



SPA - Slow Pace Application

Anleitung zum Unglücklichsein

@merelyChristina & @merelyAnna

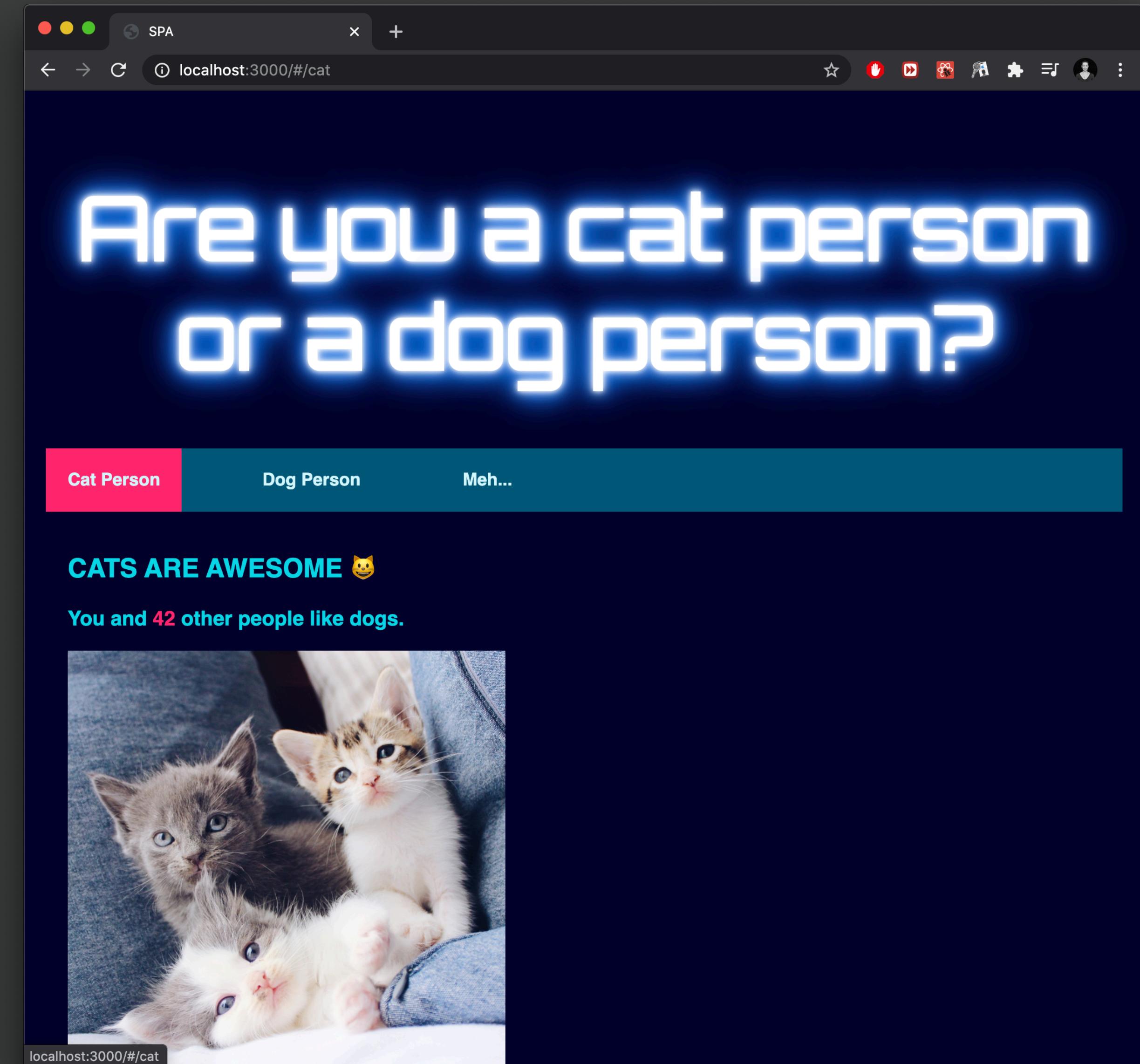
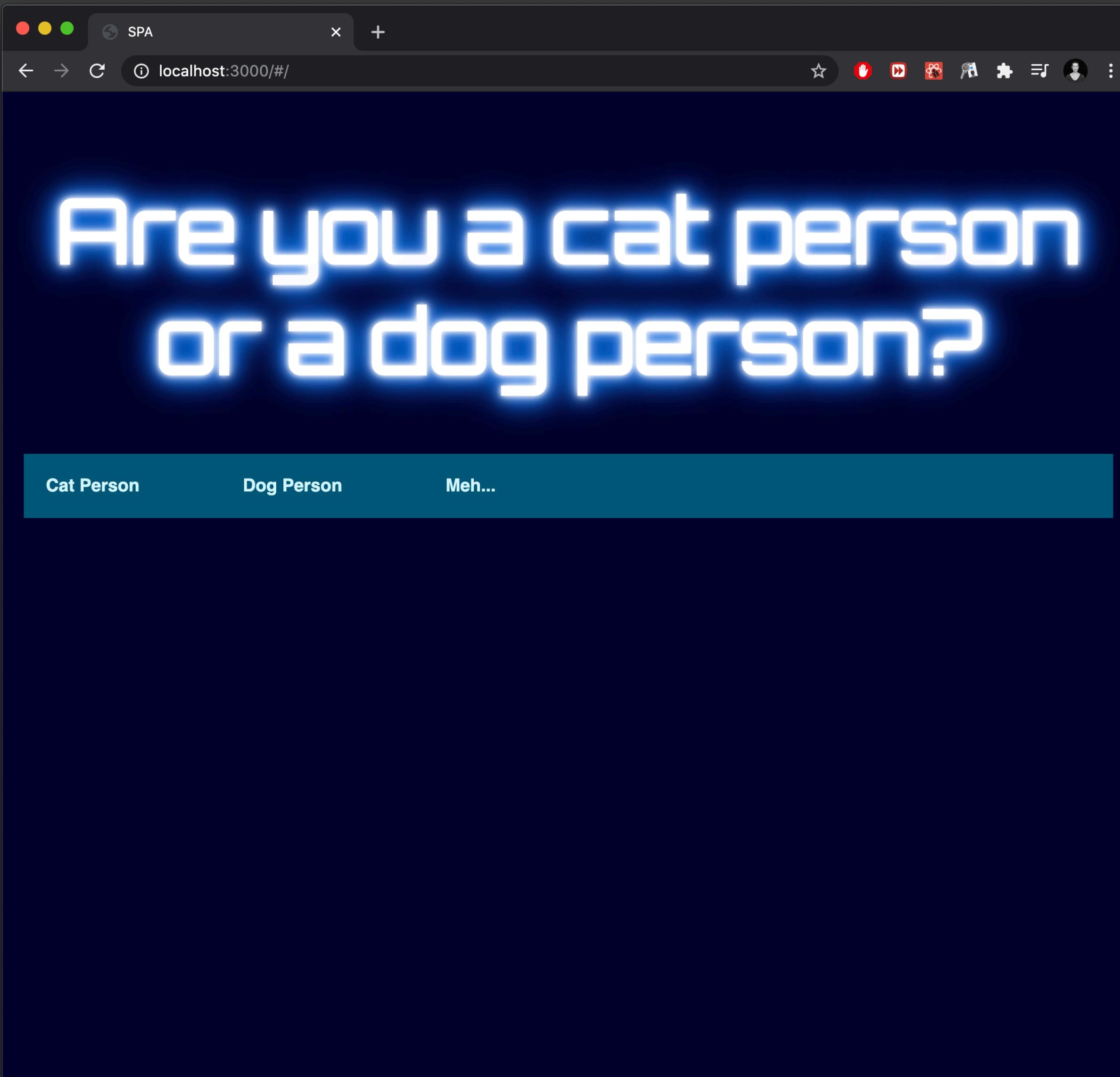
Wer sind wir?



CHRISTINA
@merelyChristina



ANNA
@merelyAnna



SPA

localhost:3000/#/dog

Are you a cat person or a dog person?

Cat Person Dog Person Meh...

DOGS ARE AWESOME 🐶

You and 42 other people like dogs.



localhost:3000/#/dog

SPA

localhost:3000/#/meh

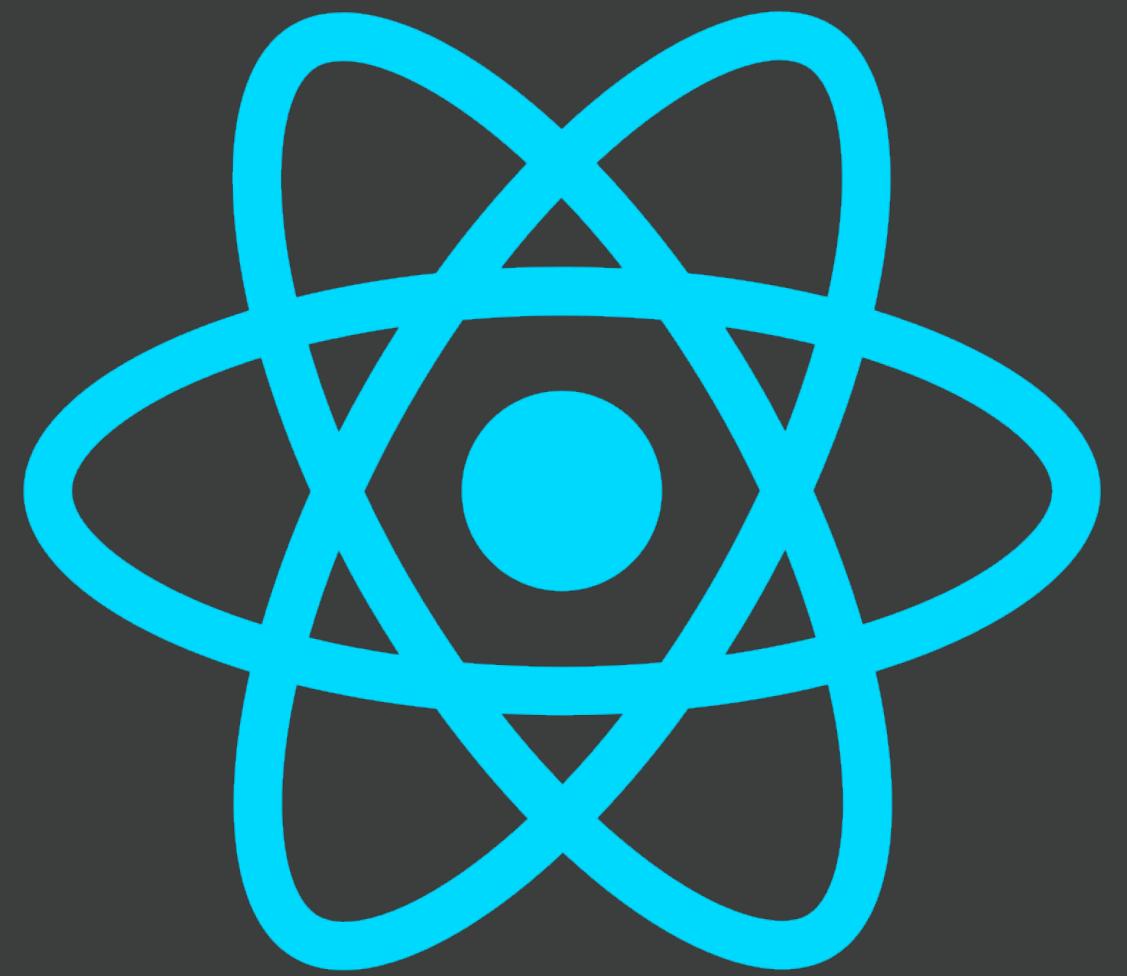
Are you a cat person or a dog person?

Cat Person Dog Person Meh...

What is wrong with you?!

localhost:3000/#/meh

SPA - SUPER POWER APPLICATION



HTML



CSS



Ziel des Abends:

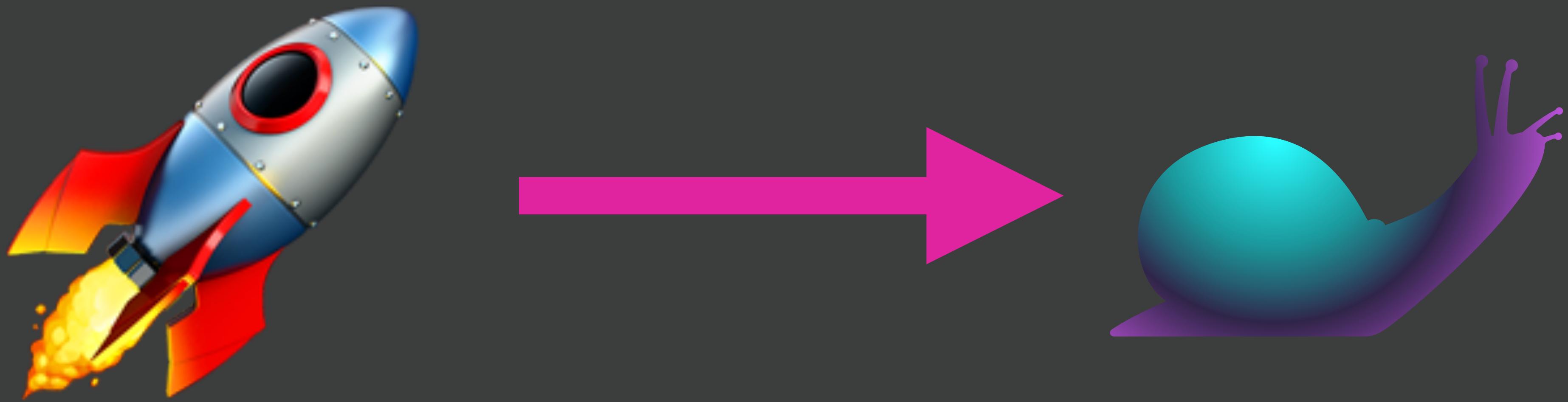
FRUSTRIERTE NUTZER



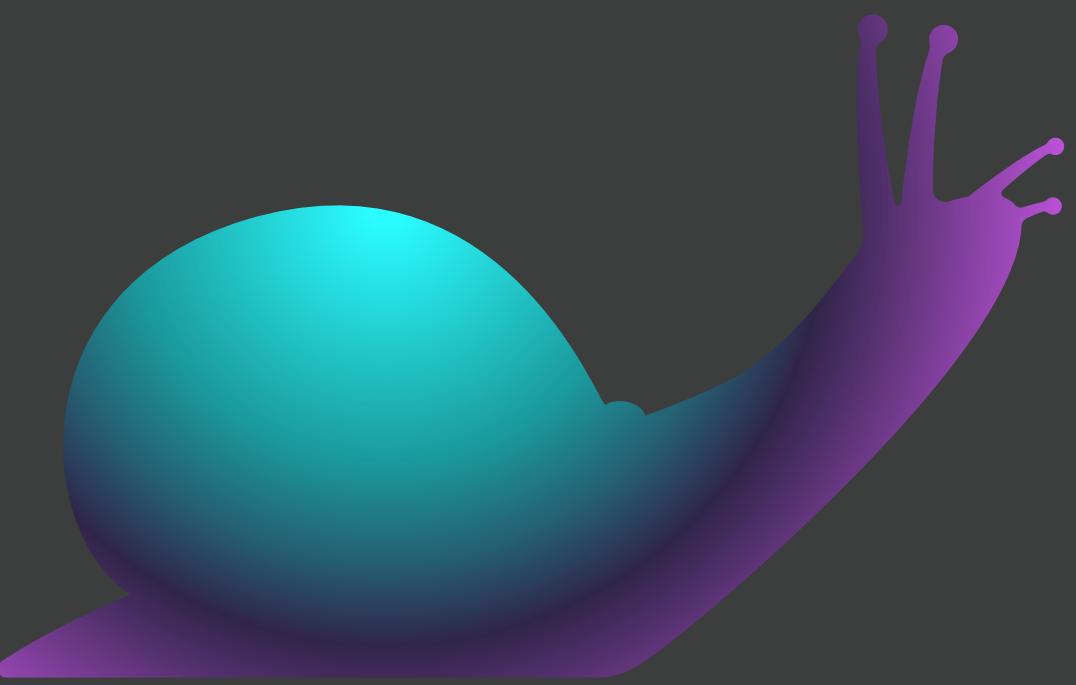
SPA - SUPER POWER APPLICATION



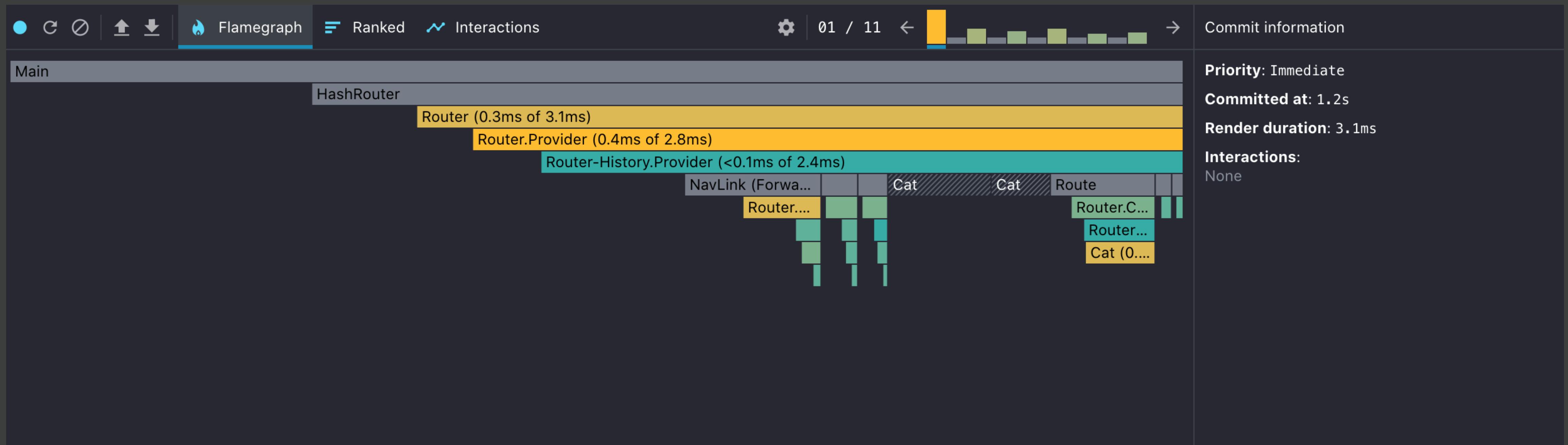
SPA - SLOW PACE APPLICATION



SPA - SLOW PACE APPLICATION

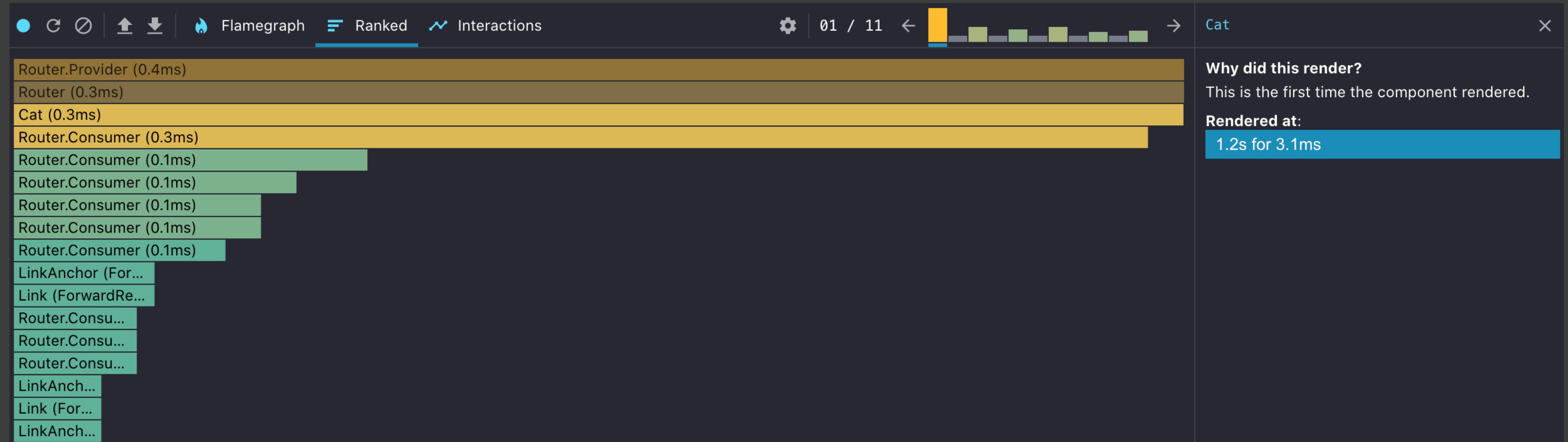


REACT PROFILER

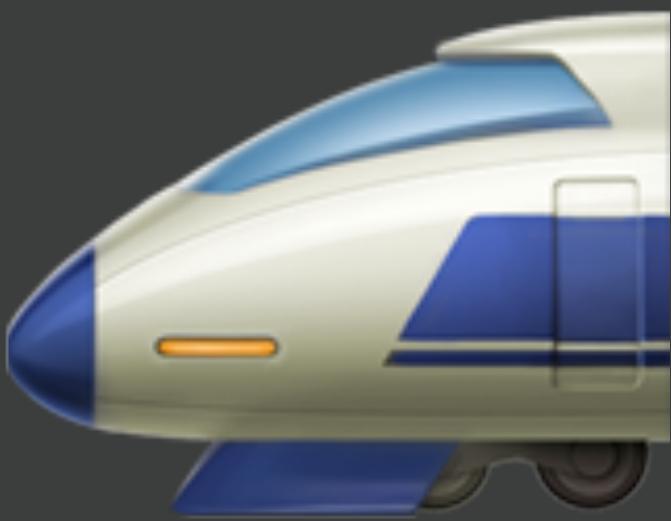
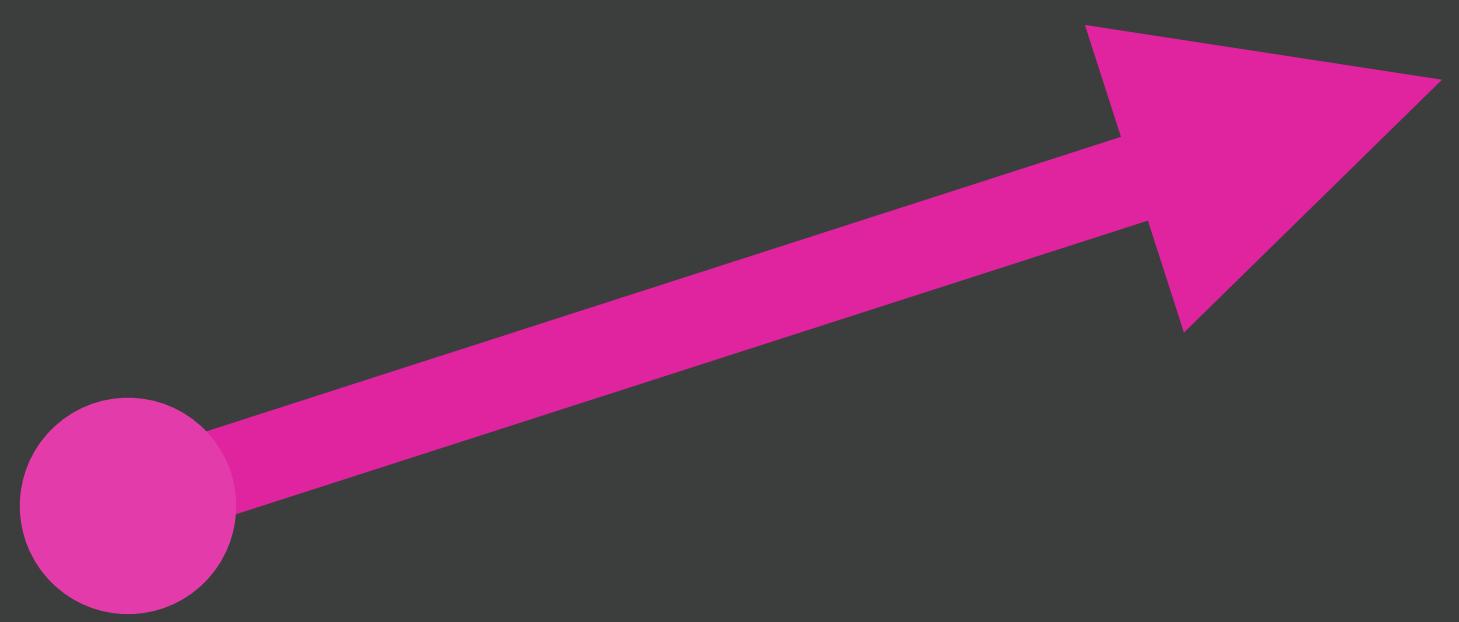
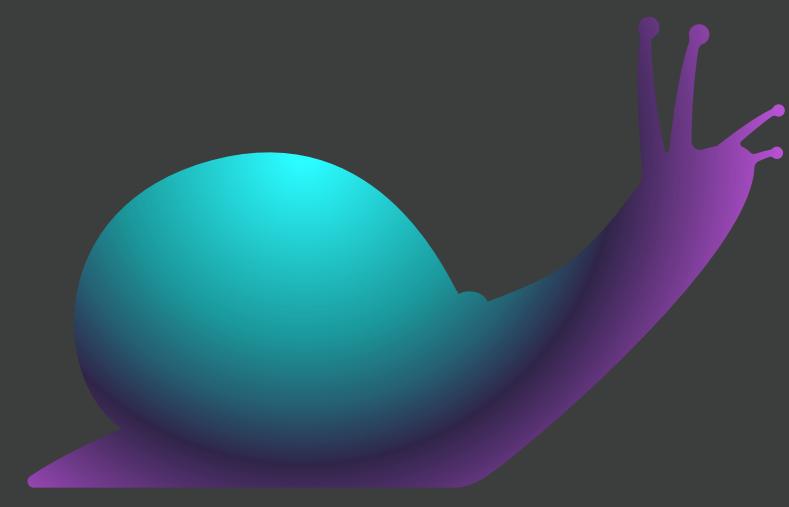


Flame Chart

REACT PROFILER



Ranked Chart



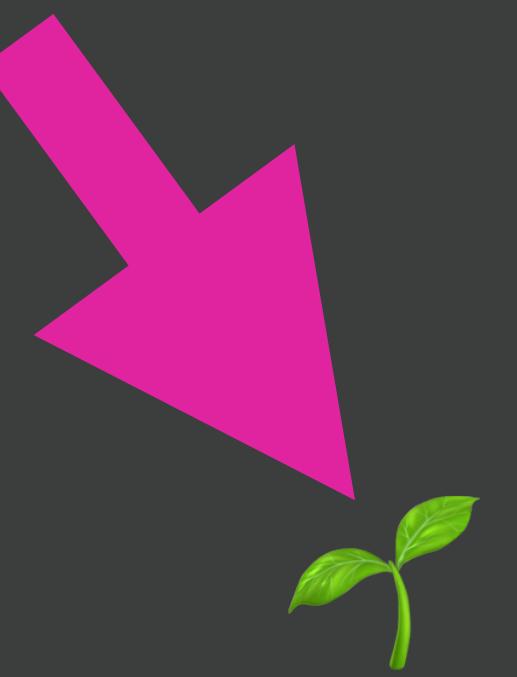
**CAN'T HAVE PRODUCTION
PROBLEMS**

**IF YOU LET YOUR USERS USE
DEVELOPMENT MODE**

memegenerator.net

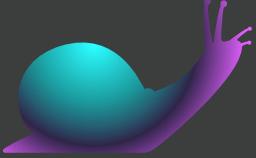


Warum ist das schnell?



