camp
tocamp

INNOVATIVE SOLUTIONS
BY OPEN SOURCE EXPERTS

Geodaten-Management mit PostGIS

Marion Baumgartner

About me

Marion Baumgartner

- Full stack GIS development
- ETL with geo-data
- https://github.com/marionb

About Camptocamp

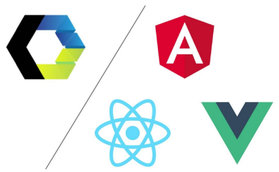
Your partner for success.

- Founded in 2001
- 190+ employees
- Offices in 3 countries:
  - Switzerland, Germany, France
- Geographic Information Systems, Enterprise Resource Planning (Odoo), IT Infrastructure Management
- A major European player in Open Source

Open Source Geospatial Software

20+ years contributions

Geospatial Open Source Software stack



OGC®
Making location count.

OpenLayers

CESIUMJS™

GeoMapFish

geOrchestra

spring

MapServer
open source web mapping

QGIS

GeoServer

About the Presentation

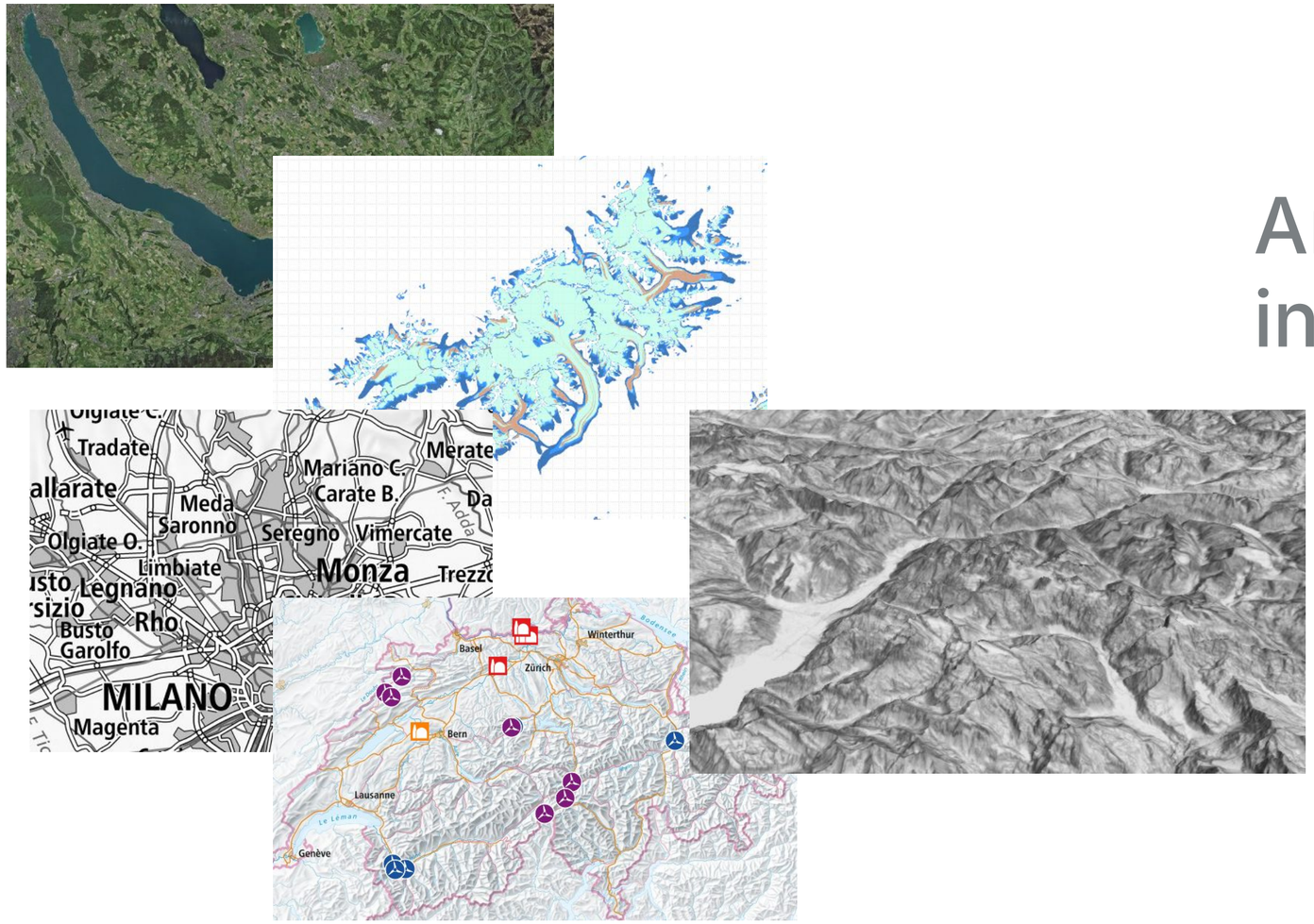
Why PostGIS

The Data Model

Testing with OSM

Harvesting the Data

What is geodata?



Any data with spatial information!

What is the format of geodata?

GML

GPX

csv

geotiff

jpg

gdb

NetCDF

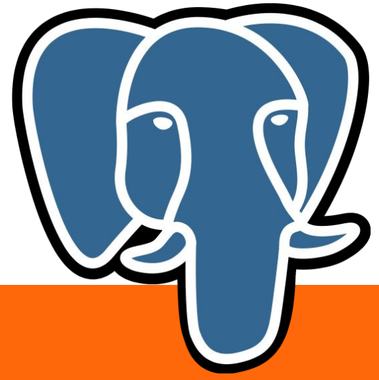
topojson

KML

shp

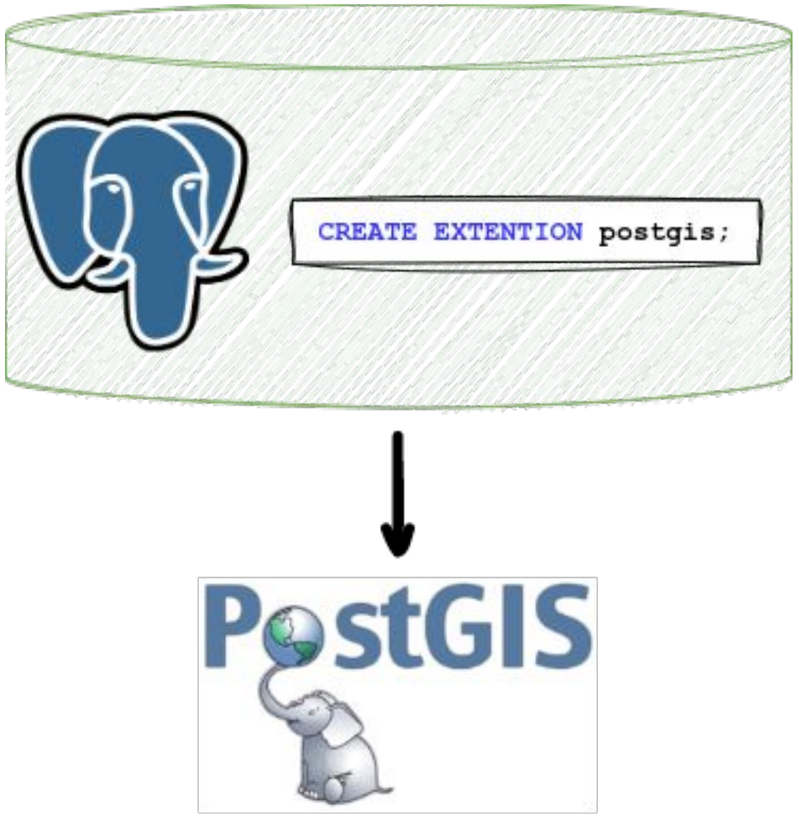
GeoPackage

osm



Why PostGIS

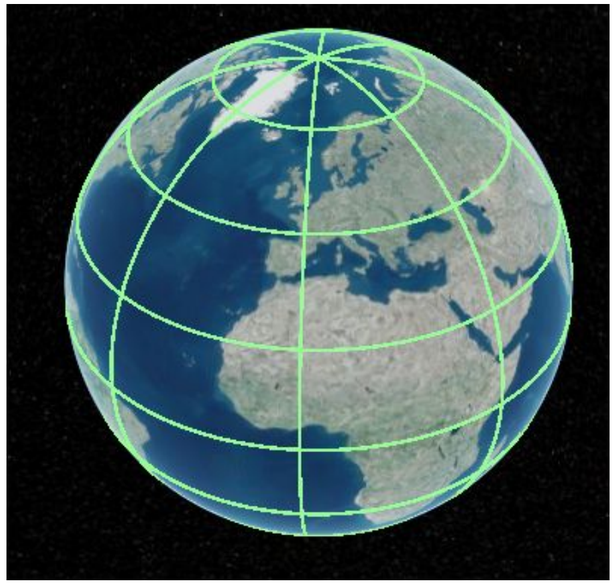
What is PostGIS?



PostGIS is a spatial database:

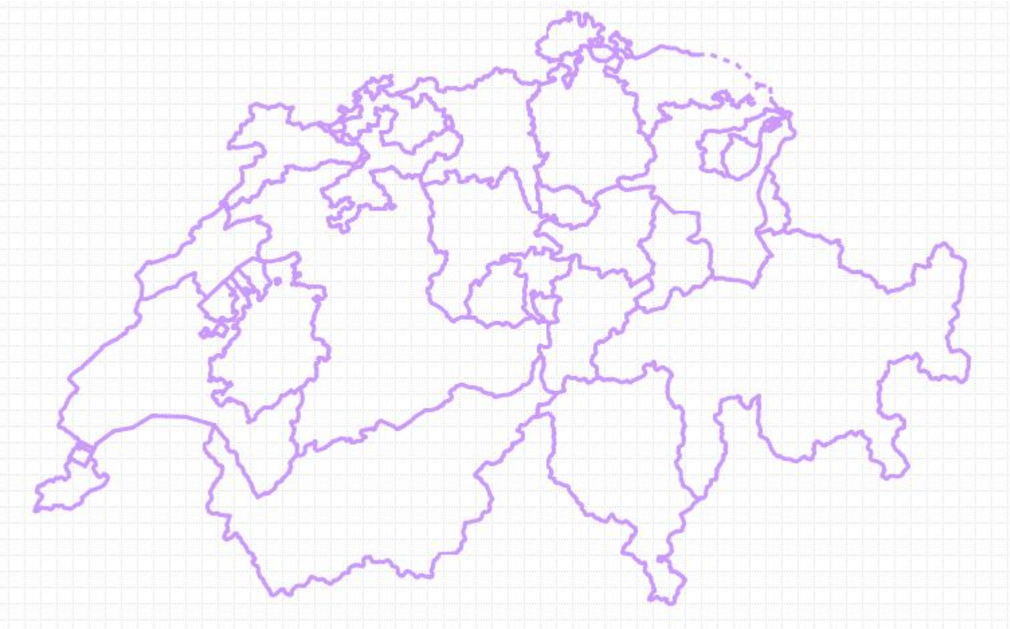
- → Spatial data types
  - geometry (point, lines, polygons)
  - raster
- → Spatial indexing
  - Optimised for spatially related data
- → Spatial functions
  - ST_...

