps

---

# Idea

Maintainability?

Framework

System



Team A

Team B



Architecture

Hard to change!

# Idea

μService

μService

μService

# Idea

μApp

μApp

μApp

## Pro

## Contra

Less Complexity

Less Coordination

Different Technologies

Separate Deployment

UI Composition

Distributed Data

Distributed System

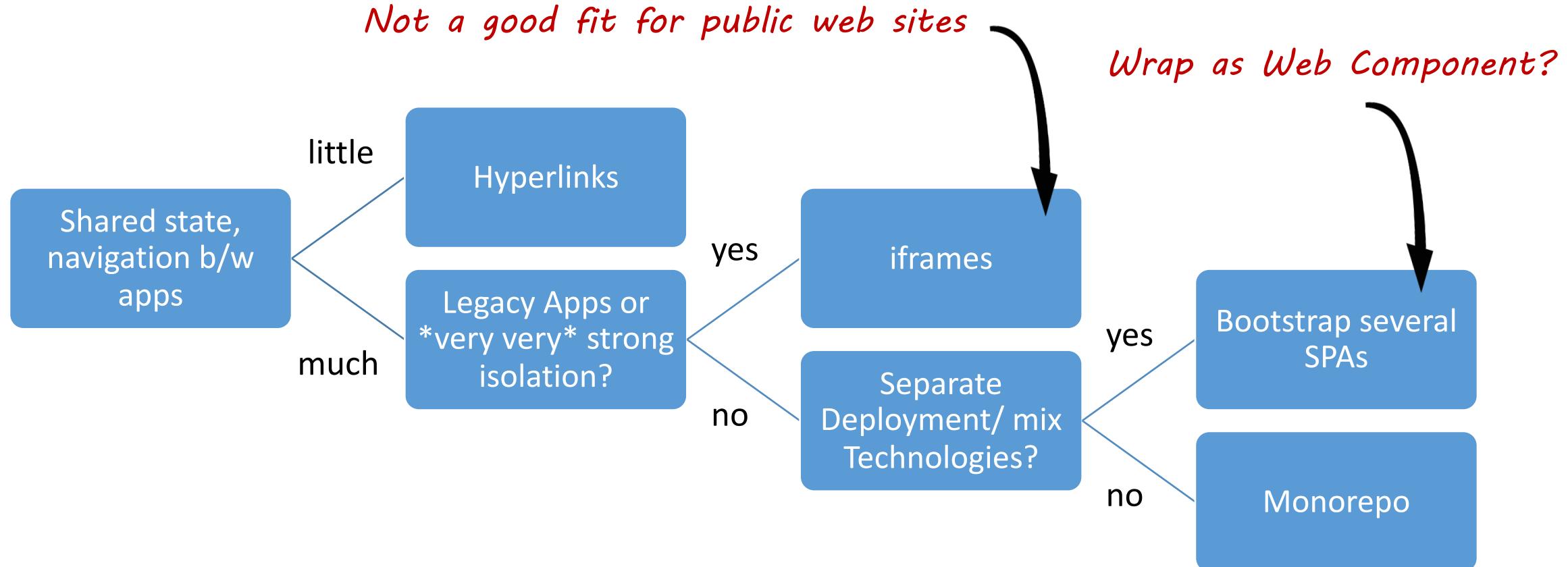
# Communication b/w Web Components

"Service Bus" in Shell



```
<client-a [state]="appState" (message)="handle(event)"></client-a>
<client-b [state]="appState" (message)="handle(event)"></client-b>
```

# Some General Advice



# Shared Widgets

Packages  
(e. g. w/ Web Components)

Lazy Loaded  
Web Components

Inner  
Sourcing

Separate  
Ways

...

# Summary

Polyfills and  
CUSTOM  
ELEMENTS\_SCHEMA

Directives for  
Two-Way-Binding

Angular Elements

Lazy & External  
Web Components

Micro Apps

# That's it for Web Components

- Any questions left over? 😊