

# PostGIS 2.0 : Géo nouvelle génération



**Vincent Picavet – Oslandia**

# Oslandia ?

- TPE française
- Spécialistes SIG Open Source
- PostGIS ( committers )
- BdD spatiales, OGC, analyse complexe...
- Quantum GIS
- Mapserver, Mapnik, TinyOWS...
- Recherche & Développement



# **Systemes d'Information Géographiques**

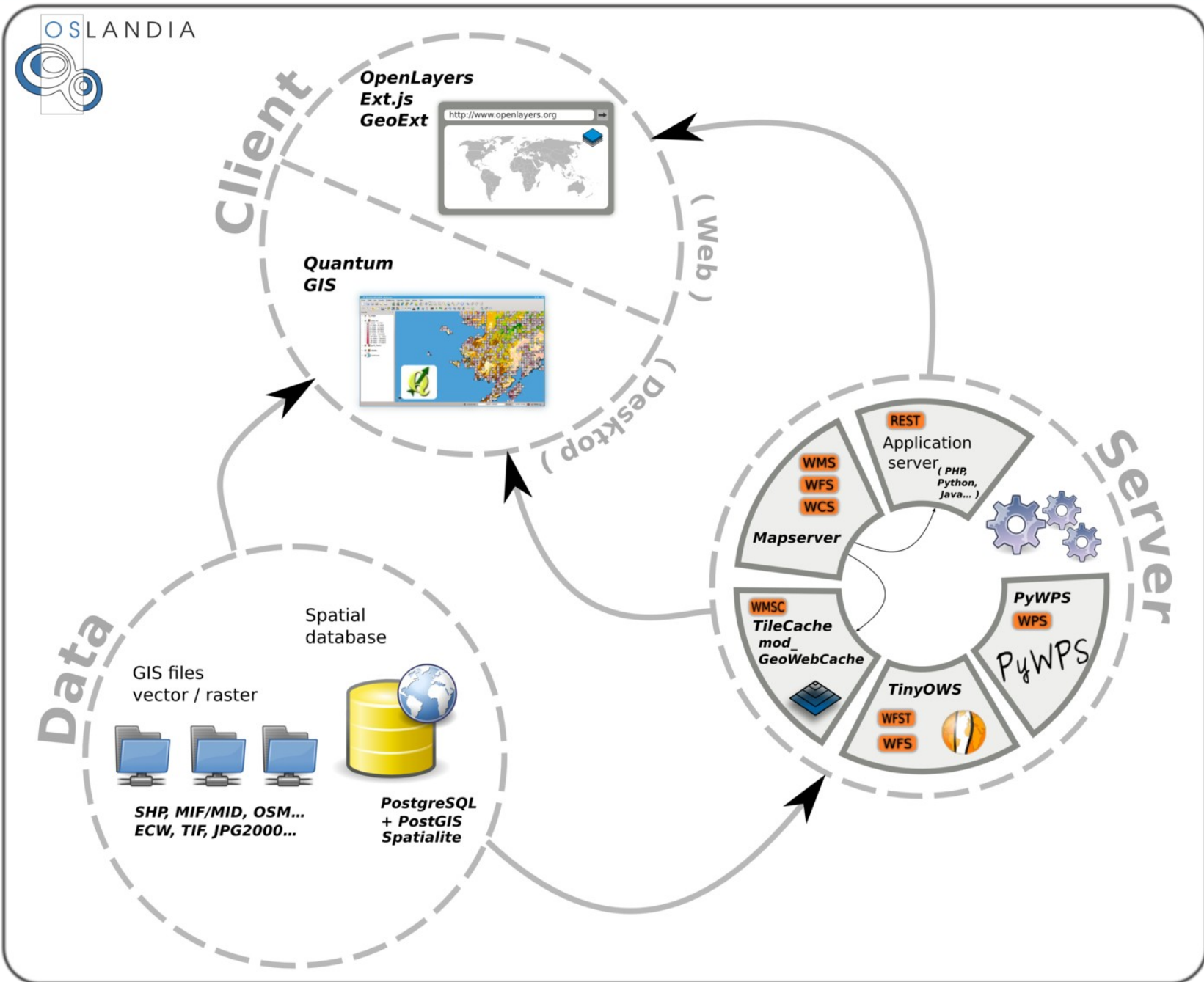


# Système d'Information Géographique

- 80% de l'information est spatiale
- Système d'information géographique
  - Capturer
  - Stocker
  - Échanger
  - Analyser
  - Visualiser
- Historiquement : collectivités
- Désormais : public, privé, tous



# SIG



# PostGIS : SGBDR spatial

- Geometrie + attributs = «feature»
- Requêtes SQL
  - Attributaires
  - Spatiales
- Forte charge
  - Données volumineuses
  - Traitements longs et complexes
- Performance acceptables
- Respect des standards



# PostGIS - Le projet



# PostGIS : standards

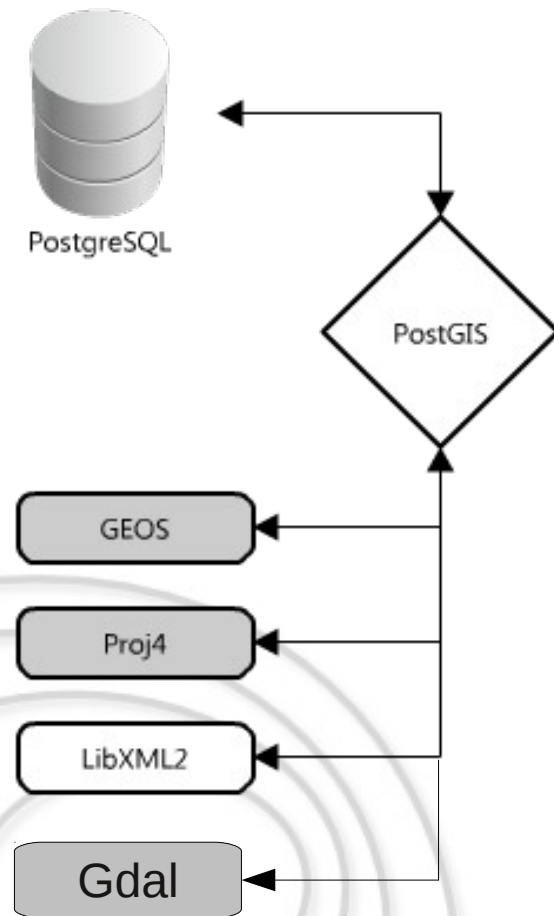
- Standards internationaux
- Spécifications
  - OGC SFS (Simple Feature for SQL)
  - ISO SQL/MM part 3
- Y sont définis :
  - Types Geometry
  - Prototypes de fonctions spatiales
  - Tables et autres pour intégrité référentielle





# PostGIS : Généralités

- Plugin PostgreSQL
- Écrit en C
- Basé sur des bibliothèques :
- Implémentation OGC SFS 1.1
  - et ISO SQL/MM
- Beaucoup d'autres choses



# PostGIS : historique

2001

Première version alpha

2003

**version 0.8** – valide pour production

2005

**version 1.0**

- Réécriture du cœur, LWGEOM
- Compatibilité OGC SFS 1.1

2006

**version 1.2**

- Direction ISO SQL/MM (curves, prefixe ST\_ ...)