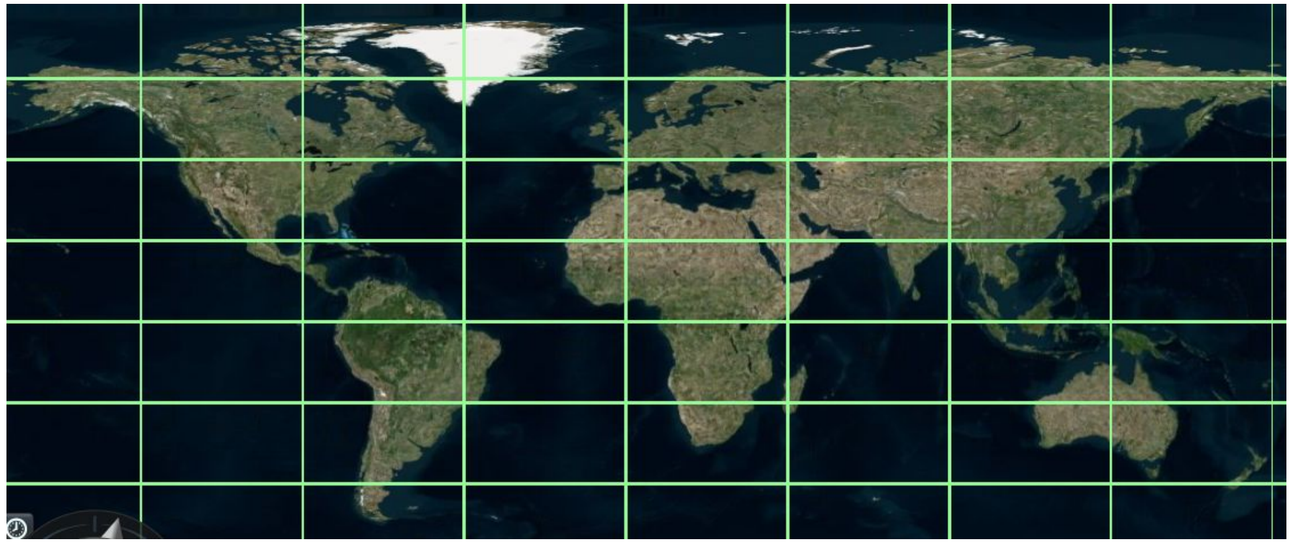
Data Types



Vector data

- → geography
- → geometry



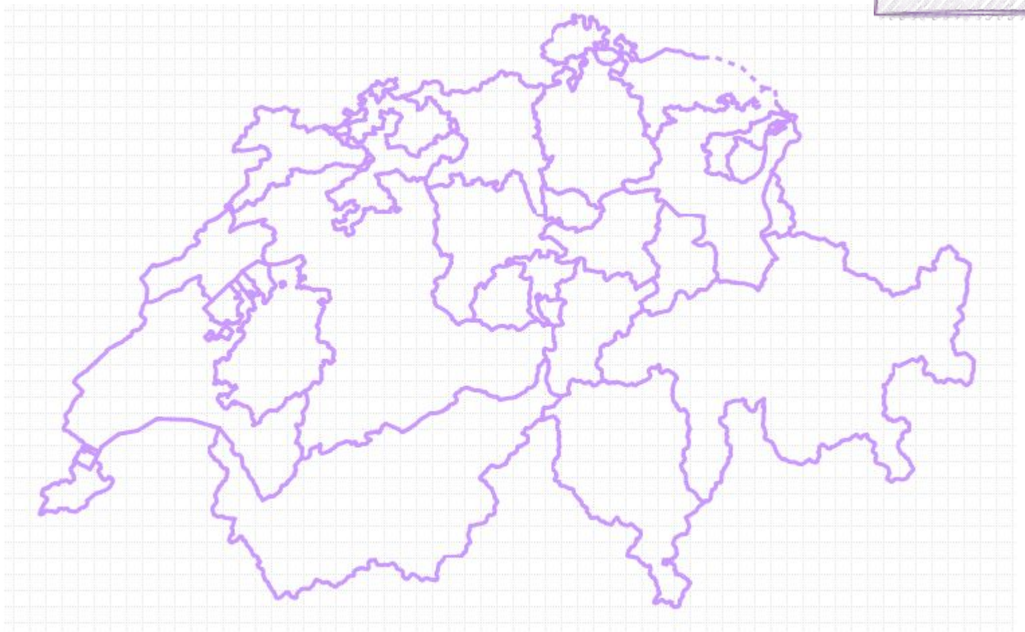


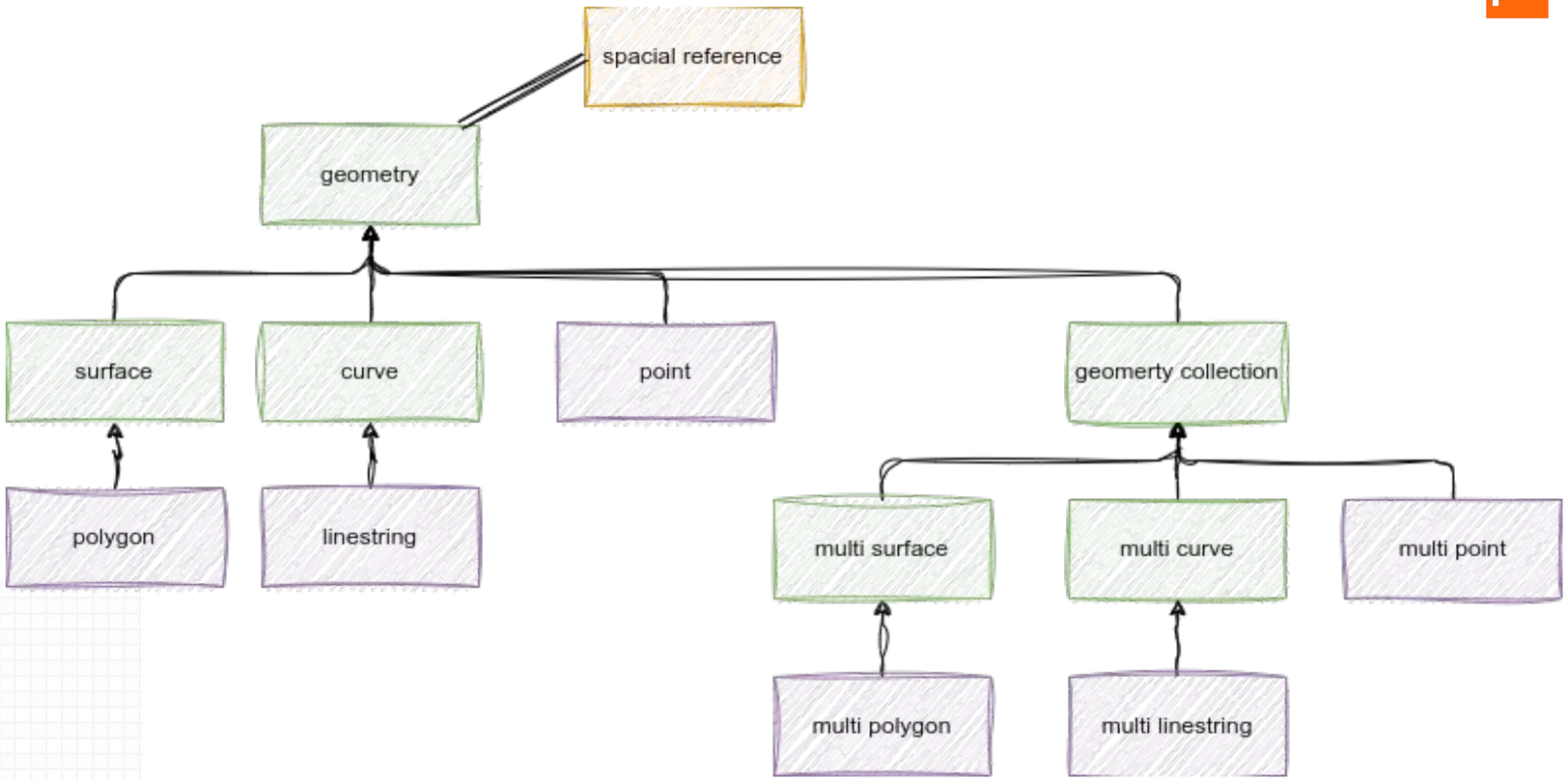
https://sandcastle.cesium.com/

Data Types

Vector data

- geography
- geometry

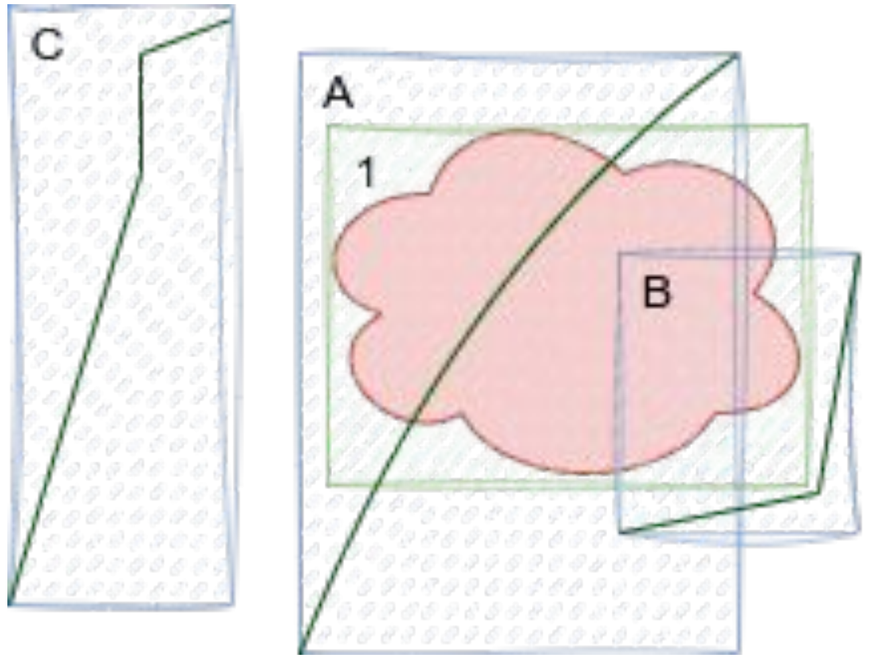




Data Types

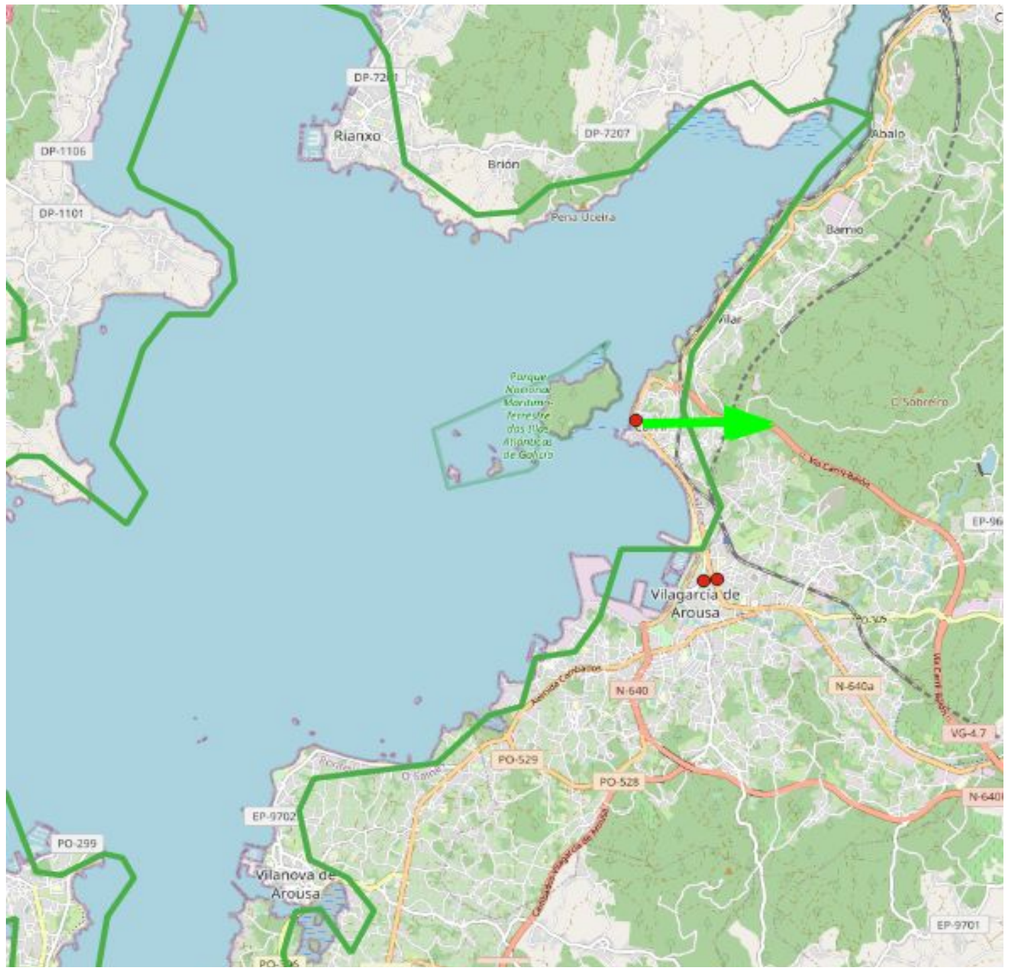
Raster data

Spatial indexes

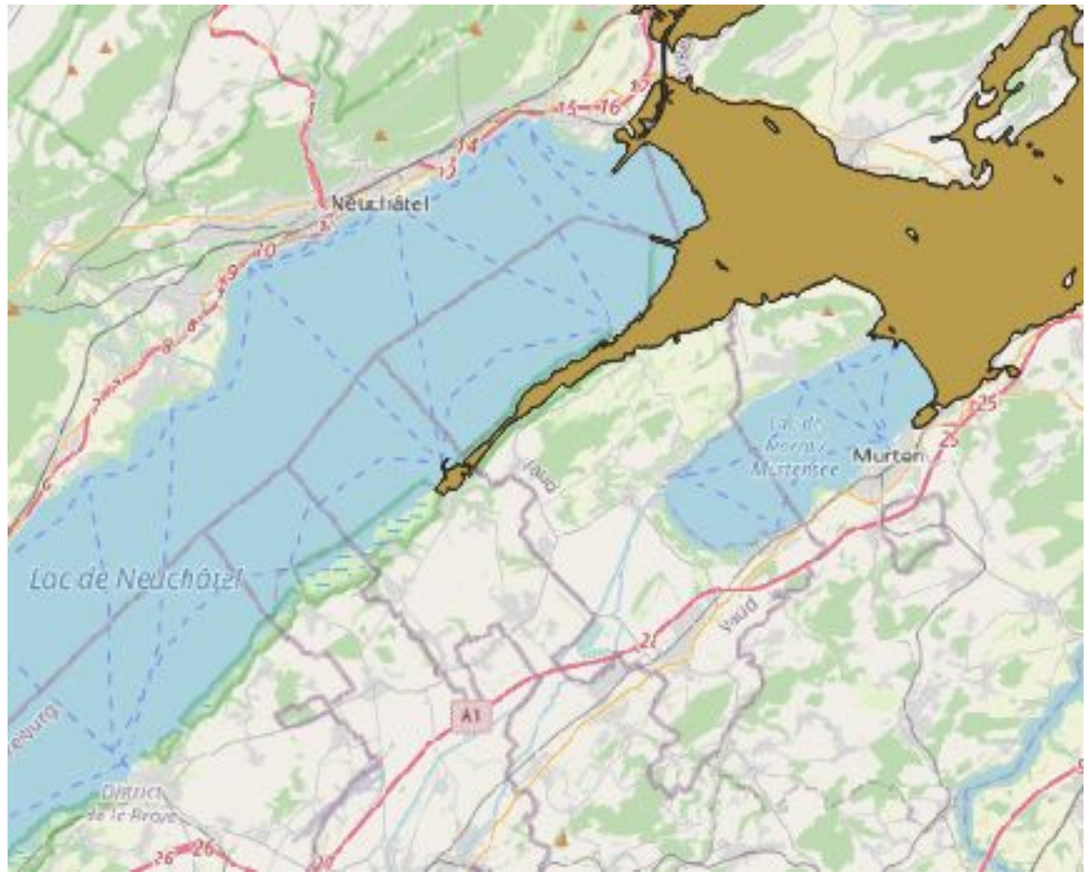


Spatial functions

```
ST_Distance(geometry A,geometry B)
```

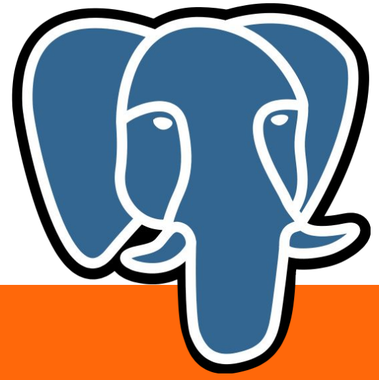



```
ST_Simplify(geometry A, float tolerance,
            [boolean preserveCollapsed])
```



Why PostGIS?

