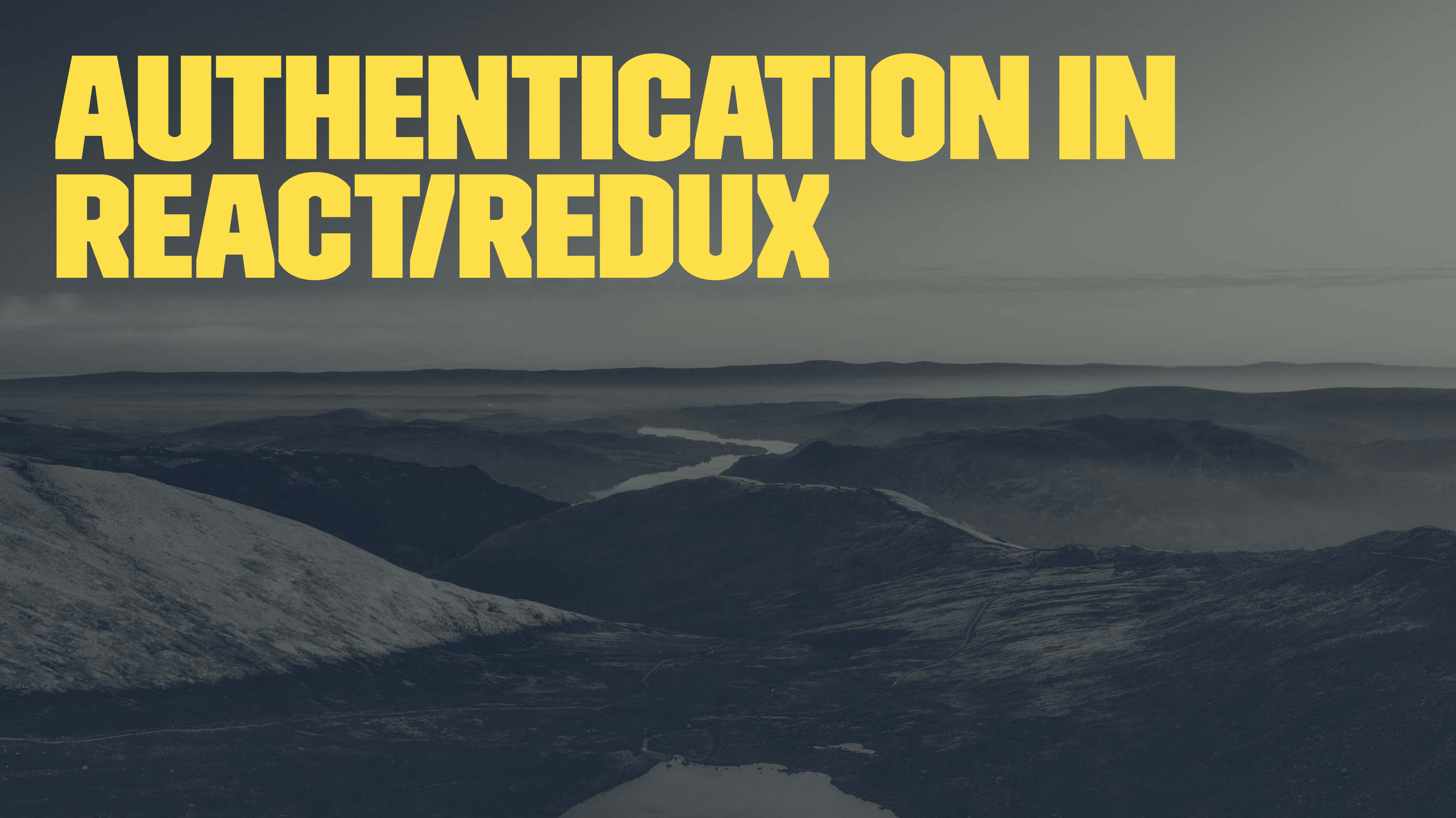
AUTHENTICATION IN REACT/REDUX



ROADMAP

» Authentication

» (Debugging)

AUTHENTICATION

- Verify identity of something
  - Who is somebody
- Identity can be
  - User
  - Different Service
  - Pets
  - ...
- Sometimes refered to as Authn

AUTHENTICATION

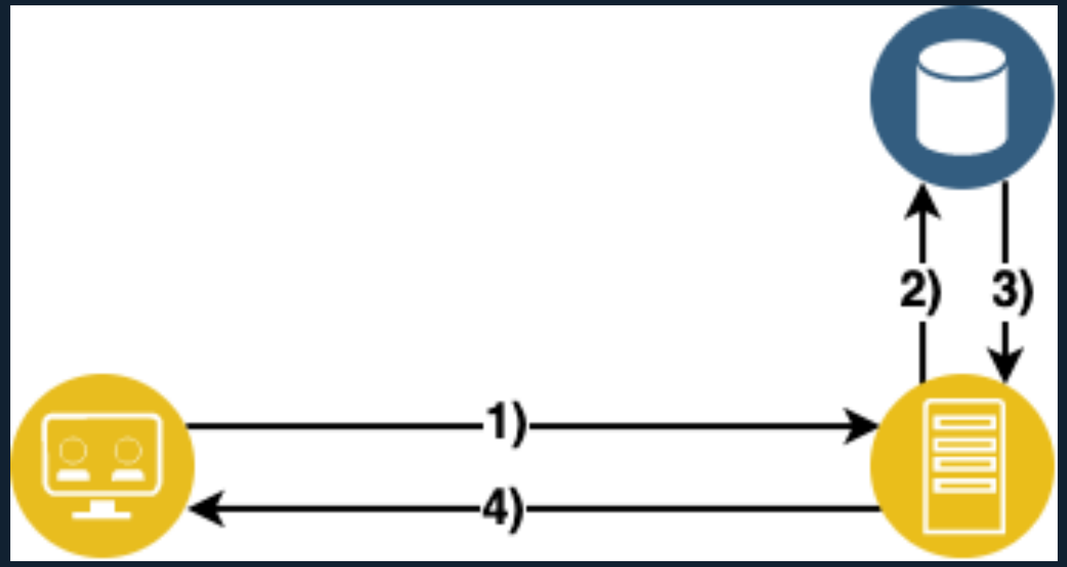
- identity is verified by credentials
- usually combination of username/password
- other data might be used for identification
  - IP Geolocation
  - 2 Factor Authentication
  - Security Keys

