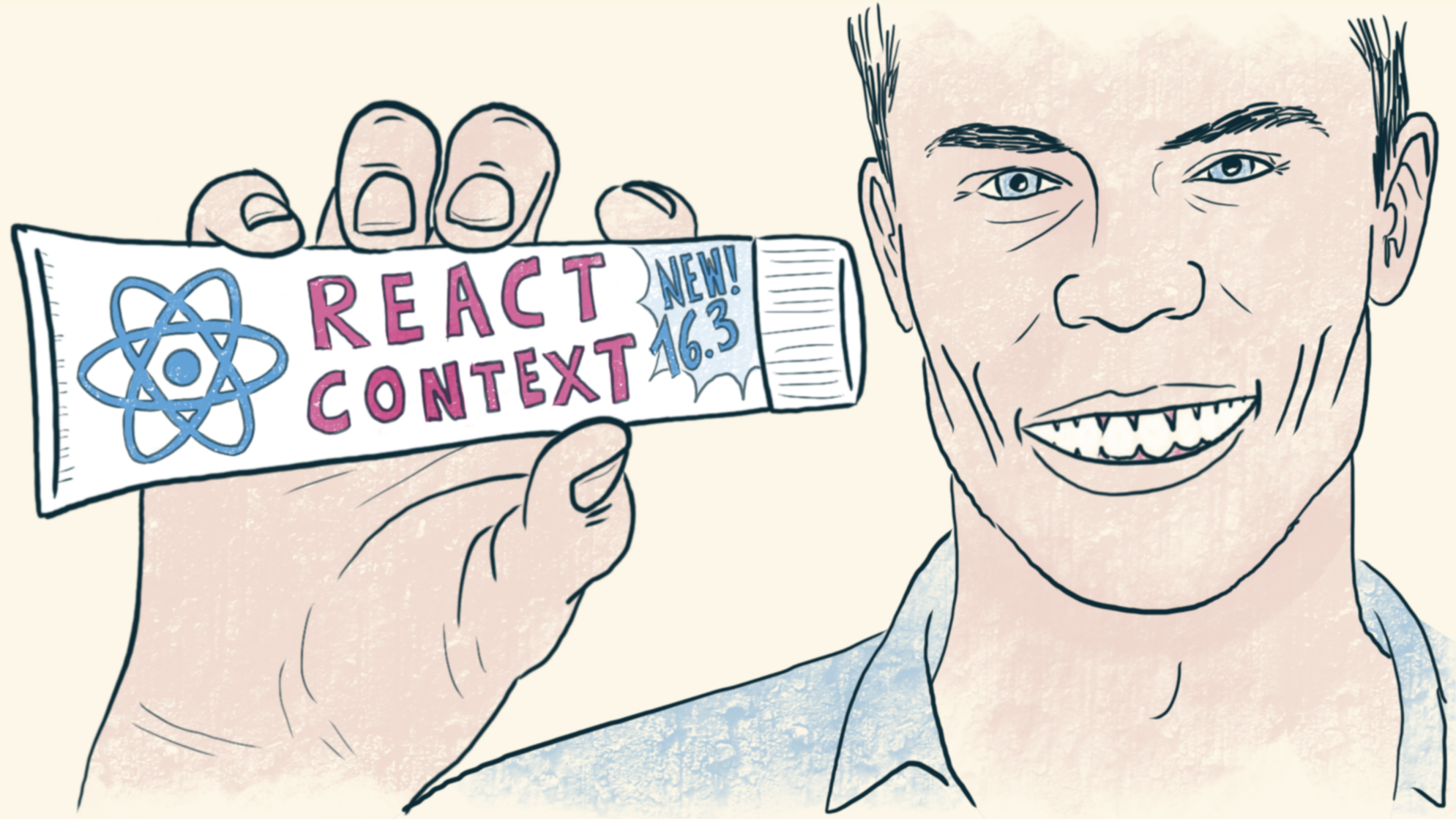
GET RICH QUICK WITH...



