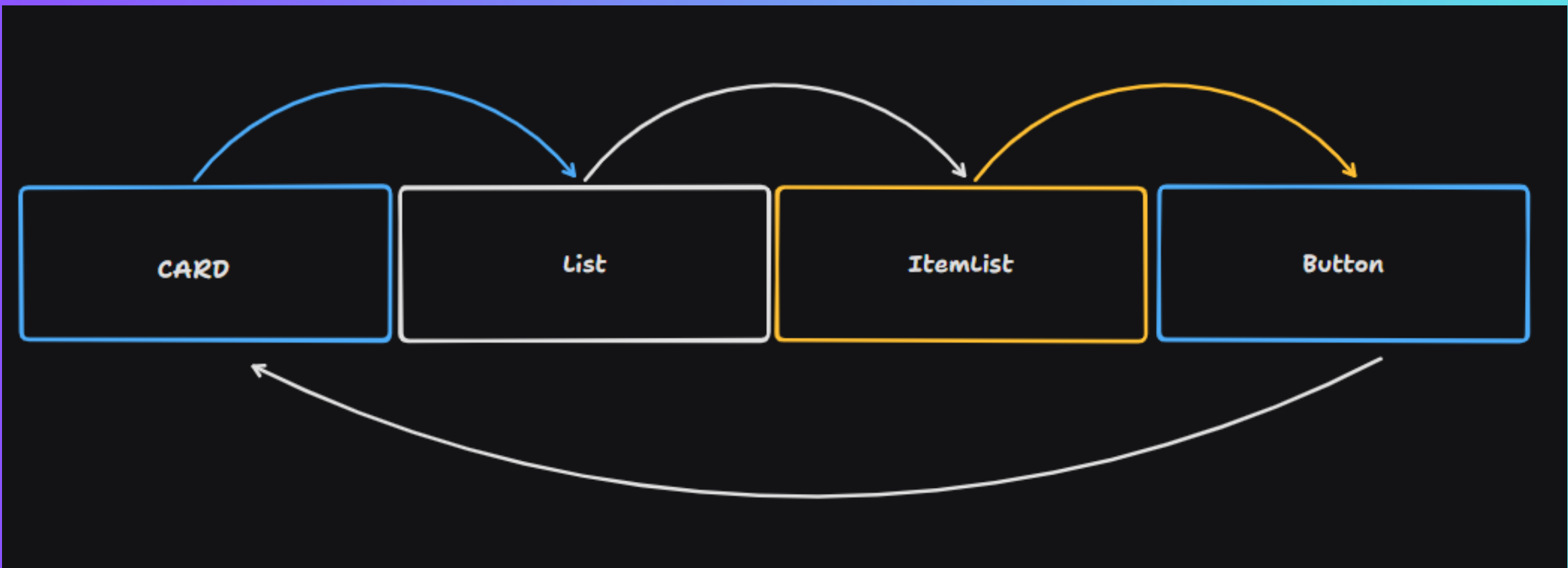
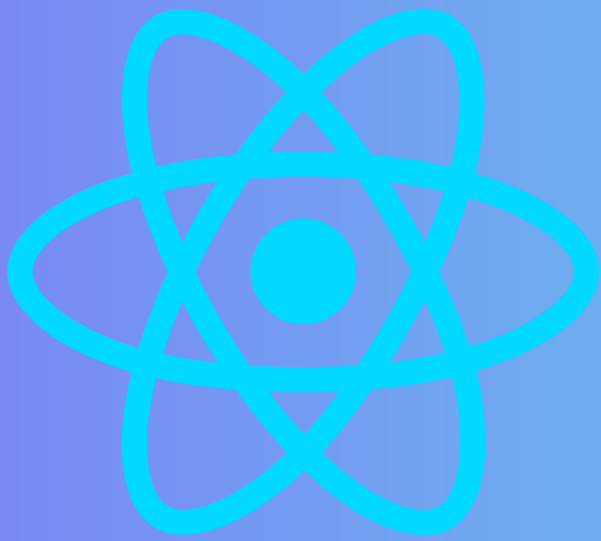




a melhor maneira de lidar com Prop drilling

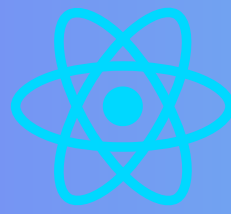




o que é Prop drilling ?