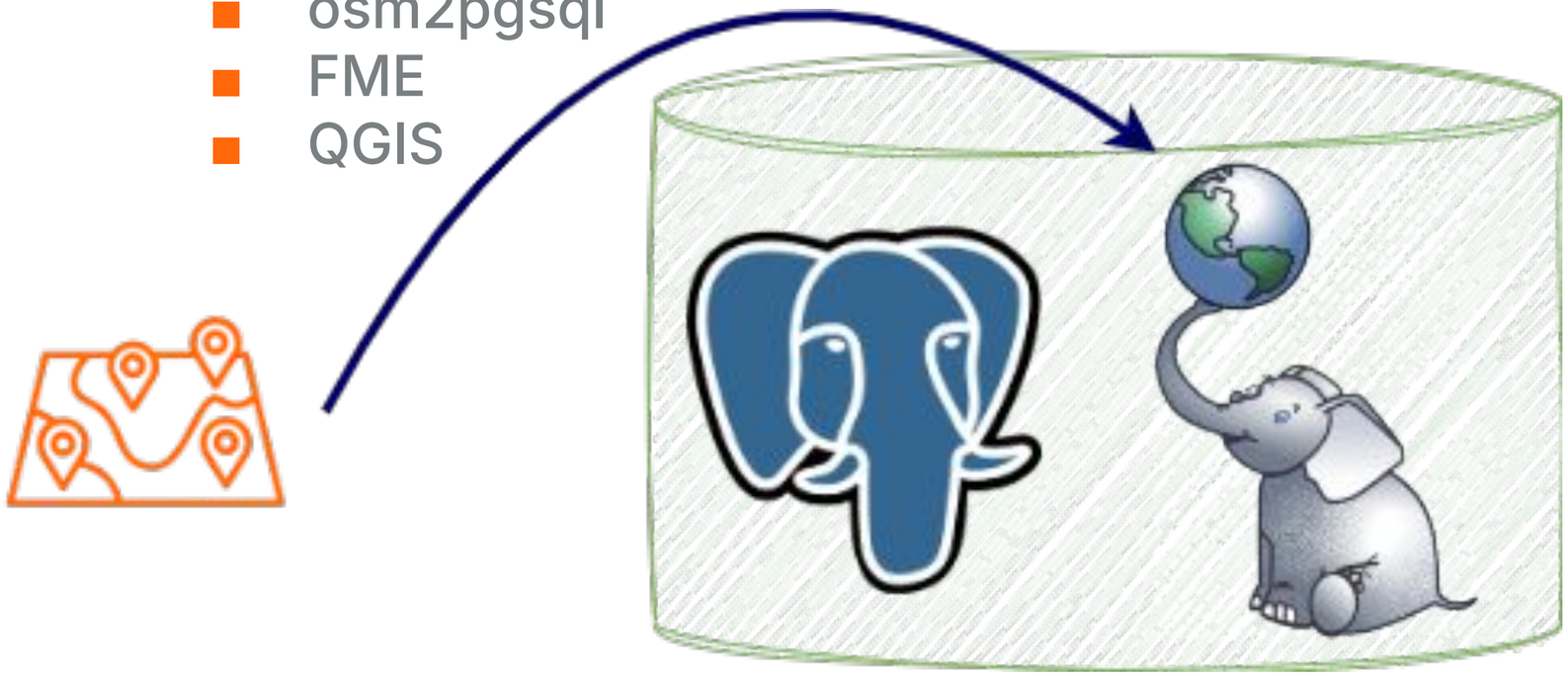
- Open source (GNU General Public License)
- A long history and active community
- Well documented
  - https://postgis.net/documentation/



The Data Model

Filling data into a PostGIS DB

- GDAL
- shp2pgsql
- osm2pgsql
- FME
- QGIS



Geodata in a Table

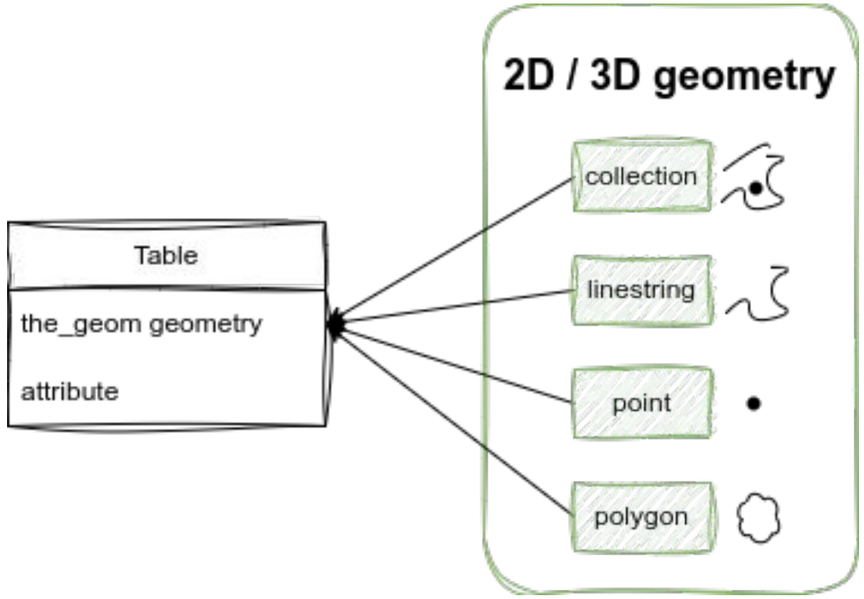
Putting **EVERYTHING** in one table and one column

- → geography
- → geometry
- → raster

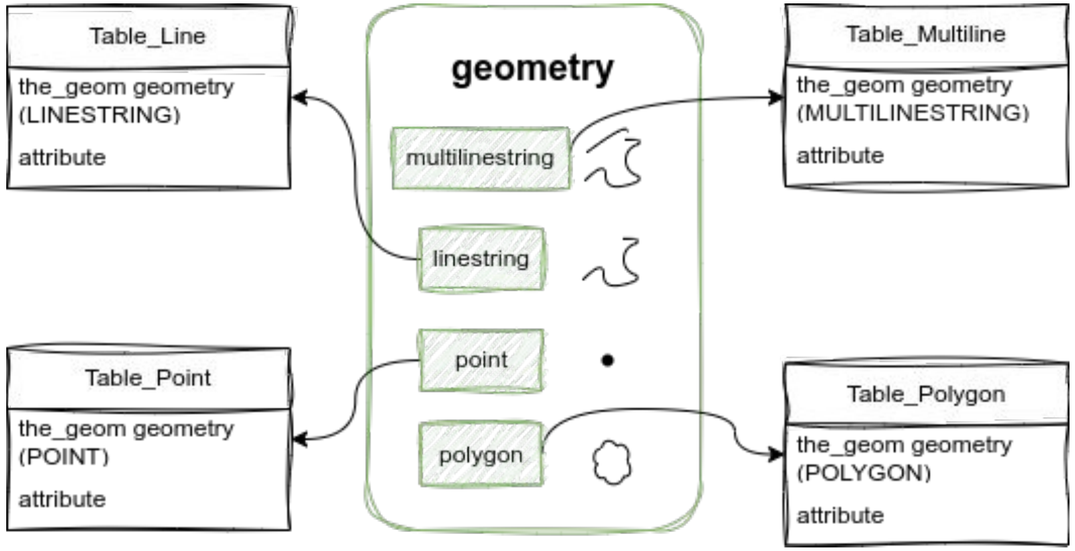
you are likely in trouble

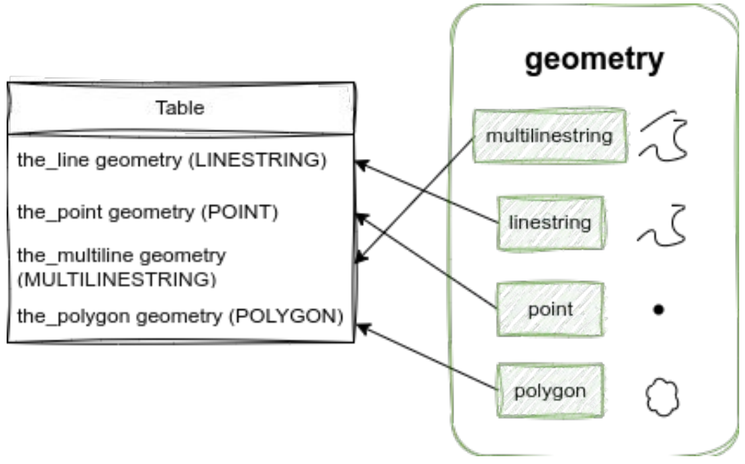
Geodata in a Table

Heterogeneous



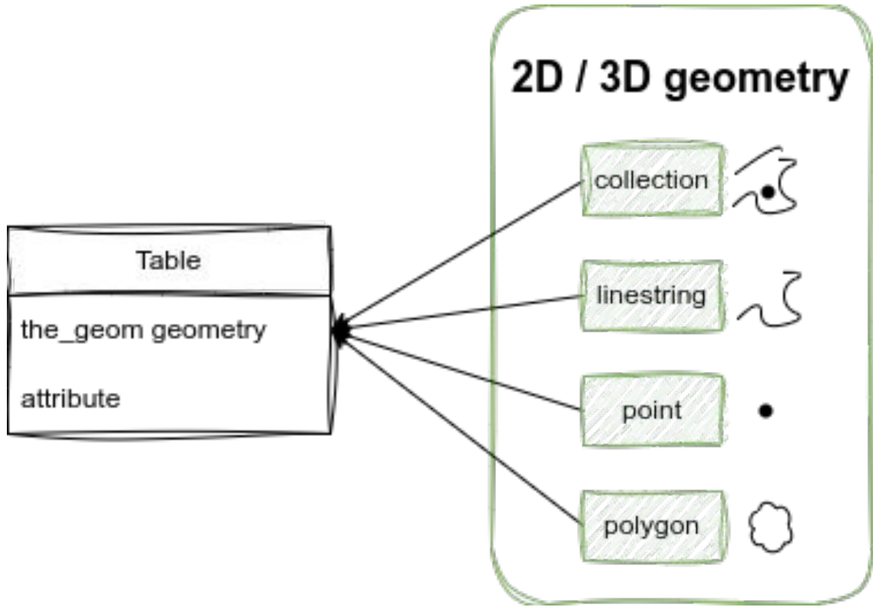
Homogeneous





Heterogeneous

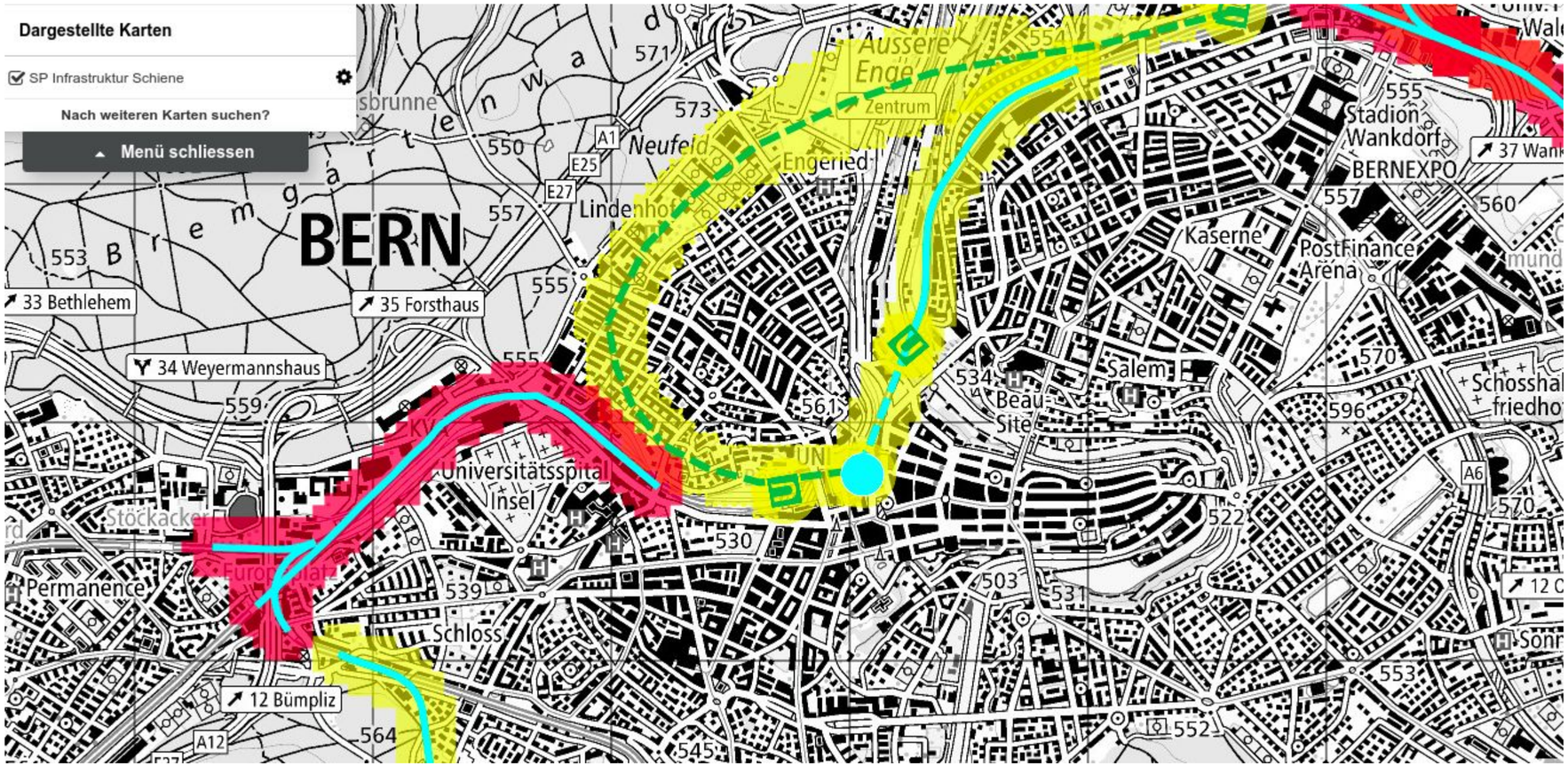
Pros & Cons



➔ Simple
- query
- model

➔ Data corruption
➔ Need to filter on types
➔ Self joins when aggregating
➔ Hard to read for many tools

