



Von WMS zu WMTS zu Vektor-Tiles

@PirminKalberer
Sourcepole AG, Zürich
www.sourcepole.ch





› Vektor Tile Demo

› <https://www.mapbox.com/maps/>



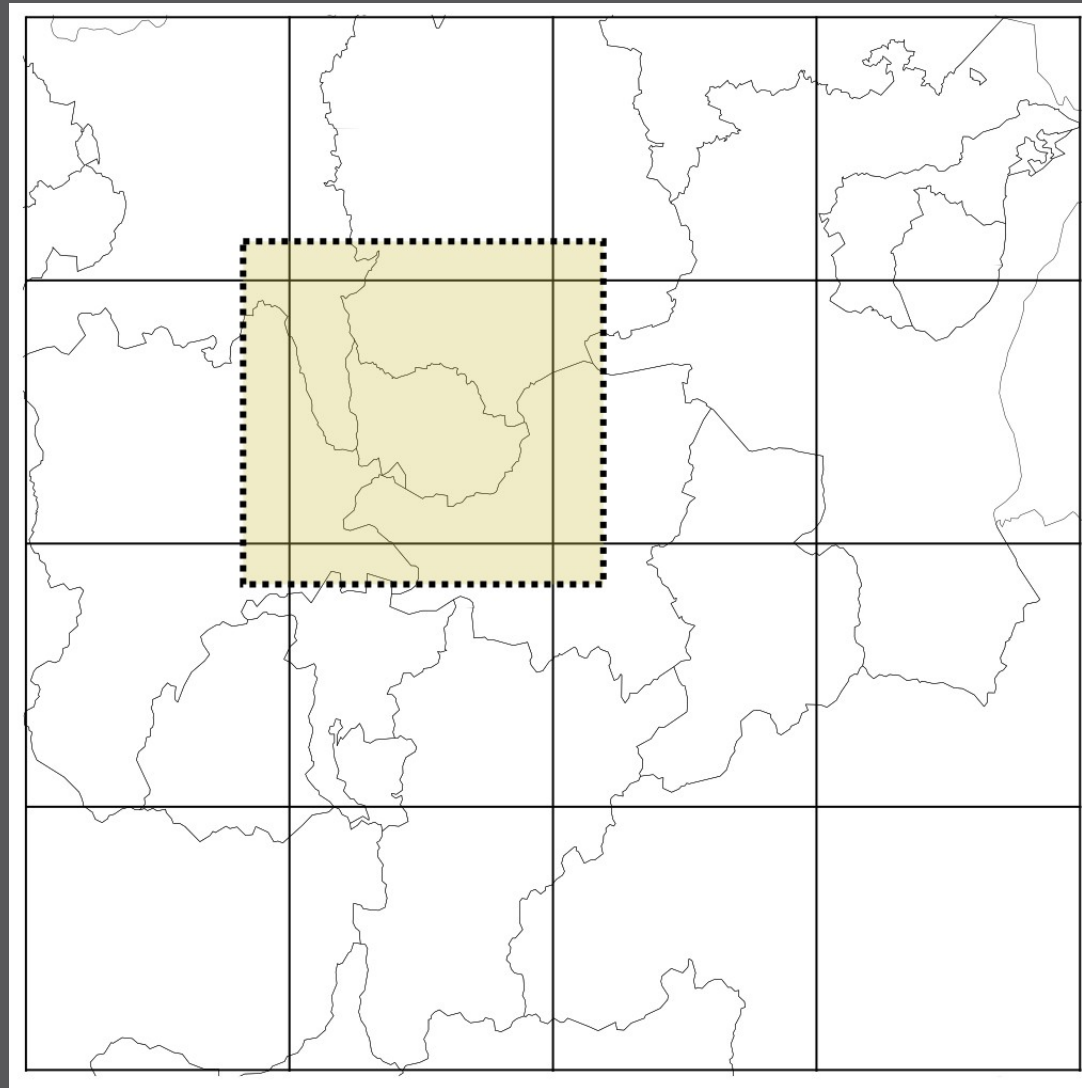
Mapbox Vector Tiles

<https://github.com/mapbox/vector-tile-spec>

- › Protocol Buffer Format (PBF, Binär, Streamable)
- › Kacheln 256x256 plus Puffer
- › Geometrie in Bildschirmkoordinaten (Ganzzahlen, ZigZag encoded)
- › Multipunkt/Multiline/Multipolygon
- › Datenattribute (Feature-ID optional)
- › Mehrere Layer pro Tile



Mapbox Vector Tiles





WMS -> WMTS -> Vektor Tiles

› WMS

- › Keine Tiling-Probleme (Labels, etc.)

