

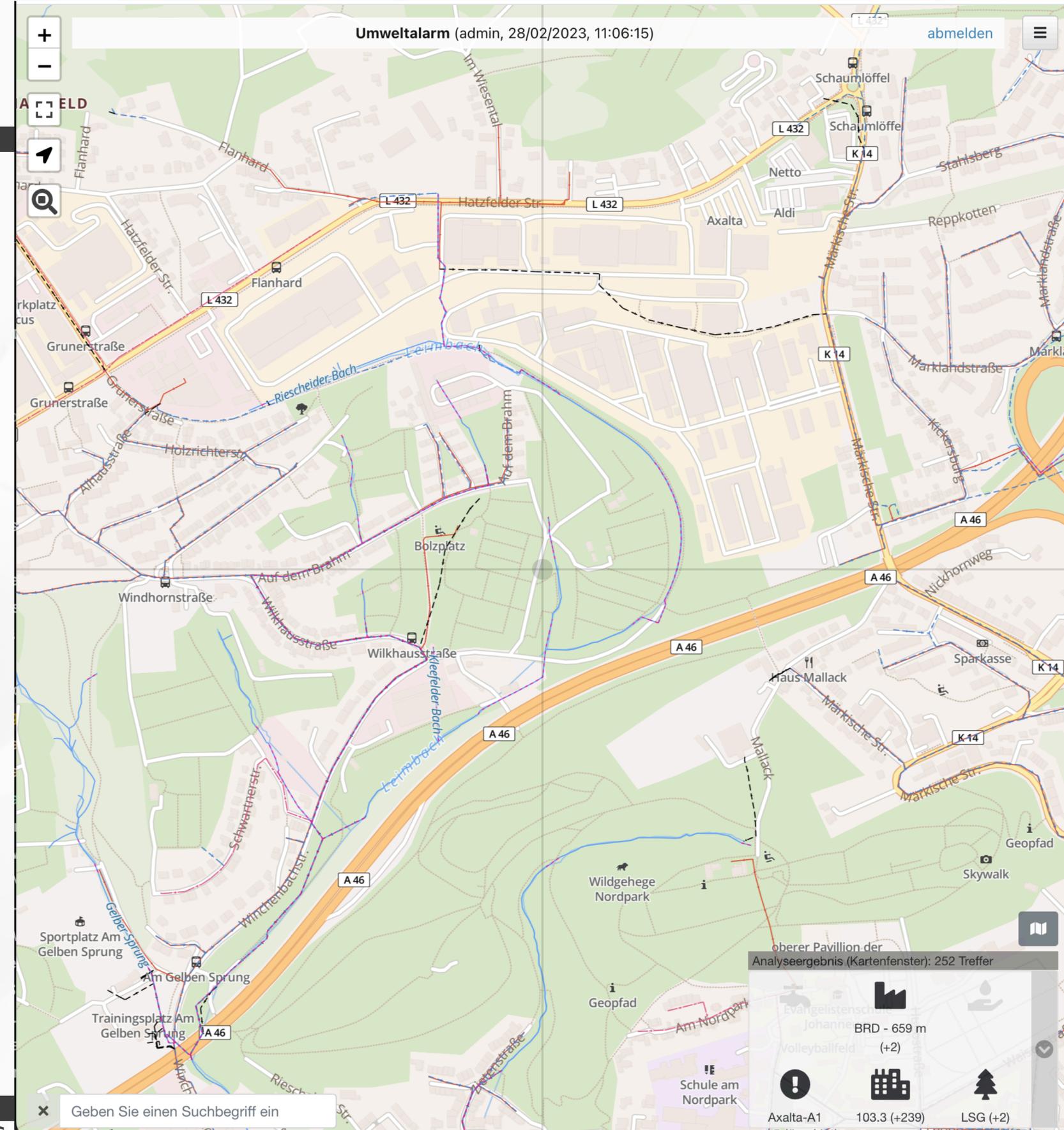
# Offline Karten mit PWA's

thorsten.hell@cismet.de | @hellth on the internet

... and it just works.

