

# **OSM2World hinter den Kulissen**

Vortrag von Tobias Knerr  
FOSSGIS 2016

# **OSM2World**

## hinter den Kulissen

Vortrag von Tobias Knerr  
FOSSGIS 2016

# OSM2World

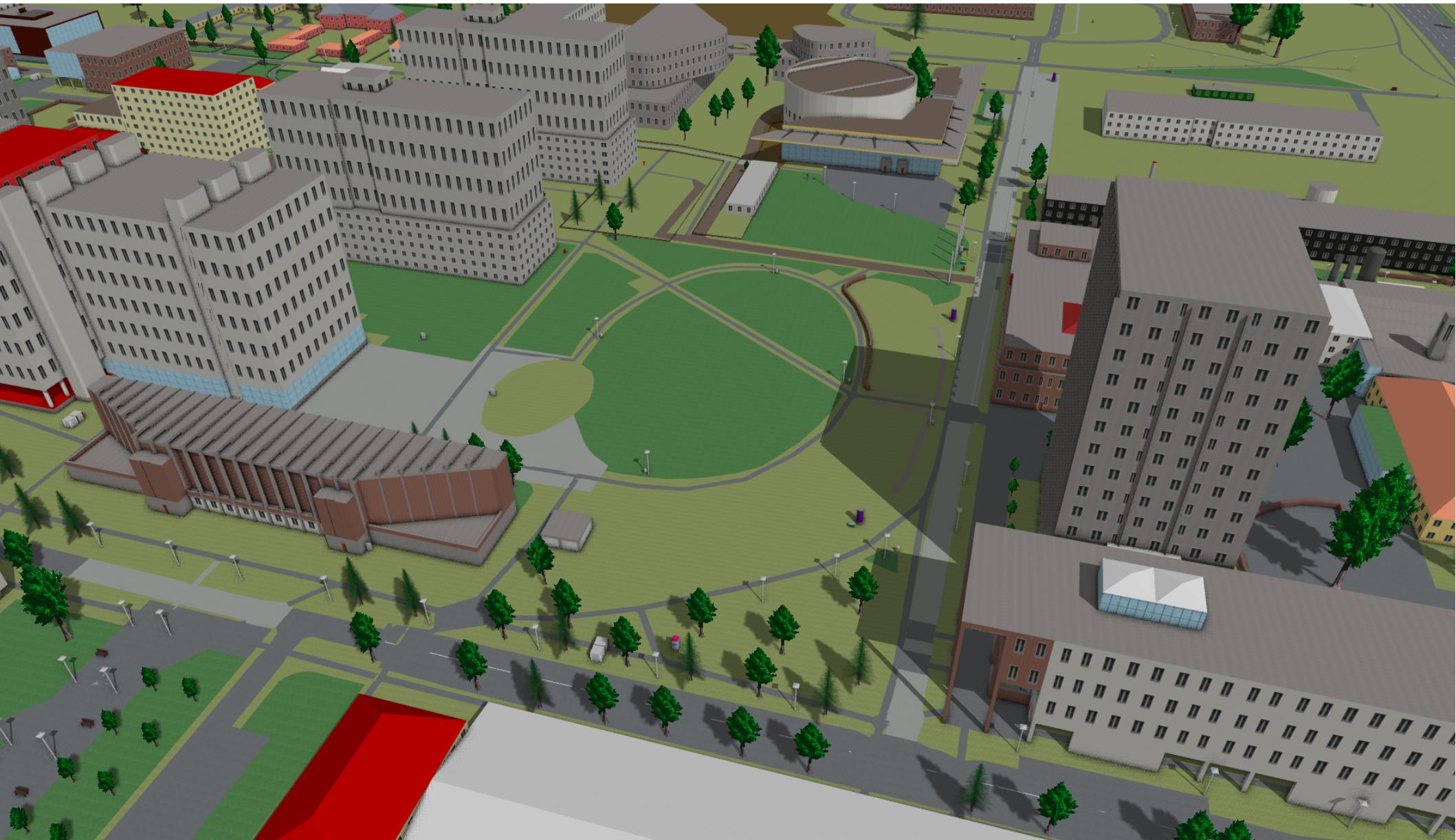
<http://osm2world.org>

- Freie Software (LGPL)
- 3D-Landschaften aus OpenStreetMap
- Exportieren oder direkt rendern

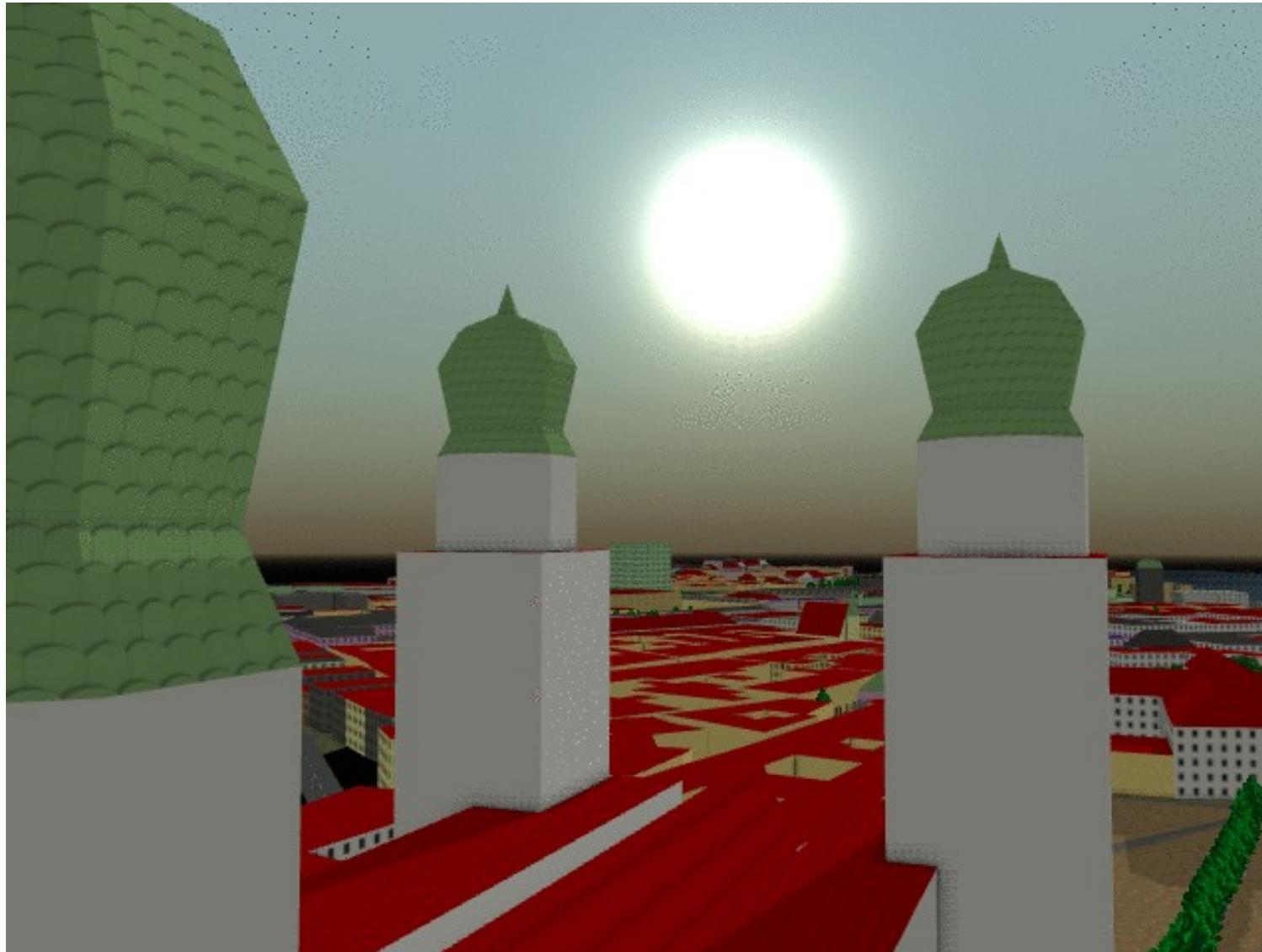
# Grafische Oberfläche

- Lokal installierte Java-Anwendung
- Lädt OSM-Daten aus diversen Quellen
- Stellt generierte Modelle via OpenGL dar
- Freie Kameraplatzierung und Navigation
- Modell-Export in alle unterstützten Formate

