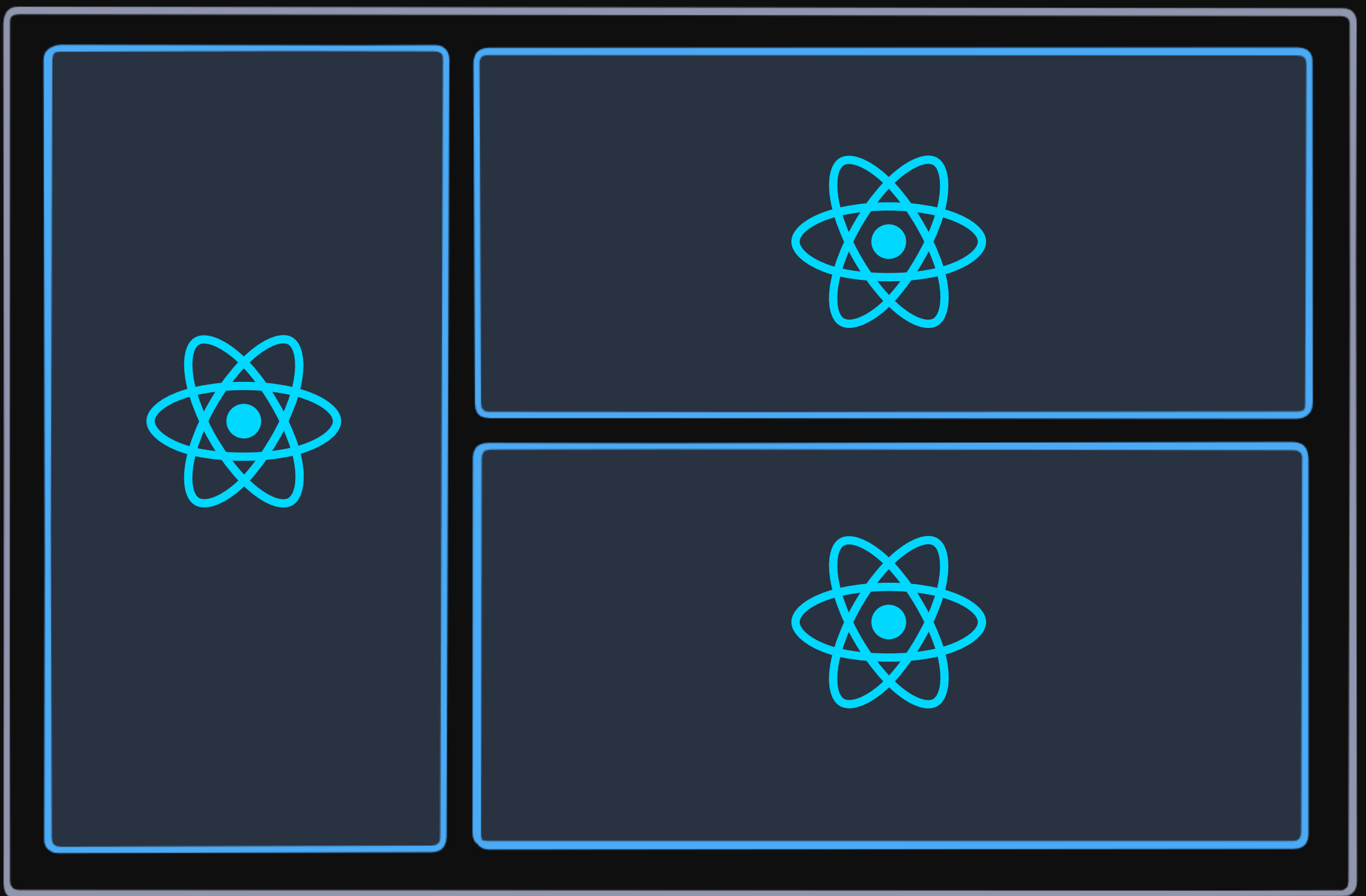


Desvendando Arquitetura de **MICRO-FRONTENDS**



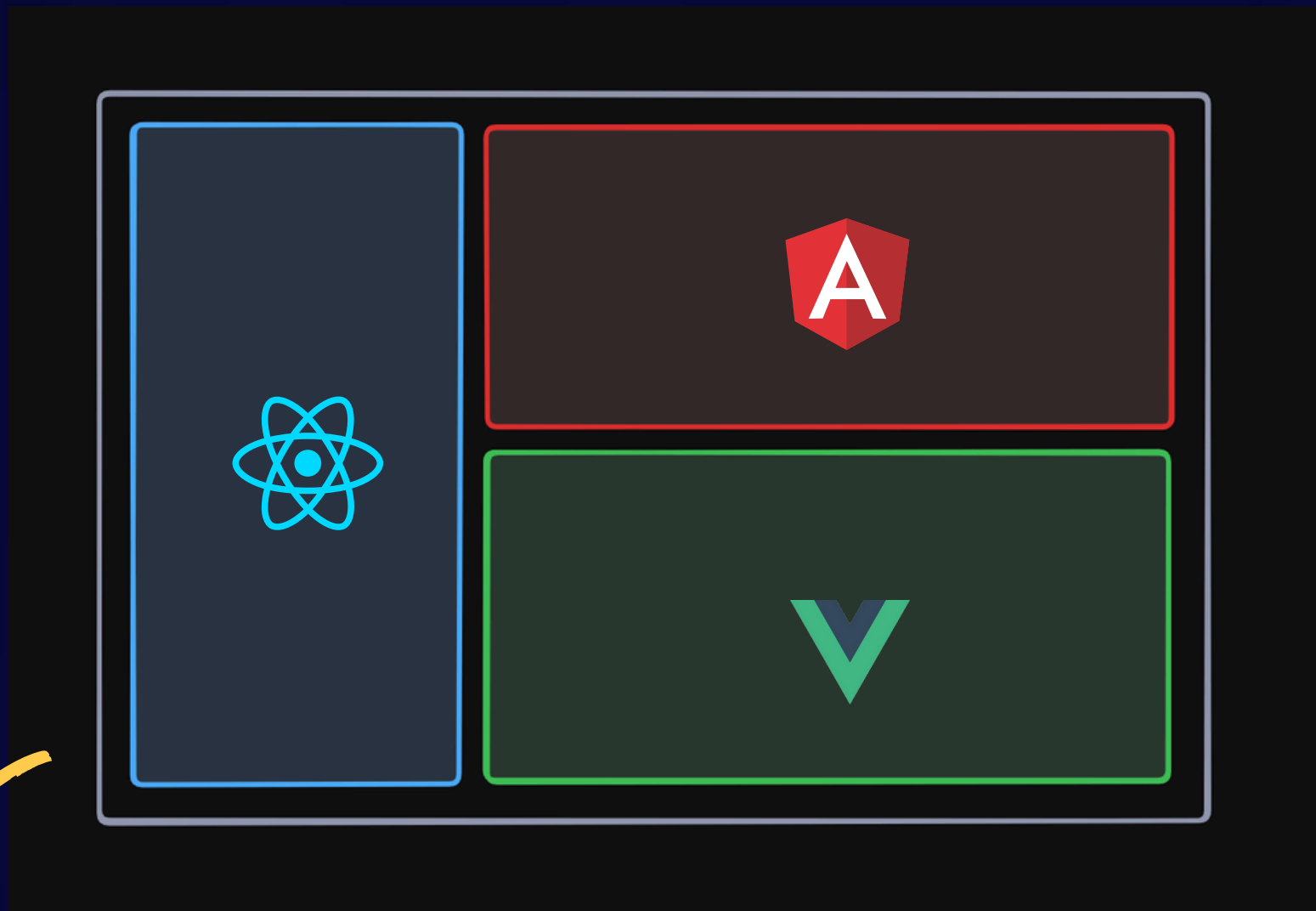
Exemplo com **React.js**



Isaac Gomes

TS

o que é **MICRO-FRONTENDS** ?



***Micro-frontends** dividem uma aplicação front-end em **partes menores, autônomas e independentes**, desenvolvidas e mantidas por equipes