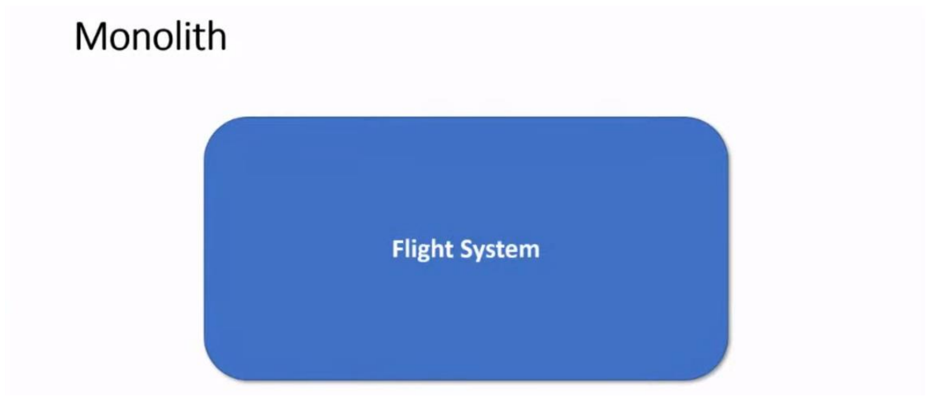
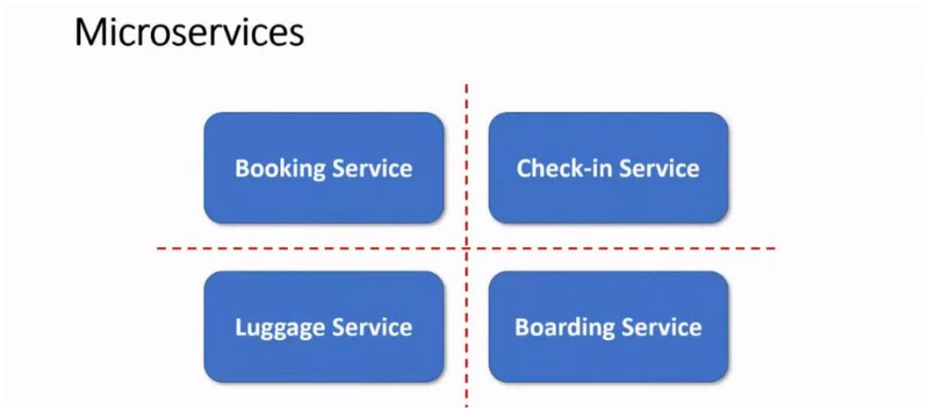
The Microfrontend Revolution
Module Federation with Angular
Manfred Steyer, ANGULARarchitects.io

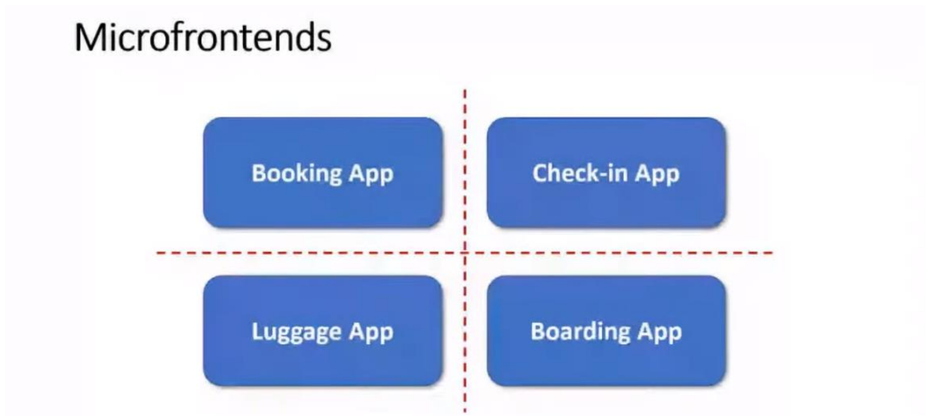
Manfred Steyer stops by to educate us on what this whole micro front end movement is all about and explain the concept of **module federation** with webpack-based approach and the benefits we can realize from it in our Angular applications.

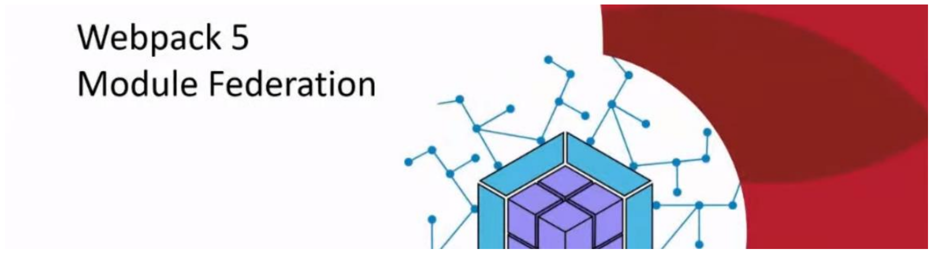
Do you remember her?

Software Engineering is a Team Sport

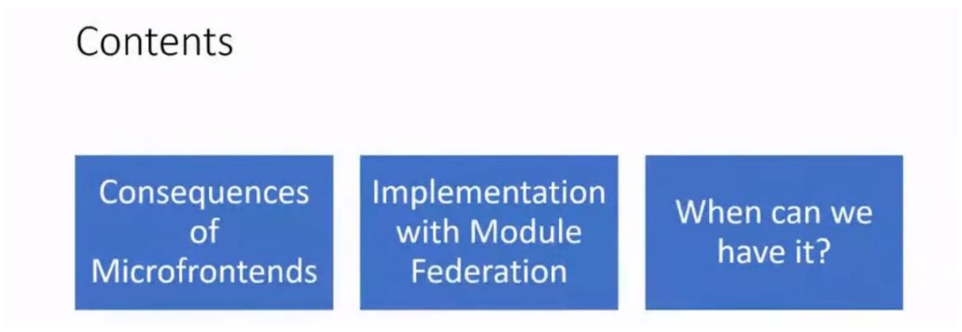








This gives a sound solution for implementing micro-frontends using Angular module federation via Webpack 5



Contents

- Consequences of Microfrontends
- Implementation with Module Federation
- When can we have it?