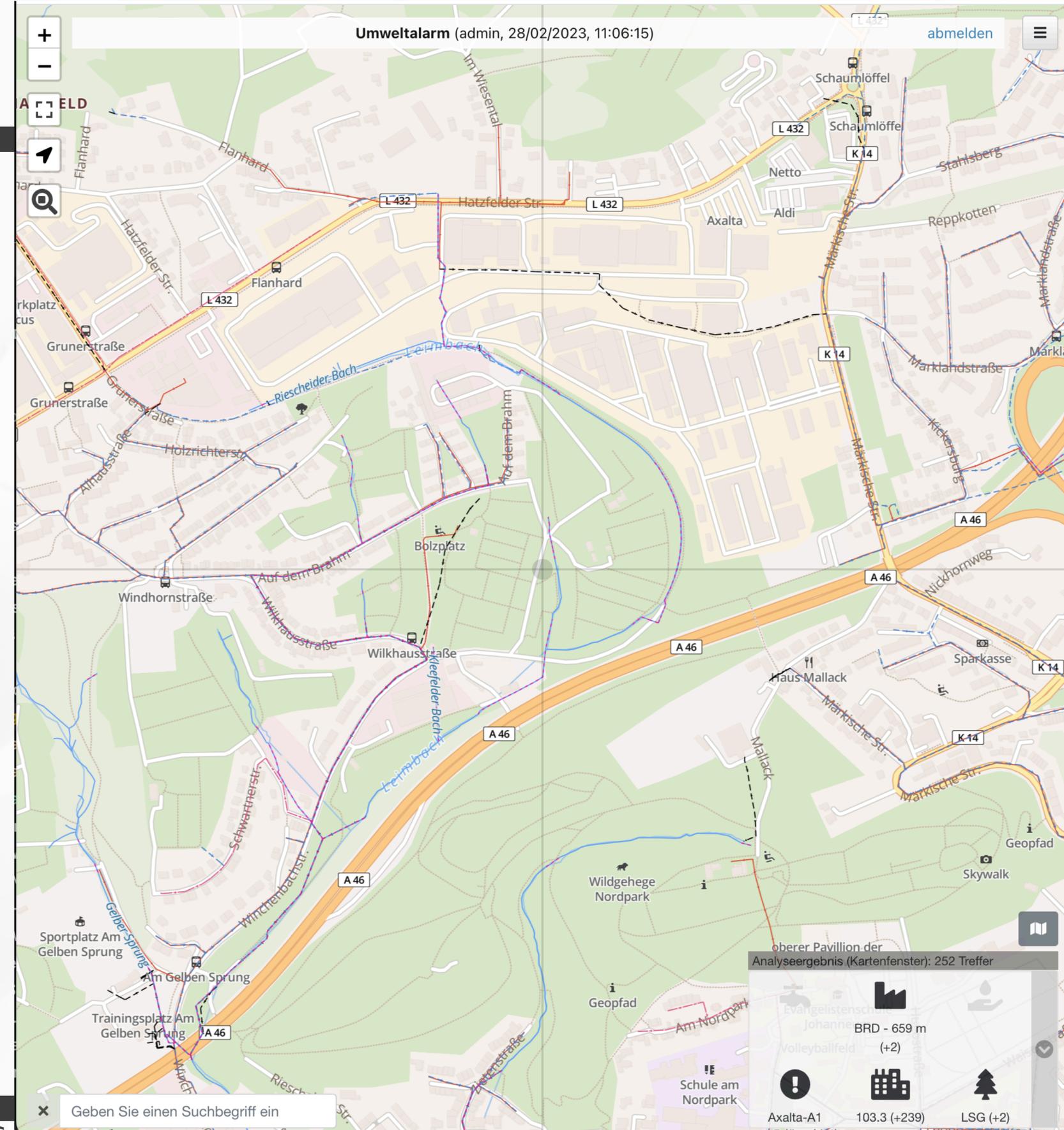
# Warum? / Anlass

- TopicMap für Umweltalarm Team @ Wuppertal
- Muss ohne Internet auskommen



# Warum? / Anlass

- TopicMap für Umweltalarm Team @ Wuppertal
- Muss ohne Internet auskommen
- Datenseite ist zu lösen (indexedDB / Dexie.js)
- App-seite ist zu lösen (PWA mit serviceWorker)
- Kartenhintergrund ???



# Wie?

- ServiceWorker?

Listing 4.4. The `fetch` event inside a Service Worker

```
1 self.addEventListener('fetch', function(event) {           1
2   if (/\.jpg$/ .test(event.request.url)) {                 2
3     event.respondWith(                                     3
4       fetch('/images/unicorn.jpg'));                       3
5   }
6 });
```

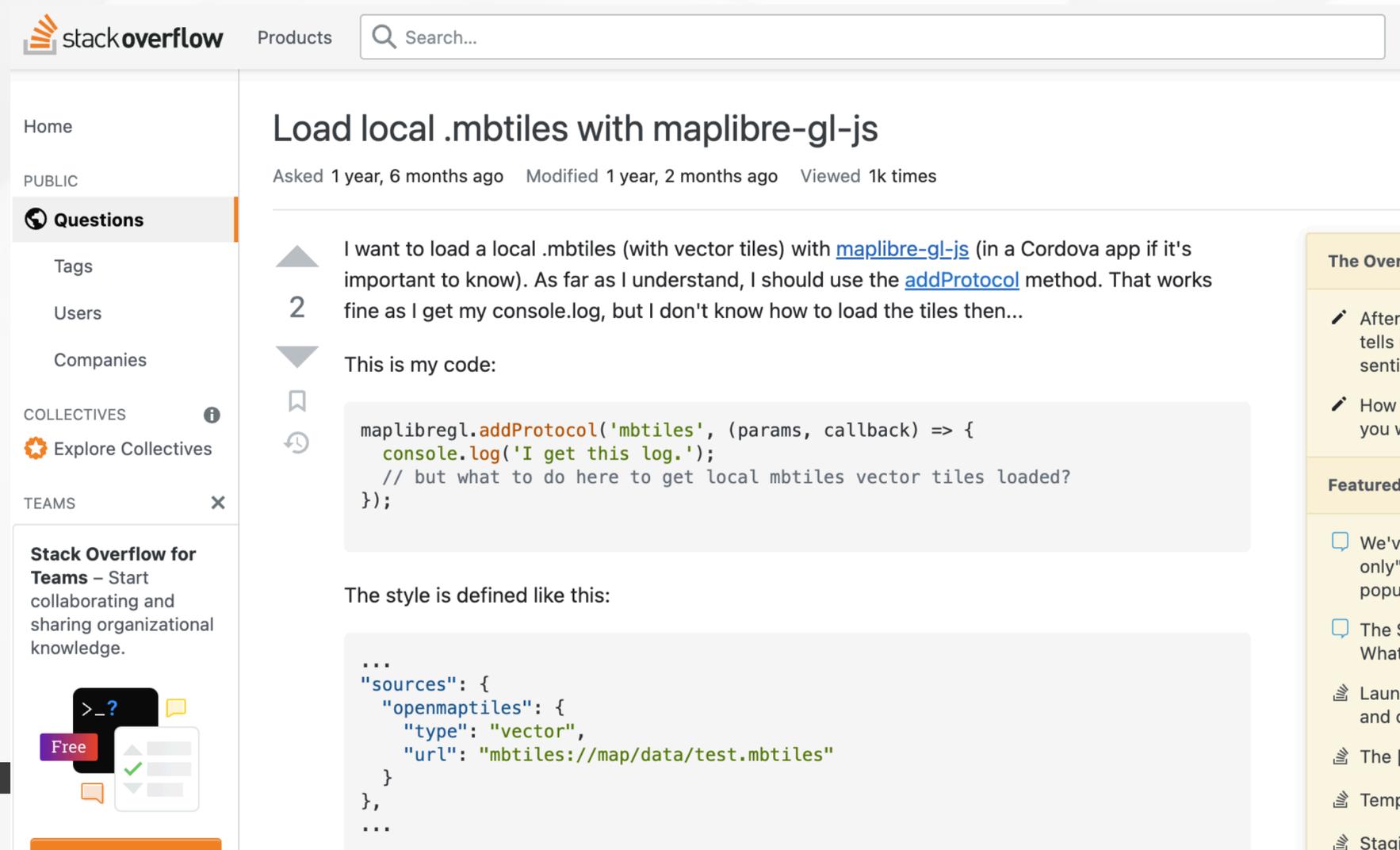
copy

# Wie?

- ServiceWorker?
  - Schlecht zu debuggen. Kein wirkliches Erfolgserlebnis. >>

# Wie?

- ~~ServiceWorker?~~
- ~~Schlecht zu debuggen. Kein wirkliches Erfolgserlebnis. >>~~



The screenshot shows a Stack Overflow page for a question titled "Load local .mbtiles with maplibre-gl-js". The question is asked 1 year, 6 months ago and has 2 answers. The user asks for help loading local .mbtiles (with vector tiles) using maplibre-gl-js in a Cordova app. They mention that using addProtocol works for console.log but they don't know how to load the tiles. The code snippet provided is:

```
maplibregl.addProtocol('mbtiles', (params, callback) => {  
  console.log('I get this log.');
```

The style is defined like this:

```
...  
"sources": {  
  "openmaptiles": {  
    "type": "vector",  
    "url": "mbtiles://map/data/test.mbtiles"  
  }  
},  
...
```

# Wie?

- ~~ServiceWorker?~~
  - ~~Schlecht zu debuggen. Kein wirkliches Erfolgserlebnis. >>~~
- `maplibreGL.addProtocol`
  - Sets a custom load tile function that will be called when using a source that starts with a custom url schema.
  - Genau das was wir brauchten \o/

## MapLibre GL JS Docs

### API REFERENCE ^

Map

[Properties and options](#)

Markers and controls

Geography and geometry

User interaction handlers

Sources

Events

### EXAMPLES

### PLUGINS

### STYLE SPECIFICATION

## addProtocol



src/index.ts

Sets a custom load tile function that will be called when using a source that starts with a custom url schema. The example below will be triggered for custom:// urls defined in the sources list in the style definition. The function passed will receive the request parameters and should call the callback with the resulting tile data, for example a pbf vector tile, non-compressed, represented as ArrayBuffer.

