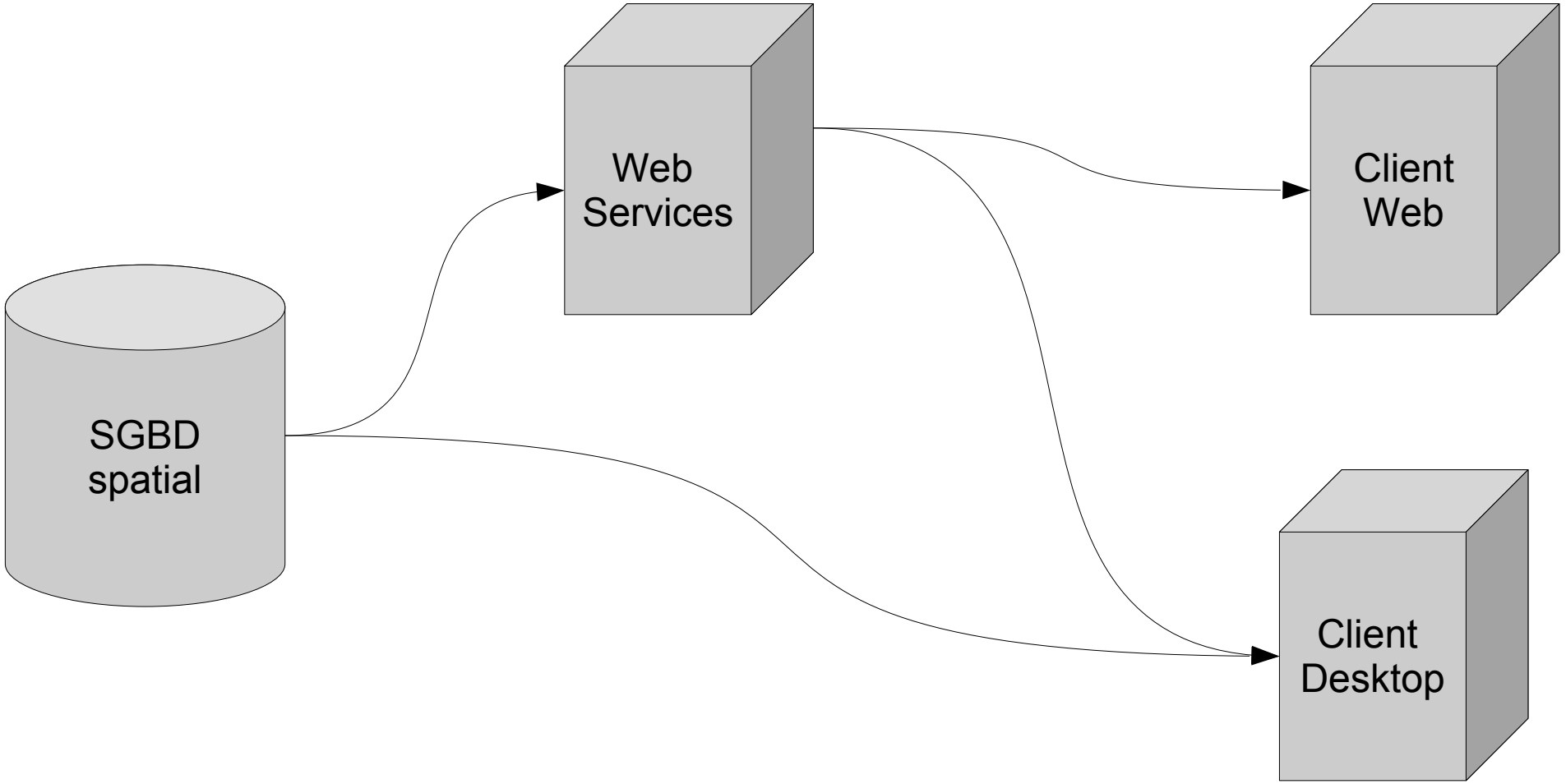
PostGIS & QGIS

Be-OpenGIS 2014 - Bruxelles

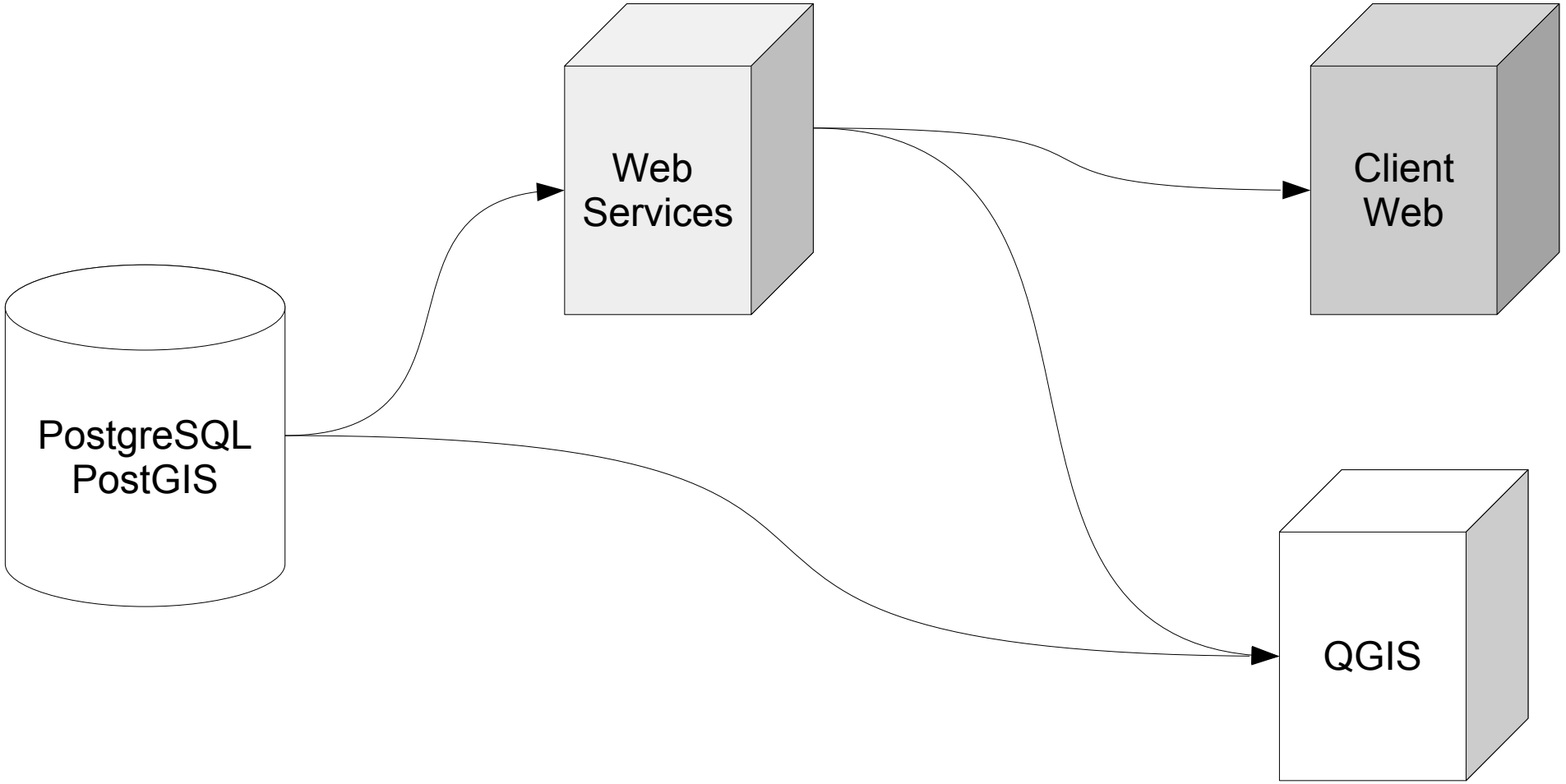
Olivier Courtin - Oslandia

Architecture SIG 'classique'



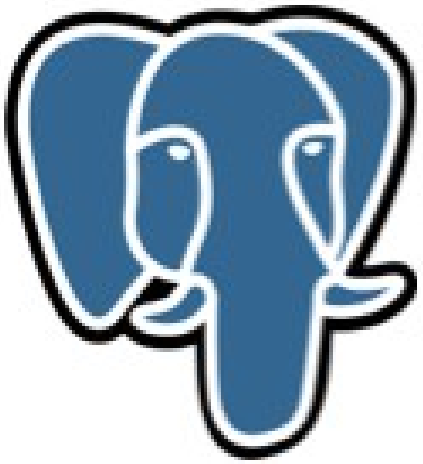
OSLANDIA

Architecture SIG 'classique'



OSLANDIA

Présentation PostGIS



PostgreSQL

+

PostGIS

OSLANDIA

Autres SGBD spatiaux

Oracle Spatial (et Locator)