2009

**version 1.4**

- PSC et incubation OSGeo

2011

**version 1.5.3**

- Geodésie

2012

**version 2.0**

- Projet OSGeo

# PostGIS : Utilisateurs

- Public
  - IGN : Institut géographique national
  - IRSN : Institut de Radioprotection et de sûreté Nucléaire
  - EEA : European Environment Agency
  - NOAA ...
- Privé
  - France Telecom, Infoterra, Digital Globe, Mediapost, NY Times...
  - Beaucoup d'autres

# PostGIS : communauté

- Mondiale
- Des milliers d'utilisateurs
- ML postgis-users très active



## Committers :

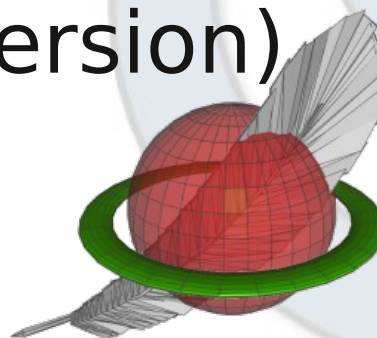
LisaSoft  
OpenGeo  
Oslandia  
CadCorp  
Paragon Corporation  
Refractions Research  
Sandro Santilli  
Sirius  
Some others and  
individuals...

# PostGIS : alternatives ?

- Oracle spatial (et locator)
- ESRI ArcSDE
- IBM DB2
- MS SQL Server (>2008)
- Actian
- Spatialite
- Sybase (dernière version)



Microsoft®  
SQL Server® 2008



# PostGIS - les bases



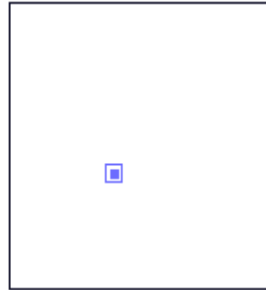
# Géométries : représentation, stockage

- Geometry (ou HEWKB)
  - Stockage natif en base
  - Format binaire encodage hexadécimal
- WKT (Well Known Text)
  - Représentation textuelle
- Dimensions
  - 2D, 3D, ou 4D
- Identifiant de projection (SRID)

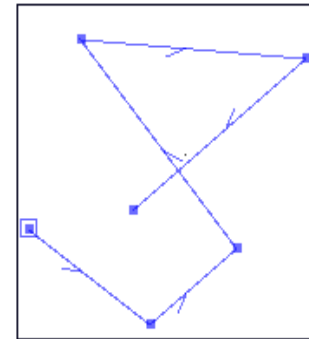


# Géometries

POINT (10 10)

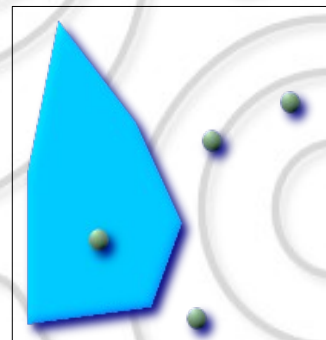
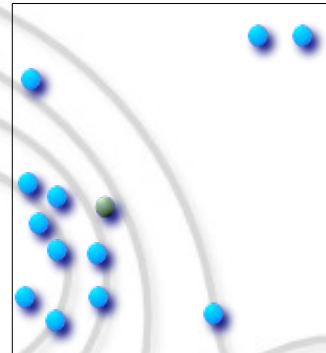
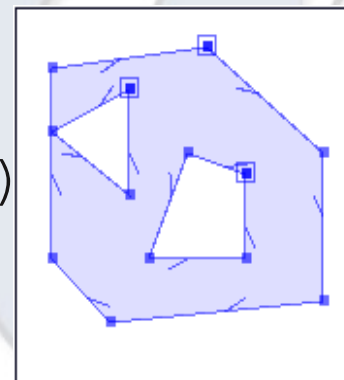


LINESTRING ( 0 5, 5 1, 9 4, 2 14, 14 13, 4 4 )



POLYGON

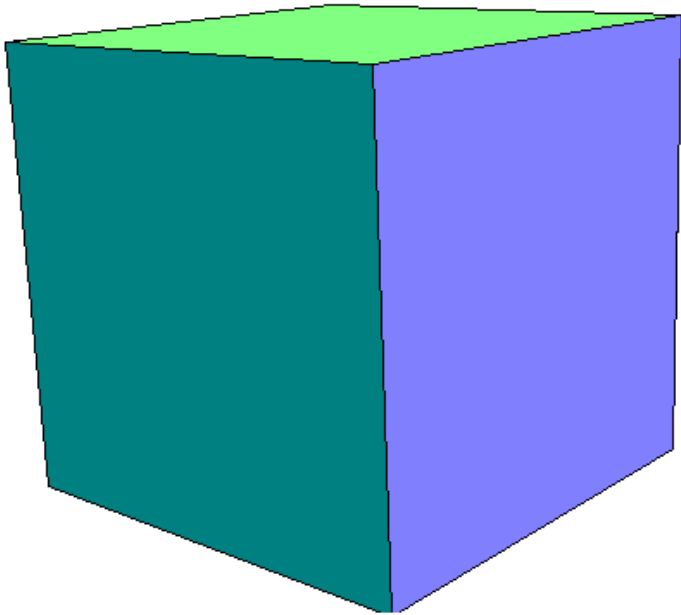
(  
 (9 13, 13 9, 13 3, 4 2, 1 4, 1 12, 9 13),  
 (5 11, 5 6, 1 9, 5 11),  
 (10 7, 10 4, 6 4, 8 8, 10 7)  
)