### Parameters

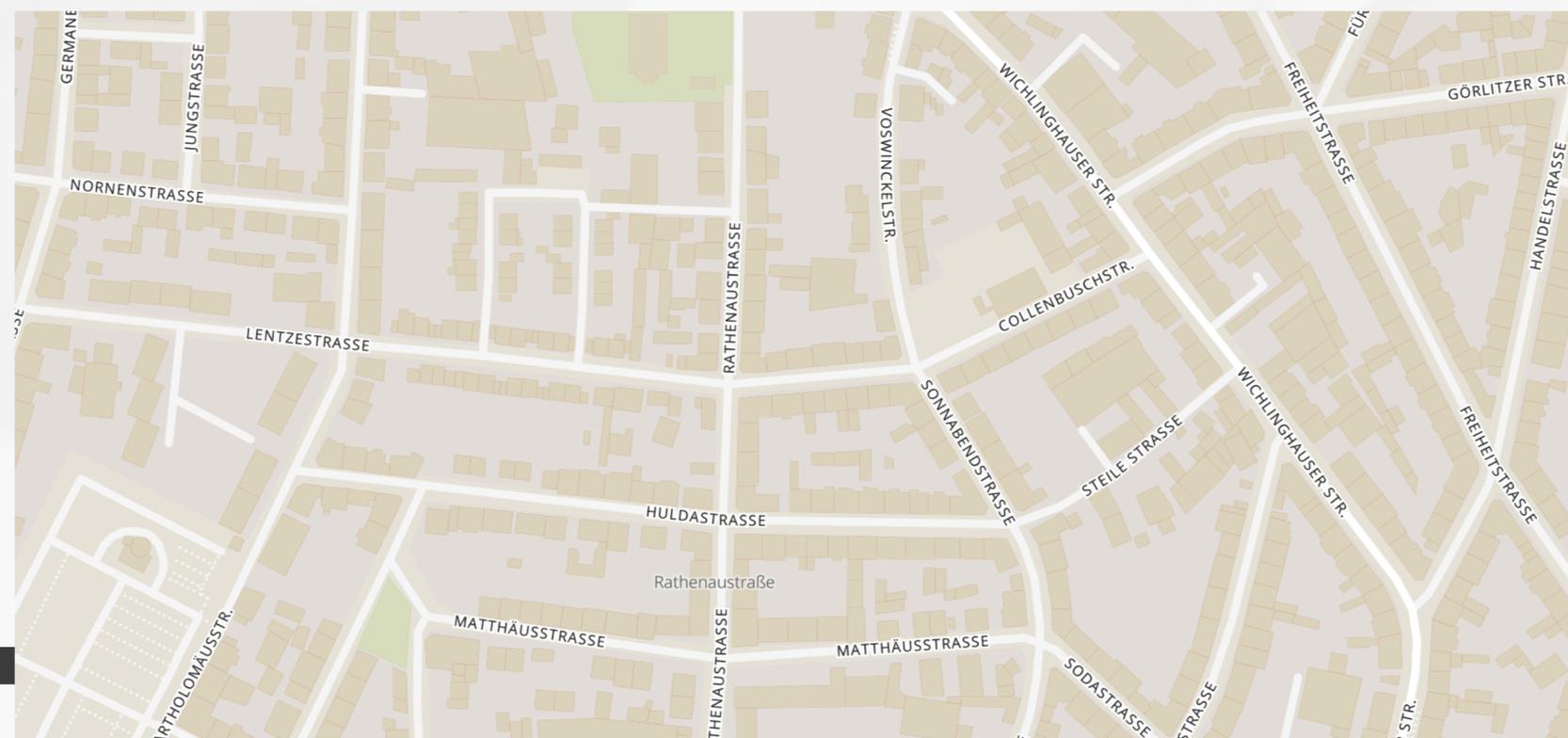
`customProtocol` ([string](#)) the protocol to hook, for example 'custom'

`loadFn` ([Function](#)) the function to use when trying to fetch a tile specified by the customProtocol

# Umsetzung (Step0)

- Einfacher MapLibre Layer in Leaflet

```
<Map style={mapStyle} center={position} zoom={18} maxZoom={25}>  
  <MapLibreLayer style="https://omt.map-hosting.de/styles/klokantech-basic/style.json" />  
</Map>
```



Path	Status	Type	Initiator	Size	T..	Waterfall
/styles/klokantech-basic/style.json	200	fetch	maplibre-gl.js:31	136 B	2...	
/data/v3.json	200	fetch	maplibre-gl.js:31	135 B	1...	
/data/v3/14/8519/5465.pbf	200	fetch	94de2ca5-59e9-4fd3-ac6e-b8a2f16...	(disk cache)	2...	
/fonts/Klokantech%20Noto%20Sans%20Regular/0-255.pbf	200	fetch	maplibre-gl.js:31	(disk cache)	2...	
/data/v3/14/8519/5464.pbf	200	fetch	94de2ca5-59e9-4fd3-ac6e-b8a2f16...	(disk cache)	4...	
/data/v3/14/8520/5464.pbf	200	fetch	94de2ca5-59e9-4fd3-ac6e-b8a2f16...	170 kB	1...	
/data/v3/14/8520/5465.pbf	200	fetch	94de2ca5-59e9-4fd3-ac6e-b8a2f16...	(disk cache)	3...	
/data/v3/14/8520/5465.pbf	200	fetch	94de2ca5-59e9-4fd3-ac6e-b8a2f16...	(disk cache)	4...	
/data/v3/14/8519/5465.pbf	200	fetch	94de2ca5-59e9-4fd3-ac6e-b8a2f16...	(disk cache)	5...	
/data/v3/14/8520/5464.pbf	200	fetch	94de2ca5-59e9-4fd3-ac6e-b8a2f16...	(disk cache)	6...	
/data/v3/14/8519/5464.pbf	200	fetch	94de2ca5-59e9-4fd3-ac6e-b8a2f16...	(disk cache)	5...	
/data/v3/14/8519/5464.pbf	200	fetch	94de2ca5-59e9-4fd3-ac6e-b8a2f16...	(disk cache)	3...	
/data/v3/14/8521/5465.pbf	200	fetch	94de2ca5-59e9-4fd3-ac6e-b8a2f16...	(disk cache)	6...	
/data/v3/14/8521/5464.pbf	200	fetch	94de2ca5-59e9-4fd3-ac6e-b8a2f16...	(disk cache)	6...	
/data/v3/14/8520/5464.pbf	200	fetch	94de2ca5-59e9-4fd3-ac6e-b8a2f16...	(disk cache)	6...	
/data/v3/14/8519/5465.pbf	200	fetch	94de2ca5-59e9-4fd3-ac6e-b8a2f16...	(disk cache)	6...	
/data/v3/14/8520/5465.pbf	200	fetch	94de2ca5-59e9-4fd3-ac6e-b8a2f16...	(disk cache)	6...	
/data/v3/14/8521/5464.pbf	200	fetch	94de2ca5-59e9-4fd3-ac6e-b8a2f16...	(disk cache)	6...	
/data/v3/14/8520/5464.pbf	200	fetch	94de2ca5-59e9-4fd3-ac6e-b8a2f16...	(disk cache)	1...	
/data/v3/14/8519/5464.pbf	200	fetch	94de2ca5-59e9-4fd3-ac6e-b8a2f16...	(disk cache)	8...	

