



# Gieß den Kiez

- Building the Backend
- 20.10.2021

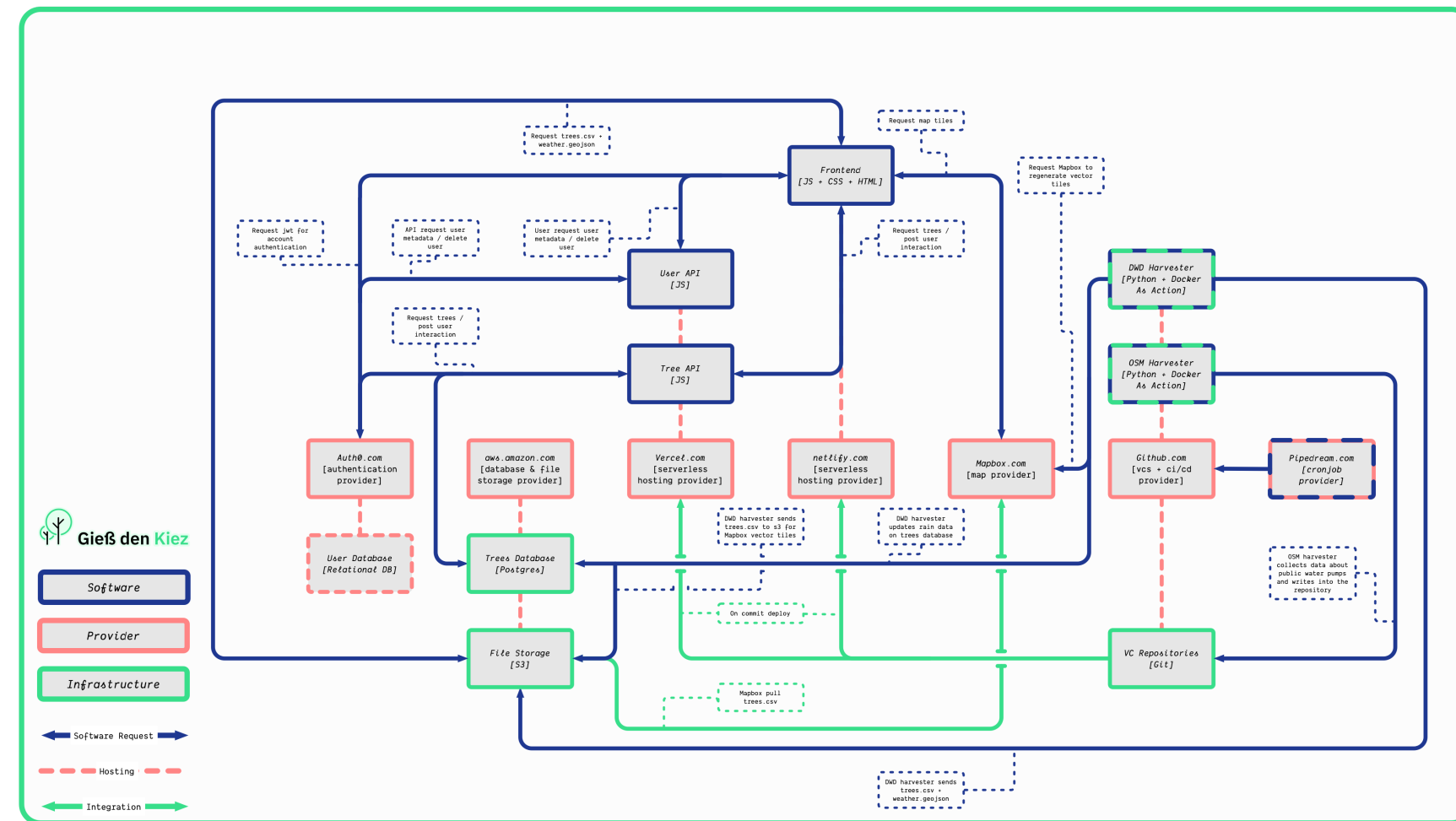
# FABIAN MORÓN ZIRFAS

- Senior Creative Technologist
- @Ideation & Prototyping Lab
- @Technologiestiftung Berlin

# T.O.C.

- ▶ Backend Überblick
- ▶ Software + Services
- ▶ Wo ist der Sourcecode?
- ▶ Wie fange ich an?
- ▶ ~~Beispiel DB & API in 6 Schritten~~
- ▶ Q & A