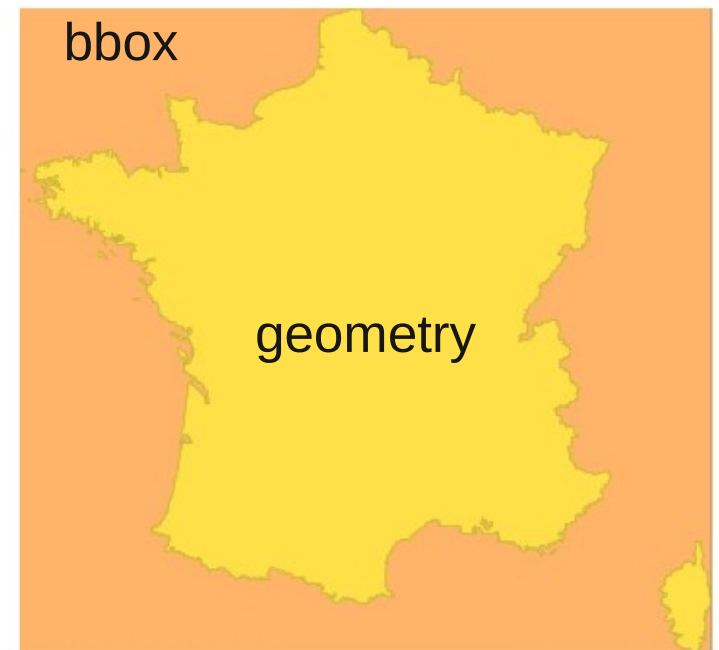
# Géométrie : Polyhedral Surface

```
PolyhedralSurface(((0 0 0, 0 0 1, 0 1 1, 0 1 0, 0 0 0)),  
                  ((0 0 0, 0 1 0, 1 1 0, 1 0 0, 0 0 0)),  
                  ((0 0 0, 1 0 0, 1 0 1, 0 0 1, 0 0 0)),  
                  ((1 1 0, 1 1 1, 1 0 1, 1 0 0, 1 1 0)),  
                  ((0 1 0, 0 1 1, 1 1 1, 1 1 0, 0 1 0)),  
                  ((0 0 1, 1 0 1, 1 1 1, 0 1 1, 0 0 1)))
```



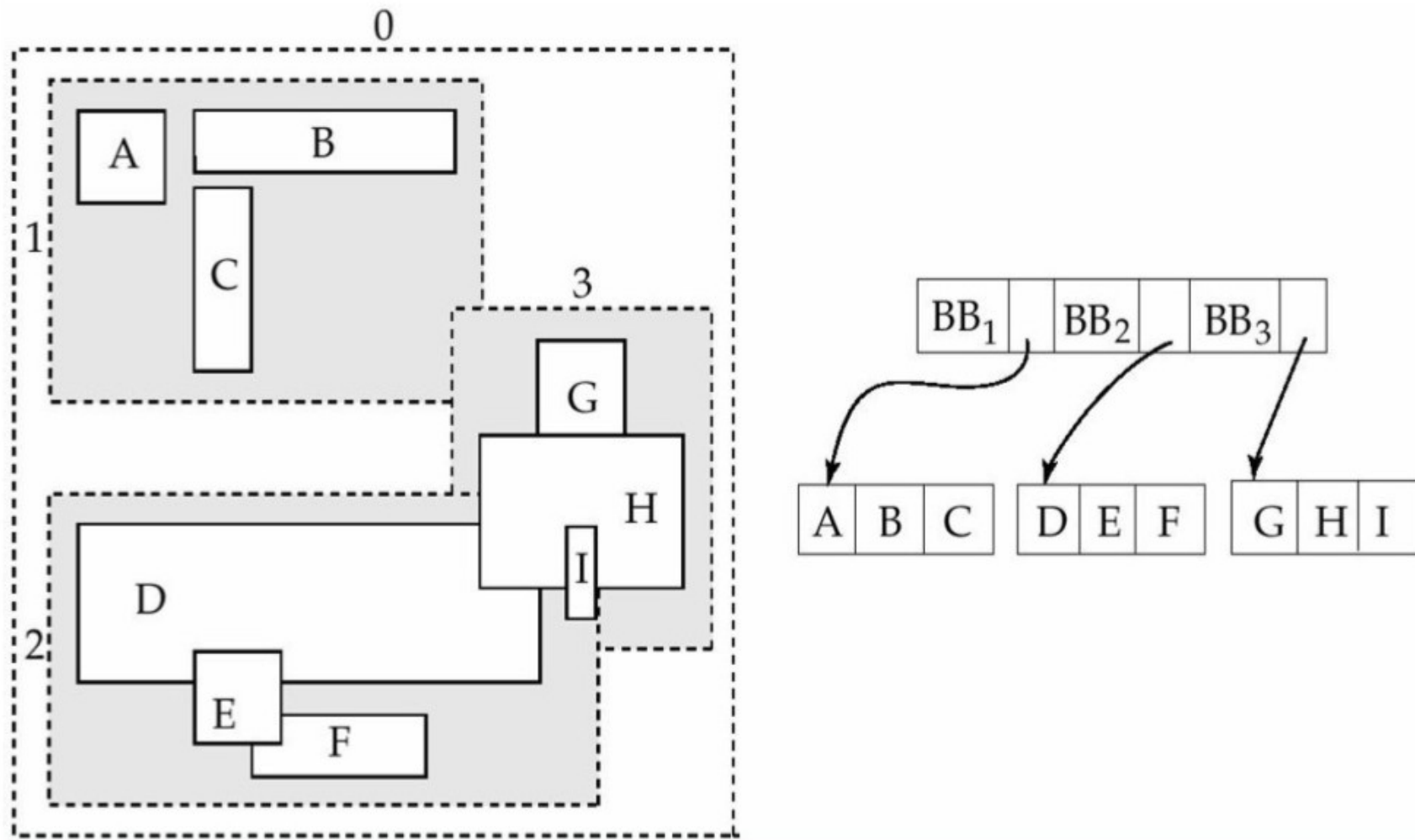
# Index spatiaux : principe & création

- Performances pour filtrage spatial
- Approximation des géométries par bbox
- Basé sur GIST
- Création index spatial :



