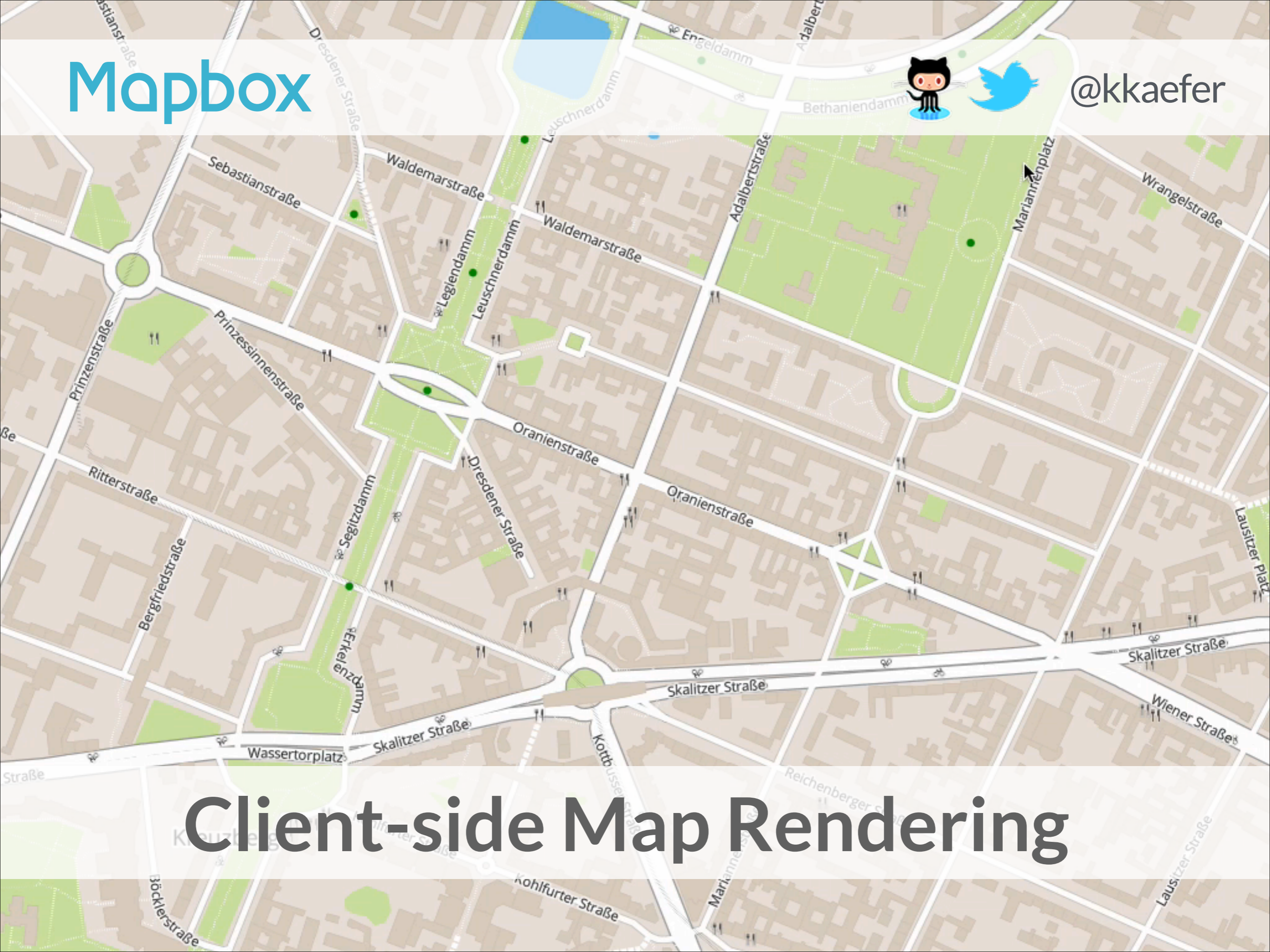


Mapbox



@kkaefer

Client-side Map Rendering



Rendering im Browser

CPU

Agg/Mapnik

Canvas

SVG

GPU

WebGL

Canvas

- Kothic JS (<http://kothic.org/>)
- Eigener Canvas-basierter Renderer
- **Nachteile**
 - Canvas ist recht langsam, > 16ms
 - Single-threaded, blockiert UI

Anti-Grain Geometry

- C++-Renderer
- Mit Emscripten nach JavaScript kompiliert
- **Vorteile:**
 - Kann im Webworker laufen
 - Prinzipiell kann man Mapnik zum laufen bekommen
- **Nachteile:**
 - Langsam, CPU-basiert