AUTHORIZATION

- what resources should somebody have access to
  - What am I allowed to do
  - eg. A person can only change its own password
- usually happens after authentication
  - Anonymous resources might be accessible without authz (eg. reading news articles)

STATEFUL AUTHENTICATION

» Session data is stored in the backend [1] [2] [3] [4]



[1] /money_transactions/ is called

[2] the session for the user is fetched from a db

[3] the session information is returned and verified

[4] result of /money_transactions/ is returned to client

STATEFUL AUTHENTICATION

- Pros:
  - Revoke session anytime
  - Easy to implement
  - Session data can be changed anytime
- Cons:
  - Increasing server resources
  - Every session needs to hit db
  - hard to integrate 3rd party apps

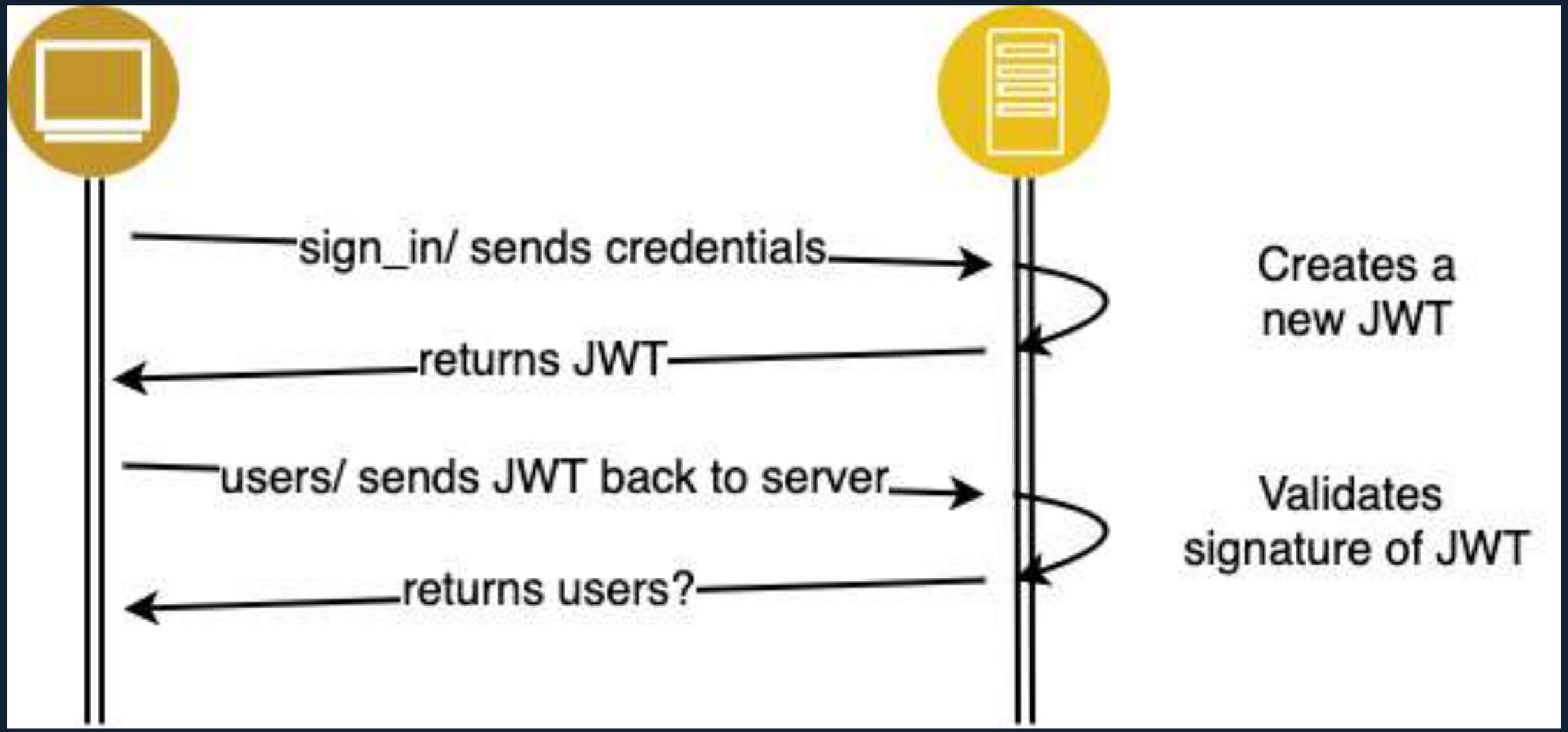
STATELESS AUTHENTICATION

- Session data is stored directly on the client
- Session data is signed and integrity can be verified
  - server only needs to verify validity
  - does not need to refetch data

STATELESS AUTHENTICATION

- Pros:
  - Lower server overhead
  - Easy to scale and integrate with 3rd party
  - 3rd party can read session data
- Cons:
  - Session can't be revoked anytime
  - More complex to implement
  - Session data can't be changed until it expires

STATELESS AUTHENTICATION



JWT (JSON WEB TOKEN)

“JSON Web Token (JWT) is a compact, URL-safe means of representing claims to be transferred between two parties.”

» A signed JSON whose validity can be verified by others

ANATOMY OF JWT

» Header Algorithm & Token type (red)

» Payload (purple)

  » whatever data needed for identification

» signature (blue)

```
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ
zdWIiOiIxMjM0NTY3ODkwIiwibmFtZSI6IkpvaG4
gRG9lIiwiaWF0IjoxNTE2MjM5MDIyfQ.SflKxwRJ
SMeKKF2QT4fwpMeJf36POk6yJV_adQssw5c
```

HEADER

» declare type JWT

» declare hashing algorithm to use

  » there are many others (see https://jwt.io/)

```
{
  "alg": "HS256",
  "typ": "JWT"
}
```