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

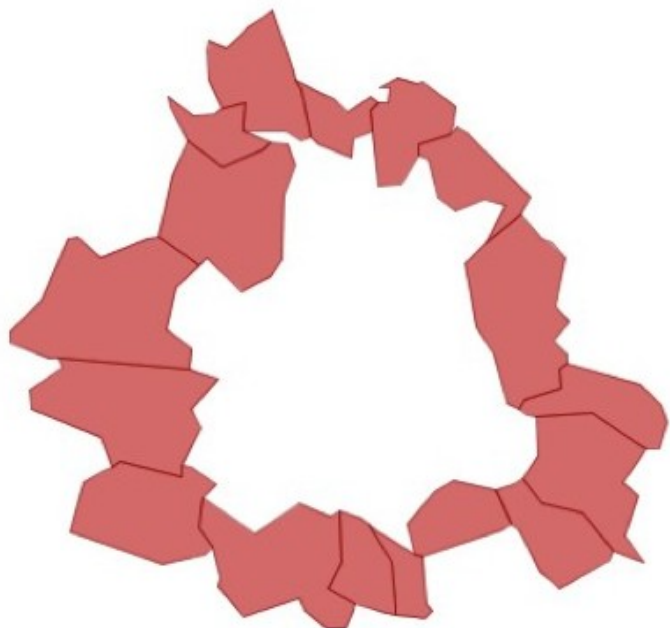
# Index spatiaux : R-tree



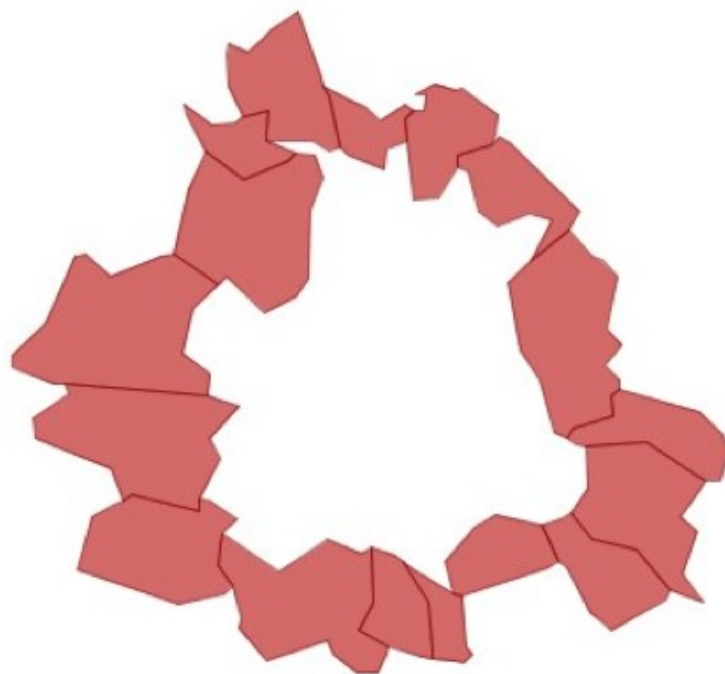
- Les bboxes sont groupées dans des régions de l'index