About me…

Manfred Steyer

Manfred Steyer, **ANGULAR***architects.io*

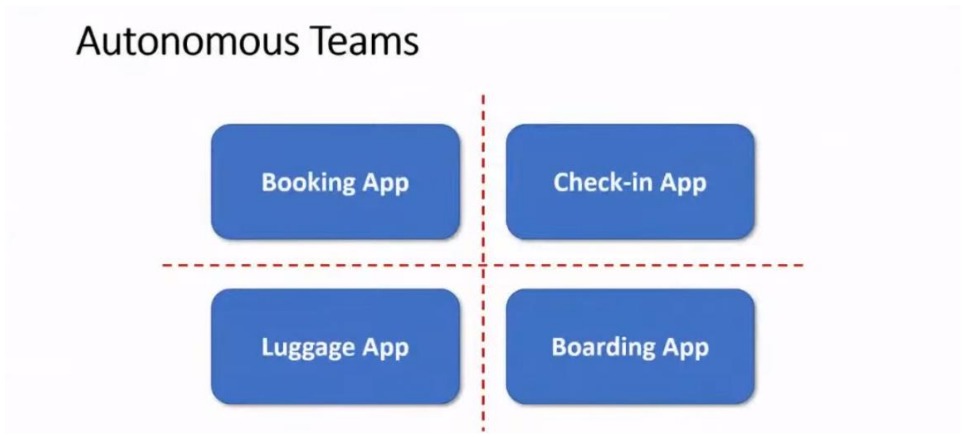
ANGULAR ARCHITECTURES
FOR ENTERPRISE-APPLICATIONS
Google Developers Experts 2019
Angular GDE

(Remote) Angular Workshops and Consulting

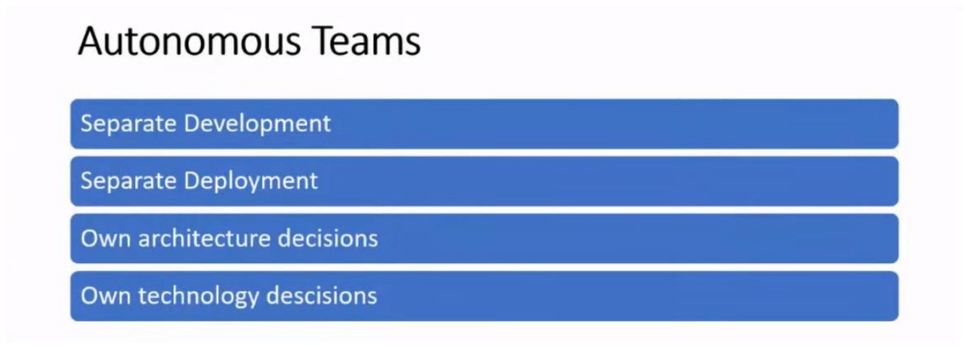
Google Developer Expert for Angular

Trusted Collaborator in the Angular Team

1: Consequences of Microfrontends

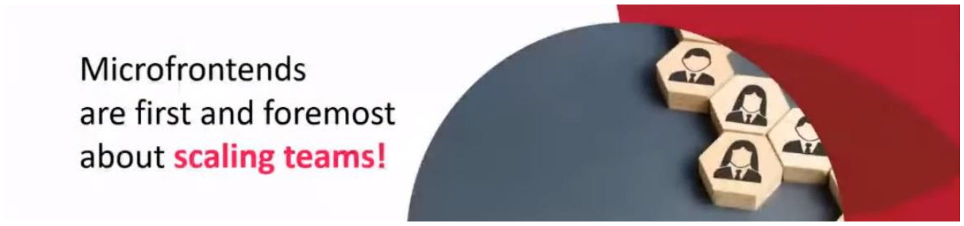


Autonomous Teams

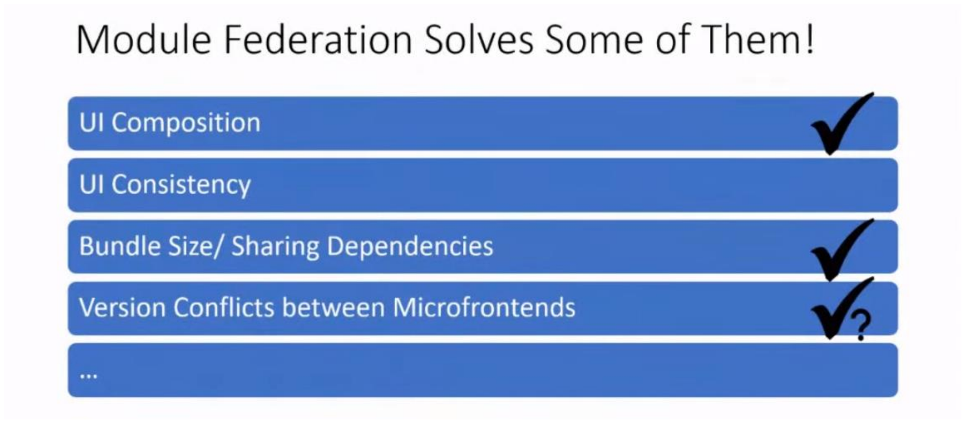


Autonomous Teams

- Separate Development
- Separate Deployment
- Own architecture decisions
- Own technology descisions