Prop drilling é um padrão em React onde os dados são passados de componentes pais para componentes filhos através das props, podendo resultar em código menos legível e manutenção mais difícil devido à necessidade de passar esses dados por vários níveis de componentes intermediários.





exemplo

```
Parent.ts

const Parent = () => {
  const [number, setNumber] = useState(42);
  return (
    <div>
      <h1>Parent Component</h1>
      <Child number={number} />
    </div>
  );
};

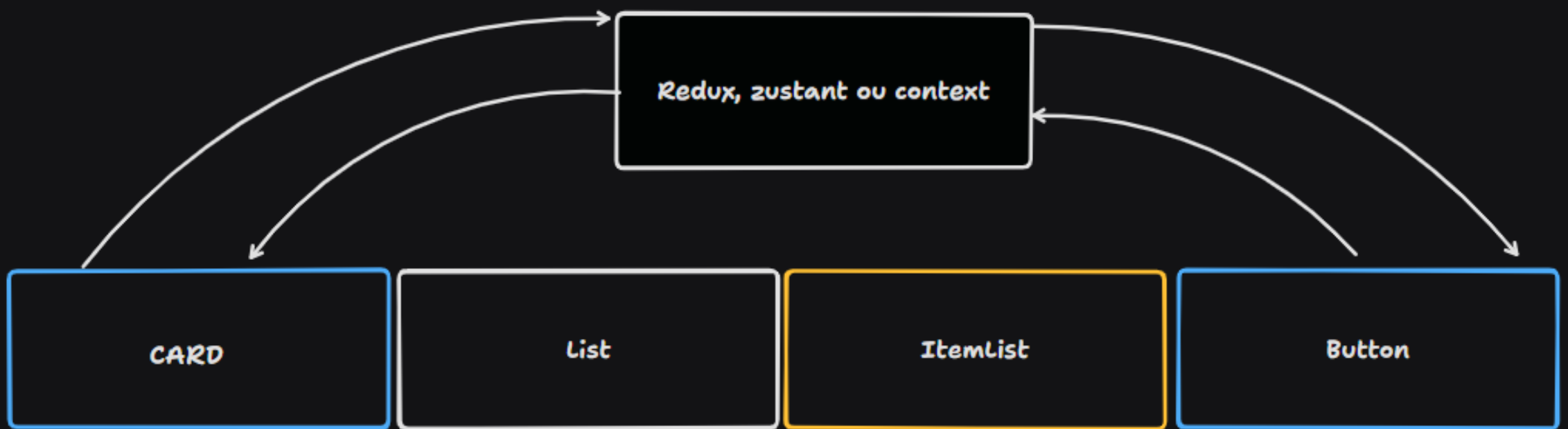
const Child = ({ number }) => {
  return (
    <div>
      <h2>Child Component</h2>
      <Grandchild number={number} />
    </div>
  );
};

const Grandchild = ({ number }) => {
  return (
    <div>
      <h3>Grandchild Component</h3>
      <p>The number passed down from Parent: {number}</p>
    </div>
  );
};
```





maneira comum de Resolver



um das soluções mais comuns são o uso de states globais usando redux, zustant ou context, assim, cortamos a trajetória e aproximando a propriedade do componente