# Index spatiaux

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



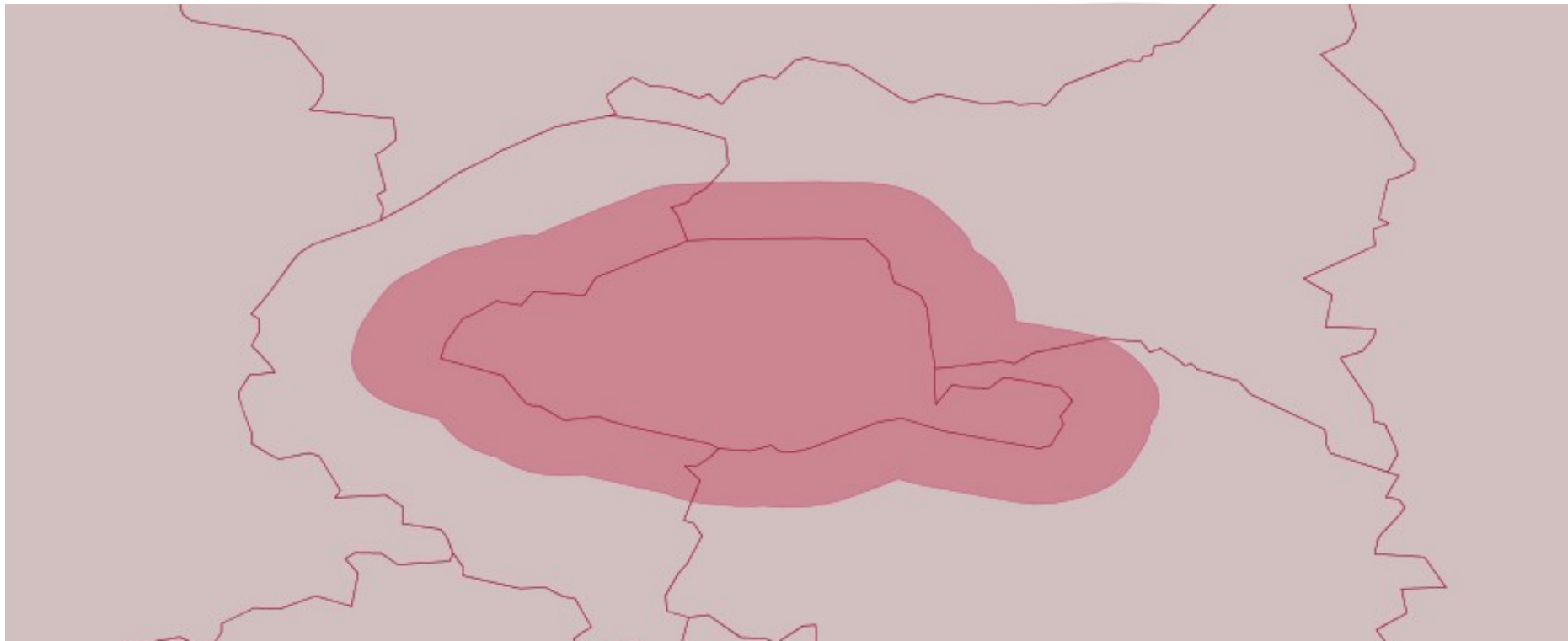
Sans index: temps = 150 ms



Avec index: temps = 30 ms

# Fonctions PostGIS : buffer

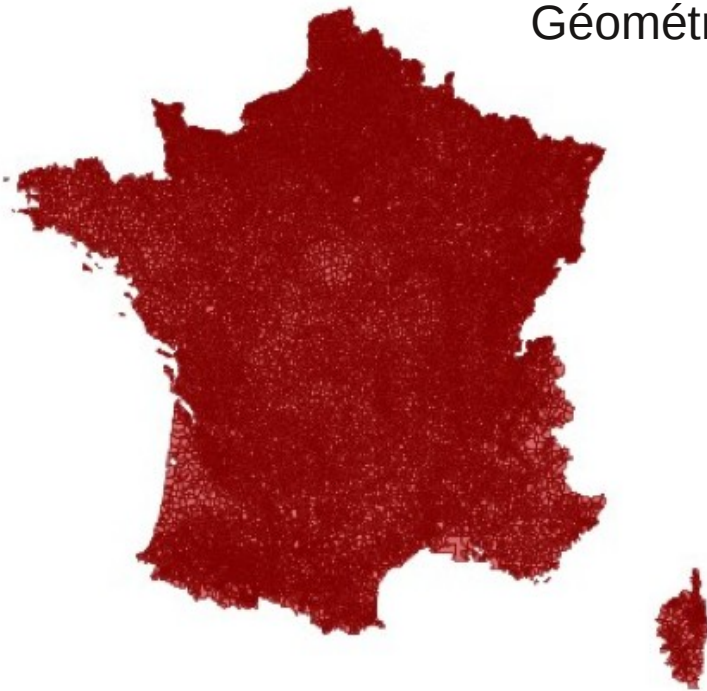
```
SELECT ST_Buffer(the_geom, 2500)  
FROM dept  
WHERE code_dept='75' ;
```



# Fonctions PostGIS : agrégats

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

Géométries des communes



Départements créés à partir des communes



# Fonctions PostGIS : intersection

```
SELECT nom_dept  
FROM dept  
WHERE ST_Intersects(the_geom,  