IBM DB2

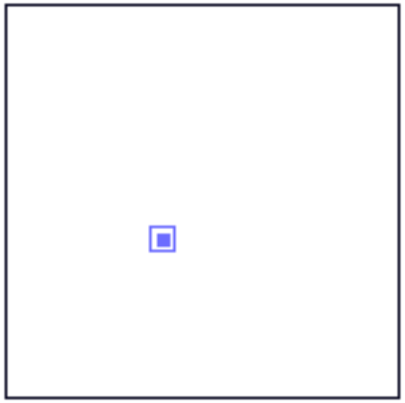
Microsoft SQLServer 2008

SpatiaLite

OSLANDIA

Géométries: Point

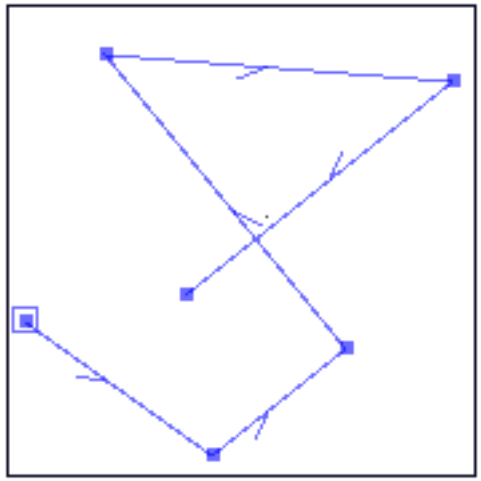
```
POINT (10 10)
```



OSLANDIA

Géométries: LineString

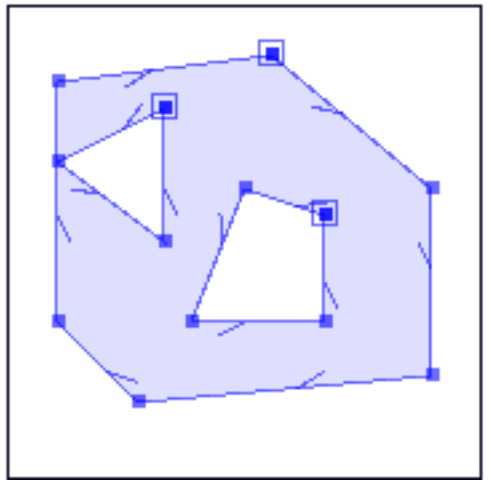
```
LINESTRING
 (
  0 5, 5 1, 9 4, 2 14, 14 13, 4 4
 )
```



OSLANDIA

Géométries: Polygon

```
POLYGON
(
 (9 13,13 9,13 3,4 2,1 4,1 12, 9 13),
 (5 11,5 6,1 9,5 11),
 (10 7, 10 4, 6 4, 8 8, 10 7)
)
```



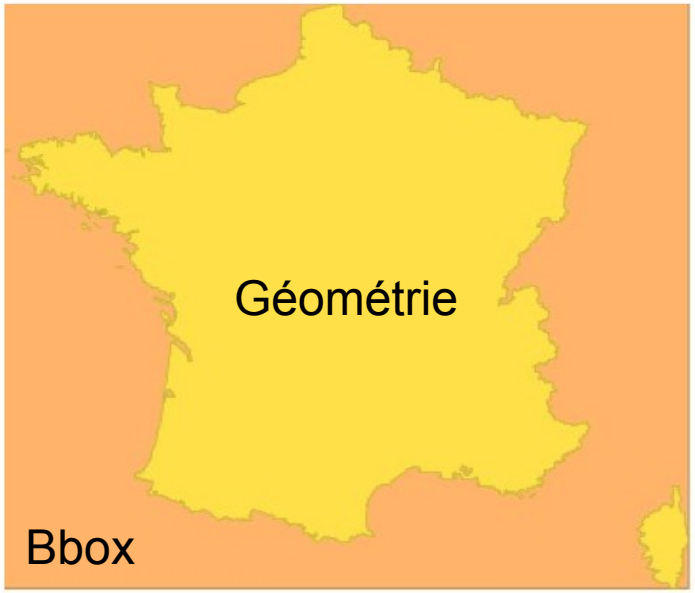
1) Le premier ring (obligatoire) correspond au ring externe
2) Les coordonnées des rings sont fermantes
3) Les rings suivants (optionels) correspondent à des 'trous'

OSLANDIA

Index spatiaux: Principe et création

Améliorer performances sur filtrage

Approxime les géométries: Bbox

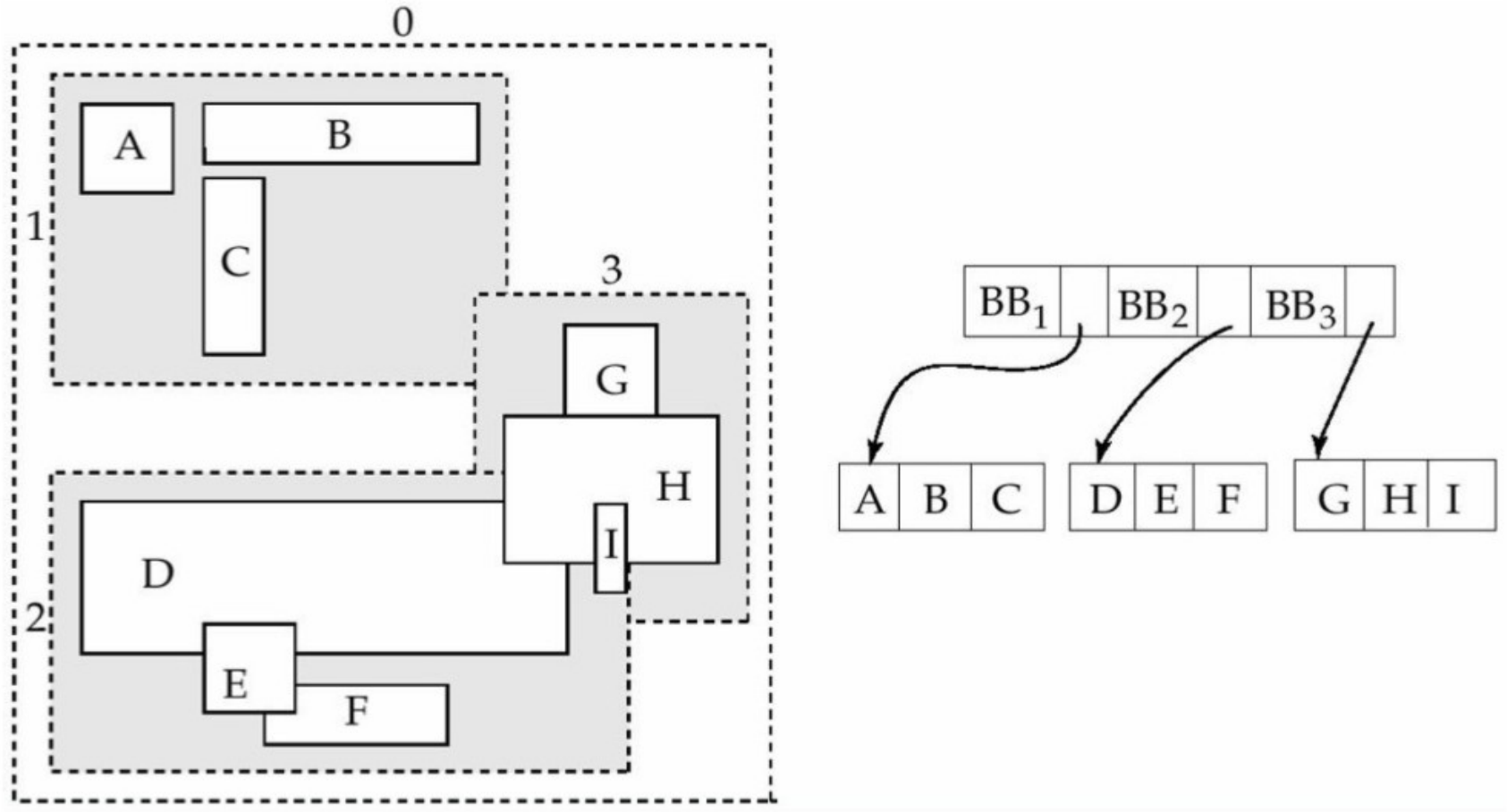


Création d'un index spatial:

```
CREATE INDEX ON table_name
USING GIST(geom_column_name);
```

OSLANDIA

Index spatiaux: R-Tree

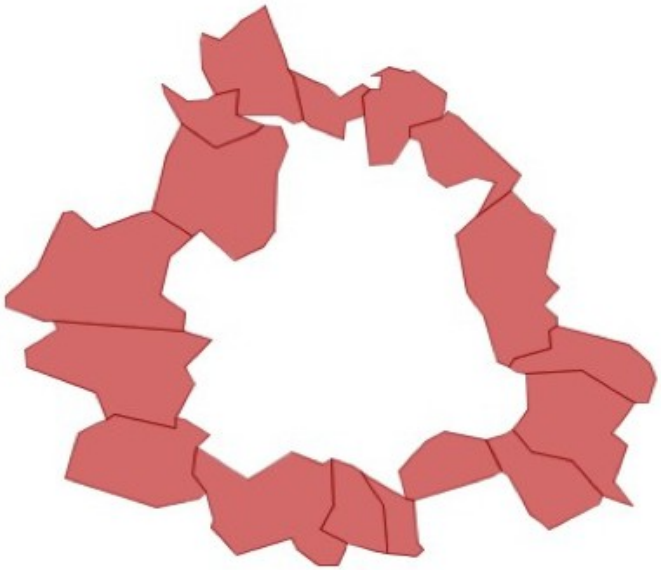


Regroupement des Bbox dans des régions de l'index

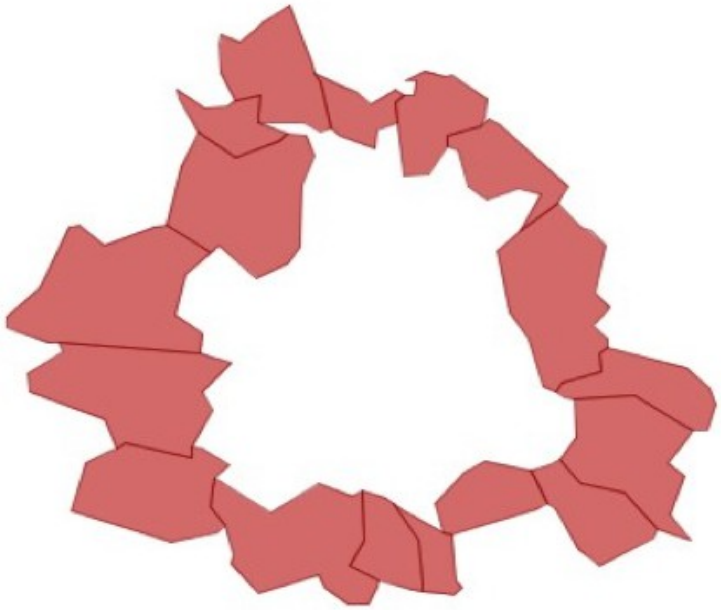
OSLANDIA

Index spatiaux

```
SELECT c1.nom FROM communes c1, communes c2
WHERE c2.nom = 'Toulouse'
AND ST_Touches(c1.geom, c2.geom);
```



