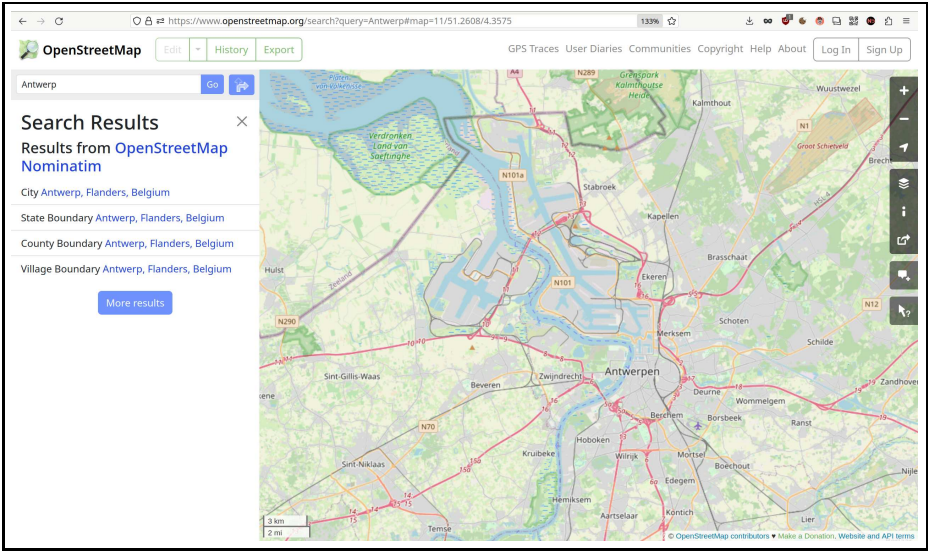
Nominatim on your laptop

Sarah Hoffmann
lonvia@denofr.de

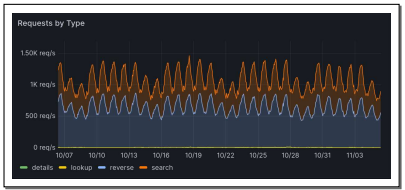
12 Nov 2023

SotM-EU
Antwerp

The Nominatim Geocoder



nominatim.openstreetmap.org: the Big Installation



Hwo About Tiny Nominatim?

(C) Raimond Spekking

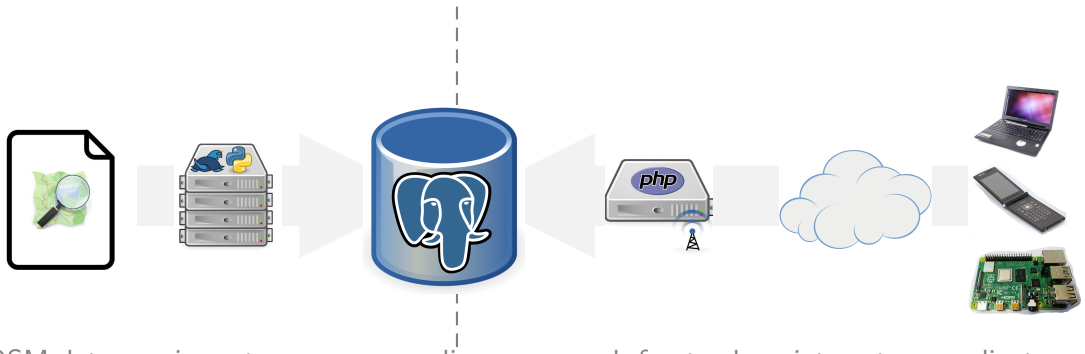
(C) Mytho88

(C) Suyash.dwivedi

with small extracts

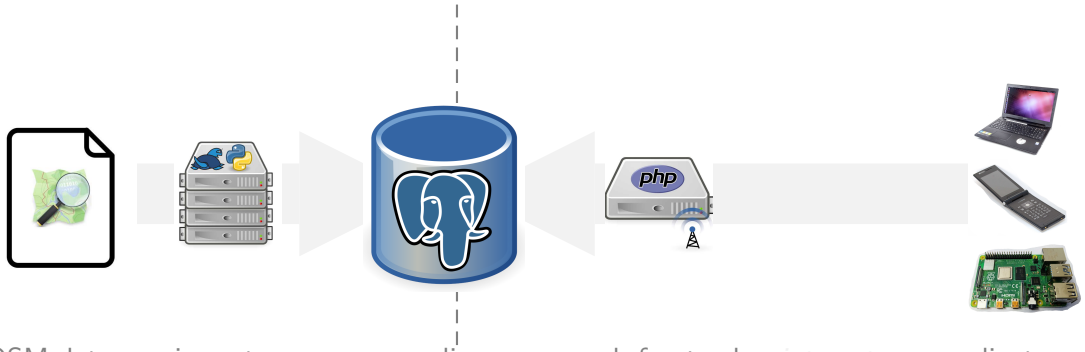
- faster than public API
- better privacy
- customizable