Patrick Hund | frontend developer at eBay Tech Berlin | @wiekatz

JS App.js src

```
import React from 'react';
import './App.css';
import config from './config';
import { AdvertisingProvider, AdvertisingSlot } from './react-pr

function App() {
  return (
    <div className="App">
      <header className="App-header">
        <h1 className="App-title">Context</h1>
      </header>
      <article className="App-article">
        <p>
          <strong>
            Context provides a way to pass data through the co
            without having to pass props down manually at ever
          </strong>
        </p>
        <p>
          In a typical React application, data is passed top-d
          child) via props, but this can be cumbersome for cer
          props (e.g. locale preference, UI theme) that are re
          components within an application. Context provides a
          values like this between components without having t
          pass a prop through every level of the tree.
        </p>
        <h2>When to Use Context</h2>
        <p>
          Context is designed to share data that can be consid
          for a tree of React components, such as the current
          user, theme, or preferred language.
```

https://5v34q9j9ln.codesandbox.io/

Context

Context provides a way to pass data through the component tree without having to pass props down manually at every level.

In a typical React application, data is passed top-down (parent to child) via props, but this can be cumbersome for certain types of props (e.g. locale preference, UI theme) that are required by many components within an application. Context provides a way to share values like this between components without having to explicitly pass a prop through every level of the tree.

When to Use Context

Context is designed to share data that can be considered “global” for a tree of React components, such as the current authenticated user, theme, or preferred language.

Note

Don’t use context just to avoid passing props a few levels down. Stick to cases where the same data needs to be accessed in many components at multiple levels.

Console Problems Tests

JS App.js src

```
13        <article className="App-article">
14          <p>
15            <strong>
16              Context provides a way to pass data through the co
17              without having to pass props down manually at ever
18            </strong>
19          </p>
20          <p>
21            In a typical React application, data is passed top-d
22            child) via props, but this can be cumbersome for cer
23            props (e.g. locale preference, UI theme) that are re
24            components within an application. Context provides a
25            values like this between components without having t
26            pass a prop through every level of the tree.
27          </p>
28          <h2>When to Use Context</h2>
29          <p>
30            Context is designed to share data that can be consid
31            for a tree of React components, such as the current
32            user, theme, or preferred language.
33          </p>
34          <p>
35            <strong>Note</strong>
36          </p>
37          <p>
38            Don’t use context just to avoid passing props a few
39            Stick to cases where the same data needs to be acces
40            components at multiple levels.
41          </p>
42        </article>
43        <
```