diferentes, permitindo o uso de **tecnologias variadas**.*



Isaac Gomes

TS

o que é **MICRO-FRONTENDS** ?



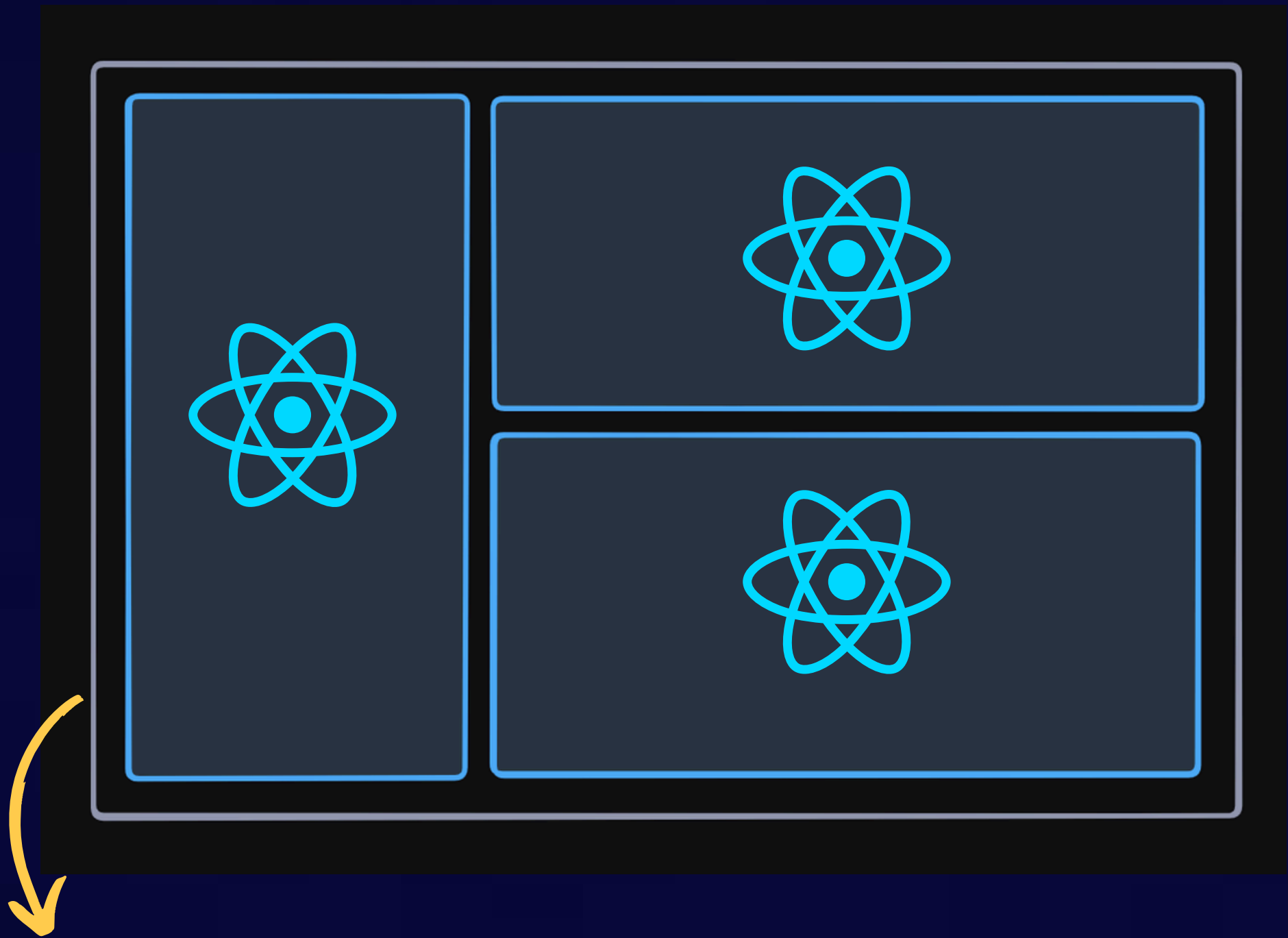
*Cada **micro-frontends** funciona como um **módulo separado**, integrado na aplicação final, promovendo escalabilidade e flexibilidade, mas exigindo cuidado com a integração e performance.*



