



DECODING WEBPACK MODULE FEDERATION

Nikhil Motiani



Hello!

I am Nikhil Motiani

- Full Stack Developer @PayPal
- JS & Web Enthusiast

You can find me at:

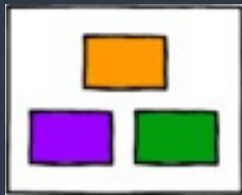


@computnik



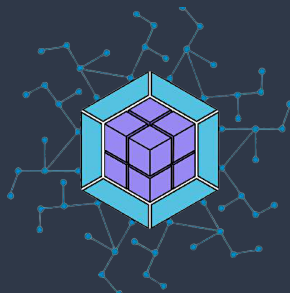
<https://computnik.dev>

AGENDA



Micro-frontends

Extending the idea of micro-services to frontend




Module Federation

How can we do micro-frontends with just webpack, using Module federation



Under the hood

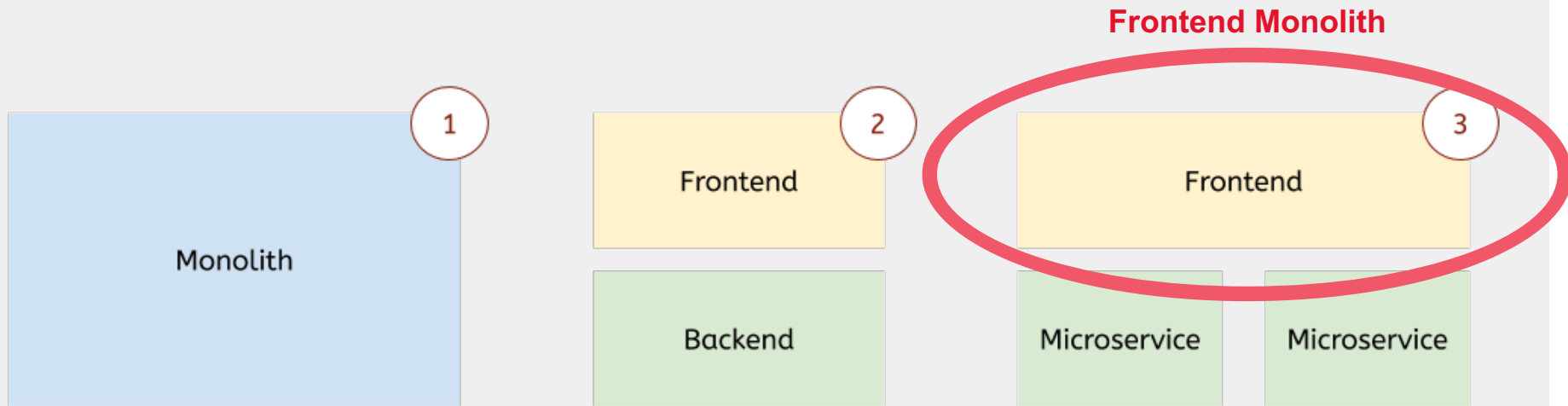
How does Module federation work internally



Building software in large organizations is **HARD!**

It requires **HIGH COORDINATION** among teams.
What helps: **Strategies** to make them **AUTONOMOUS**