https://5v34q9j9ln.codesandbox.io/

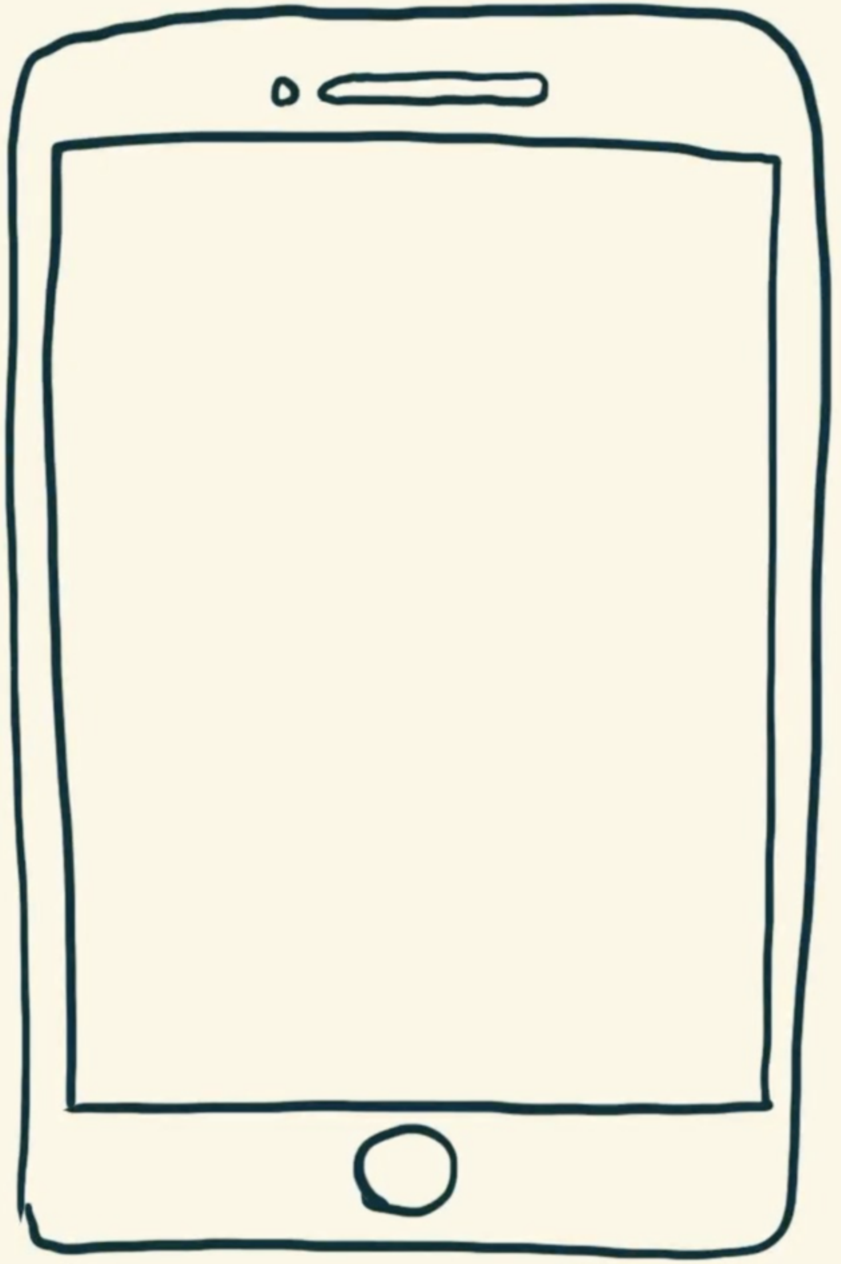
SyntaxError

/src/App.js: Unterminated JSX contents (43:10)