Sans index: temps = 150 ms

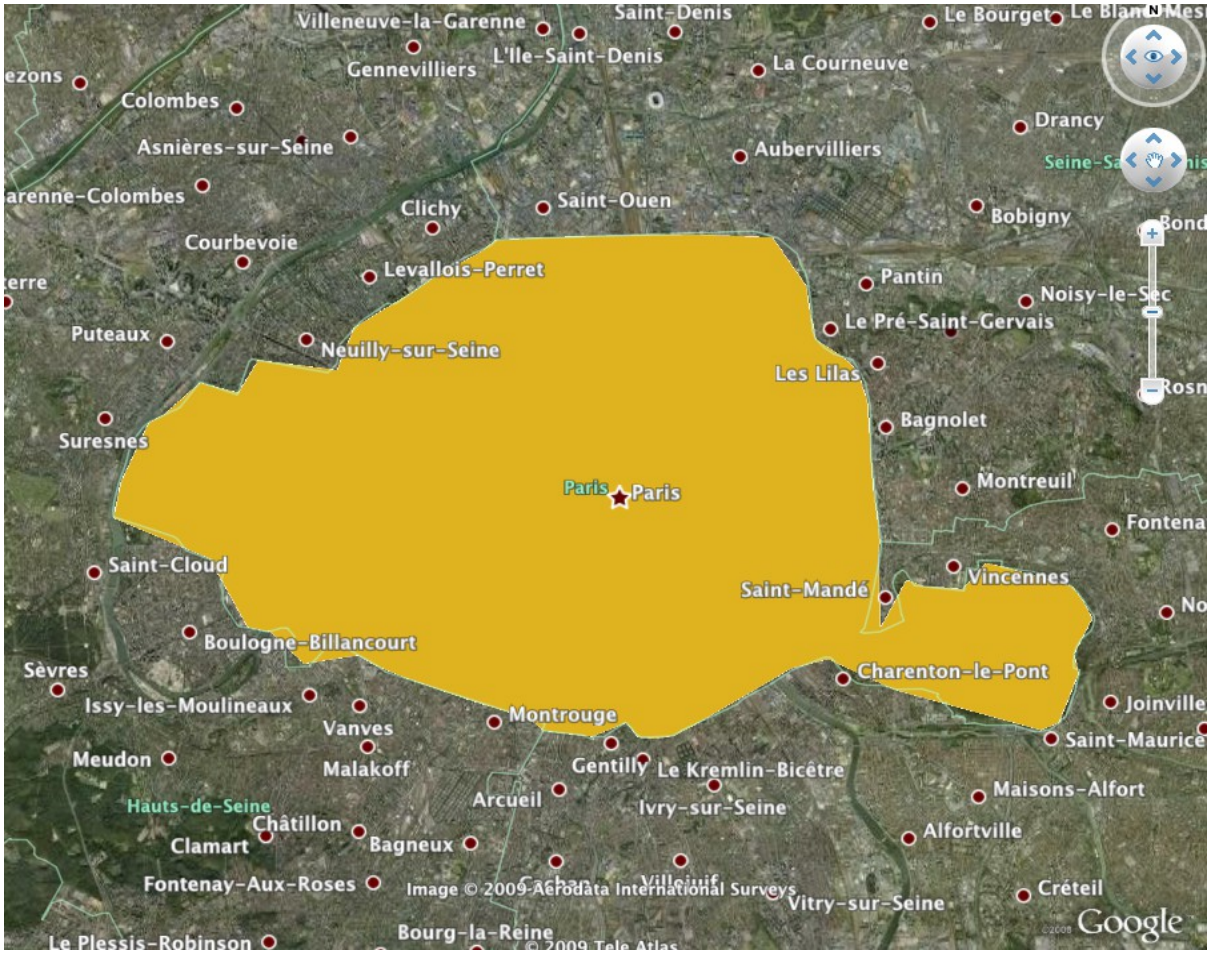


Avec index: temps = 30 ms

OSLANDIA

Export KML

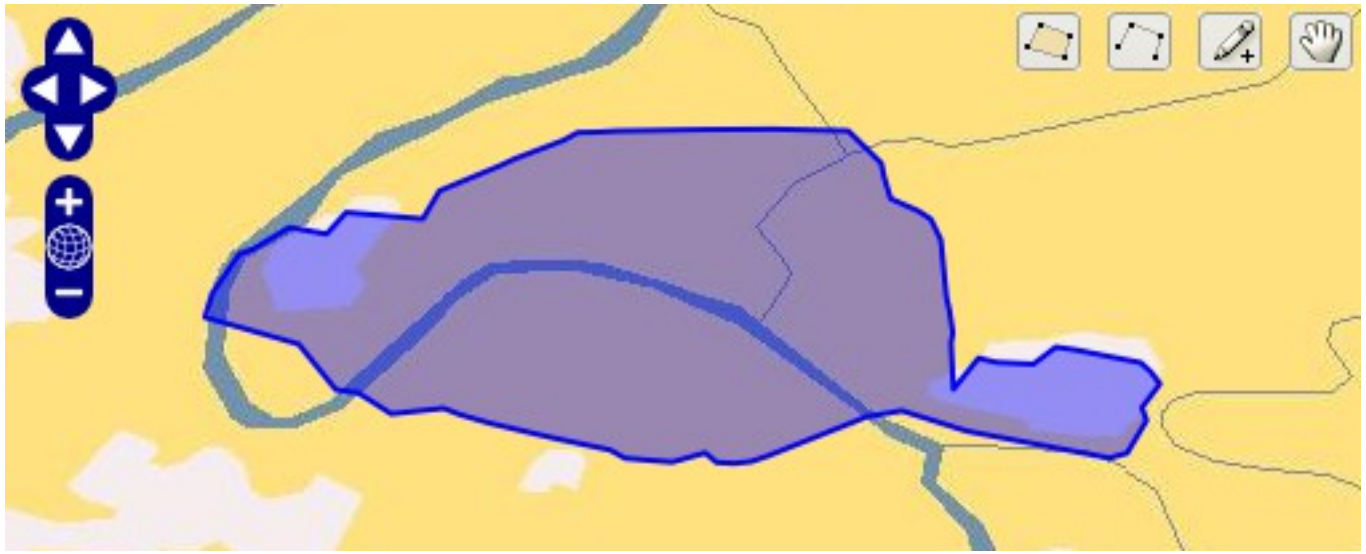
```
SELECT ST_AsKML(geom, 5)
FROM dept
WHERE code_dept='75' ;
```



OSLANDIA

Export GeoJSON

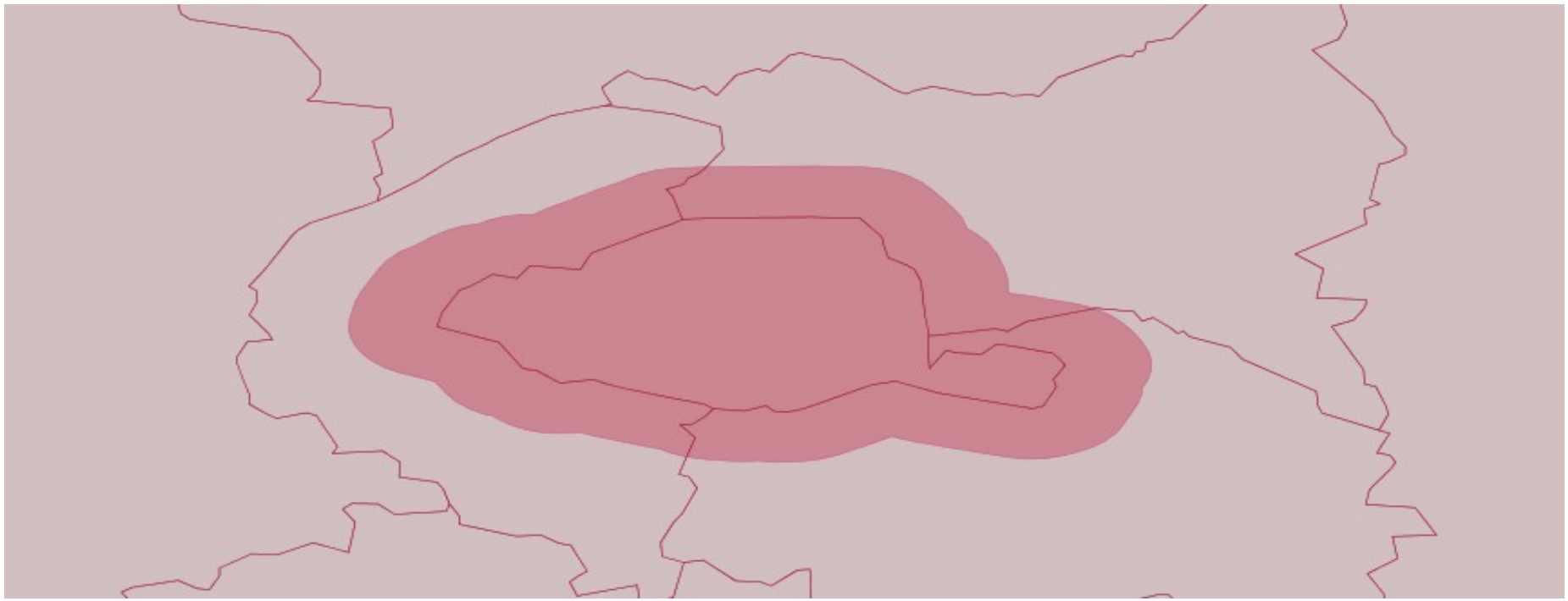
```
SELECT ST_AsGeoJSON(
            ST_Transform(geom, 4326), 5
        ) FROM dept
WHERE code_dept='75' ;
```