Micro-services: Existing solution for Backend

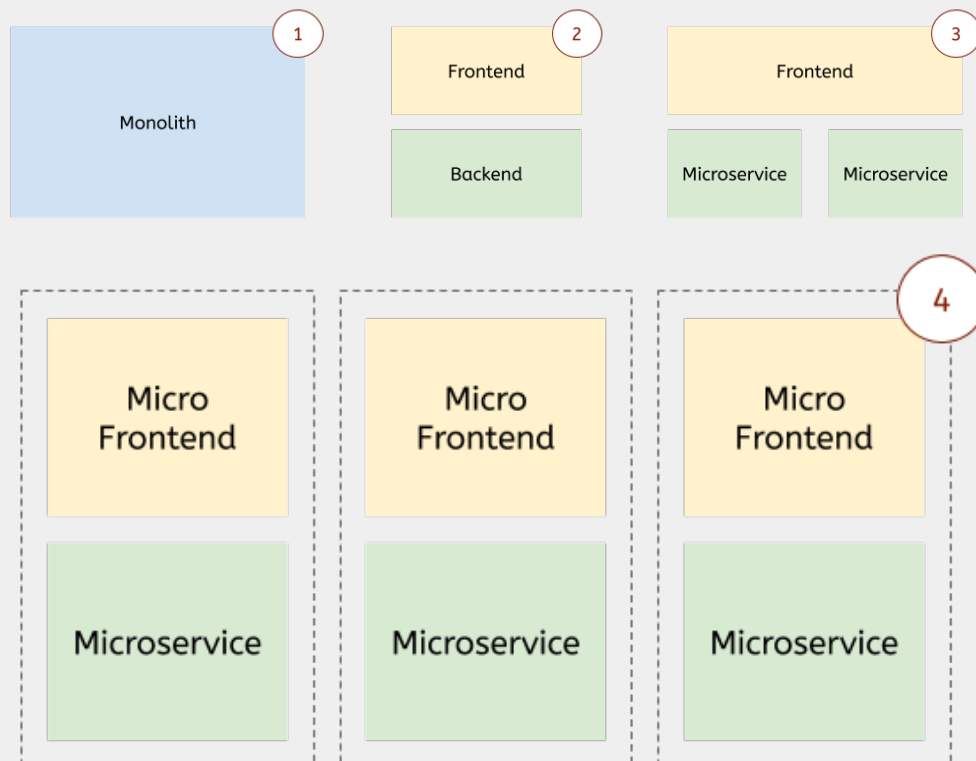




Micro-frontends

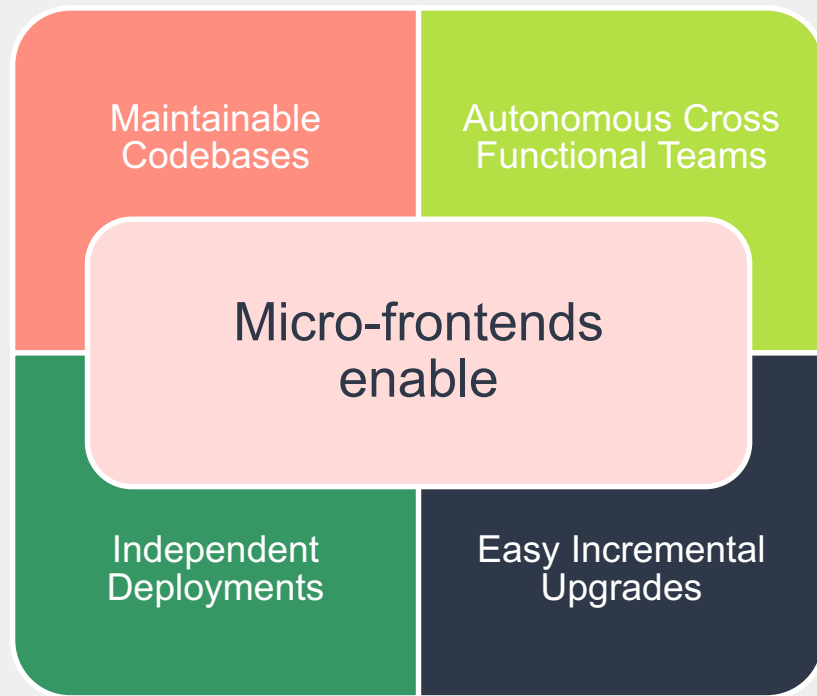
Extending concept of Micro-services to frontend

Extending the concept to Frontend: Micro-frontends



Micro-frontends?

- ❑ Term coined in 2016
- ❑ Techniques/Strategies & Recipes for:
 - Modern Web App
 - Multiple Teams
 - Shipping Features independently