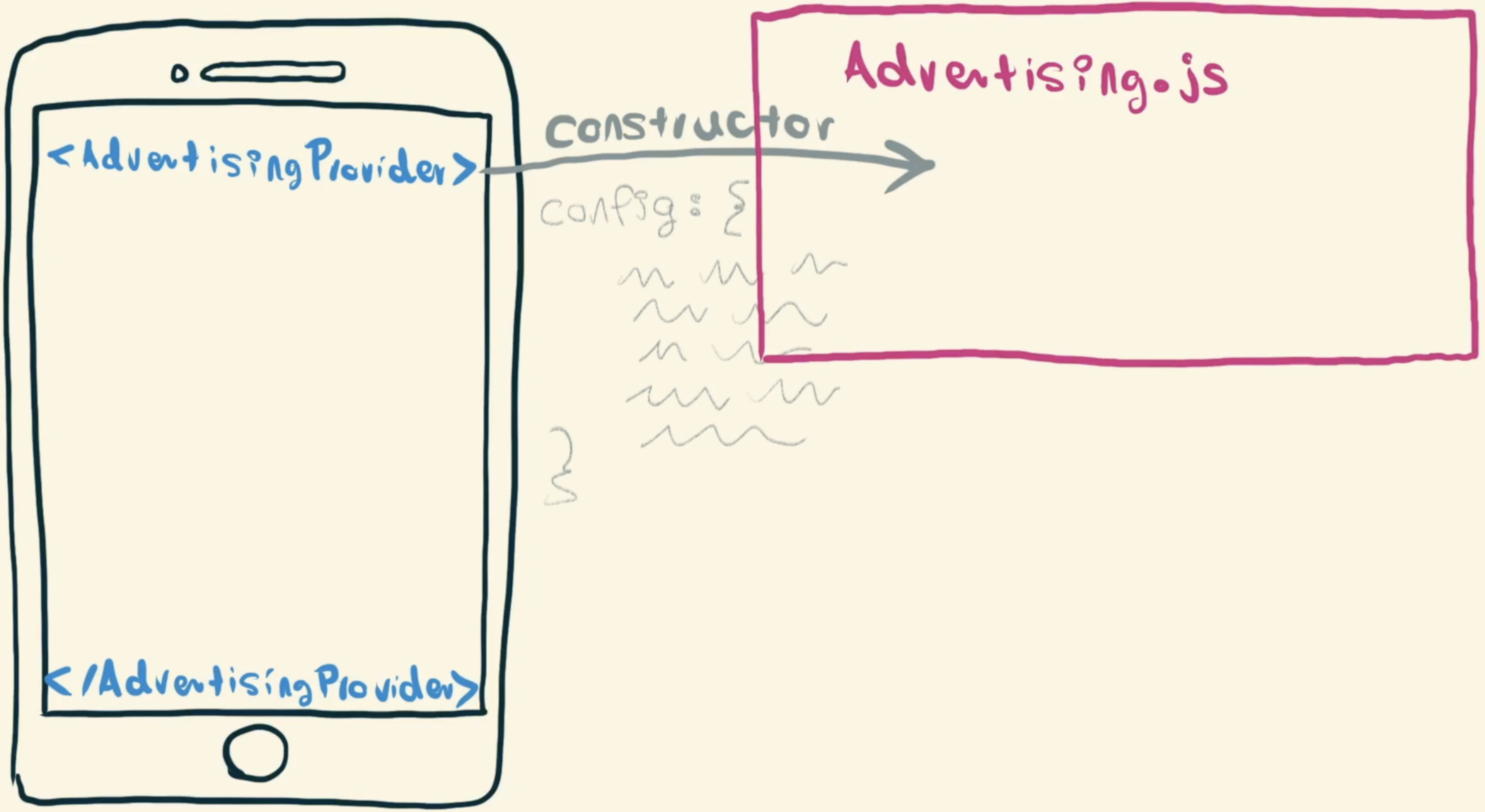
```
n
https://codesandbox.io/static/js/sandbox.2c1b959b.js:3899:50

r.<anonymous>
https://codesandbox.io/static/js/sandbox.2c1b959b.js:3900:4068

n
https://codesandbox.io/static/js/common-sandbox.5c75aa32.js:2847:811

Generator._invoke
https://codesandbox.io/static/js/common-sandbox.5c75aa32.js:2847:599

Generator.e.(anonymous function) [as next]
https://codesandbox.io/static/js/common-sandbox.5c75aa32.js:2847:990

o
https://codesandbox.io/static/js/common-sandbox.5c75aa32.js:3049:316

(anonymous function)
https://codesandbox.io/static/js/common-sandbox.5c75aa32.js:3049:464

new t
https://codesandbox.io/static/js/common-sandbox.5c75aa32.js:678:776

Worker.<anonymous>
https://codesandbox.io/static/js/common-sandbox.5c75aa32.js:3049:258
```

This screen is visible only in development. It will not appear if the app crashes in production.
Open your browser’s developer console to further inspect this error.
This error overlay is powered by `react-error-overlay` used in `create-react-app`.