Microfrontends
are first and foremost
about **scaling teams!**

Challenges

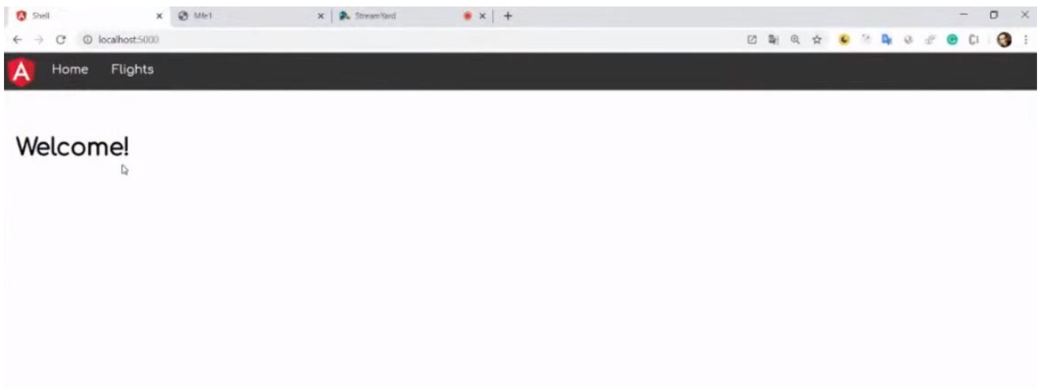

Module Federation Solves Some of Them!

UI Composition	✓
UI Consistency	
Bundle Size/ Sharing Dependencies	✓
Version Conflicts between Microfrontends	✓?
…	

Module federation in Webpack 5 solves some of the issues raised above.

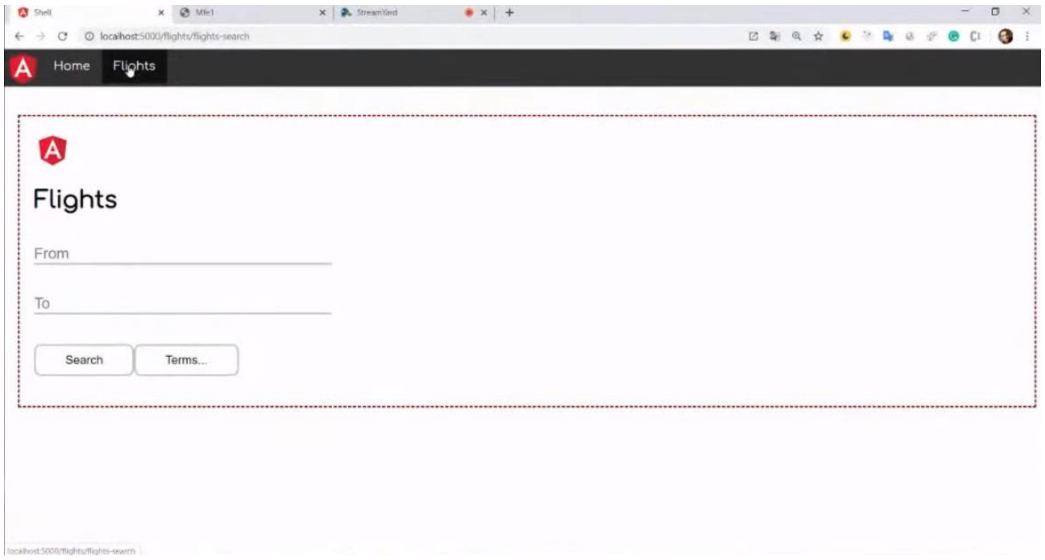
2: Webpack 5
Module Federation

DEMO

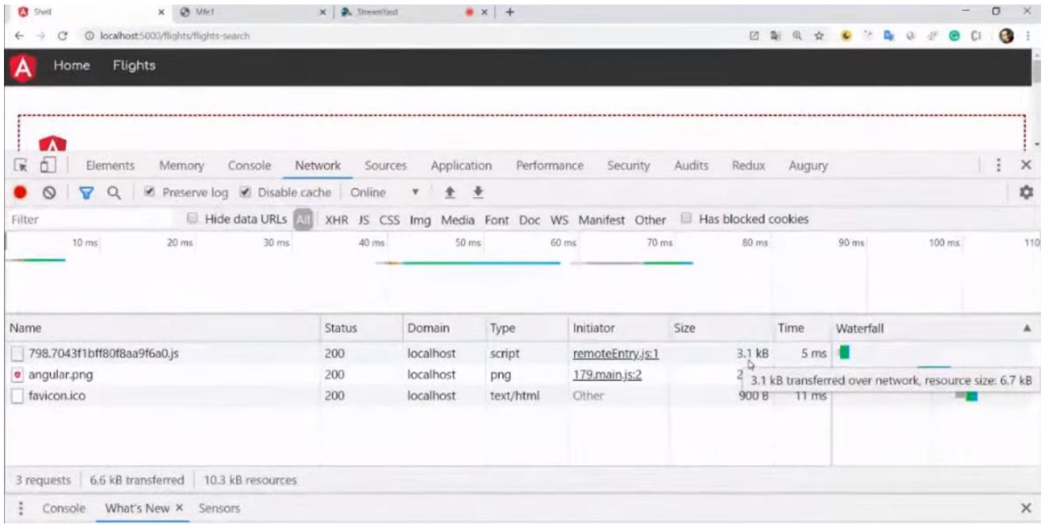


Welcome!

This our MFE shell is capable of loading the separate MFEs when needed



Each MFE has been developed, compiled and deployed separately. The shell is just loading the newest version of each MFE when needed.