Isaac Gomes

TS

Tipos de **MICRO-FRONTENDS**



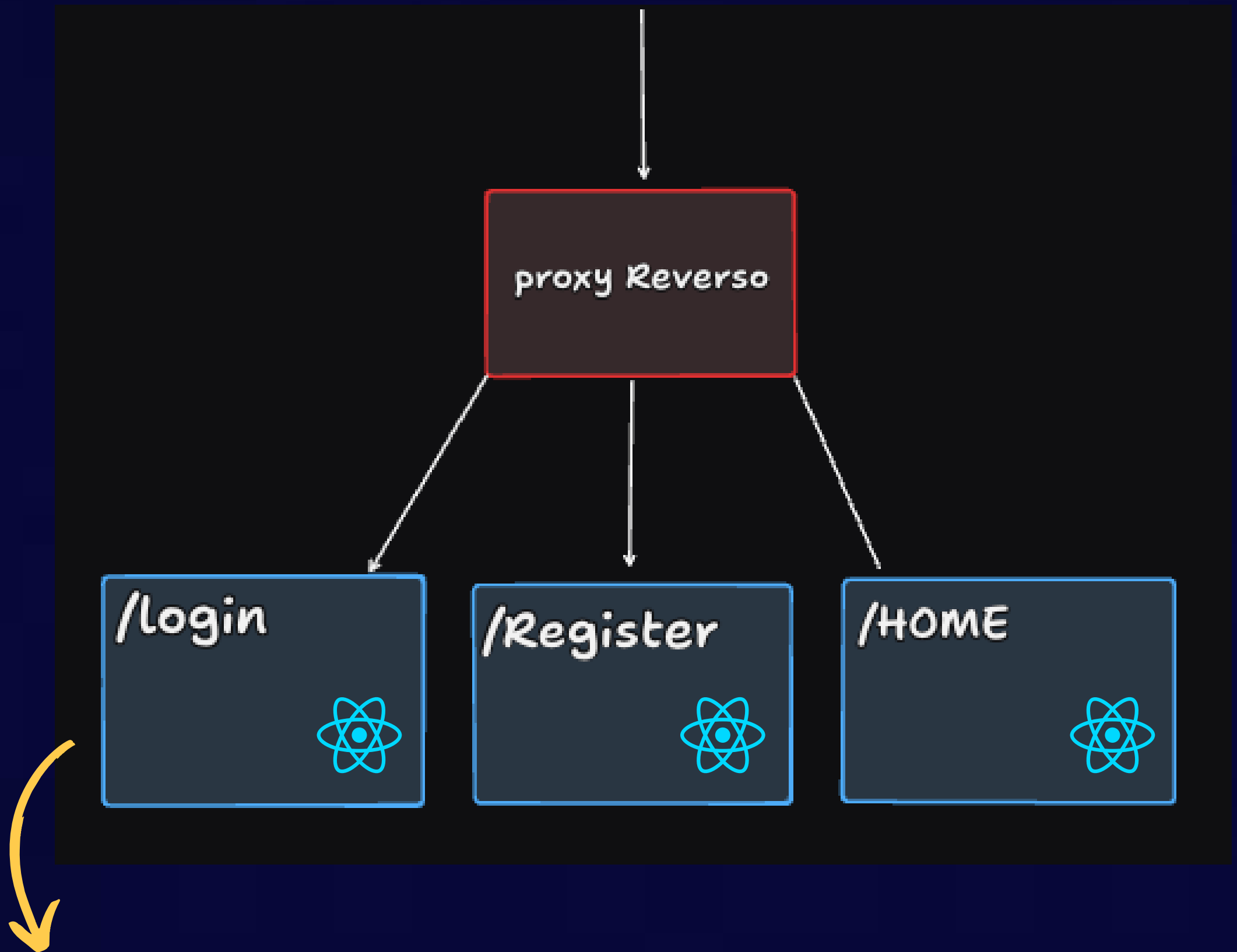
Parcel-Based MFE é uma abordagem de **micro-frontends** onde cada parte da aplicação é representada como um "**parcel**" ou **módulo** que pode ser carregado e renderizado de forma **independente** em qualquer ponto da **aplicação**



Isaac Gomes

TS

Tipos de **MICRO-FRONTENDS**



Route-Based MFE é uma abordagem de **micro-frontends** onde cada micro-frontend é associado a uma **rota específica** dentro da aplicação



Isaac Gomes

TS

como funciona **MICRO-FRONTENDS** com outras arquiteturas que vimos aqui?