33 / 49 requests | 170 kB / 3.3 MB transferred | 6.3 MB / 24.8 MB resources | Finish: 50.18 s | DOMContentLoaded: 615 ms | Load: 623 ms

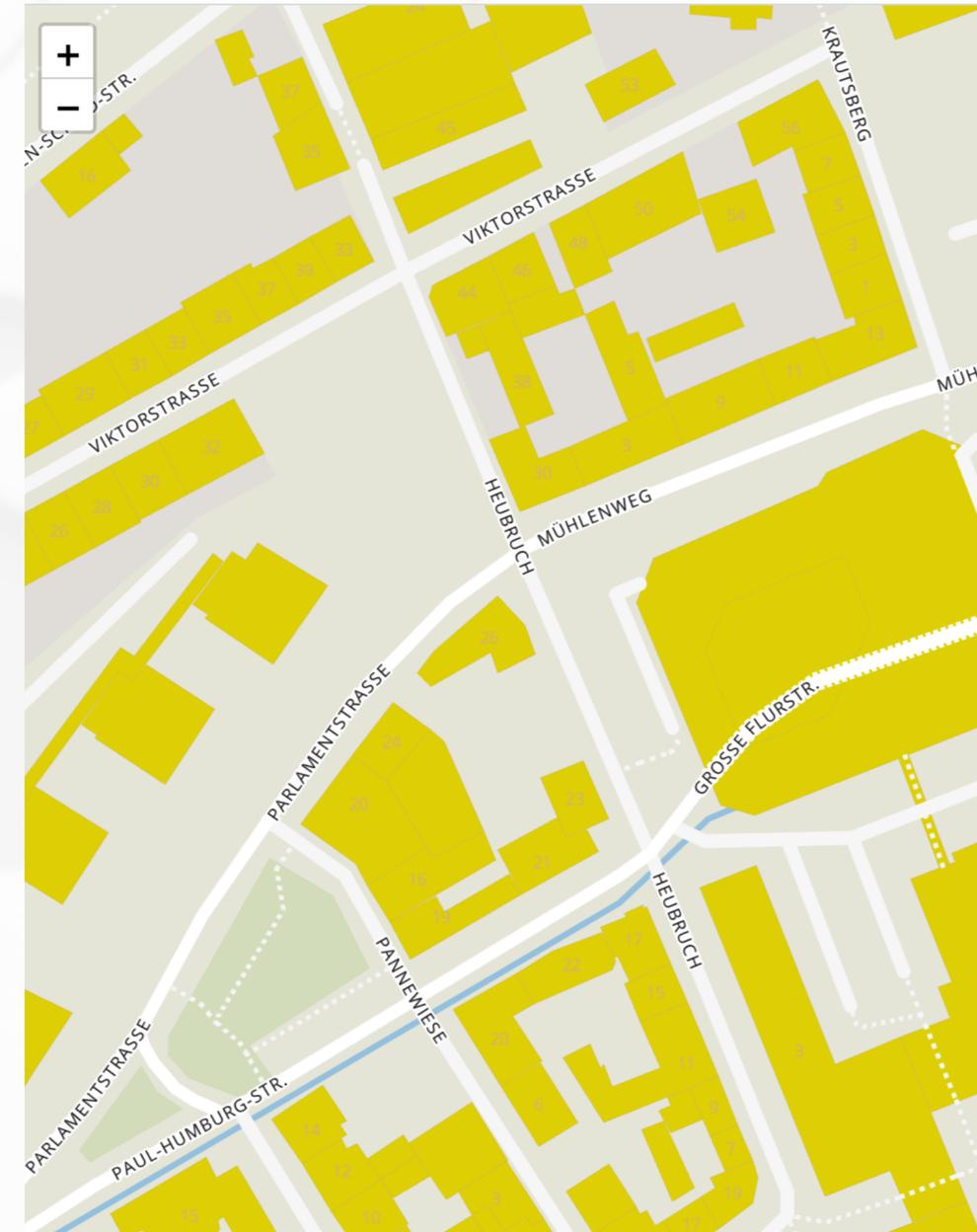
# Umsetzung (Step1)

- Einfacher MapLibre Layer mit lokalem VectorStyle Object

```
import vectorStyle from "./style";
```

```
layerConf.layers[12].paint = {  
  ...layerConf.layers[12].paint,  
  "fill-color": "rgba(222, 211, 0, 1)",  
};
```

```
<Map style={mapStyle} center={position} zoom={18} maxZoom={25}>  
  <MapLibreLayer style={layerConf} />  
</Map>
```



# Umsetzung (Step2)

- Einfacher MapLibre Layer mit lokalem VectorStyle Object

```
import vectorStyle from "./style";
```

```
const layerConf = JSON.parse(JSON.stringify(vectorStyle));
```

```
layerConf.glyphs = "custom://" + layerConf.glyphs;
```

```
layerConf.sources.openmaptiles.tiles[0] =  
  "custom://" + layerConf.sources.openmaptiles.tiles[0];
```

```
<Map style={mapStyle} center={position} zoom={18} maxZoom={25}>
```

```
  <MapLibreLayer style={layerConf} />
```

```
</Map>
```

```
Line 6:4/: 'Link' is defined but never used no-unused-vars  
✖ Fetch API cannot load custom://https://omt.map-hosting.de/data/v3/14/8519/5-  
not supported.  
✖ Error {message: 'Failed to fetch'}  
  message: "Failed to fetch"  
  ▶ [[Prototype]]: Object  
✖ Fetch API cannot load custom://https://omt.map-hosting.de/data/v3/14/8519/5-  
not supported.  
✖ Error {message: 'Failed to fetch'}  
✖ Fetch API cannot load custom://https://omt.map-hosting.de/data/v3/14/8519/5-  
not supported.  
✖ Error {message: 'Failed to fetch'}  
✖ Fetch API cannot load custom://https://omt.map-hosting.de/data/v3/14/8519/5-  
not supported.  
✖ Error {message: 'Failed to fetch'}  
✖ Fetch API cannot load custom://https://omt.map-hosting.de/data/v3/13/4259/2-  
not supported.  
✖ Error {message: 'Failed to fetch'}  
✖ Fetch API cannot load custom://https://omt.map-hosting.de/data/v3/12/2129/1-  
not supported.  
✖ Error {message: 'Failed to fetch'}  
✖ Fetch API cannot load custom://https://omt.map-hosting.de/data/v3/11/1064/6-  
not supported.  
✖ Error {message: 'Failed to fetch'}  
✖ Fetch API cannot load custom://https://omt.map-hosting.de/data/v3/10/532/34-  
supported.  
✖ Error {message: 'Failed to fetch'}  
✖ Fetch API cannot load custom://https://omt.map-hosting.de/data/v3/9/266/170-  
supported.  
✖ Error {message: 'Failed to fetch'}  
✖ Fetch API cannot load custom://https://omt.map-hosting.de/data/v3/8/133/85-  
supported.  
✖ Error {message: 'Failed to fetch'}  
✖ Fetch API cannot load custom://https://omt.map-hosting.de/data/v3/7/66/42-  
supported.  
✖ Error {message: 'Failed to fetch'}  
✖ Fetch API cannot load custom://https://omt.map-hosting.de/data/v3/14/8520/5-  
not supported.  
✖ Error {message: 'Failed to fetch'}  
✖ Fetch API cannot load custom://https://omt.map-hosting.de/data/v3/14/8520/5-  
not supported.  
✖ Error {message: 'Failed to fetch'}  
✖ Fetch API cannot load custom://https://omt.map-hosting.de/data/v3/14/8520/5-  
not supported.  
✖ Error {message: 'Failed to fetch'}
```