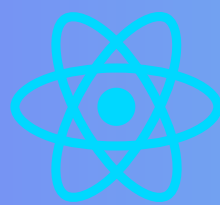




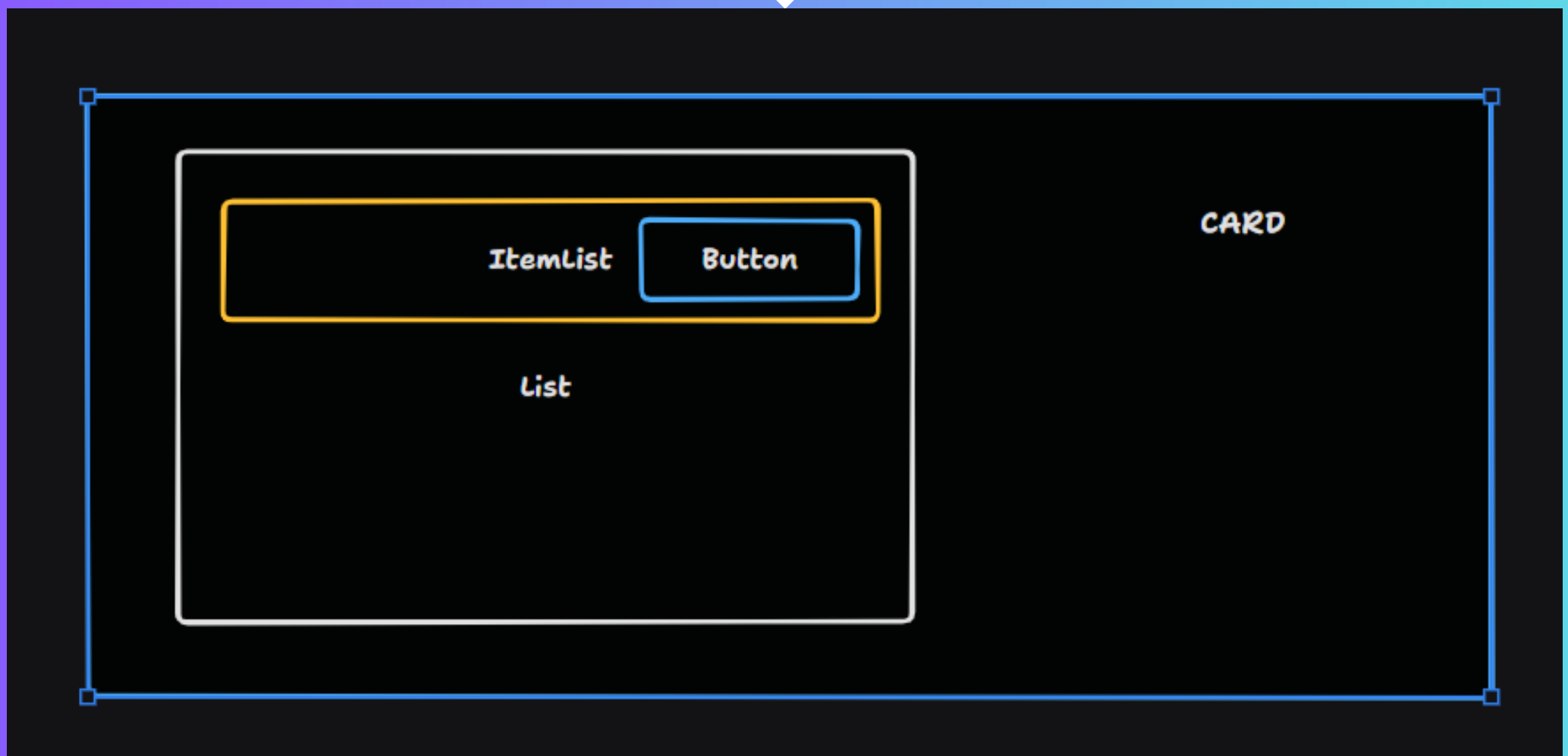
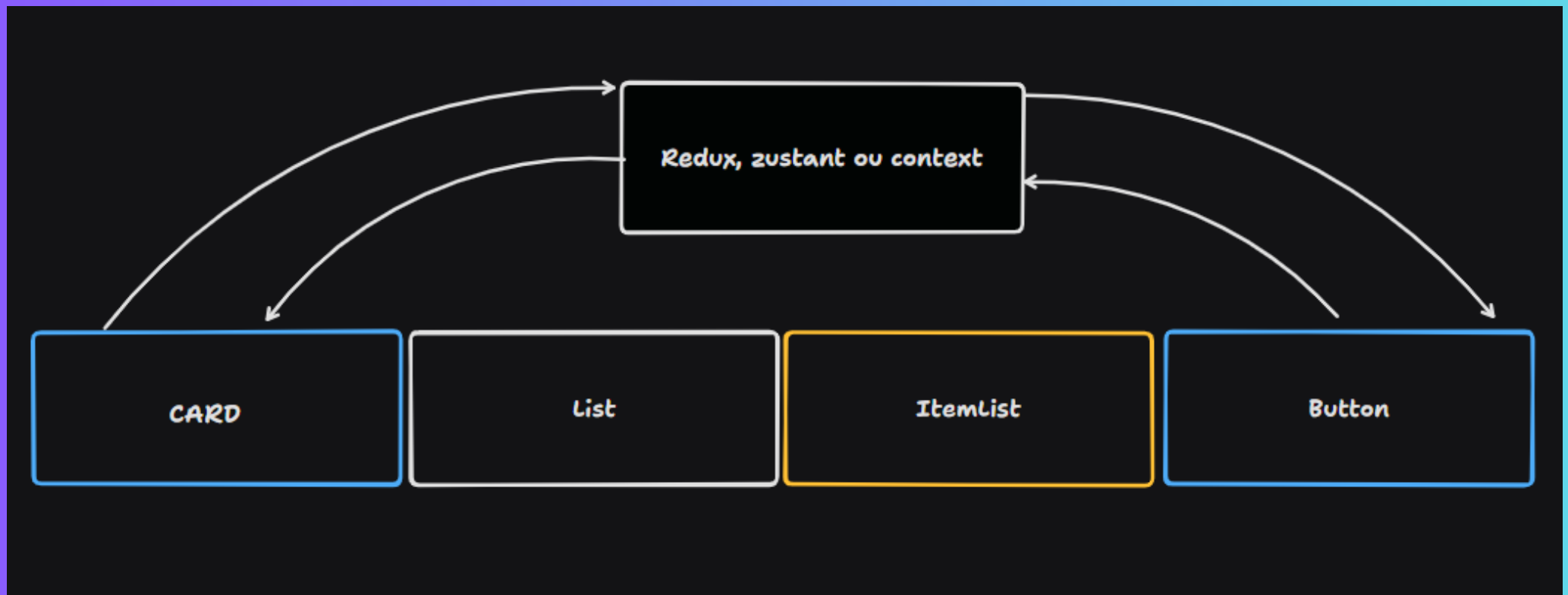
mas, será que essa é a melhor resolução?