# OpenGL-Rendering

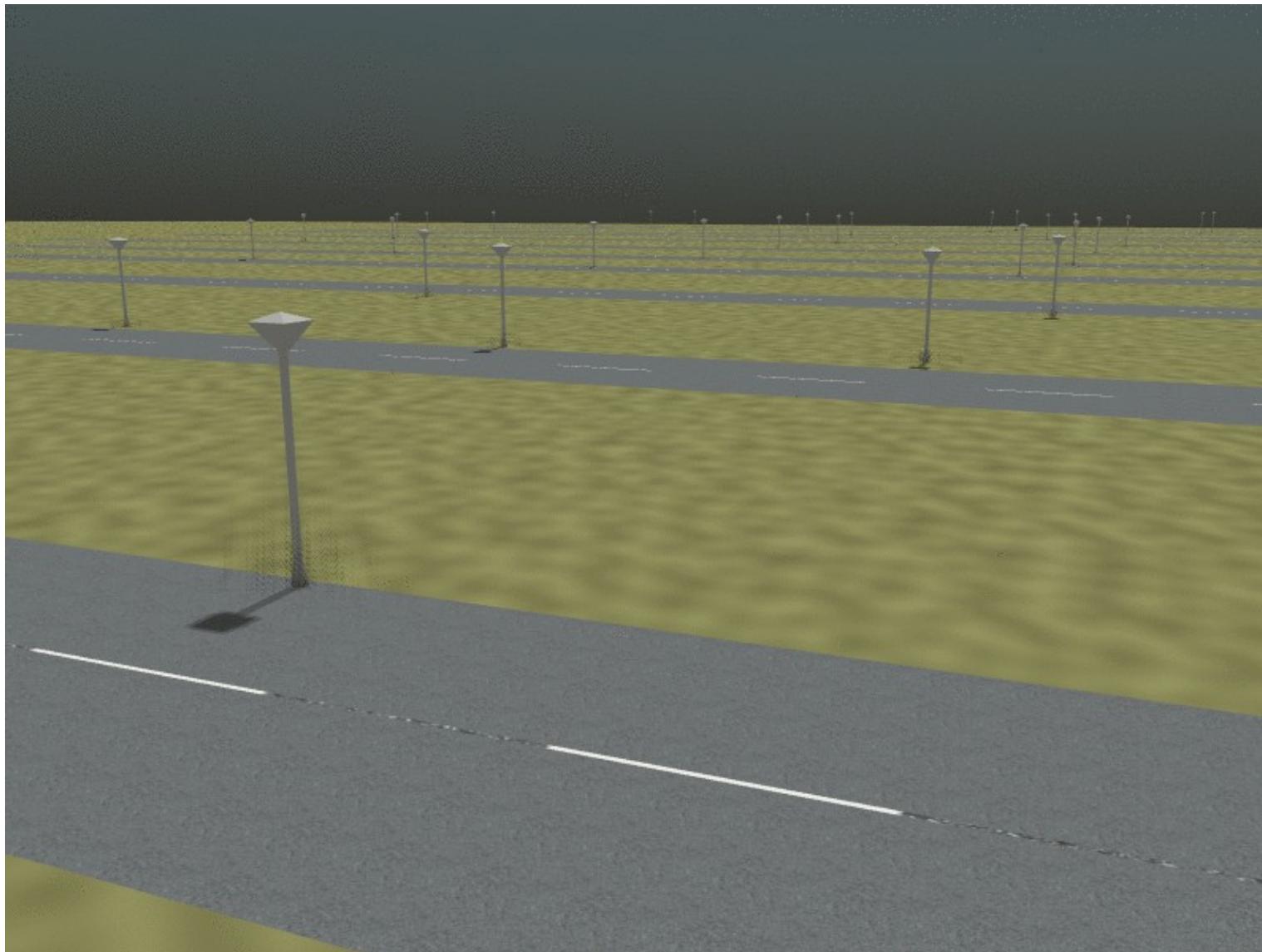


# Aktuell: Google Summer of Code

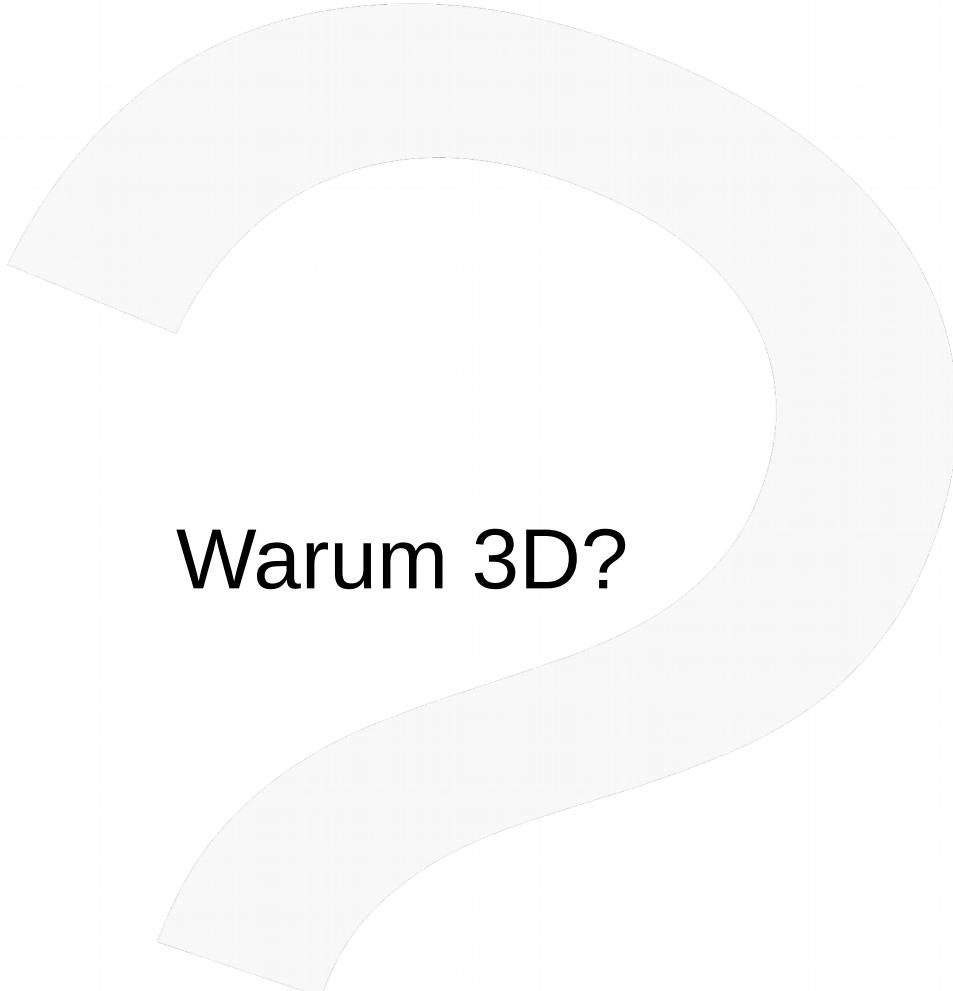


<http://www.openstreetmap.org/user/Zabot/diary/38956>

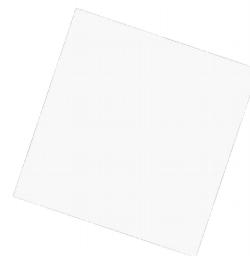
# Aktuell: Google Summer of Code



<http://www.openstreetmap.org/user/Zabot/diary/38986>



**Warum 3D?**



# Zunehmender Detailgrad der Daten

*OpenStreetMap: Hamburg vollständig  
in freier Online-Weltkarte erfasst*  
Computerbild, 2008

