

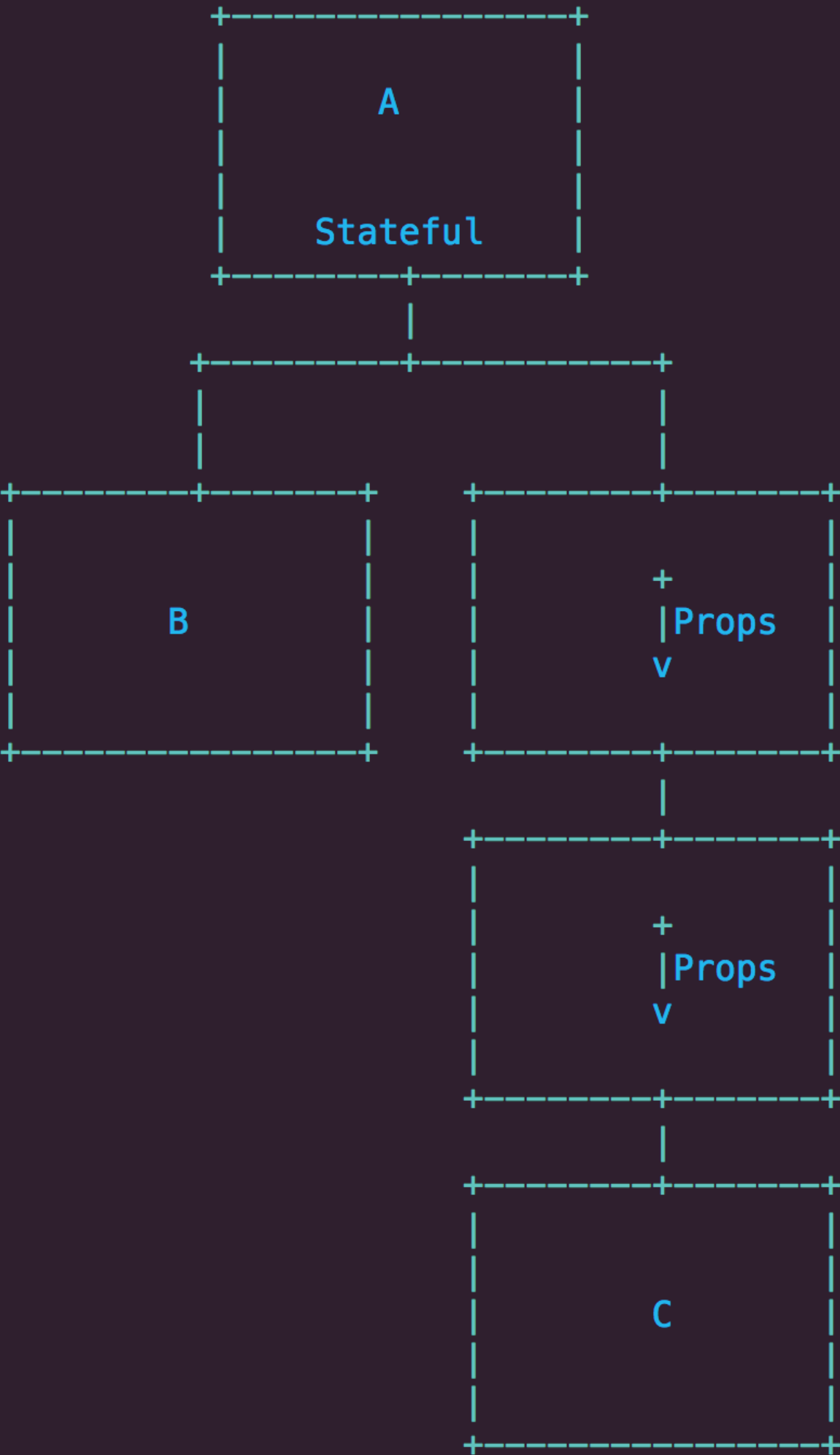
# KOMPOSITION & REACTS FLÖDE

**SAKER VI MÅSTE KUNNA**

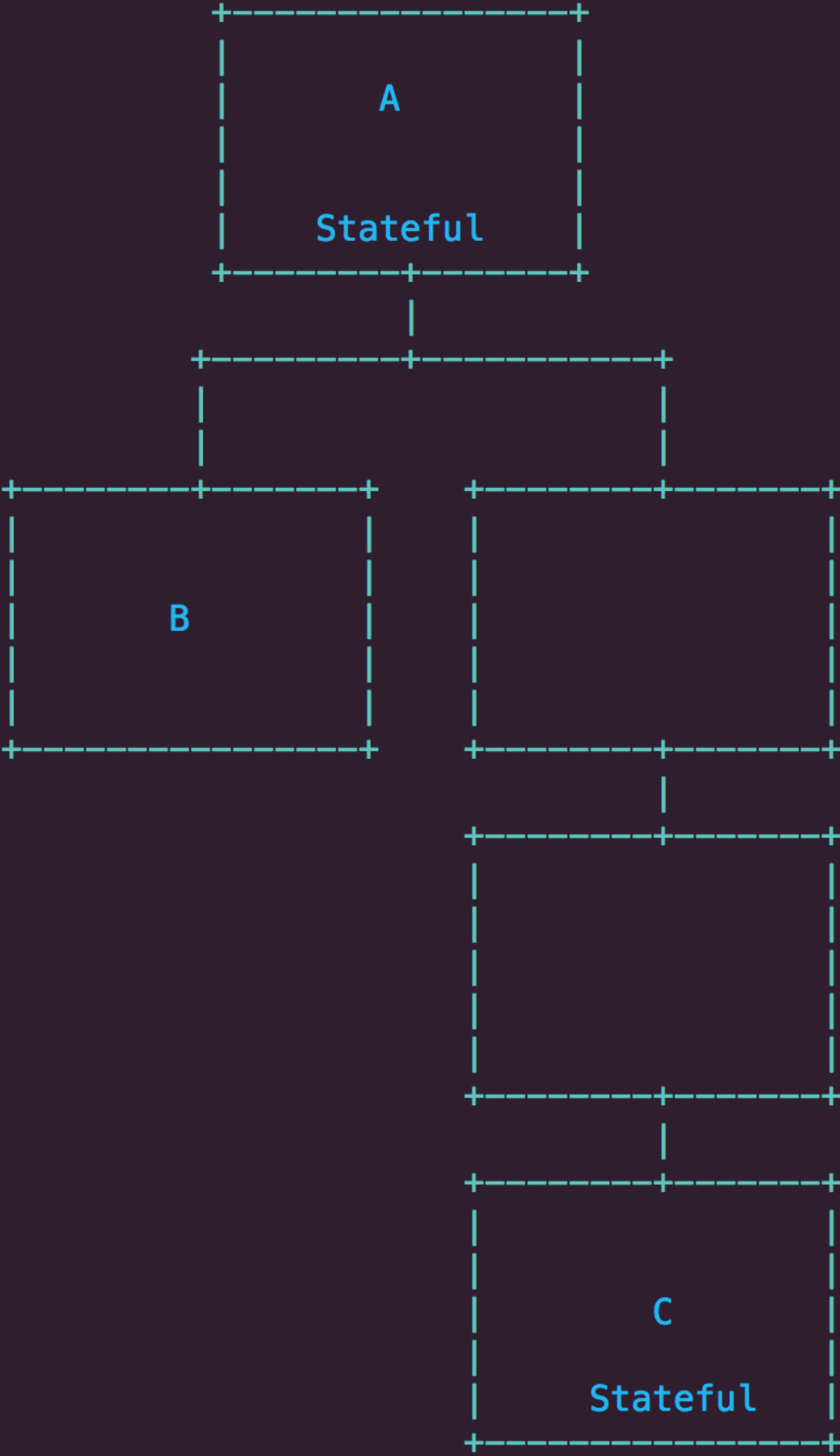
- Skapa `class` samt `function` komponenter
- Lyfta upp `state` och skicka ner det
- Skapa arrayer av komponenter: `map/filter`
- `Controlled Components`
- Skicka `state` upp 🤔

LYFT UPP STATE





B kan inte komma åt state från C  
Isf måste C först skicka till A





Men om vi måste skicka upp från en komponent

```
function Card(props) {  
  return(  
    <div key={props.id}>  
      { props.title}  
      <button  
        onClick={props.onClick}>  
        Rate 10  
      </button>  
    </div>  
  );  
}
```

## Skicka som argument

```
function Card(props) {  
  return(  
    <div key={props.id}>  
      { props.title}  
      <button  
        onClick={() => props.onClick(props.title)}>  
        Rate Title  
      </button>  
    </div>  
  );  
}
```

I föregående fall vill vi gärna skicka upp värdet

- Komponenten kan fortsätta vara stateless
- Värdet som skickas upp använd för att ändra state men tillhör inte själv state

```
function Card(props) {  
  function localClick(title) {  
    //Do more stuff, then call props.onClick  
    props.onClick(title)  
  }  
  return(  
    <button  
      onClick={localClick(props.title)}>  
      Rate Title  
    </button>  
  );  
}
```

COMPOSITION

Med klasser ärver vi

```
class App extends Component{  
  //I can do everything React.Component can do  
}
```

Detta är dock ändå gången vi ska använda arv i React

Resten ska göras med `component/function composition`