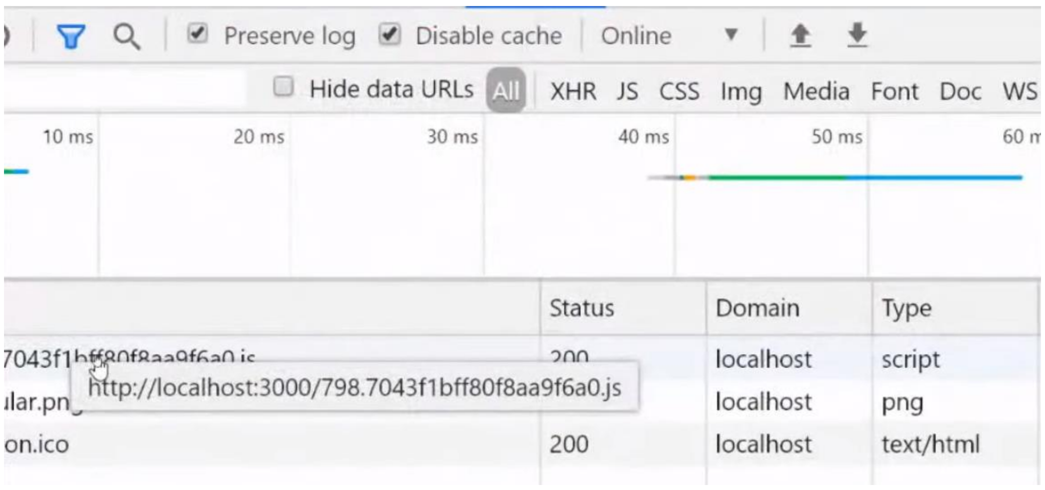


Name	Status	Domain	Type	Initiator	Size	Time
798.7043f1bff80f8aa9f6a0.js	200	localhost	script	remoteEntry.js:1	3.1 kB	5 ms
angular.png	200	localhost	png	179.main.js:2	2	
favicon.ico	200	localhost	text/html	Other	900 B	11 ms

3.1 kB transferred over network, resource size: 6.7 kB

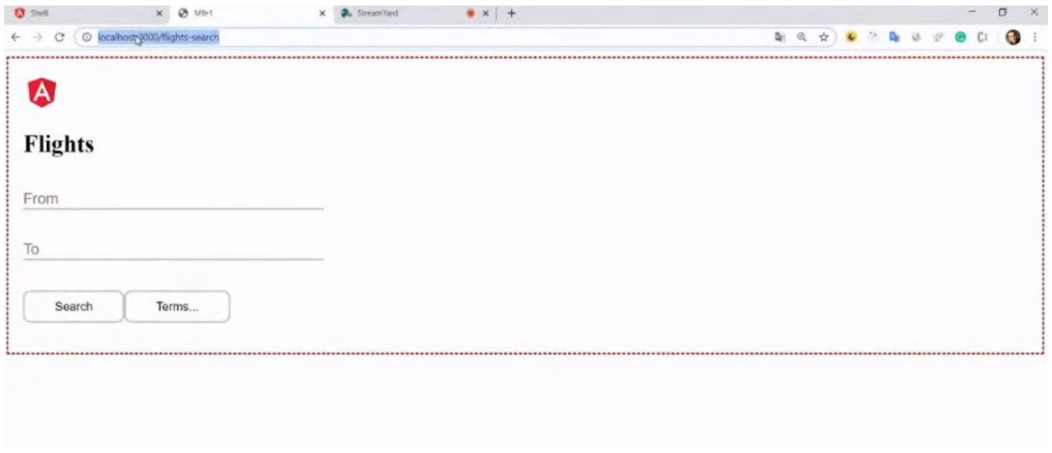
3 requests | 6.6 kB transferred | 10.3 kB resources

Each MFE chunk is loaded when needed by the shell, it looks like but not exactly lazy loading

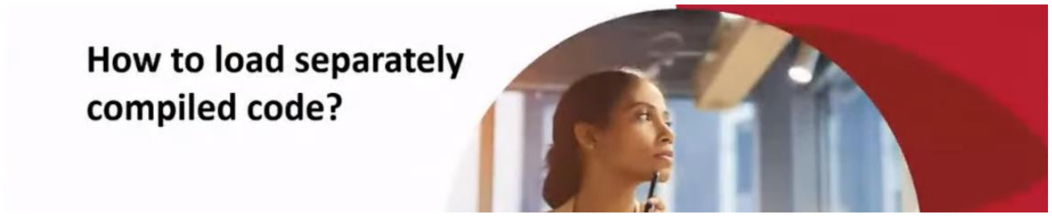


http://localhost:3000/798.7043f1bff80f8aa9f6a0.js

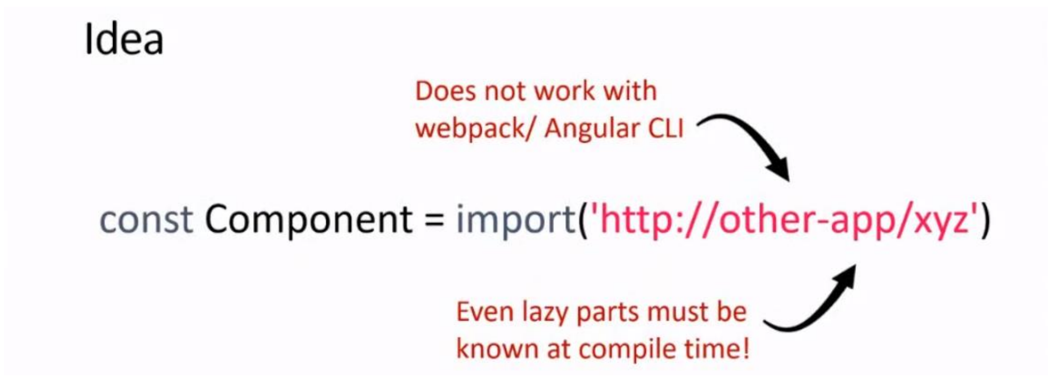
The chunks are really just loading a new endpoint



We can also run each MFE in isolation



How to load separately compiled code?



