

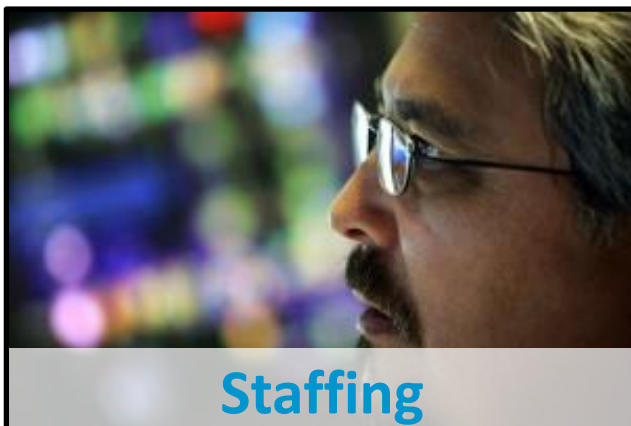
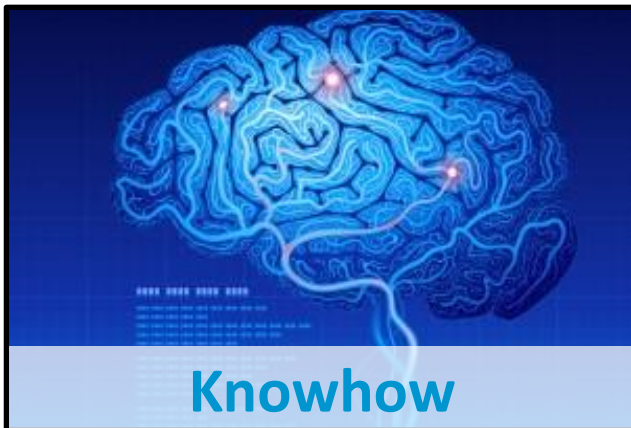
bePelias: un géocodeur local basé sur BeStAddress

Vandy Berten – Smals Research – 25-09-2025





SUPPORT FOR E-GOVERNMENT



WWW.SMALS.BE

Geocoding: concept

Geocoding = (1) Address standardization + (2) Geolocation

Geocoder = API (Application Programming Interface)

“Av. Fonsny 20, 1060 Bruxelles”



```
{“street” : “Avenue Fonsny”,  
  “housesbr” : “20”,  
  “zipcode” : “1060”,  
  “city” : “Saint-Gilles”,  
  “country” : “Belgique”,  
  “location”: [50.8358,4.3361]}
```

(1) Address standardization:

- Data quality:
 - Standardization
 - Deduplication
 - Matching
- Auto-completion (type-ahead)
- ...

(2) Geolocation:

- Visualization
- Controls
- Spatial (GIS) analytics
- ...

Or polygon

Geocoding: concept

“Av. Fonsny 20, 1060 Bruxelles”

Address standardization = Mapping
Geolocation = « side effect »/bonus

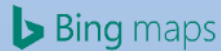
Street	Number	Zipcode	Locality	Location
...	...	...	...	...
Avenue Fonsny	20	1060	Saint-Gilles	50.8358, 4.3361
Avenue Fonsny	22	1060	Saint-Gilles	50.8359, 4.3358
...	...	...	...	...

```
{“street” : “Avenue Fonsny”,  
  “housesnbr” : “20”,  
  “zipcode” : “1060”,  
  “city” : “Saint-Gilles”,  
  “country” : “Belgique”,  
  “location”: [50.8358,4.3361]}
```

Tools

- Cloud commercial providers:

- Here we go
- Bing maps
- Google maps
- TomTom
- Sometimes free in specific conditions



- Mainly 2 open-source/open-data solutions:

- OpenStreetMap/Nominatim
- Photon (OSM + ElasticSearch)
- On-premise possible for a specific region