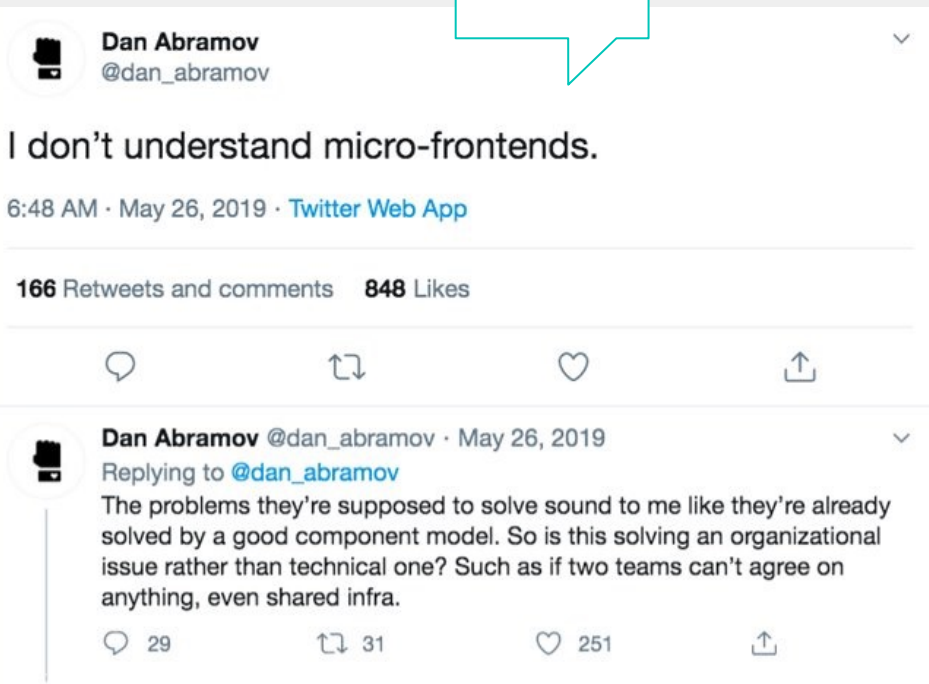


Who uses it?

- Walmart
 - SAP
 - Fiverr
 - HelloFresh
 - Starbucks
 - Upwork
 - Namecheap
 - Smapiot
 - Entando
 - CloudFactory
 - Microsoft
 - Spotify
 - SkyScanner
 - IKEA
 - Leroy Merlin
 - Zalando
 - OpenTable
 - New Relic
 - Elsevier
 - Illimity
- & ~30+ other organizations

Do I need it?

“



Tools available for Micro-frontends

Many feature rich OSS frameworks out there:



single-spa

icestark



qiankun



Piral

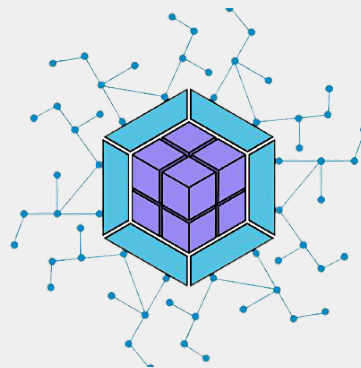


Luigi



Isomorphic Layout Composer


Evolving current tools:
Module Federation in Webpack




Sample Use-Case

The Model Store

basket: 0 item(s)






Tractor Porsche-Diesel Master 419



buy for 66,00 €

Related Products



Sample Use-Case

The Model Store

basket: 0 item(s)