Keine Warnings

Kleiner

 Wie mache ich das jetzt langsamer?

DEVELOPMENT MODE

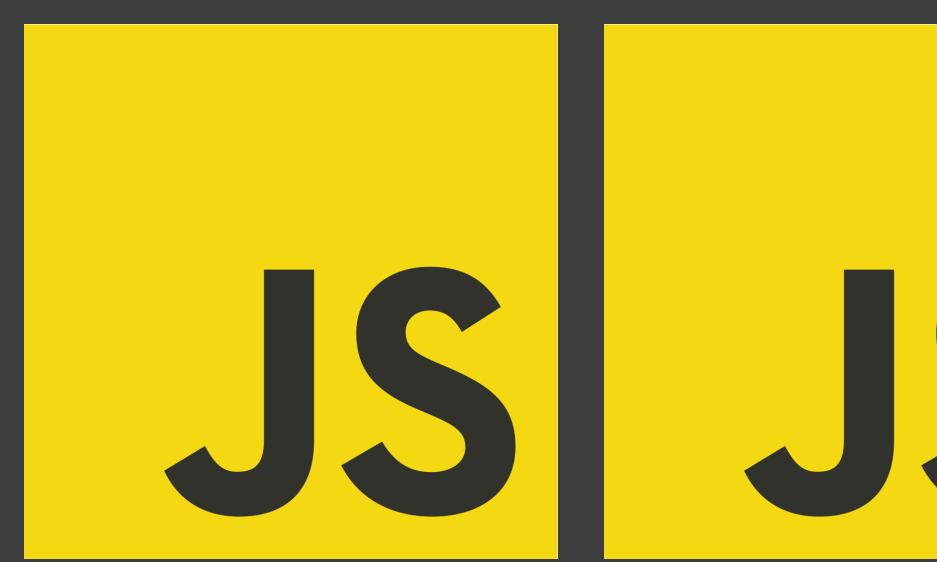
**YO DAWG, I HEARD YOU LIKE
BUNDLES**

**SO I PUT A BUNDLE INSIDE YOUR BUNDLE,
SO YOU CAN BUNDLE WHILE YOU'RE
BUNDLING**

memegenerator.net



Schnell - Bundling



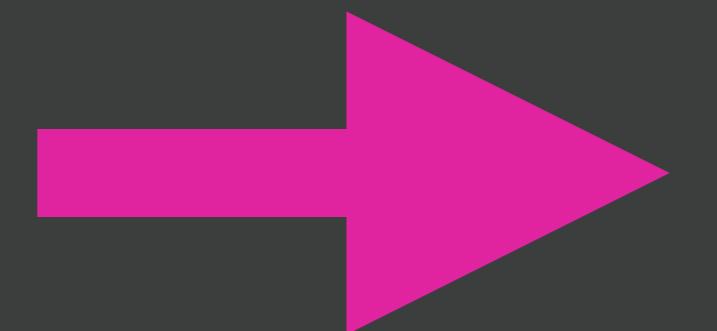
HTML



HTML



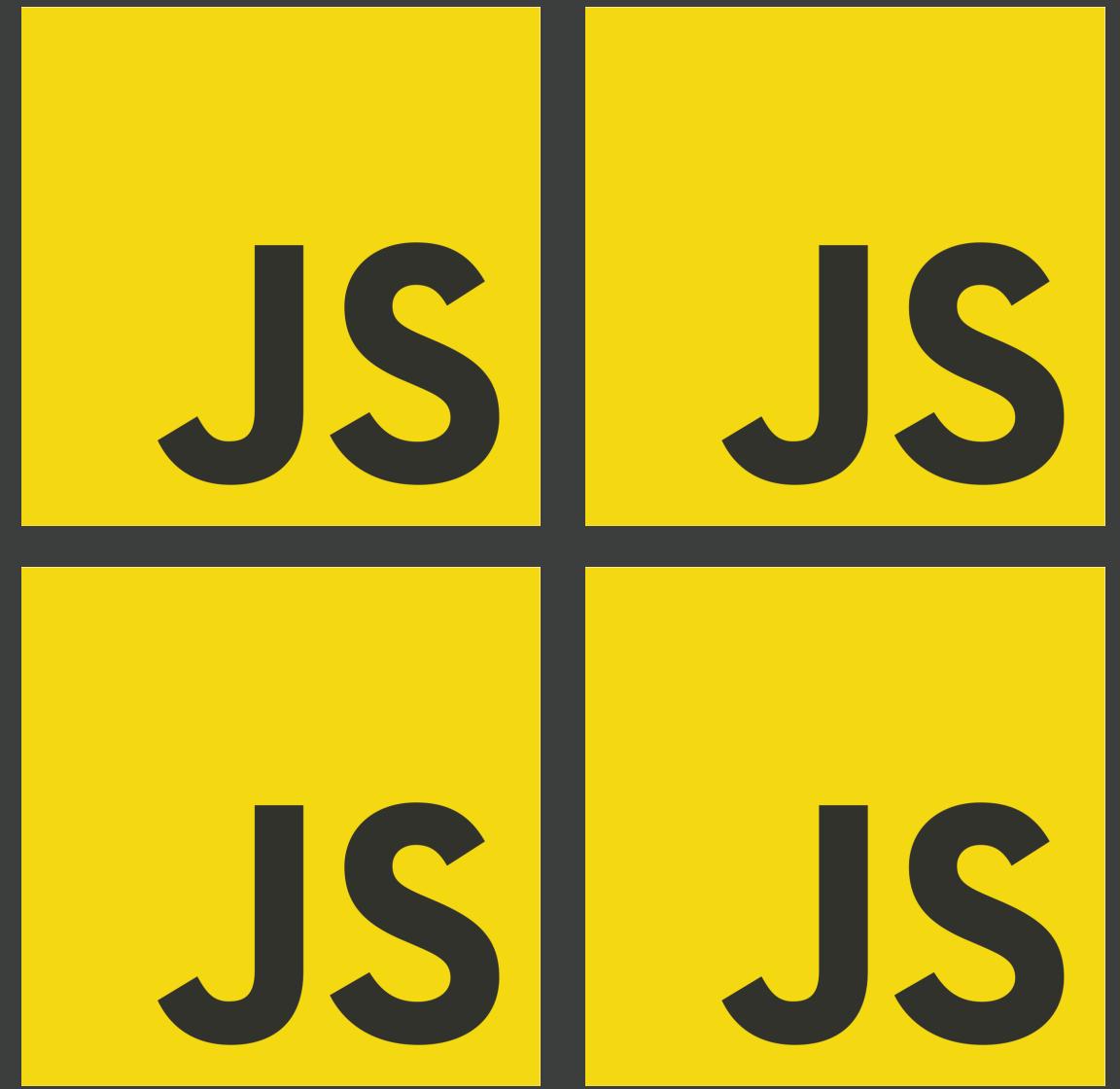
CSS



Web Bundle



Warum ist das schnell?





Warum ist das schnell?



CSS



HTML



1 Request

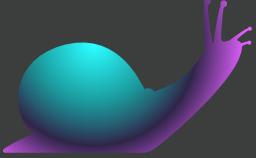
🐌 Wie mache ich das jetzt langsamer?



Viele Dateien



Keine minifizierten
Bundles



Wie mache ich das jetzt langsamer?



WITH CREATE-REACT-APP



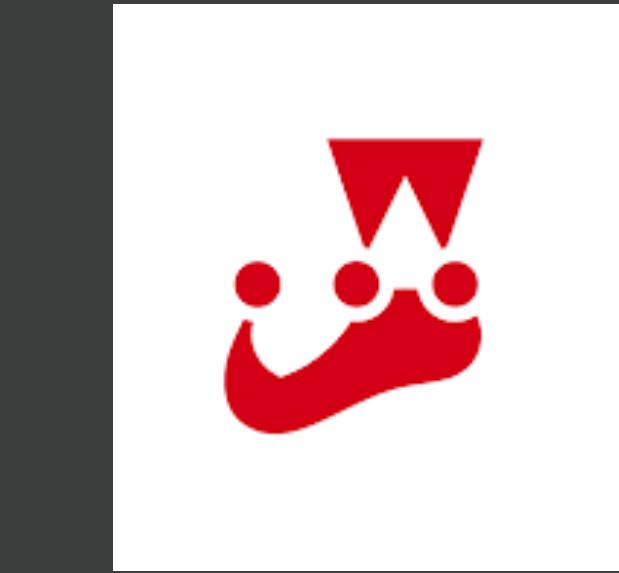
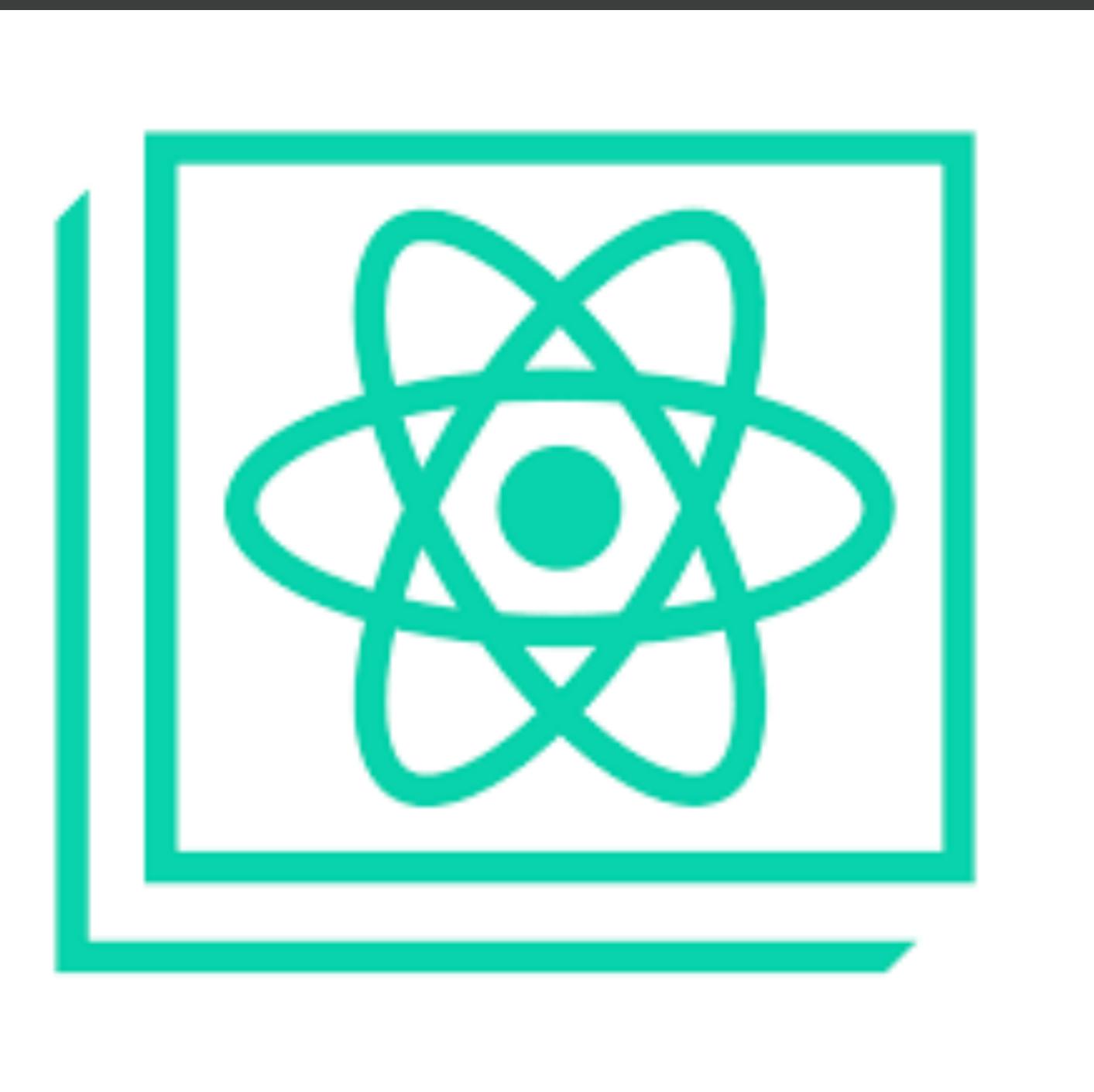
WITHOUT



Warum ist das schnell?



BABEL



Setze eine komplette React App auf



Warum ist das schnell?



Community



Entwickler

Warum macht das die SPA langsamer? 🐌



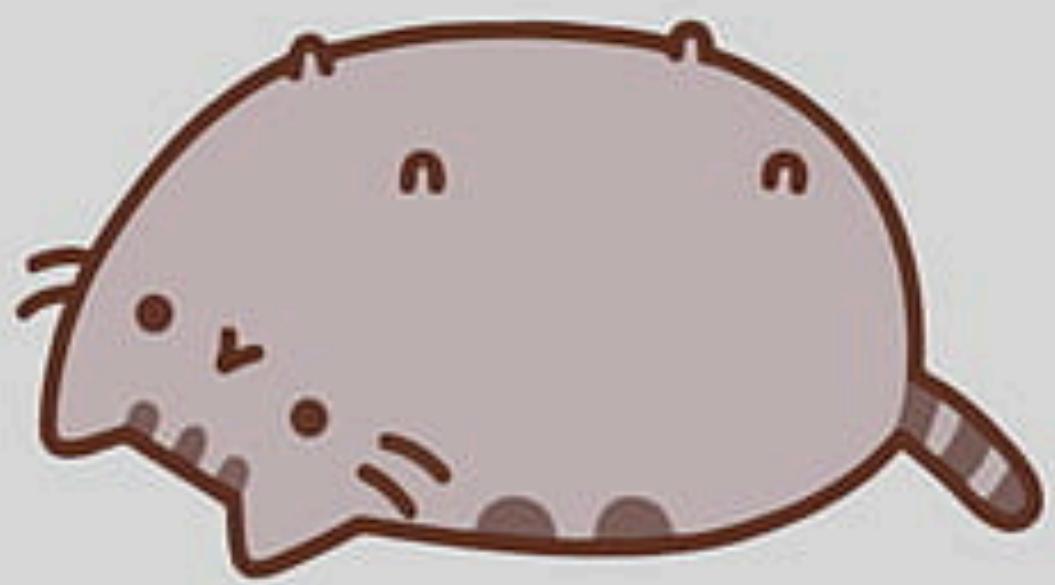
BABEL



Aufwendig und Kompliziert

LAZY LOADING

So Lazy



can't move



Schnell - Lazy Loading

```
export const MehTab = () => {
  const [value, setValue] = useState({ a: 20, b: 30 });
  return (
    <div>
      <h2>What is wrong with you?!</h2>
      <button onClick={() => setValue({ a: 20, b: 30 })}>More Mehs</button>
      ...
      <LargeImages></LargeImages>
    </div>
  );
};
```



Nicht so wichtig, langsam & wird nur hier verwendet



Schnell - Lazy Loading

```
const LargeImages = React.lazy(() => import("./LargeImages"));

export const MehTab = () => {
  const [value, setValue] = useState({ a: 20, b: 30 });
  return (
    <div>
      <h2>What is wrong with you?!</h2>
      <button onClick={() => setValue({ a: 20, b: 30 })}>More Mehs</button>
      ...
      <Suspense fallback="loading image ....">
        <LargeImages></LargeImages>
      </Suspense>
    </div>
  );
};
```



Nicht so wichtig, langsam & wird nur hier verwendet



Warum ist das schnell?

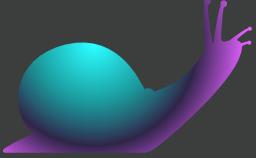
Chunk1

Core-Feature

Core-Feature

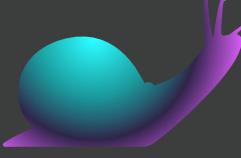
Chunk2

Large Images



Wie mache ich das jetzt langsamer?

```
export const MehTab = () => {
  const [value, setValue] = useState({ a: 20, b: 30 });
  return (
    <div>
      <h2>What is wrong with you?!</h2>
      <button onClick={() => setValue({ a: 20, b: 30 })}>More Mehs</button>
      <LargeImages></LargeImages>
    </div>
  );
};
```



Wie mache ich das jetzt langsamer?

BUNDLE

Core-Feature

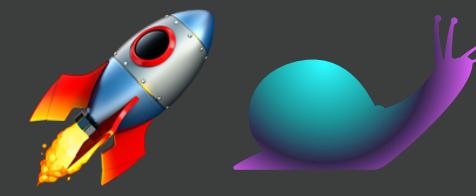
Unwichtige &
langsam

Unwichtige &
langsam

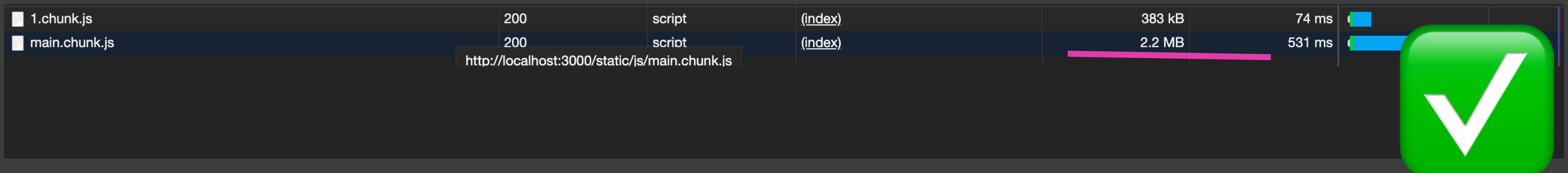
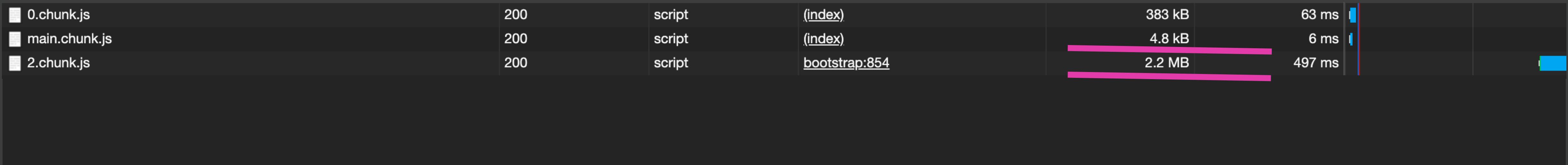
Unwichtige &
langsam

Core-Feature

Core-Feature



Hat das was gebracht?