OSLANDIA

ST_Buffer

```
SELECT ST_Buffer(geom, 2500)
FROM dept
WHERE code_dept='75';
```



OSLANDIA

Aggrégation de géométries

Les communes de France

Les communes de France
fusionnées par département

```
SELECT ST_Union(geom)
FROM commune
GROUP BY code_dept;
```

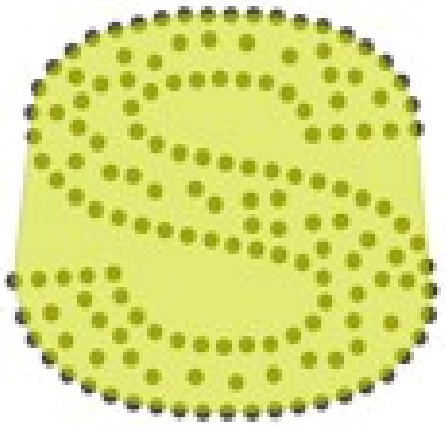
OSLANDIA

Intersection Spatiale

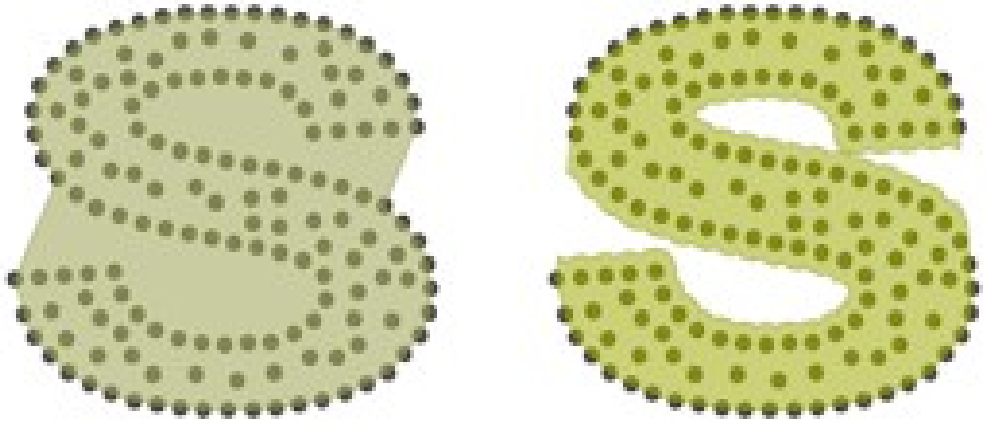
```
WITH paris AS
  (SELECT the_geom
   FROM communes
   WHERE nom='Paris')


SELECT nom FROM communes c, paris p
WHERE c.geom && p.geom
AND ST_Intersects(c.geom, p.geom);
```

OSLANDIA

ST_ConvexHull et ST_ConcaveHull



ST_ConvexHull



ST_ConcaveHull

OSLANDIA

Plus Proches Voisins (KNN)

```
SELECT nom, gid FROM geonames
ORDER BY
 geom <-> 'SRID=4326;POINT(-90 40)'::geometry

LIMIT 10;
```

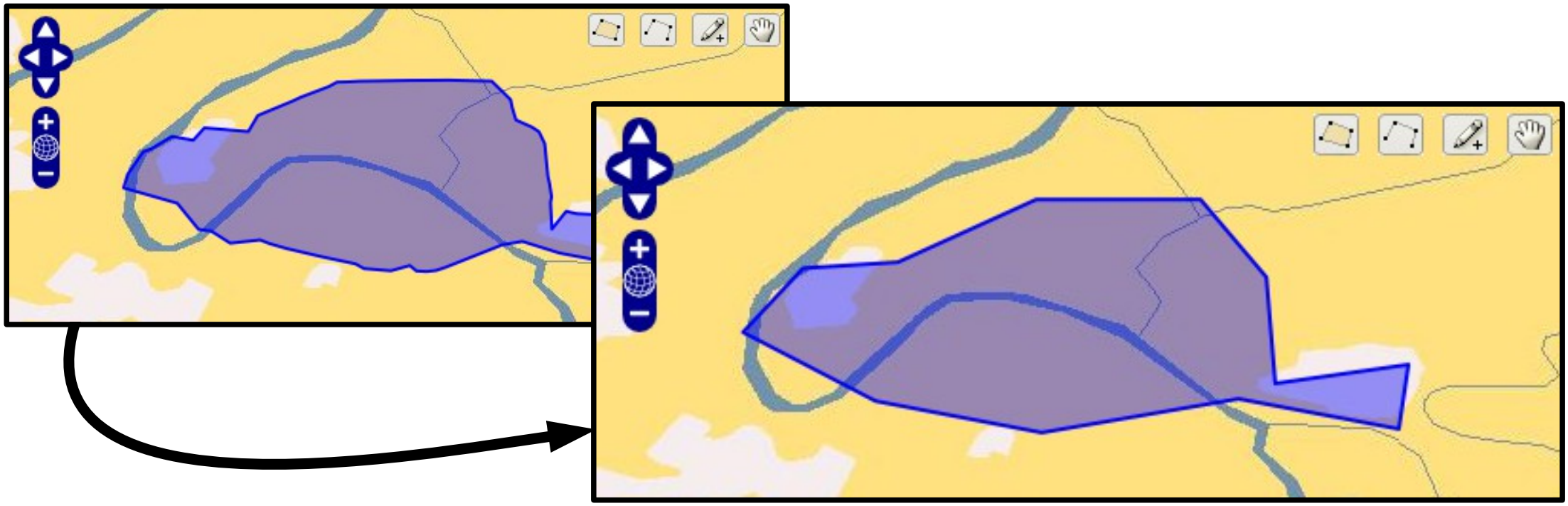
Operateur de distance: <-> or <#> (center ou bbox)

OSLANDIA

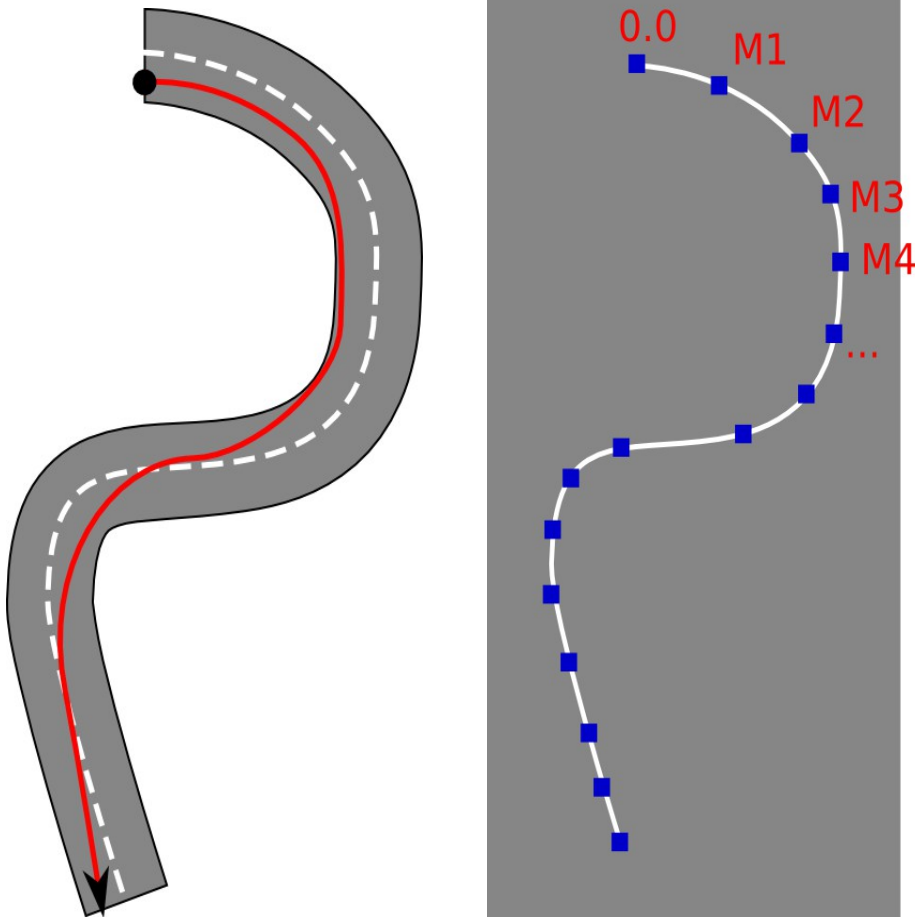
Généralisation via ST_Simplify

Algorithme Douglas-Peuker

```
SELECT ST_AsGeoJSON(
            ST_Transform(
               ST_Simplify(geom, 800),
            4326), 5)
FROM dept WHERE code_dept='75';
```



Référencement Linéaire (LRS)