Related Products



Tractor Porsche-Diesel Master
419



buy for 66,00 €



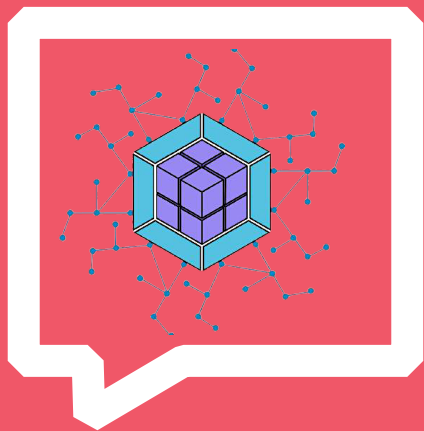
Team Product 

Team Checkout 

Team Inspire 

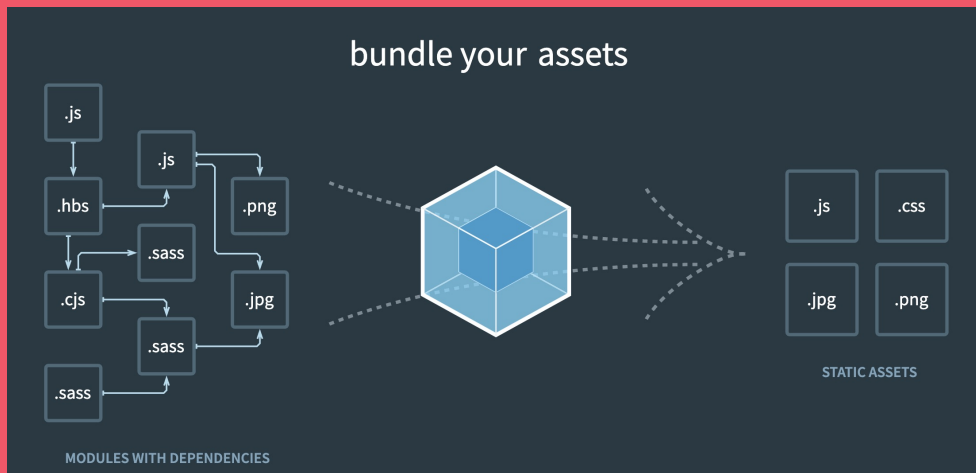


DECODING **WEBPACK** **MODULE FEDERATION**



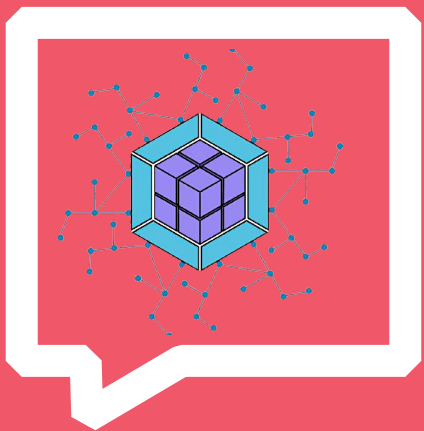
Webpack

Bundler for JS & friends



Webpack TLDR;

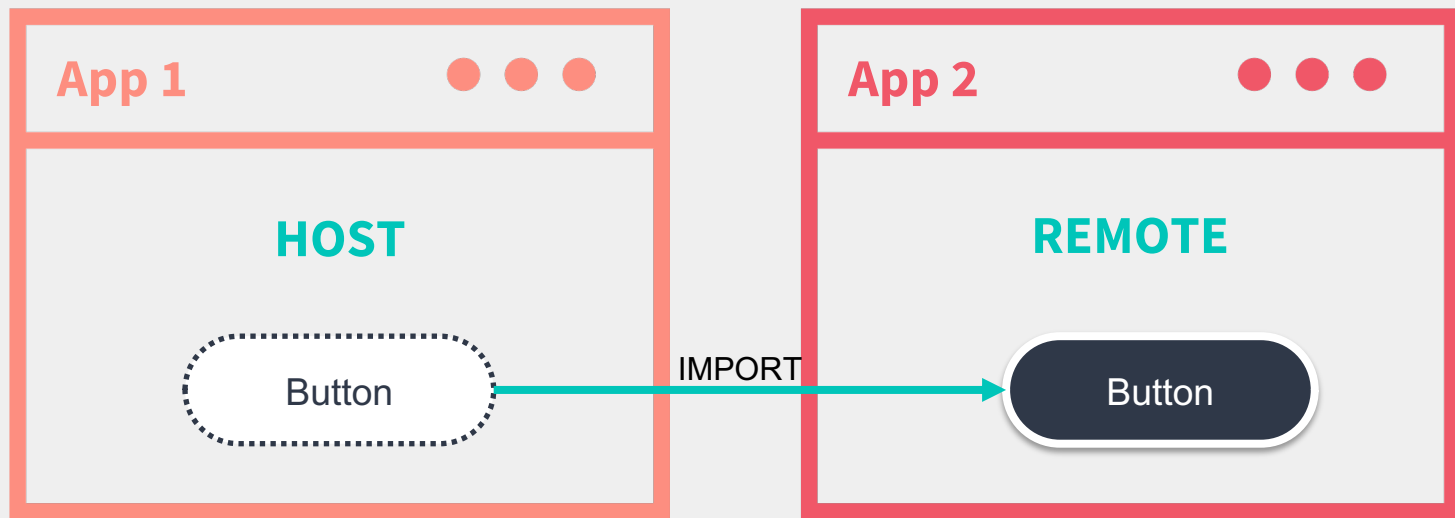
- Packs many modules into a few bundled assets.
- Code Splitting allows for loading parts of the application on demand.
- Through "loaders", modules can be commonjs, amd, es6 modules, css, images, json, LESS, ... and your custom stuff.
- Rich Plugin system. Even internal code is plugins.
- Very Popular! (*16M downloads/week, 58k Github Stars*)
- It comes shipped with various frameworks like:
 - Next.js
 - Gatsby



Module Federation

Doing Micro-frontends using Webpack

Example



From imports to remote modules

```
const RemoteButton = import('http://localhost:3002/app2/Button')
```

Doesn't Work!