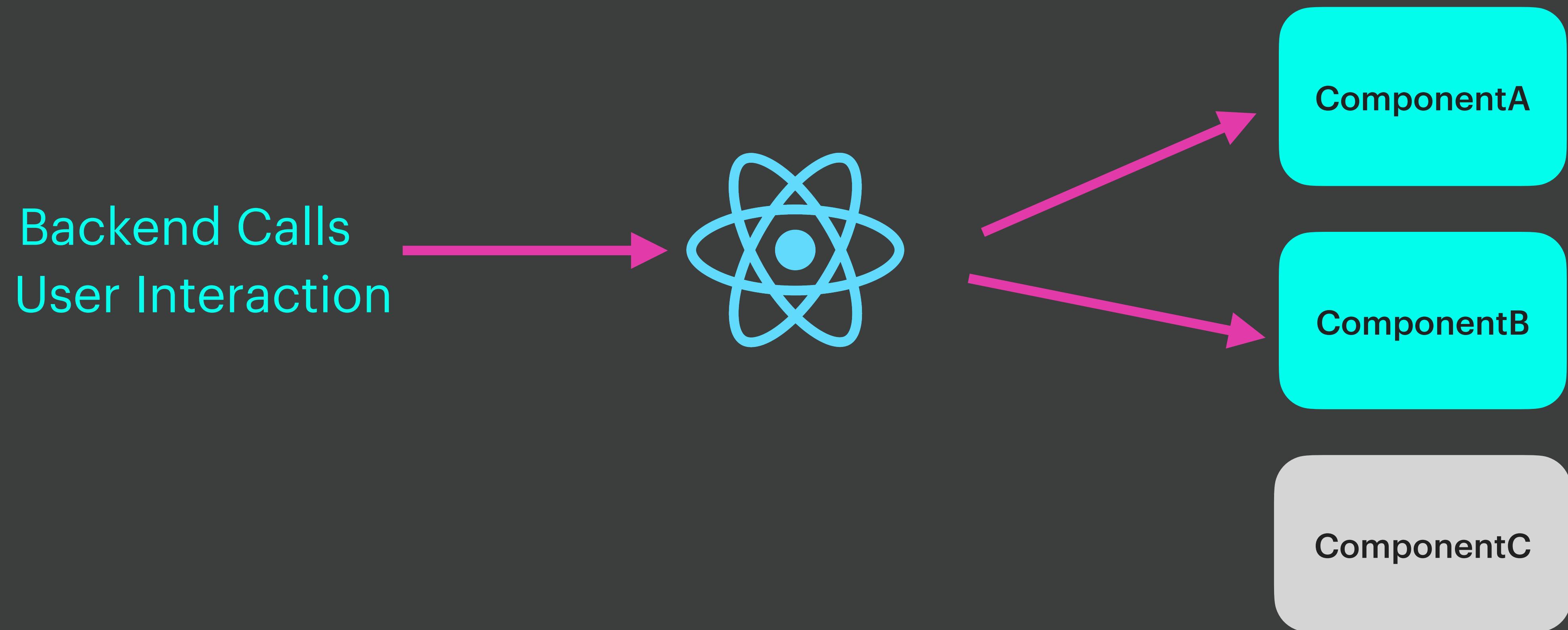
ONE COMPONENT



TO RULE THEM ALL

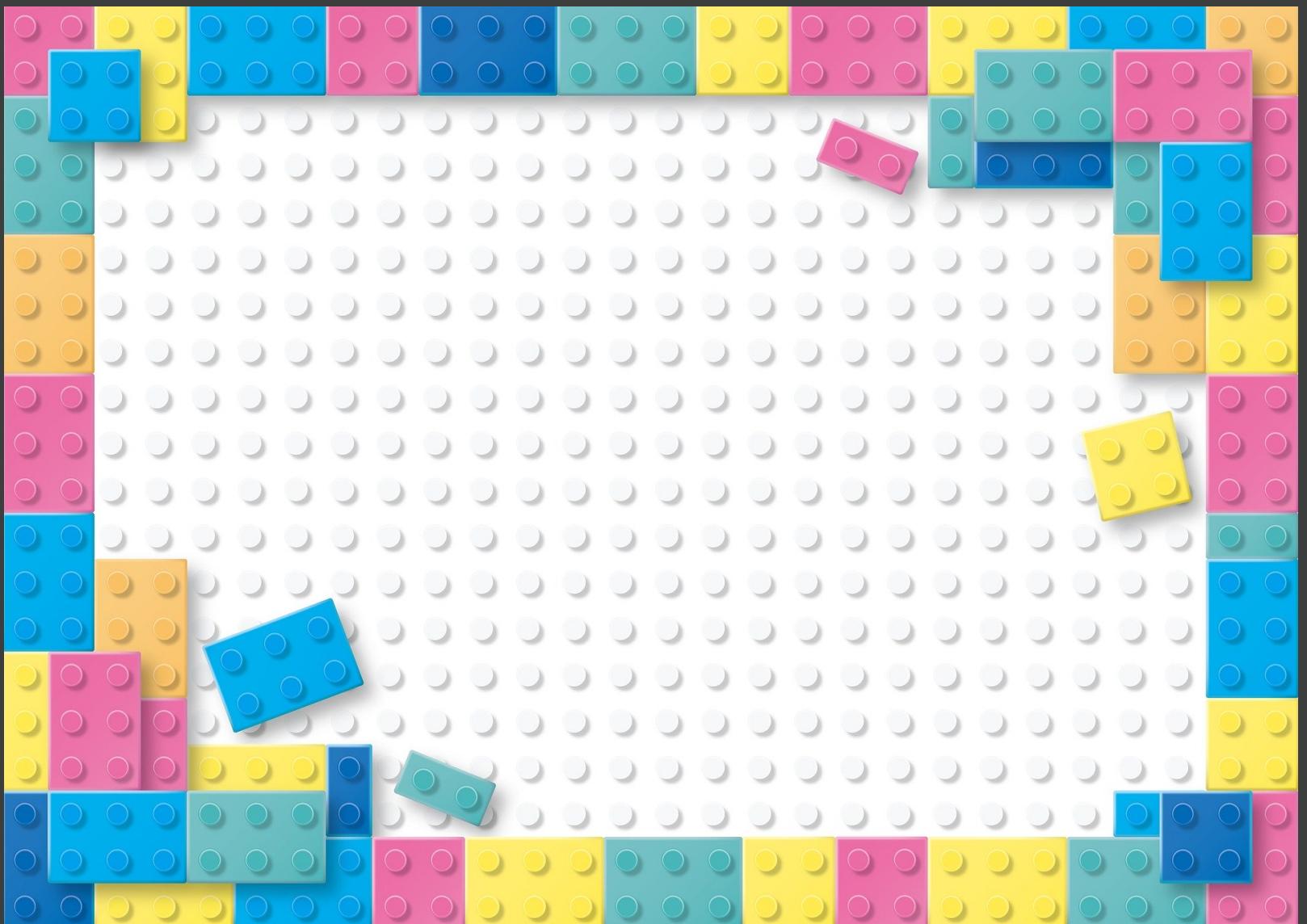


Schnell - Component Splitting





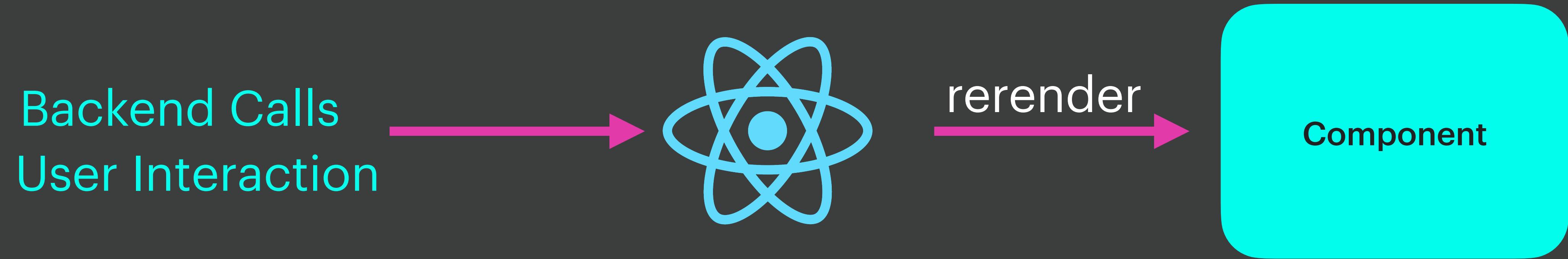
Warum ist das schnell?



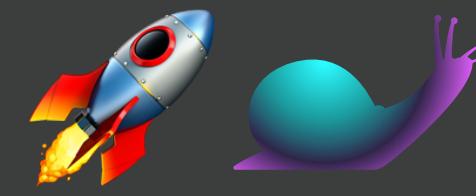
Code Splitting

Small Components

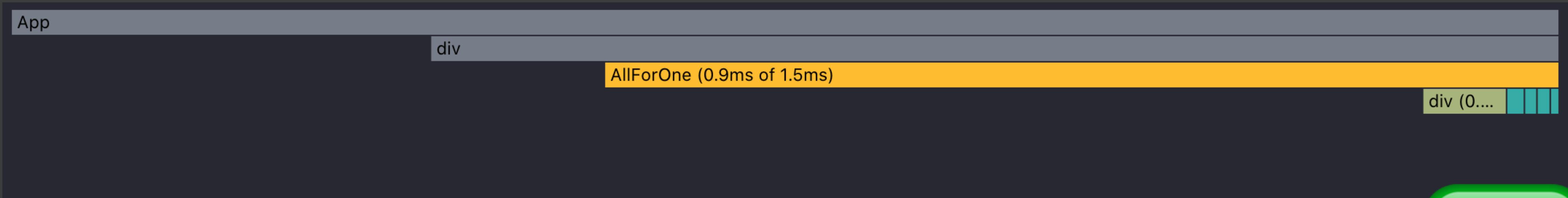
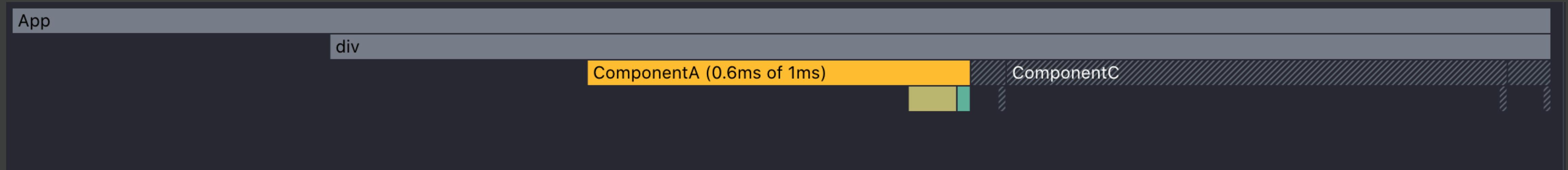
🐌 Wie mache ich das jetzt langsamer?

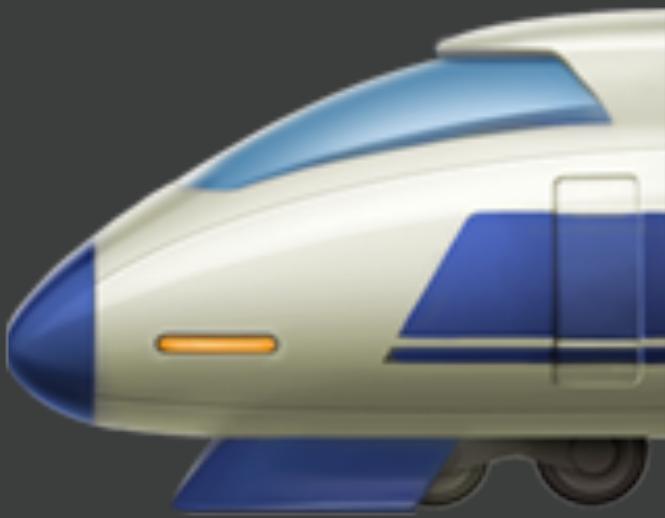
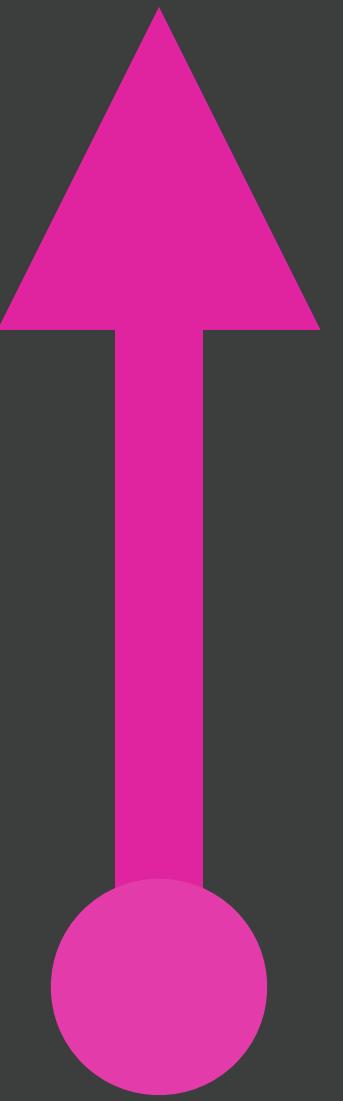
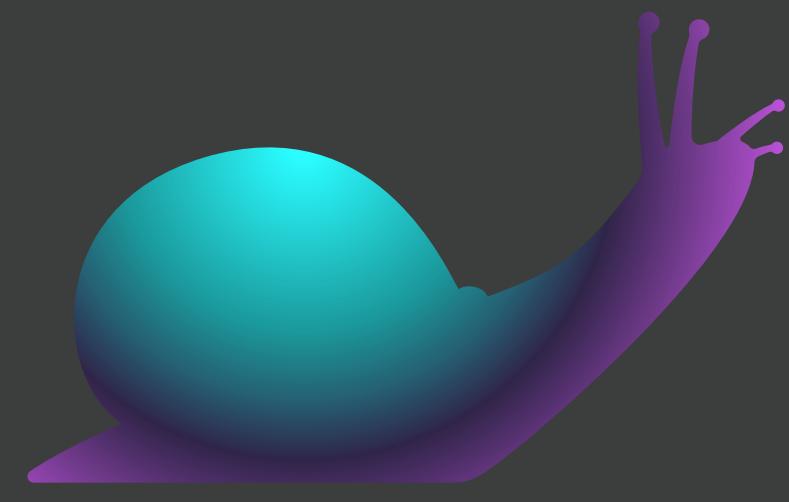


Eine Component = alles wir neu gerendert



Hat das was gebracht?





RERENDERING

RERENDERING EVERYWHERE

imgflip.com



Schnell - PureComponent

```
export class Cat extends PureComponent {  
  
  render() {  
    return (  
      <li onClick={this.props.handleClick}>  
        <h2>{this.props.name} </h2>  
        <p>{this.props.description}</p>  
      </li>  
    );  
  }  
}  
  
shouldComponentUpdate(nextProps) {  
  return (  
    this.props.name !== nextProps.name ||  
    this.props.description !== nextProps.description  
  );  
}
```



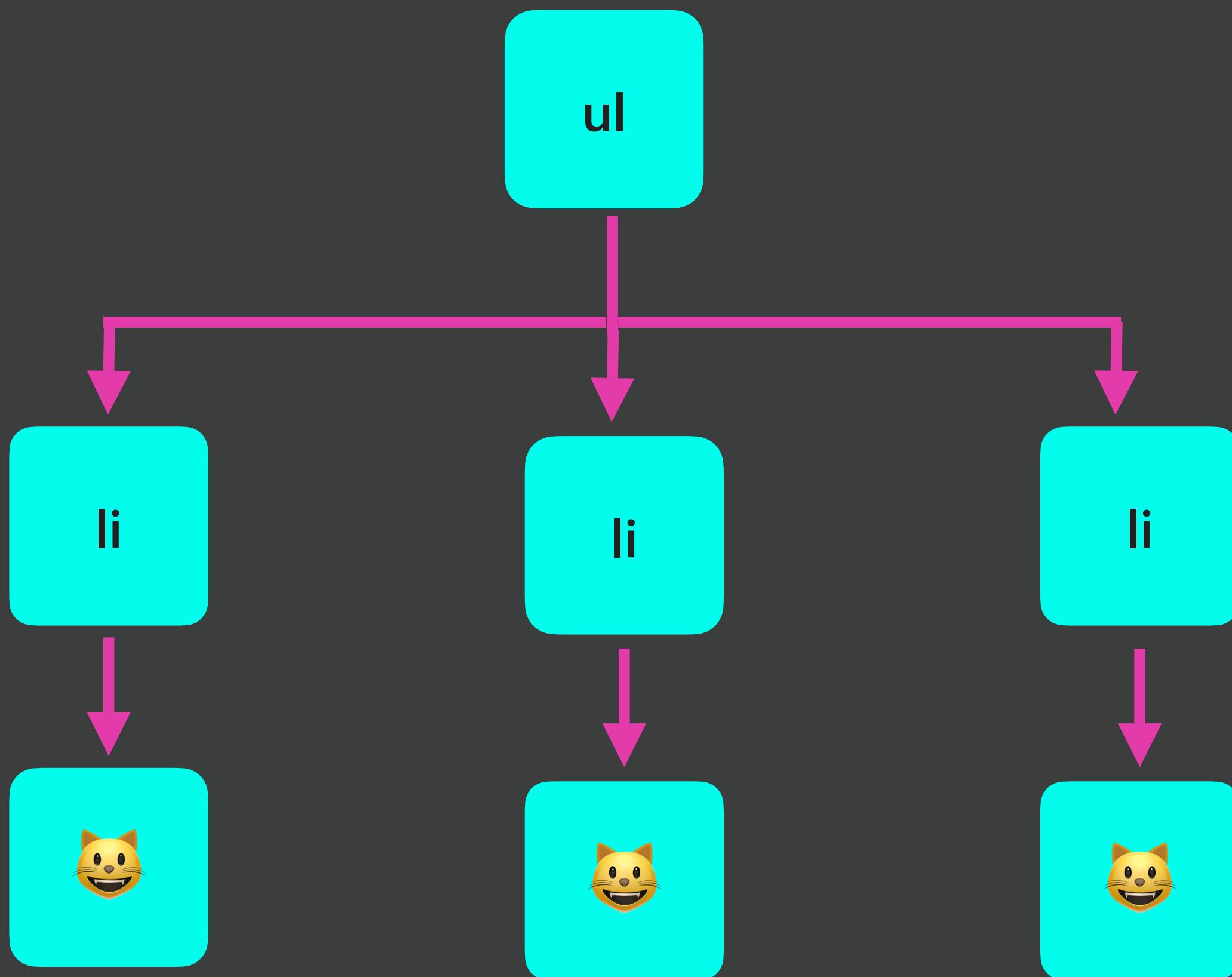
Schnell - shouldComponentUpdate

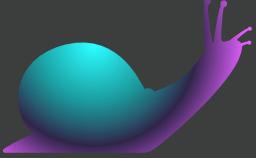
```
export class Cat extends Component {
  shouldComponentUpdate(nextProps) {
    return this.props.name !== nextProps.name;
  }

  render() {
    return (
      <li onClick={this.props.handleClick}>
        <h2>{this.props.name} </h2>
        <p>{this.props.description}</p>
      </li>
    );
  }
}
```



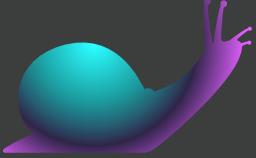
Warum ist das schnell?



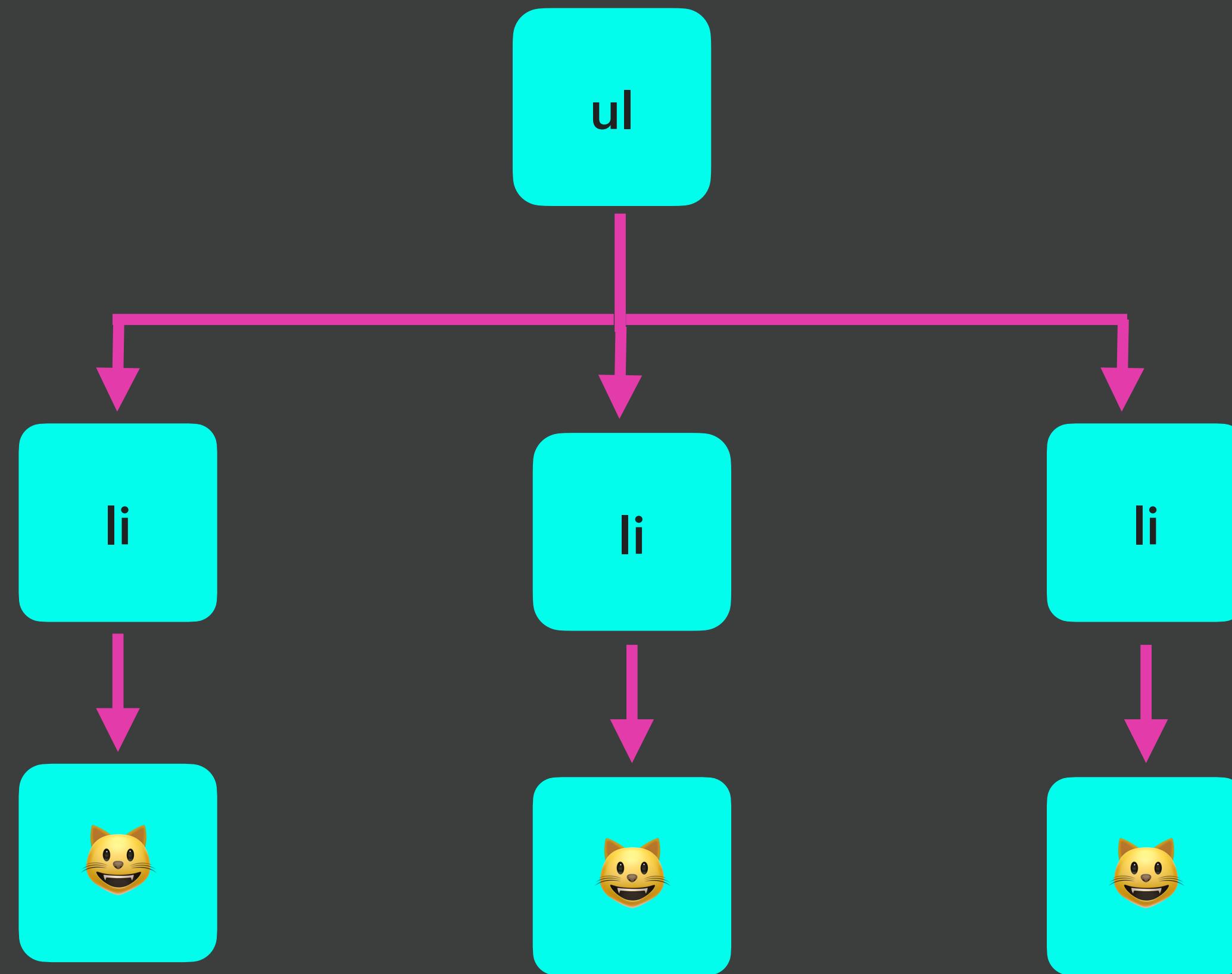


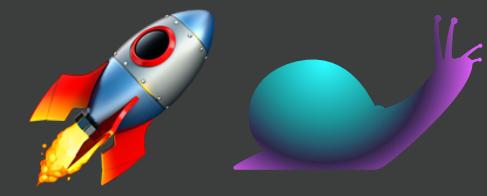
Wie mache ich das jetzt langsamer?

```
export class Cat extends Component {  
  
  shouldComponentUpdate(){  
    return true;  
  }  
  
  render() {  
    return (  
      <li onClick={this.props.handleClick}>  
        <h2>{this.props.name} </h2>  
        <p>{this.props.description}</p>  
      </li>  
    );  
  }  
}
```