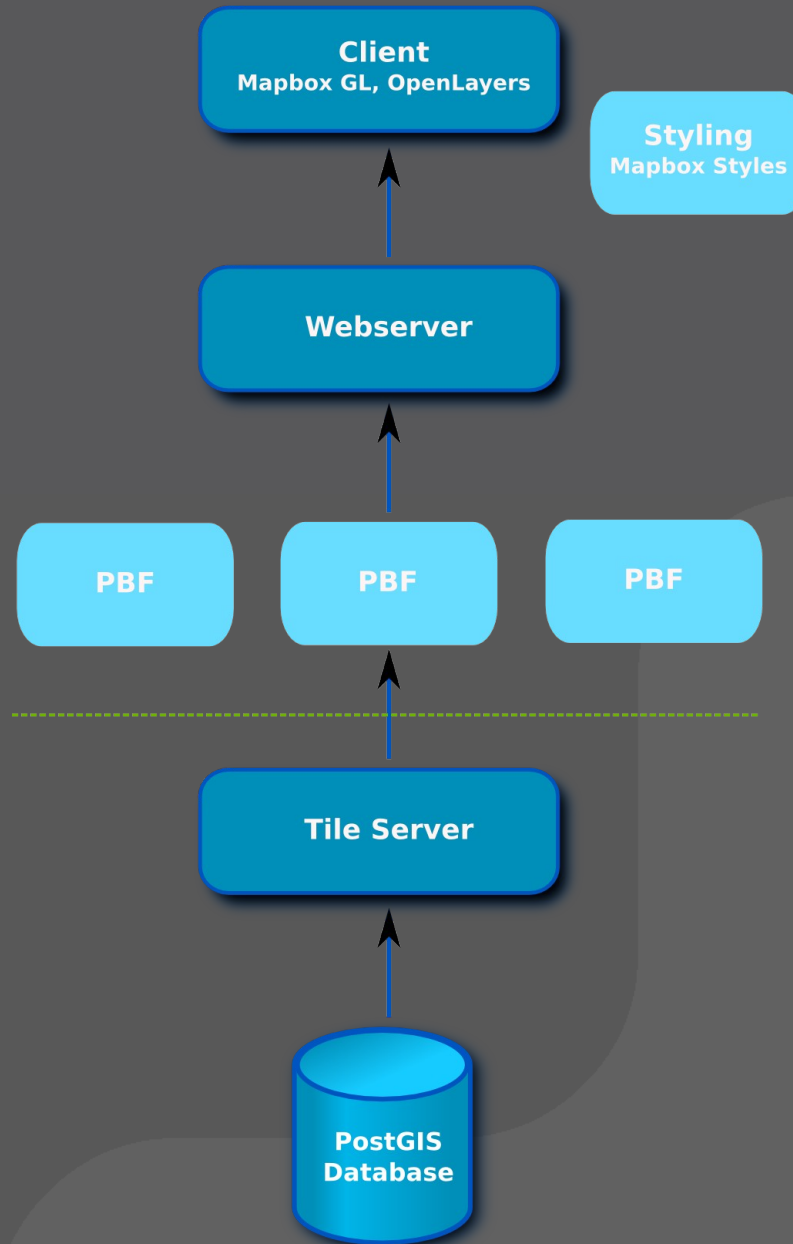
› WMTS

- › Skalierbarkeit
- › Caching (Server und Client)

› Vector Tiles

- › Skalierbarkeit
- › Caching (Server und Client)
- › Interaktivität
- › Flexible Darstellung (client-seitiges Rendering)
- › Hi-DPI

Vektor Tile Stack für Custom-Data



Vektor Tile Stack für Custom-Data

➤ MVT Implementationen:

