Microfrontends com Vite + vite-plugin-federation

Introduction, How to use and some tips!

About me

Jhonatan Morais

Passionate about creating solutions and connecting with people.

Core techs:

- PHP, GoLang and Node
- React, Vue and TS
- Design System

Hobbies:

- GameDev, Games and Animes

More about in **jpmorais.com.br**

Agenda

1) What/Why/When to use Microfrontends

2) Overview of the example architecture

3) How vite-plugin-federation works

4) Hands-on: Exposing (remote) and Consuming (host)

6) Conclusion + ideas to learn more (1:30)

What/Why/When to use Micro Frontends
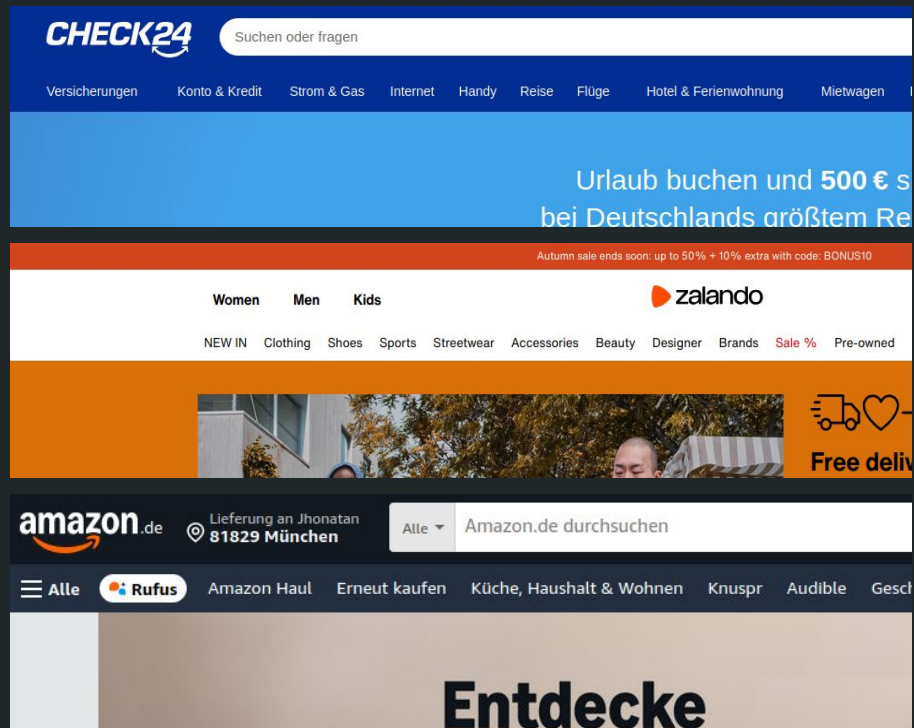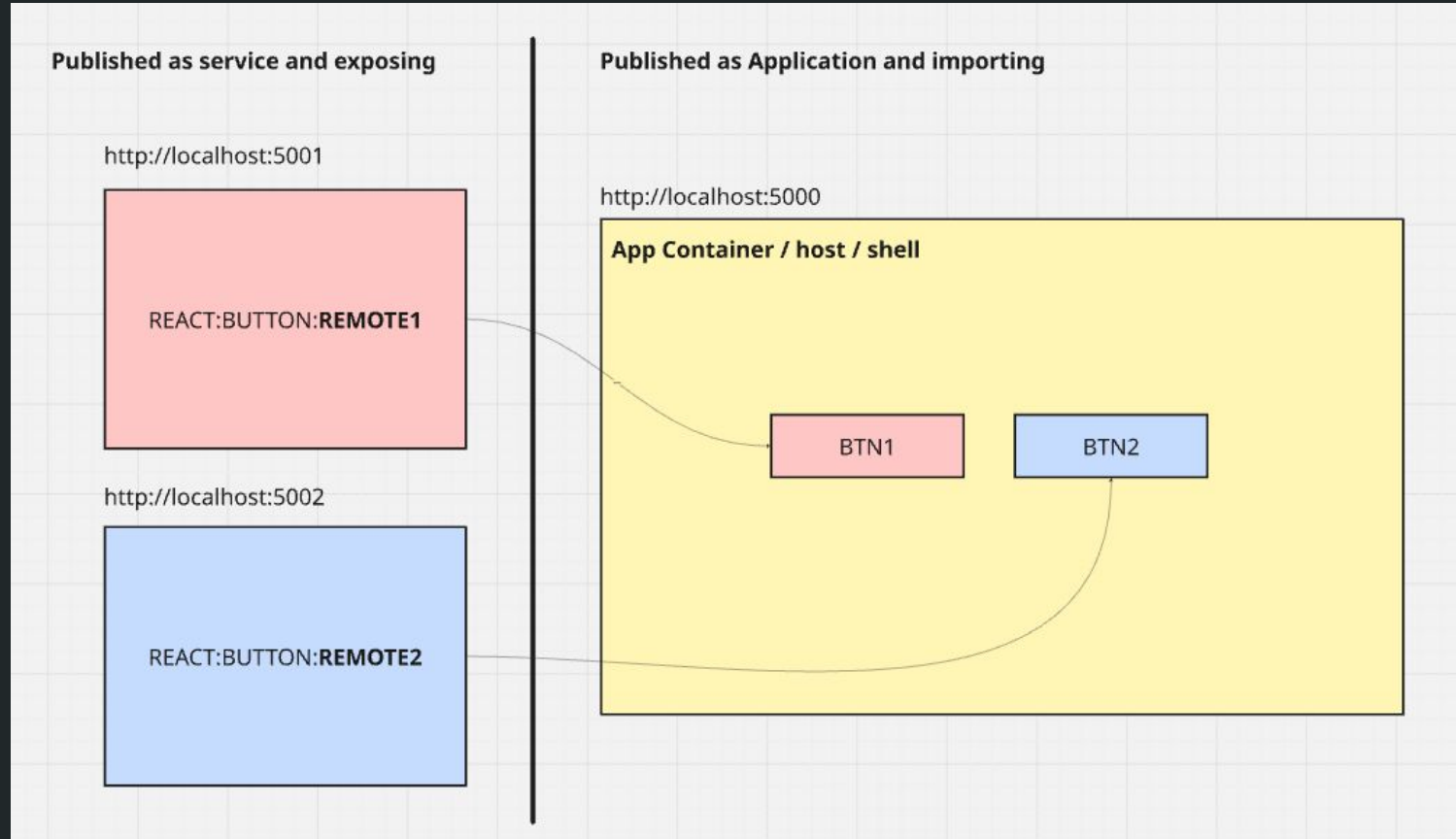
What/Why/When to use Micro Frontends