## Baum

Höhe: 9 m

Stammumfang: 1 m

Kronendurchmesser: 7 m

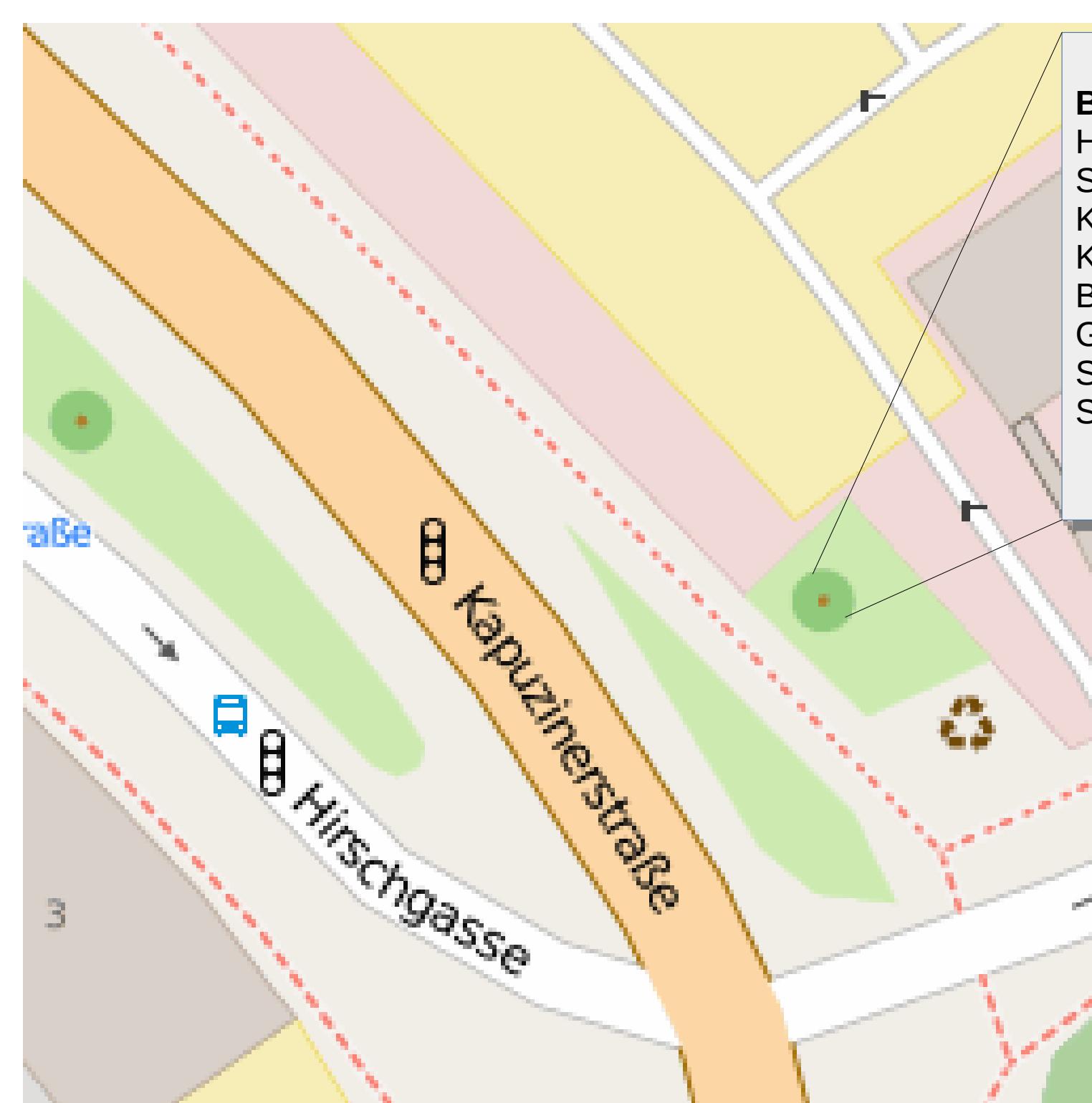
Klassifizierung: Stadtbäum

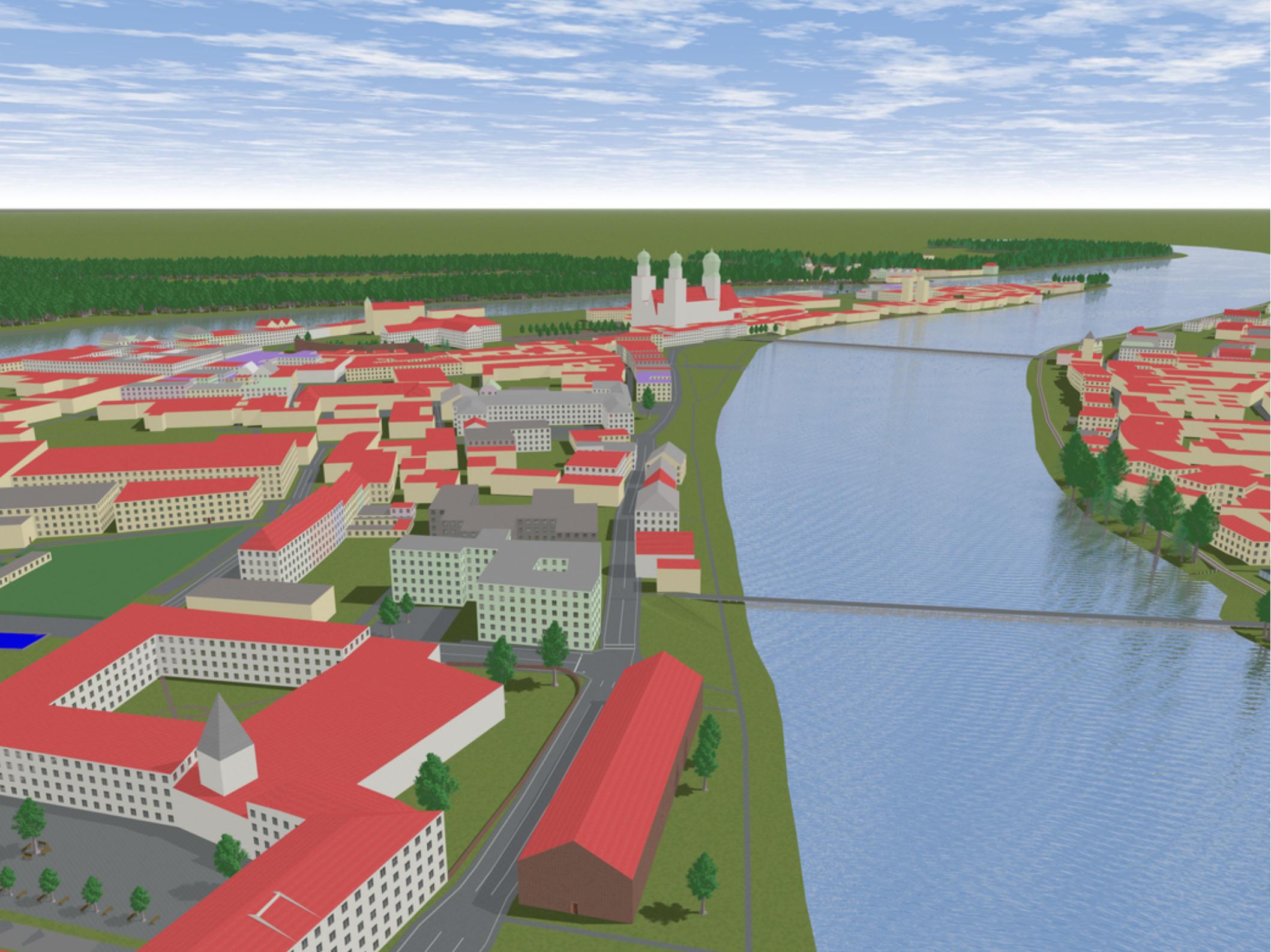
Blatttyp: Laubbaum

Genus: Betula

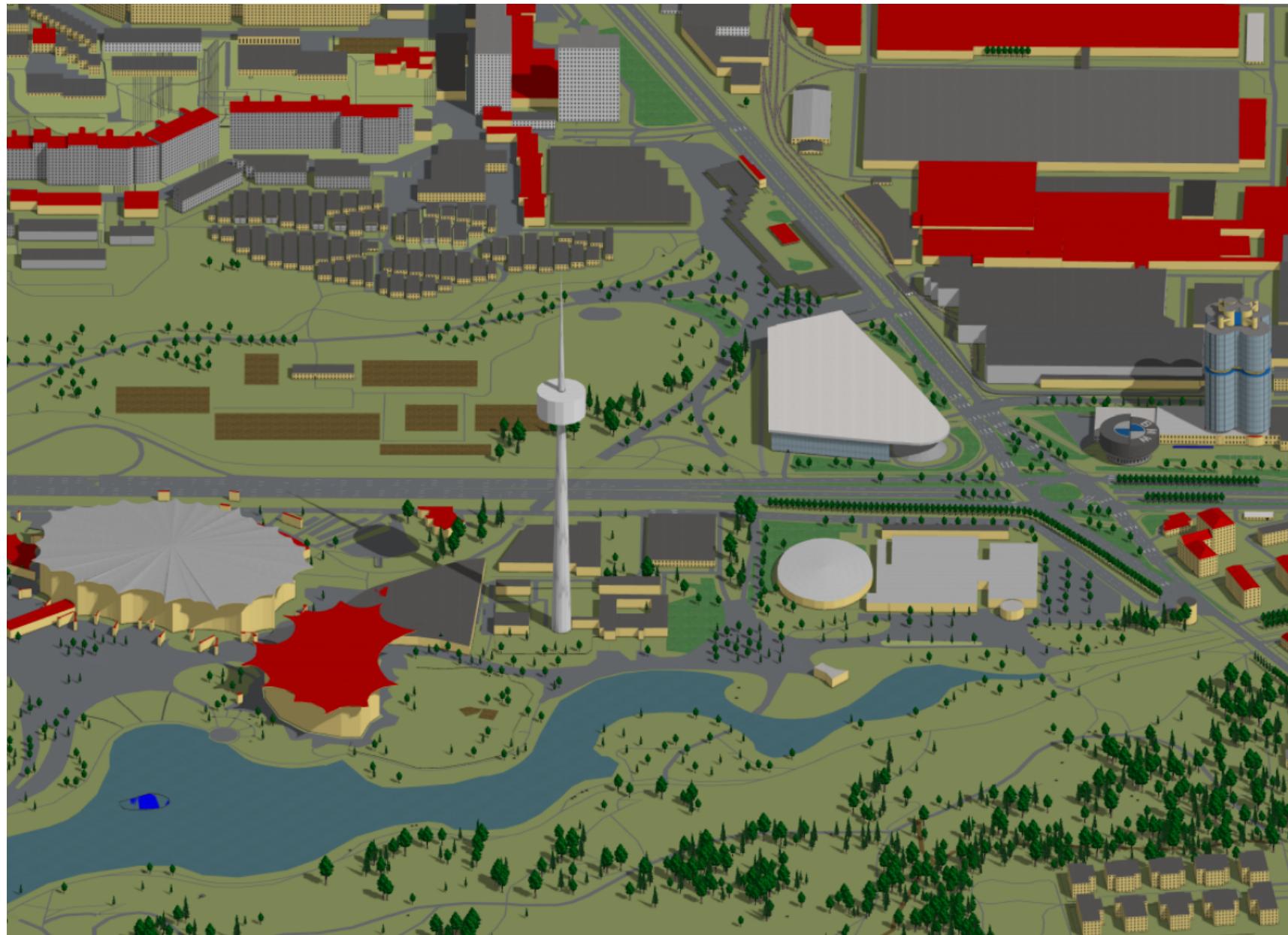
Spezies: Betula pendula

Spezies[de]: Sandbirke





# Pseudo-3D-Karten



# Spieleentwicklung

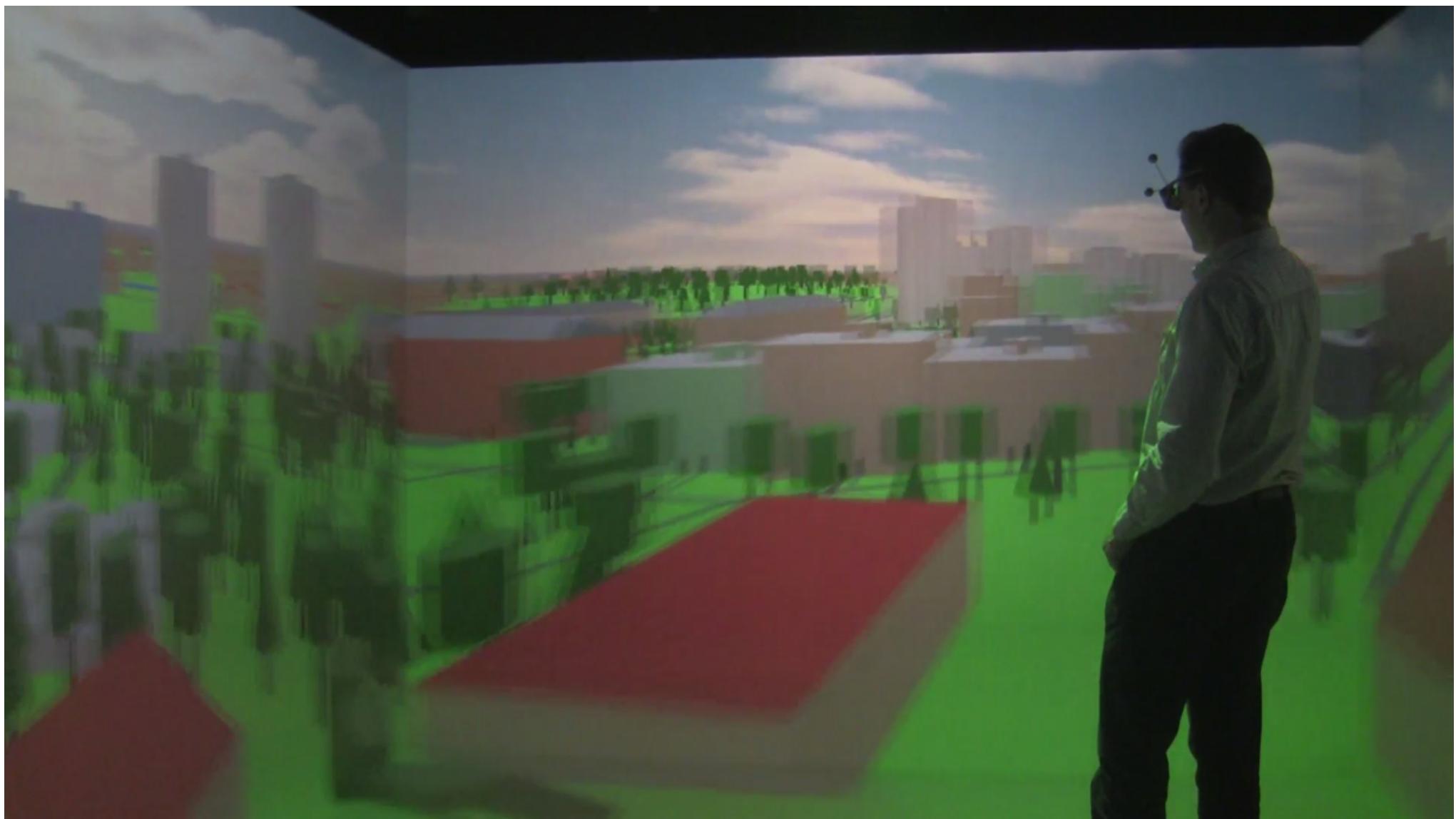


<http://wiki.openstreetmap.org/wiki/SuperTuxKart>