Console Problems Tests



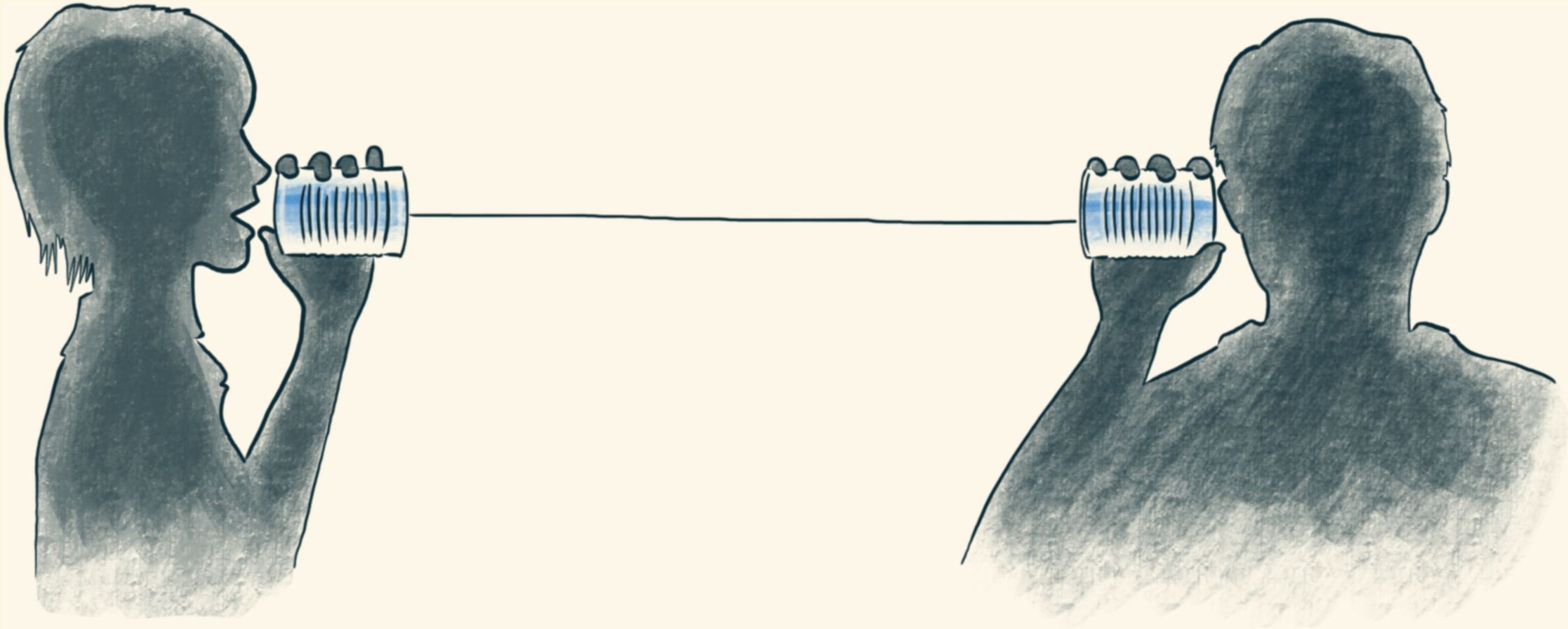

<AdvertisingProvider>
</AdvertisingProvider>


Advertising.js
constructor
<AdvertisingProvider>
config: {
}
</AdvertisingProvider>