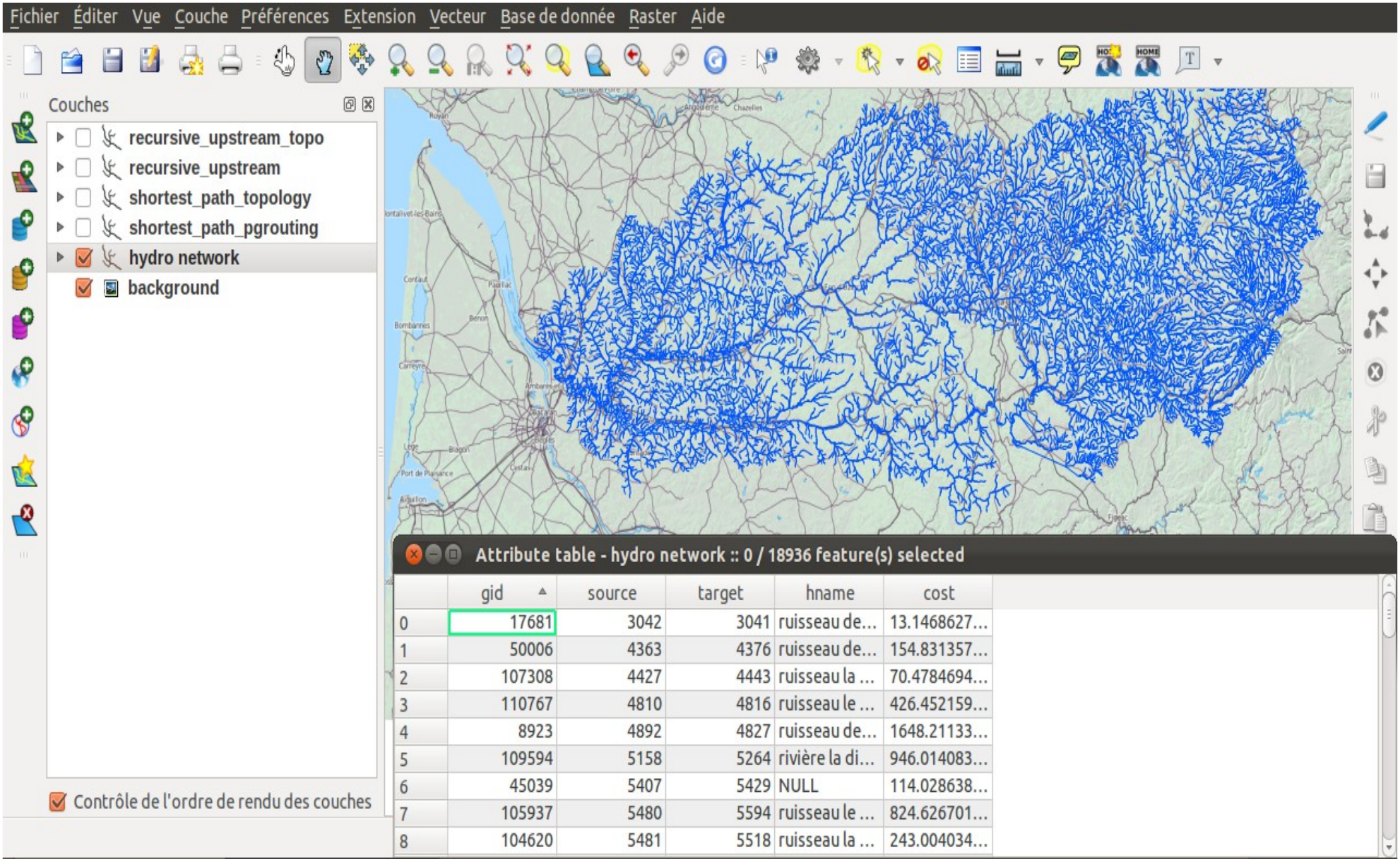


ST_Line_Interpolate_Point(linestring, location)

ST_Line_Locate_Point(LineString, Point)

OSLANDIA

Topology



	gid	source	target	hname	cost
0	17681	3042	3041	ruisseau de...	13.1468627...
1	50006	4363	4376	ruisseau de...	154.831357...
2	107308	4427	4443	ruisseau la ...	70.4784694...
3	110767	4810	4816	ruisseau le ...	426.452159...
4	8923	4892	4827	ruisseau de...	1648.21133...
5	109594	5158	5264	rivière la di...	946.014083...
6	45039	5407	5429	NULL	114.028638...
7	105937	5480	5594	ruisseau le ...	824.626701...
8	104620	5481	5518	ruisseau la ...	243.004034...

Topology

Recursive CTE

```
create table
        rec_res2 as
with recursive
        search_graph(edge_id, start_node, depth, path, length, cycle) as (
        select
                g.edge_id, g.start_node, 1 as depth, ARRAY[g.edge_id] as path
                , st_length(geom) as length, false as cycle
        from
                hydro.edge as g
        where
                edge_id = 173832
        union all
        select
                g.edge_id
                , g.start_node
                , sg.depth + 1 as depth
                , path || g.edge_id as path
                , sg.length + st_length(g.geom) as length
                , g.edge_id = ANY(path) as cycle
        from
                hydro.edge as g
        join
                search_graph as sg
        on
                sg.start_node = g.end_node
        where
                not cycle
        )
```

1

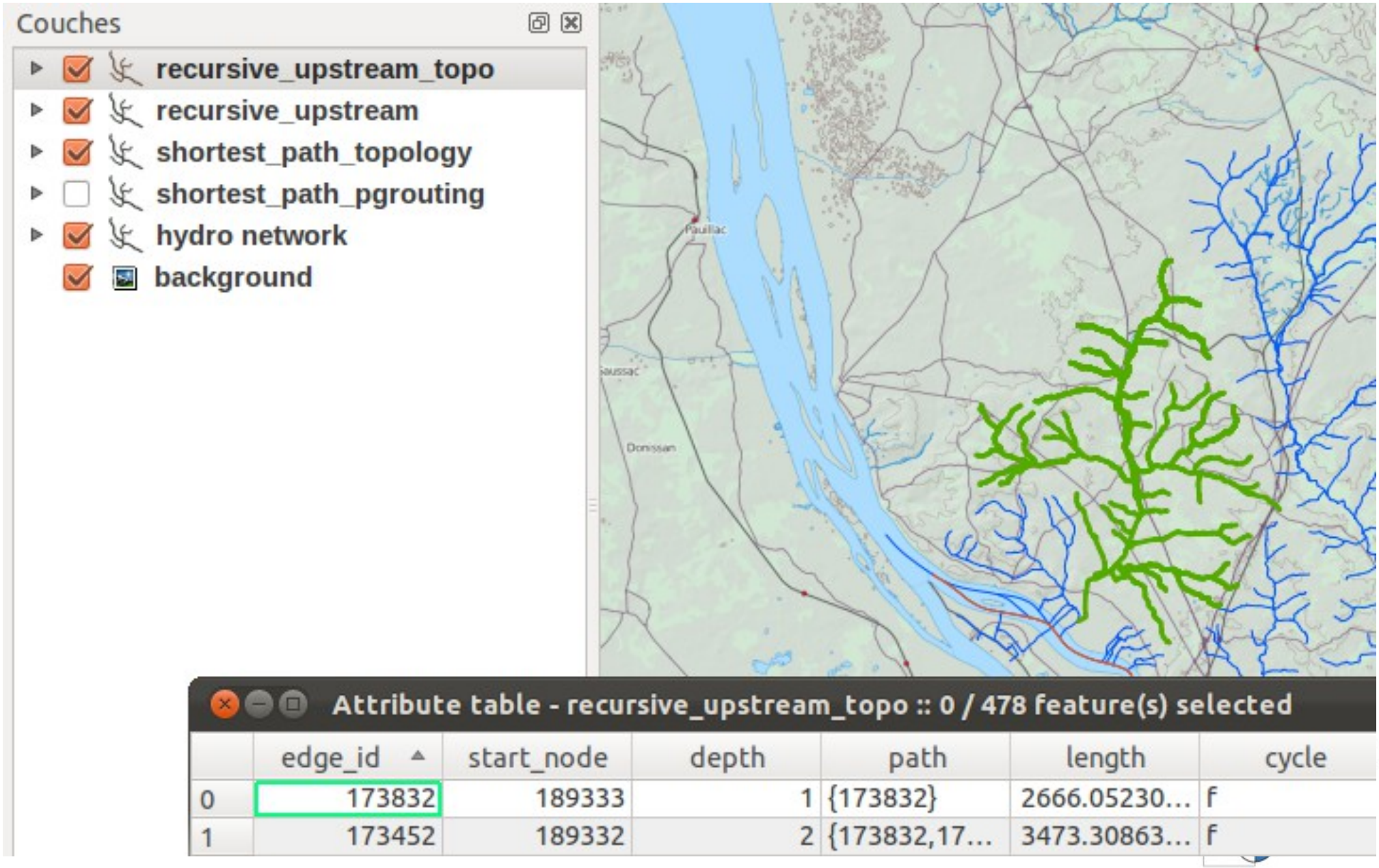
2

```
select
        sg.*
        , edge.geom as geom
from
        search_graph as sg
join
        hydro.edge as edge
on
        sg.edge_id = edge.edge_id
limit 1000;
```

3

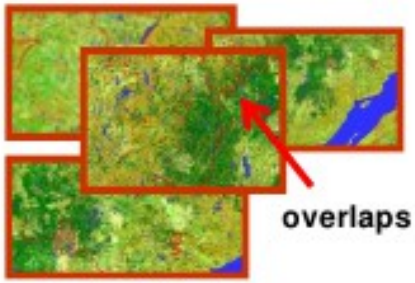
A

Topology

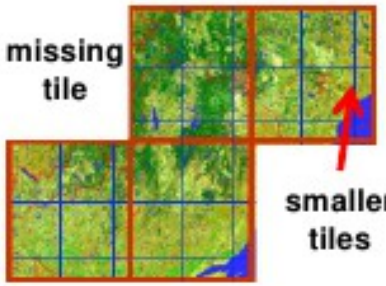

Couches
recursive_upstream_topo
recursive_upstream
shortest_path_topology
shortest_path_pgrouting
hydro network
background
Pauillac
Donissan
Attribute table - recursive_upstream_topo :: 0 / 478 feature(s) selected
edge_id
start_node
depth
path
length
cycle
0
173832
189333
1
{173832}
2666.05230…
f
1
173452
189332
2
{173832,17…
3473.30863…
f

