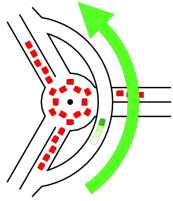


Overpass API

Workshop zur **Overpass API**



Inhalt

Einführung

Das Datenmodell, explizit und implizit

Schwerpunkt: Daten und Abfragen modellieren

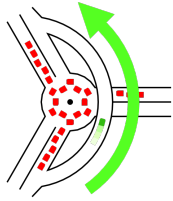
Selektoren Teil 1

Daten exportieren

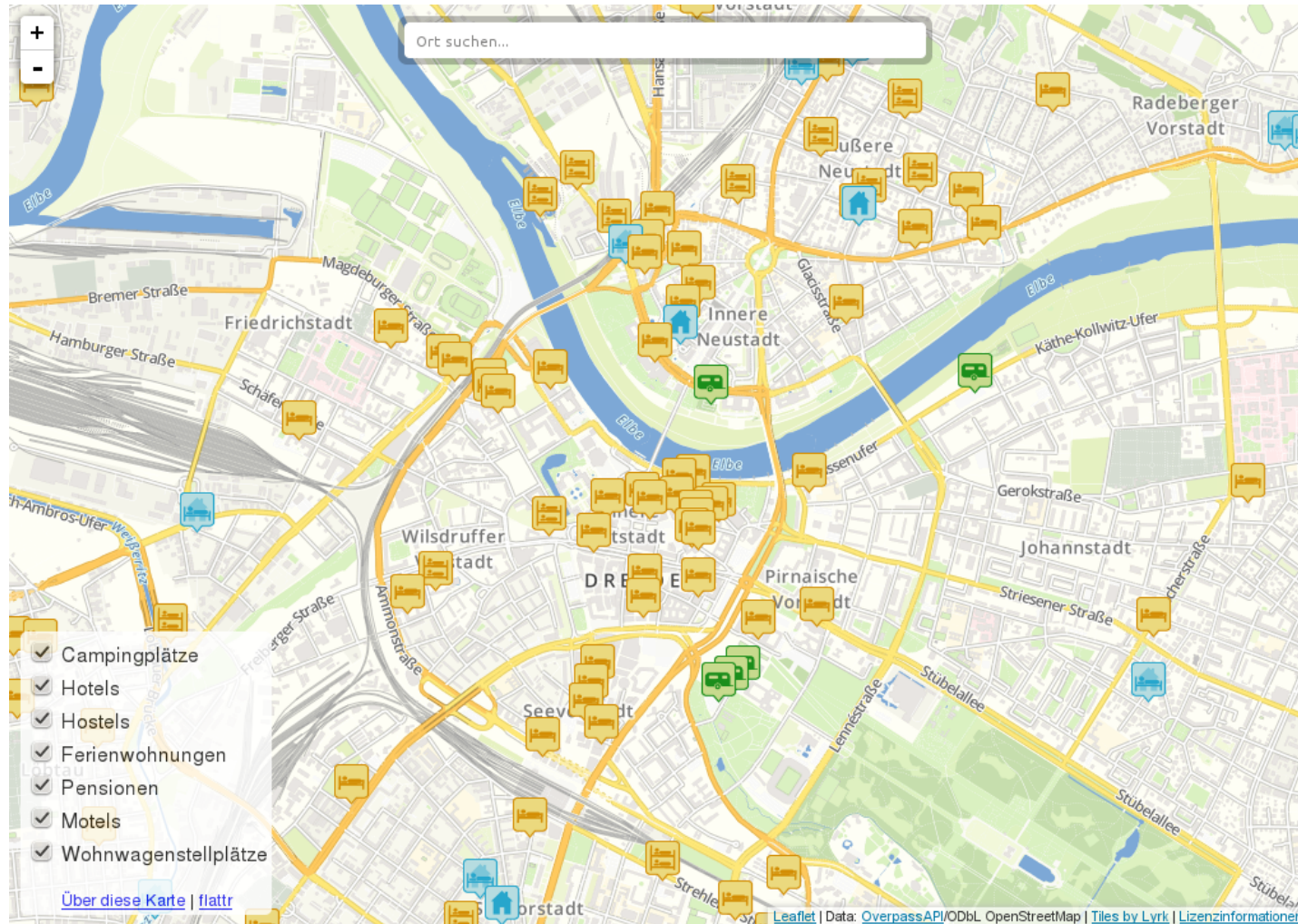
Selektoren Teil 2

Änderungen finden

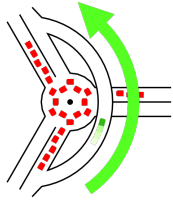
Fragen



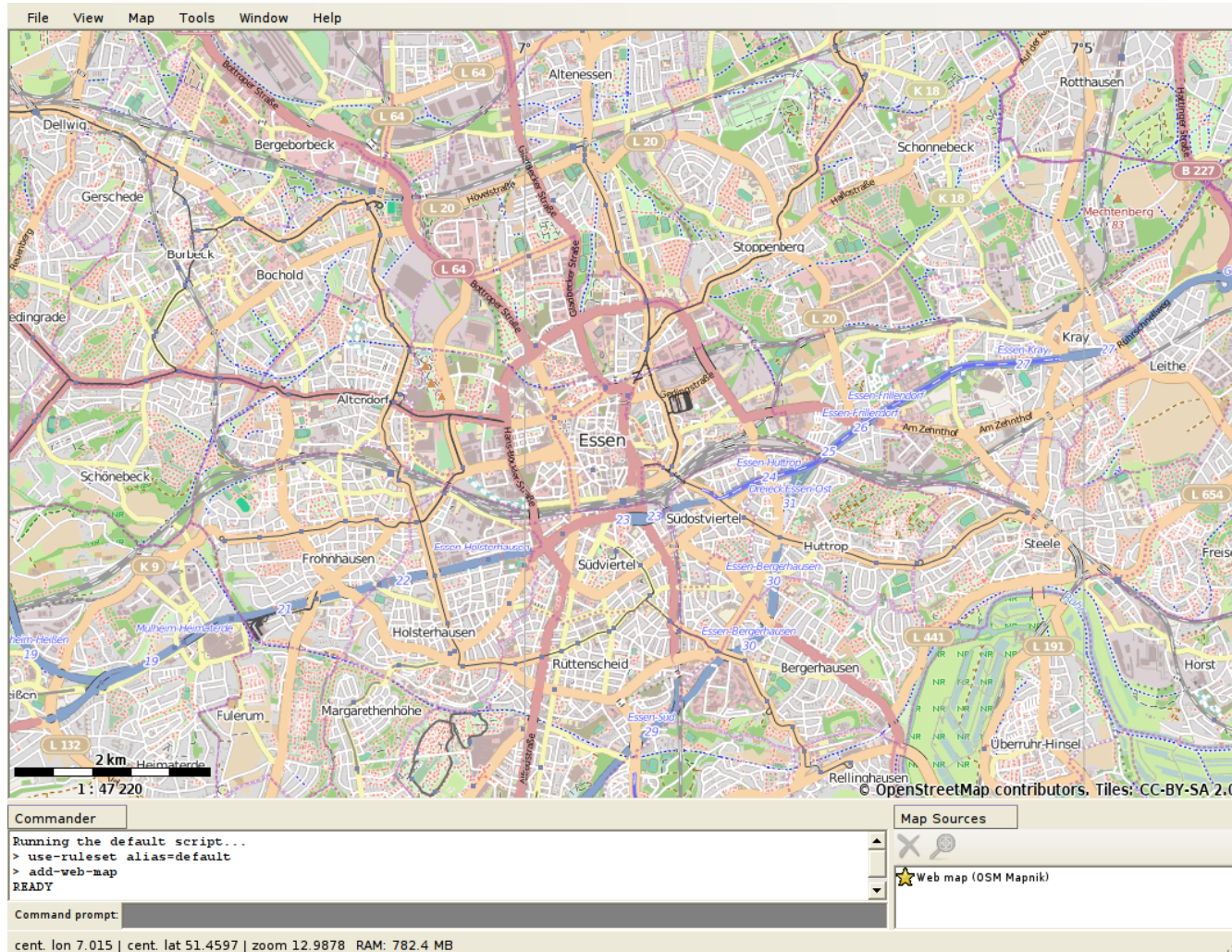
Einführung: Anwendungsbeispiele



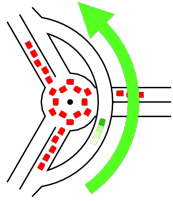
POIs als Overlay: z.B. Unterkunfts-karte