<AdvertisingProvider>
</AdvertisingProvider>

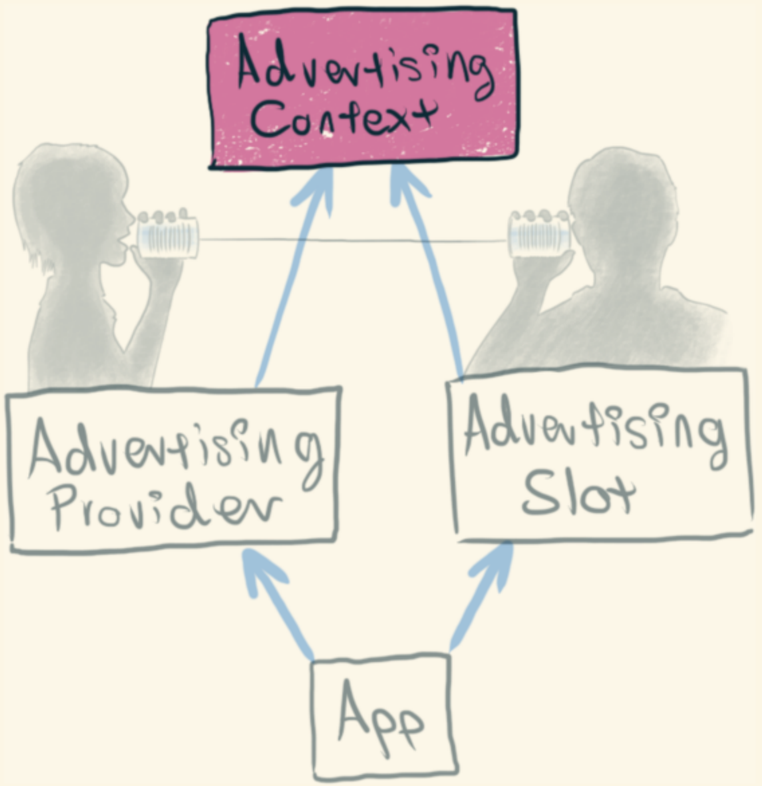

Advertising.js
config: {
}
init
Ad Server

```
const { Provider, Consumer } =
React.createContext(defaultValue);
```



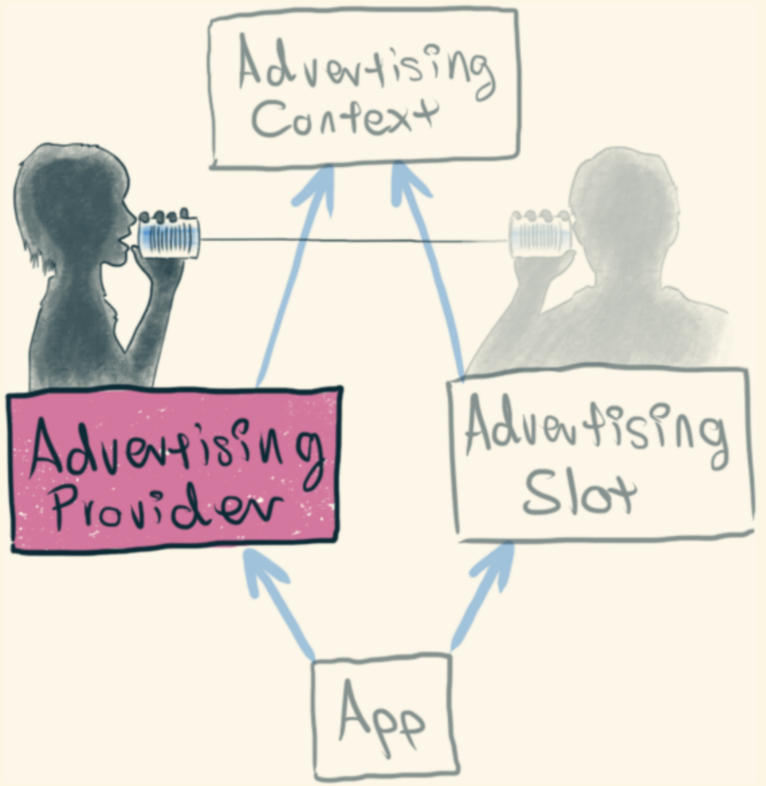
AdvertisingContext.js

```
import { createContext } from 'react';

export default createContext(() => {});
```