# Videos

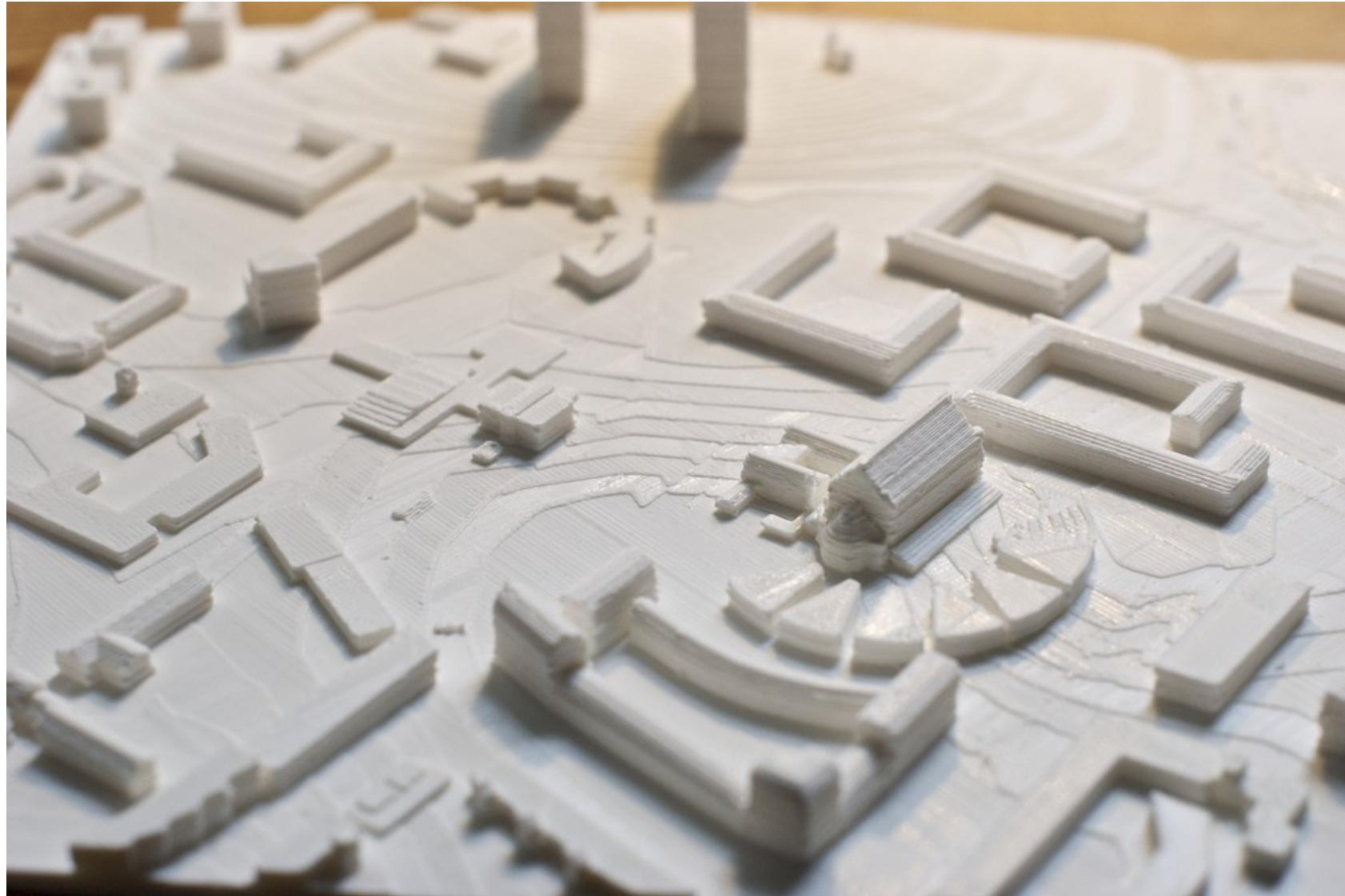


# Virtuelle Realität



<http://forum.osm.org/viewtopic.php?id=20012>

# 3D-Druck

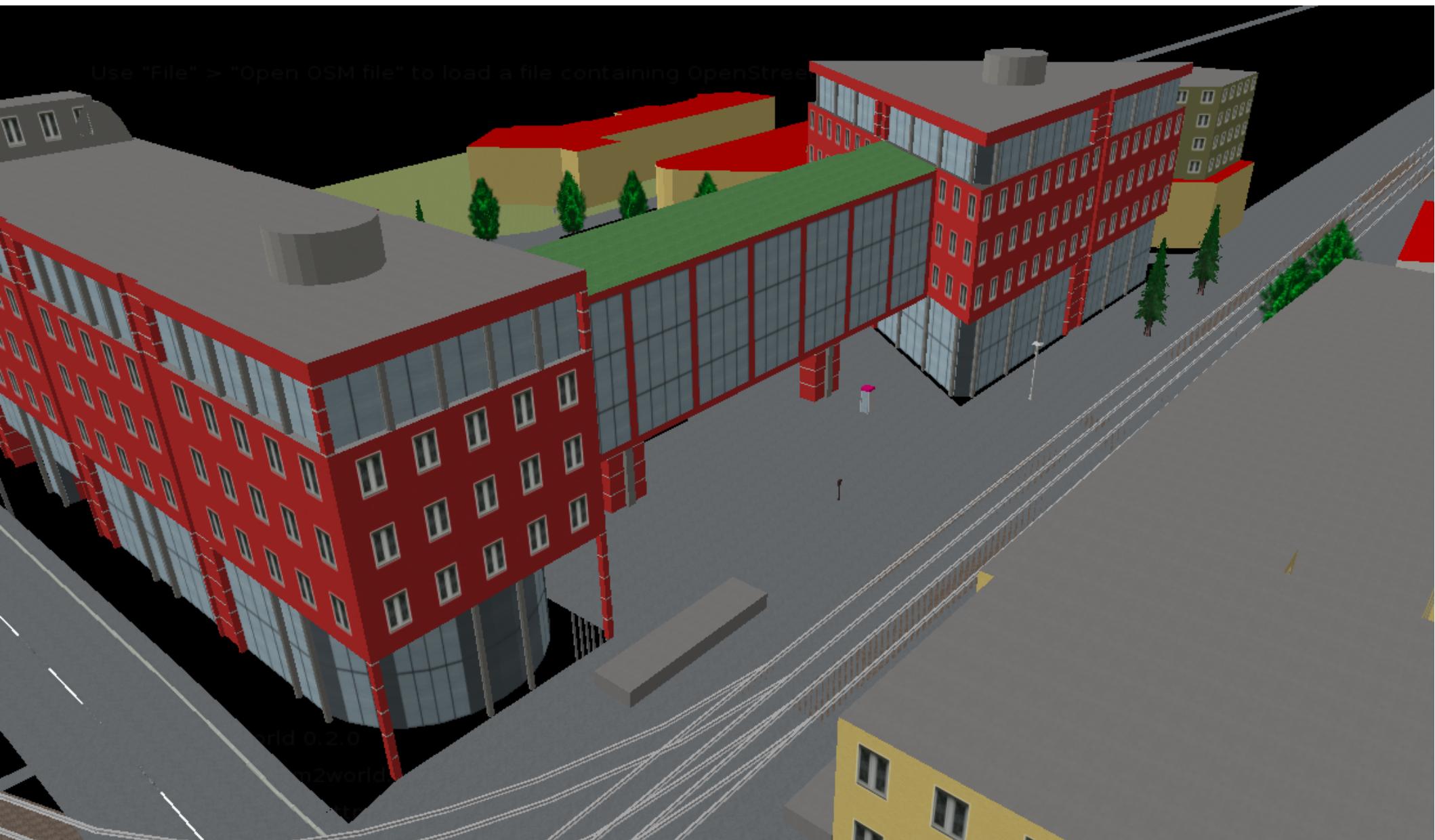


# **OSM2World**

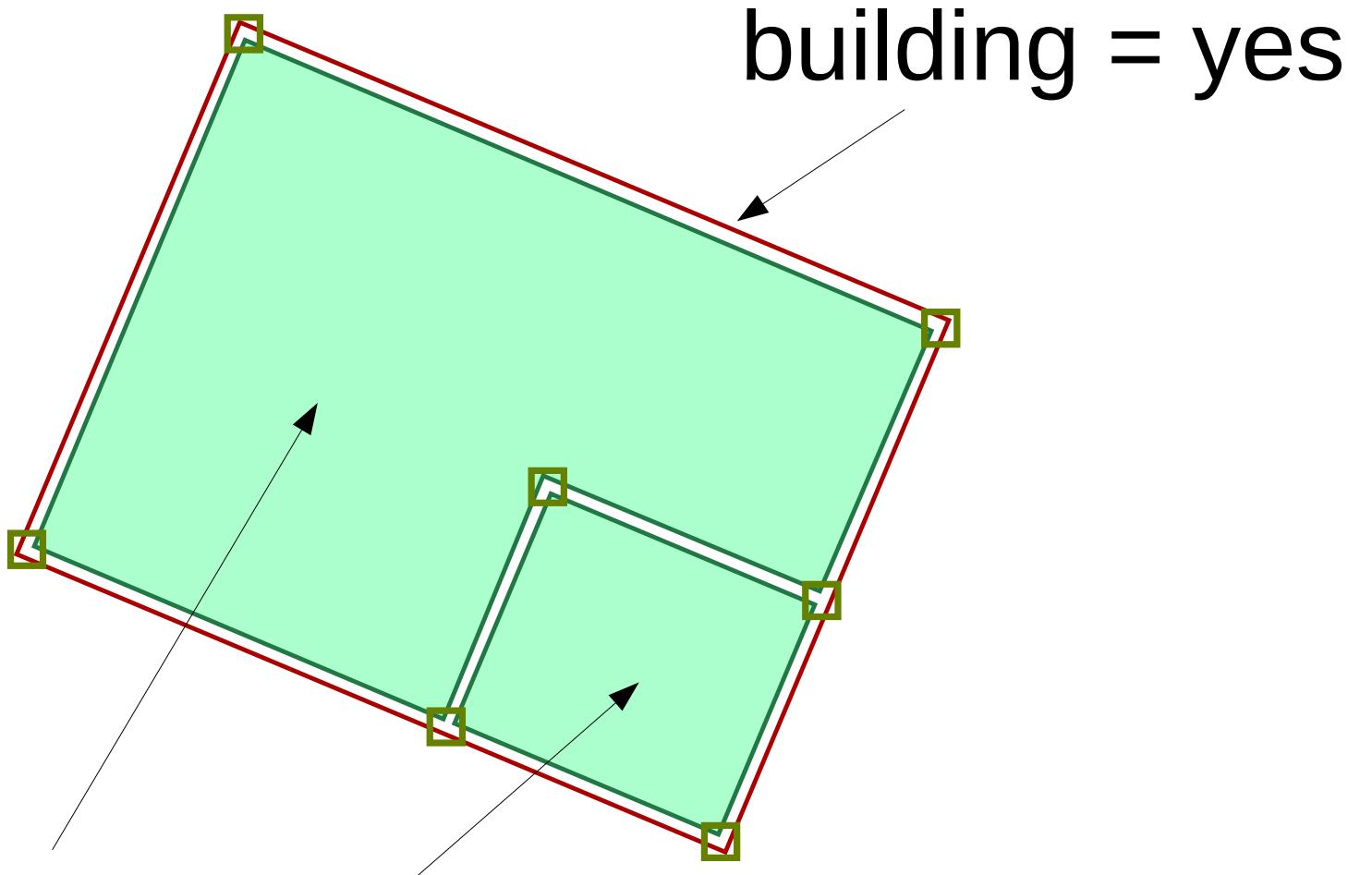
## hinter den Kulissen

Vortrag von Tobias Knerr  
FOSSGIS 2016

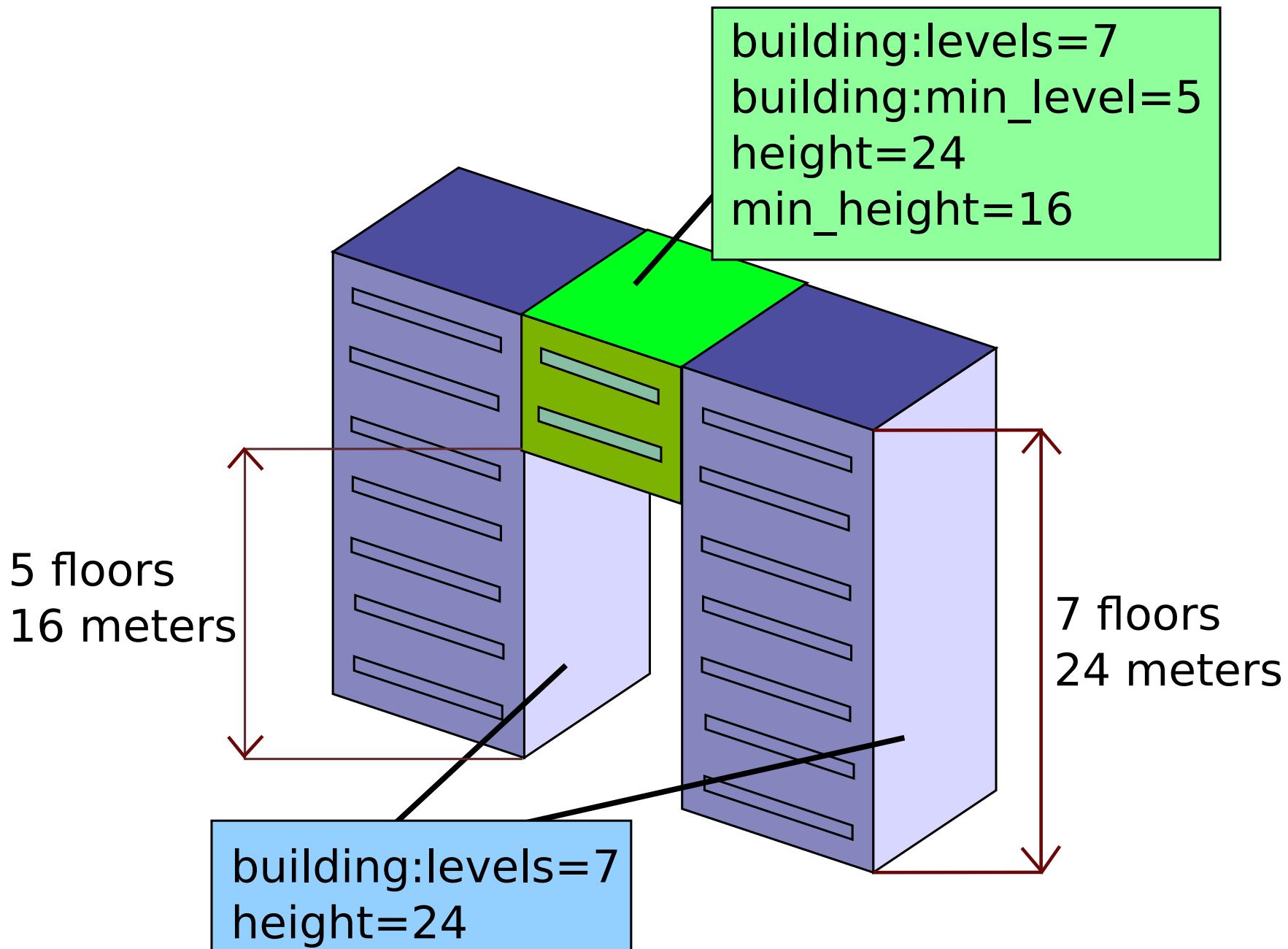
# Gebäude



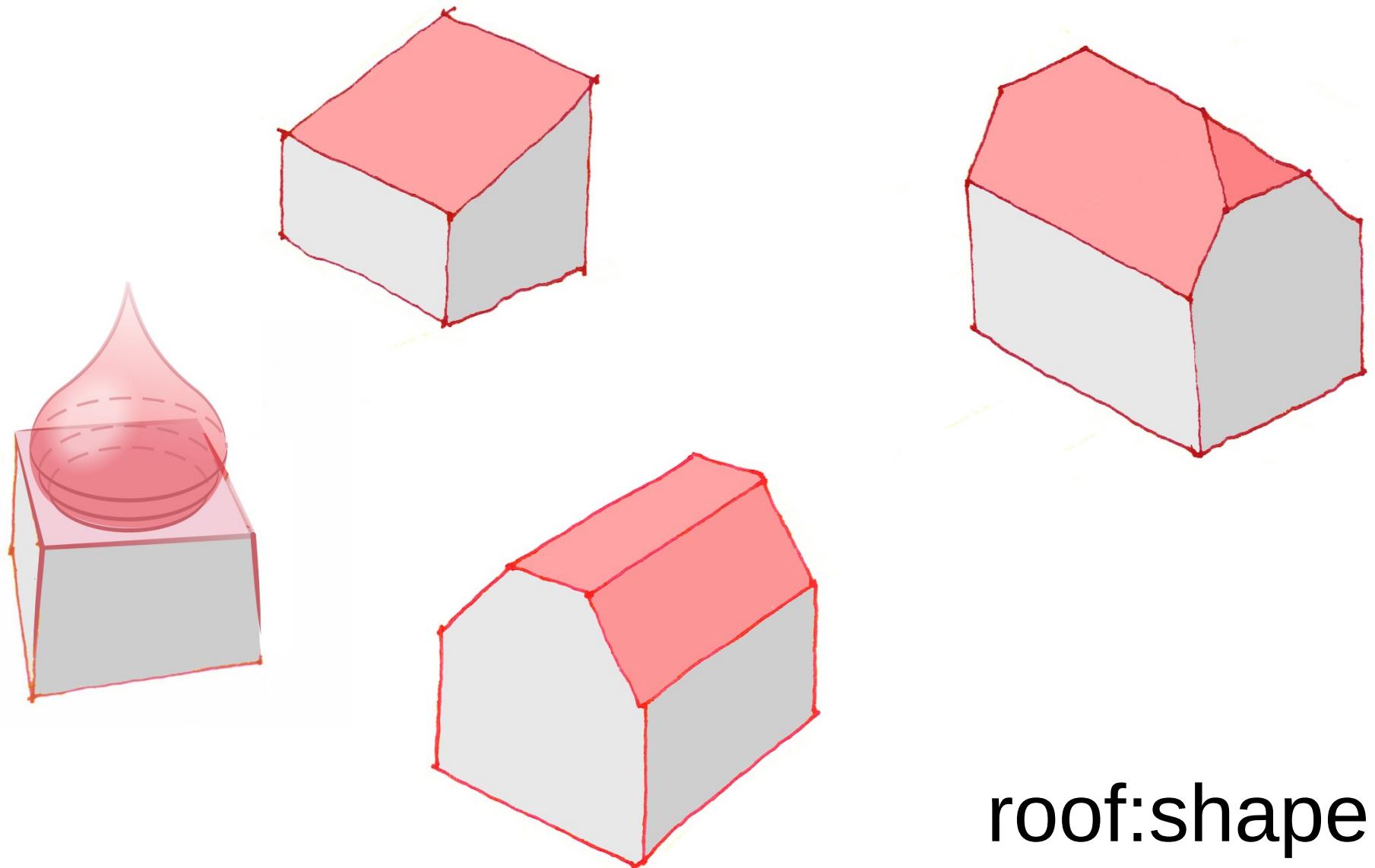