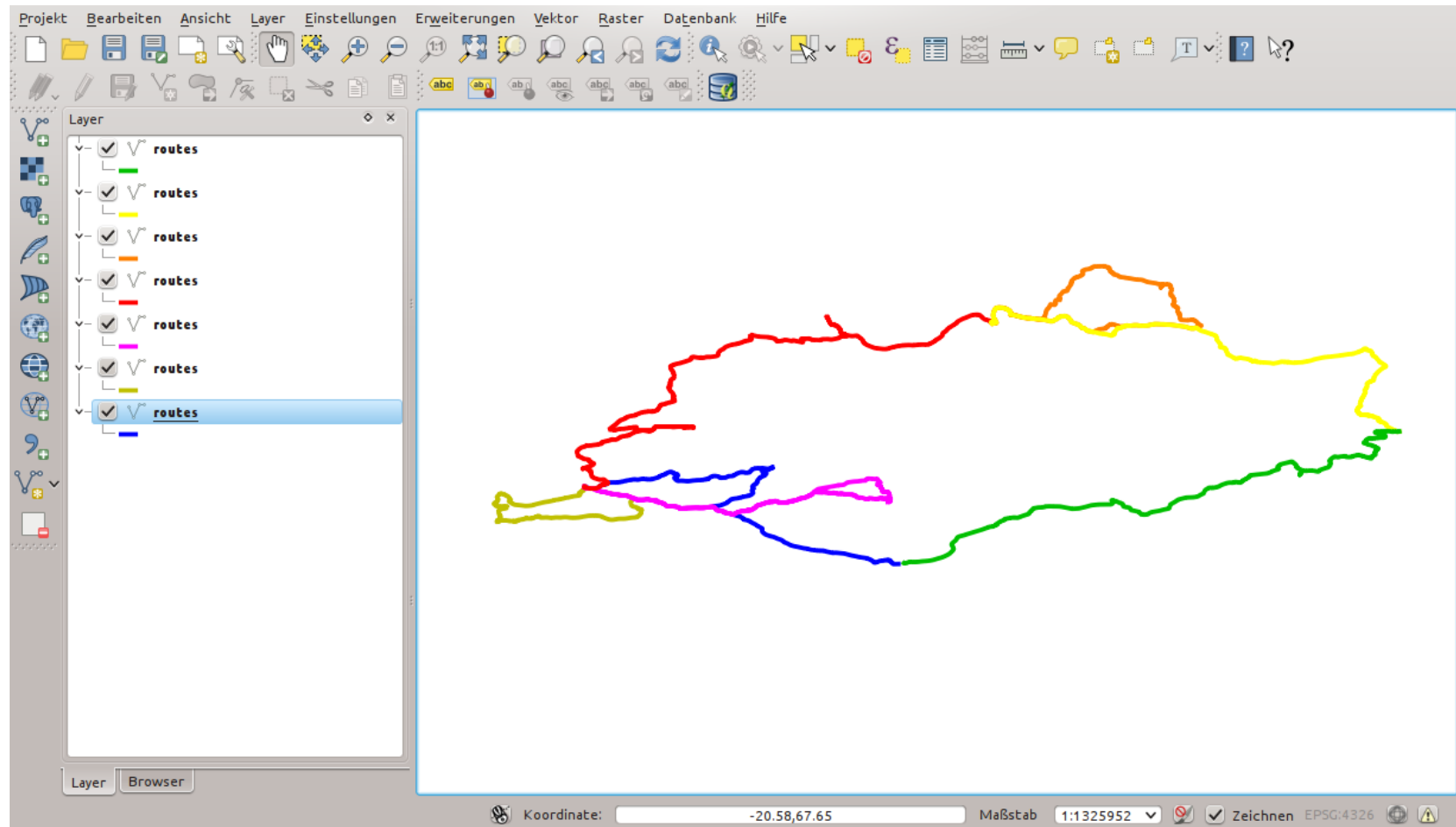
Einführung: Anwendungsbeispiele



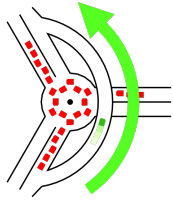
benutzerdefinierte Karten: z.B. Maperitive



Einführung: Anwendungsbeispiele



weiter konvertieren: z.B. mit QGIS in Shapefiles



Einführung: Anwendungsbeispiele

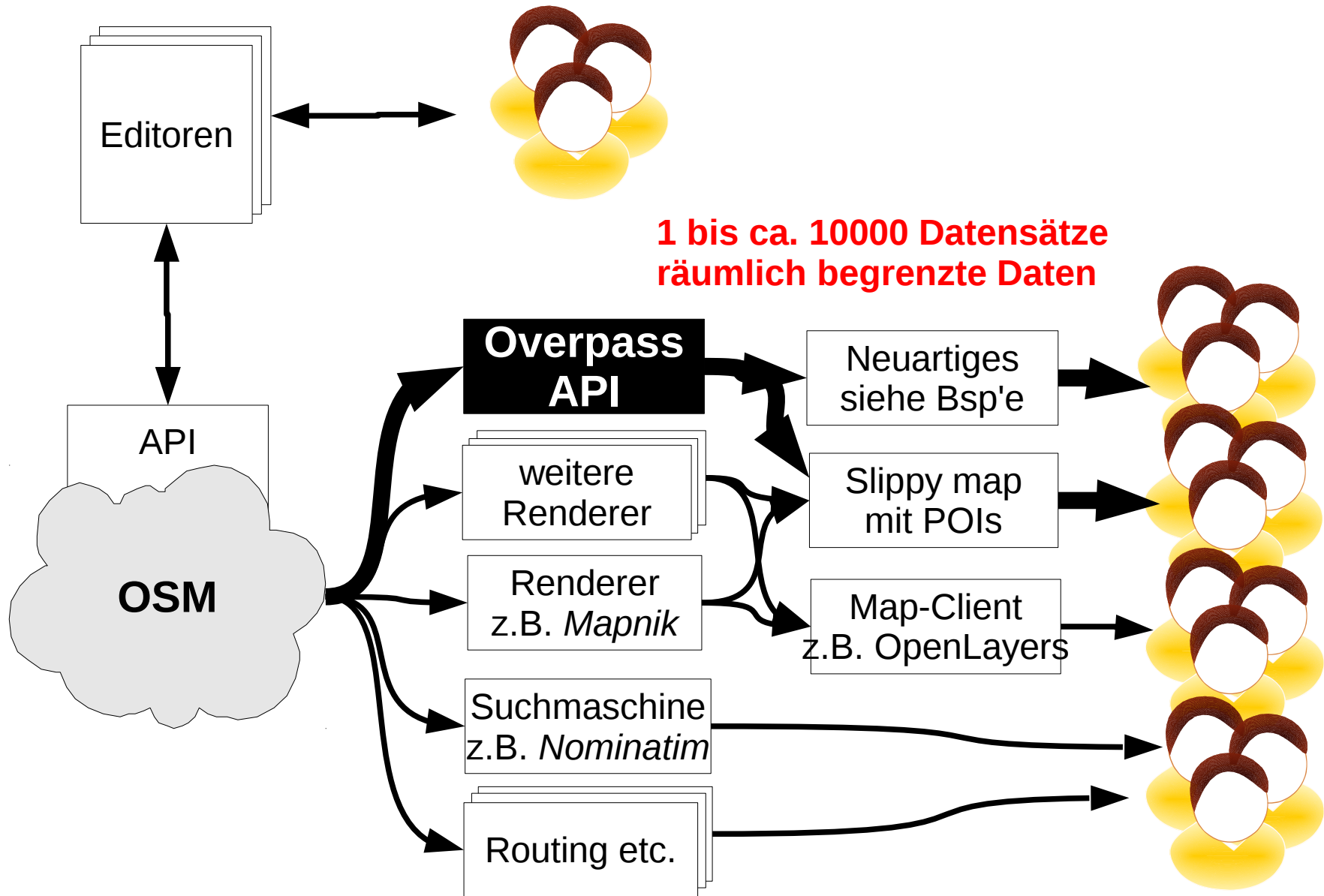
Ausführen Teilen Export Wizard Speichern Laden Einstellungen Hilfe overpass turbo Flattr this! Karte Daten

