



*Web
Components
@ the Edge*

Owen Buckley

**OpenJS World //
2022**



Hello There 🖐️



Open source and web enthusiast. Currently a UI Architect at [LendingClub](#).

- GitHub: [@thescientist13](#)
- Twitter: [@thegreenhouseio](#)
- Website: [www.thegreenhouse.io](#)

Agenda



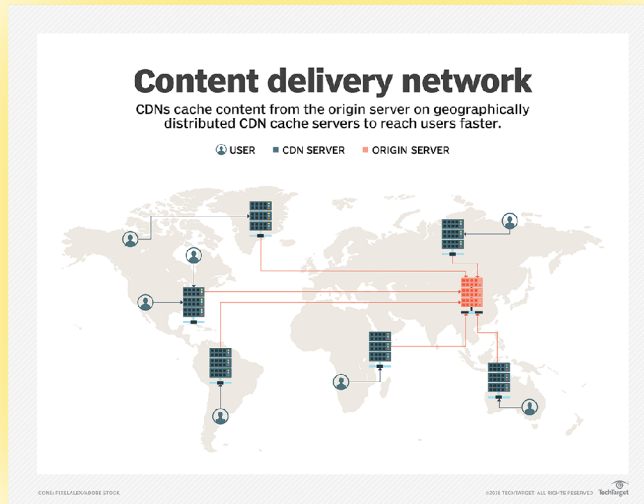
1. What is The Edge (and Serverless)?
2. Web Components (and SSR)
3. Demo
4. Considerations

5. Community

What is The Edge?



What is The Edge?



TechTarget

What is The Edge?

CDNs


- CloudFront
 - Fastly
 - Cloudflare
 - Akamai
-

```
// serverless function
export async function handler (
  const data = await fetch('http://localhost:3000')
    .then(resp => resp.json());

  return {
    statusCode: 200,
    headers: {
      'content-type': 'text/html'
    },
    body: '<ol>
```

What is The Edge?

Benefits

- Fast
- Dynamic
- On Demand
- Cost Effective 
- Cacheable

What is The Edge?

Edge Functions

- *Closer to the user*
- Limited API surface



What is The Edge?


- CDN is great for static content
- Serverless is great for on-demand compute
- Edge is great for geographic compute
- *Use them all together!*

Web Components (and SSR)



Web Components (and SSR)

Web Components

- HTML Templates
(`<template>`)
- Custom Elements
(`customElements.def`)
- Shadow DOM 
(`shadowRoot`)

```
const template = document.creat

template.innerHTML = `
  <style> footer { ... } </styl
  <footer>
    <a href="/">
      My Blog &copy; ${new Date(
    </a>
  </footer>
`;
```



```
$ node ./footer.js  
(node:44582) UnhandledPromiseRejectionWarning: ReferenceError: document is
```





```
import { renderToString } from 'react-dom/server'
import App from './App.jsx'

const html = ReactDOMServer.renderToString()

console.debug({ html });
```



```
$ node ./footer.js  
✨ Done in 1.75s.
```



Web Components (and SSR)

Use the spec, Luke

- customElements.define
- HTMLElement
- ShadowRoot
- DocumentFragment



```
import { renderToString } from 'react-dom/server'
import App from './App.jsx'

const html = ReactDOMServer.renderToString()

console.debug({ html });
```

Declarative Shadow DOM

```
<host-element>  
  <template shadowroot="open">  
    <slot></slot>  
  </template>  
  <h2>Light content</h2>  
</host-element>
```

Declarative Shadow DOM

```
<wc-card>
  <template shadowroot="open">
    <style>
      [name="title"] {
        color: green;
      }
      ::slotted(img) {
```

Web Components (and SSR)

So, to recap we can...

- ✓ Shim Node.js
- ✓ Pre-bake our HTML using web standards (no client-side JS!)
- ✓ Apply techniques like progressive

Web Components (and SSR)

so.... what are we waiting for?



Demo



Demo (Serverless Functions)

wv-at-uec-edge.thegreenhouse.io/demo



Demo (Edge Functions)

magnificent-caramel-f19440.netlify.app/demo



Considerations



Considerations

- Browser Support - Chrome / Edge
- Tiny client side polyfill
- Closer you get to the edge, the "smaller" the runtime surface area
- Somewhat manual, but community tools exist!

Community.



Community

Solutions

- [Lit SSR](#)
- [Stencil](#)
- [uhtml-ssr](#)
- [WCC \(Web Components Compiler\)](#)

Community

Communities

- [Lit \(Slack\)](#)
- [Web Components Community Group \(GitHub\)](#)
- [Project Evergreen \(GitHub\)](#)

Thank You!