BeSt Addresses

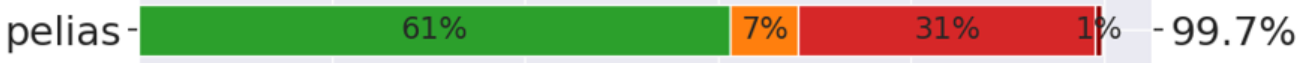
- BeSt Addresses: **BOSA** project providing an **authentic source** for **Belgian addresses**
- Gathering **regional authentic sources**: UrbiS, Adressenregister (ex CRAB), ICAR
- Each address has a **(stable) id** which should be used in other applications
- Can be used either through file **download** (public) or via **API** (public/eGov)

Our constraints

- We needed a geocoding API:
 - Local/on-premise
 - Based on BeSt Address (SPF BOSA)
 - Not written from scratch
 - Accepting large volumes
- Three options so far:
 - PhacochR (Obs. de la santé et du Social BXL + ULB/IGEAT) → Not an API (a batch R library)
 - BOSA API → Until recently, no fuzziness at all
 - Pelias + BeSt → Outdated/poor data, low robustness



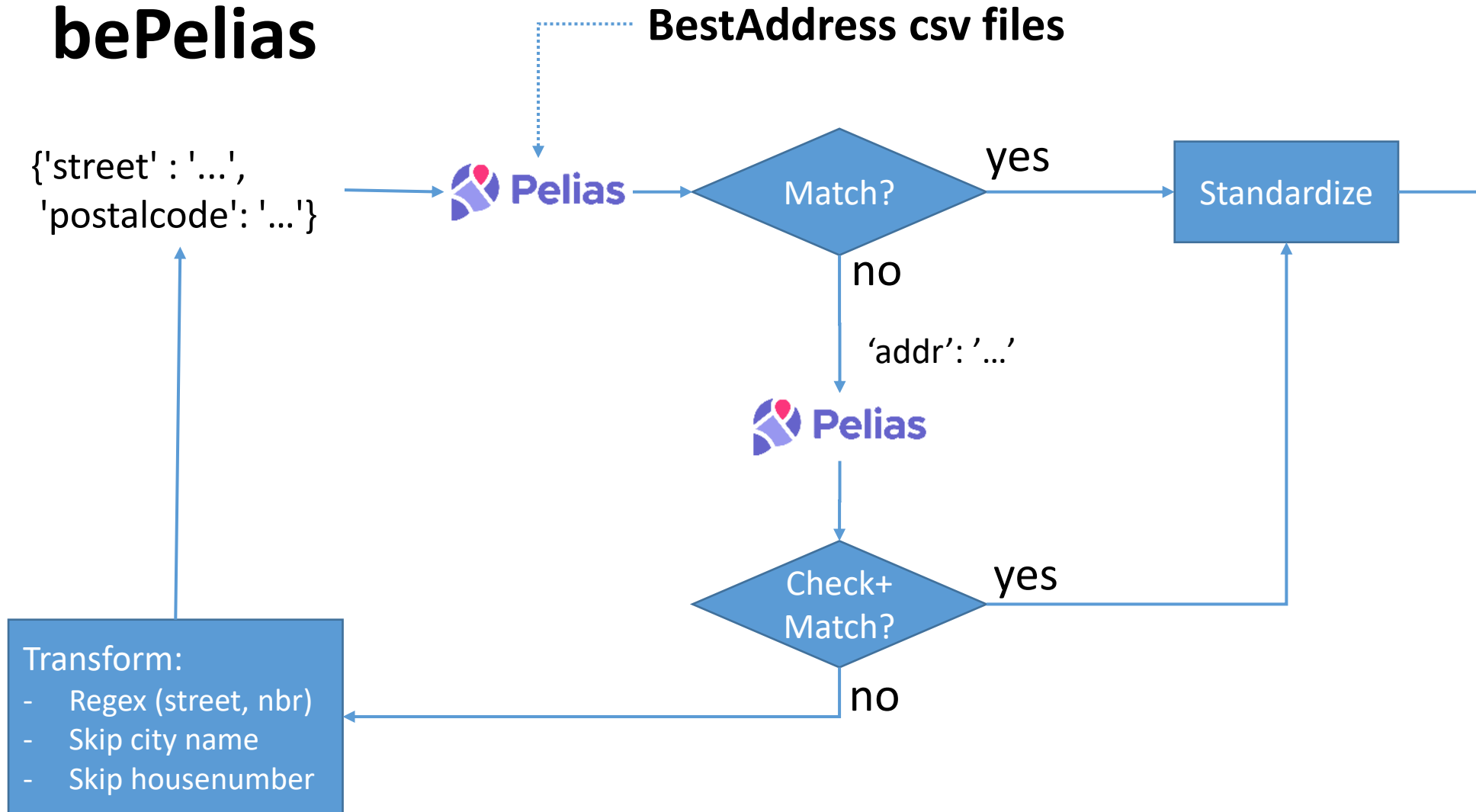
Pelias