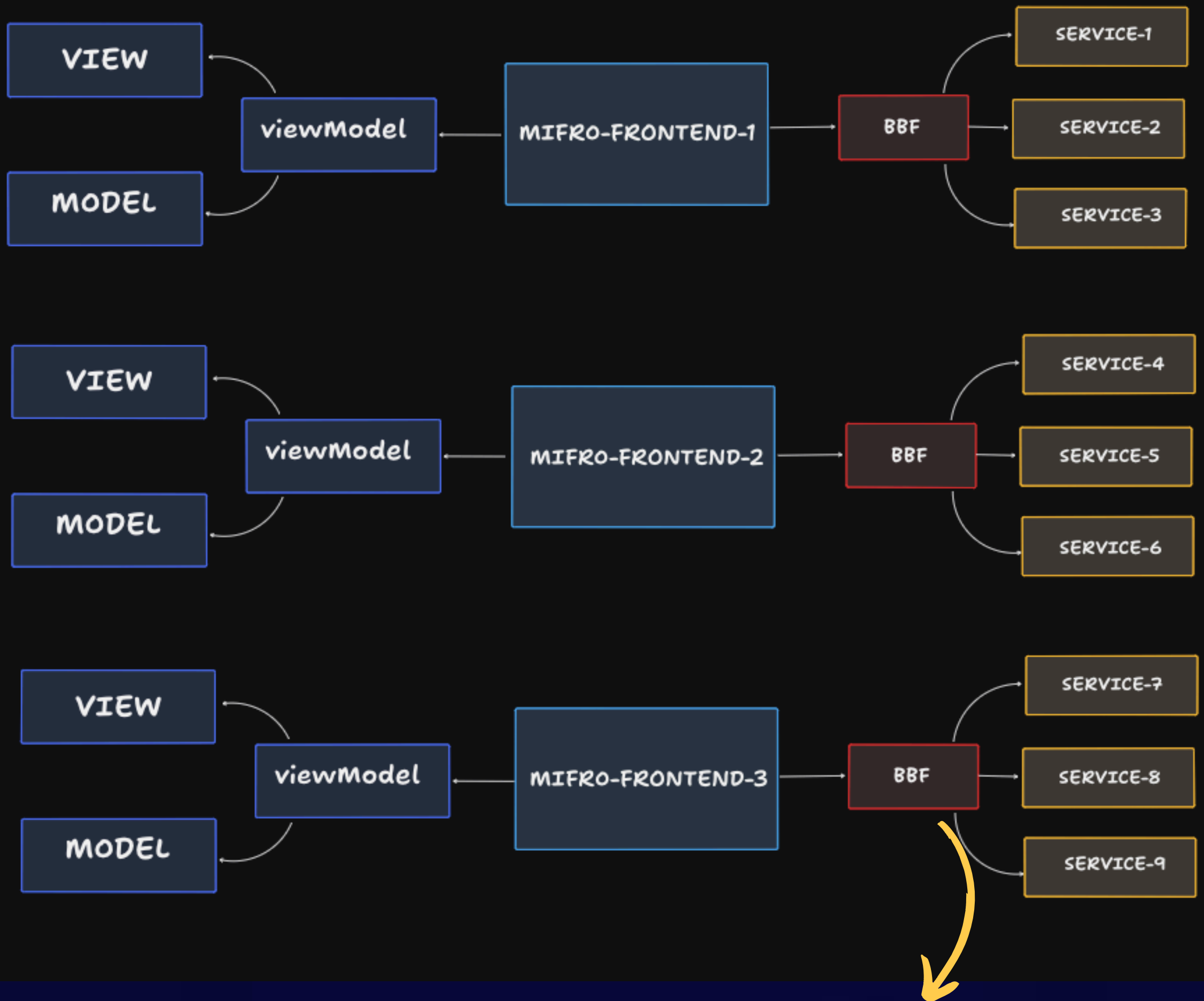
A nível de recurso podemos aplicar **MVVM** criando uma separação de (**view**, **model** e **viewModel**)



Isaac Gomes



como funciona **MICRO-FRONTENDS** com outras arquiteturas que vimos aqui?



Cada **MFE** pode ter seu **BFF**



Isaac Gomes



como funciona **MICRO-FRONTENDS** com outras arquiteturas que vimos aqui?