Istället för **is-a** så får vi **has a**

```
function Navbar(props) {  
  return(  
    <nav>  
      <NavbarItem title="Home" />  
      <NavbarItem title="About"s />  
    </nav>  
  );  
}
```

När vi använder React (och andra ramverk) så har vi två populära ord:

- **Higher Order Component (HOC)**

*a higher-order component is a function that takes a component and returns a new component.*

- **Higher Order Function**

*A function which takes a function as an argument and/or returns a function.*



```
import Button from './Button';
import makePrimary from './makePrimary';

function LoginForm(props) {
  const PrimaryButton = makePrimary(Button);
  return (<PrimaryButton title="Click me!" />);
}
```

En funktion som tar en existerande komponent och returnerar en ny komponent med nya egenskaper

Vi har flera olika sätt att komponera

```
function Button(props) {  
  
  return(  
    <button onClick={props.onClick} className={buttonType}>  
      {props.title}  
    </button>  
  );  
}
```

Vi har flera olika sätt att komponera

```
function Button(props) {  
  const buttonType = props.primary ? "btn btn-primary" : "btn";  
  return(  
    <button onClick={props.onClick} className={buttonType}>  
      {props.title}  
    </button>  
  );  
}
```

Vi har flera olika sätt att komponera

```
function Button(props) {  
  const buttonType = props.primary ? "btn btn-primary" : "btn";  
  return(  
    <button onClick={props.onClick} className={buttonType}>  
      {props.title}  
    </button>  
  );  
}  
<Button primary />
```