- Open source: <https://github.com/pelias/docker> (projects > belgium)
- **Data-source** problems (in 2023):
 - Based on OpenAddresses.io version of BestAddress CSV files (<https://batch.openaddresses.io/>), but due to some incompatibility, **not updated since Feb 2021**
 - Pelias still uses an **old OpenAddresses dataflow**, deprecated mid-2021
- **Poor data**: partial best Id, no translation, no box numbers...
- **Robustness** problem: 

Category	Percentage
Green	61%
Orange	7%
Red	31%
Grey	1%
Total	99.7%

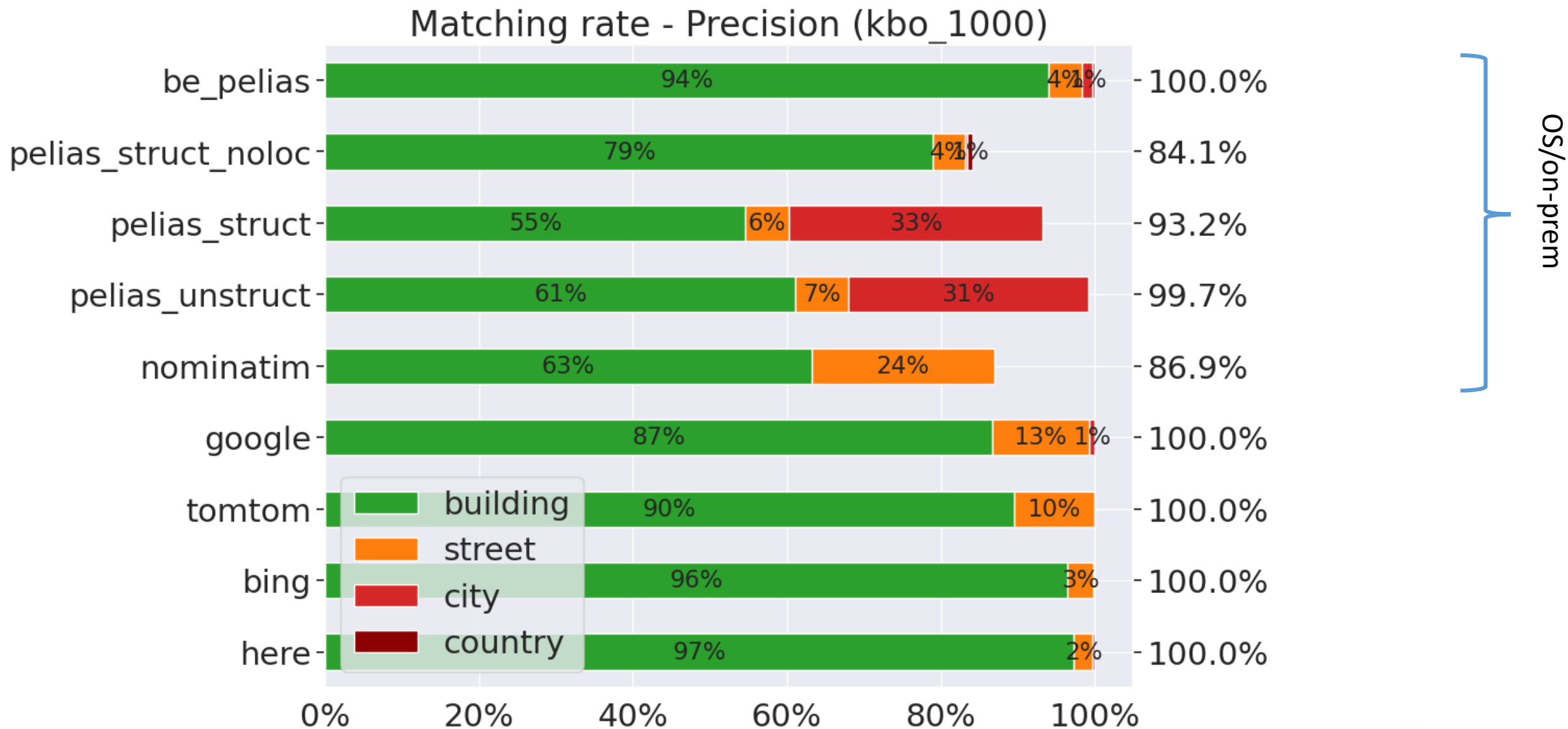
 - If **structured version** doesn't work, sometimes **unstructured version** does (and vice-versa)
 - If none works, some **simple transformations** often allow matching:
 - {"postcode": 1160, "locality": "Auderghem"} → {"postcode": 1160}
 - Cleansing text, removing parenthesis in street, non-digits in housenumber...