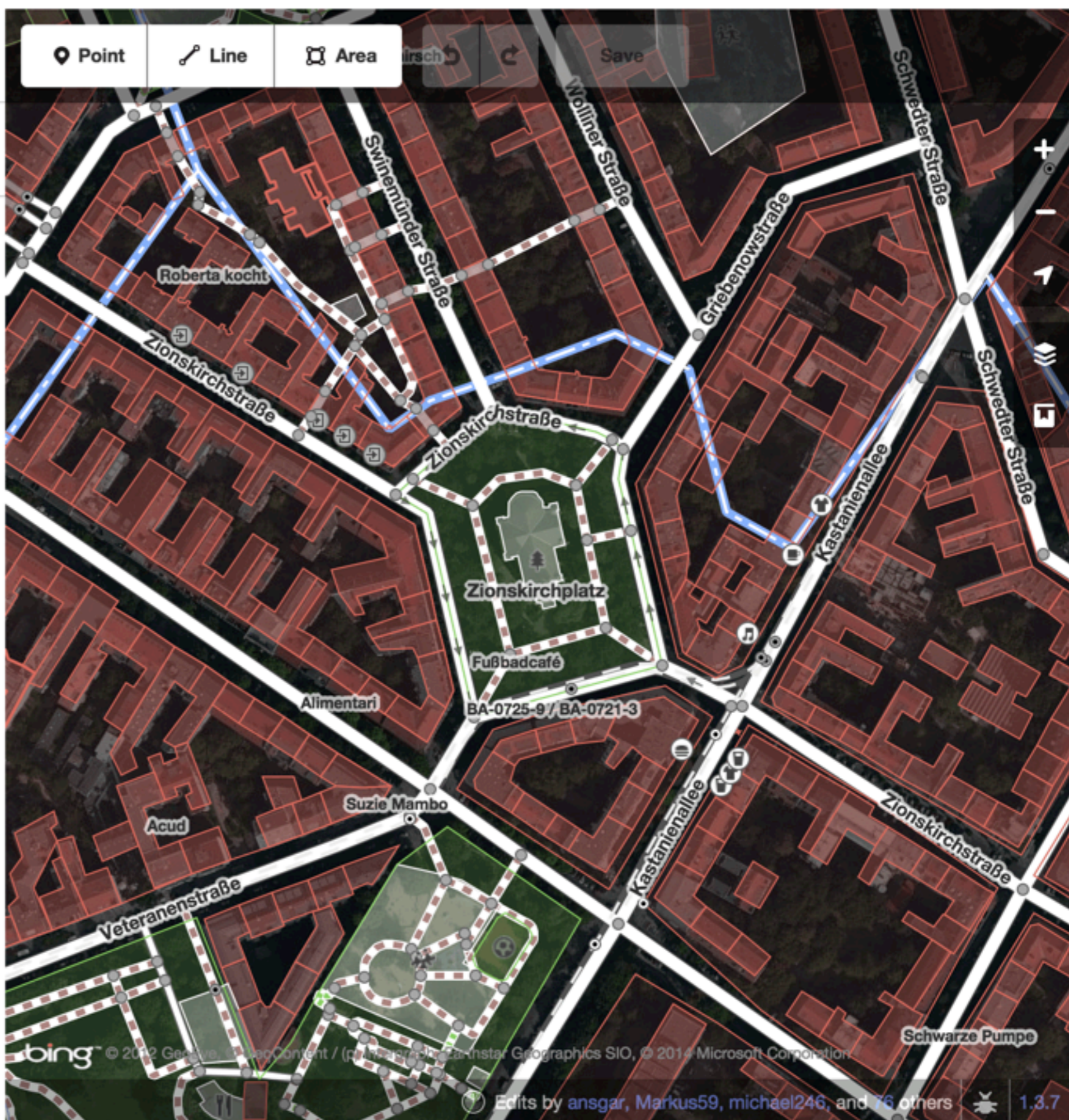
SVG

- SVG-Tiles erzeugen (<https://github.com/springmeyer/svg-tiles>)
- Als `` in Leaflet (Demo)
- **Vorteile:**
 - CSS für's Map-Styling
- **Nachteile:**
 - SVGs sind sehr langsam, schwierig zu rendern
 - SVG-Tiles sind recht groß, auch mit Kompression

Search features

Search

Point
Line
Area
Undo
Redo
Save



WebGL

- Basiert auf OpenGL ES 2.0
- In vielen Browsern verfügbar: IE, Chrome, Firefox, Opera
- Aber deaktiviert in Safari, iOS, langsamen Geräten
- **Vorteile:**
 - Hardware-Beschleunigung
 - Beliebige Kontrolle über aussehen
- **Nachteile:**
 - Enorme Komplexität
 - Extrem unterschiedliche Geräte, Treiber

WebGL

- **OpenScienceMap**
(<http://opensciencemap.org/map>)
- **Michal Migurski**
(<http://mike.teczno.com/notes/gl-solar-webgl-openstreetmap.html>)
- **Mapbox Renderer**
(Demo)