# Vorbereitung (Step4)

- Laden der vorbereiteten Daten (w.zip ~30MB) in die indexedDB

```
const offlineConfig = {
  dataStores: [
    {
      name: "Vektorkarte für Wuppertal",
      key: "wuppBasemap",
      url: "w.zip",
    },
  ],
  consoleDebug: true,
};

useEffect(() => {
  loadAndCacheOfflineMapData(offlineConfig, (key, info) => {
    console.log("loadAndCacheOfflineMapData", key, info);
  });
});
```

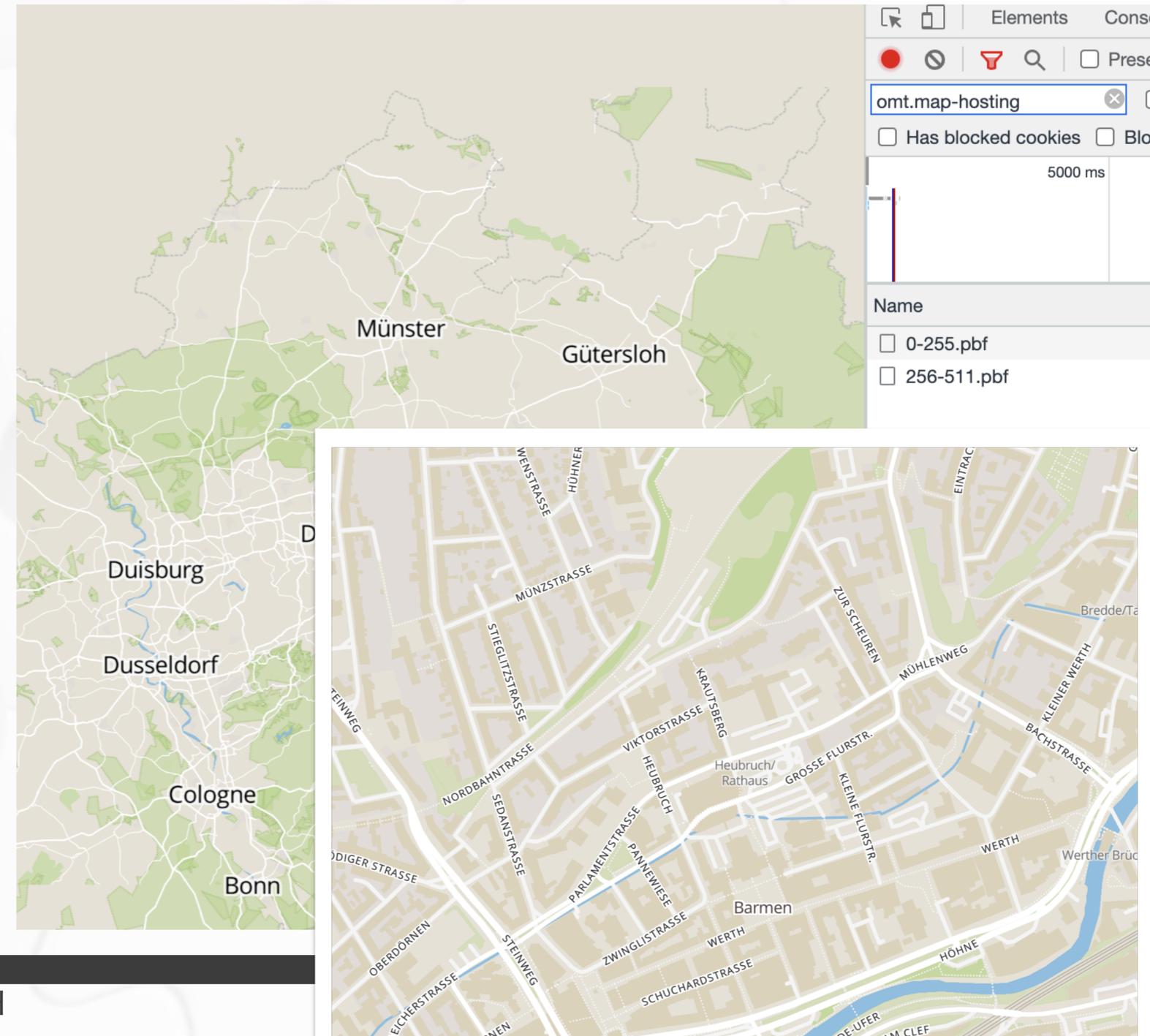
The screenshot shows the Chrome DevTools Application tab with the IndexedDB database expanded. The table below represents the data stored in the IndexedDB database.