bePelias



- Based on a **new dataflow** from BestAddress, by-passing OpenAddresses.io
- “Match?”= “is a building”. If no transformer gives a building match → keep the best one
- If coords = 0,0 → call interpolation engine

bePelias: Matching rate

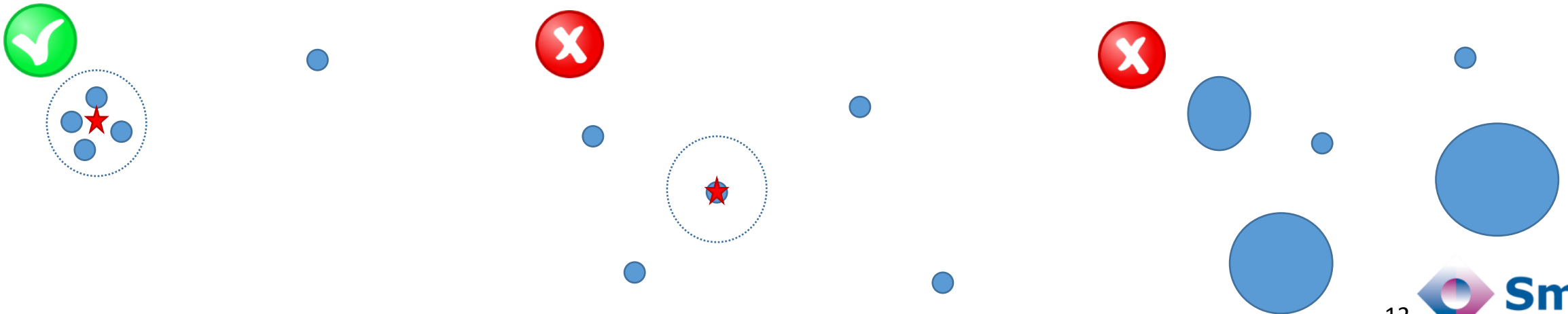


Performances

- Giving an answer does not make this answer to be **correct!**
 - Ideal situation: testing against a **reference address list**. But:
 - Building such a list is **complex!**
 - List will be (intrinsically) **biased**
- We propose a **methodology** based on a “**majority vote**”