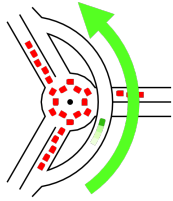
```
1 area[name="Bonn"];
2 way(area)
3 [building!~"."]
4 [boundary!~"."]
5 [barrier!~"."]
6 [amenity!~"."]
7 [leisure!~"."]
8 [landuse!~"."]
9 [railway!~"."]
10 [communication!~"."]
11 [natural!~"."]
12 ["building:part"!~"."]
13 ["roof:edge"!~"."]
14 ["roof:ridge"!~"."]
15 [waterway!~"."]
16 [highway!~"."]
17 [power!~"."]
18 [shop!~"."]
19 [historic!~"."]
20 [embankment!~"."]
21 [man_made!~"."]
22 [aeroway!~"."]
23 [tourism!~"."]
24 [public_transport!~"."]
25 [military!~"."]
26 [route!~"."]
27 ["seamark:type"!~"."]
28 ["addr:interpolation"!~"."];
29 rel(bw)->.r;
30 (._ - way(r.r));
31 (._>);
32 out;
```

geladen - Nods: 1251, Ways: 162, Relations: 0
angezeigt - POIs: 7, Linien: 111, Polygone: 51

Daten interaktiv analysieren: mit Overpass Turbo

Einführung: Das OpenStreetMap-Ökosystem für Datennutzer



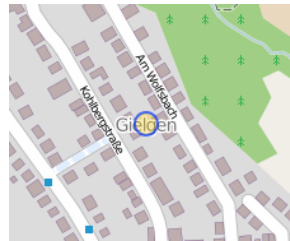


Das Datenmodell

Beispiel

Overpass QL

Node



```
node[...];
```

Way

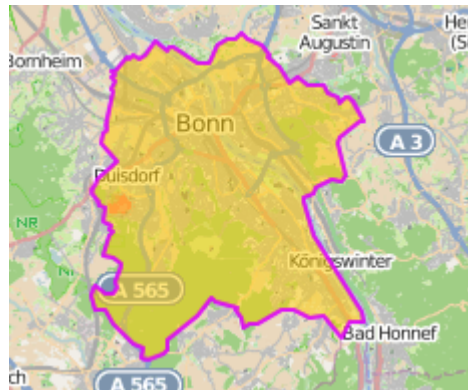
(keine Geometrie -
verweist auf Nodes)



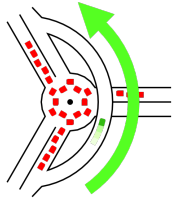
```
way[...];
```

Relation

(keine Geometrie -
verweist auf Ways
und Nodes)