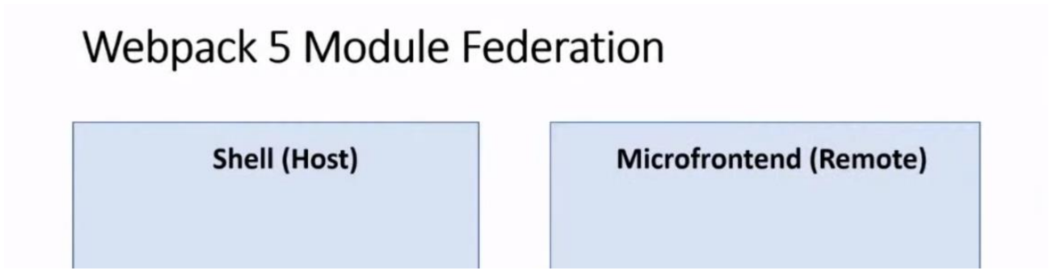
Idea

Does not work with webpack/ Angular CLI

```
const Component = import('http://other-app/xyz')
```

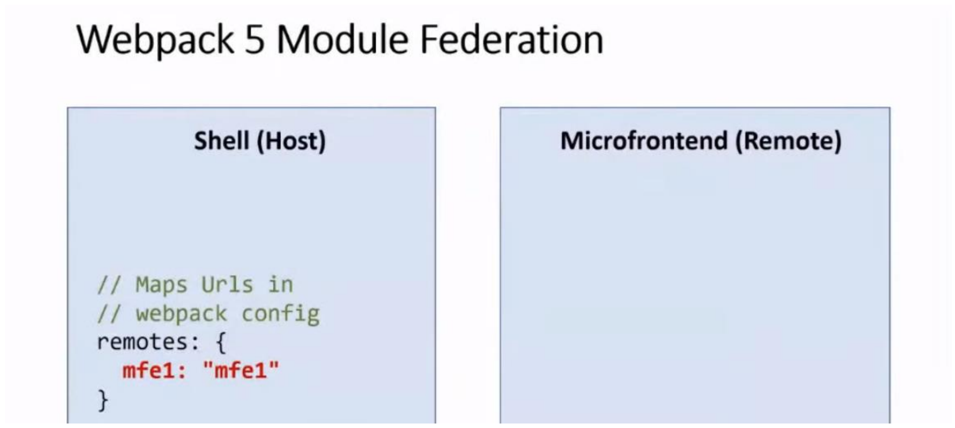
Even lazy parts must be known at compile time!

Webpack has dynamic imports that we can use but not here.



Webpack 5 Module Federation

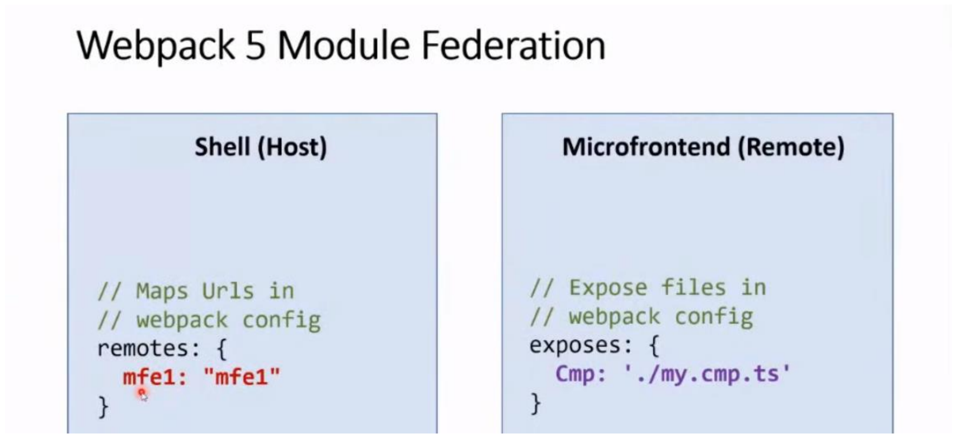
Webpack module federation clearly defines the host that will have the capability to load some remote self-contained apps that can be loaded into another app.



Webpack 5 Module Federation

```
// Maps Urls in
// webpack config
remotes: {
  mfe1: "mfe1"
}
```

Within the shell, we only need a small configuration section that defines the remote URLs/origins.

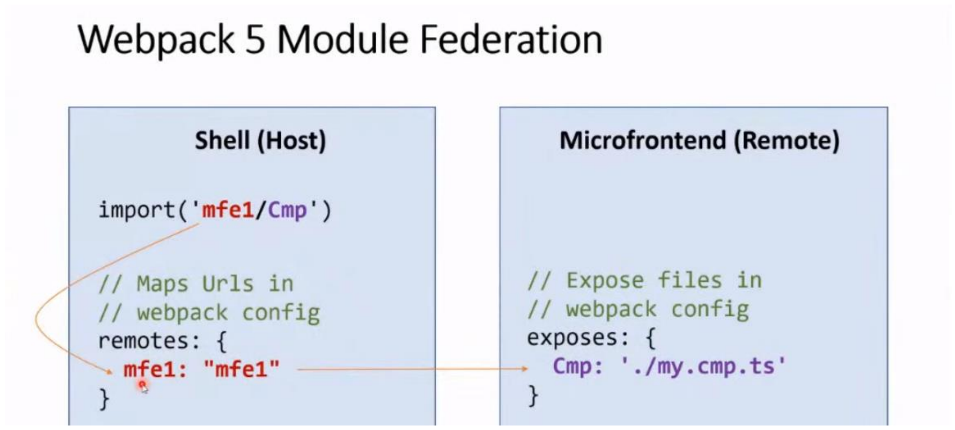


Webpack 5 Module Federation

```
// Maps Urls in
// webpack config
remotes: {
  mfe1: "mfe1"
}
```

```
// Expose files in
// webpack config
exposes: {
  Cmp: './my.cmp.ts'
}
```

The remotes/MFEs can expose things like files, components as above.