A

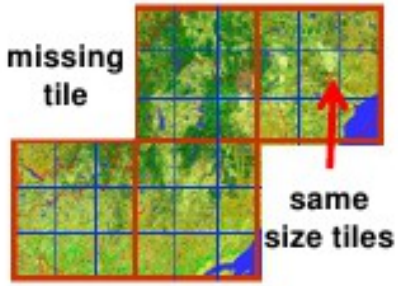
PostGIS Raster



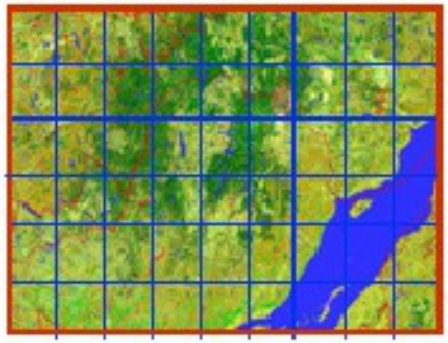
a) warehouse of untiled and unrelated images (4 images)



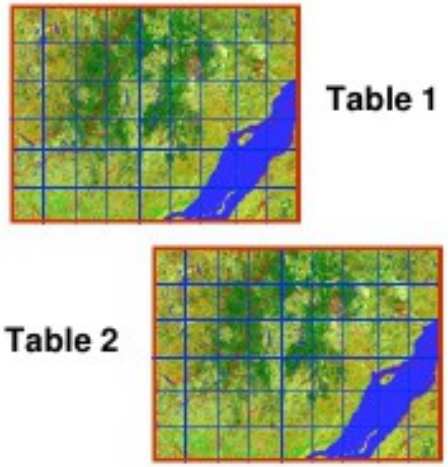
b) irregularly tiled raster coverage (36 tiles)



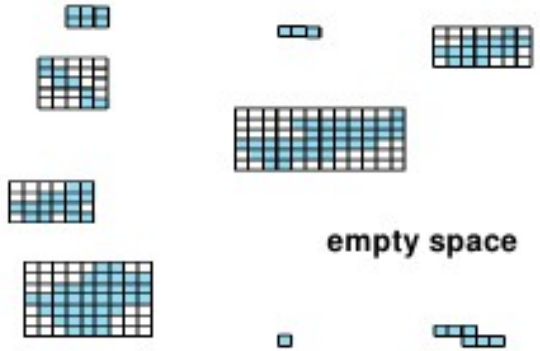
c) regularly tiled raster coverage (36 tiles)



d) rectangular regularly tiled raster coverage (54 tiles)



e) tiled images (2 tables of 54 tiles)

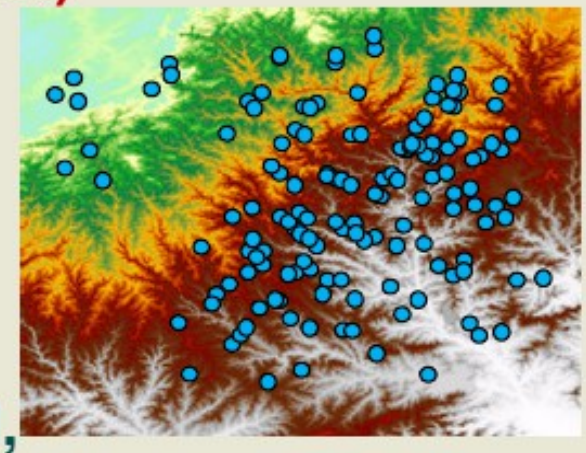


f) rasterized geometries coverage (9 lines in the table)

PostGIS Raster

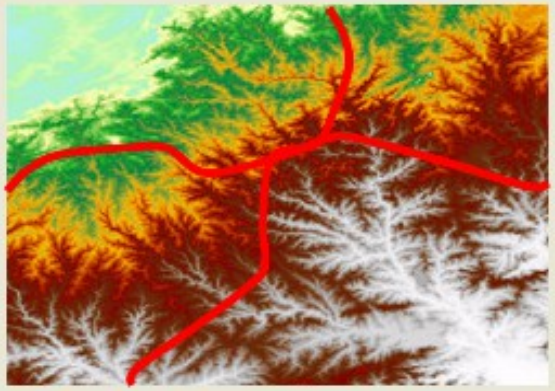
Extract ground elevation values for lidar points…

- SELECT pointID, ST_Value(rast, geom) elevation
  FROM lidar, srtm WHERE ST_Intersects(geom, rast)



Intersect a road network to extract elevation values for each road segment

- SELECT roadID,
  (ST_Intersection(geom, rast)).geom road,
  (ST_Intersection(geom, rast)).val elevation
  FROM roadNetwork, srtm WHERE ST_Intersects(geom, rast)