    (SELECT ST_Buffer(the_geom, 2500)  
    FROM dept WHERE code_dept='75' )  
);
```

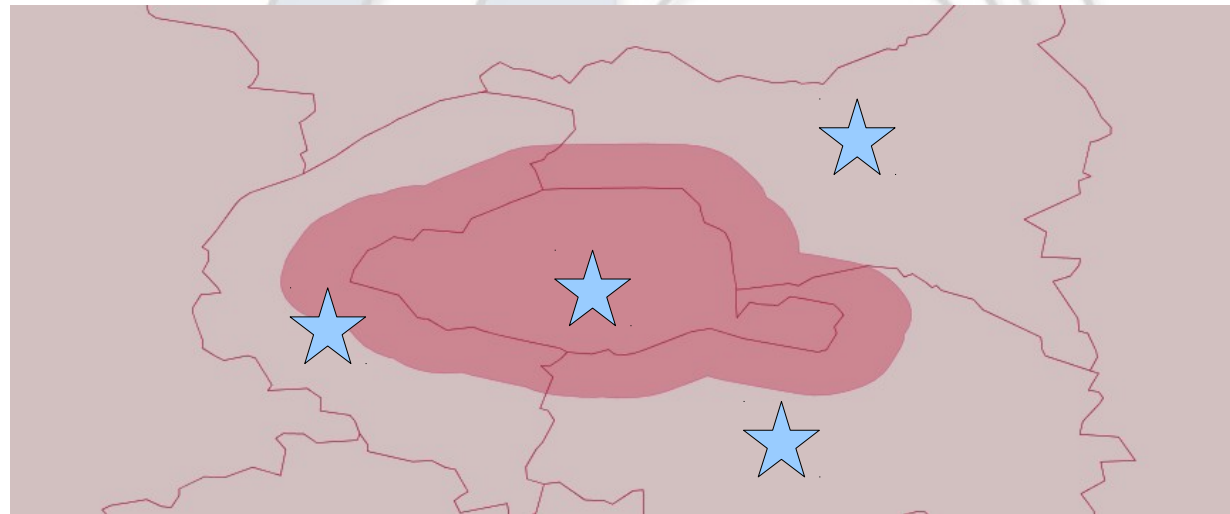
Results :

PARIS

HAUTS-DE-SEINE

SEINE-SAINT-DENIS

VAL-DE-MARNE





# PostGIS functions : distance

```
SELECT code_dept, round(  
    ST_Distance(ST_Centroid(the_geom),  
    (SELECT ST_Centroid(the_geom)  
    FROM dept WHERE code_dept='75')) / 1000)  
AS distance  
FROM dept ORDER BY distance LIMIT 4;
```

Results:

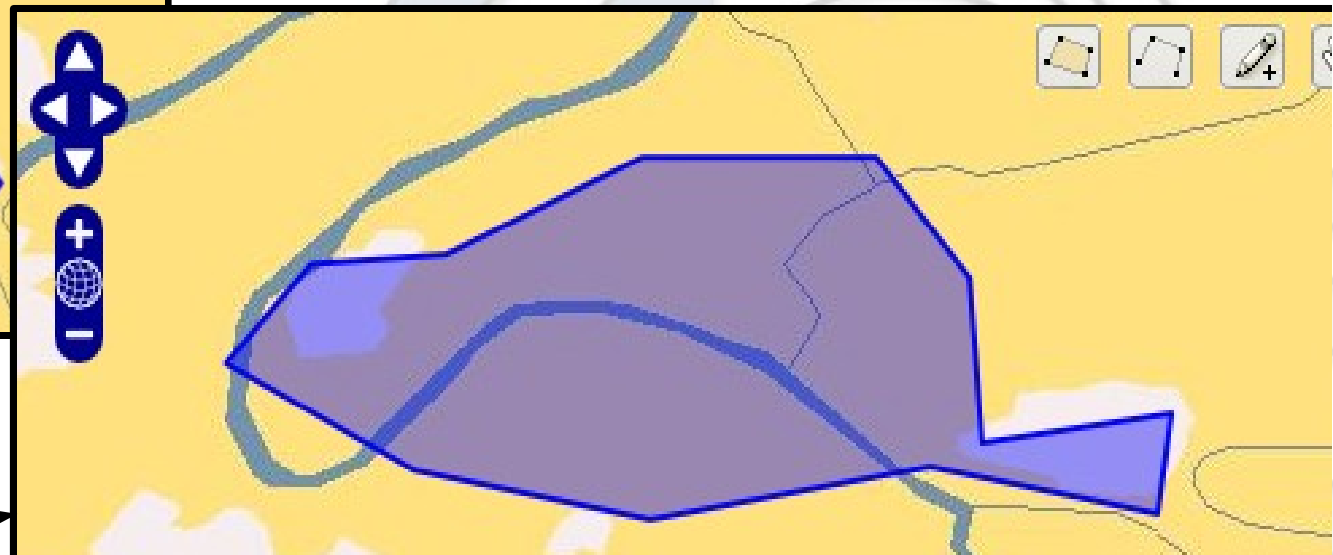
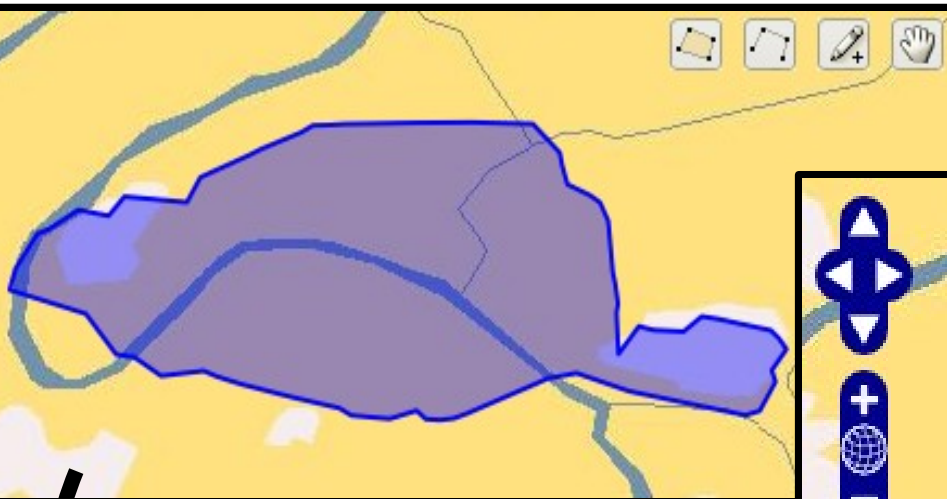
75		0
92		7
93		12
94		13