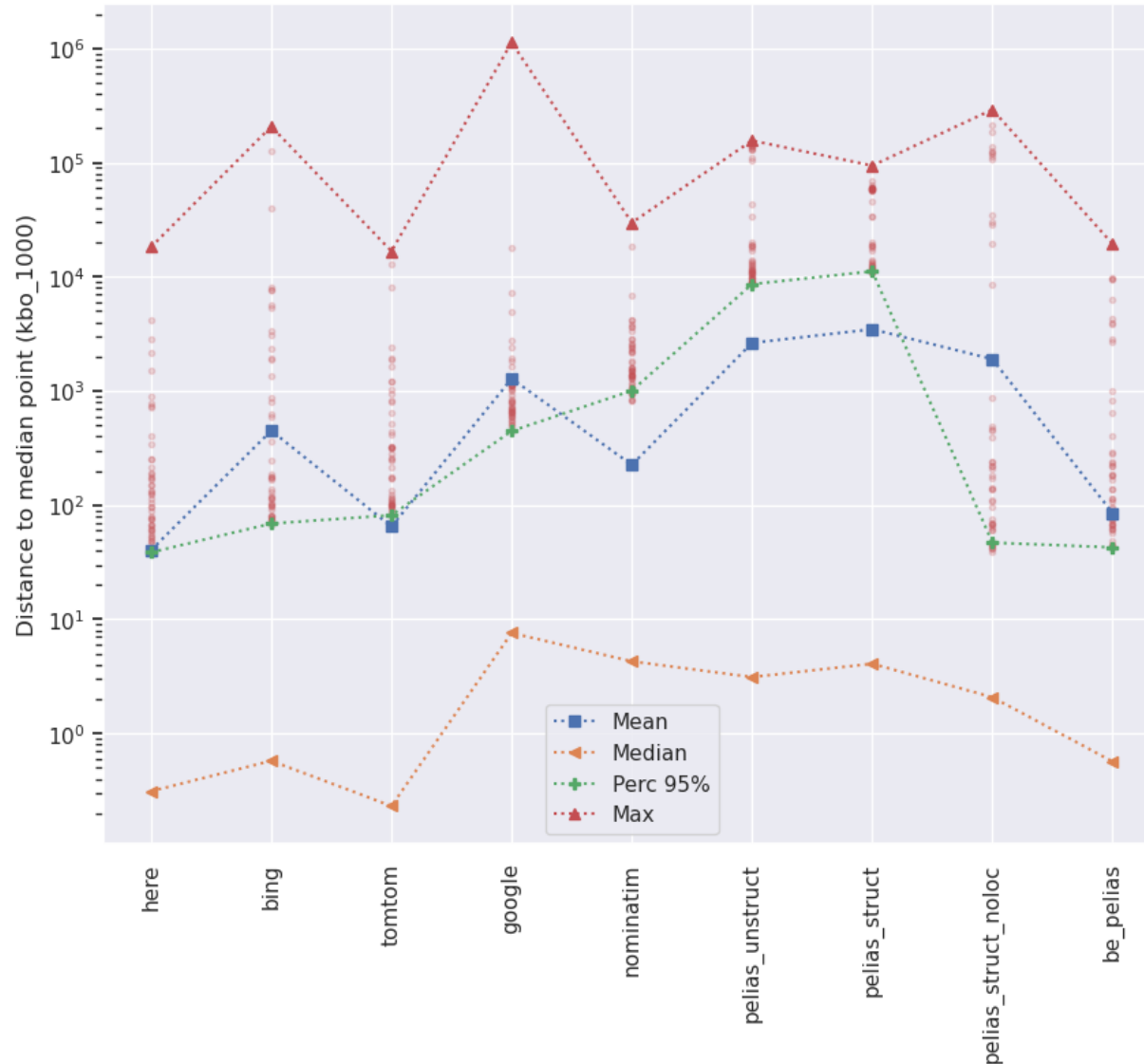
Methodology

- Collect a (not too clean) **address dataset** → 1 000 addresses from KBO
- Send this dataset to **several coders** → here, bing, nominatim, google...
- Use **median point** for each address as reference, with **some conditions**:
 - Enough (precise) matches → 957 with $\geq 3/7$ “building” matches
 - Enough answers close to median → 932 with $\geq 3/7$ within 100 m
- Compute misc. stats about « **distance to median point** »



Distance to median point

OS/on-prem



The lower the better!



