



Rendering, Hydration, and  
what the heck is Qwik?

# \$ whoami

- Ich bin Inder ;) 🤪
- Developer Advocate @ Dynatrace
- Musician
- Ionic Developer Expert | Auth0 Ambassador
- Blog at <https://inders.in/blog>



[@imsinghk7](https://twitter.com/imsinghk7)





# Rendering, Hydration, and what the heck is Qwik?

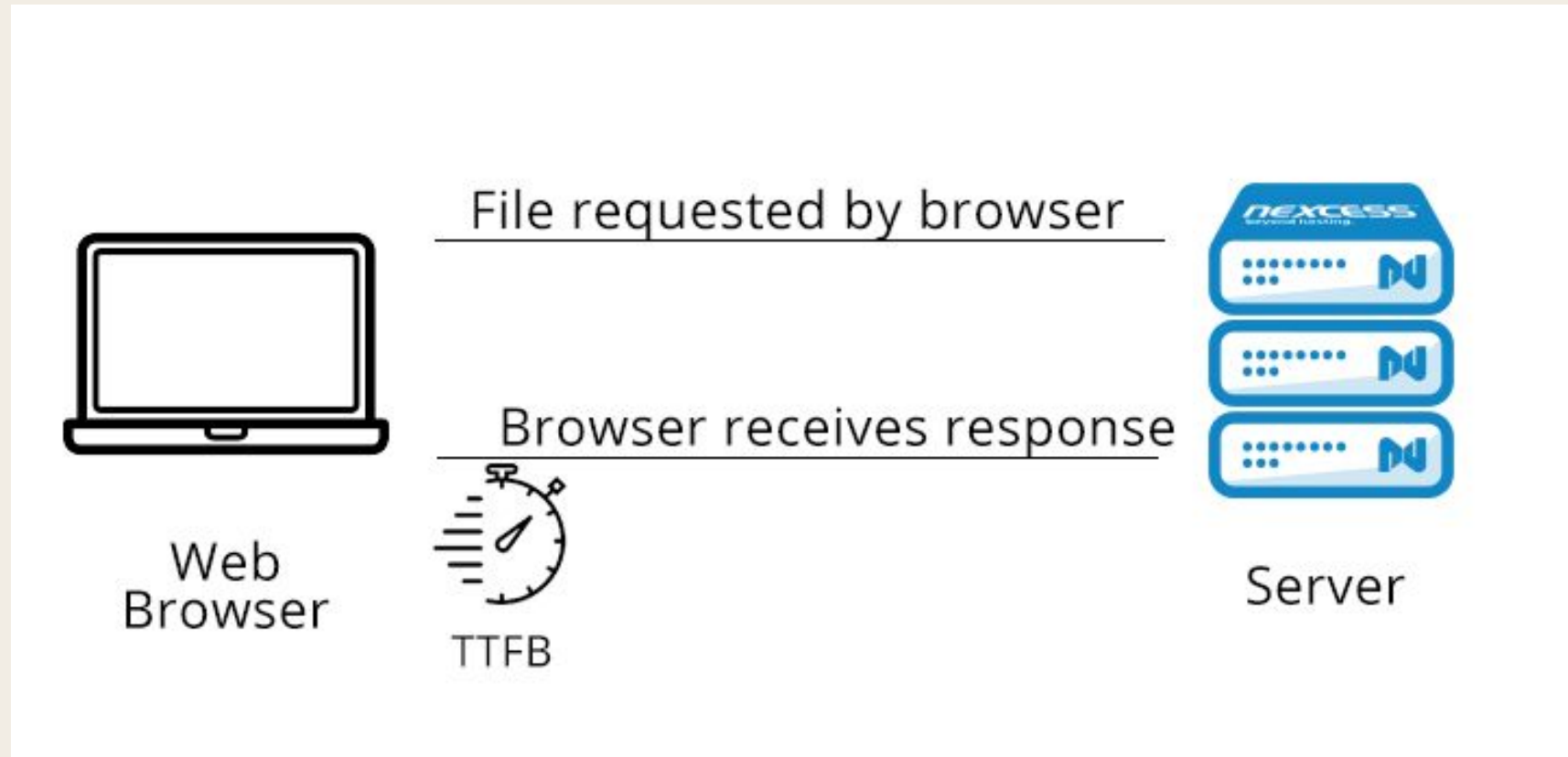
---

Road to more performant applications

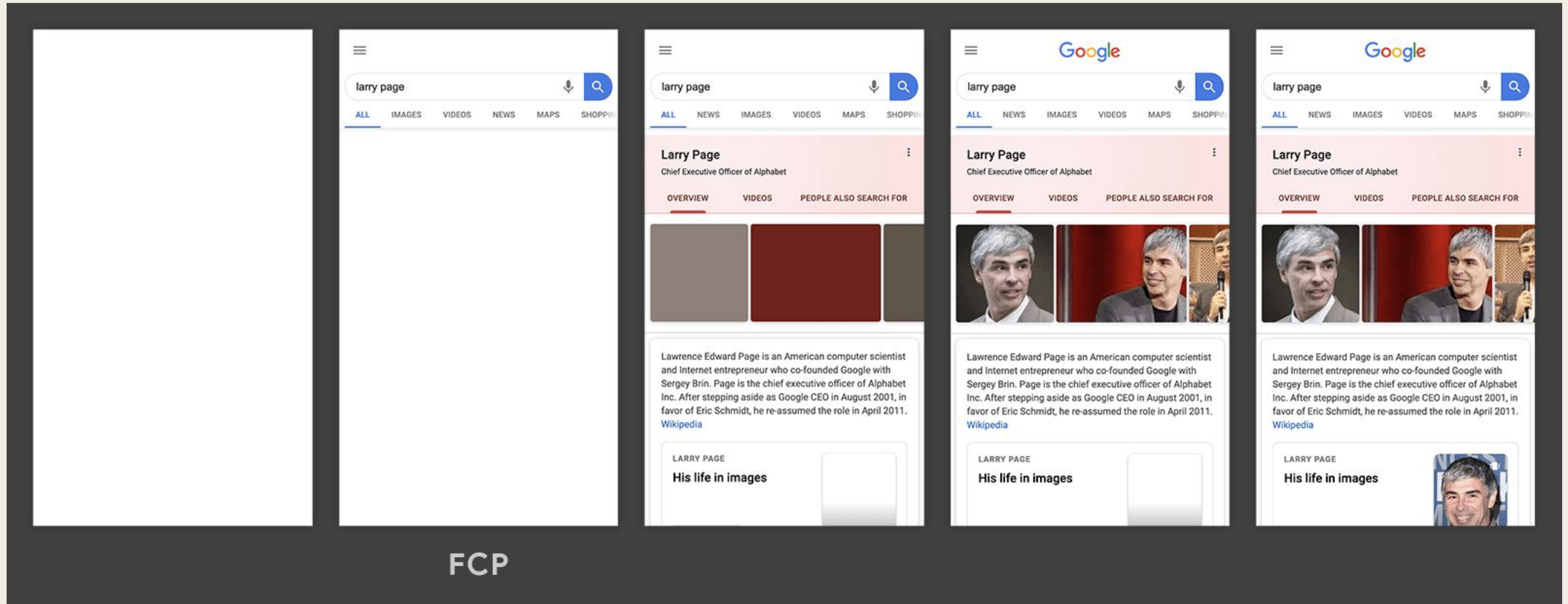


# Web vitals

# Time To First Byte (TTFB)

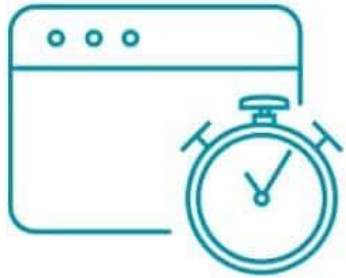


# First Contentful Paint (FCP)





# Time To Interactive (TTI)



## Page Speed Mess-Stationen





The image features a light beige background with decorative curved lines in the top-left and bottom-right corners. These lines are composed of multiple overlapping layers in shades of light green and light blue, creating a soft, abstract effect.