Even the lazy parts must be
known at BUILD TIME

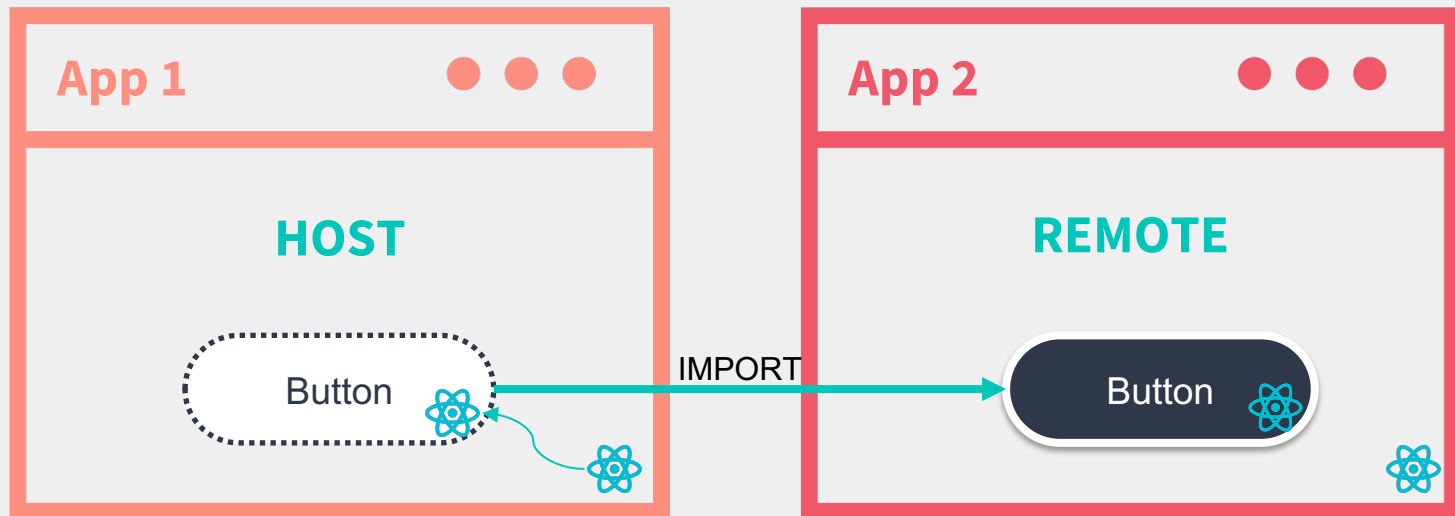
Module Federation Capabilities

Expose something

Consume Something

Share Dependencies

Example



Module Federation Plugin

Expose something

Consume Something

Share Dependencies

```
new ModuleFederationPlugin({  
  name: "app1",  
  exposes: {  
    "./Button": "./src/components/button.js",  
  },  
  remotes: {  
    app2: `app2@${getRemoteEntryUrl(3002)}`,  
  },  
  shared: ["react", "react-dom"]  
}),
```

Module Federation

HOST (App1)

```
import('app2/Button')
```

```
// webpack.config.js
```

```
// ... other config
```

```
plugins: [
```

```
  new ModuleFederationPlugin({
```

```
    name: "app1",
```

```
    remotes: {
```

```
      "app2": "app2@http://localhost:3002/app2/remoteEntry.js"
```

```
    }
```

```
  })
```

REMOTE (App2)

```
// webpack.config.js
```

```
// ... other config
```

```
plugins: [
```

```
  new ModuleFederationPlugin({
```

```
    name: "app2",
```

```
    filename: "remoteEntry.js",
```

```
    exposes: {
```

```
      "./Button": "./src/button.js"
```

```
    }
```


```
  })
```

Let's see
DEMO



Let's see
Another DEMO





Isn't that amazing?