PAYLOAD

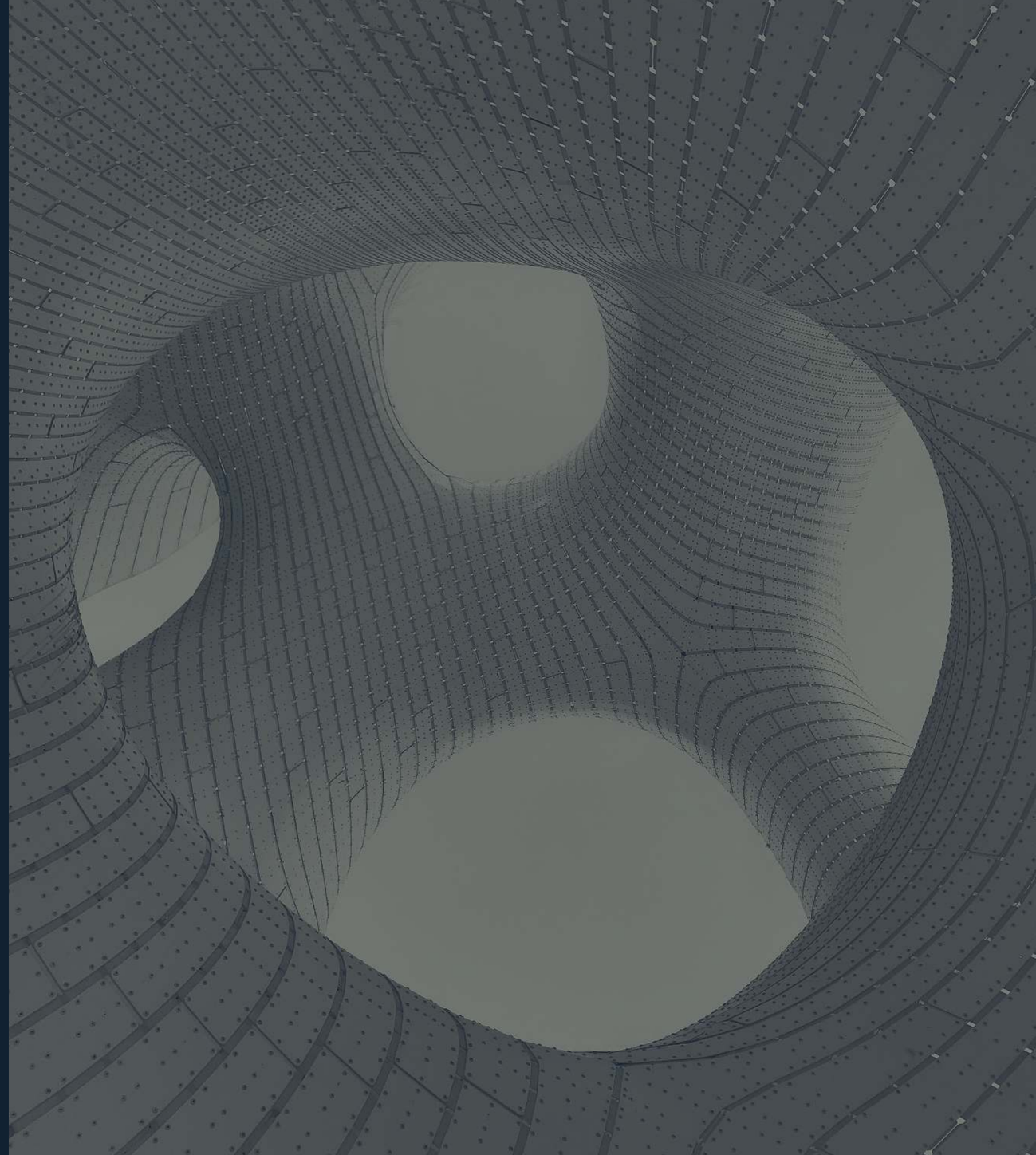
» Carries the information which we want to transmit

» Also called JWT claims

» can be read without the secret

  » don't store sensitive data in here!!!

```
{
  "id": "1234567890",
  "name": "John Doe",
  "roles": ["Admin"]
}
```



SIGNATURE

- Hash of
  - header
  - payload
  - secret
- required for data verification

ANATOMY OF JWT

```
// Header
{ "alg": "HS256", "typ": "JWT" }

// Payload (any valid JSON can be added)
{ "id": "1234567890", "name": "John Doe" }

// verify signature
// ...
```

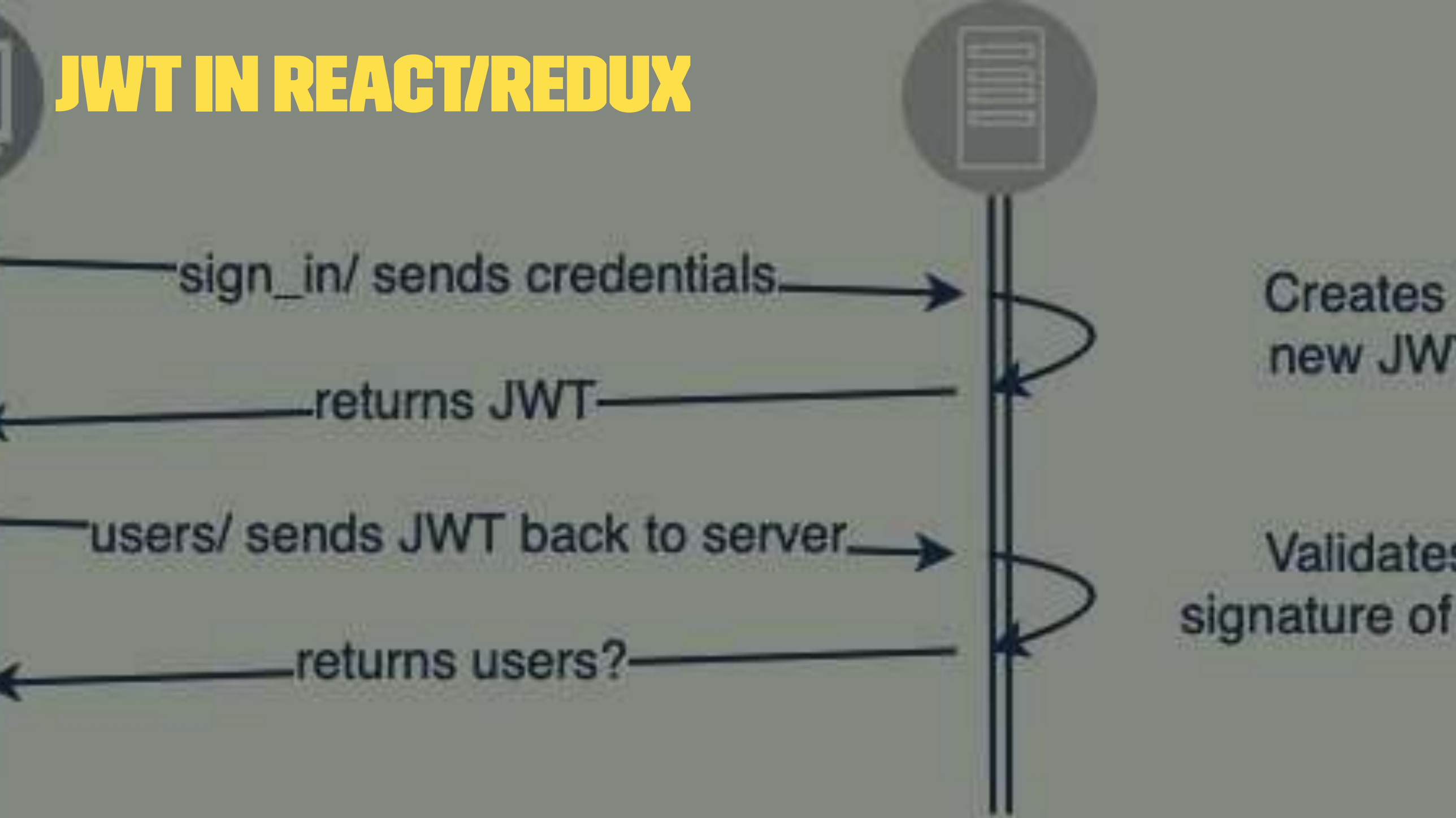
gets converted to:

```
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ
zdWIiOiIxMjM0NTY3ODkwIiwibmFtZSI6IkpvaG4
gRG9lIiwiaWF0IjoxNTE2MjM5MDIyfQ.SflKxwRJ
SMeKKF2QT4fwpMeJf36POk6yJV_adQssw5c
```