Your application code is written  
in **JavaScript**.

```
<!-- jsx -->  
<h1>Welcome to {name}</h1>
```

```
<!-- angular template -->  
<h1>Welcome to {{ name }}</h1>
```

```
<!-- vue template -->  
<template>  
  <h1>Welcome to {{ name }}</h1>  
</template>
```

```
<!-- template -->  
<h1>Welcome to {{ name }}</h1>
```

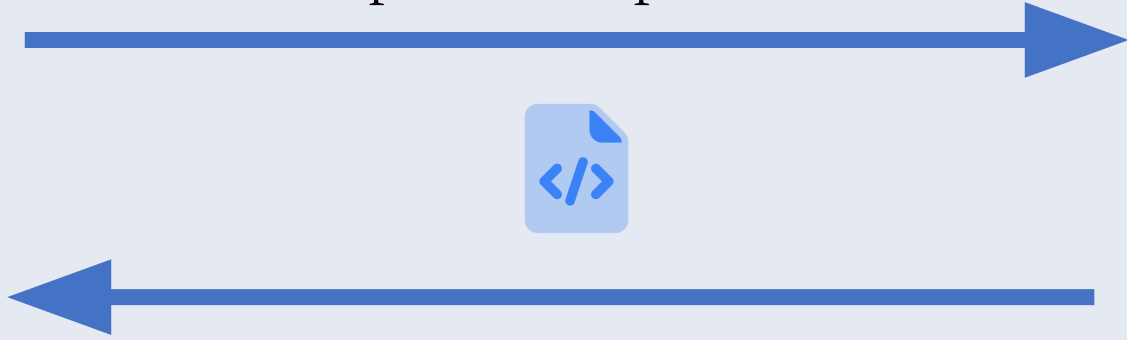
```
      <!-- html -->
<h1>Welcome to ViennaJS</h1>
```

- Initializing your app
  - By setting up the state
  - Generating HTML
  - Adding Event Listeners

# Rendering



`https://example.com`



# Client-side rendering



```
<!DOCTYPE html>
<html lang="en">
  <head>
    <title>CSR</title>
  </head>
  <body>
    <noscript>You need to enable JavaScript to run this app.</noscript>
    <div id="root"></div> } HTML
    <!-- entire app -->
    <script src="bundle.js"></script> } JavaScript
  </body>
</html>
```

# Some observations

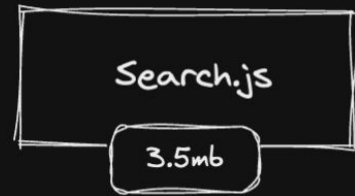
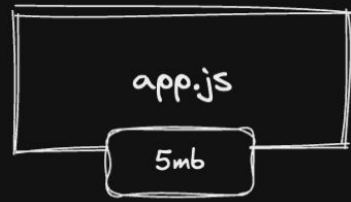
- Potentially Good TTFB Serving static content
- Potentially Bad FCP Lots of JS
- Potentially Bad TTI Lots of JS




# Lazy Loading

---







```
<!DOCTYPE html>
<html lang="en">
  <head>
    <title>Lazy Loading</title>
  </head>
  <body>
    <noscript>You need to enable JavaScript to run
this app.</noscript>
    <div id="root"></div>
    <!-- main code -->
    <script src="home.js"></script>
  </body>
</html>
```



```
// home.js  
const SearchComponent = React.lazy(  
  () => import('./SearchComponent')  
);
```

# Server-side rendering

```

<!DOCTYPE html>
<html lang="en">
  <head>
    <title>SSR</title>
  </head>
  <body>
    <div id="root">
      <h1>Welcome to ViennaJS</h1>
    </div>
    <!-- javascript code -->
    <script src="main.js"></script>
  </body>
</html>