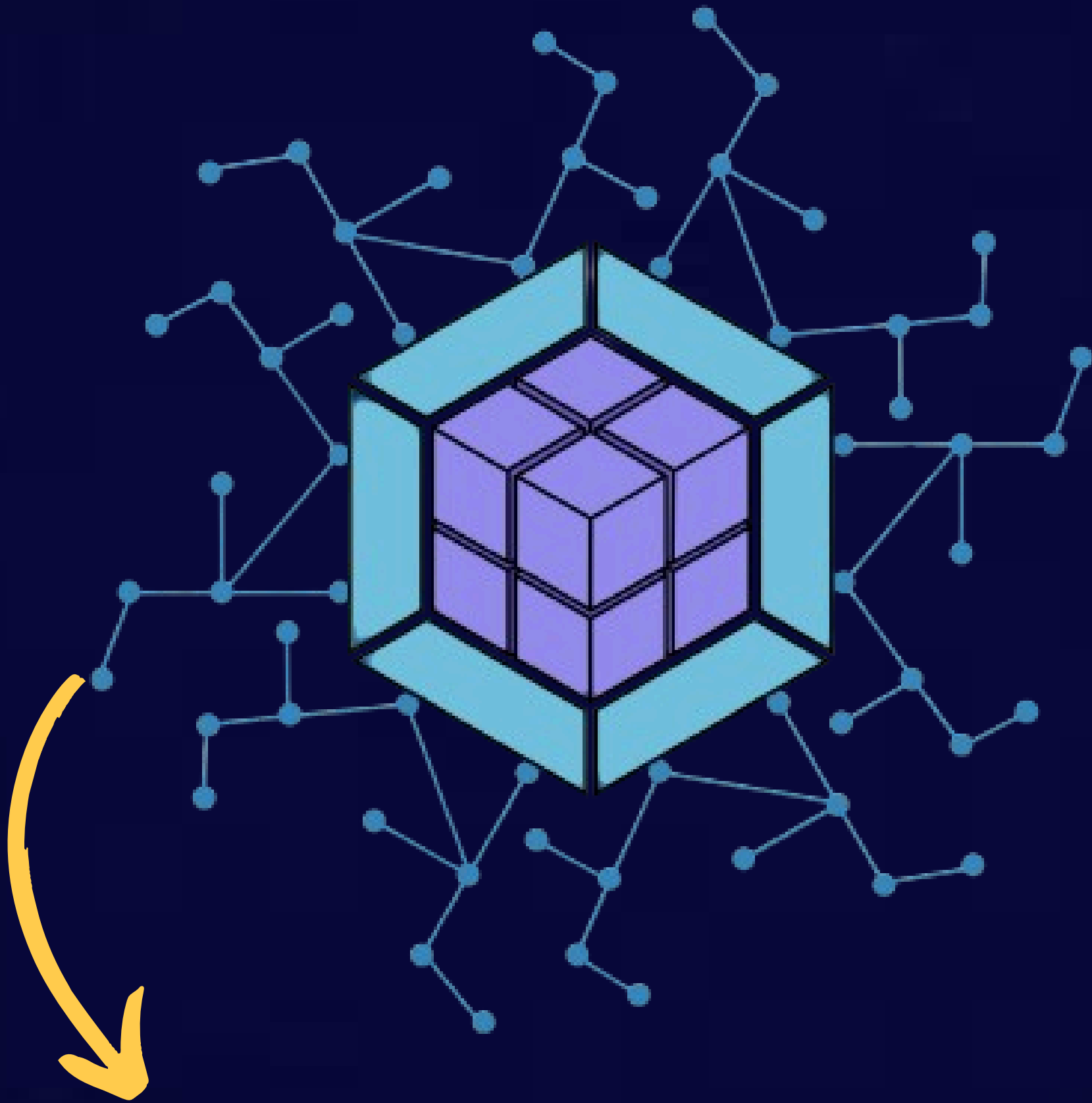
esses servicos poderiam usar
clean arch ou **ports and adapter**



Isaac Gomes



como criar **MICRO-FRONTENDS?**



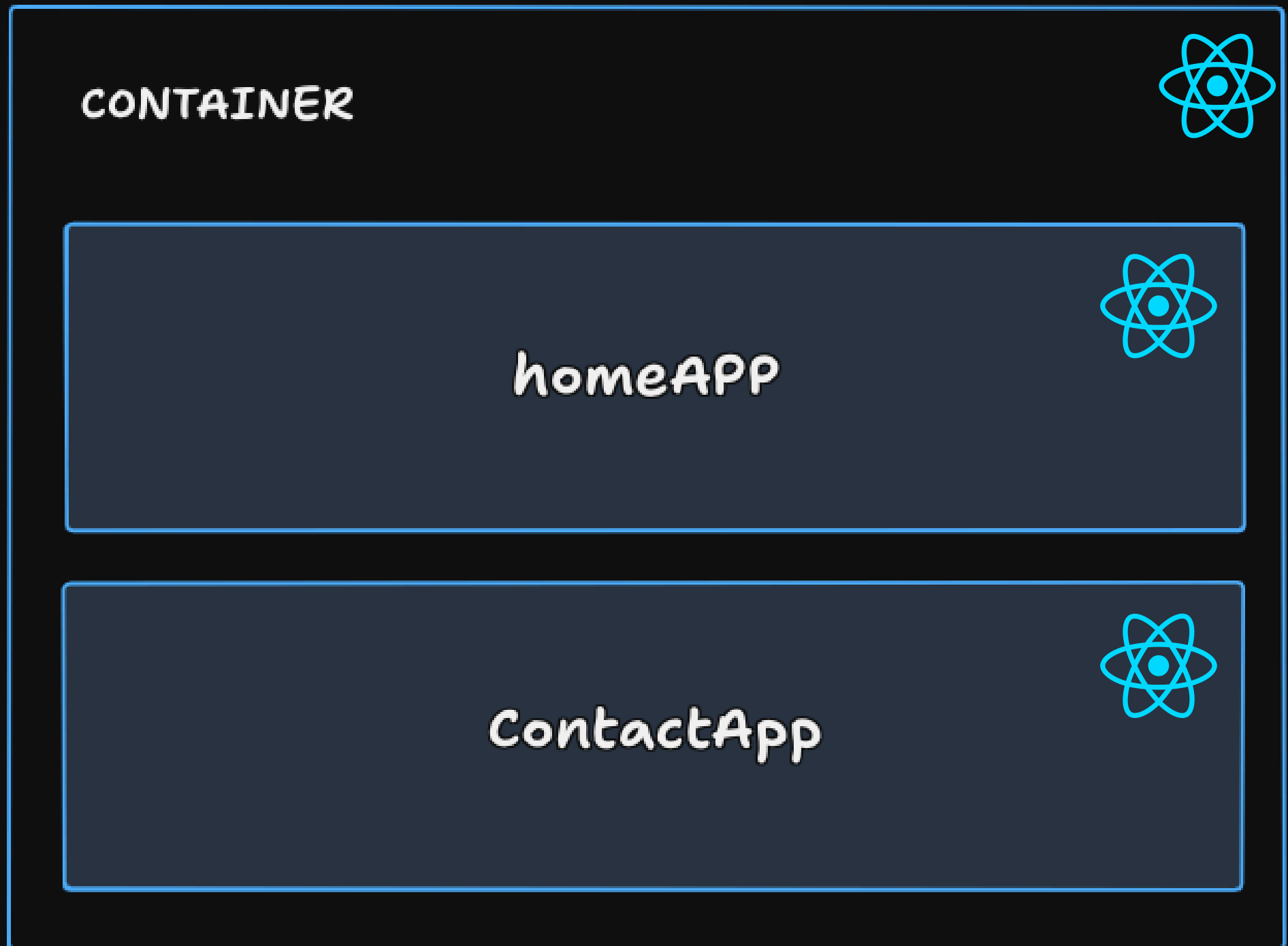
para exemplo usaremos **webpack**
(**Module Federation**)