PROS

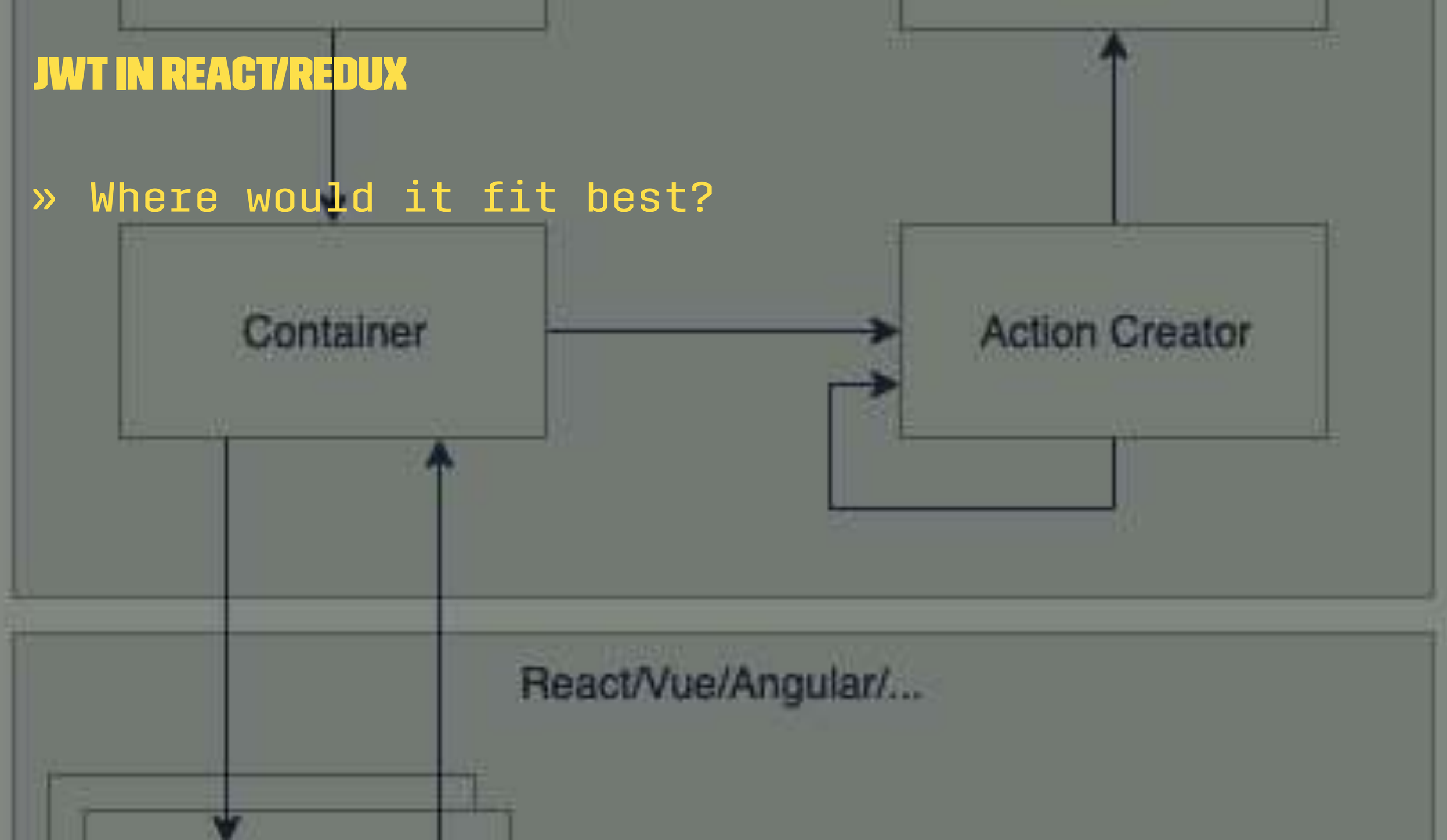
- Standard by IETF
- Scalable
  - no DB hit needed for subsequent requests
- Stateless
- Distributable
- Secure against CSRF