Heterogeneous: Example

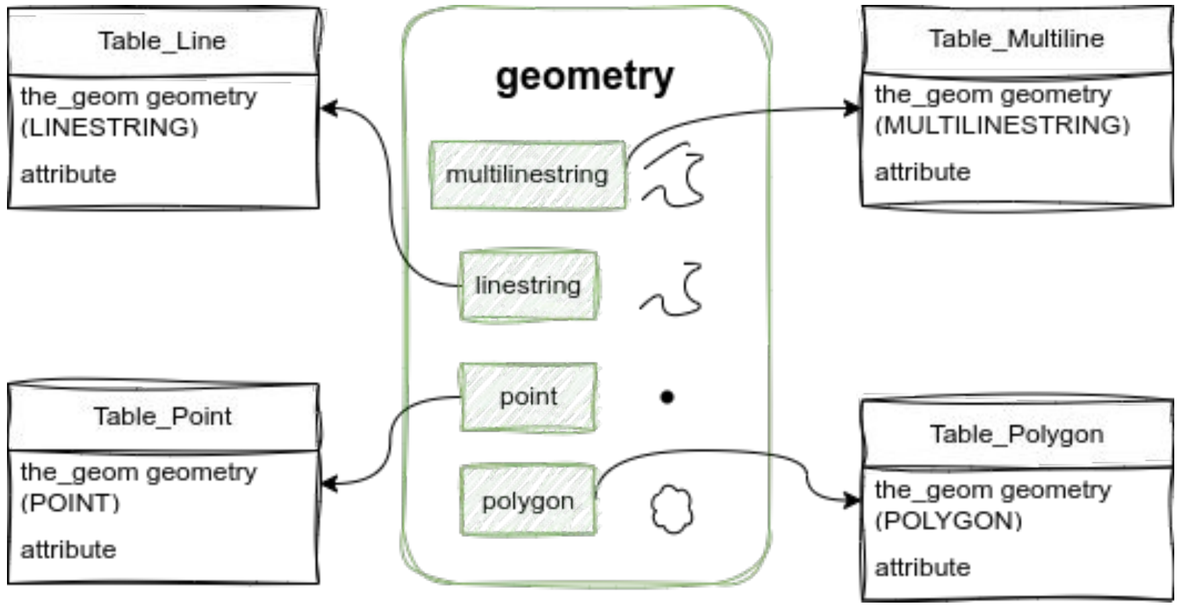


ST_GeometryType:

ST_MultiPolygon
ST_MultiPoint
ST_MultiLineString

Homogeneous

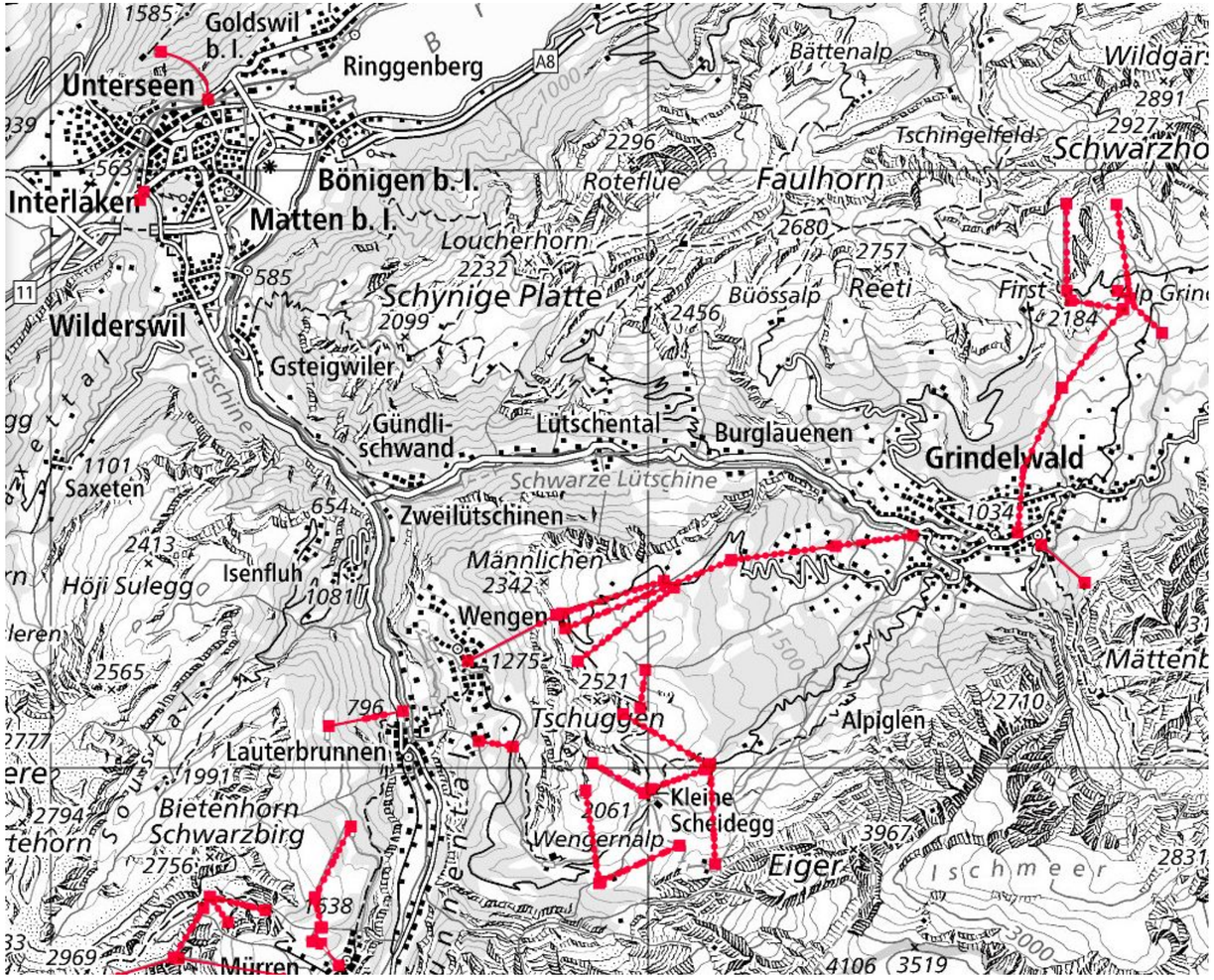
Pros & Cons



- ➔ Consistency
- ➔ Easier to access
- ➔ Performance on joining
- ➔ Easier for handling monstrous data set

- ➔ Multi-geometry queries need a union

Homogeneous Table: Example



Multiple tables with constraint on geometry type:

- ST_LineString

  OR

- ST_Point

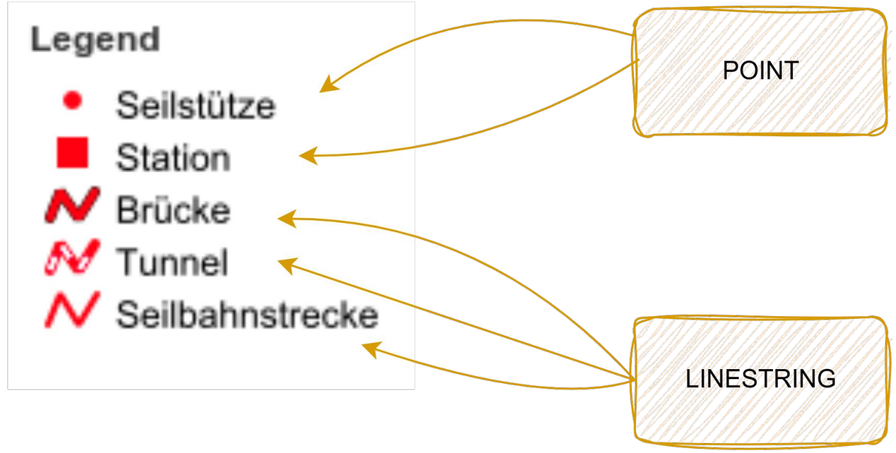


Table Inheritance

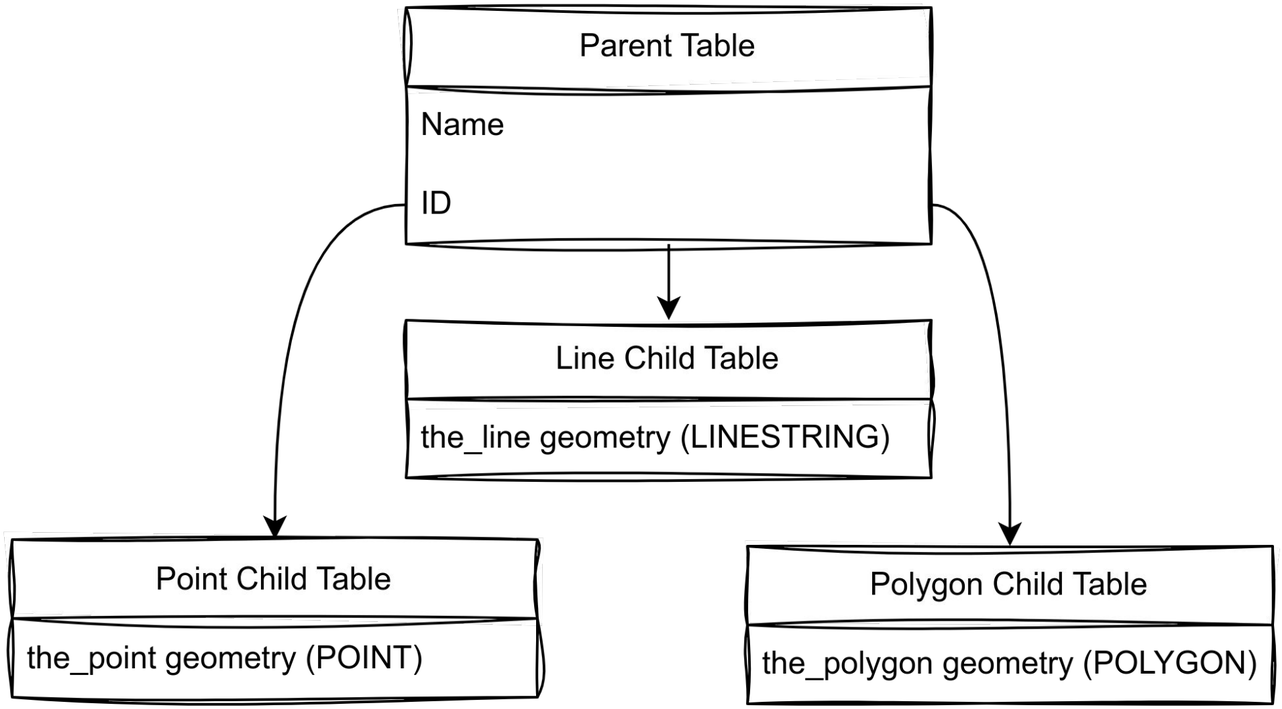


Table Inheritance

Pros & Cons

- Supported by most third party tools
- Query a hierarchy as if they were single table
- Query for specific geom type if splitting by geom

- Complex data model
- Unique in Postgresql
- Primary & foreign key constraints are not inherited
- Data is not added automatically in the correct table
- Performance