# Gebäude: Simple 3D Buildings



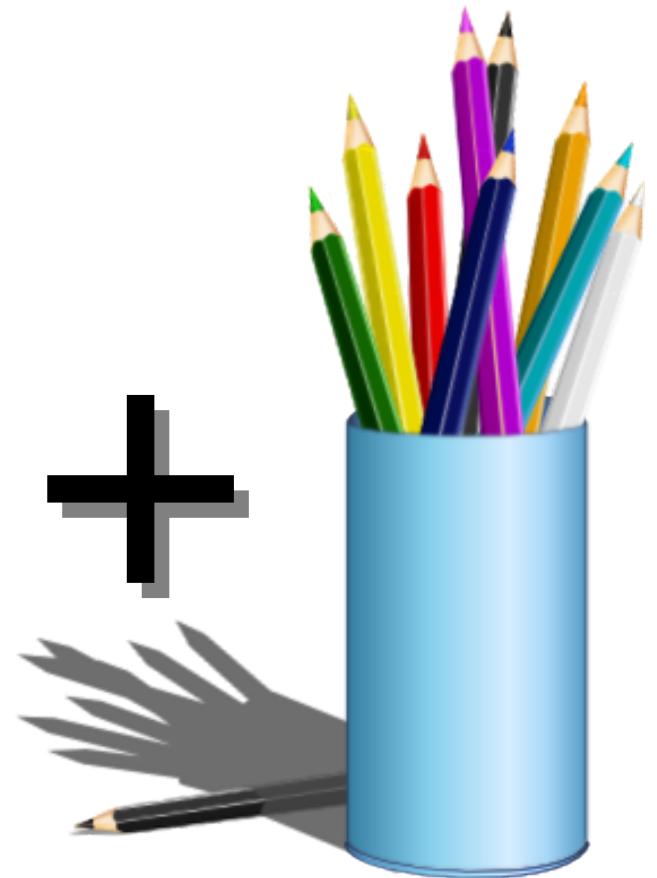
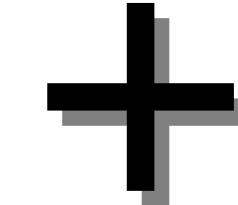
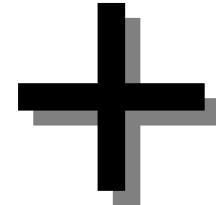
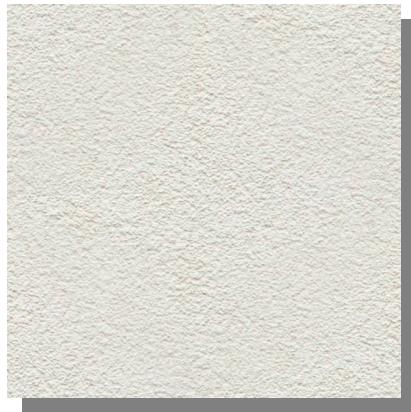
# Gebäude: Simple 3D Buildings



# Gebäude: Simple 3D Buildings



# Gebäude: Simple 3D Buildings



building:material

building:colour

# Gebäude: Building-Relation

Manchmal braucht man sie ...

... meist aber nicht!