The Geocoding Pipeline



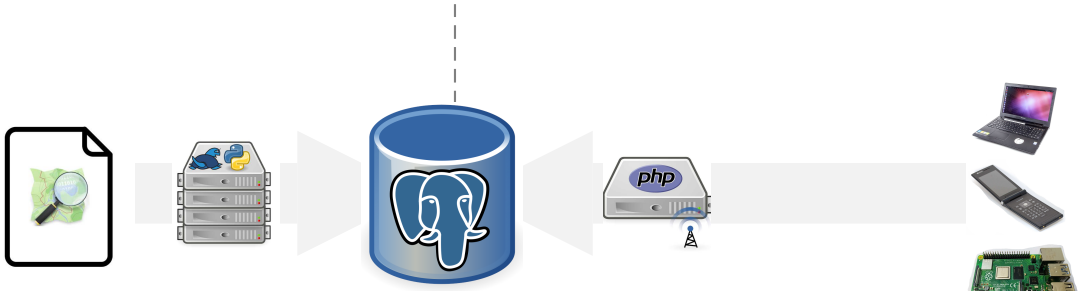
file, server, database images (C) RRZEicons

The Geocoding Pipeline



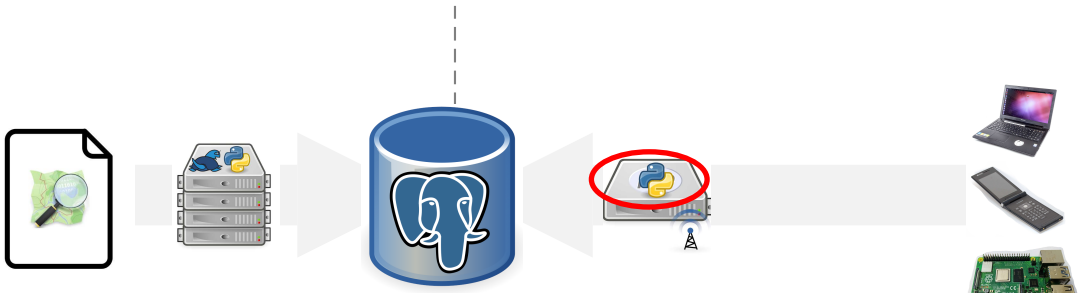
OSM data | importer | geocoding database | search frontend (webserver) | *internet* | client

The Geocoding Pipeline



OSM data | importer | geocoding database | search frontend (webserver + library) | *internet* | client

The Geocoding Pipeline



OSM data | importer | geocoding database | search frontend (webserver + library) | *internet* | client

The Geocoding Pipeline

OSM data | importer | geocoding database | search frontend (webserver + library) | *internet* | client

From PHP to Python

Why Python?

- ✓ favoured by users of Nominatim (together with Java)
- ✓ rich library support for language processing (and SQLAlchemy)
- ✓ easy to create user-extendible code
- ✗ language lock-in
- ✗ mobile support
- ✗ performance

SQLAlchemy

```php
<?php

$sSQL = 'SELECT place_id FROM';
$sSQL .= ' (SELECT place_id, rank_search,';
$sSQL .= '          ST_distance('.$sPointSQL;
$sSQL .= '                      ,geometry) as dist';
$sSQL .= ' FROM placex';
$sSQL .= ' WHERE osm_type = \'N\'';
$sSQL .= ' AND country_code = \''.$sCountryCode.'\'';
// needed to select right index
$sSQL .= ' AND rank_address between 4 and 25';
$sSQL .= ' AND rank_search between 5 and ';
$sSQL .=                    min(25, $iMaxRank);
$sSQL .= ' AND type != \'postcode\'';
$sSQL .= ' AND name IS NOT NULL ';
$sSQL .= ' AND ST_Buffer(geometry, rpd(rank_search))';
$sSQL .= '       && '.$sPointSQL;
$sSQL .= ') as a ';
$sSQL .= 'WHERE dist <= rpd(rank_search)';
$sSQL .= ' ORDER BY rank_search DESC, dist ASC';
$sSQL .= ' LIMIT 1';
```

```python
import sqlalchemy as sa

t = tables.placex
inner = sa.select(t.c.place_id, t.c.rank_search,
                  t.c.geometry.ST_Distance(pt)
                     .label('dist'))\
  .where(t.c.osm_type == 'N')\
  .where(t.c.country_code == cc)\
  .where(t.c.rank_address.between(4, 25))\
  .where(t.c.rank_search.between(5, min(maxRank, 25)))\
  .where(t.c.type != 'postcode')\
  .where(t.c.name != None)\
  .where(t.c.geometry.ST_Buffer(
                        sa.func.rpd(t.c.rank_search)
                     .intersects(pt)))
  .alias('a')

sql = sa.select(inner.c.place_id)\
  .where(inner.c.dist<=sa.func.rpd(inner.c.rank_search))
  .order_by(inner.c.rank_search.desc())
  .order_by(inner.c.dist)
  .limit(1)
```