Table Partitioning

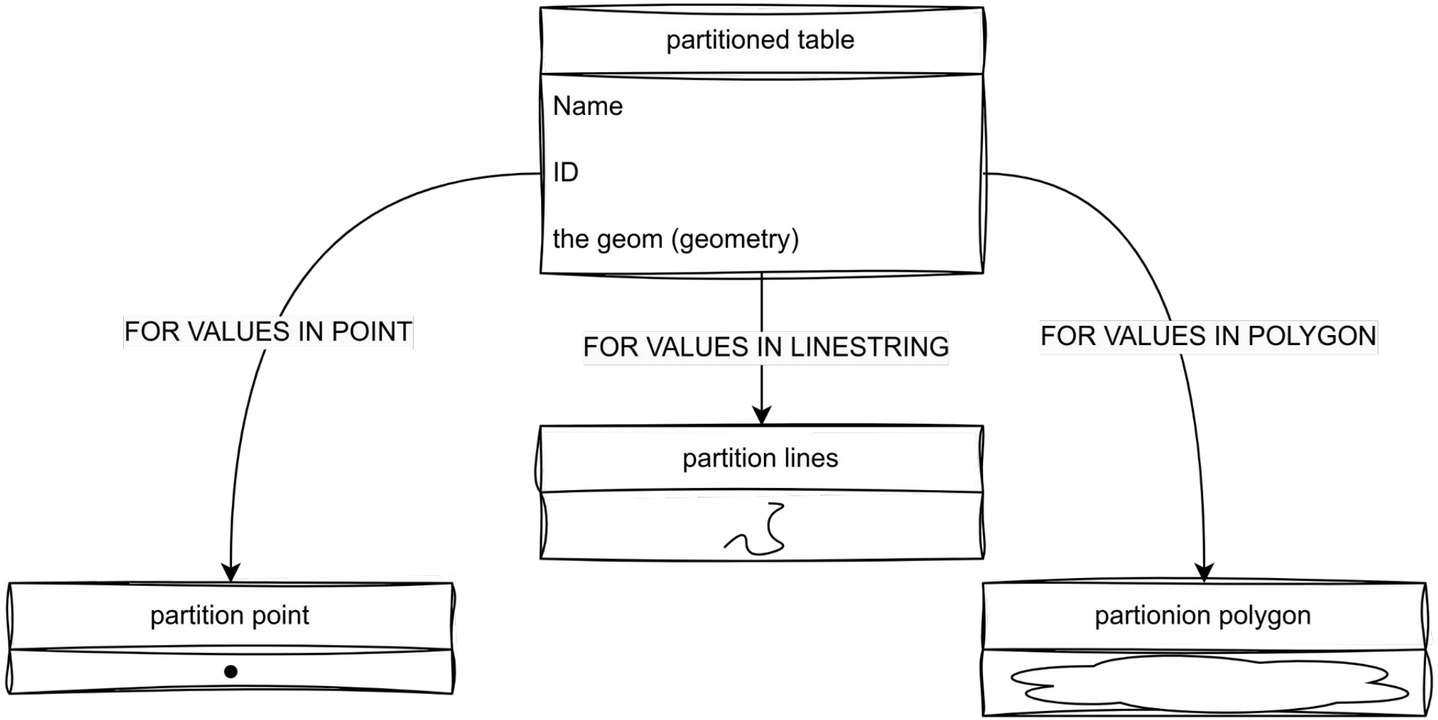
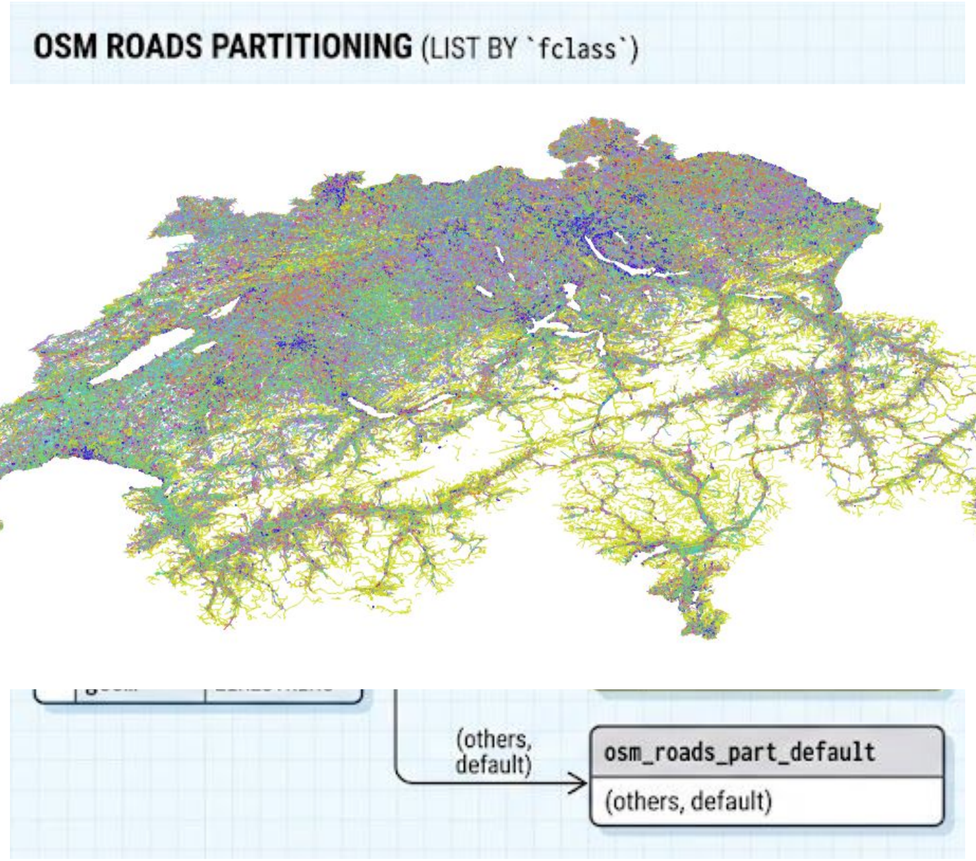
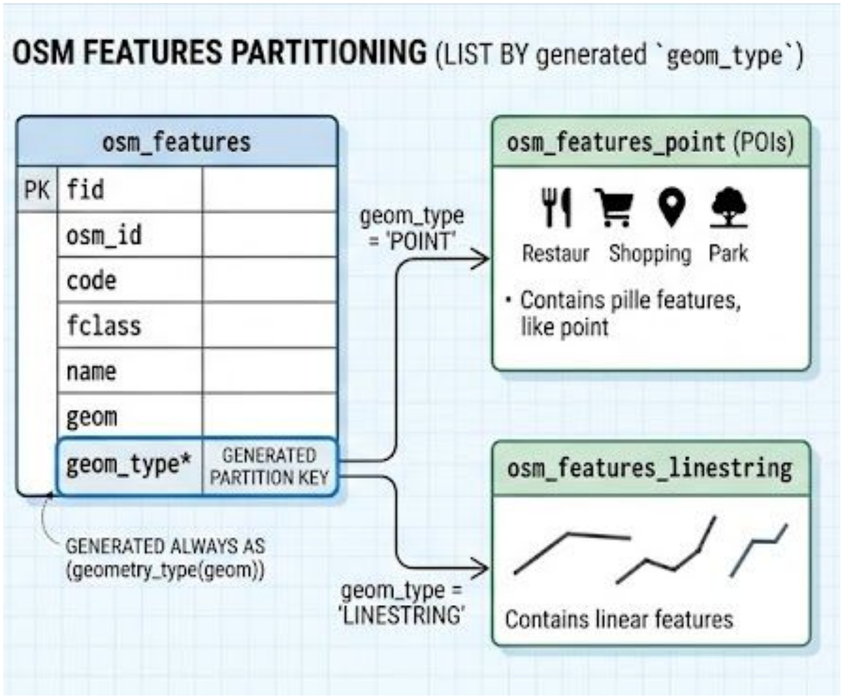
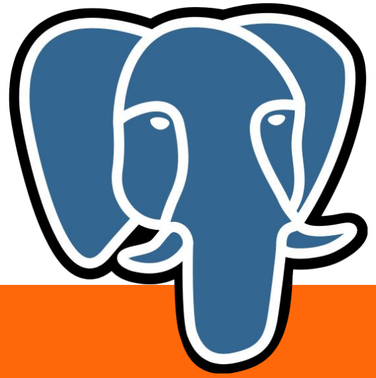


Table Partitioning: Example







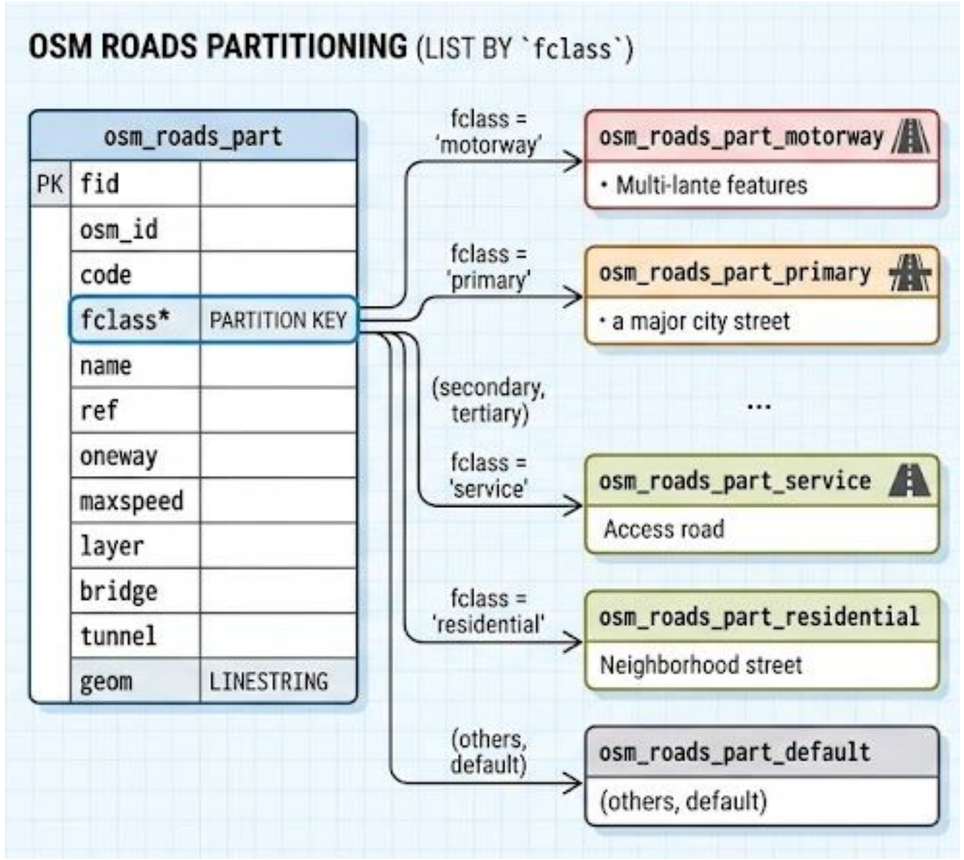
Testing with OSM Data

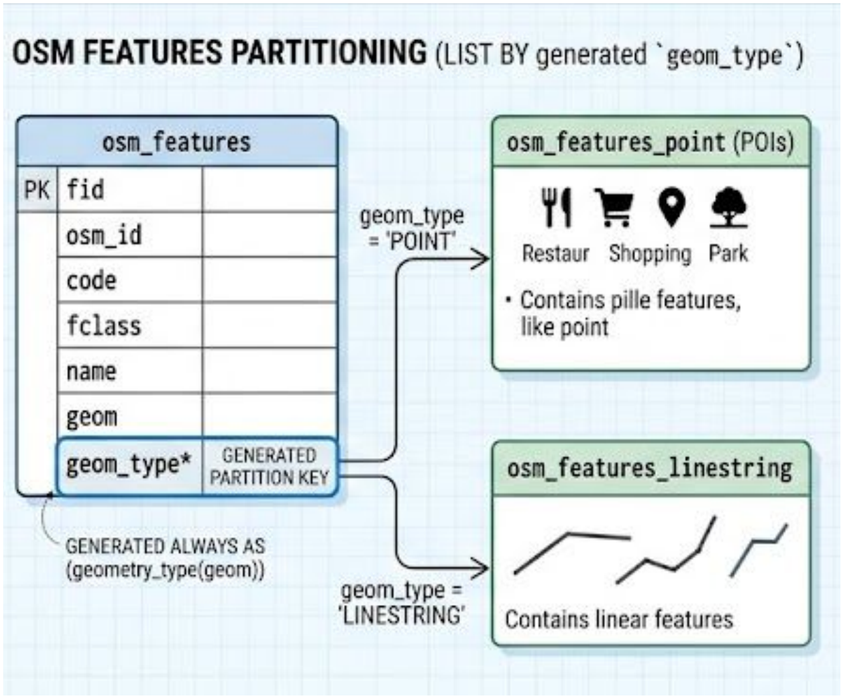
What is OSM ?

the project that creates and distributes free geographic data for the world.

We started it because most maps you think of as free actually have legal or technical restrictions on their use, holding back people from using them in creative, productive, or unexpected ways.

The Setup


OSM ROADS PARTITIONING (LIST BY `fclass`)
osm_roads_part
PK fid
osm_id
code
fclass* PARTITION KEY
name
ref
oneway
maxspeed
layer
bridge
tunnel
geom LINESTRING
fclass = 'motorway'
osm_roads_part_motorway
• Multi-lante features
fclass = 'primary'
osm_roads_part_primary
• a major city street
(secondary, tertiary)
...
fclass = 'service'
osm_roads_part_service
Access road
fclass = 'residential'
osm_roads_part_residential
Neighborhood street
(others, default)
osm_roads_part_default
(others, default)


OSM FEATURES PARTITIONING (LIST BY generated `geom_type`)
osm_features
PK fid
osm_id
code
fclass
name
geom
geom_type* GENERATED PARTITION KEY
GENERATED ALWAYS AS (geometry_type(geom))
geom_type = 'POINT'
osm_features_point (POIs)
Restaur
Shopping
Park
• Contains pille features, like point
geom_type = 'LINESTRING'
osm_features_linestring
Contains linear features

The Setup

```
>>> Row counts per partition:
           partition              |     fclass      |  rows
----------------------------------+-----------------+--------
 benchmark.osm_roads_motorway     | motorway        |   8909
 benchmark.osm_roads_motorway     | motorway_link   |   6010
 benchmark.osm_roads_primary      | primary         |  34892
 benchmark.osm_roads_primary      | primary_link    |   1117
 benchmark.osm_roads_secondary    | secondary       |  37483
 benchmark.osm_roads_secondary    | secondary_link  |    709
 benchmark.osm_roads_tertiary     | tertiary        |  50744
 benchmark.osm_roads_tertiary     | tertiary_link   |    411
 benchmark.osm_roads_residential  | living_street   |   6662
 benchmark.osm_roads_residential  | residential     | 189269
 benchmark.osm_roads_service      | cycleway        |  21388
 benchmark.osm_roads_service      | footway         | 425475
 benchmark.osm_roads_service      | path            | 219390
 benchmark.osm_roads_service      | service         | 466416
 benchmark.osm_roads_service      | track           |  76377
 benchmark.osm_roads_default      | bridleway       |    371
 benchmark.osm_roads_default      | busway          |     75
 benchmark.osm_roads_default      | pedestrian      |   7383
 benchmark.osm_roads_default      | steps           |  66075
 benchmark.osm_roads_default      | track_grade1    |  58842
 benchmark.osm_roads_default      | track_grade2    |  99535
 benchmark.osm_roads_default      | track_grade3    |  74655
 benchmark.osm_roads_default      | track_grade4    |  35269
 benchmark.osm_roads_default      | track_grade5    |  26379
 benchmark.osm_roads_default      | trunk           |   2418
 benchmark.osm_roads_default      | trunk_link      |   1061
 benchmark.osm_roads_default      | unclassified    |  70623
 benchmark.osm_roads_default      | unknown         |     37
(28 rows)
```

The Setup

```
>>> Row counts per partition:
             partition               | geom_type  |  rows
-------------------------------------+------------+---------
 benchmark.osm_features_point        | POINT      |  381480
 benchmark.osm_features_linestring | LINESTRING | 1987975
(2 rows)
```