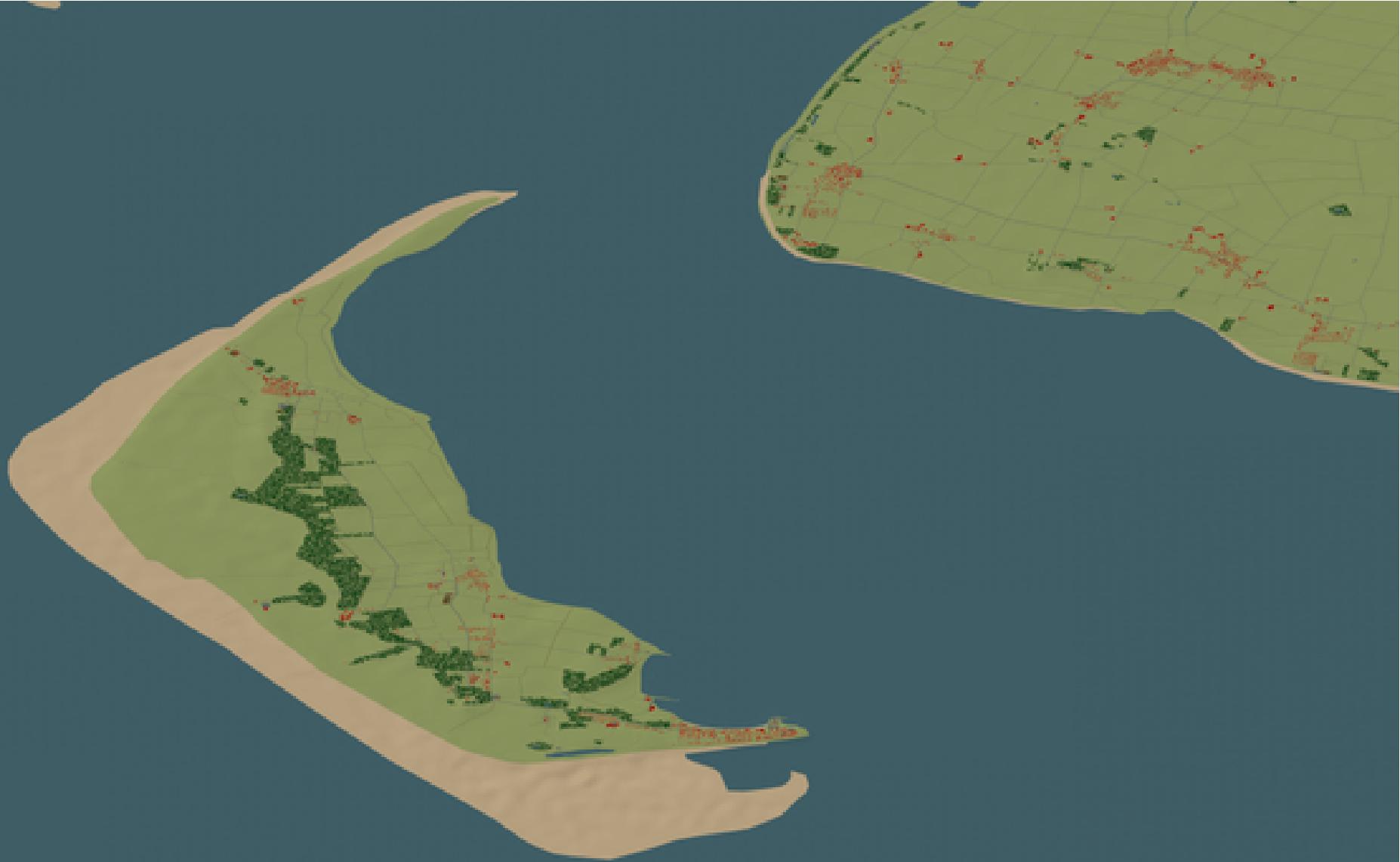
# Gebäude: Externe 3D-Modelle?



# Gebäude: Externe 3D-Modelle?

- Vorteile:
  - OSM-Daten werden einfacher: nur externer Link
  - Für manche Gebäude wohl zwingend nötig
- Aber:
  - Erlernen einer komplexen separaten Software
  - Ungenauigkeiten: Gebäude und Straßen, Nachbargebäude, Objekte und Geländemodell ...
  - Datensätze driften auseinander
- Koexistenz

# Küstenlinien



# Küstenlinien

Zur Erinnerung:

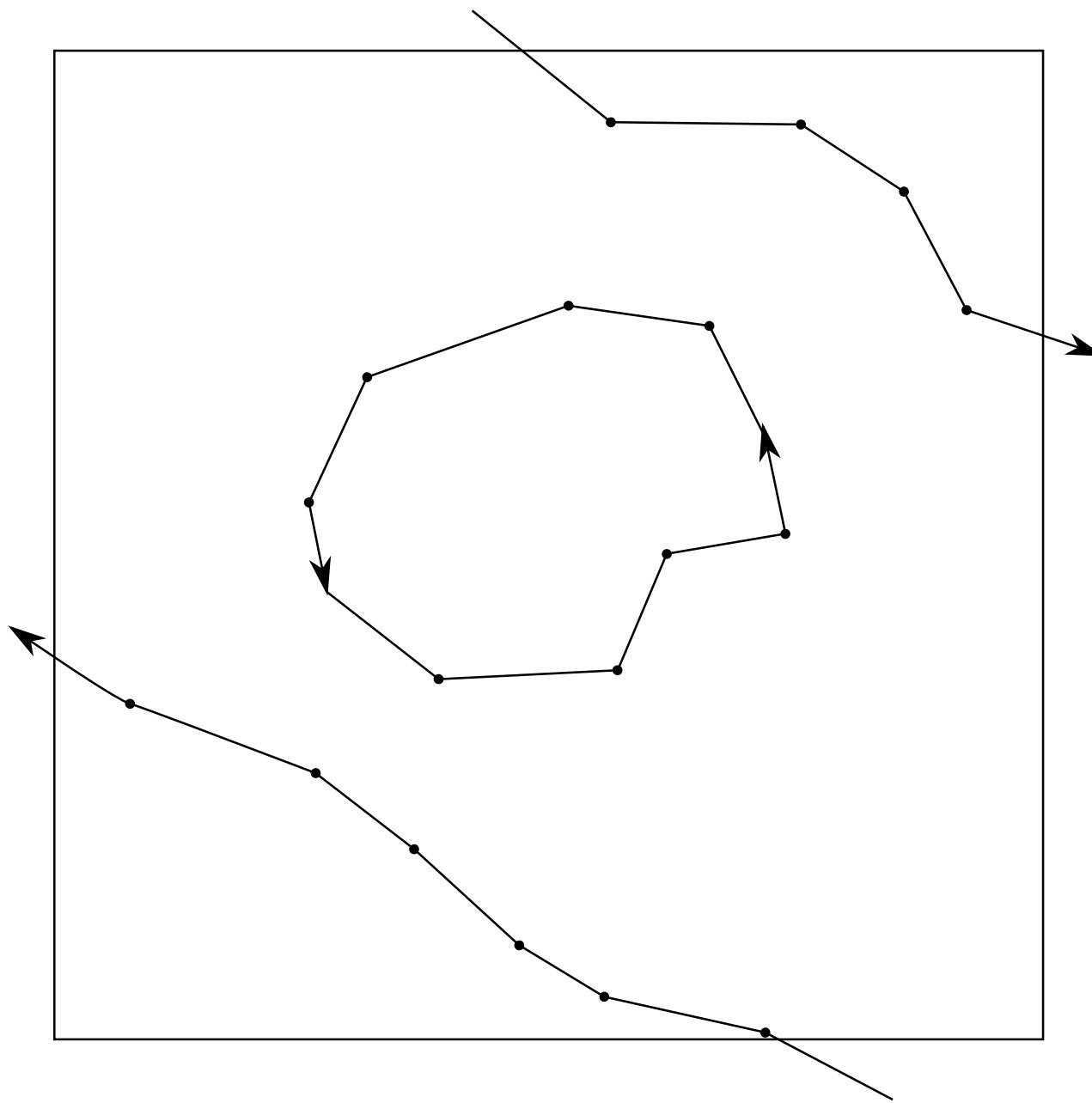
natural=coastline

Land links, Wasser rechts!

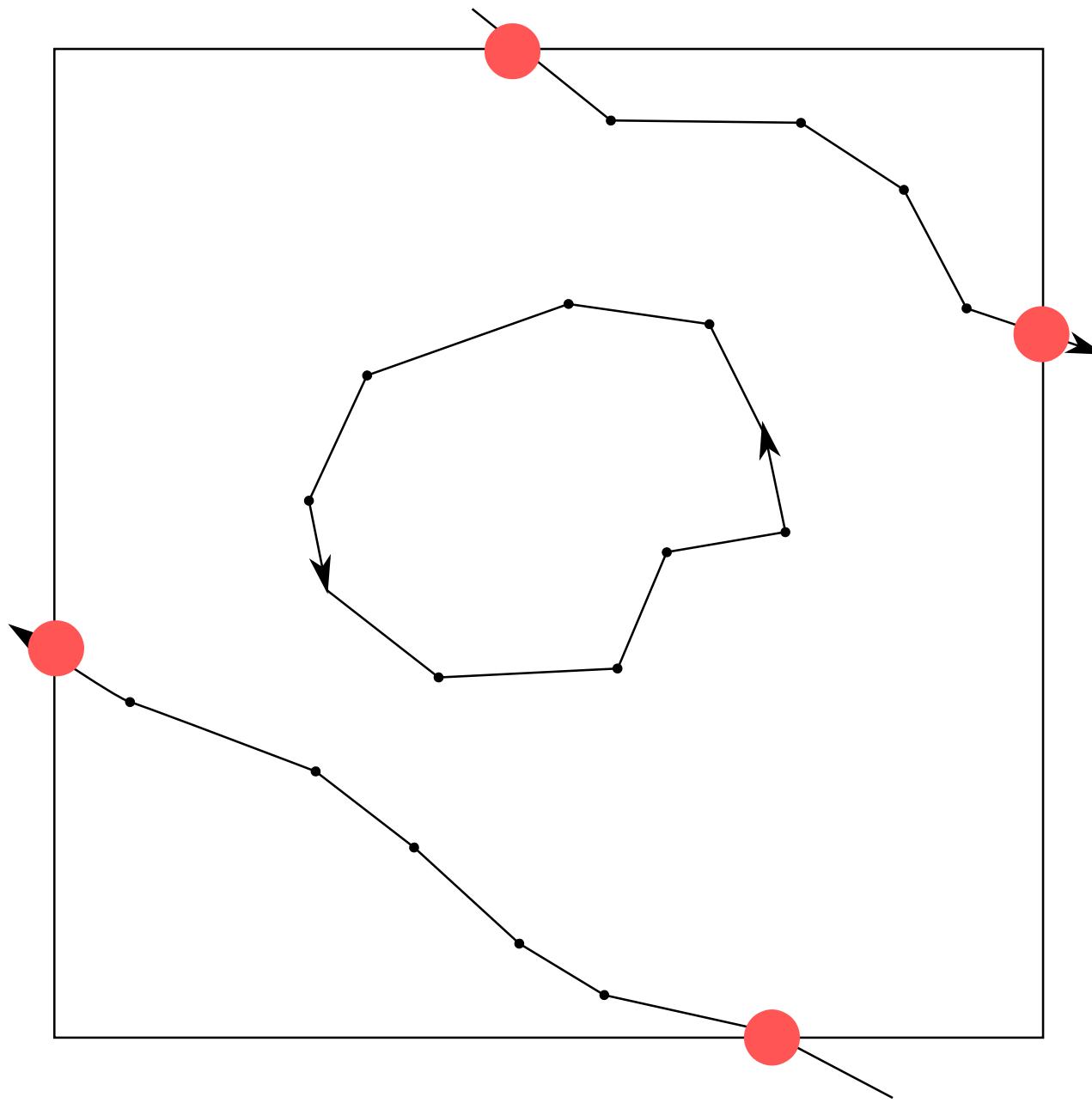
# Küstenlinien: Verarbeitung

- Kein komplettes Küsten-Polygon,  
sondern einzelne Ways
- Herausforderung:  
Im Kartenausschnitt nur (evtl. unverbundene)  
Teil-Ways enthalten  
=> wie rendern?