JWT IN REACT/REDUX

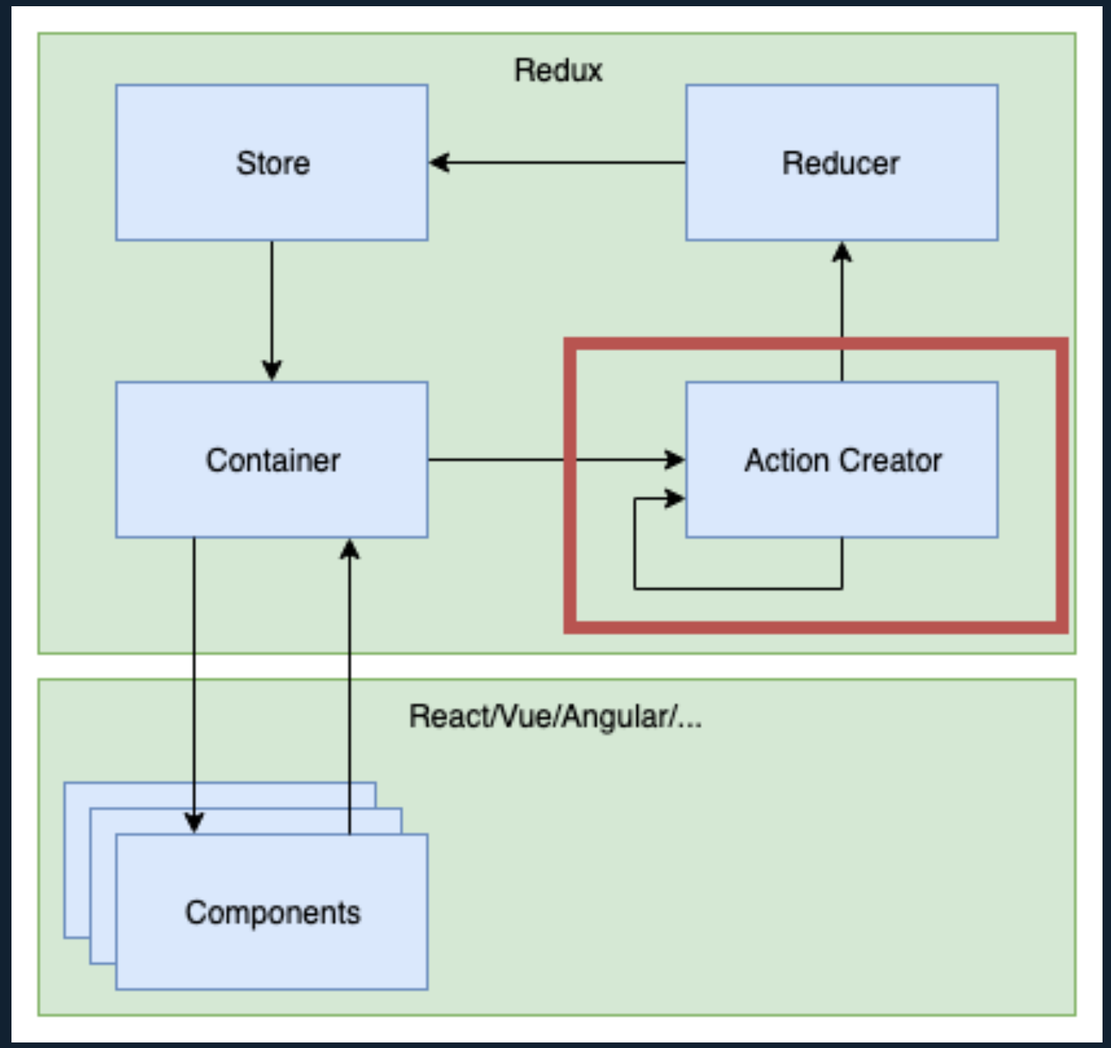


JWT IN REACT/REDUX

» Where would it fit best?



JWT IN REACT/REDUX



RECEIVING A JWT

```
const signIn = ({ email, password }) => async (dispatch) => {
    const response = await fetch("/sign_in", {
      "method": "POST",
      "headers": {
        "Accept": "application/json",
        "Content-Type": "application/json"
      },
      "body": JSON.stringify({ user: { email, password }})
    });
    const token = response.headers.get('Authorization')
    dispatch({ type: 'auth/signed_in', payload: { token }}) // still needs to be written
})
```

SENDING JWT TO BACKEND

```
const getUsers = ({ email, password }) => async (dispatch, getState) => {
  const jwtToken = getState().auth.token
  //               ^^^^^^^^^^^^^^^^^^^^^
  // get JWT token from state
  if (!jwtToken) { redirect('/sign_in') }

  const response = await fetch("/users", {
    headers: {
      'Authorization': jwtToken
      // ^^^^^^^^^^^^^^
      // add JWT token to fetch call
    },
  });
  // ...
})
```

SIGN OUT

```
const signOut = ({ email, password }) => async (dispatch) => {
  // no http call required (token needs to be removed from state in reducer)
  dispatch({ type: 'auth/signed_out'})
})
```