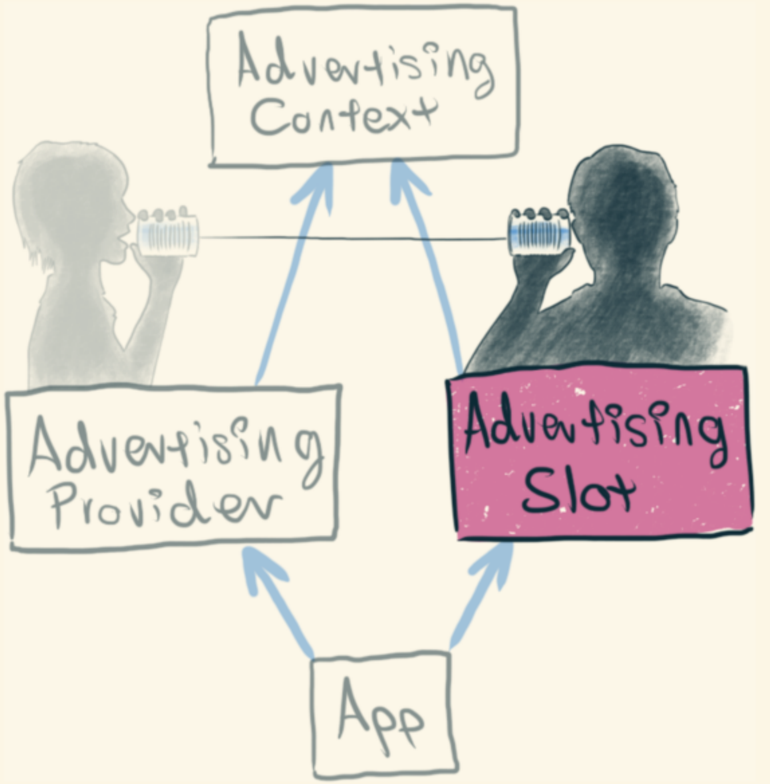
AdvertisingProvider.js

```
import React, { Component } from 'react';
import AdvertisingContext from './AdvertisingContext';
import Advertising from './Advertising';

export default class extends Component {
  constructor(props) {
    super(props);
    this.advertising = new Advertising(props.config);
  }

  render() {
    return (
      <AdvertisingContext.Provider value={this.advertising.activate}>
        {this.props.children}
      </AdvertisingContext.Provider>
    );
  }
}
```