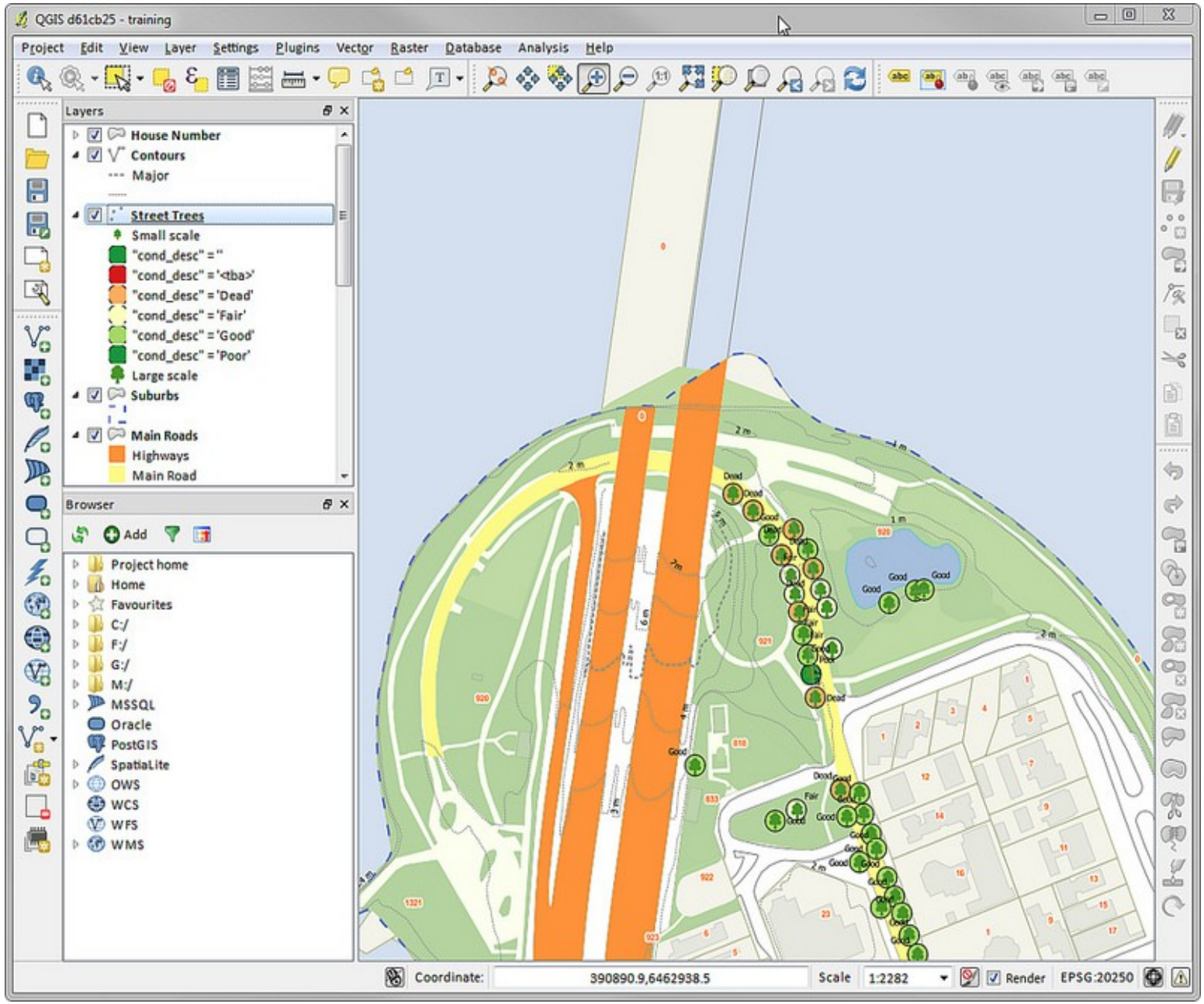


PostGIS 3D

http://vimeo.com/74869530

OSLANDIA

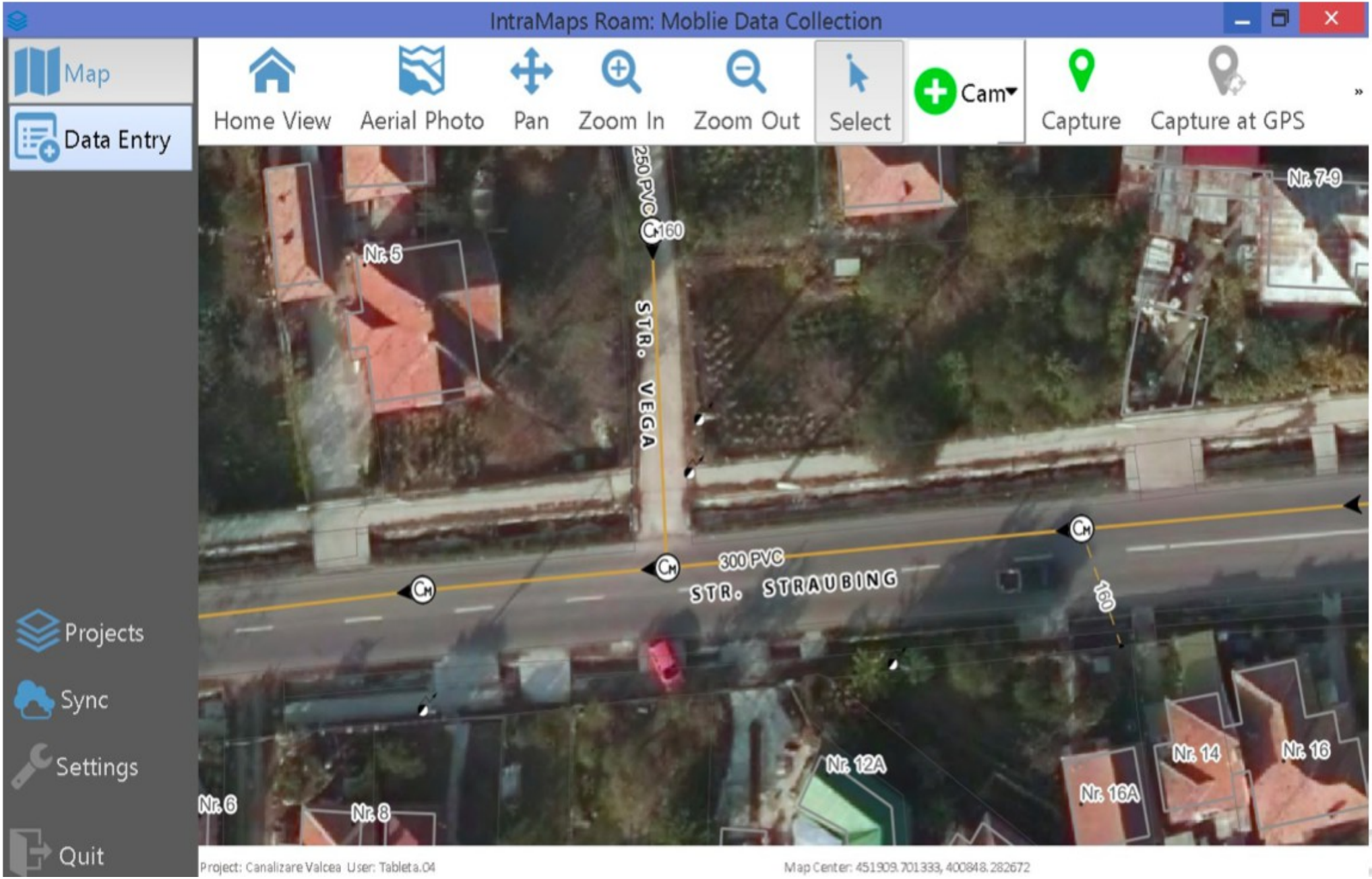
Présentation QGIS



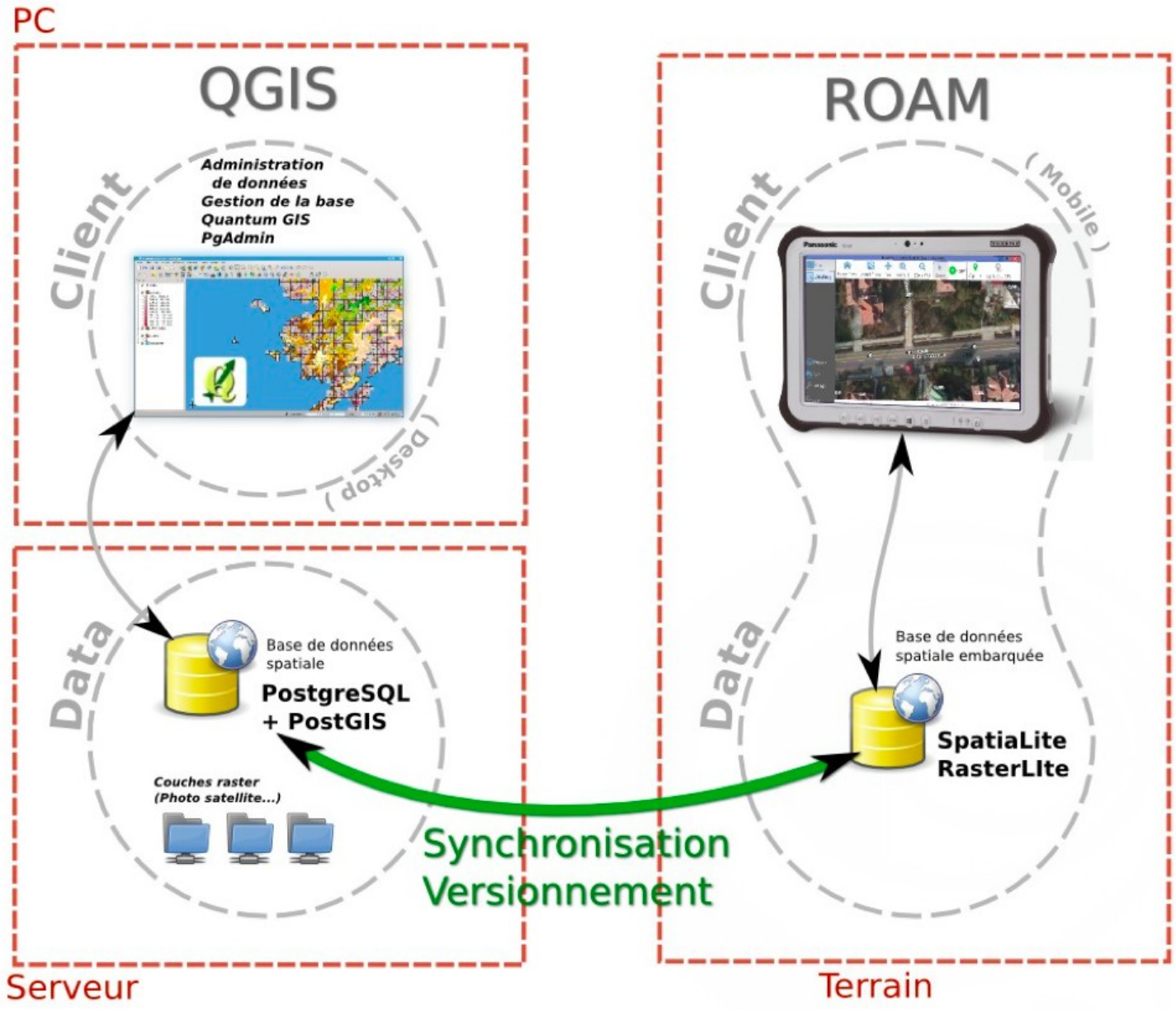
OSLANDIA


Panasonic
TOUGHPAD

QGIS IHM minimaliste métier embarquée

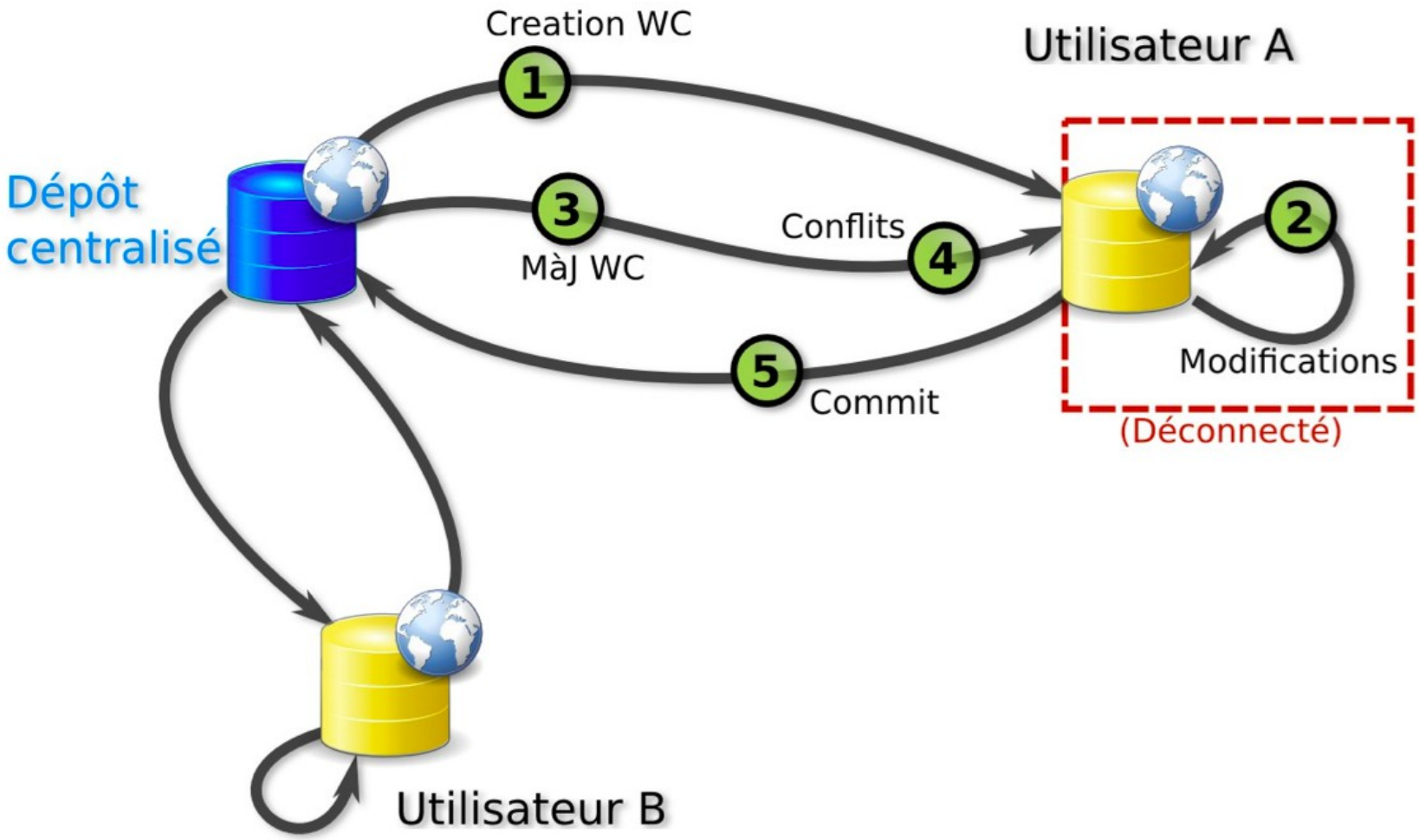


QGIS + PostGIS : Architecture Embarquée



OSLANDIA

QGIS + PostGIS: Gestion de versions et de conflits

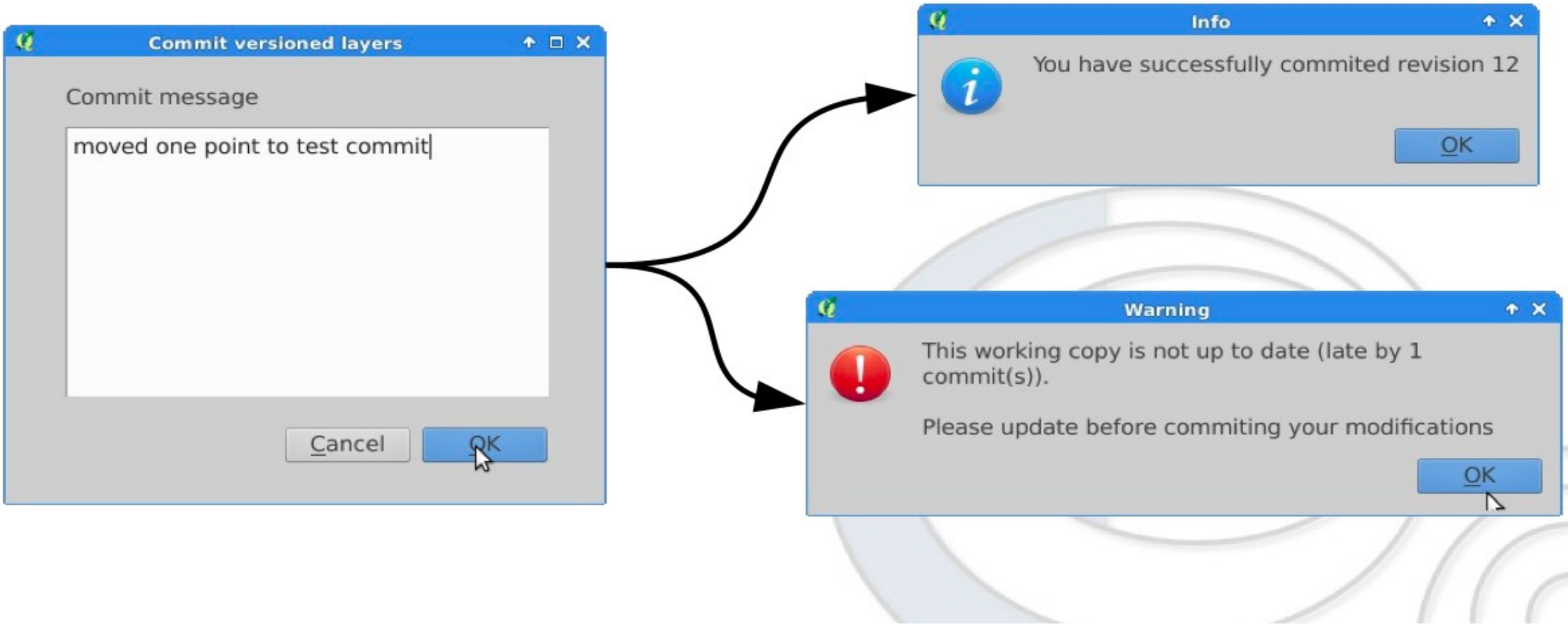