- Independent teams that need to deliver parts of the product autonomously (separate deploys).

- Need to isolate domains (Design System, Checkout, Search) with different evolution rhythms.

- Save time, money and rework

Overview of the example architecture

Published as service and exposing

http://localhost:5001

REACT:BUTTON:**REMOTE1**

http://localhost:5002

REACT:BUTTON:**REMOTE2**

Published as Application and importing

http://localhost:5000

App Container / host / shell

BTN1

BTN2

Overview of the example architecture - Configuration

Remotes

```
import { defineConfig } from 'vite'
import react from '@vitejs/plugin-react'
import federation from "@originjs/vite-plugin-federation";

// https://vitejs.dev/config/
export default defineConfig({
    plugins: [
        react(),
        federation({
            name: 'remote-app-1',
            filename: 'remoteEntry.js',
            // Modules to expose
            exposes: {
                './Button': './src/components/Button.tsx',
            },
            shared: ['react','react-dom']
        })
    ],
    build: {
        target: 'esnext'
    },
})
```

Host

```
import { defineConfig } from 'vite'
import react from '@vitejs/plugin-react'
import federation from "@originjs/vite-plugin-federation";

// https://vitejs.dev/config/
export default defineConfig({
  plugins: [
        react(),
        federation({
            name: 'host-app',
            remotes: {
                remoteApp1: "http://localhost:5001/assets/remoteEntry.js",
                remoteApp2: "http://localhost:5002/assets/remoteEntry.js",
            },
            shared: ['react','react-dom']
        })]
})
```

Overview of the example architecture - "Deploy" and usage

Keynotes

- **vite preview** in the remote apps
- **pnpm** make it a lot easier
- **libs and css** management