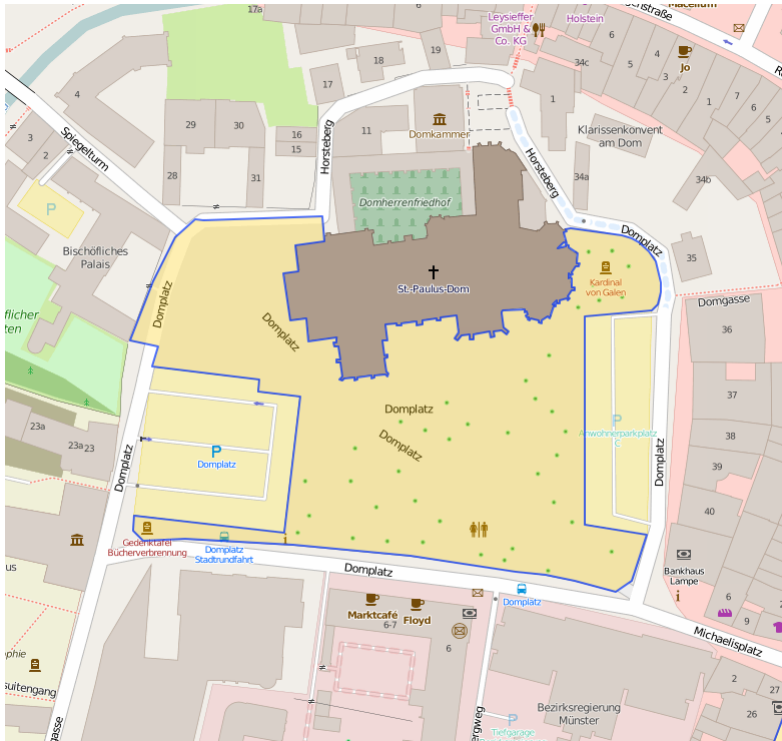


```
relation[...];
```

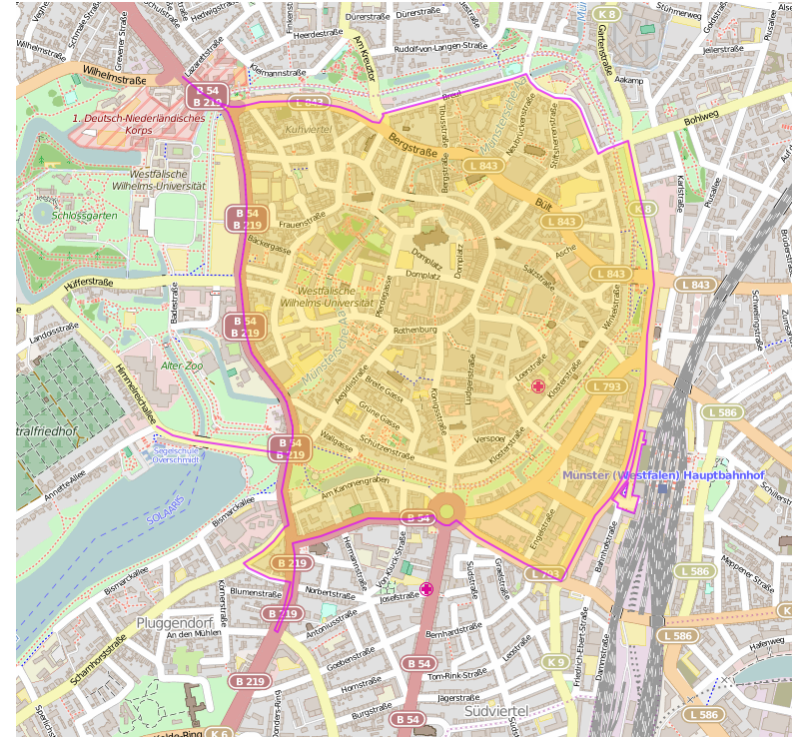



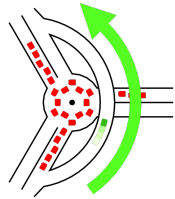
Flächen?

teilweise als
geschlossener **way**
repräsentiert



teilweise als
relation
repräsentiert





Das Datenmodell: Was nicht dazugehört

keine thematische Schichtung nach **Layern**

Warum?

Alltagsthemen überschneiden sich:

- Kaffee und Kuchen gibt's beim Bäcker, beim Café, im Restaurant ...
- Nebenstraßen sind Radwege sind Verkehrsflächen sind Nebenstraßen
- Bargeld gibt's am Geldautomat, im Bankgebäude, ggf. im Supermarkt

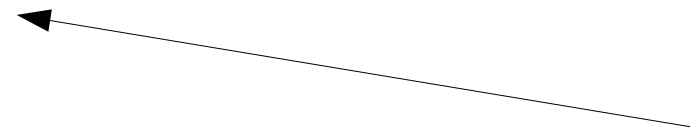
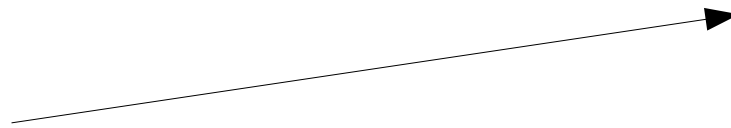
Stattdessen:

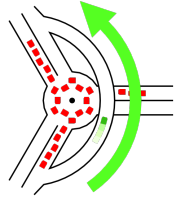
Datenmodell (re-)formulieren
nach fachlichen Kriterien
z.B.: Restaurants, Cafés, Imbiss ja,
Kneipe nein
z.B.: Tagging-Schema für Rollstühle

passende Tags recherchieren
(Wiki, Taginfo, ...)

**Kontakt zur
Community**

passende Daten sichten
(Overpass Turbo)





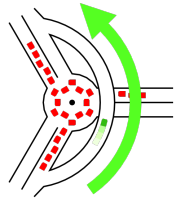
Das Datenmodell: Was nicht dazugehört

„Normalisierung“ wie in SQL,
„Kategorisierung“ wie in Wikipedia-Kategorien
Warum?