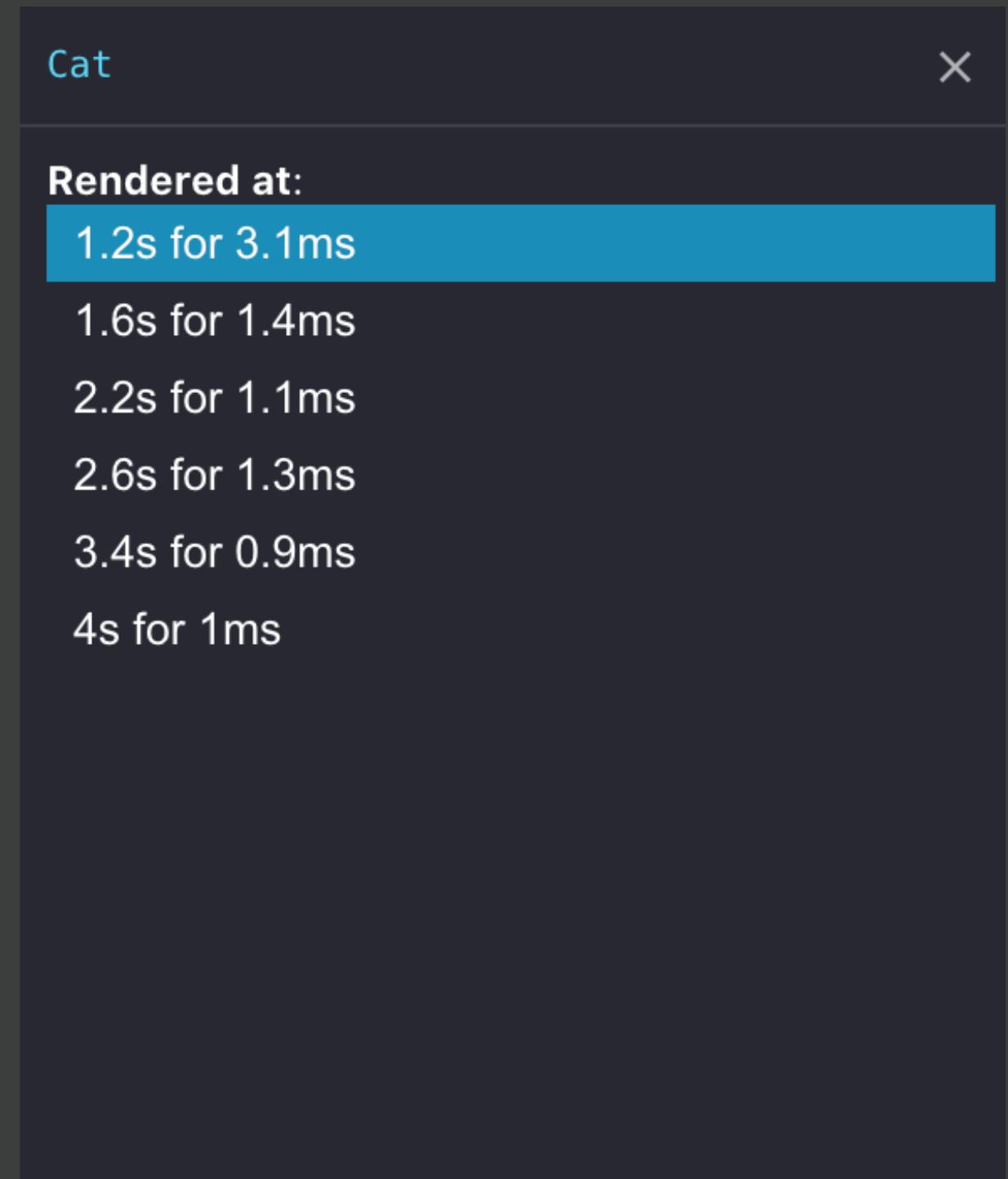
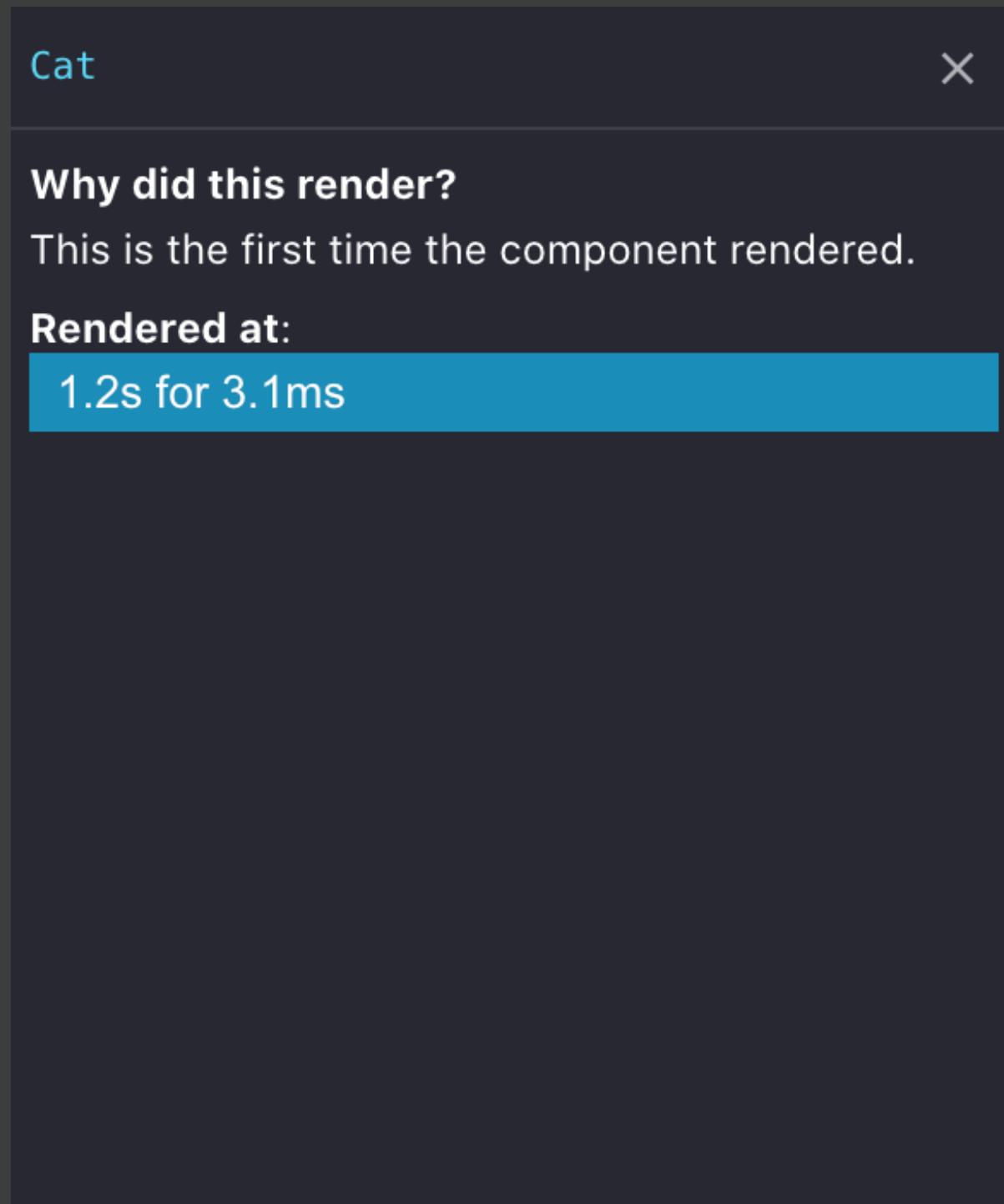


Wie mache ich das jetzt langsamer?





Hat das was gebracht?



CLASS COMPS

I'M YOU FATHER

FUNCTION COMPS

NOOOOOOOO



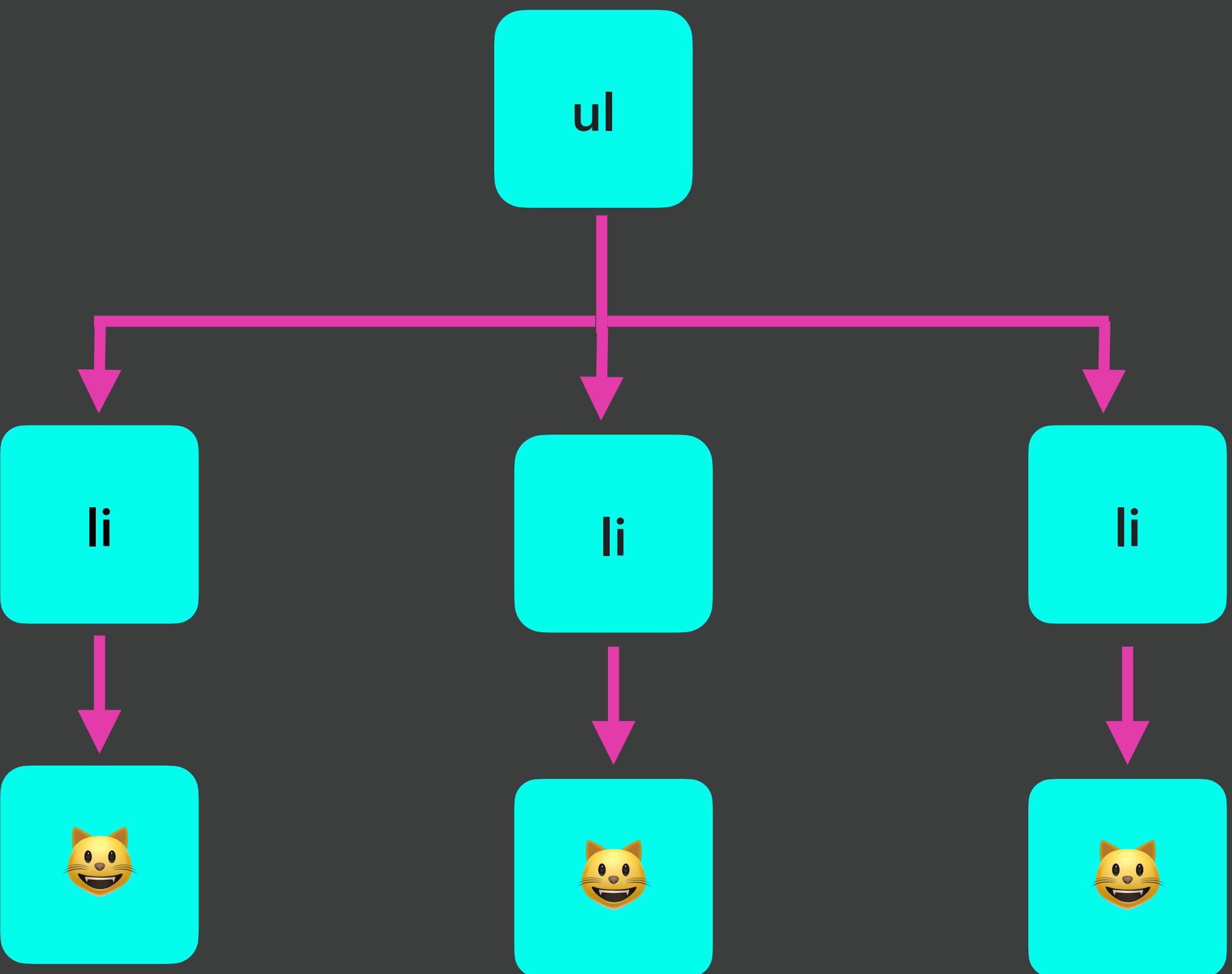
Schnell - React.memo

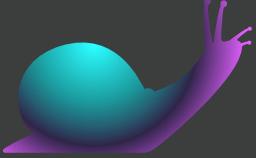
```
export const Cat = (props) => (
  <div onClick={props.handleClick}>
    <h2>{props.name} </h2>
    <p>{props.description}</p>
  </div>
);
  
export default React.memo(Cat);
```

```
this.props.name !== nextProps.name ||  
this.props.description !== nextProps.description
```



Warum ist das schnell?

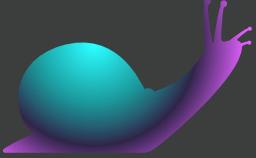




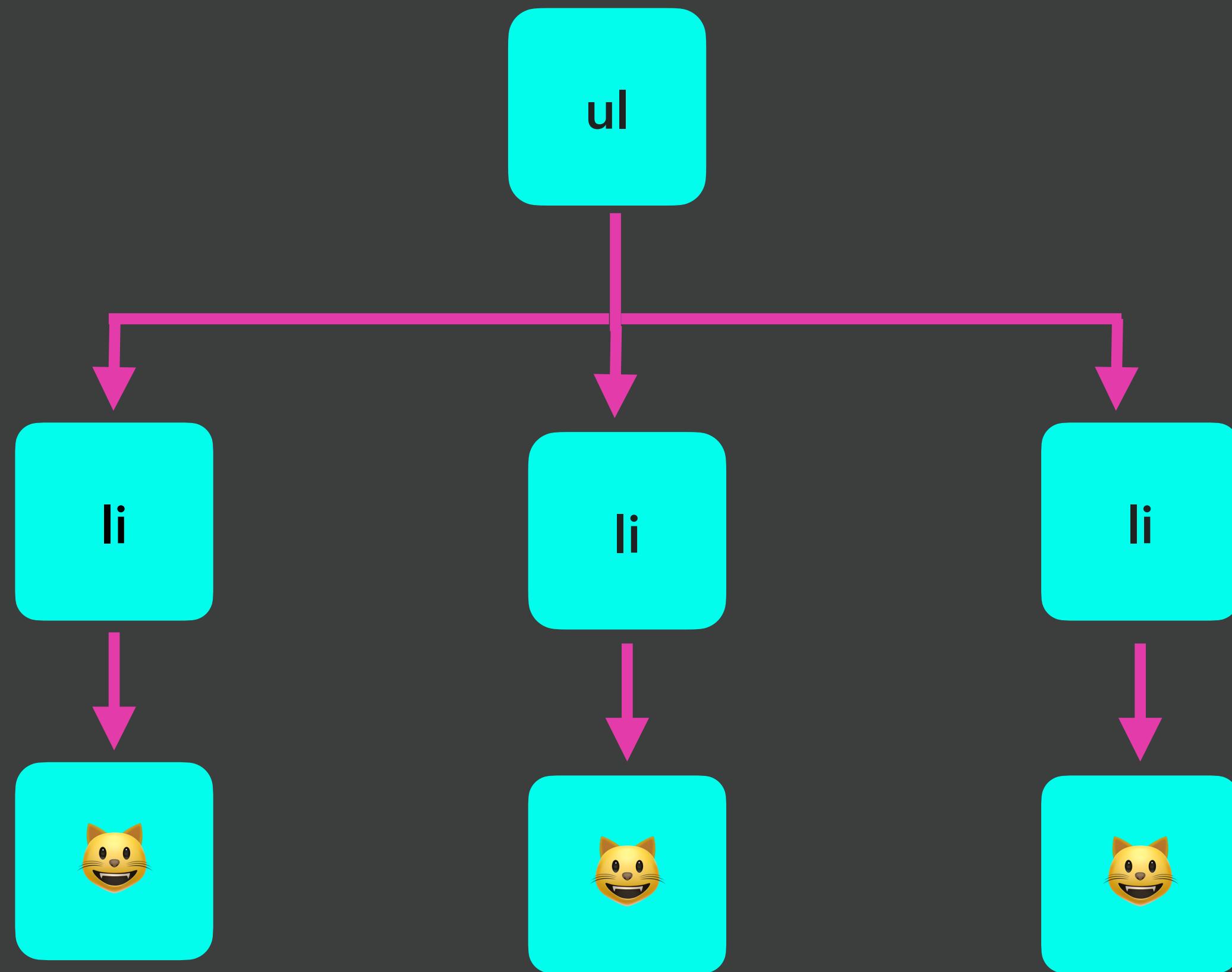
Wie mache ich das jetzt langsamer?

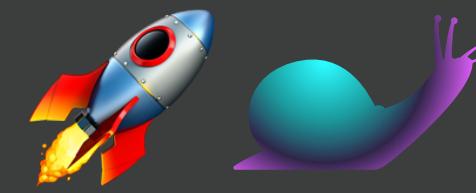
```
export const Cat = (props) => (
  <div onClick={props.handleClick}>
    <h2>{props.name} </h2>
    <p>{props.description}</p>
  </div>
);
```

```
shouldComponentUpdate(){
  return true;
}
```

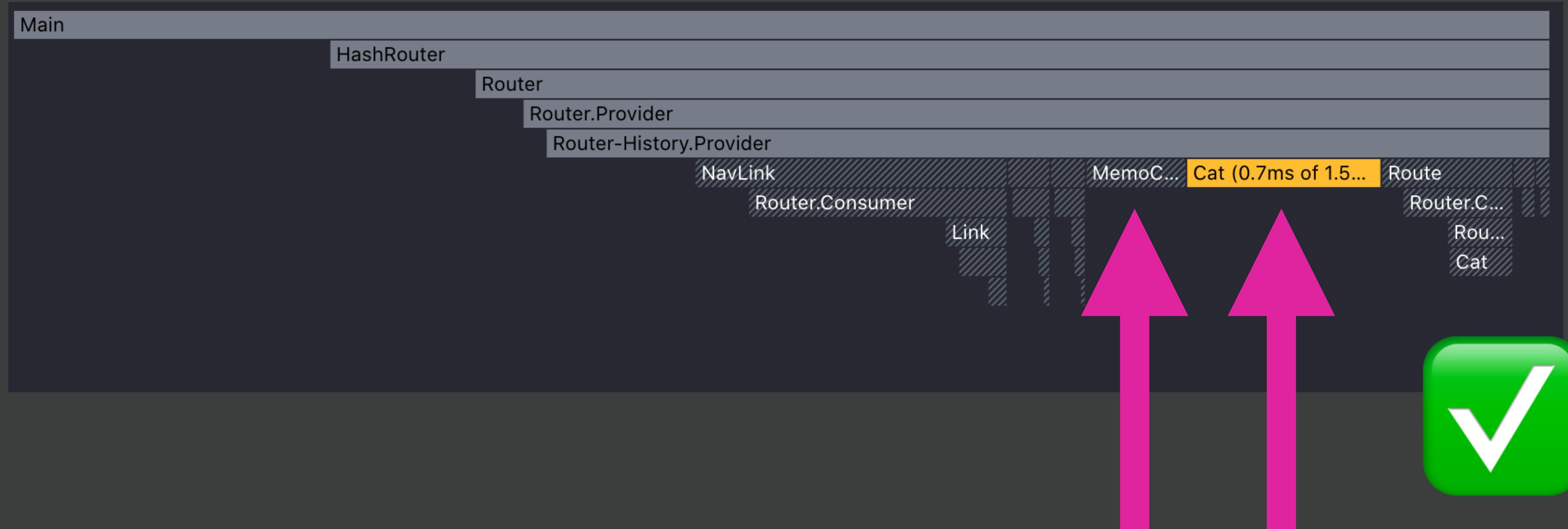


Wie mache ich das jetzt langsamer?

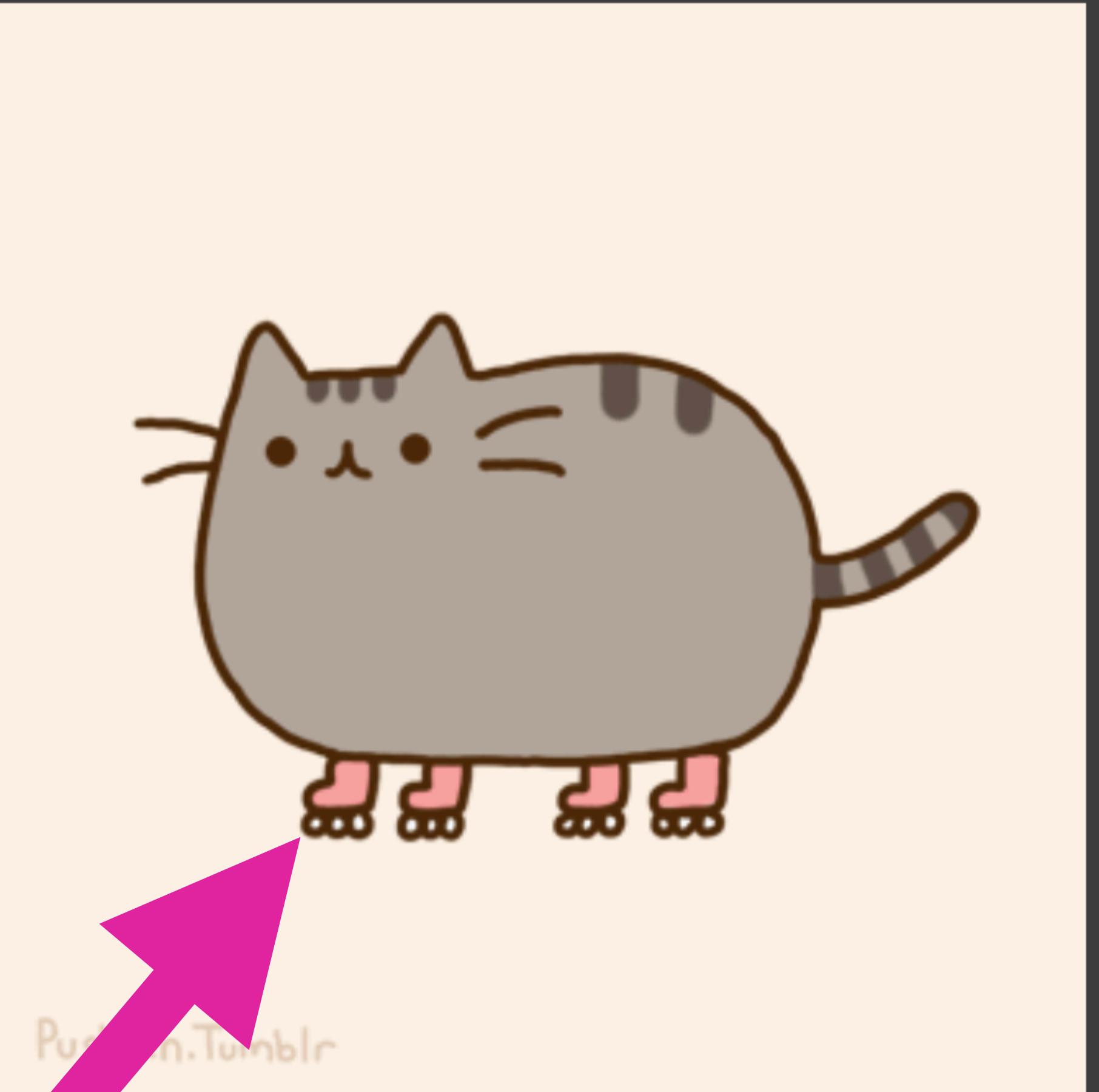




Hat das was gebracht?



INLINE FUNCTIONS



Inline skates. Hahaha. Get it?

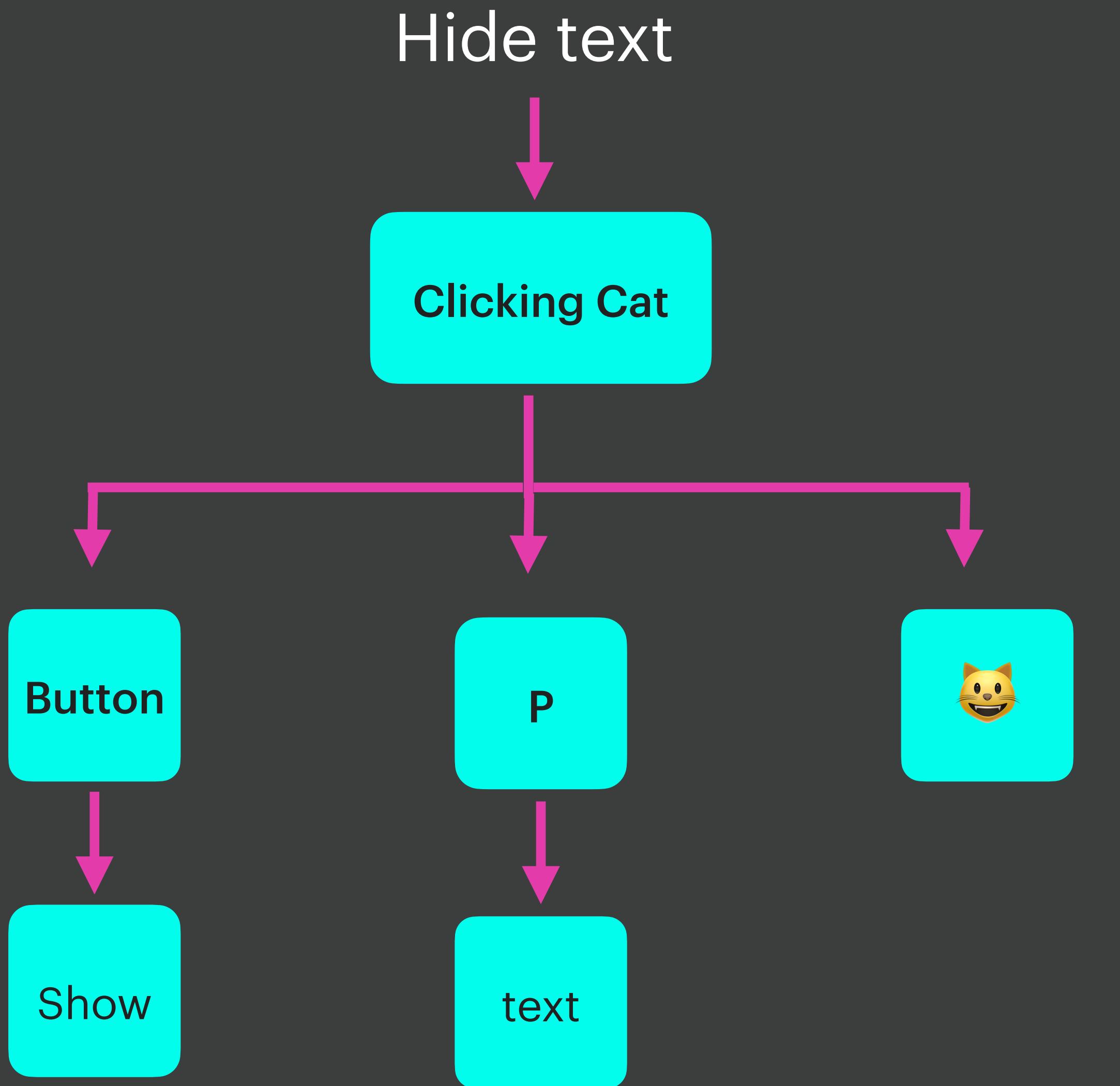


Schnell - class methode

```
export class ClickingCat extends PureComponent {  
  ...  
  
  logHandler() {  
    () => console.log("Cat was clicked");  
  }  
  
  render() {  
    return (  
      <>  
        <button onClick={() => /* update state */}>  
          {this.state.textIsVisible ? "hide text" : "showText"}  
        </button>  
        {this.state.textIsVisible && (<p> Text ...</p>)}  
        <Cat  
          name="Lulu"  
          description="A very lazy cat"  
          handleClick={this.logHandler}  
        ></Cat>  
      </>  
    );  
  }  
}
```



Warum ist das schnell?