Host

```
import MyButton1 from 'remoteApp1/Button'
import MyButton2 from 'remoteApp2/Button'

function App() {
  return (
      <div className="border-4 border-red-300 flex flex-col justify-center h-80">
          <div className='w-full text-center'>APP CONTAINTER</div>
           <div className='flex h-full w-full justify-center items-center gap-3'>

              <div className='basis-1/3 border-purple-700 border-2'>
                  <div className='text-center w-full'>APP1 - React</div>
                  <div className='my-3 mx--2'>APP1 CONTENT ...</div>
                  <MyButton1 id="my-host-custom-css-to-remote-1"/>
              </div>

              <div className='basis-1/3 border-cyan-700 border-2'>
                  <div className='text-center w-full' >APP2 - React</div>
                  <div className='my-3 mx--2'>APP2 CONTENT ... </div>
                  <MyButton2/>
              </div>

          </div>
      </div>
  )
}

export default App
```
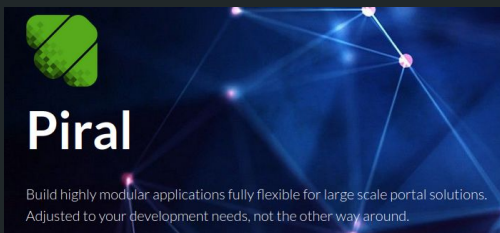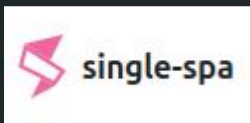
8
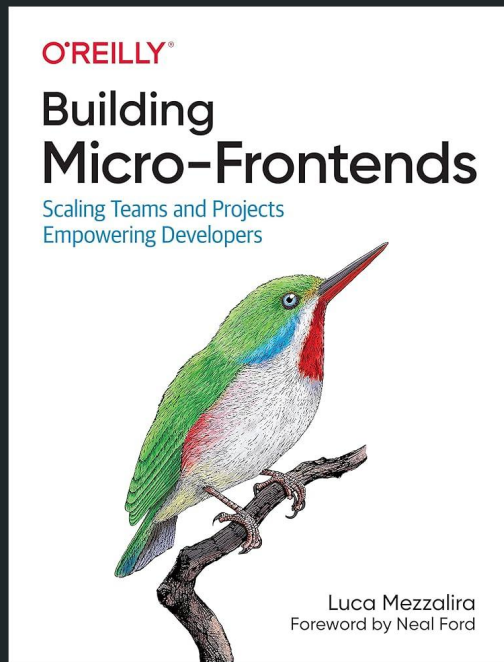
Hands-on: Exposing (remote) and Consuming (host)

Coding time

How to learn more?

talk code: https://github.com/getJv/getjv-micro-fe-react

Tools and links:
- Legacy: https://single-spa.js.org/
- OnGoing: https://piral.io/
- Tendency: https://luigi-project.io/

Plugin Repo samples:
https://github.com/originjs/vite-plugin-federation

Examples	Host	Remote
basic-host-remote	rollup + esm	rollup + esm
react-in-vue	vite + esm	vite + esm
simple-react-esm	rollup + esm	rollup + esm
simple-react-systemjs	rollup + systemjs	rollup + systemjs
simple-react-webpack	rollup + systemjs	webpack + systemjs
vue2-demo	vite + esm	vite + esm
vue3-advanced-demo	vite + esm vue-router / pinia	vite + esm vue-router / pinia
vue3-demo-esm	vite + esm	vite + esm
vue3-demo-systemjs	vite + systemjs	vite + systemjs
vue3-demo-webpack-esm-esm	vite/webpack + esm	vite/webpack + esm
vue3-demo-webpack-esm-var	vite + esm	webpack + var
vue3-demo-webpack-systemjs	vite + systemjs	webpack + systemjs
react-vite	vite + react	vite + react

O'REILLY®
Building Micro-Frontends
Scaling Teams and Projects
Empowering Developers
Luca Mezzalira
Foreword by Neal Ford

single-spa

Piral
Build highly modular applications fully flexible for large scale portal solutions.
Adjusted to your development needs, not the other way around.

Luigi
The Enterprise-Ready Micro Frontend Framework

Thank you!

Follow me!