That's great, So **How does it Work?**



The 3 internal Plugins

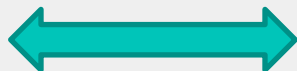
Module Federation's Capabilities - Plugins

Capabilities

Expose something

Consume Something

Share Dependencies



Plugins

Container Plugin

Container Reference Plugin

Share Plugin

ModuleFederationPlugin

```
class ModuleFederationPlugin {
  apply(compiler) {
    // ... some code
    const {name, exposes, library, filename, remotes, shared, shareScope} = options
    compiler.hooks.afterPlugins.tap("ModuleFederationPlugin", () => {
      if (Array.isArray(exposes)? exposes.length > 0: Object.keys(exposes).length > 0)) {
        new ContainerPlugin({name, library,filename,exposes}).apply(compiler);
      }
      if (Array.isArray(remotes) ? remotes.length > 0: Object.keys(remotes).length > 0) {
        new ContainerReferencePlugin({remoteType,remotes}).apply(compiler);
      }
      if (shared) {
        new SharePlugin({shared, shareScope}).apply(compiler);
      }
    });
  }
}
```

Container Plugin

Additional Compiler entry
for remote container

A module map of all the
exposed modules/assets

Capability to consume the
shared modules for
container

```
/***/ "webpack/container/entry/app2":  
/!*****!\\  
!*** container entry ***!  
\*****/  
  
var moduleMap = {  
  "./Button": () => {  
    return Promise.all([  
      require("webpack_sharing_consume_default_react_4030"),  
      require("src_Button_js")  
    ]).then(() => () =>  
      (require("./src/Button.js"))  
    );  
  }  
};
```

Container Reference Plugin

Reference to remote containers as externals

Maintains a map of References to remote containers

On demand resolution of imports to remote modules

```
/* webpack/runtime/remotes loading */
var idToExternalAndNameMapping = {
  "webpack/container/remote/app2/Button": [
    "default",
    "./Button",
    "webpack/container/reference/app2"
  ]
};

/***/ "webpack/container/reference/app2":
/*!*****!*\
!external app2@//localhost:3002/remoteEntry.js *!
\******/
__webpack_require__(
  "//localhost:3002/remoteEntry.js",
  (event) => {
    // handle errors & successes
  },
  "app2");
```

Shared Plugin

Creates a registry for shared dependencies

Loads them on demand, whenever needed

Enables Self-healing by resolving to either shared or fallback dependency

```
/* webpack/runtime/sharing */
```

```
register("react-dom", "16.14.0", () =>  
  Promise.all([  
    require("vendors-node_modules_react-dom_index_js"),  
    require("webpack_sharing_consume_default_react_react")  
  ]).then(() => () =>  
    require("../..node_modules/react-dom/index.js")  
  )  
);
```



```
import { getElement, isDevice } from 'react-native';
import { PLATFORM, FUNCTION } from 'react-native';
import { getRememberedFunding } from 'react-native';
import { getComponents, getFundingSource } from 'react-native';

import type { OnShippingChange } from 'react-native';
import { BUTTON_LAYOUT, BUTTON_SIZE } from 'react-native';
import { determineEligibleFunding } from 'react-native';
import { isSupportedNativeDevice } from 'react-native';
```

```
import { MarksElement } from 'react-native';
```

```
const DEFAULT_HEIGHT = 20;
```

```
type MarksInstance = {
  isEligible: () => boolean;
  render: (string | HTMLDivElement) => ReactElement;
```

```
};
```

```
type MarksProps = {
```

```
  fundingSource?: ?$Values<FUNDING_SOURCE>;
  onShippingChange?: OnShippingChange;
```

```
};
```

```
export type MarksComponent = (MarksProps) => ReactElement;
```

```
export const getMarksComponent = () => {
  function Marks({ fundingSource, onShippingChange }) {
    const height = DEFAULT_HEIGHT;
```

```
    const eligibility = determineEligibility(fundingSource, onShippingChange);
    const device = isDevice();
```