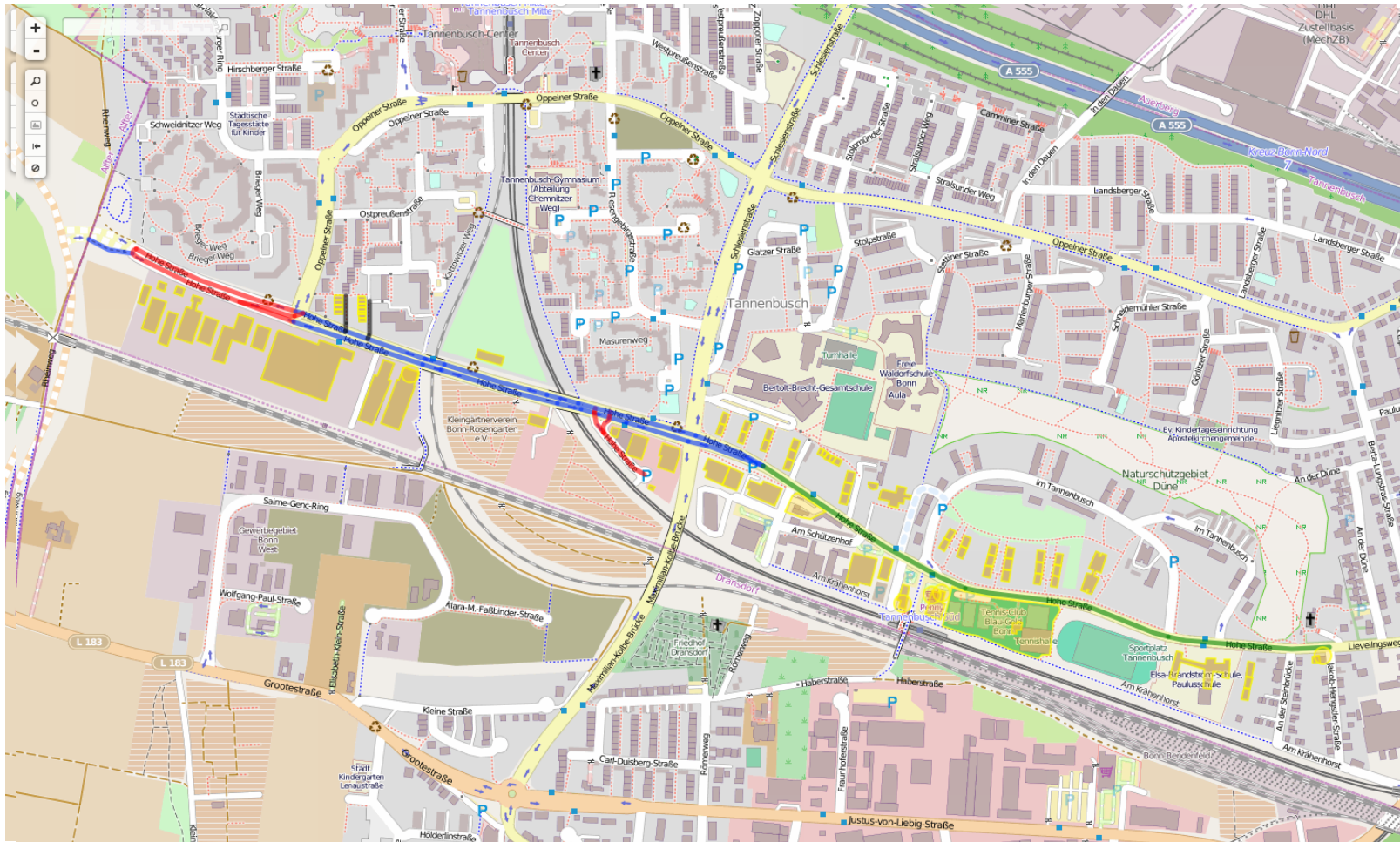
Vielseitigkeit statt Begriffsbildung:

- haben straßenbegleitende Radwege den gleichen Namen wie die Straße?
- gehören zu einem Bahnhof: der Busbahnhof, der Bahnhofsvorplatz, Geschäfte im Bahnhof, am Bahnhof, ...?
- Betriebsteile eines Bahnhofs
- Schulgelände vs. Schule

Stattdessen ...