Queries

Flat Table Queries	Partitioned Queries
Q1: COUNT(*) full scan `SELECT count(*) FROM benchmark.osm_roads;`	`SELECT count(*) FROM benchmark.osm_roads_part;`
Q2: Filter on partition key `SELECT count(*) FROM benchmark.osm_roads WHERE fclass = 'motorway';`	`SELECT count(*) FROM benchmark.osm_roads_part WHERE fclass = 'motorway';`
Q3: Filter on non-partition col `SELECT count(*) FROM benchmark.osm_roads WHERE maxspeed > 100;`	`SELECT count(*) FROM benchmark.osm_roads_part WHERE maxspeed > 100;`
Q4: Combined filter `SELECT count(*) FROM benchmark.osm_roads WHERE fclass = 'motorway' AND maxspeed > 100;`	`SELECT count(*) FROM benchmark.osm_roads_part WHERE fclass = 'motorway' AND maxspeed > 100;`
Q5: Spatial filter `SELECT count(*) FROM benchmark.osm_roads WHERE ST_DWithin(geom, ST_SetSRID(ST_MakePoint(7.45, 46.95), 4326), 0.01);`	`SELECT count(*) FROM benchmark.osm_roads_part WHERE ST_DWithin(geom, ST_SetSRID(ST_MakePoint(7.45, 46.95), 4326), 0.01);`

Queries

Flat Table Queries	Partitioned Queries
Q6: Spatial join `SELECT count(*) FROM benchmark.osm_roads r JOIN benchmark.osm_landuse l ON ST_Intersects(r.geom, l.geom) WHERE l.fclass = 'forest';`	`SELECT count(*) FROM benchmark.osm_roads_part r JOIN benchmark.osm_landuse l ON ST_Intersects(r.geom, l.geom) WHERE l.fclass = 'forest' AND r.fclass IN ('motorway', 'primary', 'secondary');`
Q7: Cross-geometry query `SELECT count(*) FROM benchmark.osm_pois p JOIN benchmark.osm_roads r ON ST_DWithin(p.geom, r.geom, 0.005) WHERE r.fclass = 'motorway' AND p.fclass = 'restaurant';`	`SELECT count(*) FROM benchmark.osm_features_point p JOIN benchmark.osm_features_linestring r ON ST_DWithin(p.geom, r.geom, 0.005) WHERE r.fclass = 'motorway' AND p.fclass = 'restaurant';`

Result Summary

Q	Scenario	Flat Table	Partitioned	Winner
Q1	Full Scan COUNT(*)	229.42 ms	**188.20 ms**	**Partitioned (~18%)**
Q2	Filter on Partition Key	1.58 ms	**0.91 ms**	**Partitioned (~42%)**
Q3	Filter on Non-Partition Col	387.74 ms	**325.05 ms**	**Partitioned (~16%)**
Q4	Combined Filter (Key + Non-Key)	7.21 ms	**1.69 ms**	**Partitioned (~77%)**
Q5	Spatial Filter (ST_DWithin)	**12.21 ms**	13.03 ms	**Flat Table (~6%)**
Q6	Spatial Join (Roads + Landuse)	5478.45 ms	**3027.46 ms**	**Partitioned (~45%)**
Q7	Cross-Geometry-Type Join	1966.70 ms	**949.17 ms**	**Partitioned (~52%)**

```
SELECT count(*) FROM benchmark.osm_roads_part
WHERE maxspeed > 100;
```

no pruning possible.

Q	Scenario	Flat Table	Partitioned	Winner
Q1	Full Scan COUNT(*)	229.42 ms	**188.20 ms**	**Partitioned (~18%)**
Q2	Filter on Partition Key	1.58 ms	**0.91 ms**	**Partitioned (~42%)**
Q3	Filter on Non-Partition Col	387.74 ms	**325.05 ms**	**Partitioned (~16%)**
Q4	Combined Filter (Key + Non-Key)	7.21 ms	**1.69 ms**	**Partitioned (~77%)**
Q5	Spatial Filter (ST_DWithin)	**12.21 ms**	13.03 ms	**Flat Table (~6%)**
Q6	Spatial Join (Roads + Landuse)	5478.45 ms	**3027.46 ms**	**Partitioned (~45%)**
Q7	Cross-Geometry-Type Join	1966.70 ms	**949.17 ms**	**Partitioned (~52%)**

```
SELECT count(*) FROM benchmark.osm_roads_part
WHERE ST_DWithin(geom, ST_SetSRID(ST_MakePoint(7.45, 46.95), 4326), 0.01);
```

Spatial queries not aligned with the partition key see no benefit

			inner	
Q1	Full Scan COUNT(*)	229.42 ms	**188.20 ms**	**Partitioned (~18%)**
Q2	Filter on Partition Key	1.58 ms	**0.91 ms**	**Partitioned (~42%)**
Q3	Filter on Non-Partition Col	387.74 ms	**325.05 ms**	**Partitioned (~16%)**
Q4	Combined Filter (Key + Non-Key)	7.21 ms	**1.69 ms**	**Partitioned (~77%)**
Q5	Spatial Filter (ST_DWithin)	**12.21 ms**	13.03 ms	**Flat Table (~6%)**
Q6	Spatial Join (Roads + Landuse)	5478.45 ms	**3027.46 ms**	**Partitioned (~45%)**
Q7	Cross-Geometry-Type Join	1966.70 ms	**949.17 ms**	**Partitioned (~52%)**

→ **Geometry-Type Separation:**

- effective at improving index density and scan efficiency.
- cleans up geometric calculations and index scans during cross-layer joins.

- **Parallel Append:** Provides modest baseline gains when pruning is not possible.

- **Strategic Alignment:** Performance benefits only accrue when query filters match the partitioning strategy.

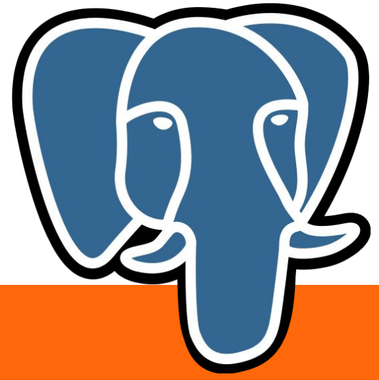
- **Beware Global Spatial Queries**: If you frequently query maps spatially without filtering on your partition key, partitioning by an attribute (like road class) will actually slow down your spatial lookups due to multi-index overhead.

- **Amount of Data:** Do not partition if the table is small

Pros & Cons

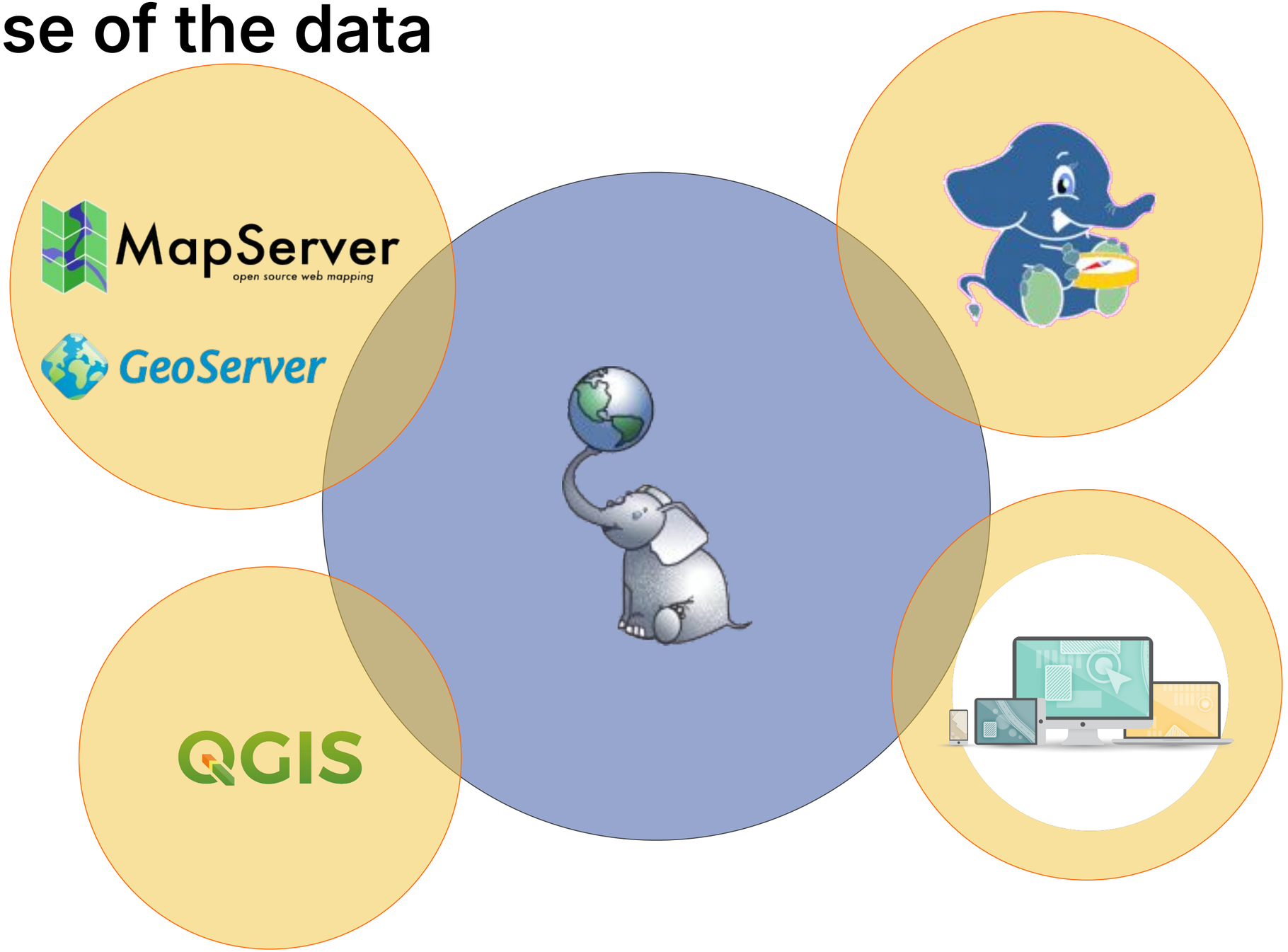
- Is understood by third-party tools (i.e QGIS)
- Generally good query performance
- Insert & update is automatically
- Indexes are inherited

- Complex data model
- No additional columns in “sub-tables”
- better used with **huge** tables
- strategy/model needs to be predefined
- Partitioning keys are limited