# Fonctions PostGIS : généralisation

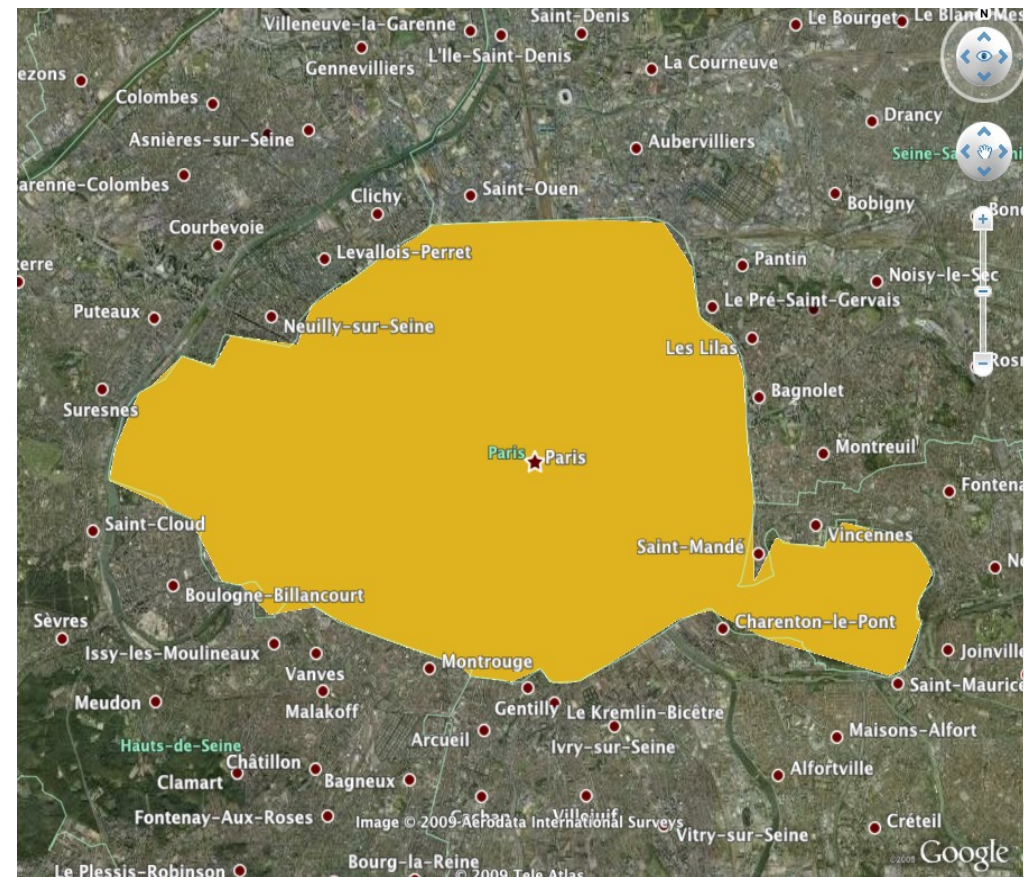
## Algorithme : Douglas-Peuker

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



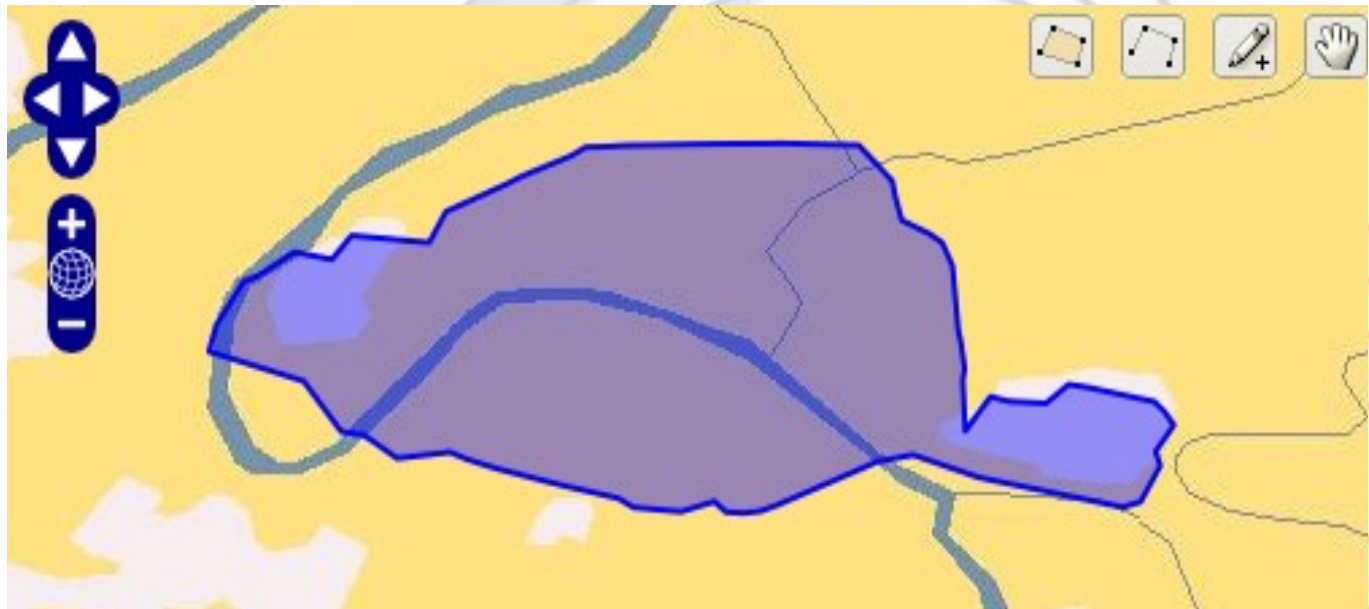
# Fonctions PostGIS : export KML

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



# Fonctions PostGIS : export GeoJSON

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





# PostGIS 2.0





# PostGIS 2.0

- PostGIS 2.0.0 : 3 avril 2012
- Après 26 mois !
- Version majeure
- Casse la compatibilité
- Une tonne de nouveautés
- De meilleures performances



**Quoi de neuf doc ?**



# Dans le cambouis

- Nouveau format de sérialisation
  - Nouveaux types géométriques (3D)
  - Correction bbox 2D uniquement
  - Correction de l'alignement des octets
- Nouveaux parsers
  - WKB
  - WKT



# Fonctionnalités

- Fonctions de gestion
- Standardisation ISO SQL/MM
- Nouvelles fonctions d'analyse
- Topologie (SQL/MM)
- 3D réelle
- Fonctions Raster / Vecteur
- Recherche indexée KNN
- TIGER (geocodeur & inverse...)



# Installation

- Installation facilitée (PG  $\geq$  9.1)
- `CREATE EXTENSION postgis ;`
- `CREATE EXTENSION postgis_topology ;`



# Gestion

- geometry\_columns → vue
- Utilisation de Typmod
- Ancien mode disponible

```
CREATE TABLE buildings (  
    gid SERIAL PRIMARY KEY  
    , geom geometry(MultiPolygon, 26986)  
);  
  
alter table buildings  
    alter column geom  
        type geometry(MultiLineString, 2154)  
        using st_setsrid(geom, 2154);
```

# Chargement

PostGIS Shapefile Import/Export Manager

PostGIS Connection

View connection details...

Import Export

Import List

Shapefile	Schema	Table	Geo Column	SRID	Mode	Rm
/home/formation/data/nyc_census_blocks	public	nyc_census_blocks	geom	0	Create	<input type="checkbox"/>
/home/formation/data/nyc_neighborhoods	public	nyc_neighborhoods	geom	0	Create	<input type="checkbox"/>
/home/formation/data/nyc_streets	public	nyc_streets	geom	0	Create	<input type="checkbox"/>
/home/formation/data/nyc_subway_stations	public	nyc_subway_stations	geom	0	Create	<input type="checkbox"/>

Add File

Options... Import About Cancel

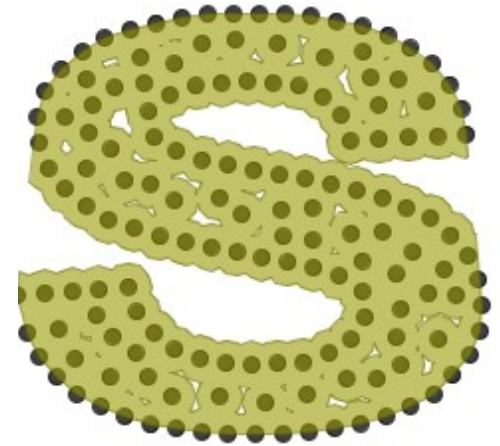
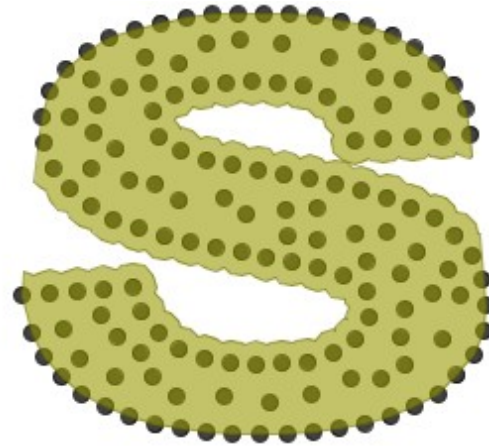
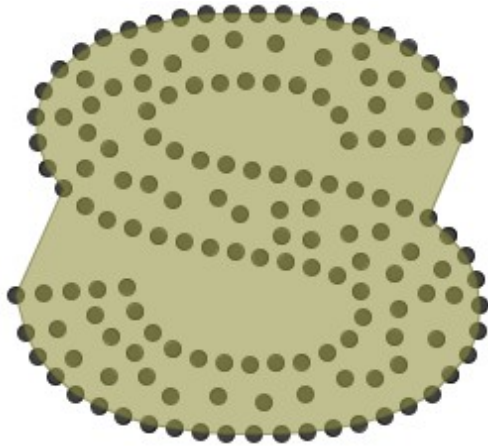
Log Window

# Fonctions

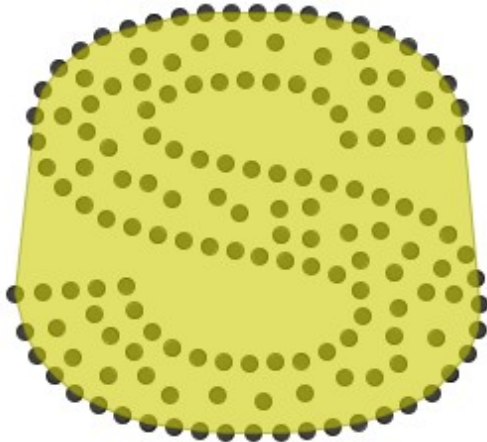
- ST\_ConcaveHull
- ST\_Snap
- ST\_Split
- ST\_MakeValid
- ST\_IsValidDetail
- ST\_OffsetCurve
- ...