```

HTML

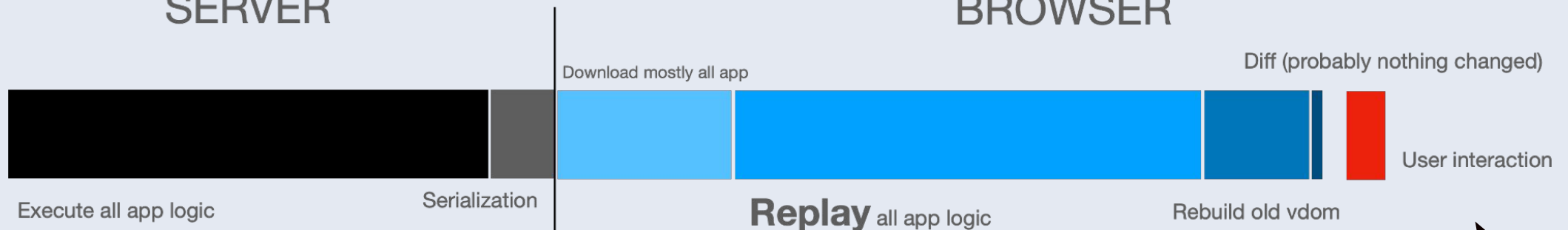
JS

Javascript execution



SERVER

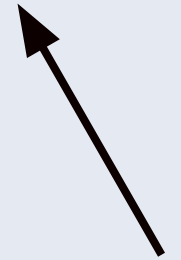
BROWSER



Rendering



Hydration



TTI



# Qwik

- HTML first framework
- Aggressive lazy loading
- No Eager Hydration
- React-like API
- Can use React, Vue and other frameworks



The background features a large, abstract graphic composed of several overlapping, semi-transparent circular bands. The bands are primarily in shades of light blue and light green, creating a layered, tunnel-like effect. The text is centered within this graphic.

# React-like API



```
import { component$, useStore, $ } from '@builder.io/qwik';

export const randomValue = () => Math.floor(Math.random() * 6 + 1);

export default component$(() => {
  const number = useStore({ list: [3, 2] });

  const generateNumber = $(() => {
    const random = randomValue();
    number.list = [...number.list, random]
    console.log(number.list);
  });

  return (
    <div>
      <h2>List: {JSON.stringify(number.list)}</h2>
      <button onClick$={generateNumber}>Generate Number</button>
    </div>
  )
});
```

} Hooks

} JSX

} Component



# Code splitting

```
import { component$, useStore, $ } from '@builder.io/qwik';

export const randomValue = () => Math.floor(Math.random() * 6 + 1);

export default component$(() => {
  const number = useStore({ list: [3, 2] });

  const generateNumber = $(() => {
    const random = randomValue();
    number.list = [...number.list, random]
    console.log(number.list);
  });

  return (
    <div>
      <h2>List: {JSON.stringify(number.list)}</h2>
      <button onClick$={generateNumber}>Generate Number</button>
    </div>
  )
});
```

# Chunk file

```
import { a as useLexicalScope, r as randomValue } from './app2.js';
export { _ as _hW } from './app2.js';

const app_component_generateNumber_amu1GzV0Qfw = () => {
  const [number] = useLexicalScope();
  const random = randomValue();
  number.list = [
    ...number.list,
    random
  ];
  console.log(number.list);
};

export { app_component_generateNumber_amu1GzV0Qfw };
```



Symbol

The background features a large, stylized circular graphic composed of multiple concentric rings. The rings are colored in shades of blue and green, with some rings being solid and others semi-transparent, creating a layered effect. The text is centered within this graphic.

HTML first



```
<body>
  <!--qv q:id=0 q:key=Ncbm:0t_0-->
  <div>
    <h2>List: [3,2]</h2>
    <button
on:click="app_component_generatenumbr_amu1gzv0qfw.js#app_component_generateNumber_amu1GzV0Qfw[0]"
      q:id="1"
    >
      Generate Number
    </button>
  </div>
  <!--/qv-->
</body>
```



# QRL


Chunk file

Symbol

app\_component\_generatenumber\_amu1gzv0qfw.js#app\_component\_generateNumber\_amu1GzV0Qfw  
[0]

Information about State(Lexical Scope)