Warum ist das schnell?

Hide text



Clicking Cat



Button

P



Show

text

```
<button onClick={() =>/ * update state */ }>  
  {this.state.textIsVisible ? "hide text" : "..."}  
</button>
```



Warum ist das schnell?

Hide text



Clicking Cat



Button



Show



P



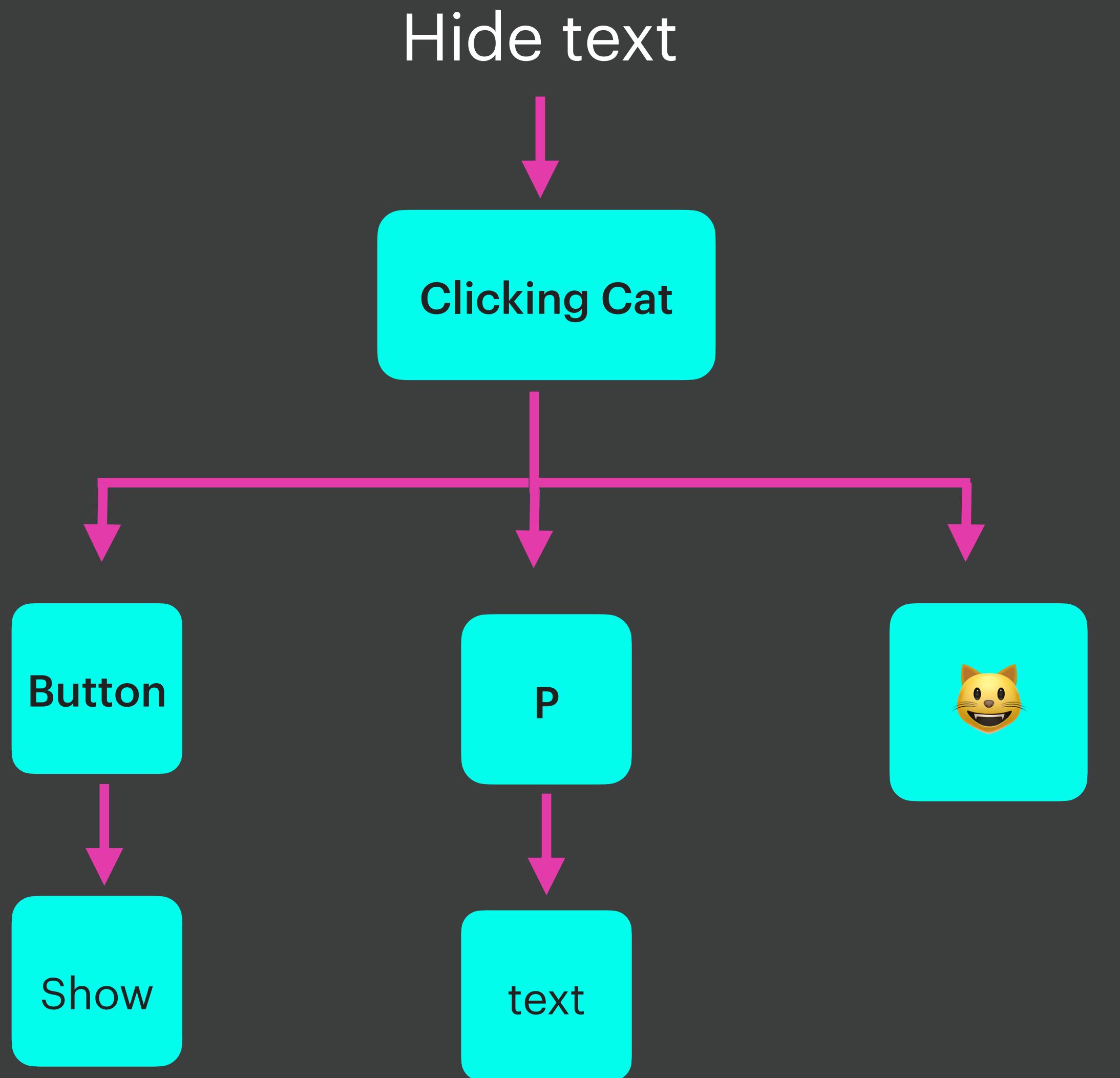
```
{this.state.textIsVisible && (<p> Text ...</p>)}
```



text

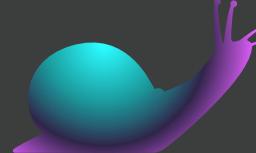


Warum ist das schnell?



```
<Cat  
  name="Lulu"  
  description="A very lazy cat"  
  handleClick={this.logHandler}  
}></Cat>;
```

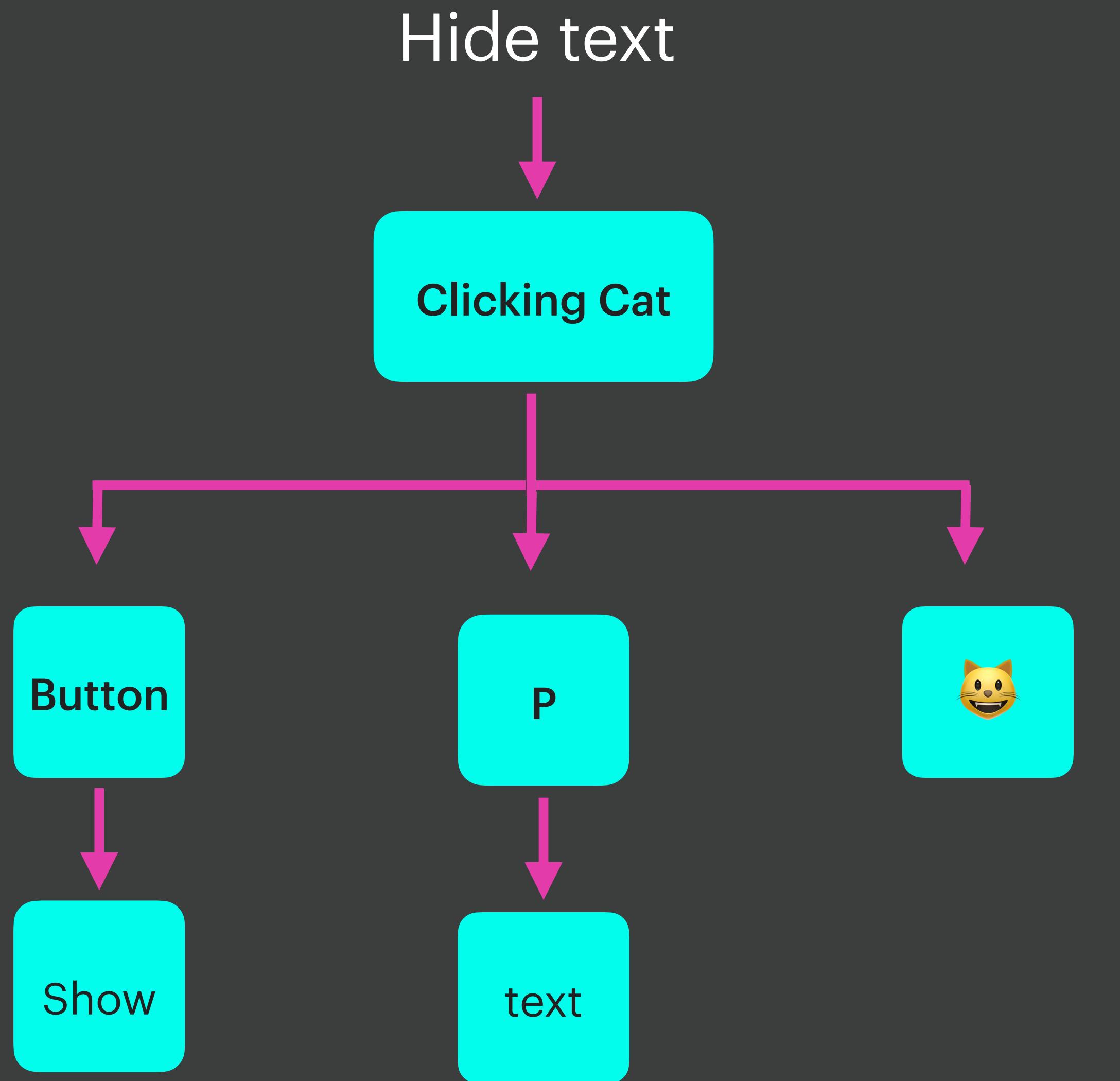
Reference ist gleich



Wie mache ich das jetzt langsamer?

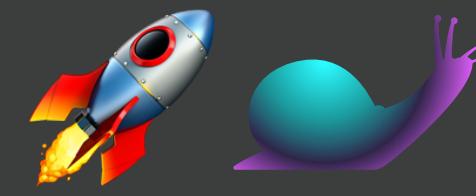
```
export class ClickingCat extends PureComponent {  
  ...  
  
  render() {  
    return (  
      <>  
        <button onClick={() =>/* update state */}>  
          {this.state.textIsVisible ? "hide text" : "showText"}  
        </button>  
        {this.state.textIsVisible && (<p> Text ...</p>) }  
        <Cat  
          name="Lulu"  
          description="A very lazy cat"  
          handleClick={() => console.log("Cat was clicked") }  
        ></Cat>  
      </>  
    );  
  }  
}
```

🐌 Wie mache ich das jetzt langsamer?

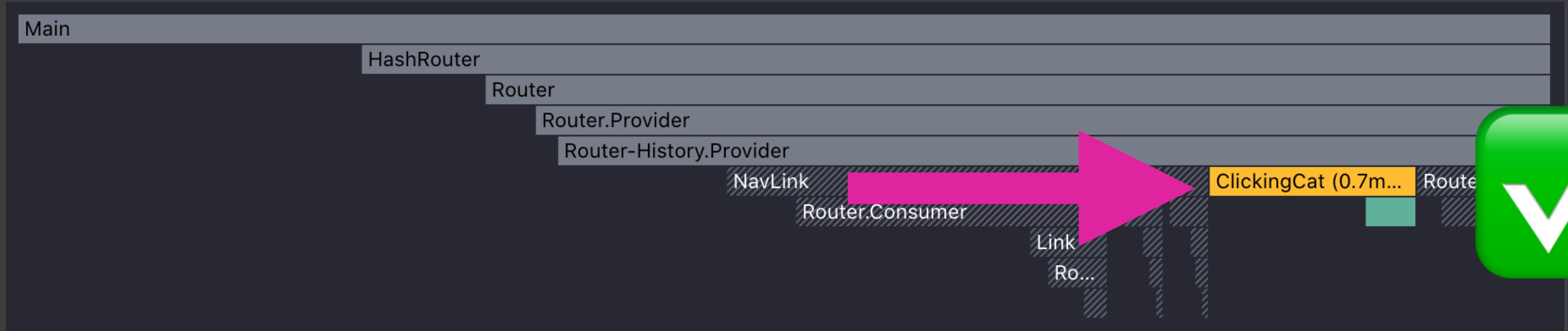
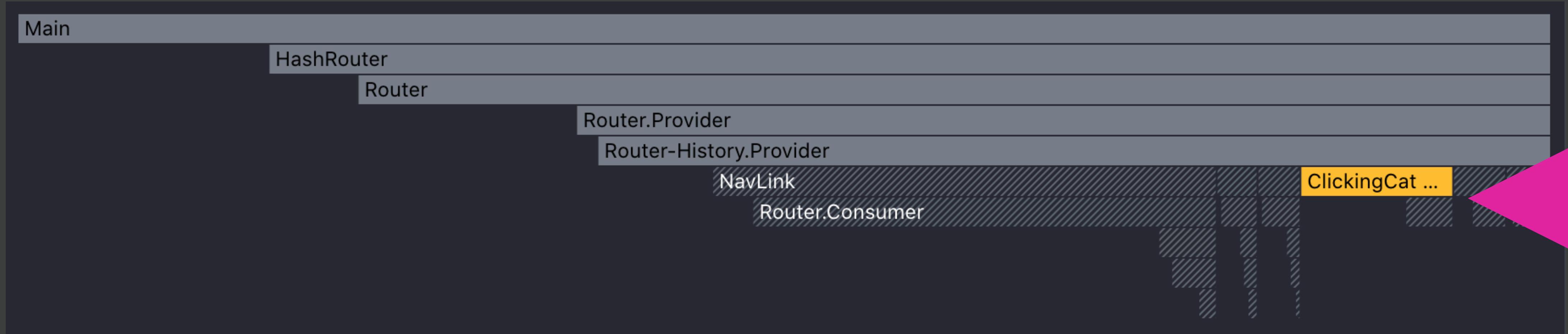


```
<Cat  
  name="Lulu"  
  description="A very lazy cat"  
  handleClick={() => console.log("clicked")}  
}></Cat>;
```

Neue Funktion bei rendering



Hat das was gebracht?



INLINE FUNCTIONS



Hahaha. Get it? Still no? Okay...



Schnell - UseCallback

```
export const ClickingCatFunctionComp = () => {
  ...
  const logHandler = useCallback(() => console.log("click"), []);
  return (
    <>
    ...
    <Cat
      name="Lulu"
      description="A very lazy cat"
      handleClick={() => this.logHandler}
    ></Cat>
    </>
  );
};

export default React.memo(ClickingCatFunctionComp);
```

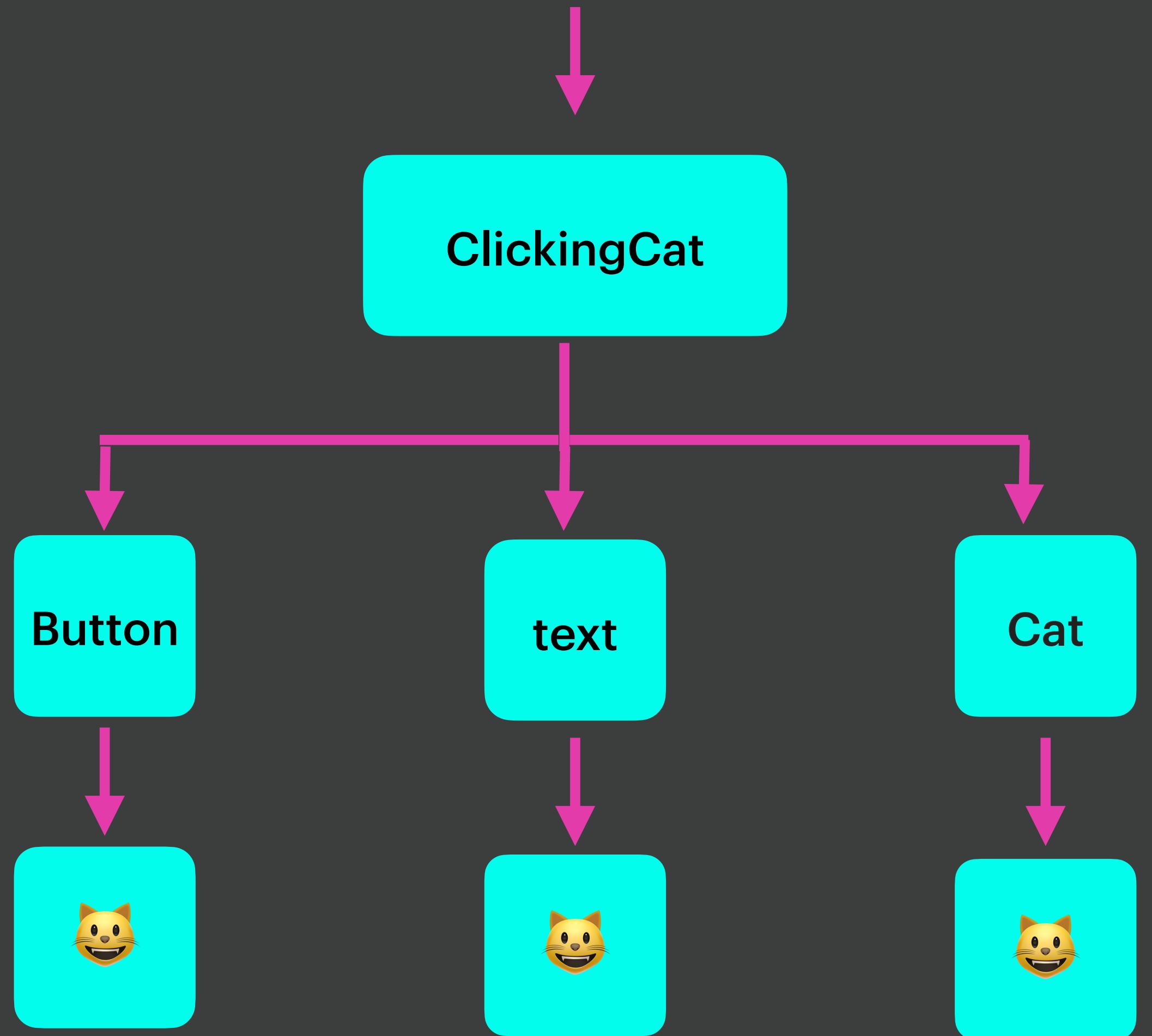


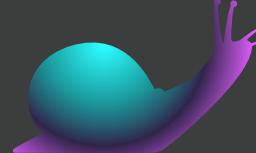
Speichert Funktion ref



Warum ist das schnell?

Update visibility of text





Wie mache ich das jetzt langsamer?

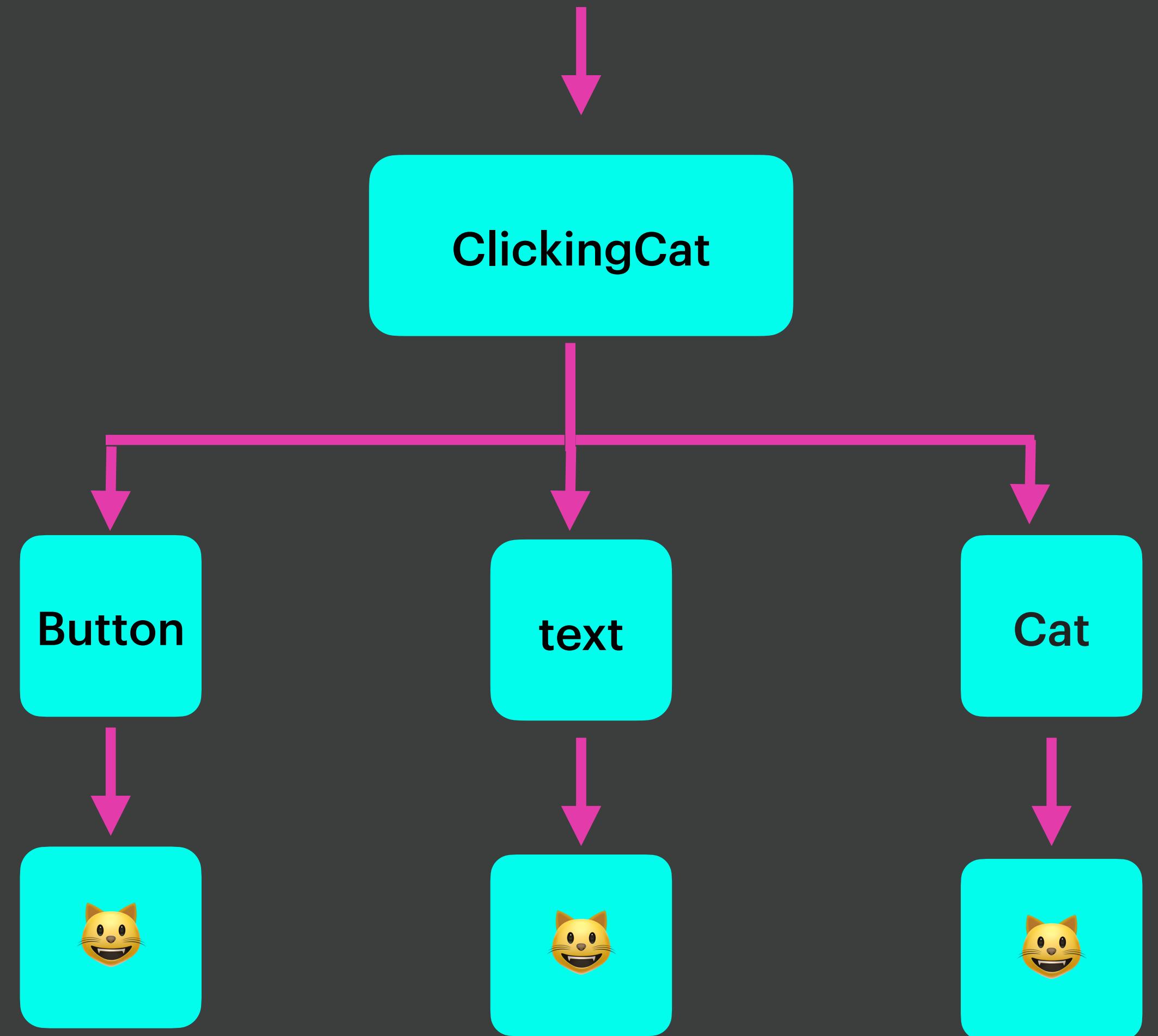
```
export const ClickingCatFunctionComp = () => {
  const [textIsVisible, setTextVisibility] = useState(true);

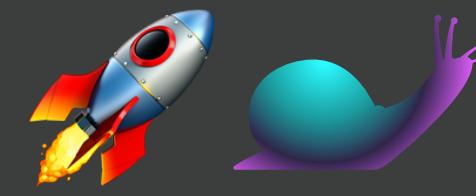
  return (
    <>
      <button onClick={() => setTextVisibility(!textIsVisible)}>
        {textIsVisible ? "hide text" : "showText"}
      </button>
      {textIsVisible && <p>text</p>}
      <Cat
        name="Lulu"
        description="A very lazy cat"
        handleClick={() => console.log("click")}
      ></Cat>
    </>
  );
};

export default React.memo(ClickingCatFunctionComp);
```

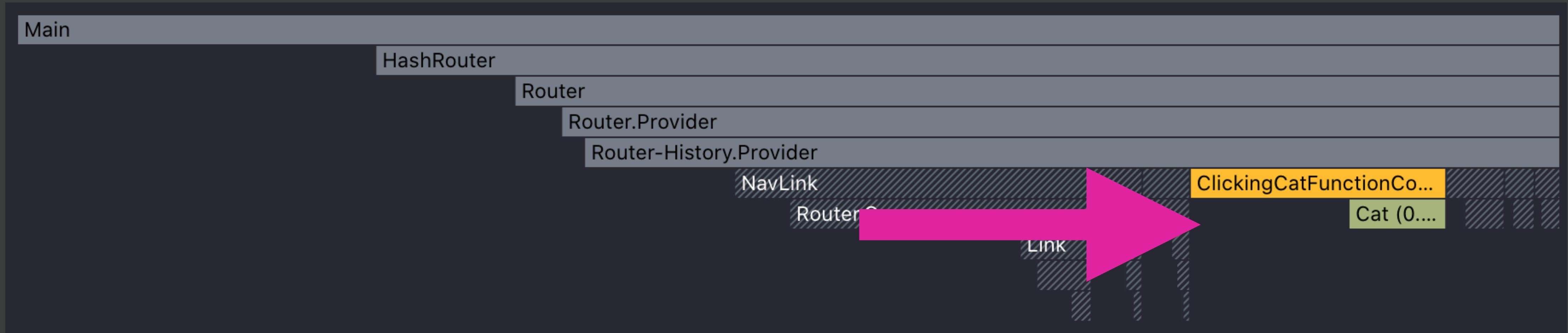
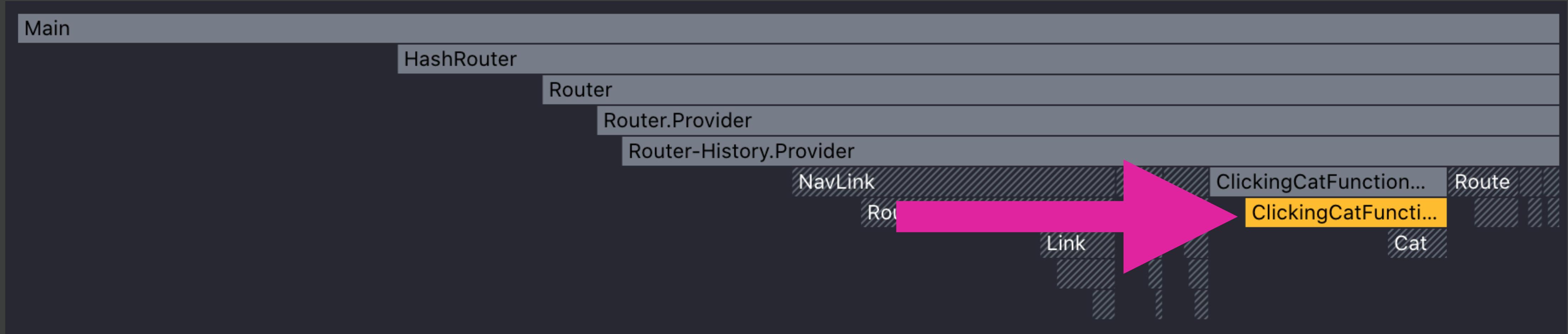
Warum macht das die SPA langsamer? 🐌