#	Key (Key path: "key")	Value
0	"md5.wuppBasemap"	{key: 'md5.wuppBasemap', value: 'cc59af2bf4a18ee0c074f62963f3c83'}
1	"tiles/10/531/340.pbf"	{key: 'tiles/10/531/340.pbf', value: Uint8Array(588)}
2	"tiles/10/531/341.pbf"	{key: 'tiles/10/531/341.pbf', value: Uint8Array(12668)}
3	"tiles/10/531/342.pbf"	{key: 'tiles/10/531/342.pbf', value: Uint8Array(6470)}
4	"tiles/10/532/340.pbf"	{key: 'tiles/10/532/340.pbf', value: Uint8Array(4594)}
5	"tiles/10/532/341.pbf"	{key: 'tiles/10/532/341.pbf', value: Uint8Array(195410)}
6	"tiles/10/532/342.pbf"	{key: 'tiles/10/532/342.pbf', value: Uint8Array(44303)}
7	"tiles/10/533/340.pbf"	{key: 'tiles/10/533/340.pbf', value: Uint8Array(753)}
8	"tiles/10/533/341.pbf"	{key: 'tiles/10/533/341.pbf', value: Uint8Array(9889)}
9	"tiles/10/533/342.pbf"	{key: 'tiles/10/533/342.pbf', value: Uint8Array(3644)}
10	"tiles/11/1063/681.pbf"	{key: 'tiles/11/1063/681.pbf', value: Uint8Array(875)}
11	"tiles/11/1063/682.pbf"	{key: 'tiles/11/1063/682.pbf', value: Uint8Array(17402)}
12	"tiles/11/1063/683.pbf"	{key: 'tiles/11/1063/683.pbf', value: Uint8Array(15337)}
13	"tiles/11/1063/684.pbf"	{key: 'tiles/11/1063/684.pbf', value: Uint8Array(16888)}
14	"tiles/11/1064/681.pbf"	{key: 'tiles/11/1064/681.pbf', value: Uint8Array(6283)}
15	"tiles/11/1064/682.pbf"	{key: 'tiles/11/1064/682.pbf', value: Uint8Array(104606)}
16	"tiles/11/1064/683.pbf"	{key: 'tiles/11/1064/683.pbf', value: Uint8Array(153720)}
17	"tiles/11/1064/684.pbf"	{key: 'tiles/11/1064/684.pbf', value: Uint8Array(58311)}
18	"tiles/11/1065/681.pbf"	{key: 'tiles/11/1065/681.pbf', value: Uint8Array(4673)}
19	"tiles/11/1065/682.pbf"	{key: 'tiles/11/1065/682.pbf', value: Uint8Array(114381)}
20	"tiles/11/1065/683.pbf"	{key: 'tiles/11/1065/683.pbf', value: Uint8Array(108312)}
21	"tiles/11/1065/684.pbf"	{key: 'tiles/11/1065/684.pbf', value: Uint8Array(59820)}
22	"tiles/11/1066/681.pbf"	{key: 'tiles/11/1066/681.pbf', value: Uint8Array(719)}
23	"tiles/11/1066/682.pbf"	{key: 'tiles/11/1066/682.pbf', value: Uint8Array(15518)}
24	"tiles/11/1066/683.pbf"	{key: 'tiles/11/1066/683.pbf', value: Uint8Array(11332)}
25	"tiles/11/1066/684.pbf"	{key: 'tiles/11/1066/684.pbf', value: Uint8Array(6562)}
26	"tiles/12/2127/1363.pbf"	{key: 'tiles/12/2127/1363.pbf', value: Uint8Array(1720)}
27	"tiles/12/2127/1364.pbf"	{key: 'tiles/12/2127/1364.pbf', value: Uint8Array(34694)}
28	"tiles/12/2127/1365.pbf"	{key: 'tiles/12/2127/1365.pbf', value: Uint8Array(25445)}
29	"tiles/12/2127/1366.pbf"	{key: 'tiles/12/2127/1366.pbf', value: Uint8Array(18871)}
30	"tiles/12/2127/1367.pbf"	{key: 'tiles/12/2127/1367.pbf', value: Uint8Array(35647)}

# Umsetzung (Step4)

- Intercepting und Laden der Tiles aus der indexedDB (Dexie.js)

```
useEffect(() => {
  loadAndCacheOfflineMapData(offlineConfig, (key, info) => {
    console.log("loadAndCacheOfflineMapData", key, info);
  });
  ...
  maplibreGl.addProtocol("indexedDB", (params, callback) => {
    let url = params.url.replace("indexedDB://", "");
    console.log("indexeddb protocol interception", url);
    //Just do the tiles for now (starting with "https://omt.map-hosting.de/data/v3/")
    if (url.startsWith("https://omt.map-hosting.de/data/v3/")) {
      const dbKey = url.replace(
        "https://omt.map-hosting.de/data/v3/",
        "tiles/"
      );
      // local anonymous async block ---
      (async () => {
        const hit = await db[OBJECTSTORE].get(dbKey);
        if (hit) {
          callback(null, hit.value.buffer, null, null);
        }
      })();
      // -----
    } else {
      fetchOnline(url, callback);
    }
  });
});
```



# Umsetzung (Step5)

- ... wenn nichts in der DB ist, fetchOnline

```

// local anonymous async block ---
  (async () => {
    const hit = await db[OBJECTSTORE].get(dbKey);
    if (hit) {
      callback(null, hit.value.buffer, null, null);
    } else {
      fetchOnline(url, callback);
    }
  })();
// -----
```

# Umsetzung (Step6)

- offlineFetch in helper Function ausgelagert (wertet Ruleset aus)

```
const offlineConfig = {
  dataStores: [
    {
      name: "Vektorkarte für Wuppertal",
      key: "wuppBasemap",
      url: "w.zip",
    },
  ],
  consoleDebug: true,
  rules: [
    {
      origin: "https://omt.map-hosting.de/fonts/Klokantech Noto",
      cachePath: "fonts/0pen",
      realServerFallback: false,
    },
    {
      origin: "https://omt.map-hosting.de/fonts",
      cachePath: "fonts",
      realServerFallback: false,
    },
    {
      origin: "https://omt.map-hosting.de/data/v3",
      cachePath: "tiles",
      realServerFallback: false,
    },
  ],
};
```

```
maplibreGl.addProtocol("indexedDB", (params, callback) => {
  let url = params.url.replace("indexedDB://", "");
  console.log("indexeddb protocol interception", url);
  customOfflineFetch(url, offlineConfig).then((buffer) => {
    if (buffer) {
      callback(null, buffer, null, null);
    } else {
      callback(null, new ArrayBuffer(), null, null);
    }
  });
});
```

# Umsetzung (Step7)

- realServerFallback: true

```
const offlineConfig = {
  dataStores: [
    {
      name: "Vektorkarte für Wuppertal",
      key: "wuppBasemap",
      url: "w.zip",
    },
  ],
  consoleDebug: true,
  rules: [
    {
      origin: "https://omt.map-hosting.de/fonts/Klokantech Noto",
      cachePath: "fonts/0pen",
      realServerFallback: true,
    },
    {
      origin: "https://omt.map-hosting.de/fonts",
      cachePath: "fonts",
      realServerFallback: true,
    },
    {
      origin: "https://omt.map-hosting.de/data/v3",
      cachePath: "tiles",
      realServerFallback: true,
    },
  ],
};
```