Supports database adaption for PostgreSQL, SQLite and others.

Performance: PHP vs. Python

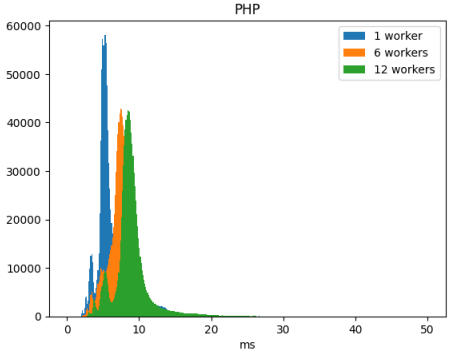
API call / request duration(*)	PHP	Python(1st try)	Python(master)
reverse geocoding	6.3 ms	13.8 ms	8.9 ms
forward search	28.1 ms	37.0 ms	32.7 ms

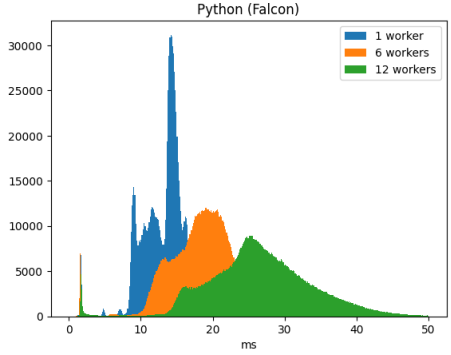
Unequal conditions:

- SQL string concatenation (PHP) vs. SQL library (Python)
- search algorithm not the same

(*) average over 10.000 real-world requests as seen on osm.org

Performance: Multitasking





The Nominatim Python Library

since 4.3 release (experimental)

A simple example: Tagging your photo with locations

```
from pathlib import Path
import exifread
from imgtag import ImgTag
import nominatim.api as napi

with open(FILENAME, "rb") as fh:
    y, x = exifread.utils.get_gps_coords(
                exifread.process_file(fh, details=False))

api = napi.NominatimAPI(Path('.'))
result = api.reverse((x, y), zoom=25, address_details=True)

address_parts = result.address_rows.localize(napi.Locales(['en']))

tags = ImgTag(filename=FILENAME)
tags.add_tags(address_parts)
tags.close()
```

Different Kinds of Search

- free-text search

```
search('BluePoint, Antwerp'))
```

- structured search

```
search\_address(city='Berchem', street='Filip Williotstraat 9', postcode='2600')
```

- specific type of place near a named place

```
search\_category([('amenity', 'event\_venue')], near\_query='Antwerp')
```

- specific type of place near a location

```
search\_category([('amenity', 'event\_venue')], viewbox=(51.1, 4.4, 51.2, 4.5))
```

And all other API functions

- point to address

```
reverse((x, y))
```

- OSM object to address

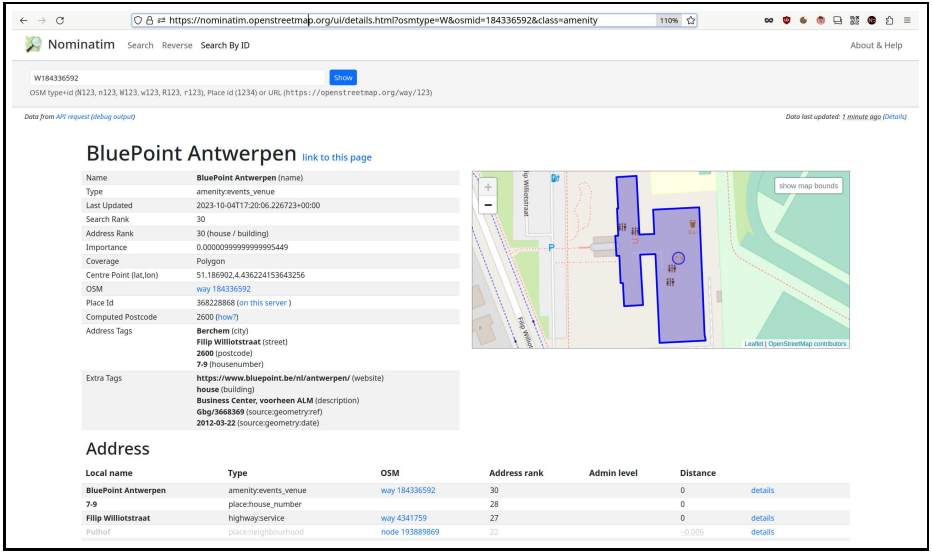
```
lookup([OsmID('N', 3452), OsmID('W', 555)])
```