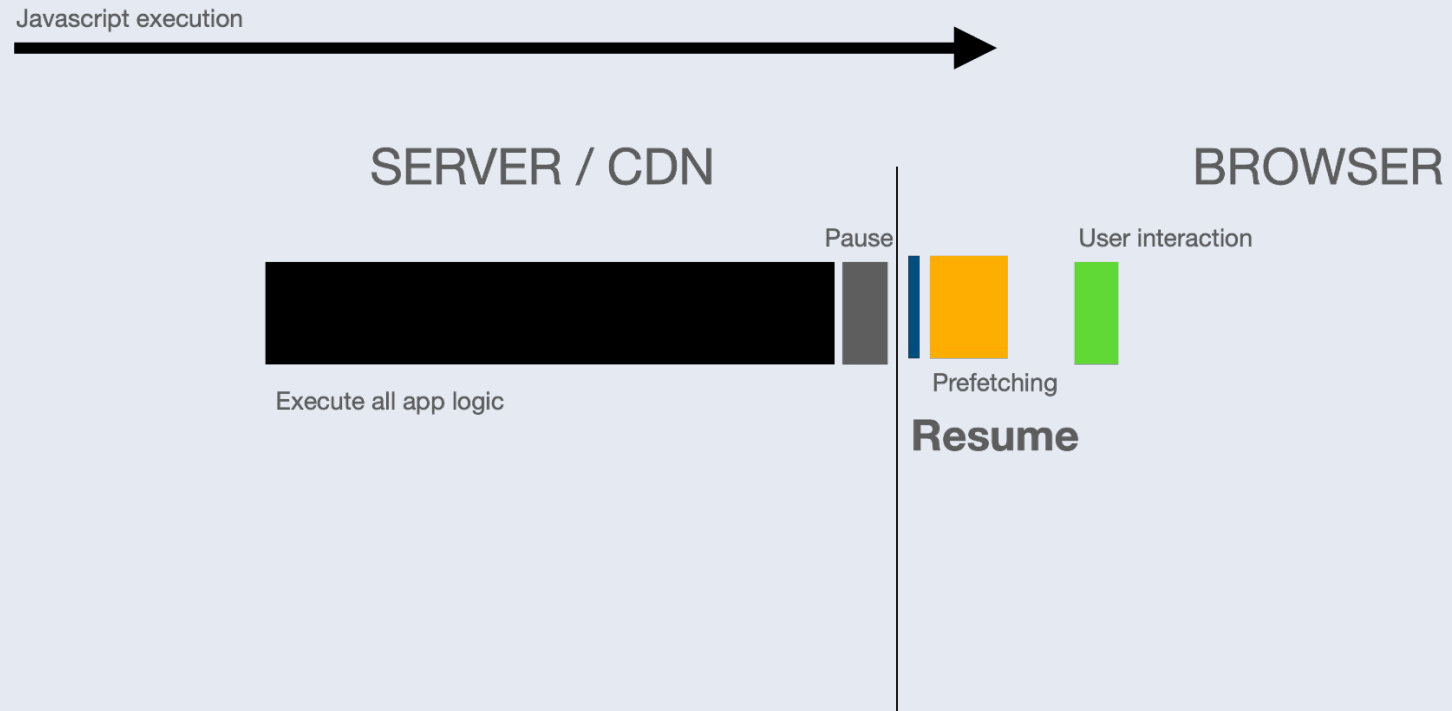
# State

```
  
<script type="qwik/json">  
  {  
    "refs": { "1": "0!" },  
    "ctx": { "0": { "h": "4", "s": "0!" } },  
    "objs": [  
      { "list": "3" },  
      3,  
      2,  
      ["1", "2"],  
      "\u0002app_component_ncbm0trxwgc.js#app_component_Ncbm0Trxwgc"  
    ],  
    "subs": [["0 #0 list"]]  
  }  
</script>
```



Bringing it all together

# Resumable (Qwik)



# Recap

- Web vitals
- Rendering
- Hydration
- Intro to Qwik

To know more <https://qwik.builder.io>



[@imsinghk7](https://twitter.com/imsinghk7)