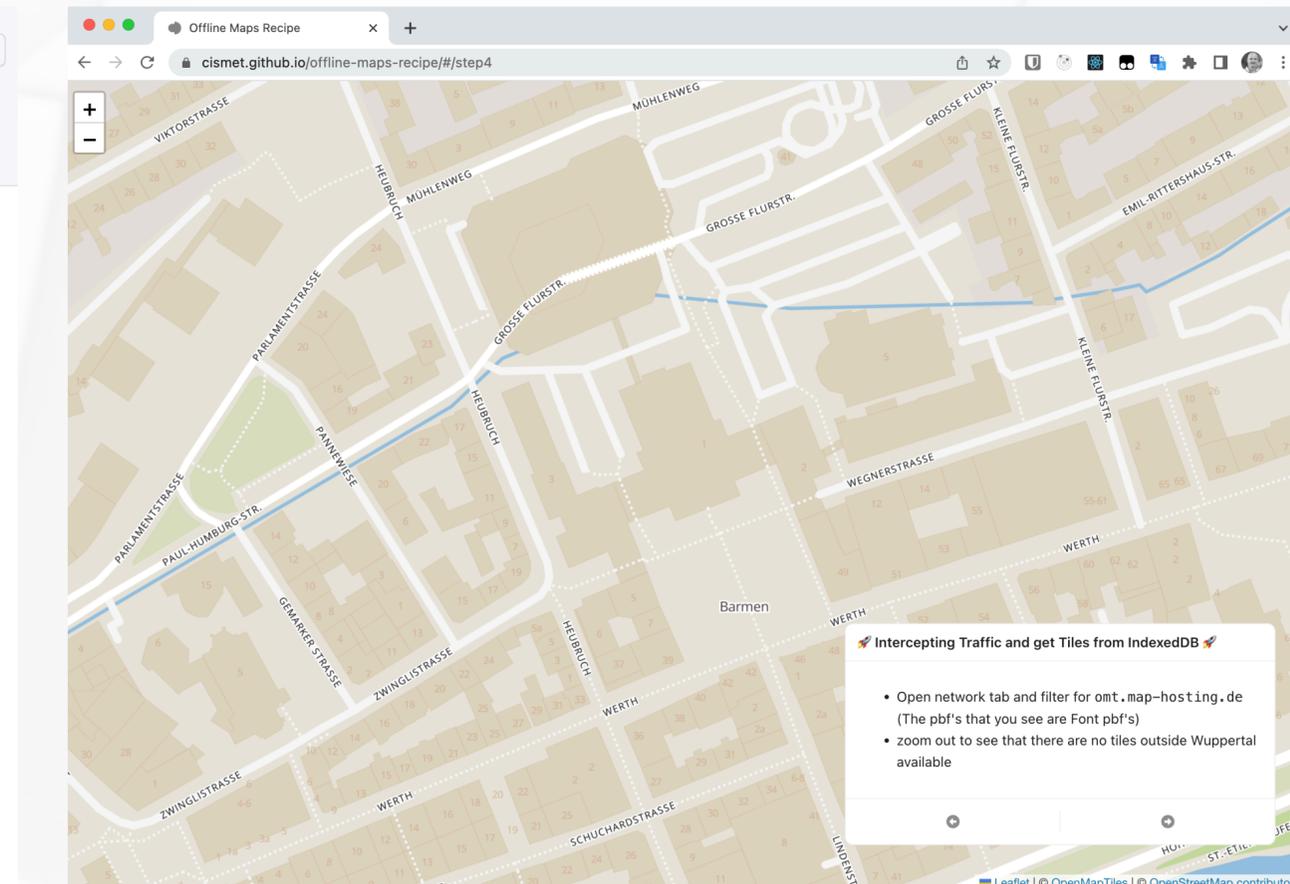
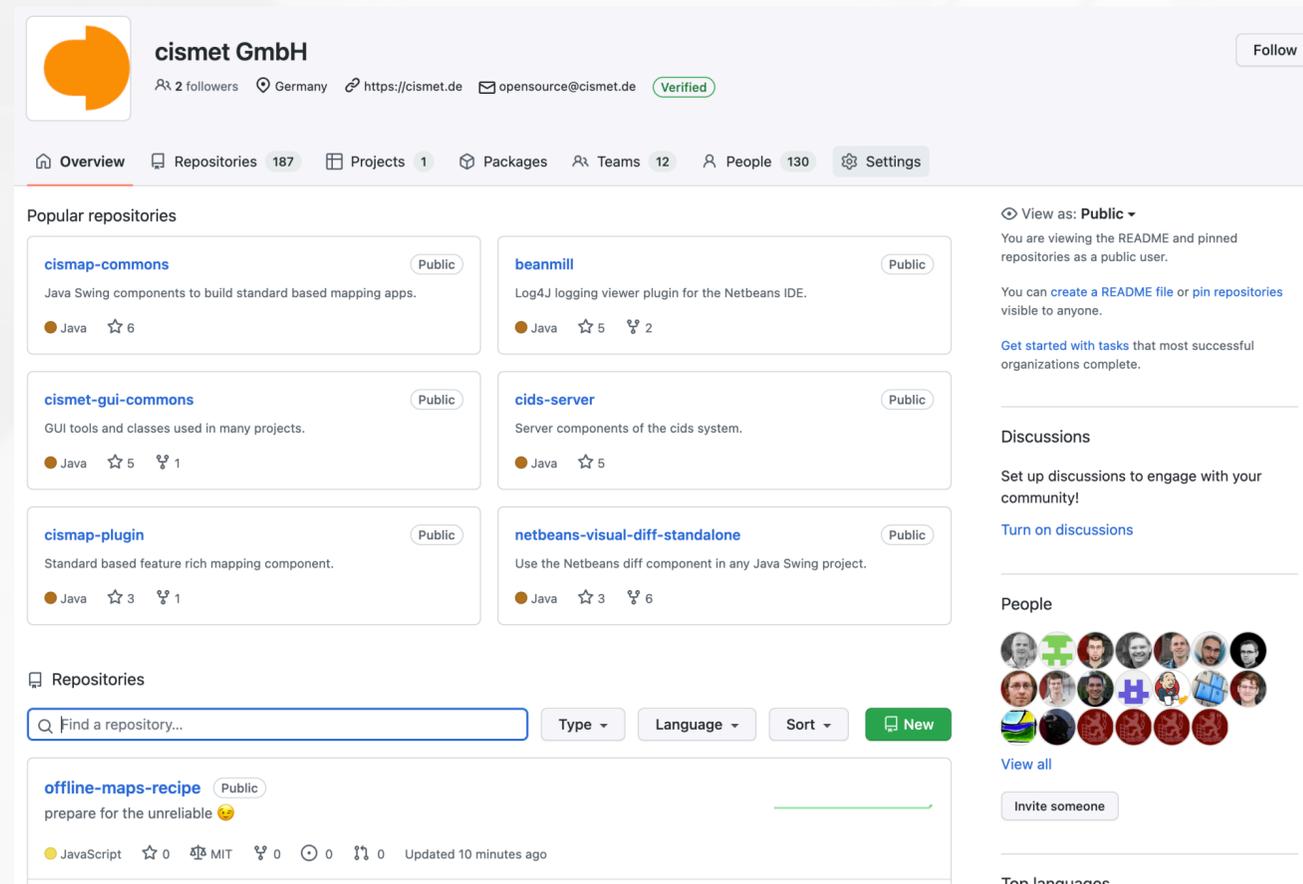
```
maplibreGl.addProtocol("indexedDB", (params, callback) => {
  let url = params.url.replace("indexedDB://", "");
  console.log("indexeddb protocol interception", url);
  customOfflineFetch(url, offlineConfig).then((buffer) => {
    if (buffer) {
      callback(null, buffer, null, null);
    } else {
      callback(null, new ArrayBuffer(), null, null);
    }
  });
});
```

# Demoprojekt

- GitHub Projekt: <https://github.com/cismet/offline-maps-recipe>
- Laufende Instanz: <https://cismet.github.io/offline-maps-recipe>



- Was fehlt noch:
  - PWA Setup



- Sprites über ServiceWorker prefetching (funktioniert nicht über addProtocol 🙄)