SIGN OUT CAVEATS

» items in store need to be removed manually

  » eg. previous money transactions/users

  » otherwise there is a possible data leak

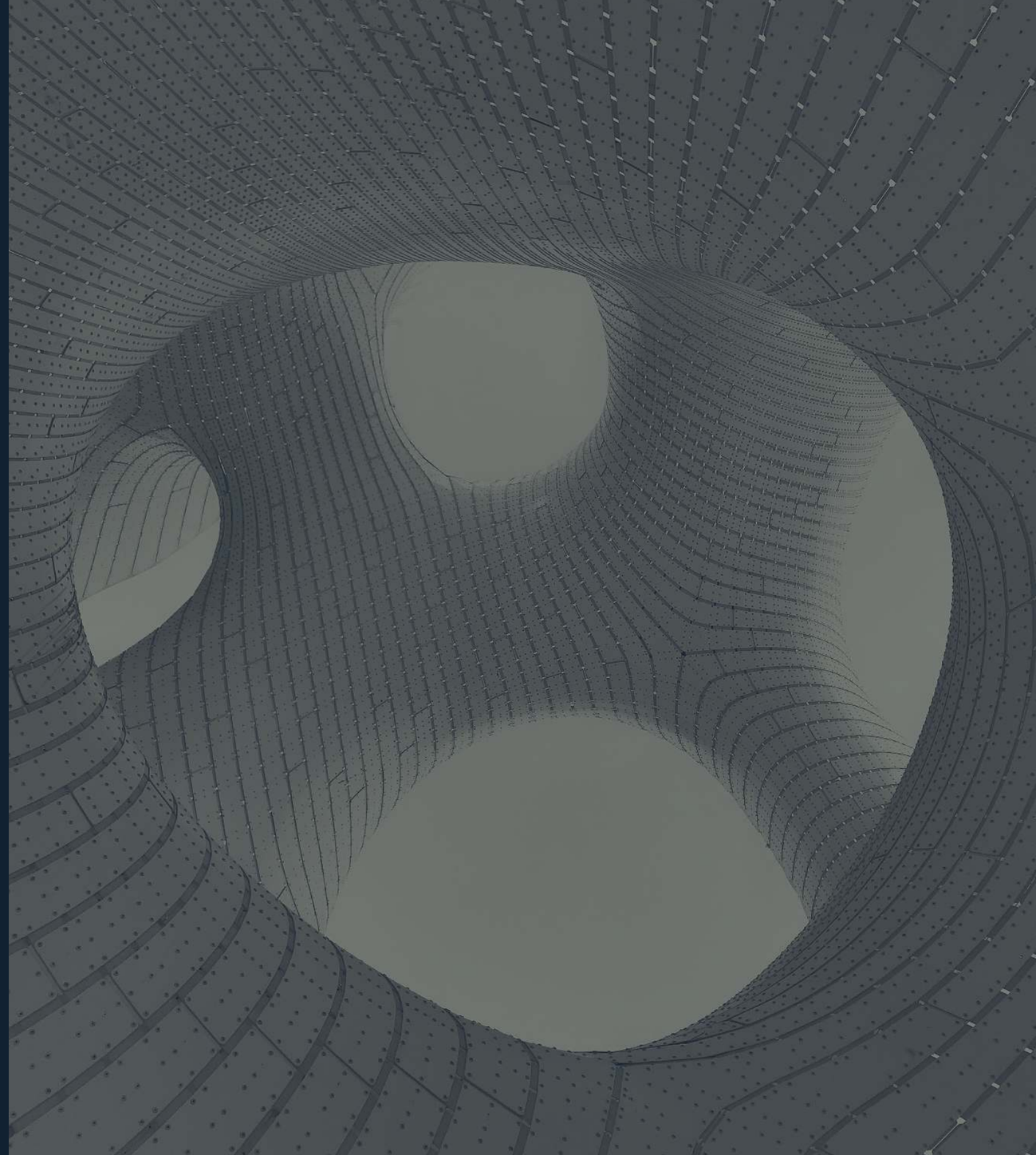
```
const initialState = {}
const userReducer = (previousState = initialState, action) => {
  switch(action.type) {
    // ...
    case 'auth/signed_out':
      return initialState
    // ...
  }
}
```

AUTHENTICATION WITH FIREBASE

FIREBASE
INSTALL FIREBASE TOOLS

» `npm install -g firebase-tools`

» `firebase login`



FIREBASE
INITIALIZE PROJECT IN FIREBASE

- » firebase init
- » Select features
  - » Database
  - » Hosting
  - » Emulators
- » Create new project
  - » select name + 5 random characters at the end

FIREBASE
INITIALIZE PROJECT IN FIREBASE

» What do you want to use as your public directory?

  » enter build

» Configure as a single-page app

  » enter y

» Set up automatic builds

  » enter n for now

FIREBASE
EMULATOR SETUP

- Select
  - Authentication Emulator
  - Database Emulator

FIREBASE
EMULATOR SETUP

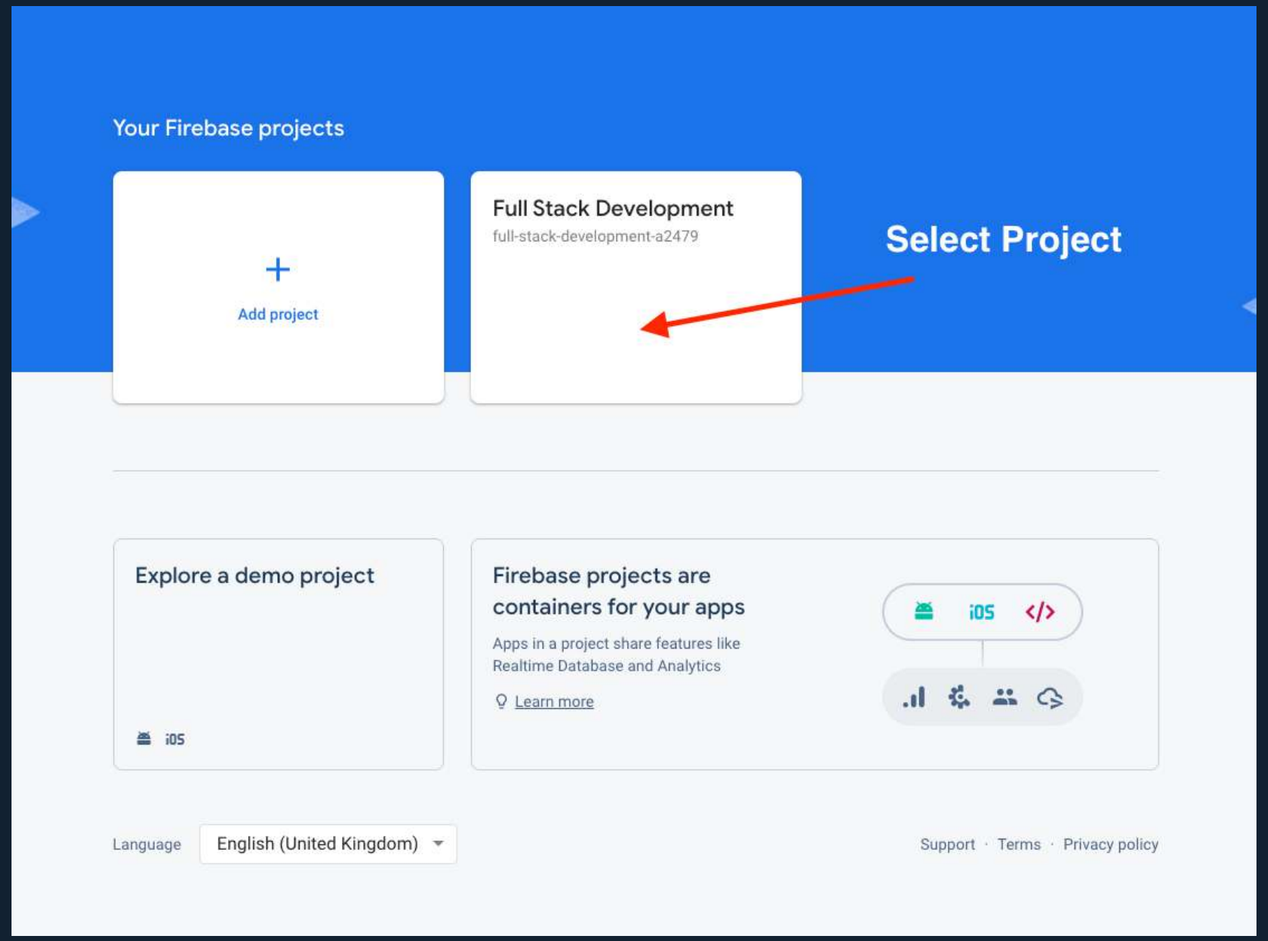
```
Which Firebase emulators do you want to set up? Press Space to select
emulators, then Enter to confirm your choices. Authentication
Emulator, Database Emulator?
- Which port do you want to use for the auth emulator? 9099
- Which port do you want to use for the database emulator? 9000
- Would you like to enable the Emulator UI? Yes
- Which port do you want to use for the Emulator UI (leave empty to use any
  available port)?
- Would you like to download the emulators now? (y/N) y
```