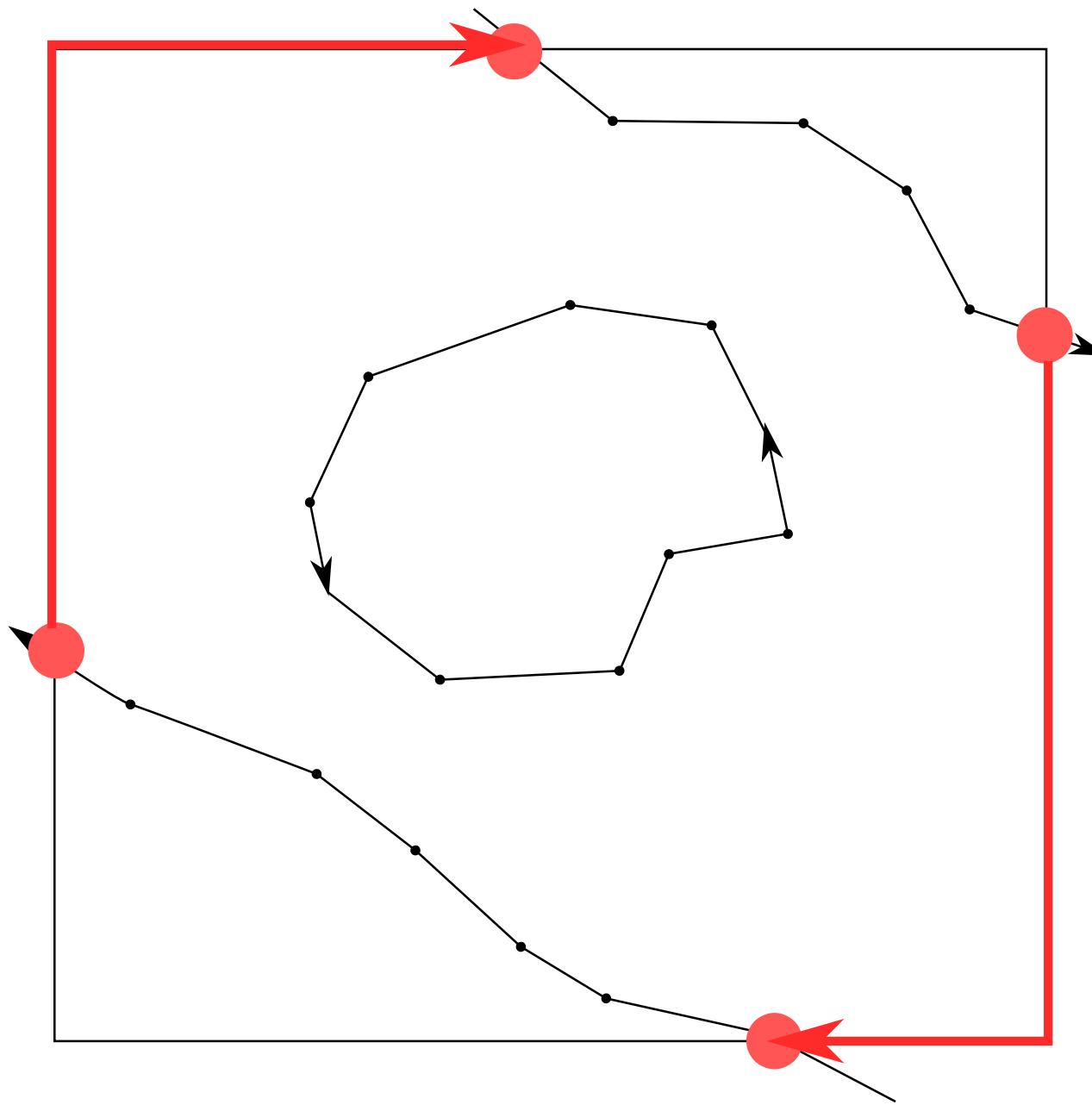
# Küstenlinien: Verarbeitung



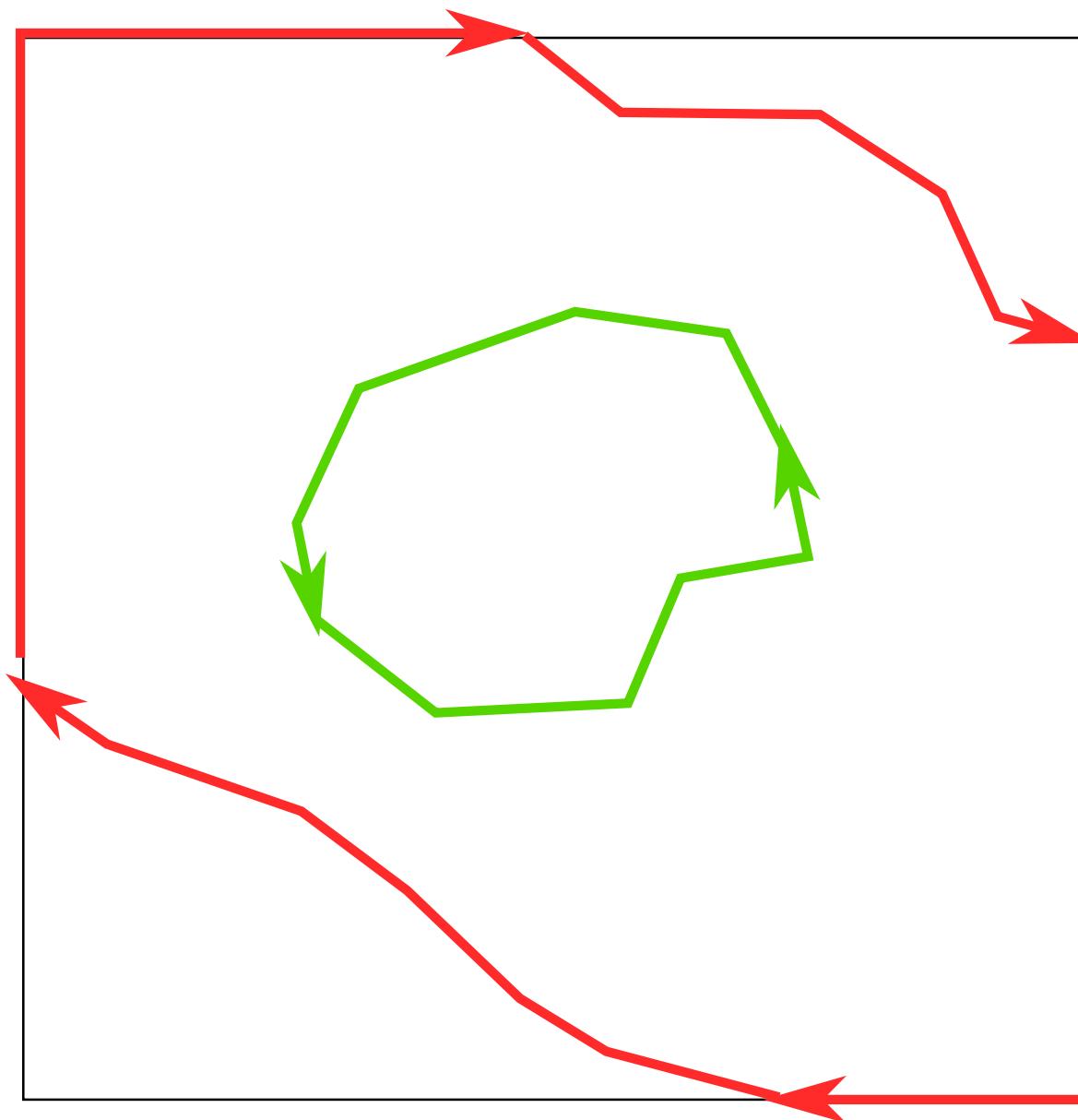
# Küstenlinien: Verarbeitung



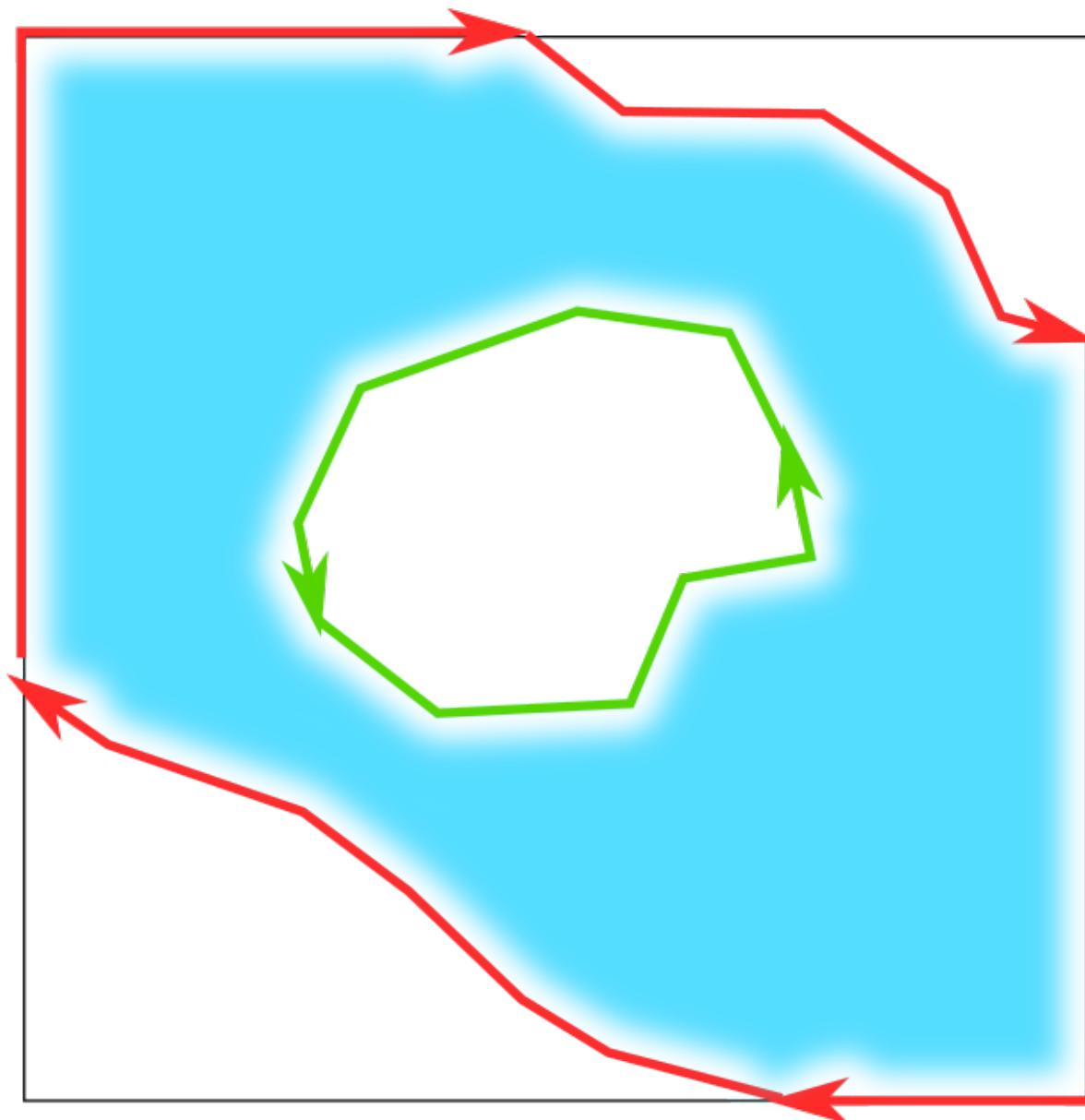
# Küstenlinien: Verarbeitung



# Küstenlinien: Verarbeitung



# Küstenlinien: Verarbeitung



# Küstenlinien: Verarbeitung

- Kein komplettes Küsten-Polygon,  
sondern einzelne Ways
- Herausforderung:  
Im Kartenausschnitt nur (evtl. unverbundene)  
Teil-Ways enthalten ✓
- Herausforderung Teil 2:  
Was tun mit Kacheln *ohne* Küstenlinie?

# Küstenlinien: Kacheln ohne Linie

- Derzeitige Heuristik:
  - Wenn Kacheln bestimmte Objekte enthalten, sind sie Meer (oder Land)
  - Default: Land
- Problem:
  - Es ist fast bei keinem Objekt eindeutig!
  - Beispiel: Flüsse, Bojen, Unterwasserkabel, ...

# Küstenlinien: Kacheln ohne Linie



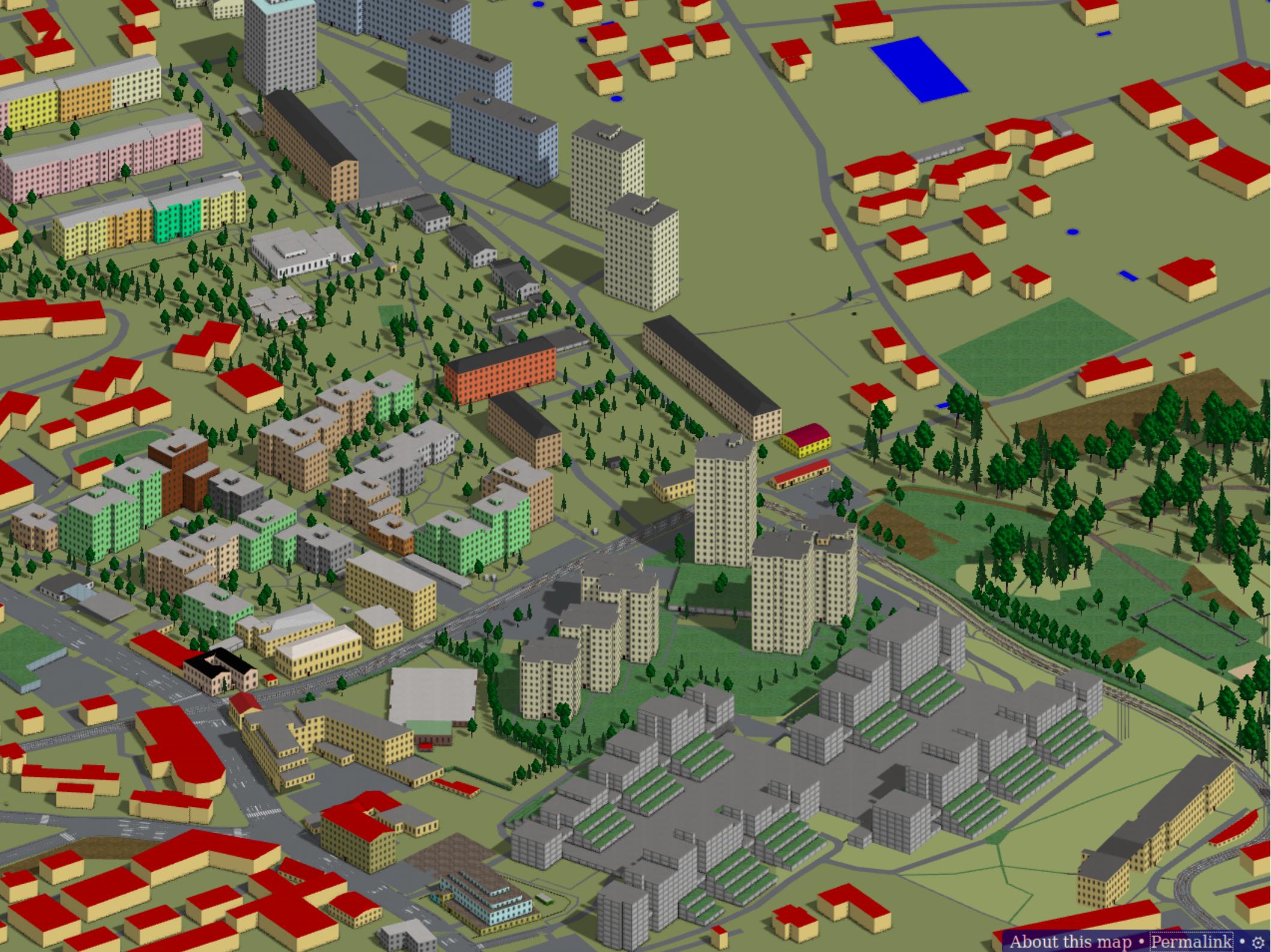
# Küstenlinien: Kacheln ohne Linie

- Geplante Lösung:
  - Küstenlinien-Download (z.B. von openstreetmapdata.com)
  - Rastern und Flag (1 Bit) pro Kachel
- Probleme:
  - Hängt von Konsistenz der globalen Coastlines ab
  - Löst das Problem nur für Onlinekarte

# Küstenlinien: Nebenbemerkung

- Unvollständige Multipolygone
  - Kann vorkommen bei manuellem Download
  - Problem für OSM2World
  - Grund: keine definierte links-rechts-Eigenschaft!
- Gedankenspielerei:
  - API 7: partieller Download?

Onlinekarte  
(<http://maps.osm2world.org/>)