# Bei Fragen zu dem Thema oder ...



## Vielen Dank

cismet GmbH  
Im Kleegarten 6  
66636 Tholey

cismet Cityoffice  
SAR.FACTORY - Ursulinenstraße 35  
66111 Saarbrücken

[thorsten.hell@cismet.de](mailto:thorsten.hell@cismet.de)

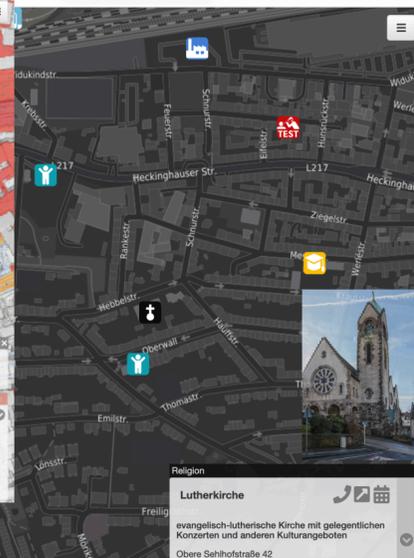
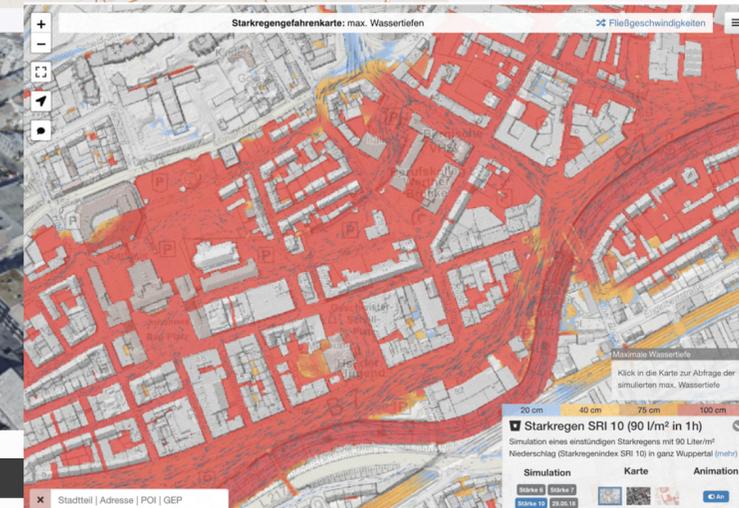
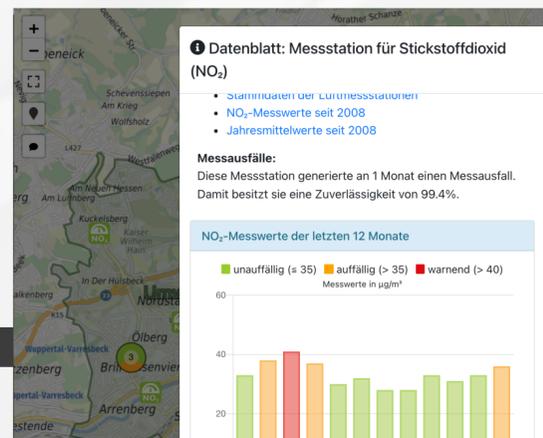
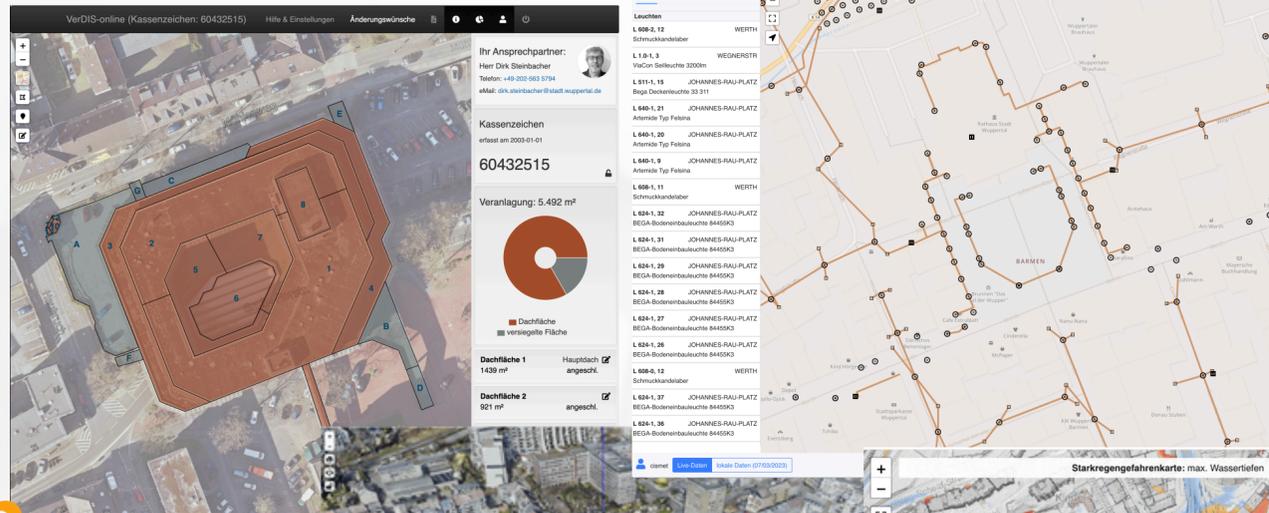
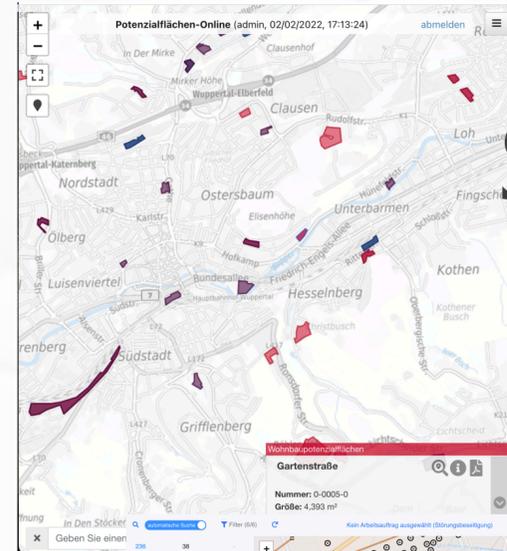
...bei Interesse an einem spannenden Job 😊: einfach ansprechen!



OpenSource GIS Entwicklung

MapLibre  
Leaflet  
CesiumJS  
100% Remote  
DigitalTwin  
PostGIS  
Docker  
React

Student / Geek / Nerd  
Fachinformatiker  
Bachelor / Master  
Teams



And it just works ...



Vielen Dank

cismet GmbH  
Im Kleegarten 6  
66636 Tholey

cismet Cityoffice  
SAR.FACTORY - Ursulinenstraße 35  
66111 Saarbrücken

[thorsten.hell@cismet.de](mailto:thorsten.hell@cismet.de)

# Offline Package erzeugen

- osm.pbf herunterladen
- mbtiles erzeugen (Openmaptiles oder Tilemaker)
- auspacken (mb-util)
- auspacken der einzelnen pbf's
- Fonts dazulegen
- packen
- md5 berechnen und in Datei schreiben