FIREBASE EMULATOR SETUP

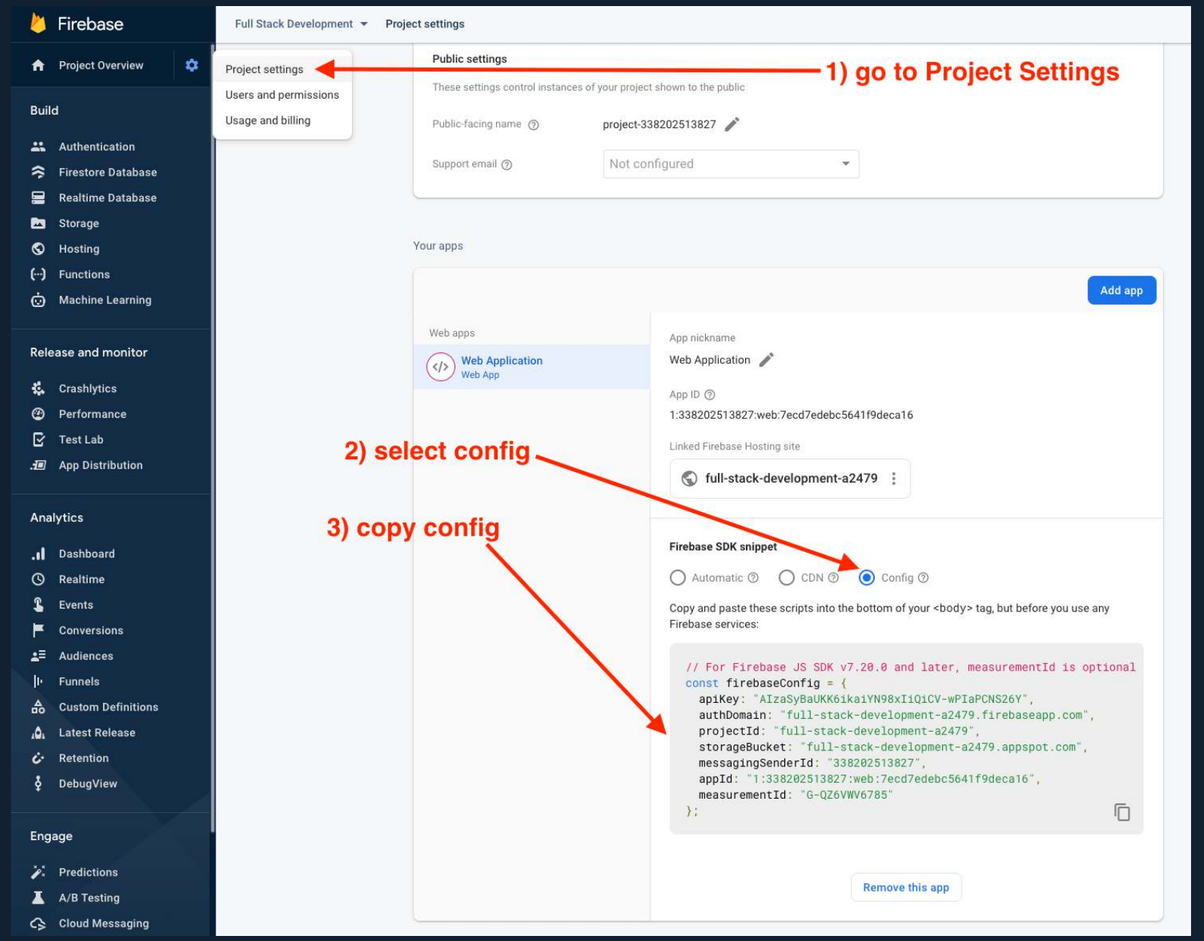
» Adapt start scripts in package.json

```
"start:emulators": "firebase emulators:start",
"start": "concurrently -n firebase,app,storybook 'npm run start:emulators' 'npm run start:app' 'npm run start:storybook'",
```

FIREBASE
GET FIREBASE CONFIG 1/3



FIREBASE
GET FIREBASE CONFIG 2/3



FIREBASE
GET FIREBASE CONFIG 3/3

» paste config in src/firebase.json [6]

```
// src/firebase.js

const firebaseConfig = {
  apiKey: "AIzaSyBaUKK6ikaiYN98xIiQiCV-wPIaPCNS26Y",
  authDomain: "full-stack-development-a2479.firebaseapp.com",
  projectId: "full-stack-development-a2479",
  storageBucket: "full-stack-development-a2479.appspot.com",
  messagingSenderId: "338202513827",
  appId: "1:338202513827:web:7ecd7edebc5641f9deca16",
  measurementId: "G-QZ6VWV6785"
};
```

[6] different for each user

FIREBASE
INSTALL FIREBASE SDK

```
» npm i firebase firebase-tools
```

```
// src/firebase.js
import firebase from 'firebase'
import 'firebase/auth'

const firebaseConfig = {
  apiKey: "AIzaSyBaUKK6ikaiYN98xIiQiCV-wPIaPCNS26Y",
  authDomain: "full-stack-development-a2479.firebaseapp.com",
  projectId: "full-stack-development-a2479",
  storageBucket: "full-stack-development-a2479.appspot.com",
  messagingSenderId: "338202513827",
  appId: "1:338202513827:web:7ecd7edebc5641f9deca16",
  measurementId: "G-QZ6VWV6785"
};

firebase.initializeApp(firebaseConfig)
```