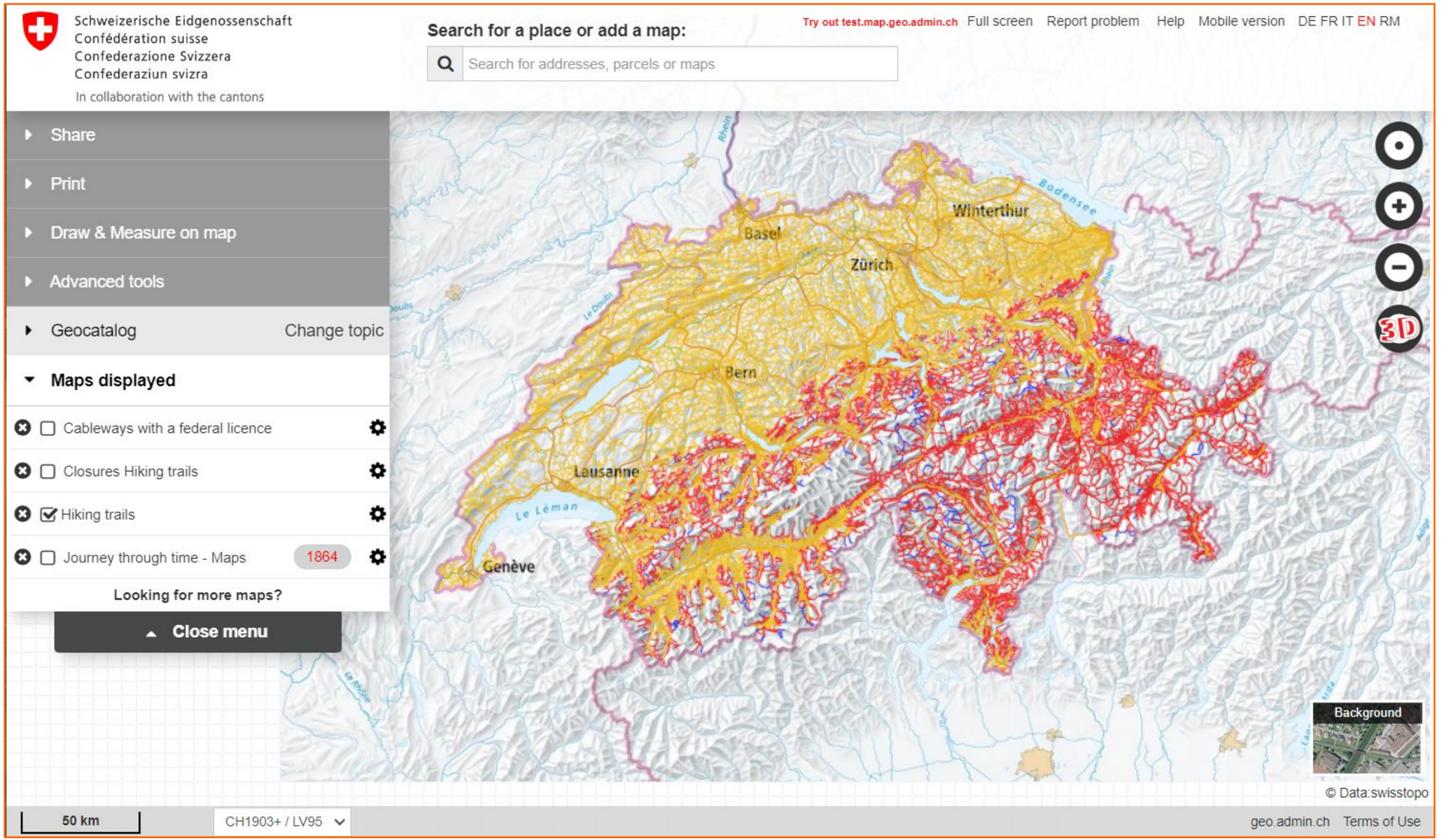


Harvesting the Data

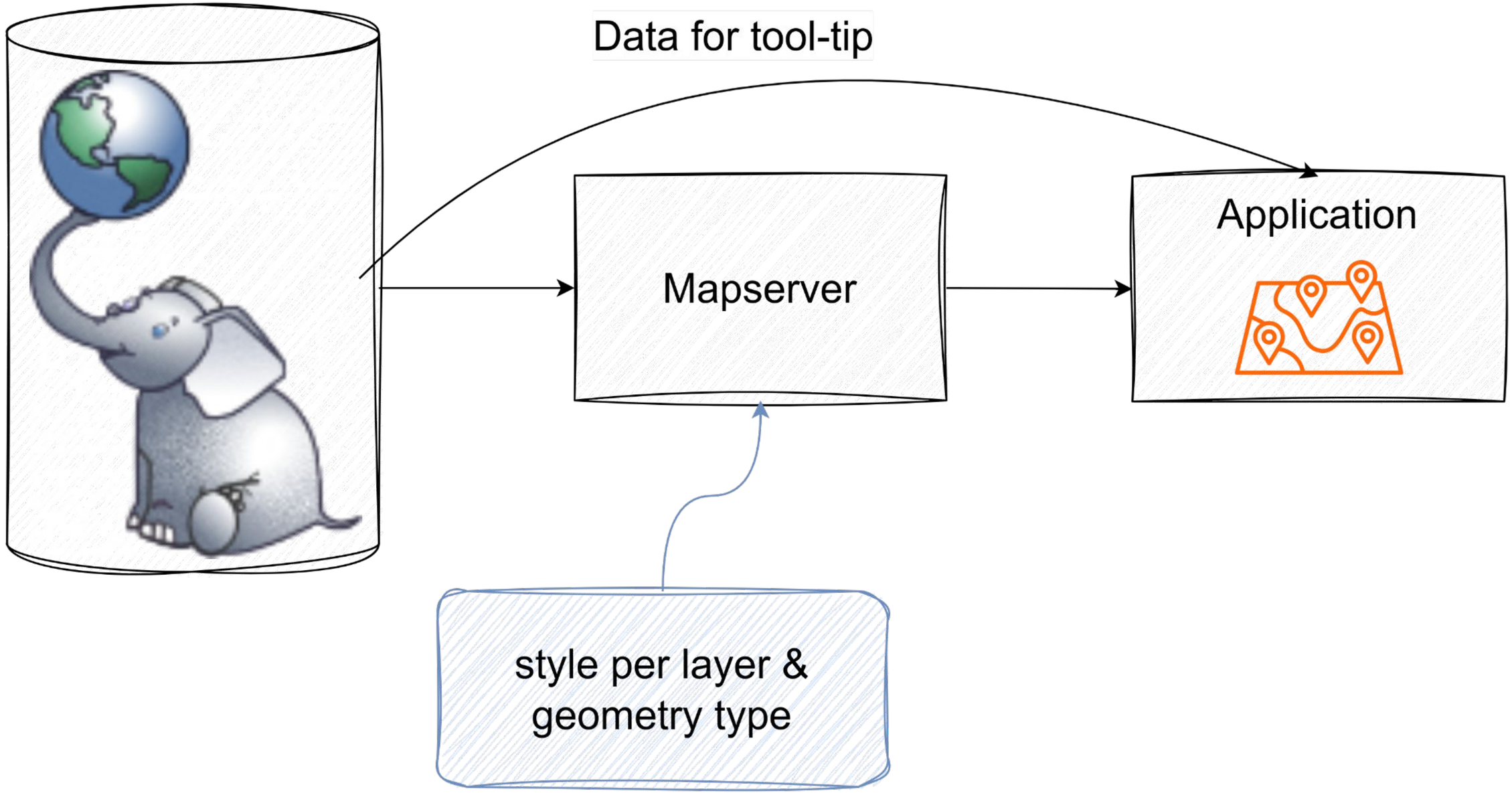
Make use of the data



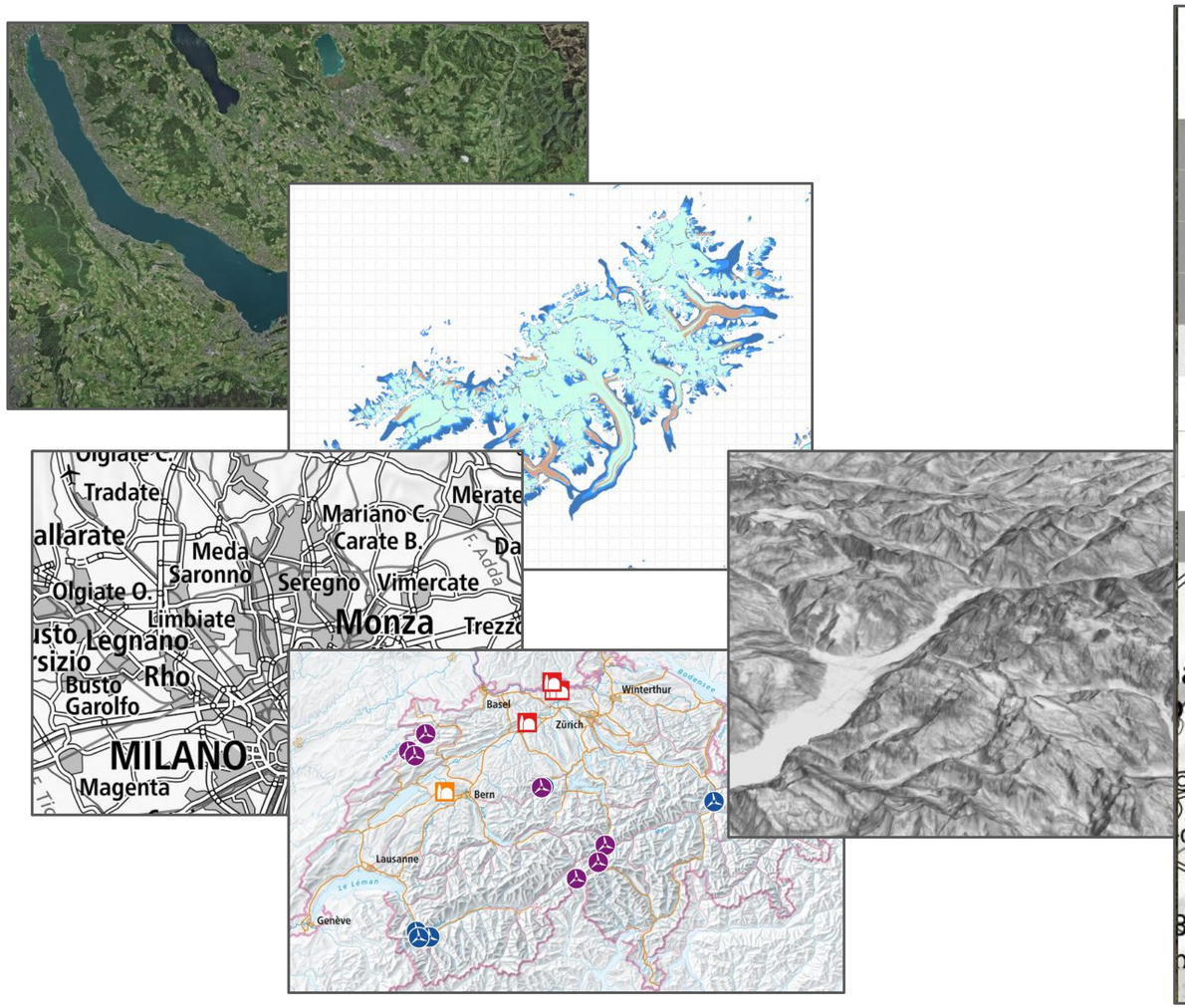
Swisstopo: A large GDI

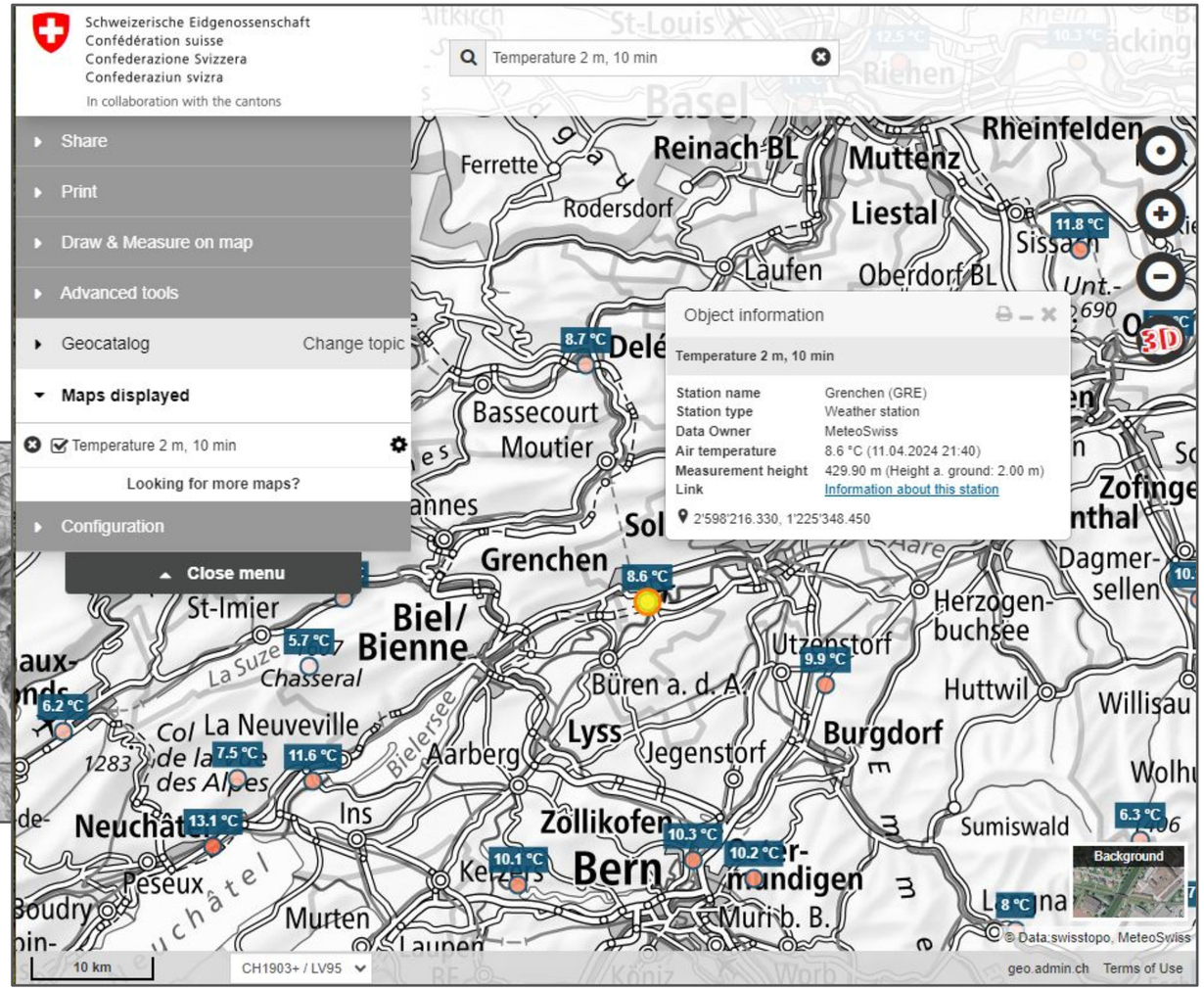


Swisstopo: A large GDI

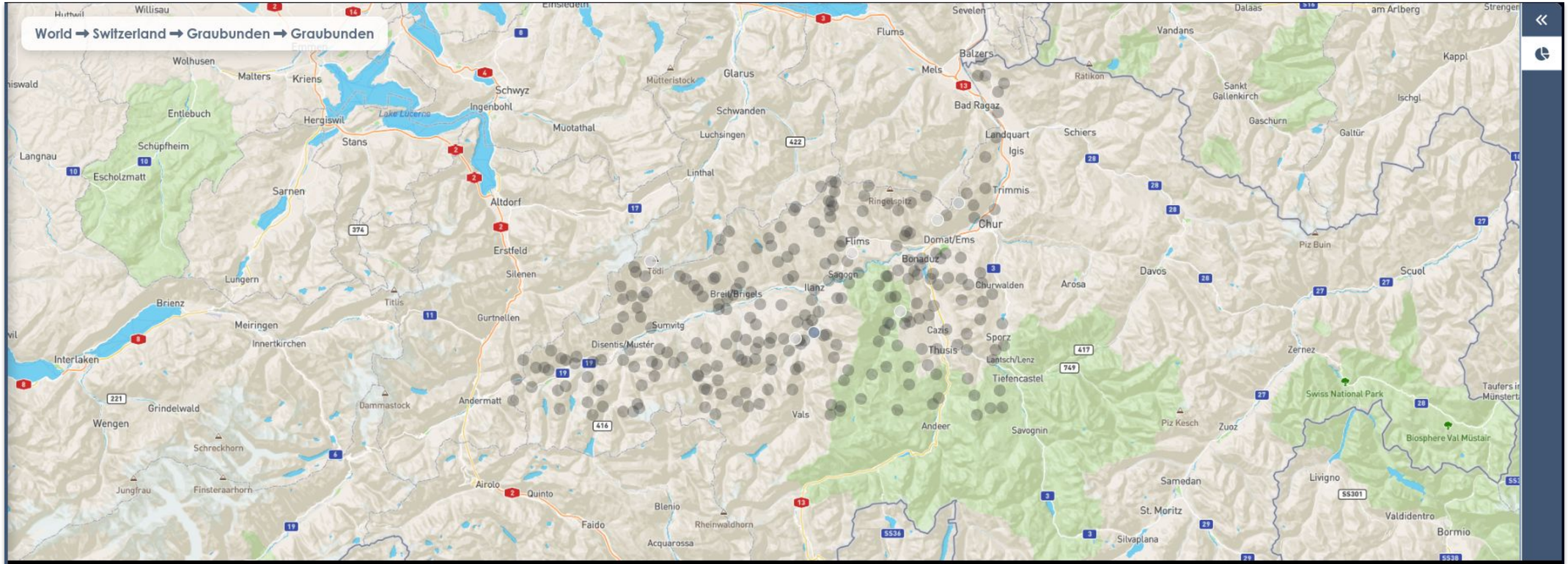


Swisstopo: A large GDI





Aggregate Data with a Geospatial Relation

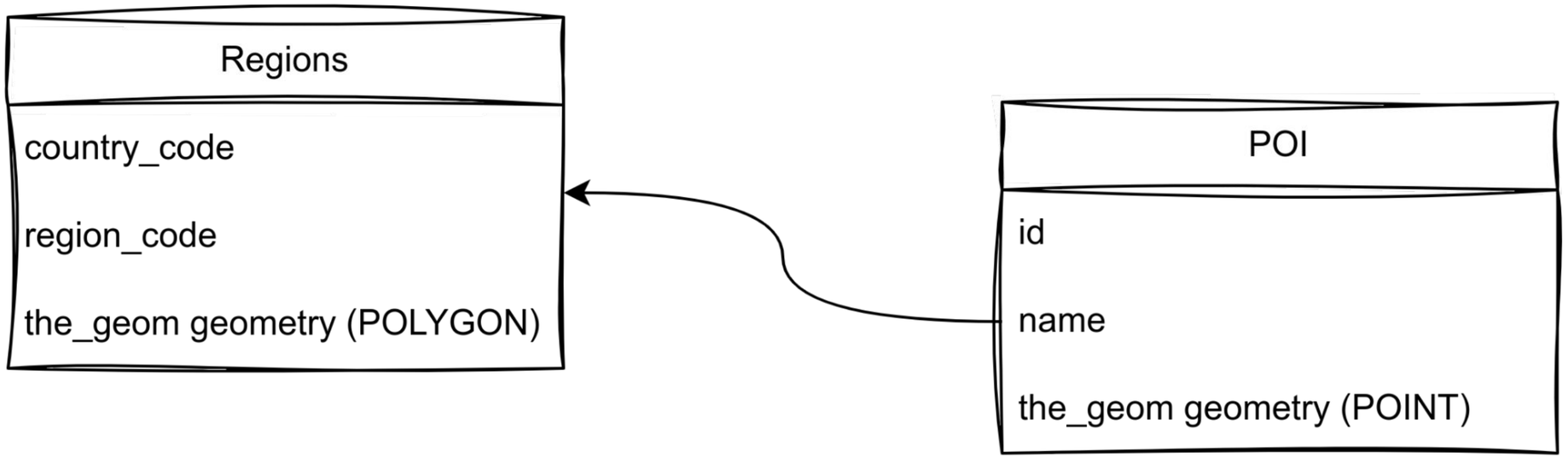


Outlet	ERP	GIS	Infrastructure	Other	Subscriptions	Total
Besnard et Fils	€2,490	€4,385	€223	€158	€4,292	€11,548
Martineau Guerin S.A.R.L.		€14,087		€600		€14,687
Meyer				€0		€0
Pinto Jourdan SARL				€0		€0
Raymond	€36,066	€61,190	€2,858	-€682	€28,743	€128,174
Schubert GbR	€35,256	€20,179	€11,307	€1,200	€149,137	€217,080
Ullrich Braun KG	€253,136	€575,992	€7,817	-€2,158	€201,008	€1,035,794

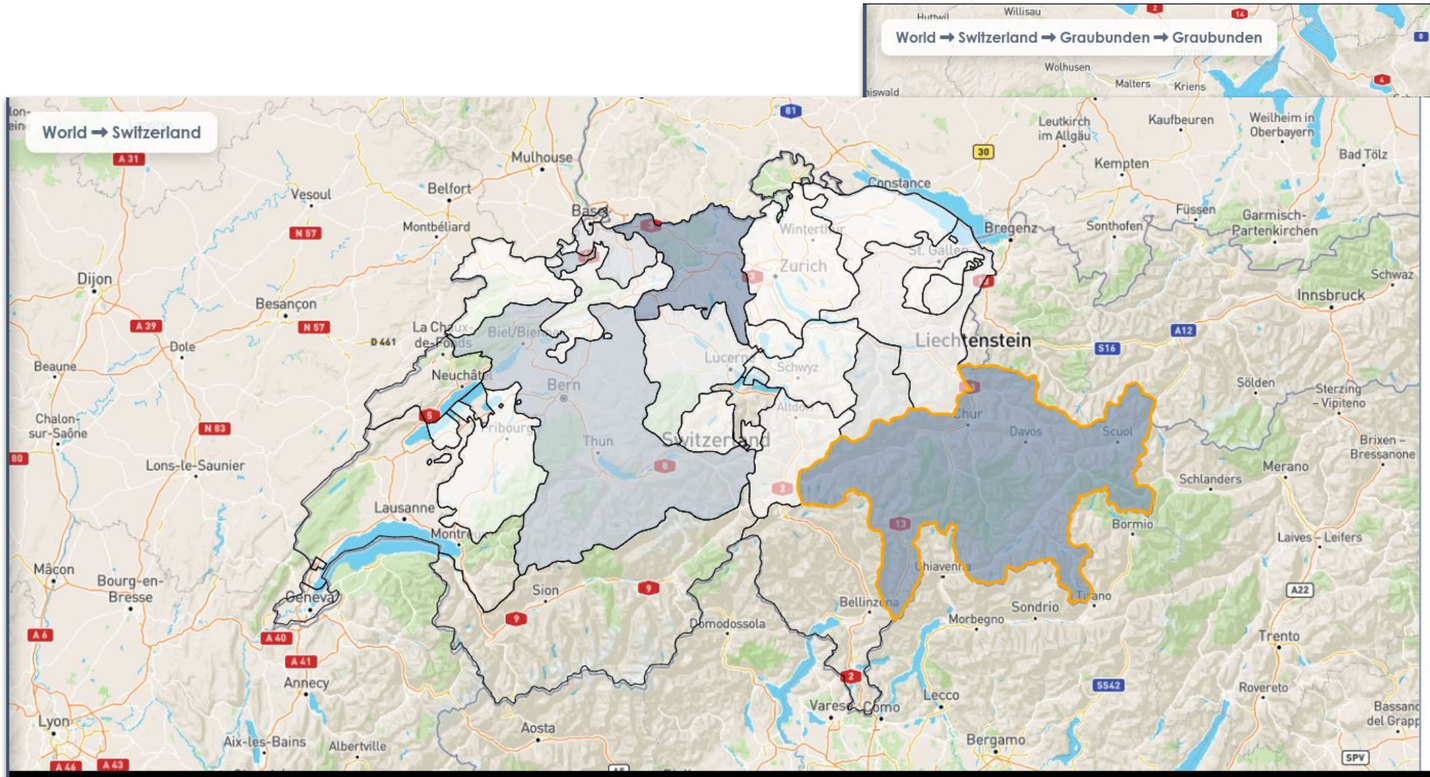
+ OpenStreetMap boundary data

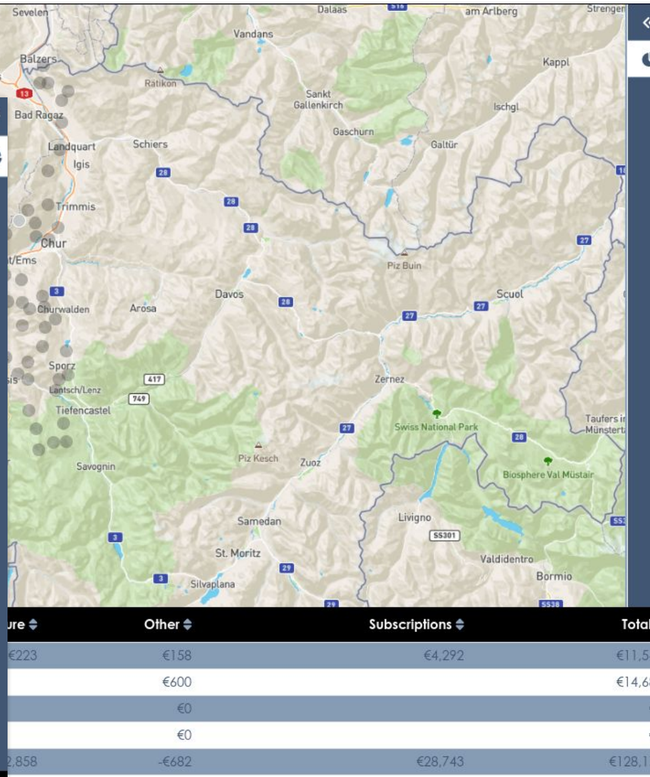
Aggregate Data with a Geospatial Relation

- Access data and styling through OpenLayers
- Homogeneous tables



Aggregate Data with a Geospatial Relation

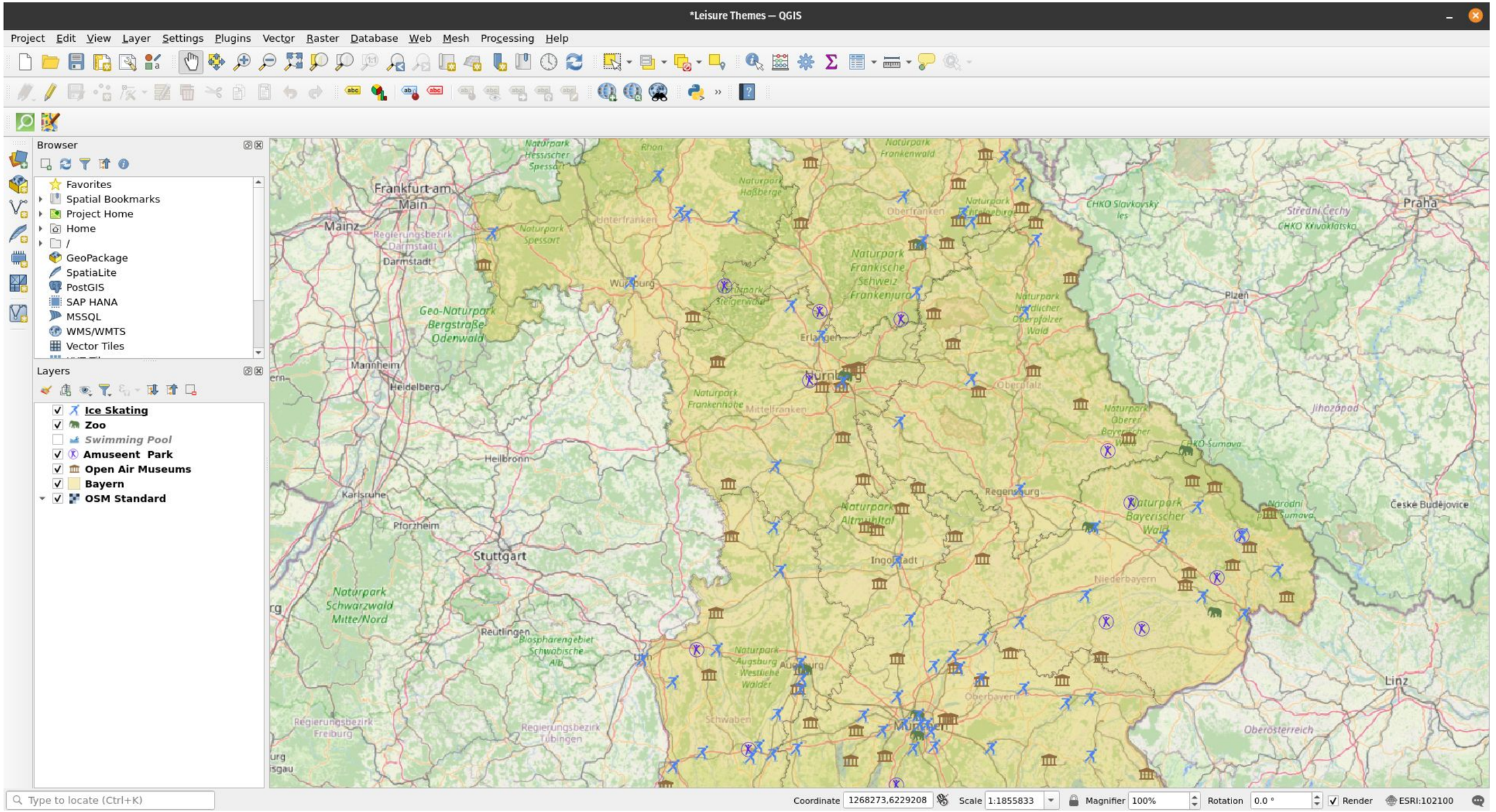




Region	ERP	GIS	Infrastructure	Other	Subscriptions	Total