Isaac Gomes

TS

como criar **MICRO-FRONTENDS**?



para exemplos vamos criar **3 APPs**
uma **home**, **contact** e um **Container**



Isaac Gomes

TS

como criar **MICRO-FRONTENDS**?

```
new ModuleFederationPlugin({  
  name: 'ContactApp',  
  filename: 'remoteEntry.js',  
  exposes: {  
    './ContactApp': './src/Contact  
  },  
})
```



criamos um projeto com **webpack** e agora vamos configurar o **ModuleFederation** dizendo qual vai ser o nome e o que vamos expor



Isaac Gomes

TS

como criar **MICRO-FRONTENDS?**

```
const Contact = () => {  
  return (  
    <Card>  
        
      <CardBody>  
        <CardTitle tag="h5">John Doe</CardTitle>  
        <CardSubtitle className="mb-2 text-muted" tag="h6">  
          Software Engineer  
        </CardSubtitle>  
        <CardText>  
          Email: john.doe@example.com  
          <br />  
          Phone: (123) 456-7890  
        </CardText>  
        <Button color="primary">Contact</Button>  
      </CardBody>  
    </Card>  
  )  
}
```

Aqui esta o nosso componente de **Contact**



Isaac Gomes

TS

como criar **MICRO-FRONTENDS**?

```
new ModuleFederationPlugin({  
  name: 'HomeApp',  
  filename: 'remoteEntry.js',  
  exposes: {  
    './HomeApp': './src/Home.  
  },  
})
```



agora vamos configurar a nossa **HomeApp**



Isaac Gomes

TS

como criar **MICRO-FRONTENDS?**

```
const Home = () => {  
  return (  
    <Nav tabs>  
      <NavItem>  
        <NavLink href="#" active>  
          Link  
        </NavLink>  
      </NavItem>  
      <NavItem>  
        <NavLink href="#">Link</NavLink>  
      </NavItem>  
      <NavItem>  
        <NavLink href="#">Another Link</NavLink>  
      </NavItem>  
      <NavItem>  
        <NavLink disabled href="#">  
          Disabled Link  
        </NavLink>  
      </NavItem>  
    </Nav>  
  )  
}
```

Aqui esta o nosso componente de **Home**



Isaac Gomes

TS

como criar **MICRO-FRONTENDS**?

```
new ModuleFederationPlugin({  
  name: 'App',  
  remotes: {  
    HomeApp: 'HomeApp@http://localhost:9002/remoteEntry.js',  
    ContactApp: 'ContactApp@http://localhost:9003/remoteEntry.js',  
  },  
}),
```

agora vamos criar nosso **container** e no **ModuleFederation** dele vamos declarar o que vão acessar "**remotes**"



Isaac Gomes

TS

como criar **MICRO-FRONTENDS**?

```
import HomePage from 'HomeApp/HomeApp'  
import ContactApp from 'ContactApp/ContactApp'  
  
const App = () => {  
  return (  
    <div>  
      <p>CONTAINER</p>  
      <HomePage />  
      <ContactApp />  
    </div>  
  )  
}
```