FIREBASE
INSTALL FIREBASE SDK

```
» npm i firebase firebase-tools
```

```
// src/firebase.js
import firebase from 'firebase'
import 'firebase/auth'

const firebaseConfig = {
  apiKey: "AIzaSyBaUKK6ikaiYN98xIiQiCV-wPIaPCNS26Y",
  authDomain: "full-stack-development-a2479.firebaseapp.com",
  projectId: "full-stack-development-a2479",
  storageBucket: "full-stack-development-a2479.appspot.com",
  messagingSenderId: "338202513827",
  appId: "1:338202513827:web:7ecd7edebc5641f9deca16",
  measurementId: "G-QZ6VWV6785"
};

firebase.initializeApp(firebaseConfig)
```

FIREBASE
INITIALIZE FIREBASE AUTH [7]

```
// src/firebase.js
// ...

firebase.initializeApp(firebaseConfig)

export const auth = firebase.auth()

if (process.env.NODE_ENV === 'development') {
//  ^^^^^^^^^^^^^^^^^^^^
// when app is started in development mode
// use the local emulator
  auth.useEmulator('http://localhost:9099')
}
```

[7] complete firebase.js https://gist.github.com/webpapaya/d317f23e993a29055766b00074a5a5eb

FIREBASE
INITIALIZE FIREBASE AUTH

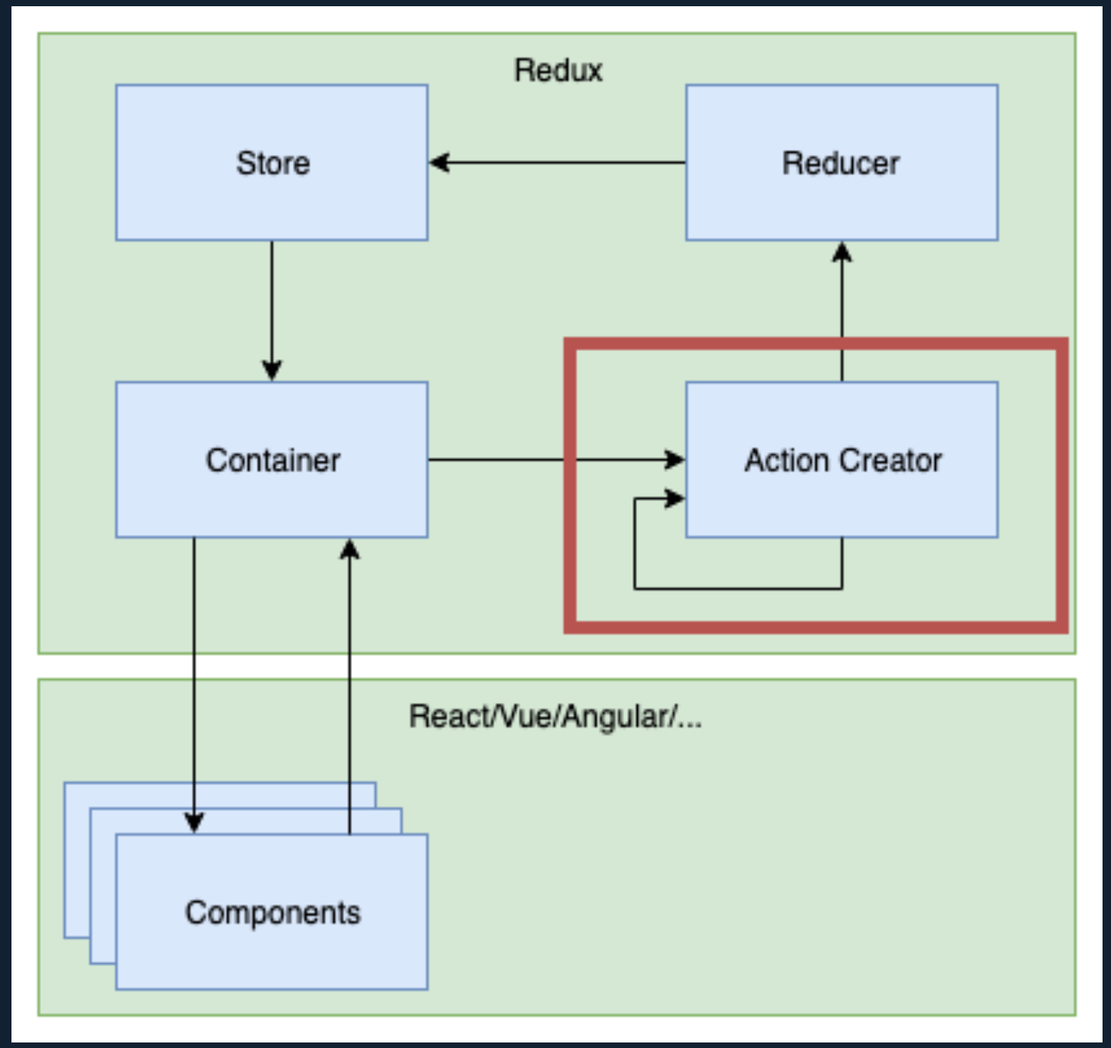
```
import { auth } from '../../firebase.js'

// with our auth instance we can
// signIn
auth.signInWithEmailAndPassword('email@mail.com', 'superSecret')

// signUp
auth.createUserWithEmailAndPassword('email@mail.com', 'superSecret')

// signOut
auth.signOut()
```

FIREBASE AND REDUX



FIREBASE AND REDUX
BUILD AN ACTION CREATOR FOR SIGN UP

```
// src/firebase.js
// ...

import { auth } from './firebase'
const signUp = ({ email, password }) => async (dispatch) => {
  const result = await auth.createUserWithEmailAndPassword(email, password)
  dispatch({
    type: 'user/signedUp'
  })
}
```

HOMEWORK

» see wiki

FEEDBACK

» Questions: tmayrhofer.lba@fh-salzburg.ac.at

» https://de.surveymonkey.com/r/8TW92LL