! No stats if no building level results
→ « bad » geocoders are even worse

Deployment

- Available on <https://github.com/SmalsResearch/bePelias/>
- Docker + Python + FastAPI
- Deploy :
 - `./scripts/build.sh` # Build pelias & bepelias docker images (~30 minutes)
 - `./scripts/feed.sh` # Prepare files from Bosa and load them (~1h30)
 - `./scripts/run.sh` # Run Pelias and bePelias API (with default parameters)
- Many variants (one region, update data, two-machines build...)
- Live (test) instance available on request
- Unstructured mode available

Call

[https://\[host:IP\]/REST/bepelias/v1/geocode?
streetName=Avenue+Fonsny&houseNumber=20&postCode=1060&postName=Saint-Gilles](https://[host:IP]/REST/bepelias/v1/geocode?streetName=Avenue+Fonsny&houseNumber=20&postCode=1060&postName=Saint-Gilles)

```
{  items: [{
    bestId:      "https://databrussels.be/id/address/219307/7",
    street:      {name: {fr: "Avenue Fonsny",    nl: "Fonsnylaan"},
                  id: "https://databrussels.be/id/streetname/4921/2"},
    municipality: {name: {fr: "Saint-Gilles",    nl: "Sint-Gillis"},
                  code: "21013",
                  id: "https://databrussels.be/id/municipality/21013/14"},
    postalInfo:  {name: {fr: "Saint-Gilles",    nl: "Sint-Gillis"},
                  postalCode: "1060"},
    housenumber: "20",
    status:      "current",
    coordinates: {lat: 50.83583, lon: 4.33845},
    precision:   "address"}],
  total: 1,
  callType: "struct",
  inAddr: {address: "Avenue Fonsny, 20",
            locality: "",
            postalcode: "1060"},
  peliasCallCount: 3,
  transformers: "clean;no_city",
  self: "..."} }
```

OpenAPI

bePelias API 1.0.0 OAS 3.1

[/REST/bepelias/v1/openapi.json](#)

A service that allows geocoding (postal address cleansing and conversion into geographical coordinates), based on Pelias and BestAddresses.

Code available on <https://github.com/SmalsResearch/bePelias/>

[Vandy BERTEN - Website](#)

[Send email to Vandy BERTEN](#)

Servers

/REST/bepelias/v1 ▾

default



GET **/geocode** Geocode



GET **/geocode/unstructured** Geocode Unstructured



GET **/reverse** Geocode Reverse



GET **/searchCity** Search City



GET **/id/{bestid}** Get By Id



GET **/health** Health



OpenAPI

Schemas ^		
BePeliError	>	Expand all object
BoxInfo	>	Expand all object
Coordinates	>	Expand all object
GeocodeOutput	>	Expand all object
GetByldOutput	>	Expand all object
Health	>	Expand all object
HealthDetails	>	Expand all object
Item	>	Expand all object
Municipality	>	Expand all object
Name	>	Expand all object
PartOfMunicipality	>	Expand all object
PostallInfo	>	Expand all object
ReverseGeocodeOutput	>	Expand all object
SearchCityOutput	>	Expand all object
Street	>	Expand all object
ValidationError	>	Expand all object
HttpValidationError	>	Expand all object

Conclusions

- bePelias: on-premise geocoder (REST API) based on BeSt Address (Belgium)
- Simple deployment
- Similar performance as main cloud actors
- Only for addresses (not for « Smals, Saint-Gilles » or « Resto à Ixelles »)

Special thanks to « Crisis Centrum » (M. Ducobu, G. Kruwialis) for their input

Vandy BERTEN

vandy.berten@smals.be

<https://www.smalsresearch.be/author/berten/>

Smals, ICT for society

02 787 57 11

Fonsnylaan 20 / Avenue Fonsny 20

1060 Brussel / 1060 Bruxelles