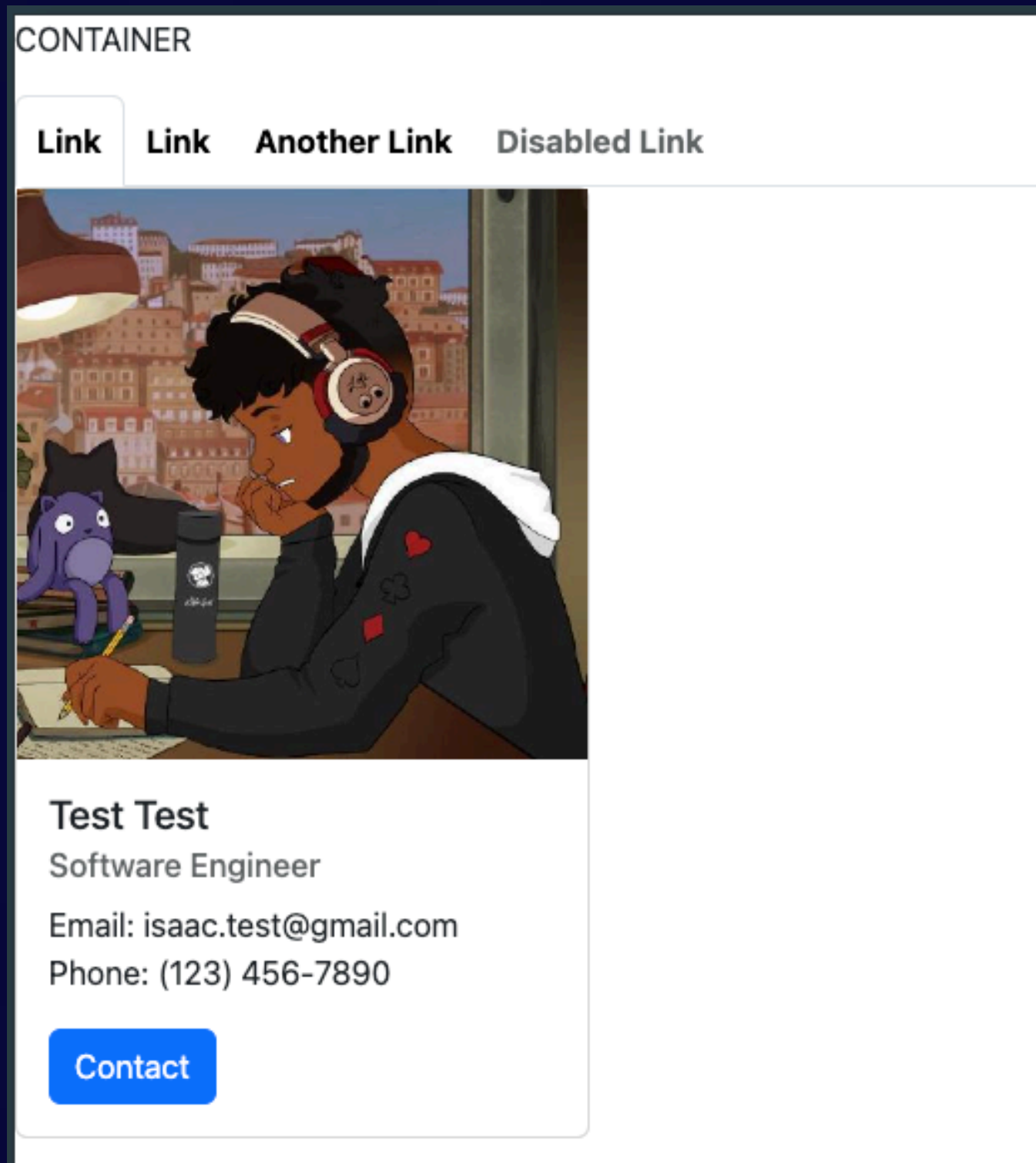
agora é só usar os **MF criados**



Isaac Gomes

TS

como criar **MICRO-FRONTENDS**?