OSLANDIA

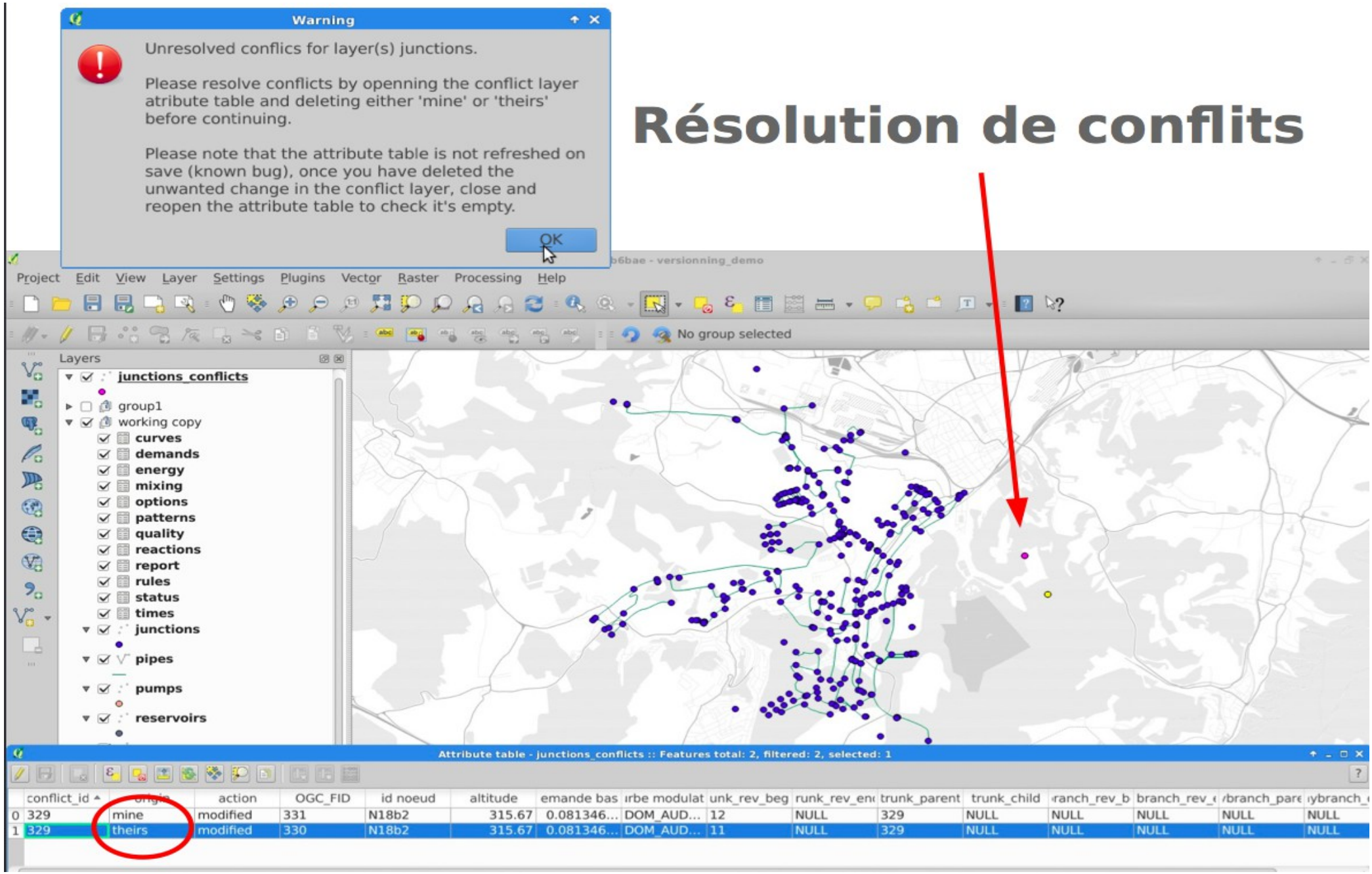
QGIS + PostGIS: Gestion de versions et de conflits

Commit

- sur WC à jour
- insertion des éléments dans les tables
- métadonnées de révision



QGIS + PostGIS: Gestion de versions et de conflits

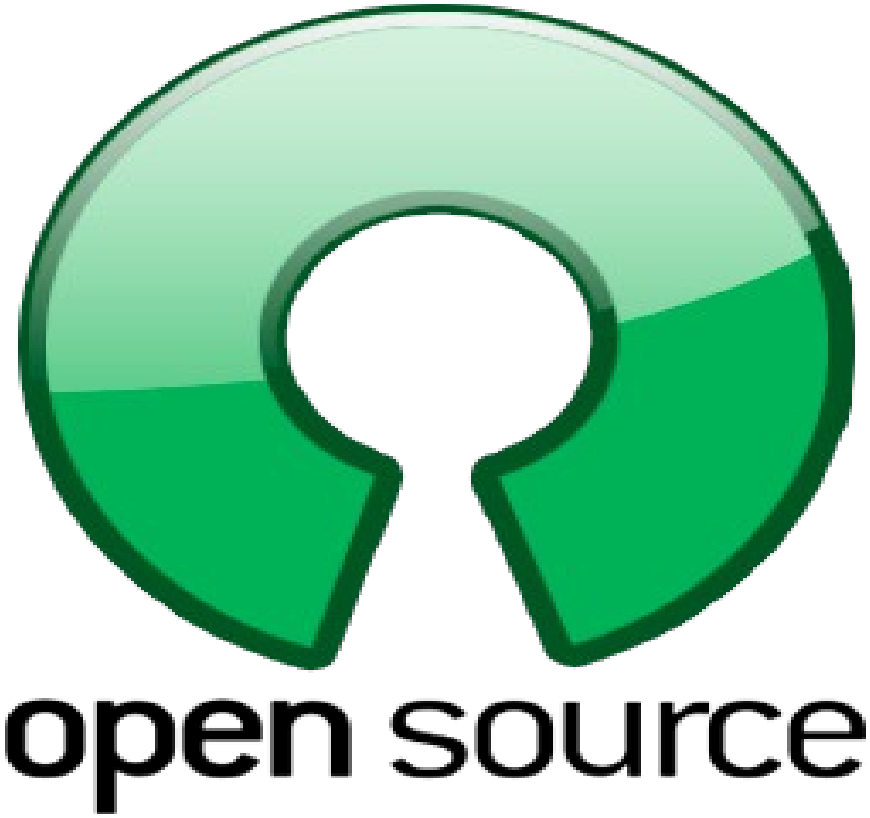


QGIS avec plugin et librairie métier, EPANET

http://vimeo.com/87754967

OSLANDIA

Pourquoi Utiliser de l'Open Source en SIG ?



OSLANDIA

www.oslandia.com