JS@

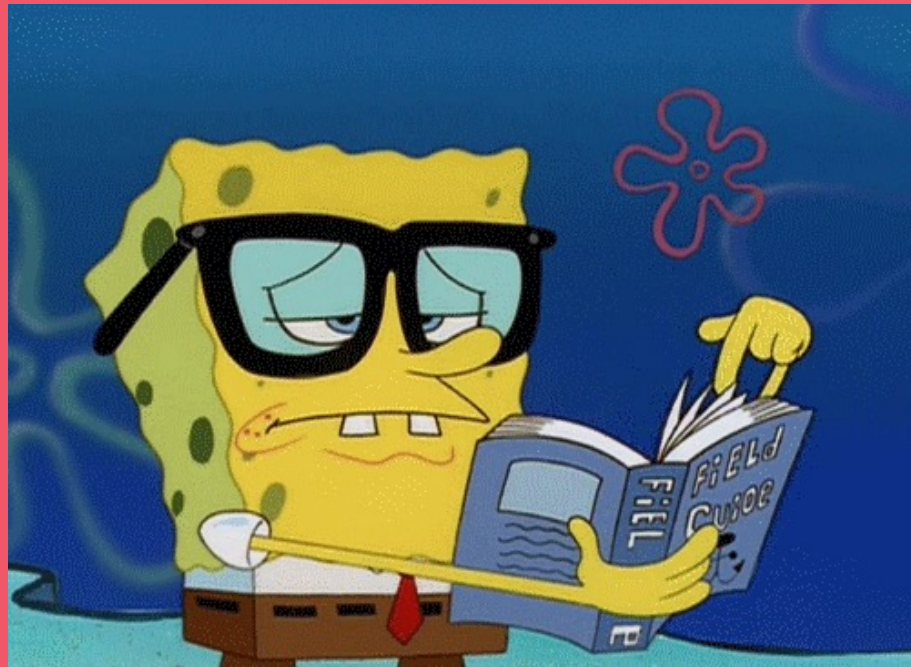
2021

#JsPayPal2021NikhilMotiani3880

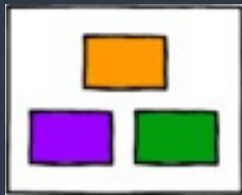


Review

What did we learn?

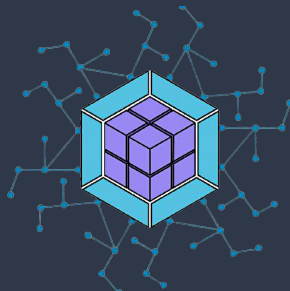


Let's review some concepts



Micro-frontends

Extending the idea of micro-services to frontend



Module Federation

How can we do micro-frontends with just webpack, using Module federation



Under the hood

How does Module federation work internally



Next steps



Where to go from here?

Evaluate

- if you need Micro-frontends? And what parts of your frontend needs it?
- Do you want a full-blown solution (like OSS frameworks we saw earlier)
- Or Extend current tooling like webpack (if you are using it, already)

Other areas to consider:

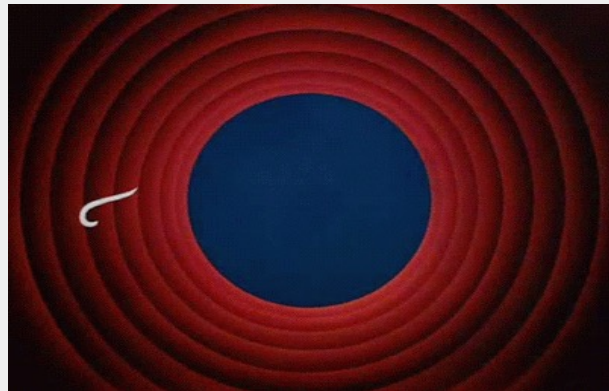
- Isolated CSS (CSS Modules, CSS in JS, etc.)
- Coherent UI, Style guides/ patterns
- Static vs Dynamic remotes
- Integration testing
- Performance





Thanks

Any Questions?



You can find me at:



@computnik



<https://computnik.dev>

Appendix



Extras!!



References

Links to resources

References

1. [Webpack's Module Federation Docs](#)
2. [Module Federation Community Examples](#)
3. [Step by Step Micro-frontends using webpack module federation](#)
4. [Webpack: Upcoming changes in 2021](#)
5. Code demos shown - [Sharing Button Demo](#), [Dashboard demo](#)
6. Some great YouTube videos:
 1. [Shared App Shell, State, Routing and Components](#)
 2. [Webpack 5 Module Federation - Zack Jackson - CityJS Conf 2020](#)
 3. [Full Site Federation in Webpack 5 by Jack Herrington](#)
 4. [Everything's a plugin - Sean Larkinn](#)