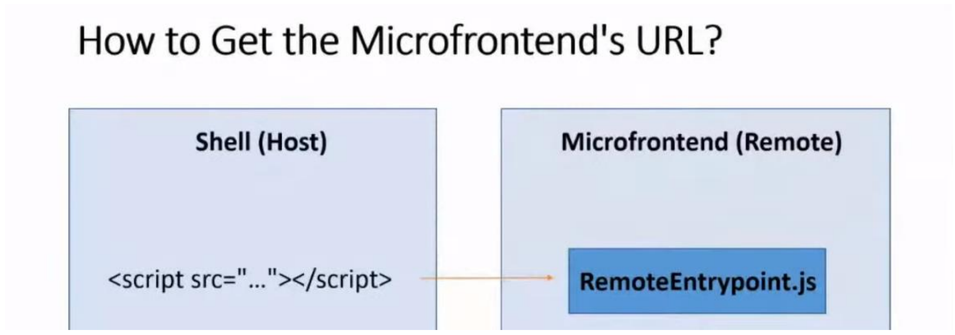


Webpack 5 Module Federation

```
import('mfe1/Cmp')

// Maps Urls in
// webpack config
remotes: {
  mfe1: "mfe1"
}
```

```
// Expose files in
// webpack config
exposes: {
  Cmp: './my.cmp.ts'
}
```

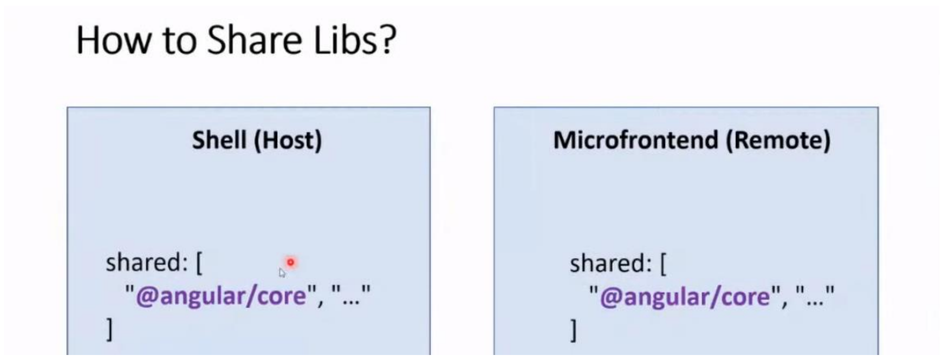
This now allows the shell to import and load views like mfe1.Cmp from different MFEs



How to Get the Microfrontend's URL?

```
<script src="..."></script>
```

When compiling the MFEs, we can get the RemoteEntrypoint.js file via a script or dynamic script tag in the host.



How to Share Libs?

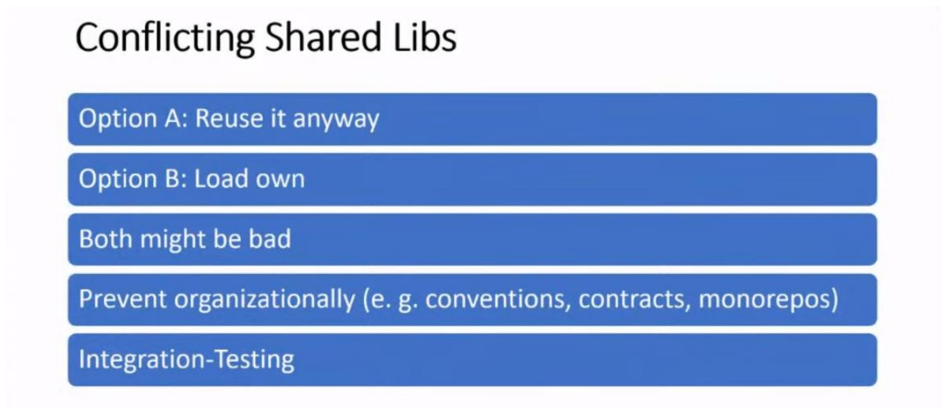
Shell (Host)

```
shared: [
  "@angular/core", "..."
]
```

Microfrontend (Remote)

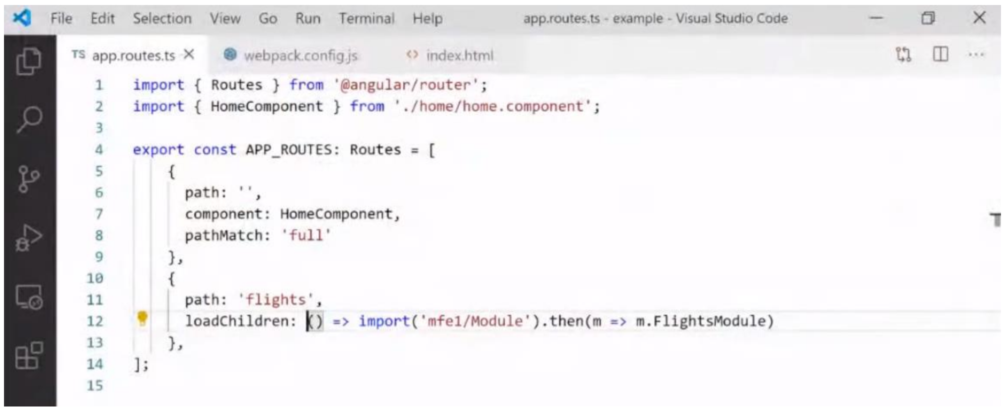
```
shared: [
  "@angular/core", "..."
]
```

A MFE will not load a library if that library is already listed in the shell's **webpack.config**'s shared array as above.