- Dump full information of place

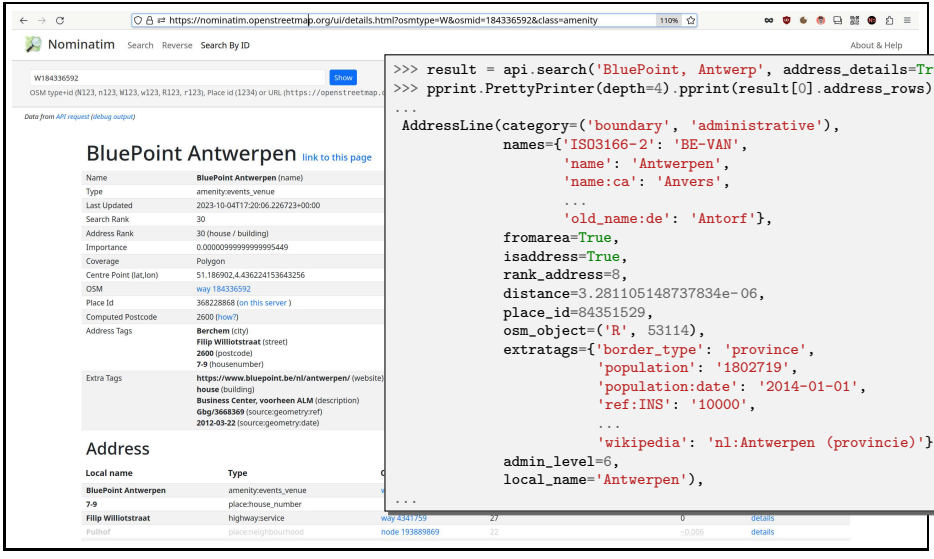
```
details(PlaceID(2345756))
```

... with many filters and constraints

Rich Result Information



Rich Result Information




```
>>> result = api.search('BluePoint, Antwerp', address_details=True)
>>> pprint.PrettyPrinter(depth=4).pprint(result[0].address_rows)
...
 AddressLine(category=('boundary', 'administrative'),
             names={'ISO3166-2': 'BE-VAN',
                    'name': 'Antwerpen',
                    'name:ca': 'Anvers',
                    ...
                    'old_name:de': 'Antorf'},
             fromarea=True,
             isaddress=True,
             rank_address=8,
             distance=3.281105148737834e-06,
             place_id=84351529,
             osm_object=('R', 53114),
             extratags={'border_type': 'province',
                        'population': '1802719',
                        'population:date': '2014-01-01',
                        'ref:INS': '10000',
                        ...
                        'wikipedia': 'nl:Antwerpen (provincie)'},
             admin_level=6,
             local_name='Antwerpen'),
...
```

Nominatim inside SQLite

PostgreSQL → SQLite

```
nominatim convert -o admin.sqlite
```

Extract / Database size (*)	PostgreSQL	SQLite
Germany	19 GB	18 GB
Admin	12 GB	12 GB

(*) non-updateable database, reverse-geocoding only

Using the SQLite database

```
me@home> echo 'NOMINATIM_DATABASE_DSN=sqlite:dbname=admin.sqlite' >> .env
me@home> python3 tag_photo_place.py
```

or

```
me@home> export NOMINATIM_DATABASE_DSN=sqlite:dbname=admin.sqlite
me@home> python3 tag_photo_place.py
```

or

```
...
api = nominatim.api.NominatimAPI(
          Path('.'),
          environ={'NOMINATIM_DATABASE_DSN': 'sqlite:dbname=admin.sqlite'})
...
```

Performance

Extract / Requests per sec (*)	PostgreSQL	SQLite
Germany	454	23
Admin	149	21

(*) Random reverse coordinates near Germany/around the world.

What's next?

- SQLite forward search
- reducing database size
- improving indexes for faster queries
- GeoPy support?

Thank you.

Nominatim Website: `https://nominatim.org`

`lonvia@denofr.de`

Thanks to our supporters

NGI ZERO ENTRUST

OpenCage

Graphhopper

komoot