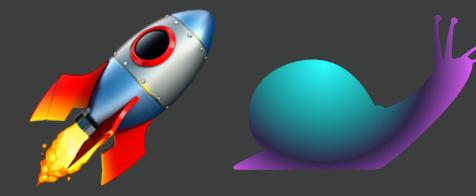
Update visibility of text



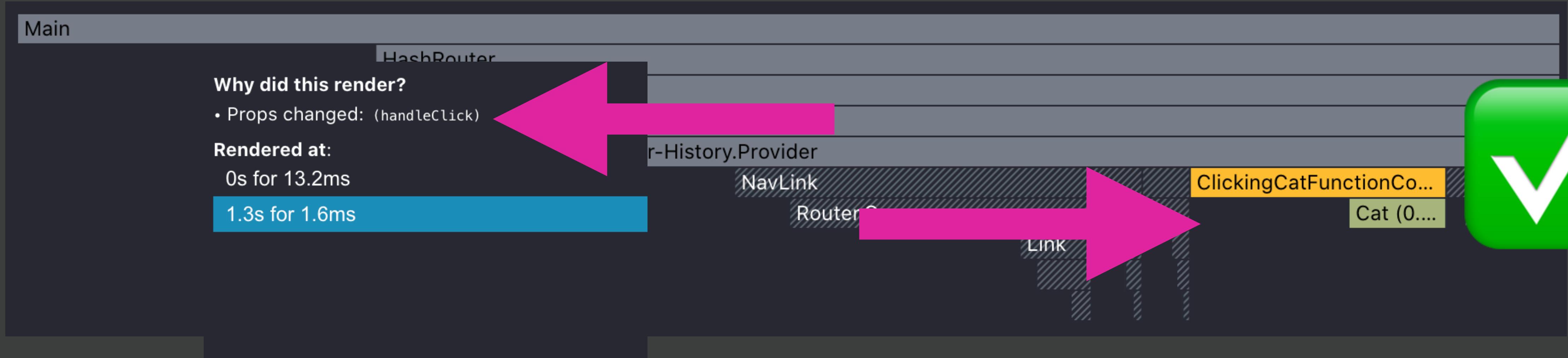
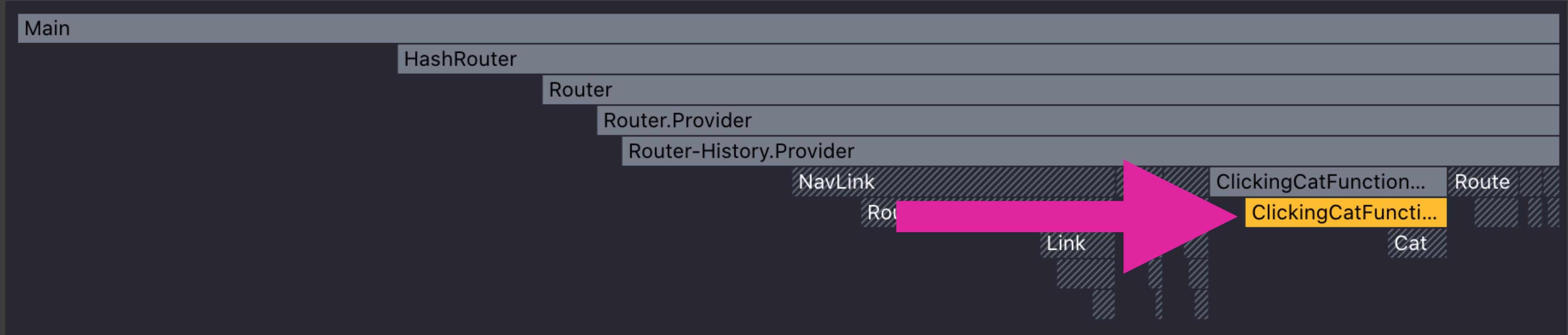


Hat das was gebracht?

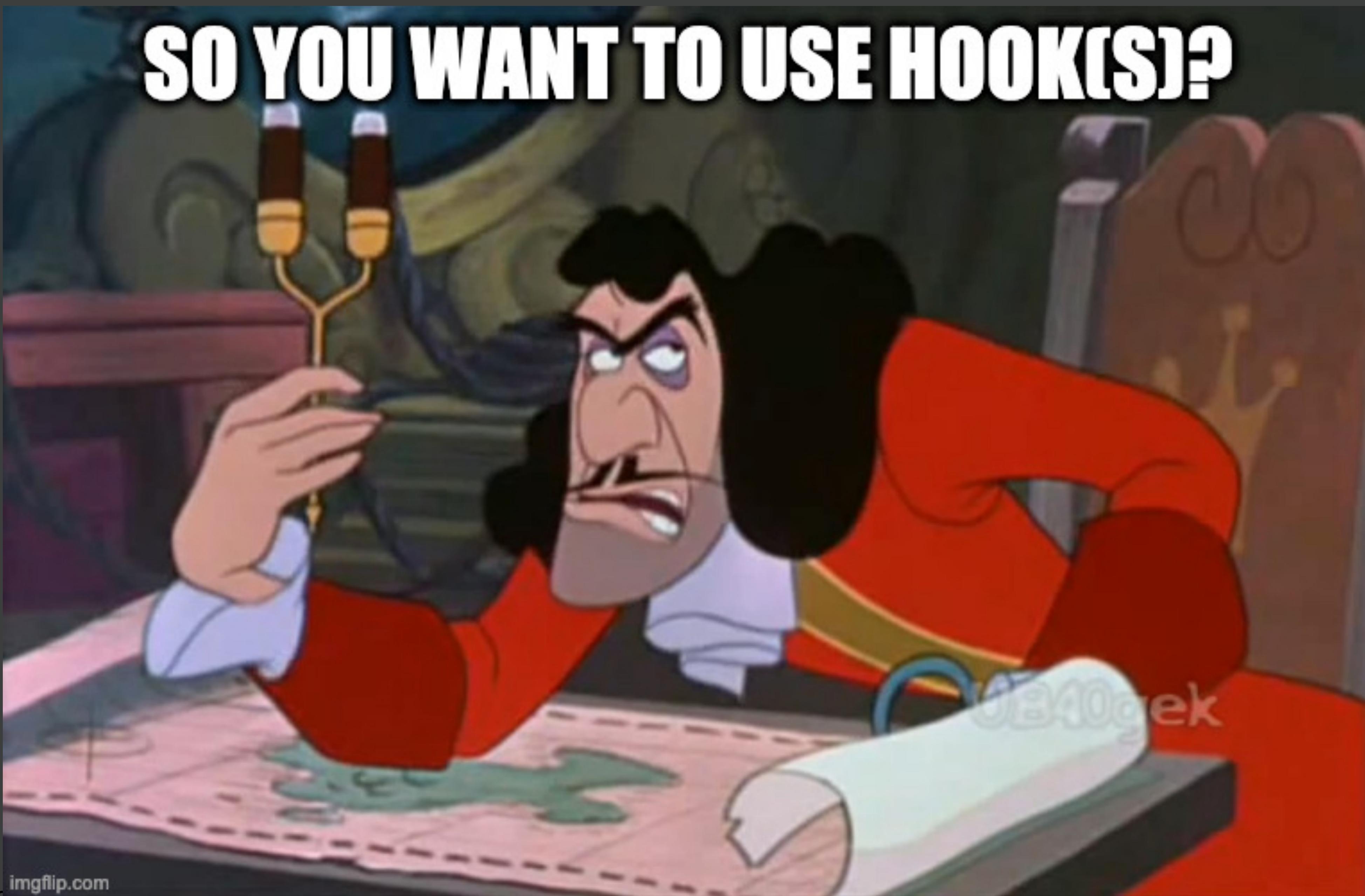




Hat das was gebracht?



SO YOU WANT TO USE HOOK(S)?





Schnell - Effects

```
export const MehTabWithEffect = () => {
  const [text] = useState("blubber");
  const [comment, setComment] = useState("my comment to the text");

  setInterval(() => setComment(comment === "miep" ? "test" : "miep"), 3000);
  return (
    <div>
      <h2>What is wrong with you?!</h2>
      <EffectComponent text={text} comment={comment}></EffectComponent>
    </div>
  );
};
```



Schnell - Effects

Update comment



```
const EffectComponent = ({ text, comment }) => {
  const [counter, setCounter] = useState(0);

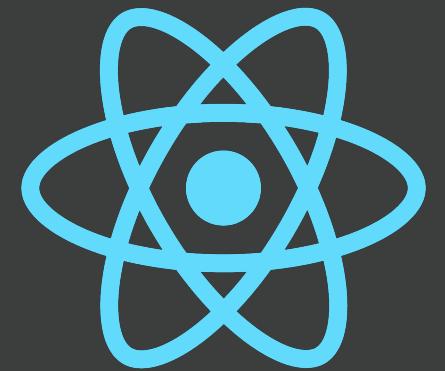
  useEffect(() => {
    setCounter(counter + 1);
  }, [text]);
  return (
    <p>
      {text} is rendererd {counter} times
      {comment}
    </p>
  );
};
```

What is wrong with you?!

blubber is rendererd 0 times my comment to the text



Warum ist das schnell?



`useEffect(callback, [dependencies])`



Aufruf

`useEffect(callback, [a, b])`

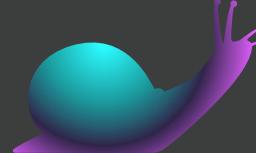
a,b

`useEffect(callback, [])`

einmal

`useEffect(callback)`

Immer



Wie mache ich das jetzt langsamer?

Update comment

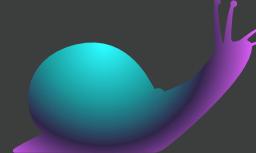


```
const EffectComponent = ({ text, comment }) => {
  const [counter, setCounter] = useState(0);

  useEffect(() => {
    setCounter(counter + 1);
  });
  return (
    <p>
      {text} is rendererd {counter} times
      {comment}
    </p>
  );
};
```

What is wrong with you?!

blubber is rendererd 1 timesmy comment to the text



Wie mache ich das jetzt langsamer?

Update comment

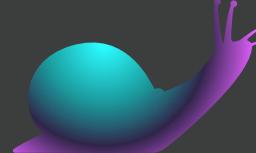


```
const EffectComponent = ({ text, comment }) => {
  const [counter, setCounter] = useState(0);

  useEffect(() => {
    setCounter(counter + 1);
  });
  return (
    <p>
      {text} is rendererd {counter} times
      {comment}
    </p>
  );
};
```

What is wrong with you?!

blubber is rendererd 2 times my comment to the text



Wie mache ich das jetzt langsamer?

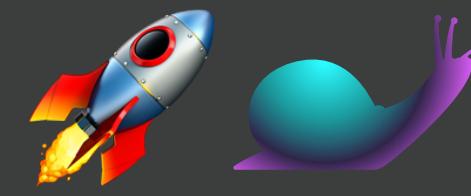
Update comment



```
const EffectComponent = ({ text, comment }) => {
  const [counter, setCounter] = useState(0);

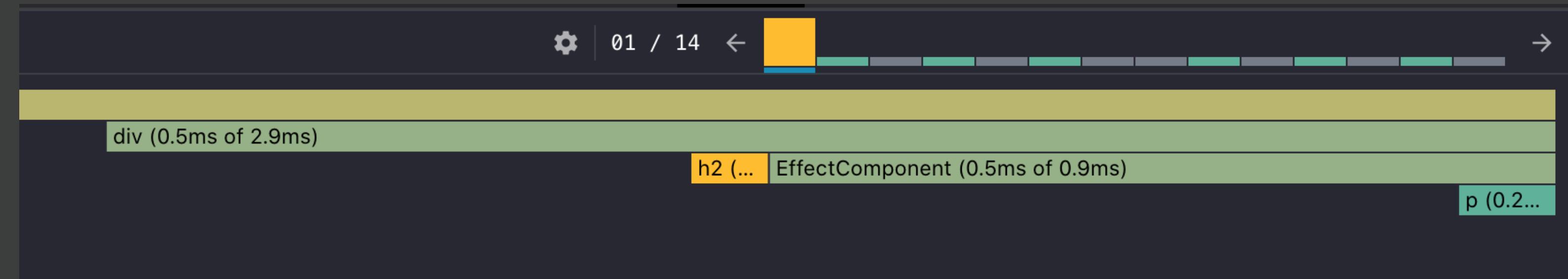
  useEffect(() => {
    setCounter(counter + 1);
  });
  return (
    <p>
      {text} is rendererd {counter} times
      {comment}
    </p>
  );
};
```

rendererd 2 times my

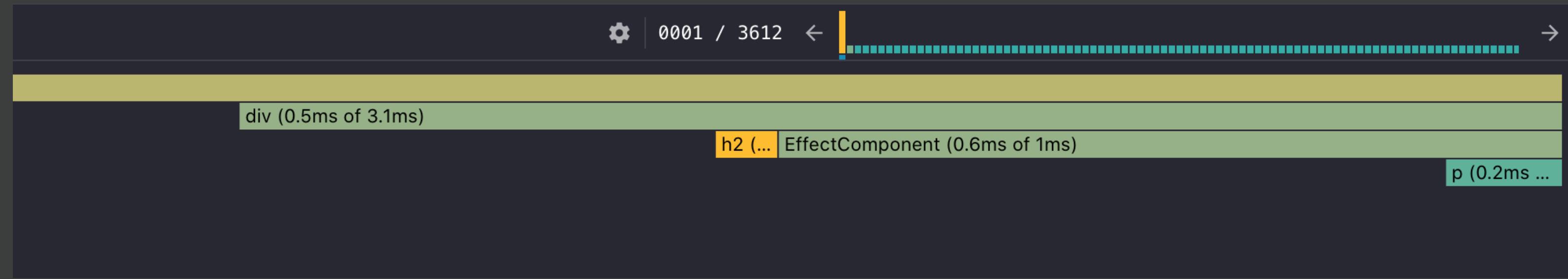


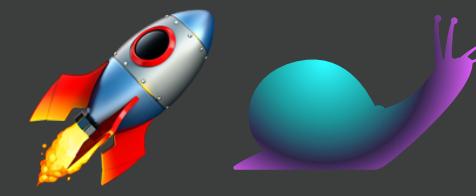
Hat das was gebracht?

`useEffect(callback, [])`



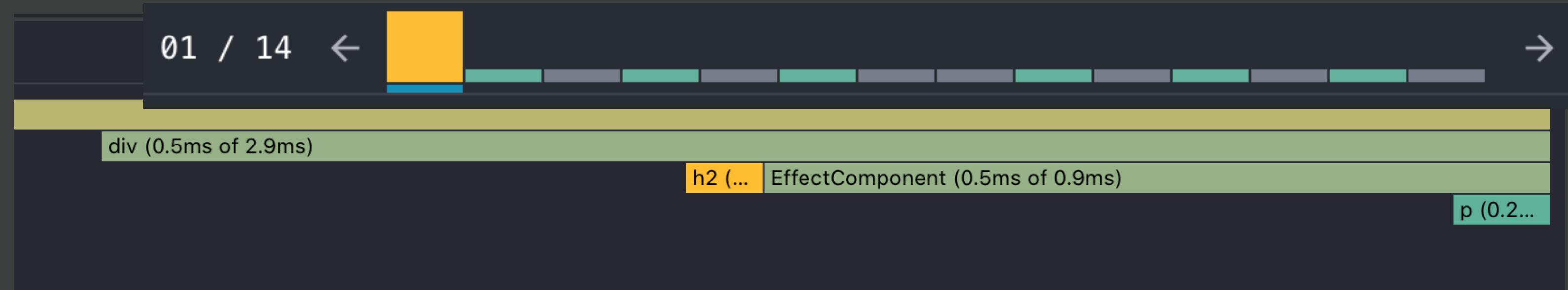
`useEffect(callback)`



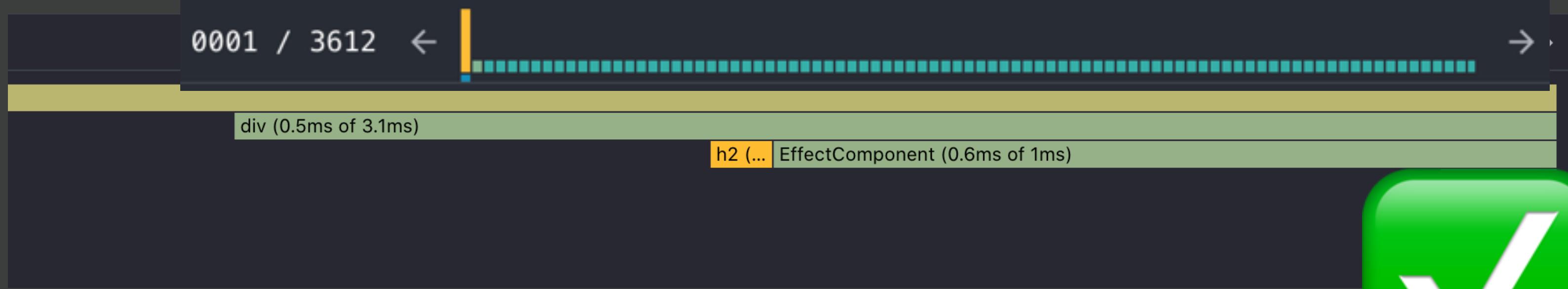


Hat das was gebracht?

`useEffect(callback, [])`



`useEffect(callback)`





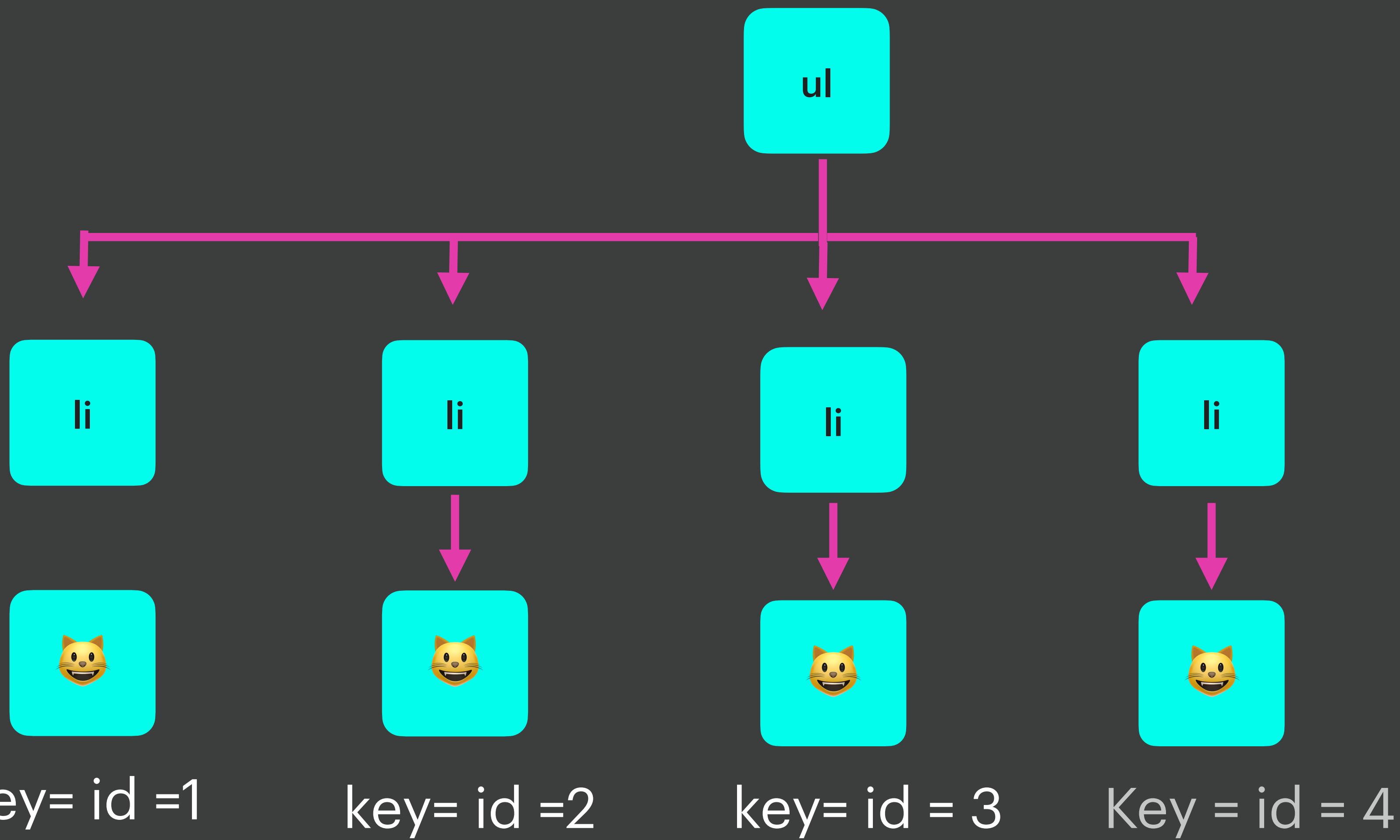


Schnell - key property

```
export const CatsList = () => {
  const [catList, setCatList] = useState(CATS);
  return (
    <ul>
      {catList.map((cat, index) => (
        <li key={cat.id}>
          <h2>{cat.name}</h2>
          <img src={cat.image} alt={cat.name}></img>
          <p>{cat.description}</p>
          <button
            onClick={() =>
              setCatList(catList.filter((catEntry) => catEntry !== cat))
            }
          >
            removeCat
          </button>
        </li>
      ))}
    </ul>
  );
};
```

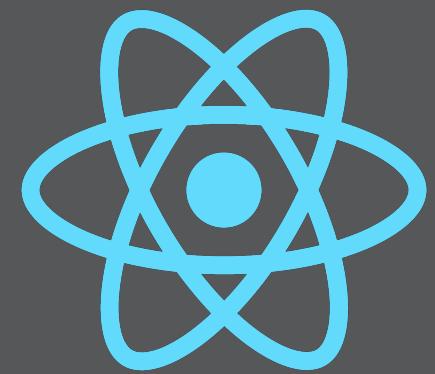


Warum ist das schnell?