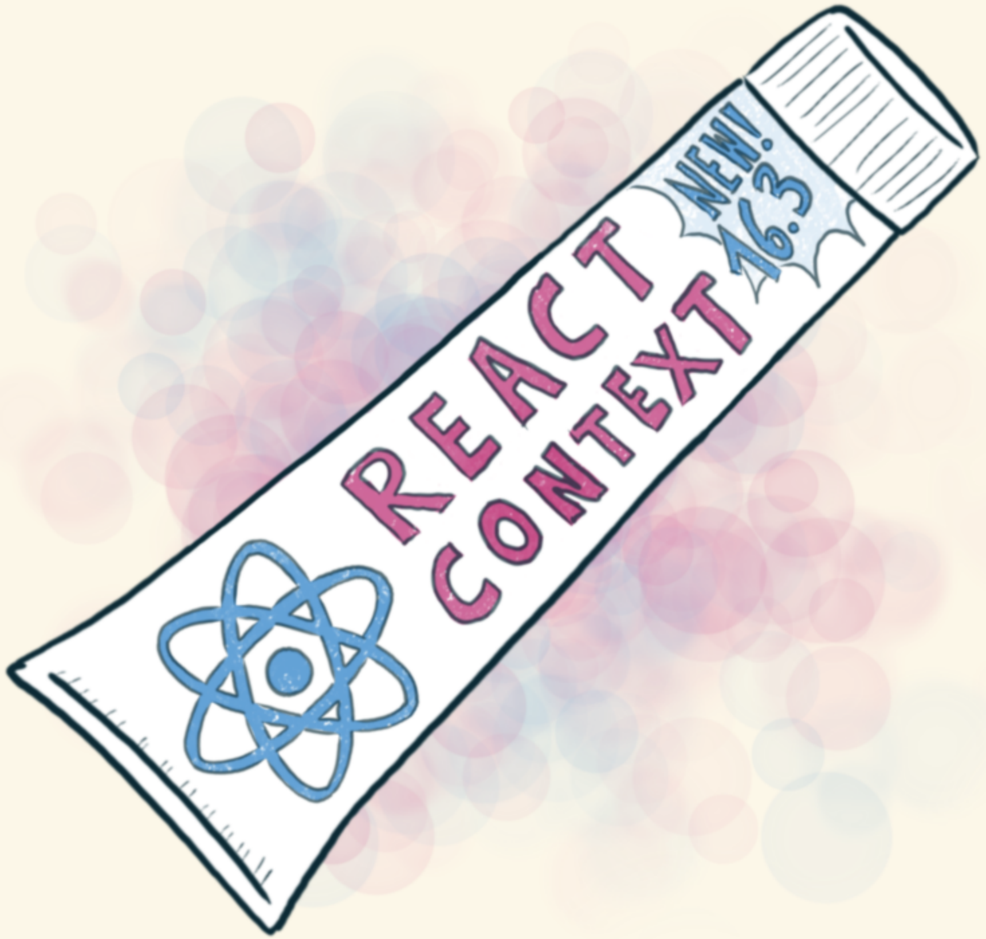
AdvertisingSlot.js

```
import React, { Component } from 'react';
import AdvertisingContext from './AdvertisingContext';

class AdvertisingSlot extends Component {
  componentDidMount() {
    const { activate, id } = this.props;
    activate(id);
  }

  render() {
    return <div id={this.props.id} />
  }
}

export default props => (
  <AdvertisingContext.Consumer>
    {activate => <AdvertisingSlot {...props} activate={activate} />}
  </AdvertisingContext.Consumer>
);
```



```
// Old context API
import React, { Component } from 'react';
import PropTypes from 'prop-types';

class MyProvider extends Component {
  getChildContext() {
    return { message: 'hello world' };
  }
  render() {
    return this.props.children;
  }
}

MyProvider.childContextTypes = {
  message: PropTypes.string
};

class MyConsumer extends Component {
  render() {
    return <h1>{this.context.message}</h1>
  }
}

MyConsumer.contextTypes = {
  message: PropTypes.string
};

export default () => (
  <MyProvider>
    <MyConsumer />
  </MyProvider>
);
```

```
// New context API
import React, {
  Component,
  createContext
} from 'react';

const { Provider, Consumer } = createContext();


function MyProvider({ children }) {
  return (
    <Provider value="hello world">
      {children}
    </Provider>
  );
}


function MyConsumer(props) {
  return (
    <Consumer>
      {message => <h1>{message}</h1>}
    </Consumer>
  );
}


export default () => (
  <MyProvider>
    <MyConsumer />
  </MyProvider>
);
```



DON'T USE IT FOR EVERYTHING!

Other usecase examples besides *getting rich quick*:

- Internationalization
- Layout themes
- Tracking

react-prebid

Library for ad placements with Prebid header bidding in React applications.

To use it, you need to have a Doubleclick for Publishers (DFP) ad server set up, along with configuration to use Prebid in place. Please refer to the Prebid documentation for details.

Usage

Installing the Package

Assuming you already have an application that uses React, install the *react-prebid* package as a production dependency with npm:

```
npm install --save react-prebid
```

github.com/technology-ebay-de/react-prebid

THANK YOU!

Follow me on Twitter:
@wiekatz

Patrick Hund, frontend developer at eBay Tech Berlin

View the slides:
technology-ebay-de.github.io/get-rich-quick-with-react-context/

View the code examples:
codesandbox.io/s/488y5n5kw
codesandbox.io/s/3rz877y8jm