<https://github.com/mapbox/awesome-vector-tiles>

Vektor Tile Server für Custom-Data

- › node-mapnik (Kartotherian, tesseract)
- › Tilezen tileserver
- › Tegola
- › t-rex
- › GeoServer
- › PostGIS ST_AsMVT



node-mapnik

- › **Node-mapnik: Bindings to mapnik for node.js**
- › **Kartotherian: Wikipedia Maps Tile Server**
- › **tessera: tilelive-based tile server**
- › **Basieren auf Mapbox-Komponenten**
- › **Wenig Dokumentation**
- › **Layer-Konfiguration: Mapnik layer definition (XML) normalerweise generiert aus YAML**
- › **Vektor Tile Formate: MVT**
- › **Datenquellen: PostGIS**



Tilezen tileserver

- › Python server von Mapzen
- › Layer definitions: SQL in jinja2 templates, YAML
- › Layer-Konfiguration: SQL (jinja2 Templates), YAML
- › Vektor Tile Formate: MVT, TopoJSON, GeoJSON
- › Datenquellen: PostGIS
- › Umfangreiche Geometrie Post-Processing Funktionen

- › **Programmiersprache Go**
- › **Layer-Konfiguration: SQL in TOML**
- › **Vektor Tile Formate: MVT**
- › **Datenquellen: PostGIS**
- › **Single Executable**
- › **Unterstützung mehrerer Datenquellen**

- › **Programmiersprache Rust**
- › **Layer-Konfiguration: SQL in TOML**
- › **Vektor Tile Formate: MVT**
- › **Datenquellen: PostGIS**
- › **Single Executable**
- › **Eingebauter Webserver mit Viewer**
- › **Cache Seeding Tool**
- › **Custom Tiling Schemas für Nicht-Merkator Projektionen**
- › **Embedded Mapbox JS Styling**



- › Programmiersprache Java
- › Layer-Konfiguration: SQL in XML (Web-Backend)
- › Vektor Tile Formate: MVT, GeoJSON
- › Datenquellen: PostGIS, u.v.m



PostGIS ST_AsMVT

- › PostGIS Funktionen (C)
- › Layer-Konfiguration: -
- › Vektor Tile Formate: MVT
- › Datenquellen: PostGIS
- › Benutzbar für eigene Server-Implementation



Workflow mit t-rex (1)

› Installation:

- › Download binary from <https://github.com/pka/t-rex/releases>
- › Unzip

› Start server:

```
t_rex serve --dbconn postgresql://user@host/database
```



Workflow mit t-rex (2)

› Tiles served at:

`http://localhost:6767/{layer}/{z}/{x}/{y}.pbf`

› List of layers and built-in viewer:

`http://localhost:6767/`



Workflow mit t-rex (3)

› Generate a configuration template:

```
t_rex genconfig --dbconn postgresql://user@host/database
```

› Run with configuration file:

```
t_rex serve --config myconfig.cfg
```



Workflow mit t-rex (4)

› Generate tile cache:

```
t_rex generate --config myconfig.cfg
```

› Create MBTiles File:

```
mb-util --image_format=pbfs /tmp/mvtcache/ne ne.mbtiles
```



Styling / Viewer

- › **Mapbox Styles (JSON)**
 - › Viewer:
 - › Mapbox GL JS
 - › OpenLayers 3/4
 - › Style Editor (OSS)
 - › Maputnik
- › **Mapzen Tangram Styles (YAML)**
 - › Viewer:
 - › Tangram
 - › Style Editor (OSS)
 - › Tangram Play



Credits / Links

- › **Paul Norman:**
 - › <http://www.paulnorman.ca/blog/2016/11/serving-vector-tiles/>
- › <https://www.mapbox.com/vector-tiles/>
- › <https://mapzen.com/projects/vector-tiles/>
- › <https://github.com/tilezen/tileserver>
- › <http://tegola.io/>
- › <https://github.com/pka/t-rex>
- › <http://docs.geoserver.org/latest/en/user/extensions/vectortiles/>
- › http://postgis.net/docs/manual-dev/ST_AsMVT.html



Danke!



@PirminKalberer