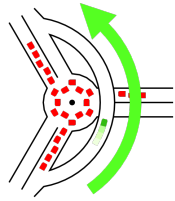


Das Datenmodell: String-Gleichheit

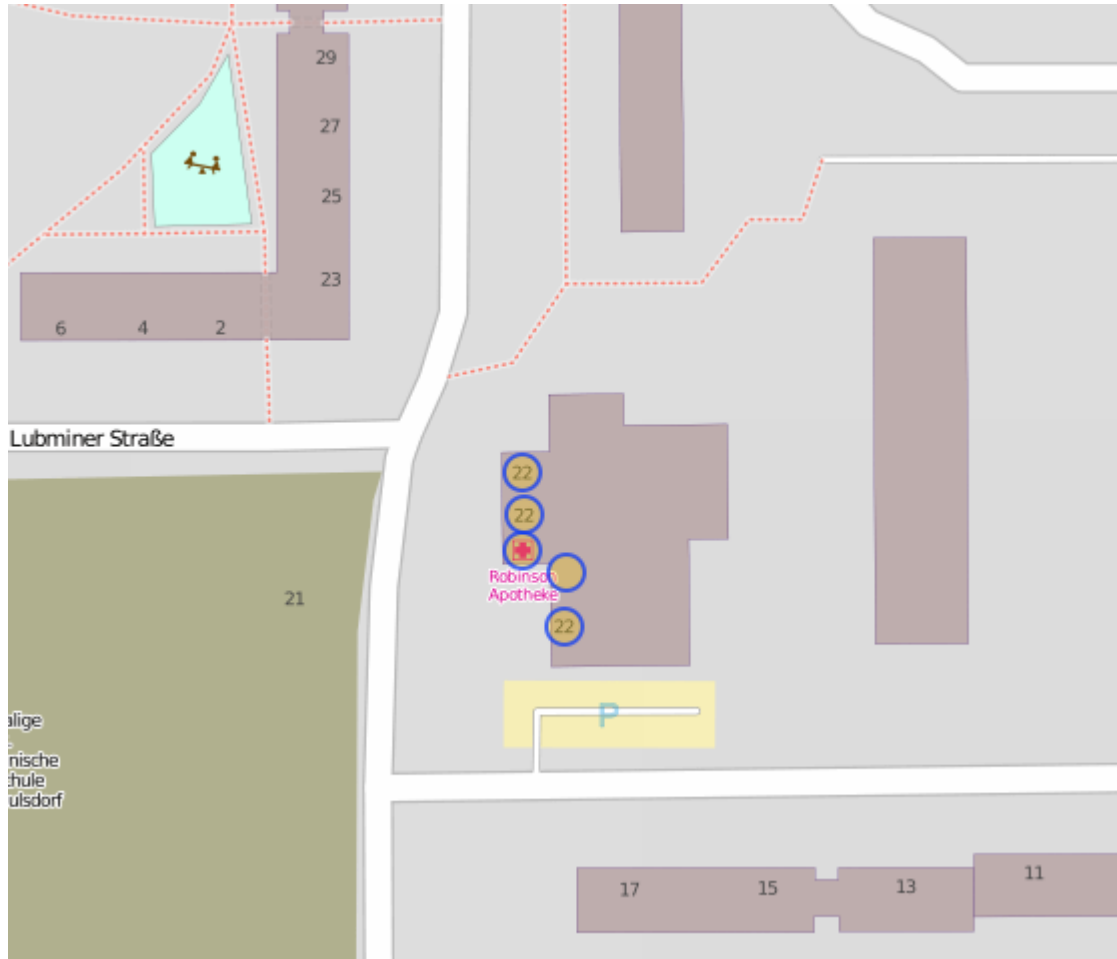


[name="Hohe Straße"]

[„addr:street“="Hohe Straße"]



Das Datenmodell: räumlicher Bezug



liegt in



verbunden mit



schneidet



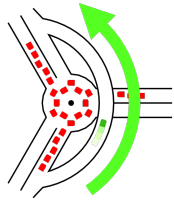
„nahe bei“



„fußläufig zu“

Leicht: Apotheke liegt **in** Berlin

aber: an welcher Straße?



Selektoren, Teil 1

zum Üben: *alle Bäckereien in Dresden*

Vorarbeiten:
Suche nach „osm wiki Bäckerei“,
Öffnen der OSM-Wiki-Seite.

DE:Tag:shop=bakery

Verfügbare Sprachen — Tag:shop=bakery

- čeština • Deutsch • English • español • français • italiano • polski • português • suomi • русский • 日本語 • العربية

Andere Sprachen — Helfen Sie uns, dieses Wiki zu übersetzen

Eine **Bäckerei**. Ein Laden, der Brot und andere Backwaren verkauft. Oft werden auch Kuchen angeboten, auch belegte Brötchen. Eine Bäckerei backt üblicherweise in ihren Geschäftsräumen (Backstube) täglich frisches Brot und Brötchen sowie anderes Kleingebäck.

Mit zunehmender Industrialisierung des Handwerks dürfte diese strenge Definition nicht zu halten sein. Der Durchschnittskunde erkennt oft nicht, ob die in einer Bäckerei verkauften Backwaren tatsächlich in den Geschäftsräumen gebacken oder angeliefert werden. Viele Kunden achten eher auf den Preis. Aber es gibt auch wieder immer mehr Kunden, die auf Qualität und Geschmack achten. Urban entstehen immer mehr Back-Shops diverser Franchise-Ketten. Auf dem Land und Kleinstädtisch finden sich aber noch viele eigenständige Bäckereien mit eigener Backstube. Mit untenstehenden optionalen Tags kann dies unterschieden werden.

Keine Bäckerei in diesem Sinne sind self-service Brotbackautomaten oder Backwarentresen beim Discounter, selbst wenn mehrere Backöfen dort in Nebenräumen in Betrieb sind. Das ist mittlerweile normales Angebot. Eigentlich auch nicht **self-service Back-Shops** diverser Ketten. Bis zu einem neuen Proposal dahingehend werden diese Shops jedoch weiterhin als bakery getaggt, wenn möglich, wie untenstehend spezifiziert.

Contents [hide]
1 Wie mappen?
2 Tags die in diesem
3 Optionale Tags
4 Fotos
5 Rendering
6 Ähnliche Tags

Wie mappen?
Setze einen Punkt oder zeichne den Umriss des Gebäudes. Füge `shop=bakery` und `name=*` hinzu.

shop = bakery

Beschreibung
Eine Bäckerei

Darstellung in openstreetmap-carto

Gruppe: Shops

Für diese Elemente

Sinnvolle Kombinationen

- name=*

Wikidata
Q274399
Status: akzeptiert

taginfo [More...]

	138 427	4.81 %
	19 941	2.43 %
	66	0.88 %

Werkzeuge für diesen Tag

- taginfo • CH • CZ • HU • PL
- overpass-turbo

Alle Objekte mit Tag *shop=bakery*

Öffnen der Website
<https://overpass-turbo.eu>

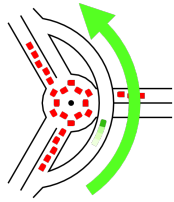
Eingeben:
`area[name="Dresden"];
nwr[shop=bakery](area);
out center;`

„Ausführen“!

Eingeben:
`nwr[shop=bakery]({{bbox}});
out center;`

„Ausführen“!