Aargau	€214,333	€526,586	€49,056	€1,134	€533,980	€1,325,089
Appenzell Ausser-rhoden				€0		€0
Appenzell Inner-rhoden	€5,071	€6,383	€177	€0	€3,438	€15,068
Basel-landschaft	€60,784	€74,909	€14,107	€4,327	€174,303	€328,430
Bern	€140,097	€269,857	€11,771	€8,497	€224,359	€654,580
Fribourg	€2,962	€7,233	€1,038	€0	€8,703	€19,936
Glarus	€46,112	€75,455	€5,204	€431	€94,079	€221,281
Graubunden	€326,949	€675,832	€22,205	-€883	€383,179	€1,407,283
Jura	€1,079			-€1,079		€0

ure	Other	Subscriptions	Total
€223	€158	€4,292	€11,548
	€600		€14,687
	€0		€0
	€0		€0
2,858	-€682	€28,743	€128,174
1,307	€1,200	€149,137	€217,080
7,817	-€2,158	€201,008	€1,035,794

Data Analysis with Desktop Tools



camptocamp
INNOVATIVE SOLUTIONS
BY OPEN SOURCE EXPERTS

Thanks for your attention.

https://github.com/camptocamp

https://www.camptocamp.com

Marion Baumgartner
marion.baumgartner@camptocamp.com

Literature References

- PostGIS in Action by Regina O. Obe, Leo S. Hus
- https://postgis.net/docs
- https://www.postgis.net/workshops/postgis-intro/
- https://map.geo.admin.ch/
- https://www.swisstopo.admin.ch/de/swisstopo-in-zahlen

INNOVATIVE SOLUTIONS
BY OPEN SOURCE EXPERTS