# BACKEND ÜBERBLICK



# VORRAUSSETZUNGEN

Tool	Kommentar
<a href="#">nvm</a>	Verwaltung von Node.js Versionen
<a href="#">asdf</a>	Verwaltung von CLI Versionen
<a href="#">Git</a>	Versionskontrolle
<a href="#">Node.js</a>	Ausführung
<a href="#">AWS CLI</a>	Abhängigkeit für Terraform
<a href="#">Terraform</a>	Erzeugung von Infrastruktur
<a href="#">auth0.com</a> Account	Auth Provider
<a href="#">vercel.com</a> Account	Hosting Provider
<a href="#">netlify.com</a> Account	Hosting Provider
<a href="#">mapbox.com</a> Account	Karten Provider
<a href="#">AWS</a> Account	DB und Datei Speicherung Provider
<a href="#">GitHub</a> Account	VCS + CI/CD

# WO IST DER SOURCECODE?

Provider	Infrastruktur	Repository ( <a href="https://github.com/technologiestiftung/">https://github.com/technologiestiftung/</a> )
Mapbox	Karten	
Auth0	Autentifizierung	
GitHub	Versionskontrolle & CI/CD	
<del>AWS</del>	<del>Datenbank</del>	giessdenkiez-de-aws-rds-terraform
AWS	Datei Speicherung	giessdenkiez-de-aws-s3-terraform
DWD	Regendaten	giessdenkiez-de-dwd-harvester
OSM	Wasserpumpendaten	giessdenkiez-de-osm-pumpen-harvester
Vercel	Backend Hosting	tsb-trees-api-user-management
Vercel	Backend Hosting	giessdenkiez-de-postgres-api
Netlify	Frontend Hosting	giessdenkiez-de

**WIE FANGE ICH AN?**



▶▶▶ ZUM WIKI ▶▶▶

# **BEISPIEL DB & API IN 6 SCHRITTEN**



Hier könnten Drachen hausen!

# **1. DATENBANK ERZEUGEN**

**AWS ODER NICHT?**



**RENDER.COM**

- Username
- Passwort
- Host
- Port
- Datenbank Name



**ALS postgresql CONNECTION STRING**

```
postgresql://[USER]:[PASSWORD]@[HOST]:[PORT]/[DATABASE]?schema=[SCHEMA]
```

## **2. AUTH0.COM API**

- ▶ Audience
- ▶ Issuer
- ▶ JWKSUri

# 3. QUELLCODE

```
git clone https://github.com/technologiestiftung/giessdenkiez-de-postgres-api.git gdk-api  
cd gdk-api  
npm ci
```

## 3.1 ENVIRONMENT VARIABLEN

```
cp .env.sample .env
```

in .env

```
# this is for the local dev environmet
port=5432
user=fangorn
database=trees
password=ent
host=localhost
# this is for prisma - the pattern is
# postgresql://USER:PASSWORD@HOST:PORT/DATABASE?schema=SCHEMA
DATABASE_URL="postgresql://fangorn:ent@localhost:5432/trees?schema=public"
# you will find these in auth0.com
jwksuri=https://your-fancy-tenant.eu.auth0.com/.well-known/jwks.json
audience=your-audience
issuer=https://your-fancy-tenant.eu.auth0.com/
```

## 4. TABELLEN & DATEN

```
npx prisma db push --preview-feature --skip-generate  
npx prisma db seed --preview-feature
```



## 5. DEPLOY

```
npx vercel
```

# 5.1 ENVIRONMENT VARIABLEN

```
# the user for the postgres db
npx vercel env add user
# the database name
npx vercel env add database
# the database password
npx vercel env add password
# the host of the db, aws? render.com? localhost?
npx vercel env add host
# defaults to 5432
npx vercel env add port
# below are all taken from auth0.com
npx vercel env add jwksuri
npx vercel env add audience
npx vercel env add issuer
```

## 5.2 DEPLOY

```
npx vercel --prod
```



## 6. TEST

```
code --install-extension humao.rest-client  
code docs/api.http
```

## **6.1 TEST AUTH**

## **6.1.1 AUTH0 APPLICATION**

## 6.1.2 ENVIRONMENT VARIABLE

in .env

```
# These can be obtained from Auth0 if you create a new machine to machine  
# application that has access to your API  
client_id=abc123  
client_secret=abc123
```



## 6.1.3 TOKEN HOLEN

```
code docs/api.http
```

## 6.1.4 AUTHENTIFIZIERTE ANFRAGE

in .env

```
# below variables are for testing the api only  
# this token can be obtained by running the POST request to  
# see docs/api.http for more info  
# https://giessdenkiez.eu.auth0.com/oauth/token  
token=a.b.c
```

**Q & A**

**DANKE**

für Ihre Aufmerksamkeit