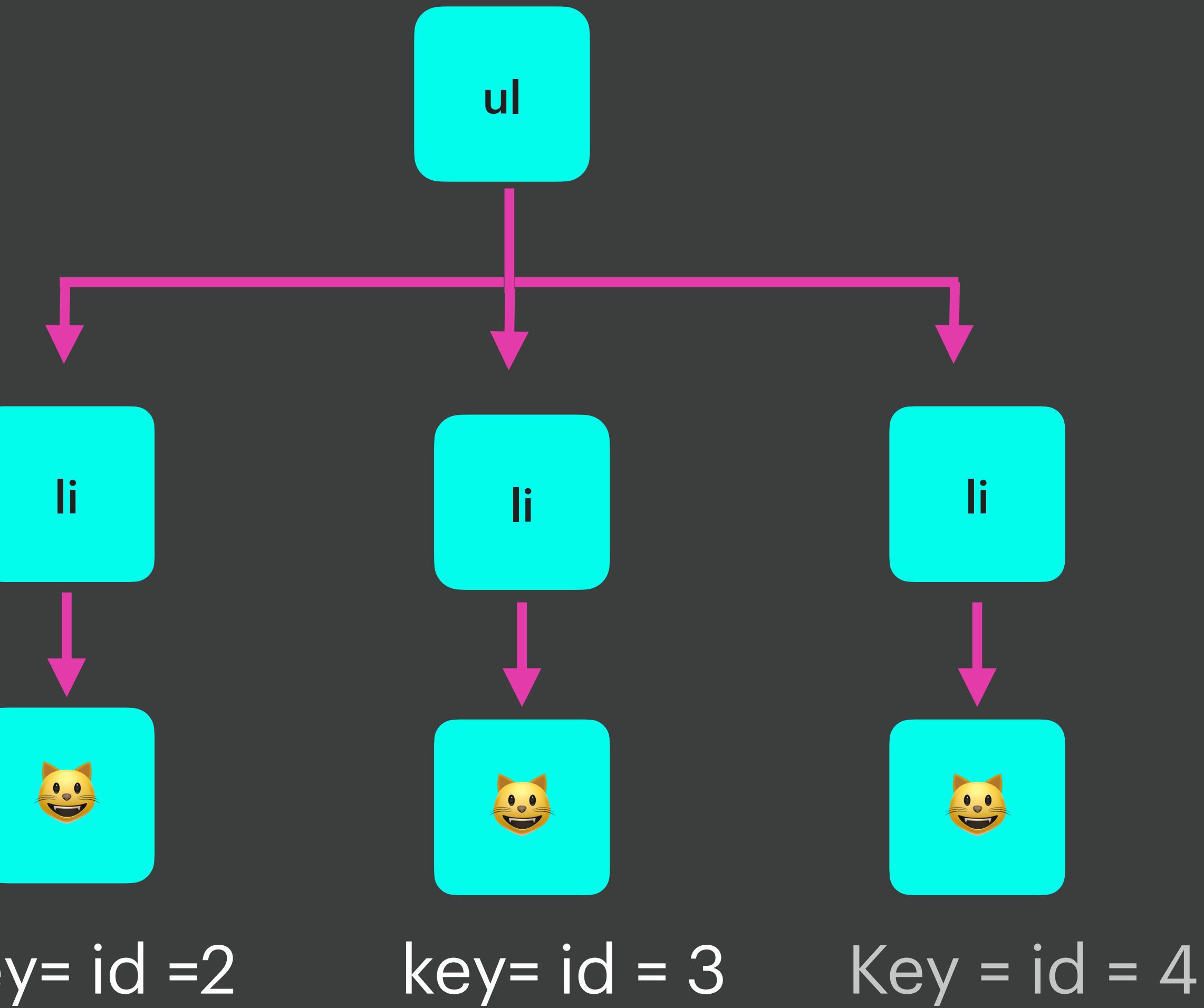




Warum ist das schnell?

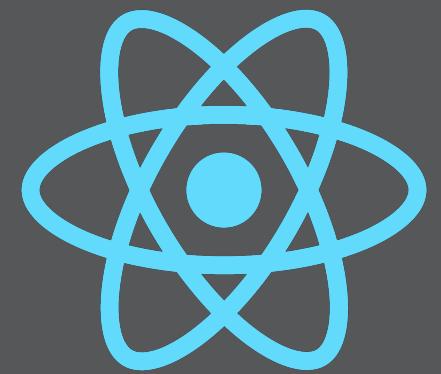


Re-Render:
Key != PreKey



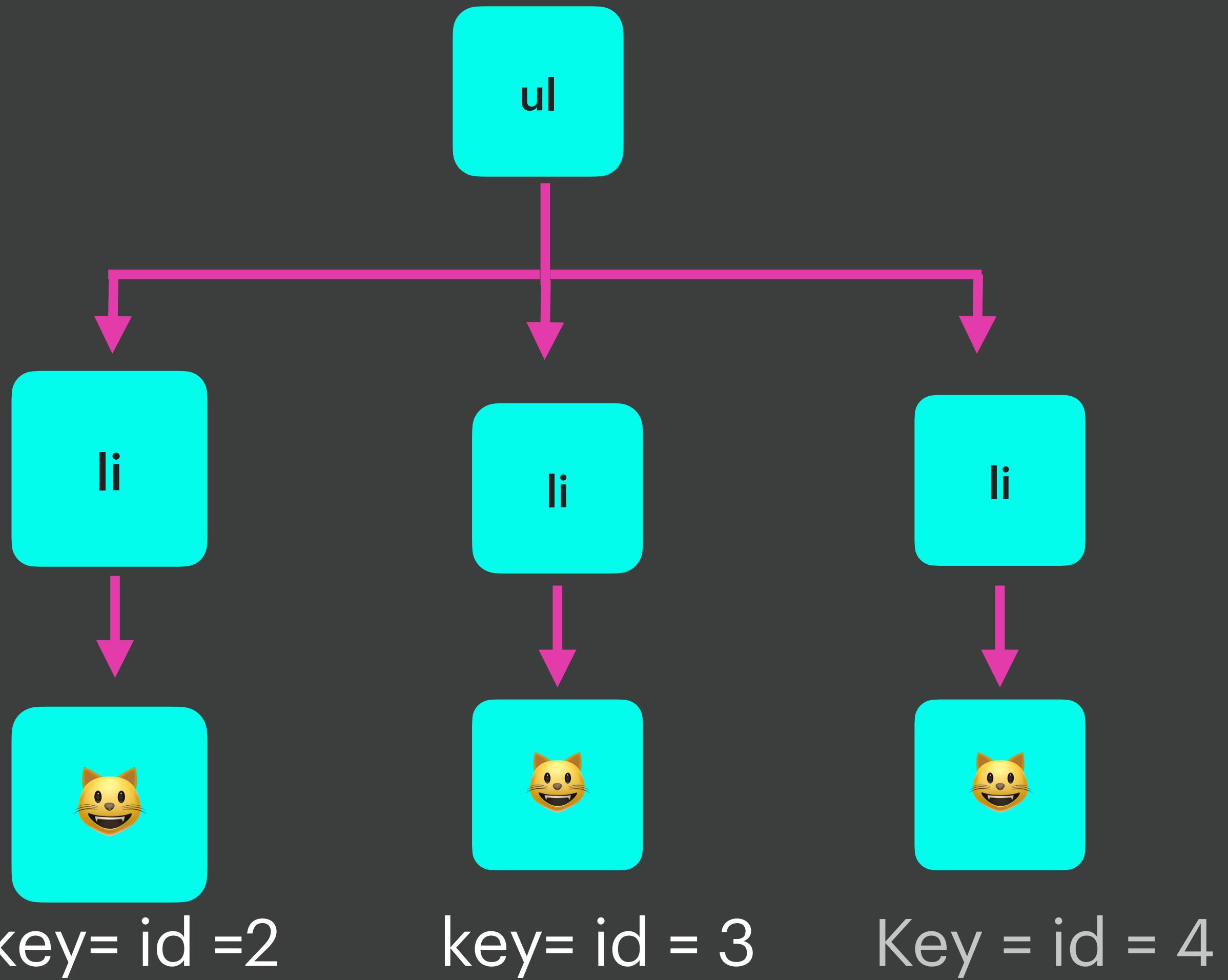


Warum ist das schnell?



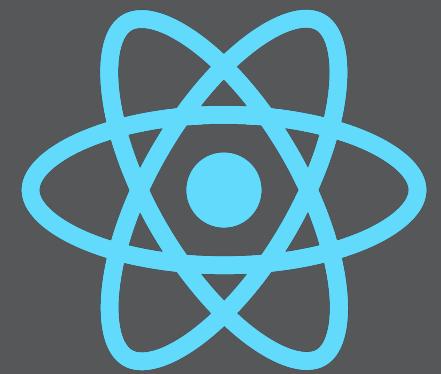
Re-Render:
Key != PreKey

key=2, old = 2
2 === 2



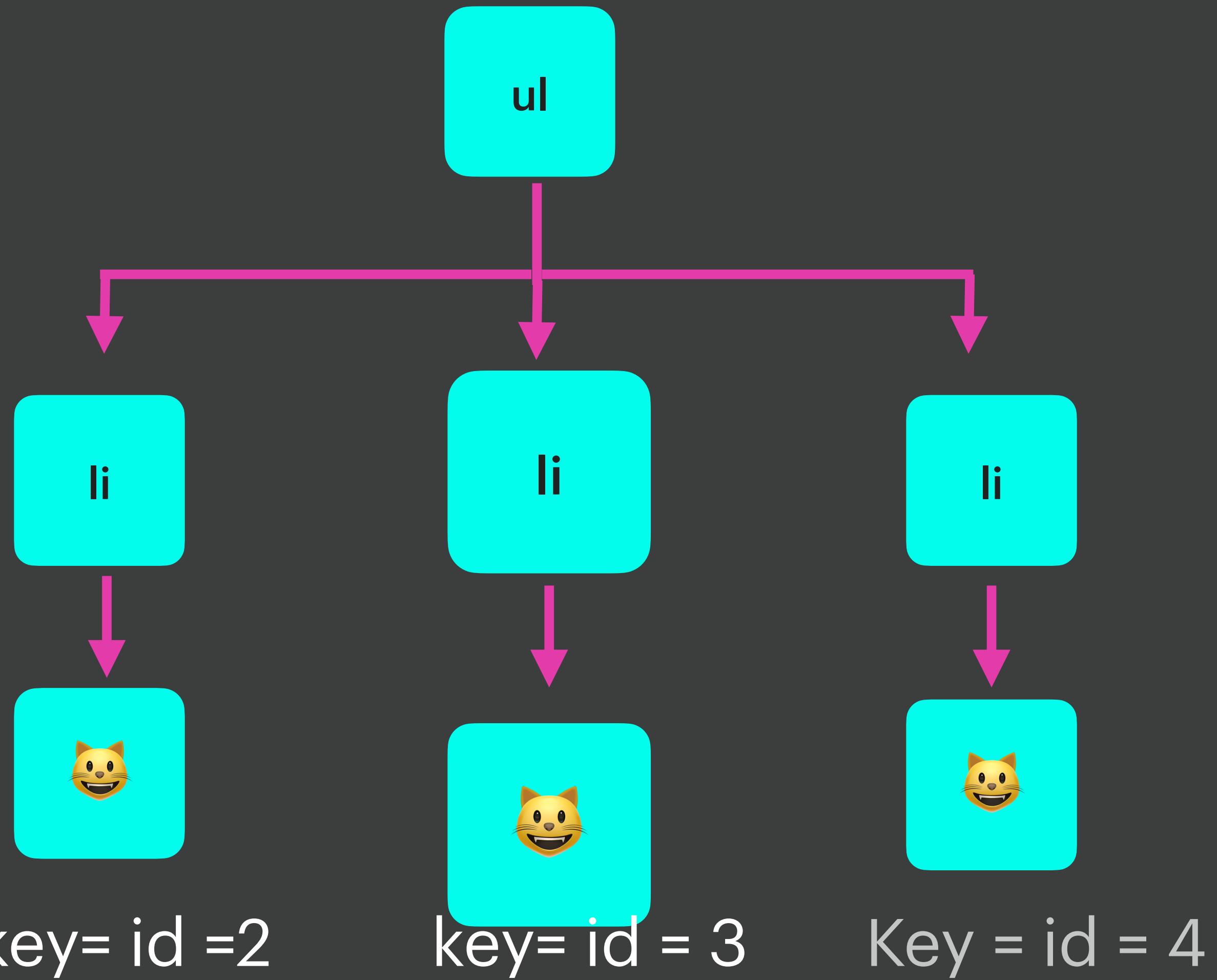


Warum ist das schnell?



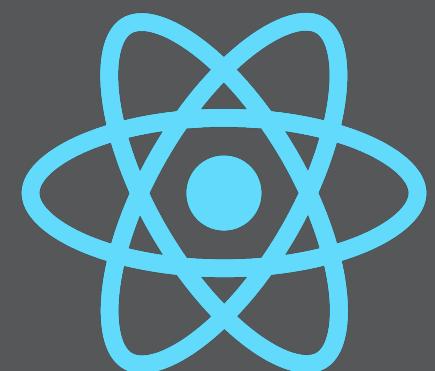
Re-Render:
Key != PreKey

key=3, old = 3
3 === 3





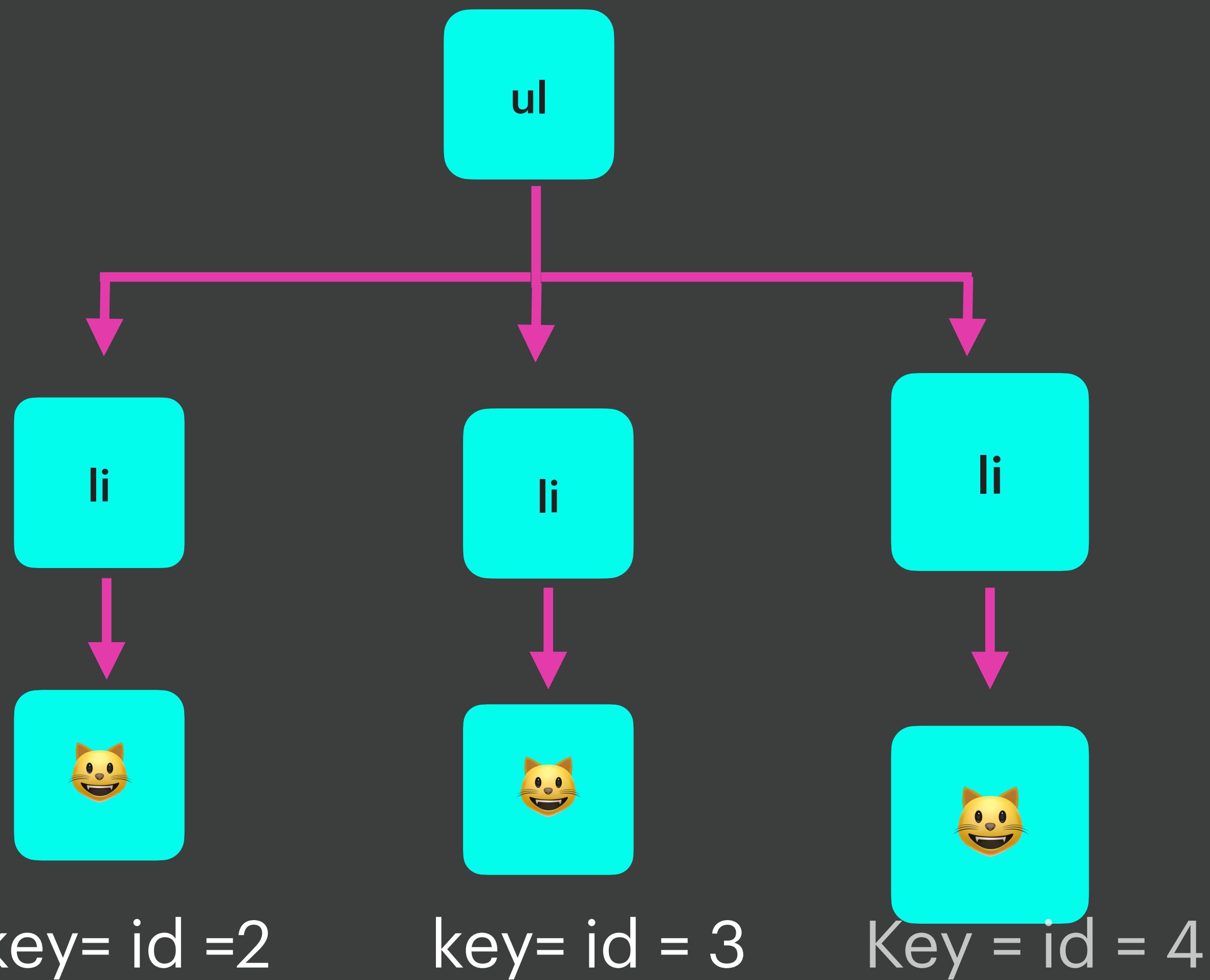
Warum ist das schnell?

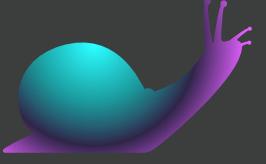


Re-Render:
Key != PreKey

key=4, old = 4

4== 4

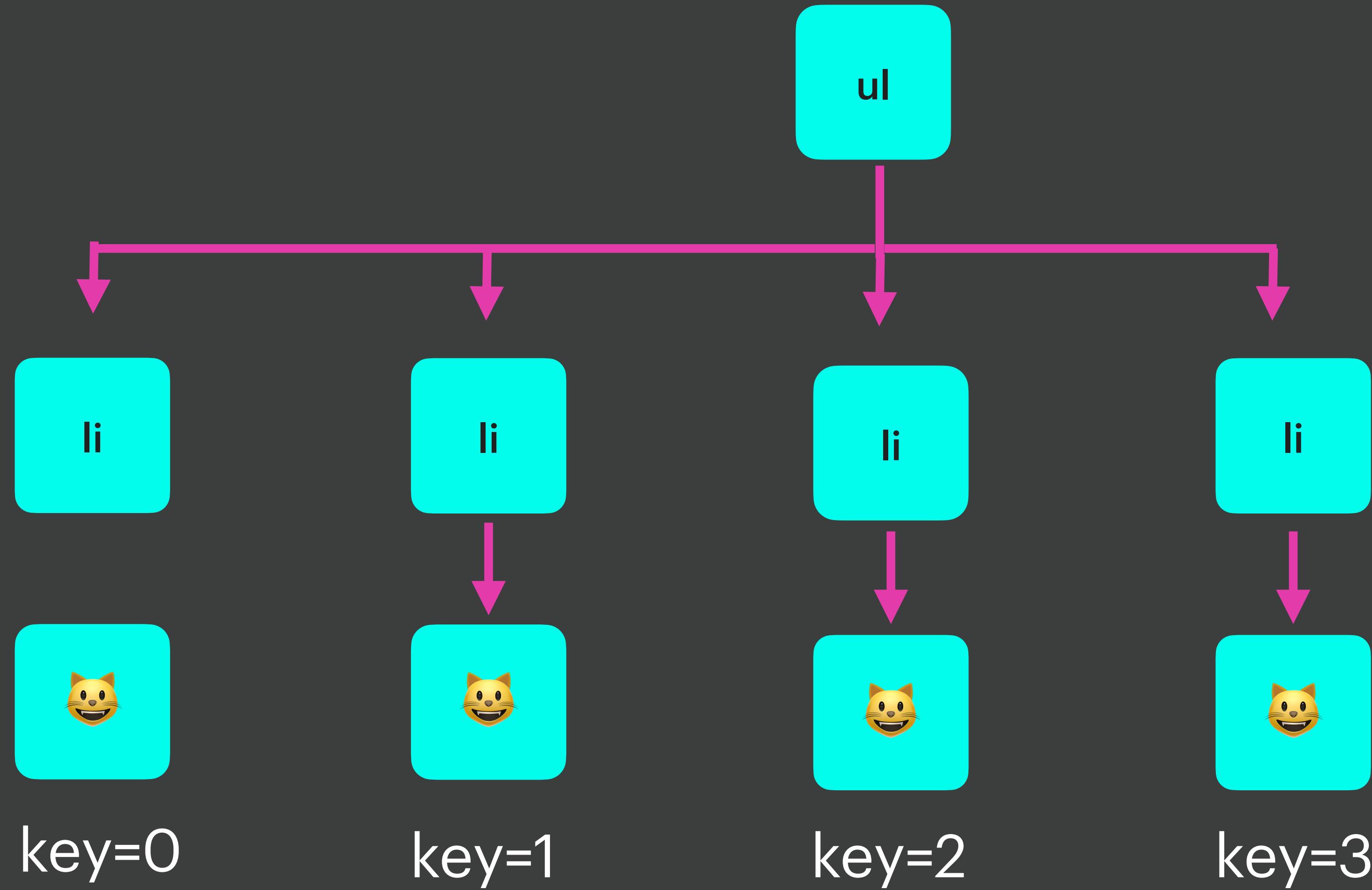




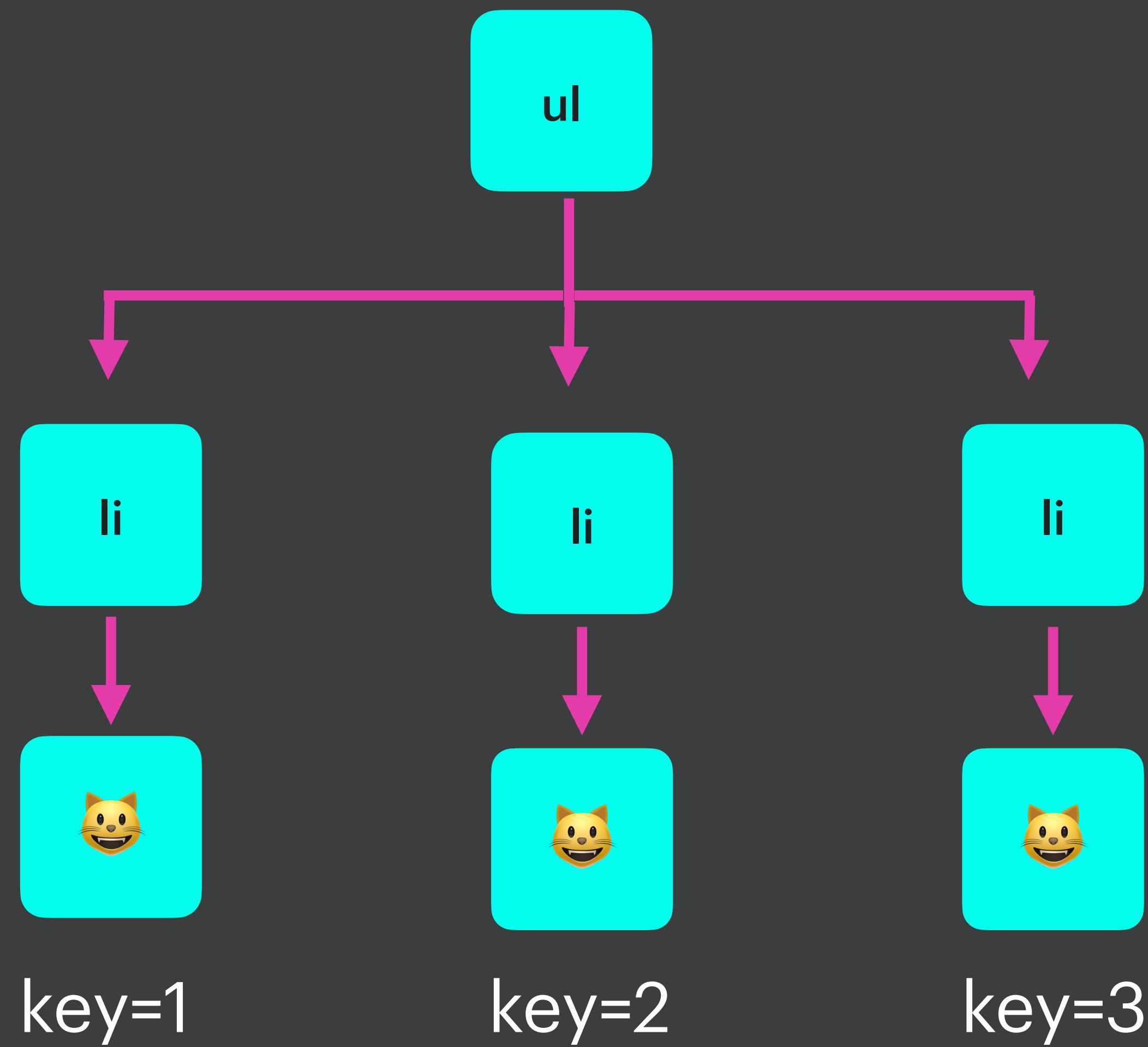
Wie mache ich das jetzt langsamer?

```
export const CatsList = () => {
  const [catList, setCatList] = useState(CATS);
  return (
    <ul>
      {catList.map((cat, index) => (
        <li key={index}>
          <h2>{cat.name}</h2>
          <img src={cat.image} alt={cat.name}></img>
          <p>{cat.description}</p>
          <button
            onClick={() =>
              setCatList(catList.filter((catEntry) => catEntry !== cat))
            }
          >
            removeCat
          </button>
        </li>
      ))}
    </ul>
  );
};
```

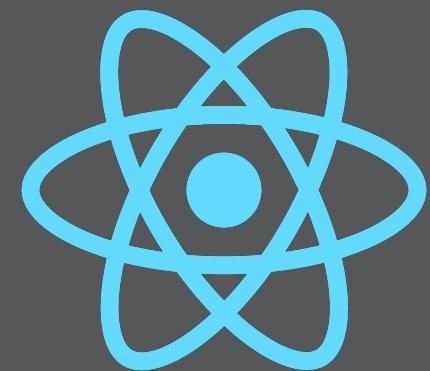
🐌 Wie mache ich das jetzt langsamer?