Conflicting Shared Libs

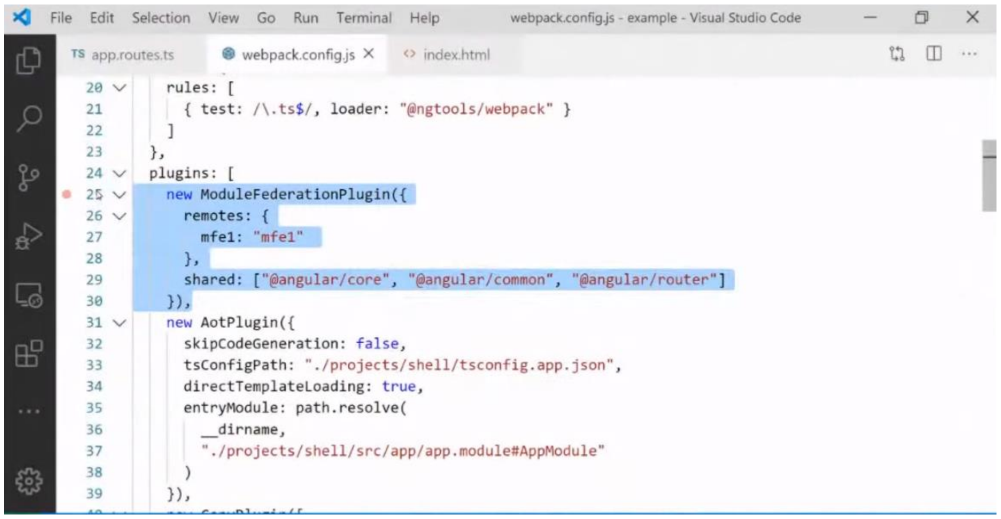
- Option A: Reuse it anyway
- Option B: Load own
- Both might be bad
- Prevent organizationally (e. g. conventions, contracts, monorepos)
- Integration-Testing



File Edit Selection View Go Run Terminal Help
app.routes.ts - example - Visual Studio Code
app.routes.ts
webpack.config.js
index.html

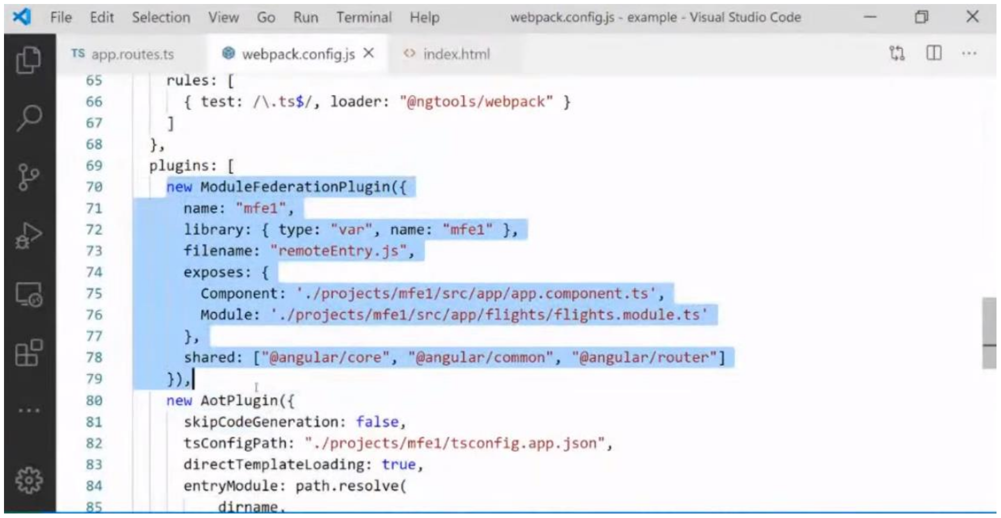
```
import { Routes } from '@angular/router';
import { HomeComponent } from './home/home.component';

export const APP_ROUTES: Routes = [
    {
      path: '',
      component: HomeComponent,
      pathMatch: 'full'
    },
    {
      path: 'flights',
      loadChildren: () => import('mfe1/Module').then(m => m.FlightsModule)
    },
];
```

This is the code for our shell and the routes of the shell defined, the shell will lazy load the MFEs that are actually an external app with its own routing configuration


File Edit Selection View Go Run Terminal Help
webpack.config.js - example - Visual Studio Code
app.routes.ts
webpack.config.js
index.html

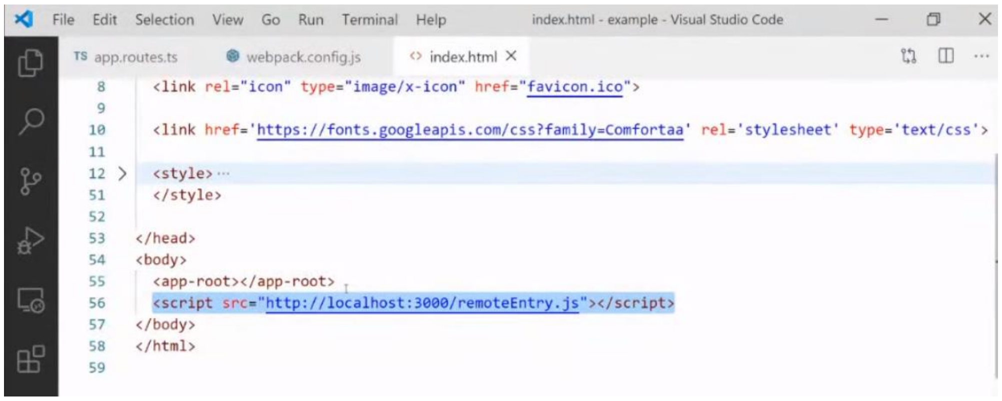
```
    rules: [
      { test: /\.ts$/, loader: "@ngtools/webpack" }
    ]
  },
  plugins: [
    new ModuleFederationPlugin({
      remotes: {
        mfe1: "mfe1"
      },
      shared: ["@angular/core", "@angular/common", "@angular/router"]
    }),
    new AotPlugin({
      skipCodeGeneration: false,
      tsConfigPath: "./projects/shell/tsconfig.app.json",
      directTemplateLoading: true,
      entryModule: path.resolve(
        __dirname,
        "./projects/shell/src/app/app.module#AppModule"
      )
    }),
```