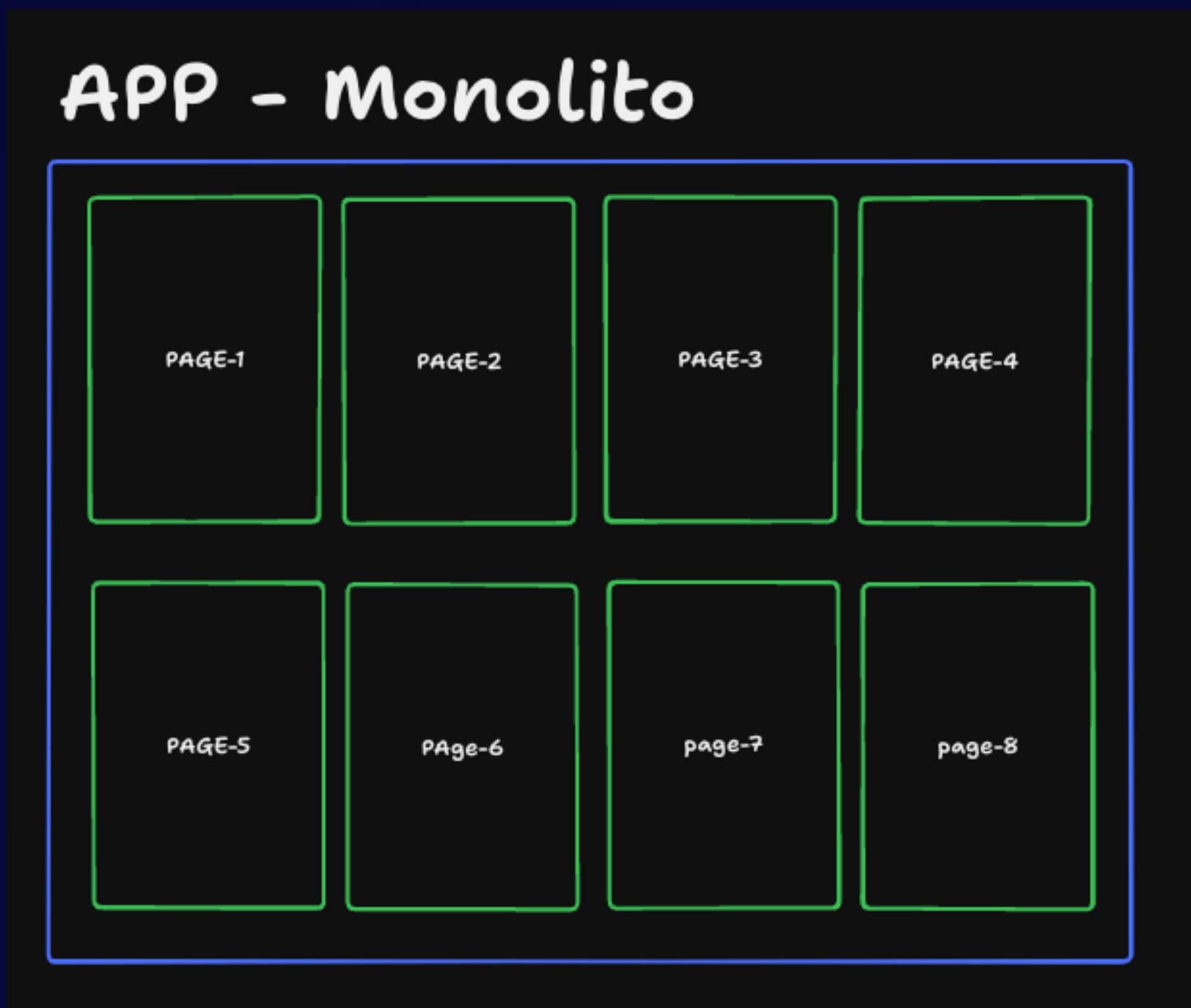
agora temos **2 MF**
rodando nessa tela



Isaac Gomes

TS

MONOLITO / MICRO-FRONTENDS



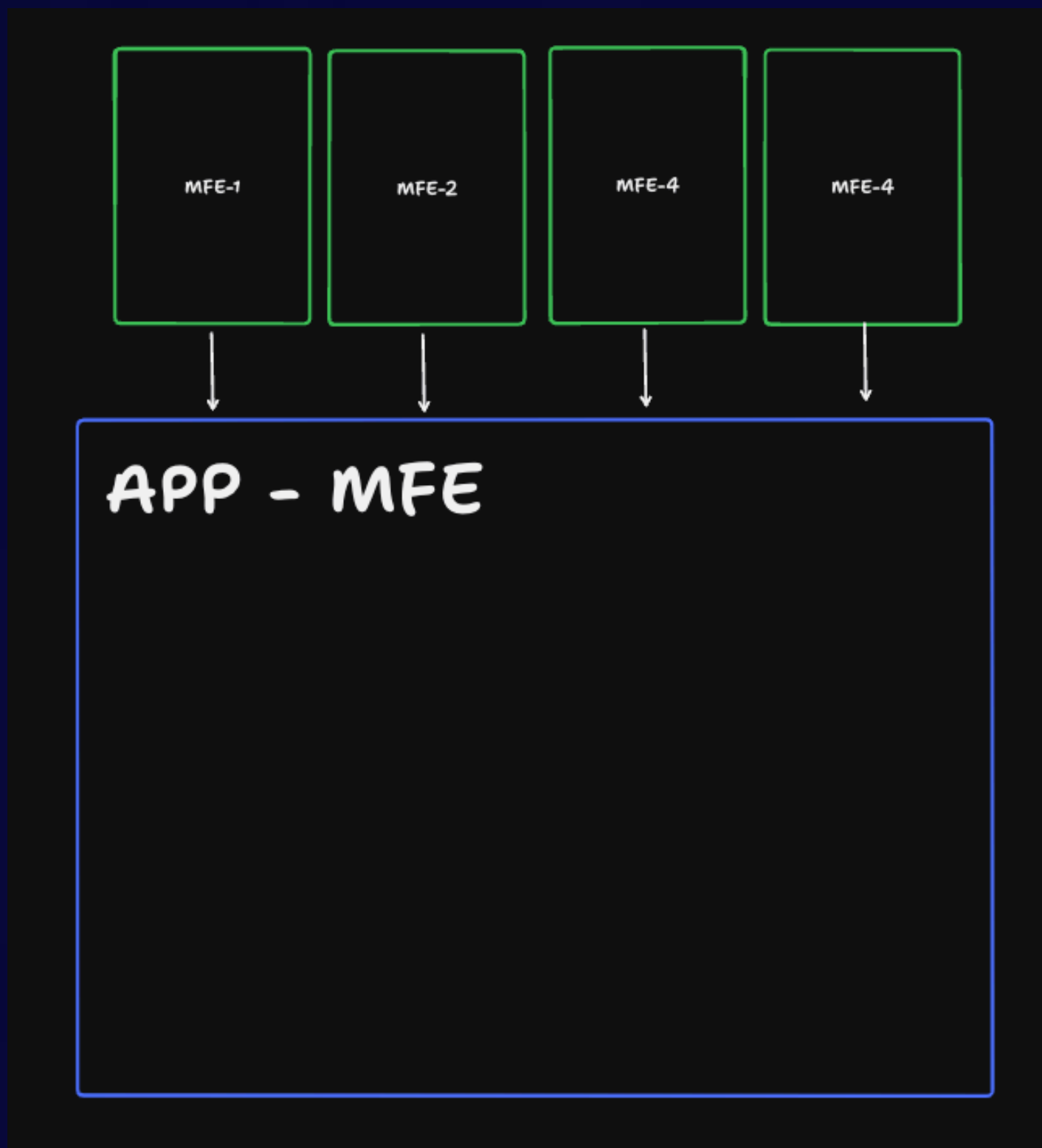
então aqui podemos ver a diferença de quando temos um unico **monolito frontend** que tem uma **Codebase única**



Isaac Gomes

TS

MONOLITO / MICRO-FRONTENDS



e um projeto com **MFE** com essa
divisão da codebase mais
modular e independente



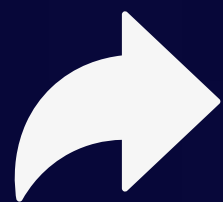
Isaac Gomes

TS

Gostou?



Curta



Compartilhe



Salve



Isaac Gomes

TS