Datenerzeugung

Vektorkacheln

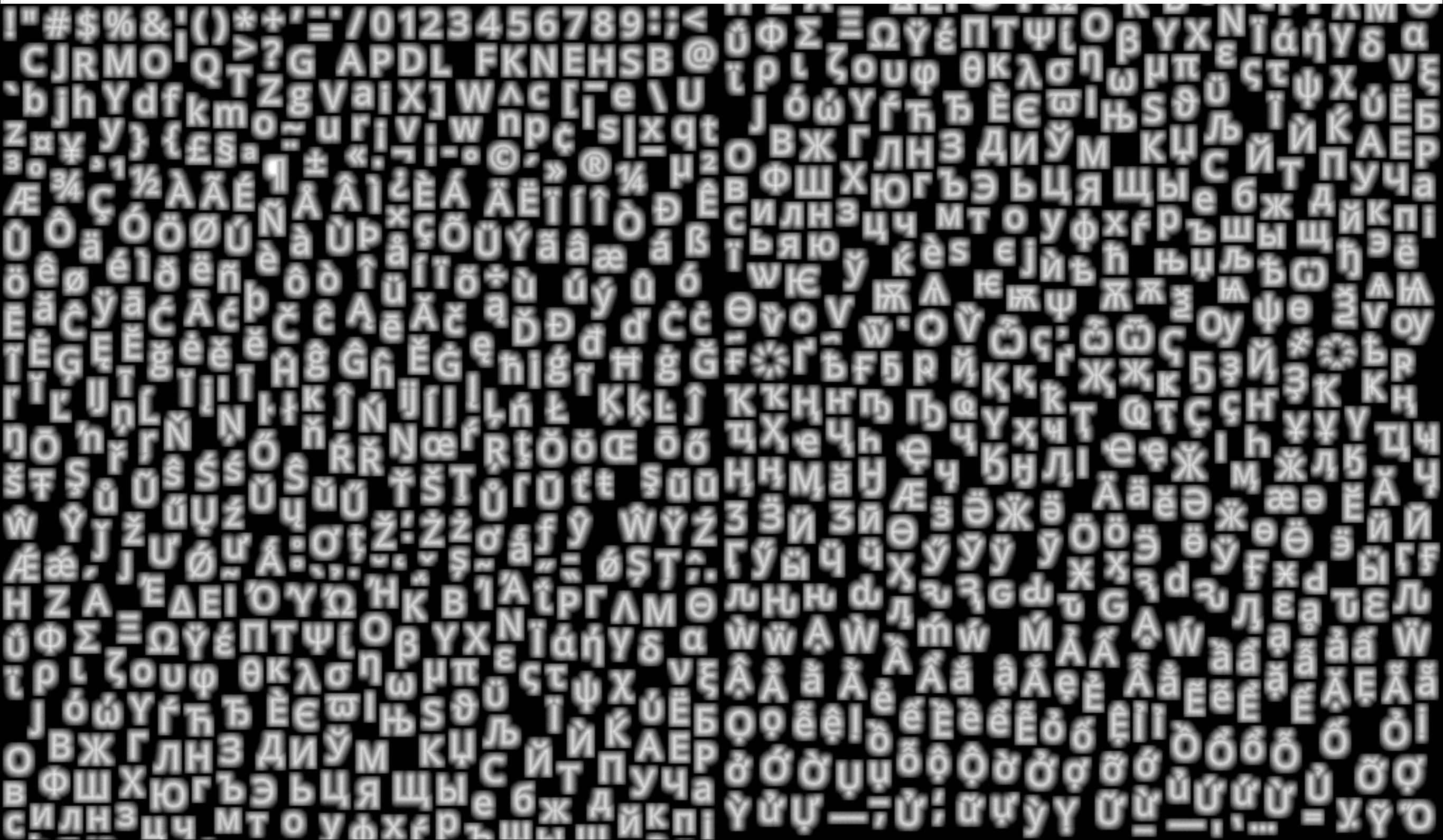
- Vektorkacheln als “Rendering Target”
 - Mapnik lädt und verarbeitet Daten
 - <https://github.com/mapbox/mapnik-vector-tile>
- TileStache
 - <http://tilestache.org/doc/TileStache.Goodies.VecTiles.html>

Text Rendering



the Free Type Project

Textur-Atlas



R

Demo

Text Shaping

Unicode-Charakter



Font-Glyph


Sigma → σοφός ← Sigma

Kontextabhängiges Shaping

Unshaped: مِ يَحْرُلْ اِنْ مَحْرُلْ اِهْلْ اَمْ سِبْ

Shaped: يَسْمُ اَللّٰهُ الرَّحْمٰنُ الرَّحِيْمُ

ابش 1234 دَفق



Bidirektionaler Text

ସି ସ ଧ ଧ ଧ
ସି ସ ଧ ଧ ଧ

Gestapelte Diakritische Zeichen

ସି ସ ଧ ଧ
ସି ସ ଧ ଧ

Vielen Dank. **Fragen?**



Mapbox