This is what we need in our webpack.config file


File Edit Selection View Go Run Terminal Help
webpack.config.js - example - Visual Studio Code
app.routes.ts
webpack.config.js
index.html

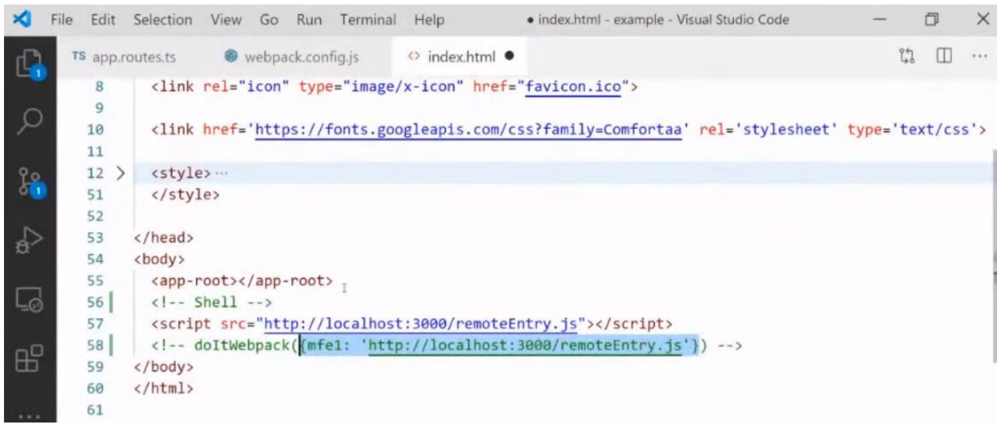
```
    rules: [
      { test: /\.ts$/, loader: "@ngtools/webpack" }
    ]
  },
  plugins: [
    new ModuleFederationPlugin({
      name: "mfe1",
      library: { type: "var", name: "mfe1" },
      filename: "remoteEntry.js",
      exposes: {
        Component: './projects/mfe1/src/app/app.component.ts',
        Module: './projects/mfe1/src/app/flights/flights.module.ts'
      },
      shared: ["@angular/core", "@angular/common", "@angular/router"]
    }),
    new AotPlugin({
      skipCodeGeneration: false,
      tsConfigPath: "./projects/mfe1/tsconfig.app.json",
      directTemplateLoading: true,
      entryModule: path.resolve(
        __dirname,
```

We then do the same thing in the configuration for our MFEs


File Edit Selection View Go Run Terminal Help
index.html - example - Visual Studio Code
app.routes.ts
webpack.config.js
index.html

```
  <link rel="icon" type="image/x-icon" href="favicon.ico">

  <link href='https://fonts.googleapis.com/css?family=Comfortaa' rel='stylesheet' type='text/css'>

  <style>···
  </style>

</head>
<body>
  <app-root></app-root>
  <script src="http://localhost:3000/remoteEntry.js"></script>
</body>
</html>
```

We need to load the remoteEntry.js in our index.html file as above


File Edit Selection View Go Run Terminal Help
● index.html - example - Visual Studio Code
app.routes.ts
webpack.config.js
index.html

```
  <link rel="icon" type="image/x-icon" href="favicon.ico">

  <link href='https://fonts.googleapis.com/css?family=Comfortaa' rel='stylesheet' type='text/css'>

  <style>···
  </style>

</head>
<body>
  <app-root></app-root>
  <!-- Shell -->
  <script src="http://localhost:3000/remoteEntry.js"></script>
  <!-- doItWebpack({mfe1: 'http://localhost:3000/remoteEntry.js'}) -->
</body>
</html>
```

3: When can we have it?

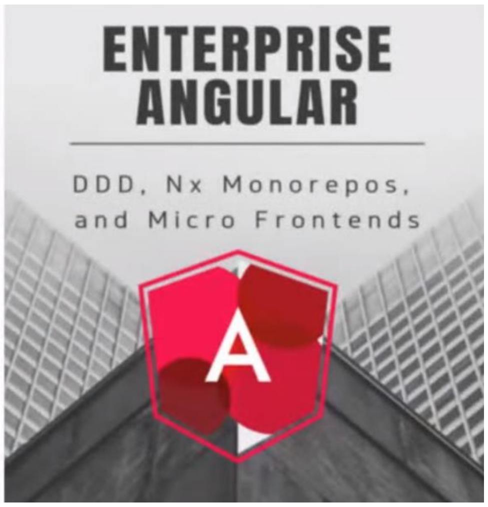
Well ...

- Webpack 5 is currently beta
- Shown examples: PoC w/ custom webpack conf + patched CLI lib
- CLI: Not before version 11 (fall 2020)
- Squeeze federation config into CLI's webpack config
- Custom Builder (e. g. ngx-build-plus)

Free eBook

Updated for Module Federation and Alternatives

ANGULARarchitects.io/book



Conclusion

Main Purpose of µFrontends: Scaling Teams

Federation: Import From Other App

Sharing Libs

Take Care of Conflicts

Be like Bonnie and think first!

Evaluate whether you need µFrontends

No: Majestic Monolith

Yes: Consider Module Federation

Contact and Downloads

[web] **ANGULAR***architects.io*

[twitter] ManfredSteyer

Slides & Examples

Remote Company Workshops and Consulting