About this map • [Permalink](#) •



# Onlinekarte: Historie

- Eine Inspiration: osm2pov
- Daher: PovRay-Dateien als erstes Zielformat
- Versuch der Erstellung einer Onlinekarte
- PovRay + OSM2World ist rechenzeitintensiv

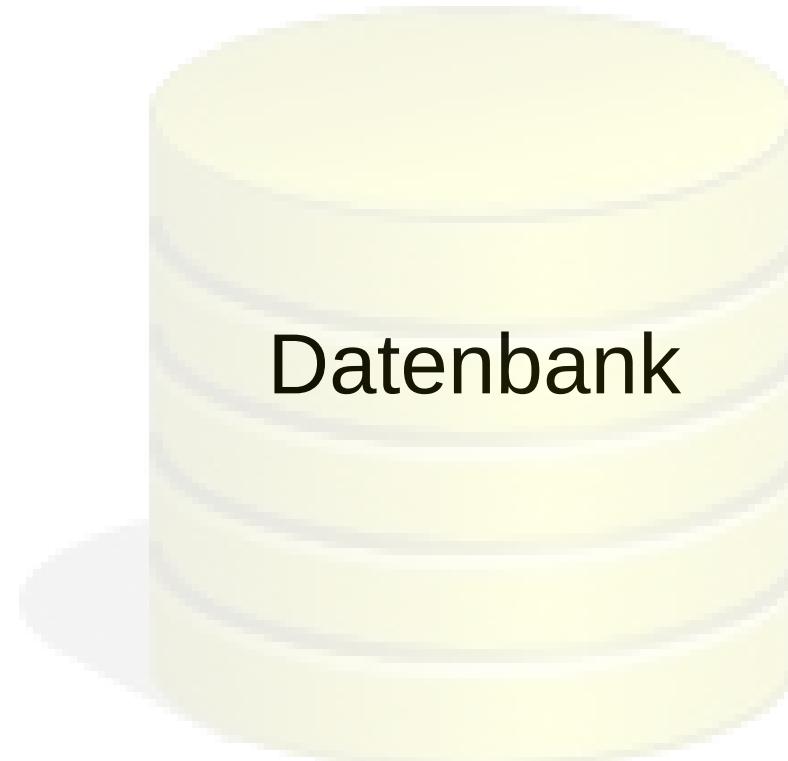
=> Bilder müssen *vorberechnet* sein,  
warten bis zur Anfrage undenkbar

# Onlinekarte: Kachelberechnung

- Noch immer genutztes Verfahren:
  - Download von DACH-Extrakt (Geofabrik)
  - Splitten in Datenkacheln
  - Datenkacheln mit Änderungen in Queue
  - OSM2World arbeitet Queue ab
- Optimierungskriterien bisher:
  - Durchschnittsgeschwindigkeit
  - Viele Kacheln zugleich: z13-18, Himmelsrichtung

# Onlinekarte: Historie

- Probleme:
  - Viele Kacheln werden nie angefragt
  - Festplattenplatz als limitierender Faktor
- Lösungsansatz: Kacheln on Demand
  - OSM2World öffnet Port für Anfragen
  - Einsatz von Tirex
  - DB-Backend



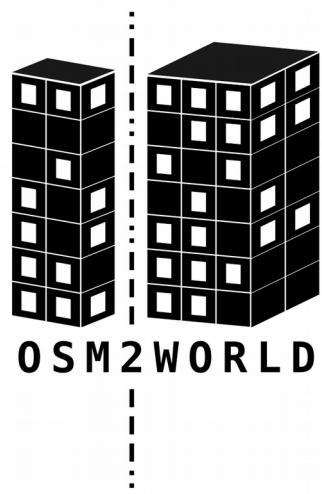
Datenbank

# Datenbank

- Aktuell unterstützte Datenquellen:
  - .osm (inkl. JOSM-Variante)
  - .osm.bz2
  - .osm.pbf
  - Overpass
  - Java-API
  - PostgreSQL? Osm2pgsql?

# Datenbank: osm2pgsql

- Tabellschema ausgelegt auf spezielle Rendering-Anforderungen (2D):
  - Ways liegen als Geometrie vor, nicht als Liste von Nodes
  - Relationen analog
- Workaround für OSM2World:
  - Benutzung des Parameters –slim
  - Bei Anfrage: Join mit zusätzlichen Tabellen



[osm2world.org](http://osm2world.org)