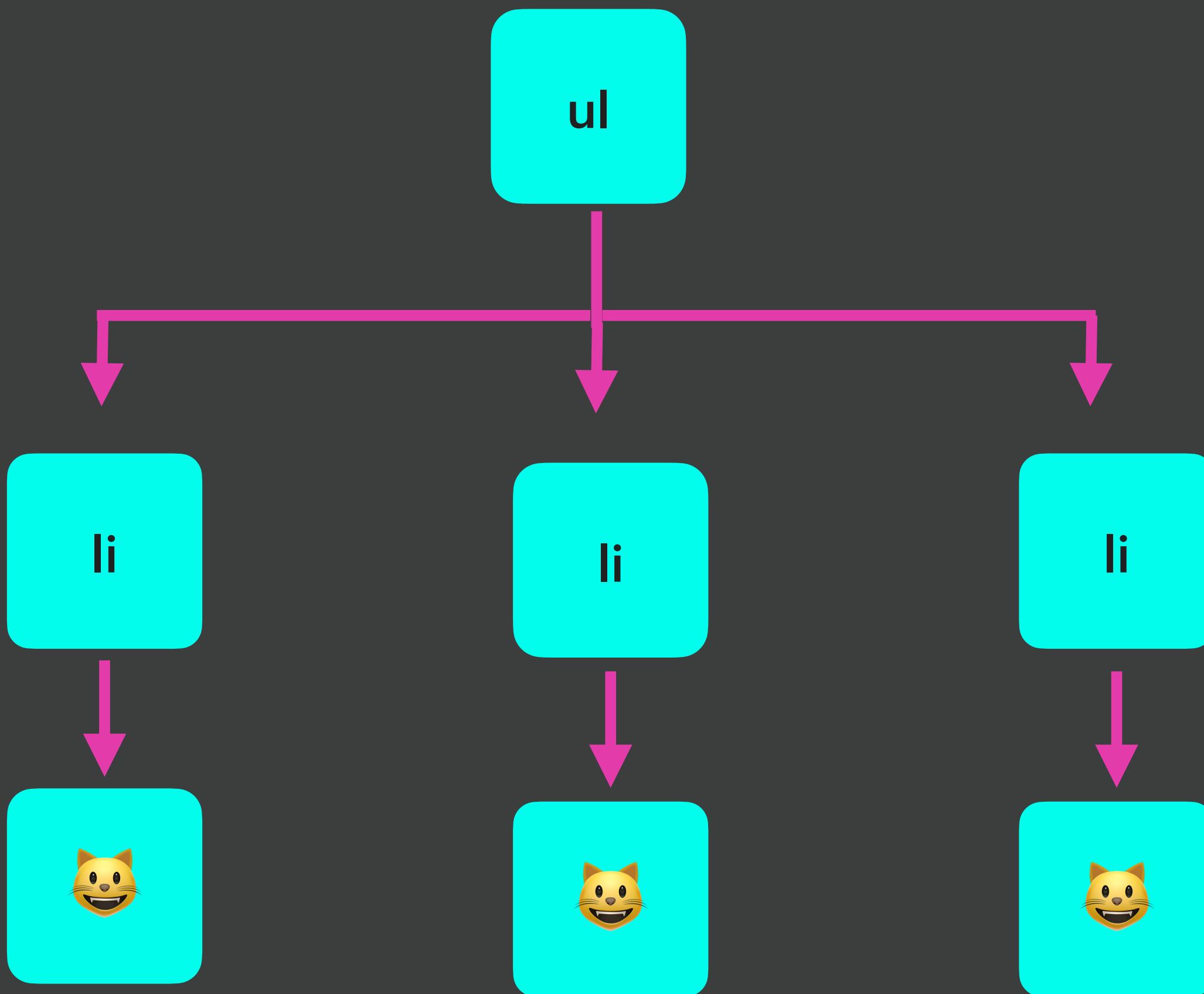
🐌 Wie mache ich das jetzt langsamer?



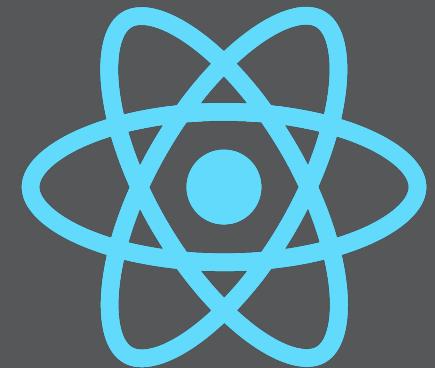
🐌 Wie mache ich das jetzt langsamer?



Re-Render:
Key != PreKey

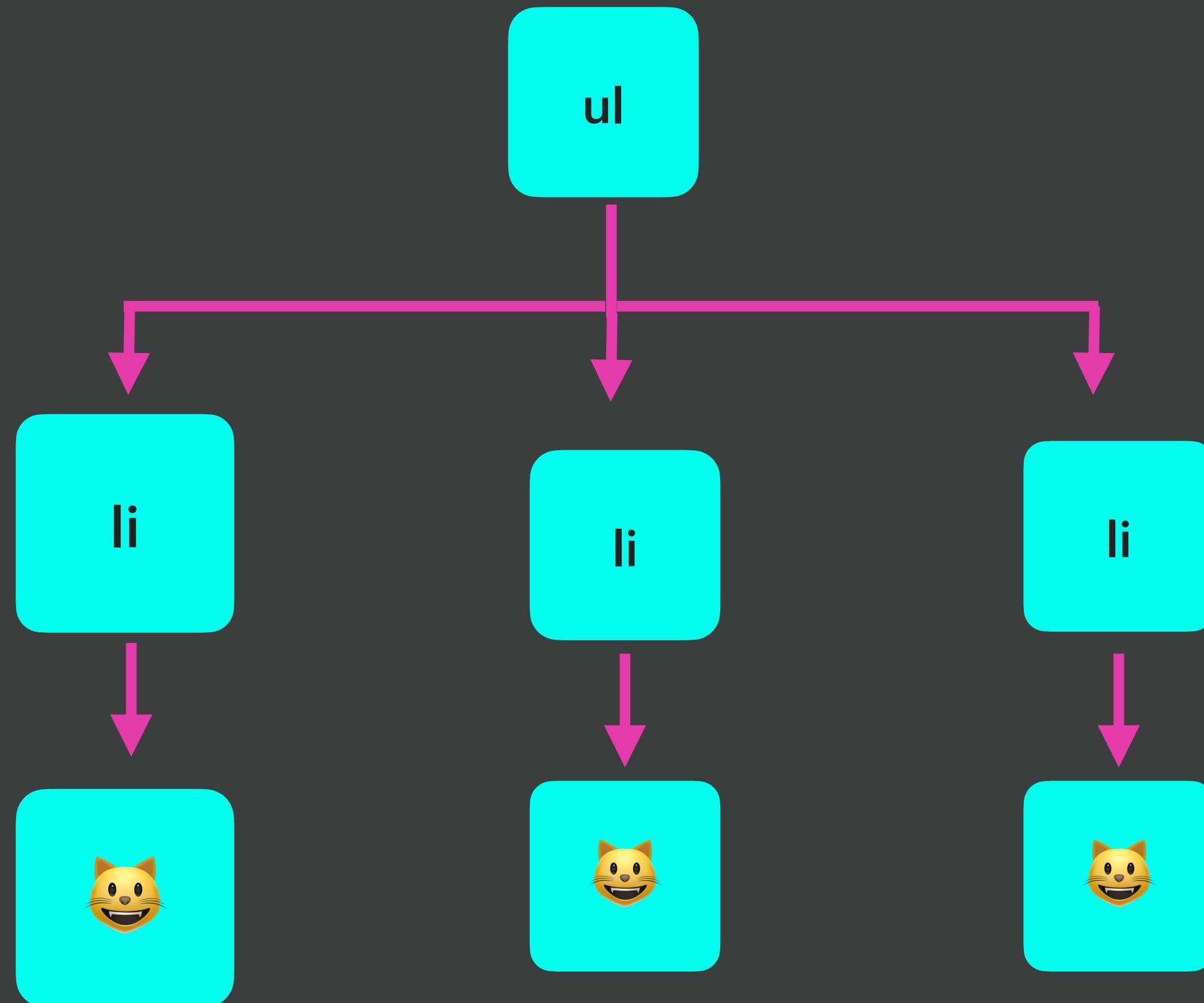


🐌 Wie mache ich das jetzt langsamer?

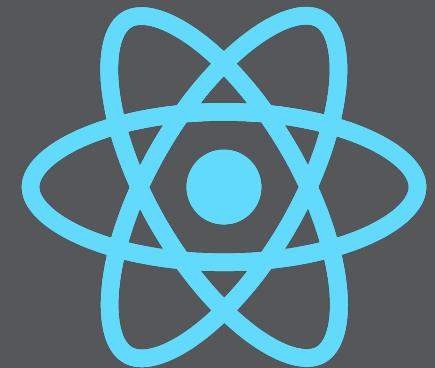


Re-Render:
Key != PreKey

key=0, old = 1
0 === 1

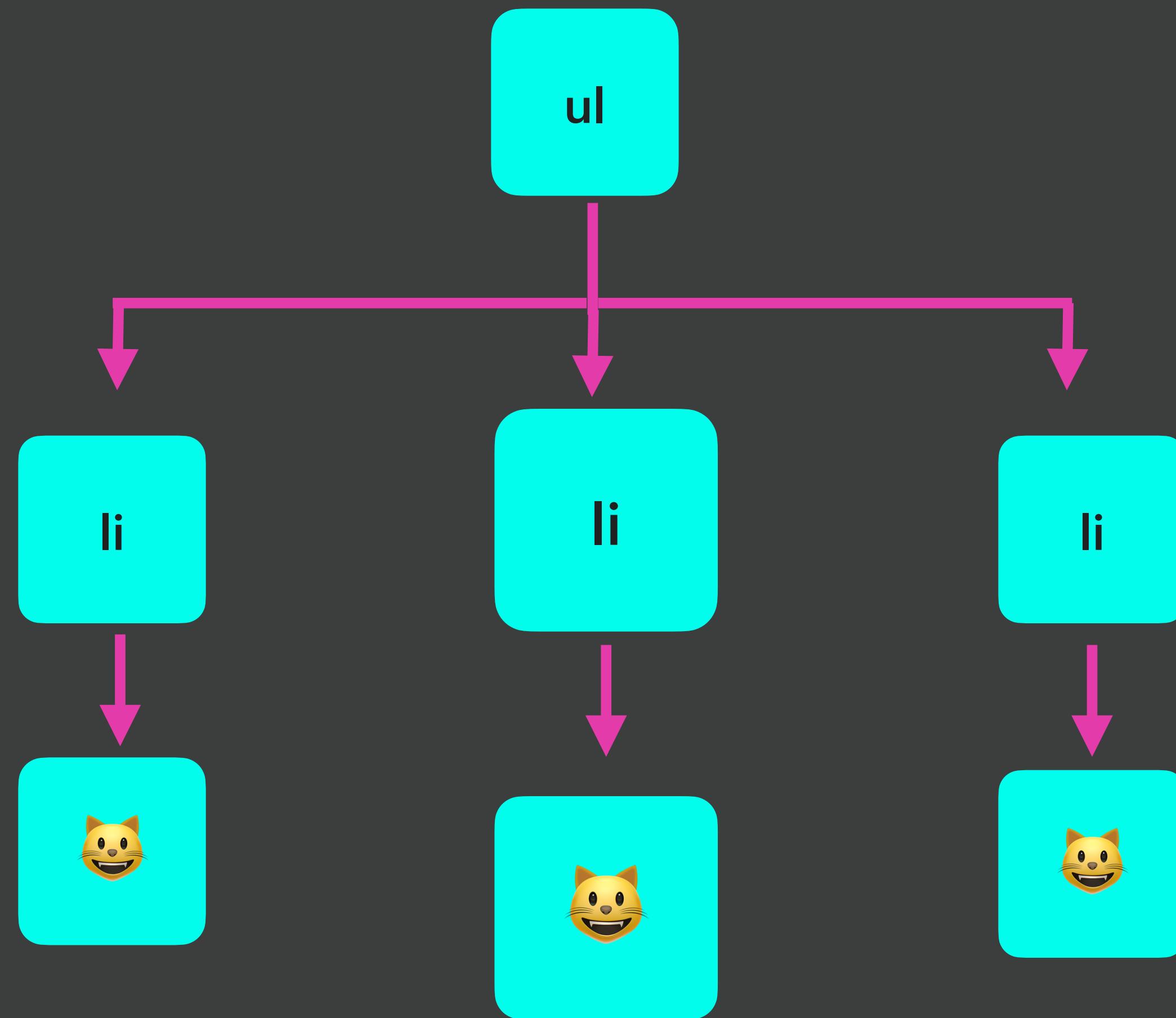


🐌 Wie mache ich das jetzt langsamer?

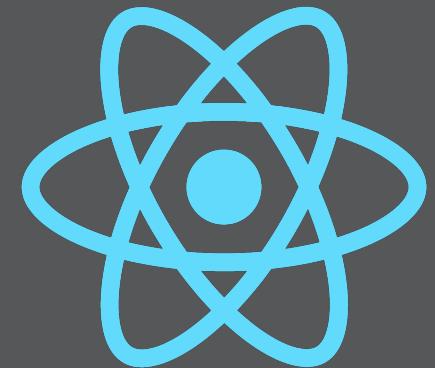


Re-Render:
Key != PreKey

key=1, old = 2
1 === 2

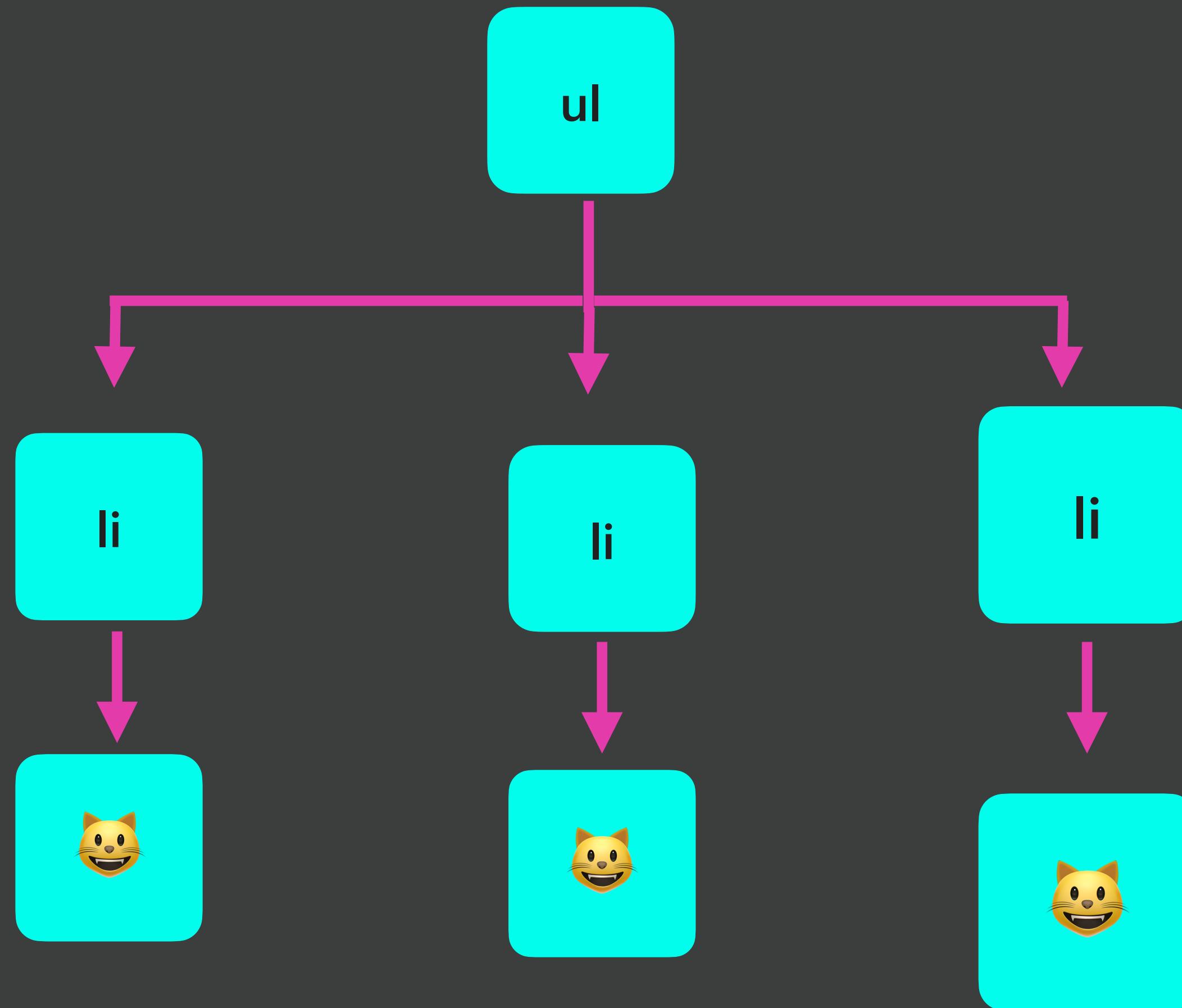


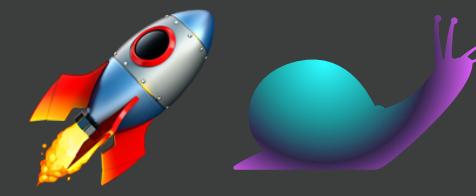
🐌 Wie mache ich das jetzt langsamer?



Re-Render:
Key != PreKey

key=2, old =3
2 === 3



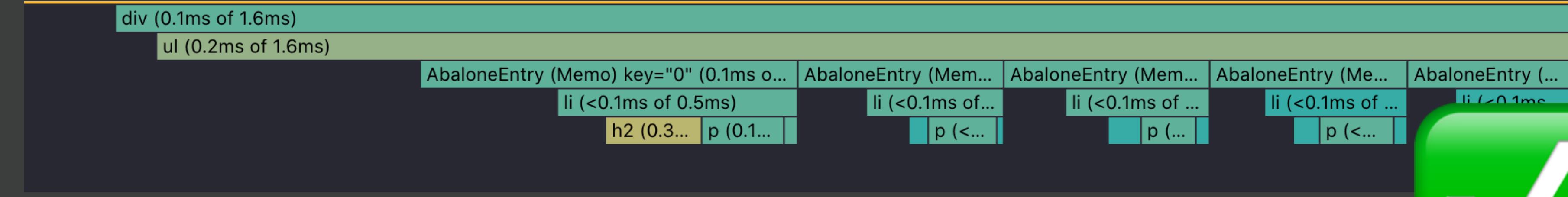


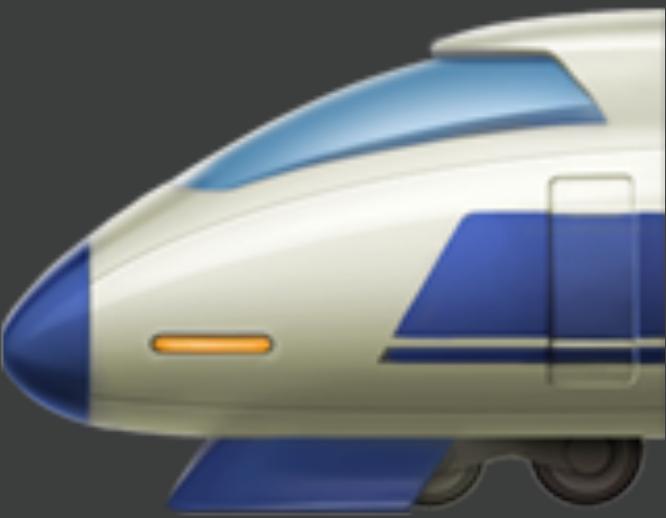
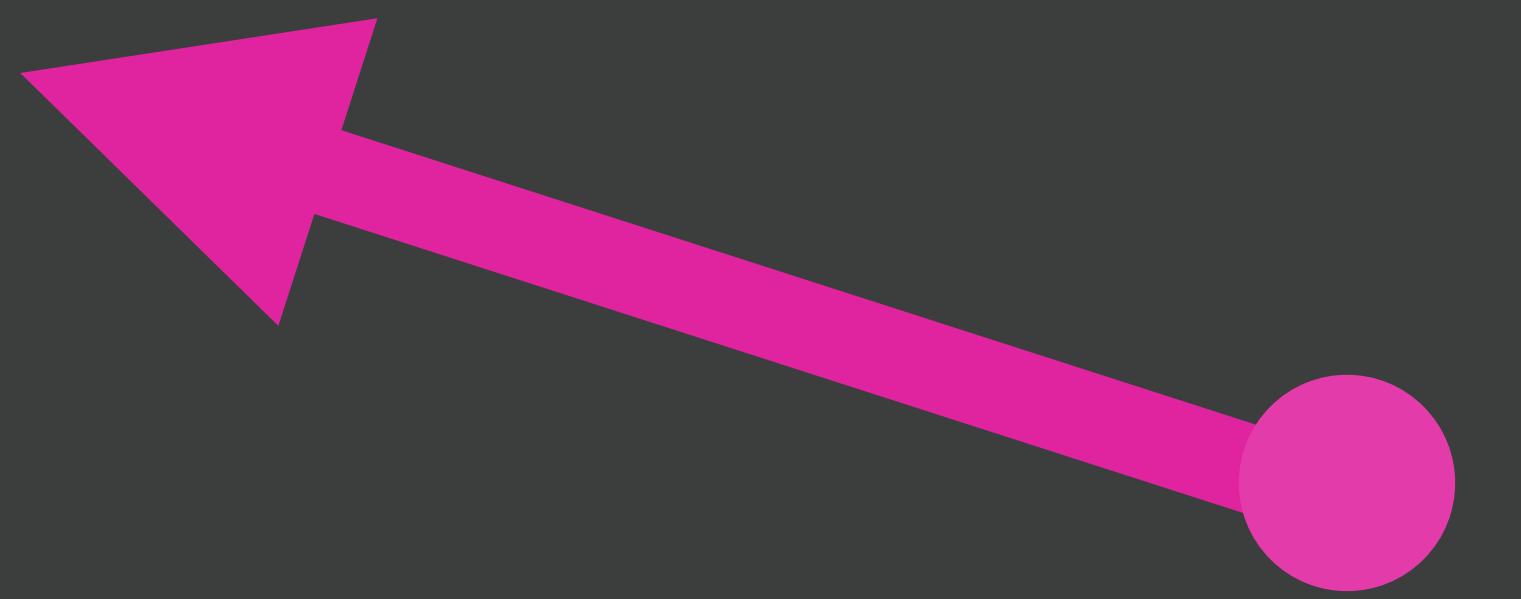
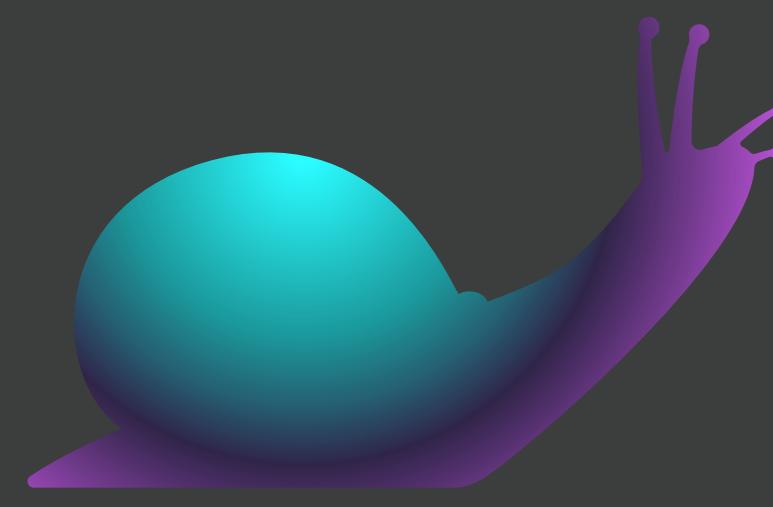
Hat das was gebracht?

Key = id
+
Memo



Key = index





EVERY GROUP PROJECT



IN SCHOOL YOU HAVE EVER DONE

THE END



FRAGEN?

Sprich uns an! :)



CHRISTINA
○ merelyChristina



ANNA
○ merelyAnna