**com essa resolução temos maior
verbosidade e menor clareza, quem já se
deparou com o redux mal implementado
para resolver esse tipo de problema sabe
a dor de cabeça que é...**





uma das maneiras de resolver é ao invés de aproximar a propriedade do componente



é aproximar o componente da propriedade usando padrões como "Composition", assim, trazendo menor complexidade e maior clareza





Exemplo

```
Parent.ts

const Parent = () => {
  const [number, setNumber] = useState(42);
  const handleClick = () => setNumber(prevNumber => prevNumber + 1);
  return (
    <div>
      <h1>Parent Component</h1>
      <Child number={number} handleClick={handleClick}>
        <Grandchild />
      </Child>
    </div>
  );
};

const Child = ({ number, handleClick, children }) => {
  return (
    <div>
      <h2>Child Component</h2>
      <p>The number passed down from Parent: {number}</p>
      <button onClick={handleClick}>Increment Number</button>
      {children}
    </div>
  );
};

const Grandchild = () => {
  return (
    <div><h3>Grandchild Component</h3></div>
  );
};
```

ao usar children aproximamos o
componente da propriedade





o que padrões como composition
pattern busca resolver?



pattern como composition buscam
resolver o problema "propcalypse"



o que é propcalypse?

A "propcalypse" é um termo usado na
comunidade de desenvolvimento React
para descrever uma situação na qual
um componente React recebe uma
grande quantidade de props
(propriedades), tornando-se difícil de
gerenciar e entender





como isso se relaciona com "prop drilling" já que o objetivo é tratar o "propcalypse"?

```
Modal.ts

export const Modal = {
  Modal,
  ModalContent,
  ModalTitle,
  ModalAction
}
```

quando quebramos as responsabilidades deixando de ter "um componente que usa componente que usa outro componente" com isso aproximamos os componentes das propriedades e assim dos seus usos





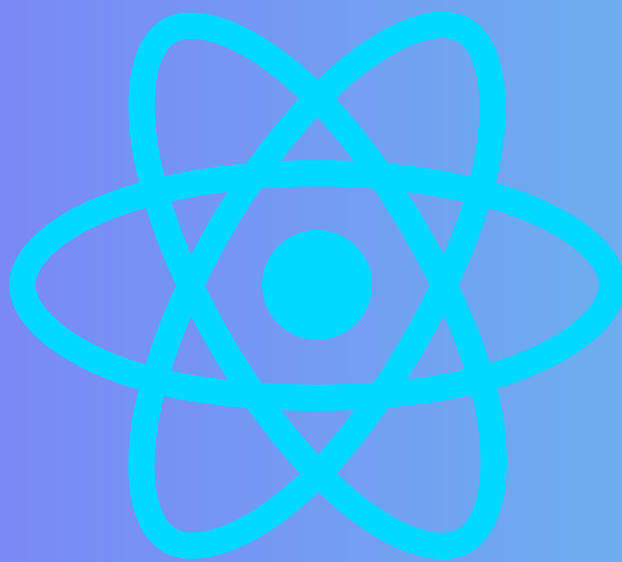
exemplo

```
Modal.ts

export const Modal = {
  Modal,
  ModalContent,
  ModalTitle,
  ModalAction
}
```

ao invés de termos um componente de modal com um "propcalypse" de opções de footer, header e button quebramos ele em uma composition e por consequencia quebramos o problema do Prop drilling que aconteceria exemplo no footer onde teríamos o Modal > FooterModal > Button pq aproximamos as props dos seus usos





**mas, será que alguma lib conhecida
usa esse pattern?**





shadcn-ui

```
shadcn.ts

import {
  Dialog,
  DialogContent,
  DialogDescription,
  DialogFooter,
  DialogHeader,
  DialogTitle,
  DialogTrigger,
} from "@components/ui/dialog"
```





chakra-ui

```
● ● ● chakra.ts

import {
  AlertDialog,
  AlertDialogBody,
  AlertDialogFooter,
  AlertDialogHeader,
  AlertDialogContent,
  AlertDialogOverlay,
  AlertDialogCloseButton,
} from '@chakra-ui/react'
```





next-ui

nextui.ts

```
import {  
  Modal,  
  ModalContent,  
  ModalHeader,  
  ModalBody,  
  ModalFooter  
} from "@nextui-org/react";
```