Loaded — nodes: 313, ways: 10, relations: 0
Deployed — pois: 323, lines: 0, polygons: 0



Selektoren, Teil 1

alle Objekte mit bestimmten Tag:

```
nwr[shop=bakery]({{bbox}});
```

immer erforderlich: Typ des Objekts (**node**, **way**, **relation**, **nwr**)

Tagkombinationen („und“) durch Aneinanderhängen:

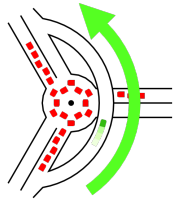
```
node[shop=bakery][wheelchair=yes];
```

auch mit Kartenausschnitt:

```
node[shop=bakery]({{bbox}});
```

(**{{bbox}}**) gibt es nur in Overpass Turbo:

ist eigentlich Länge und Breite: (**51.9, 7.5, 52.0, 7.7**)



Selektoren, Teil 1

alle gerade gefundenen Flächen:

```
nwr[shop=bakery](area);
```

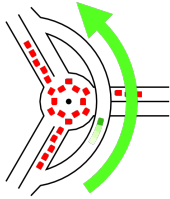
d.h. Ergebnis hängt von vorhergehender Zeile ab:

```
area[name=„Dresden“];
```

```
out center;
```

ähnlich: (around:...), (n), (w), (r), (bw), (br) und weitere

Pseudo-Typ für Flächen: **area**



Daten exportieren

```
nwr[shop=bakery](area);
```

```
out center;
```

steuert, welche Daten ausgegeben werden

Bitte ausprobieren:

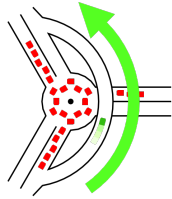
```
way[shop=bakery]({{bbox}});
```

```
out center;
```

... an eine Fundstelle heranzoomen,

```
way[shop=bakery]({{bbox}});
```

```
out geom;
```



Daten exportieren

Overpass Turbo:

```
way[shop=bakery]({{bbox}});  
out geom;
```

JOSM:

```
way[shop=bakery]({{bbox}});  
(. _ ; >);  
out meta;
```

Tabellenkalkulation:

```
[out:csv(;;lat,;;lon,name)];  
way[shop=bakery]({{bbox}});  
out center;
```

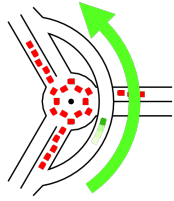
GeoJSON:

```
[out:json];  
way[shop=bakery]({{bbox}});  
convert item ::geom=poly(geom()),::=::;  
out geom;
```

Bitte in

<https://overpass-turbo.de>

ausführen und per Reiter „Daten“ anzeigen!



Selektoren, Teil 2

zum Üben: *alle Konditoren in Dresden*

Vorarbeiten:
Suche nach „osm wiki Konditor“,
Öffnen der OSM-Wiki-Seite.

DE:Tag:shop=pastry

The screenshot shows the OSM Wiki page for the tag 'shop=pastry'. It includes a language selector at the top, a description of the tag, a list of related tags, and a table of tag statistics. A large red arrow points from the left towards the 'taginfo' table.

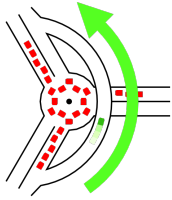
Tag	Anzahl	Anteil
shop=pastry	6 336	0.22 %
shop=bakery	871	0.11 %
shop=confectionery	4	0.05 %

shop=pastry
liefert nur wenige Treffer!

Suche liefert auch
amenity=cafe mit zusätzlich *cuisine=cake*
als Möglichkeit:

```
area[name="Dresden"];  
nwr[amenity=cafe][cuisine=cake](area);  
out center;
```

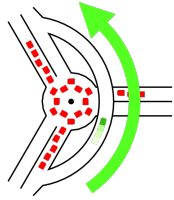
Finden Sie eine dritte Möglichkeit auf der
Wiki-Seite zu *shop=bakery*!



Selektoren, Teil 2

```
area[name="Dresden"]->.a;  
(  
  nwr[shop=pastry](area.a);  
  nwr[amenity=cafe][cuisine=cake](area.a);  
  nwr[shop=bakery][pastry_shop=yes](area.a);  
);  
(._;>);  
out meta;
```

Sind das wirklich alle?



Plausibilitätsprüfung

Nach Suche per Suchmaschine:
Was ist mit „Café Maaß“ in der Oschatzer Straße?

In <https://overpass-turbo.eu> dahin per
`area[name="Dresden"];`
`way[name="Oschatzer Straße"](area);`
`out geom;`
und Klick auf die Lupe

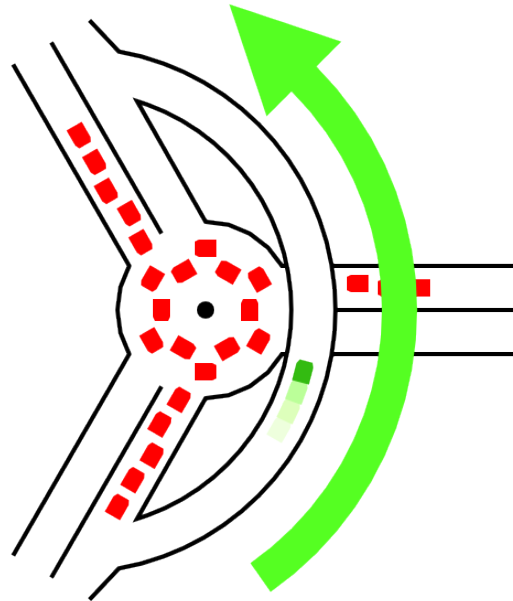
Und jetzt alles dort, was „Maaß“ im Namen hat:
`nwr[name~"Maaß"]({{bbox}});`
`out geom;`
Keine Treffer?!?

Zoomen auf Hausnr 13 und alle Daten holen:
`nwr({{bbox}});`
`out geom;`

Fazit:

**Nicht unterscheidbar
von einer Bäckerei.**

**Nicht immer sind
die Daten gut genug.**



Overpass API

http://wiki.osm.org/wiki/Overpass_API

Danke für Ihre Aufmerksamkeit!