# Enveloppes et courbes



Enveloppe concave



Enveloppe convexe



Courbes avec offset

# Nettoyer des données

- Avant : ST\_Buffer(the\_geom, 0)
- Après :
  - ST\_MakeValid()
  - ST\_RemoveRepeatedPoints()
  - ST\_IsValidReason()
  - ST\_IsValidDetail()

```
SELECT ST_IsValid(geom),ST_IsValidReason(geom) FROM
  (SELECT ST_GeomFromText('POLYGON ((0 0, 0 10, 10 10, 10 0, 0 0),(20 20, 20 30, 30 30, 30 20, 20 20))') as geom) as foo;
```

st_isvalid	st_isvalidreason
f	Hole lies outside shell at or near point (20.0, 20.0, NaN)

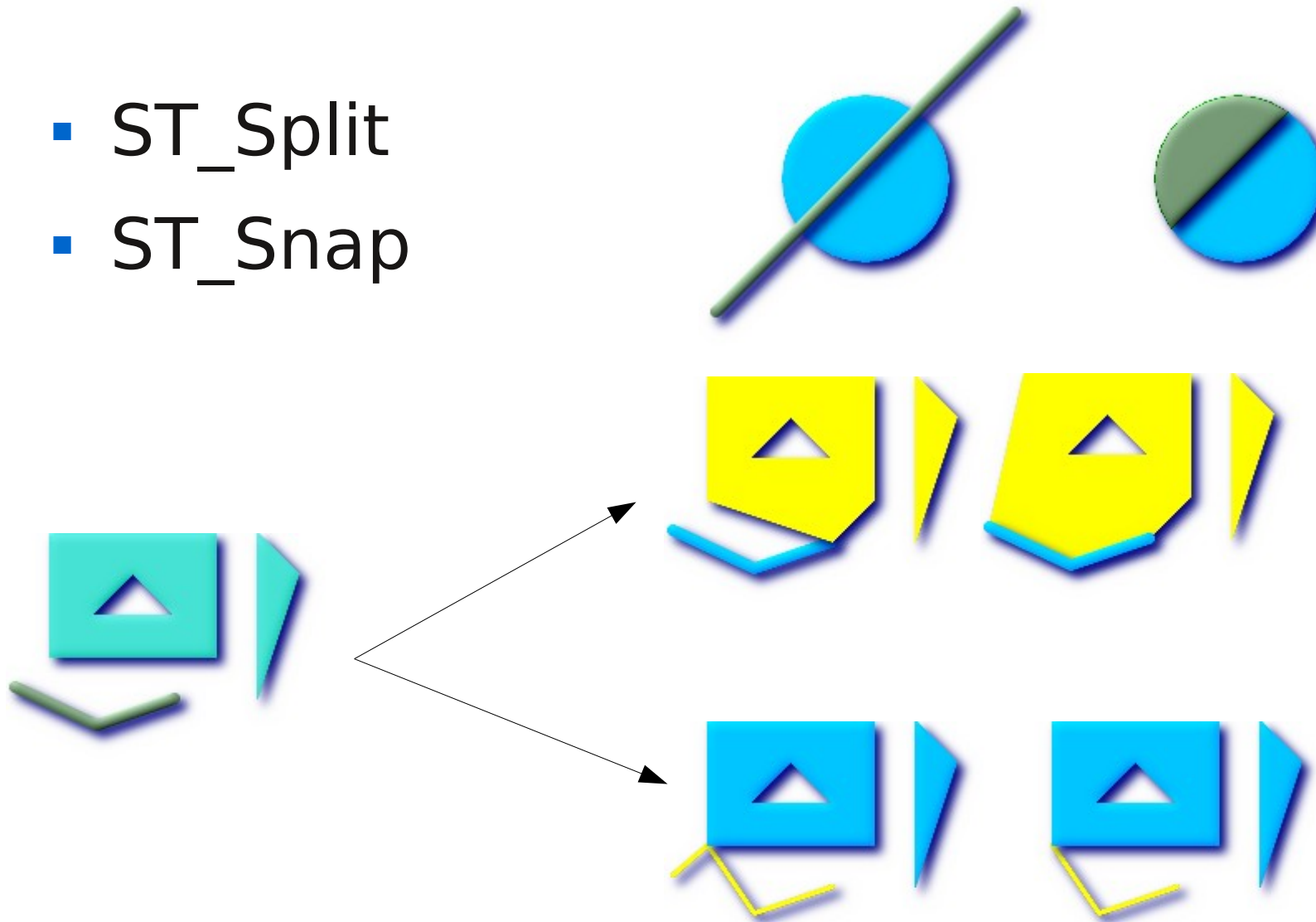
```
SELECT * FROM ST_IsValidDetail('LINESTRING(...)');
```

gid	reason	location
5330	Self-intersection	POINT(32 5)



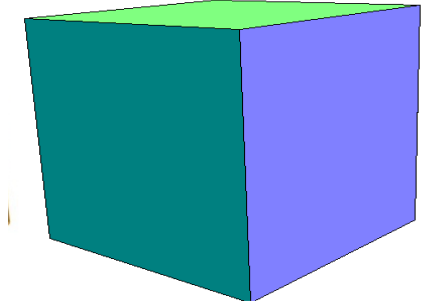
# Split & Snap

- ST\_Split
- ST\_Snap



# 3D pour de vrai

- 3D «réelle» dans PostGIS
- Standards ISO et OGC
  - ISO 19125, SQL/MM, SFS 1.2.0
- Première étape
- Nouveaux types et fonctions





- Nouveaux types :
  - TRIANGLE, POLYHEDRALSURFACE, TIN
- Nouvelles fonctions :
  - ST\_3DDistance, ST\_3DIntersects, ST\_3DDWithin, ST\_3DClosestPoint...
  - Import/export : ST\_AsGML, ST\_AsX3D...
- Nouveaux opérateurs
  - &&&
- Indexes spatiaux : nd-indexes

# TIGER

- Geocodeur
- Geocodeur inverse
- Import de TIGER vers PostGIS topology
- Mise à jour de l'outil d'import



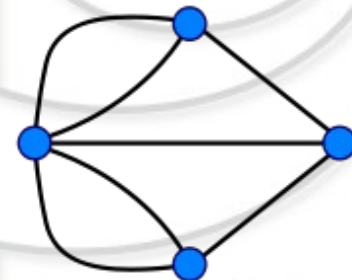
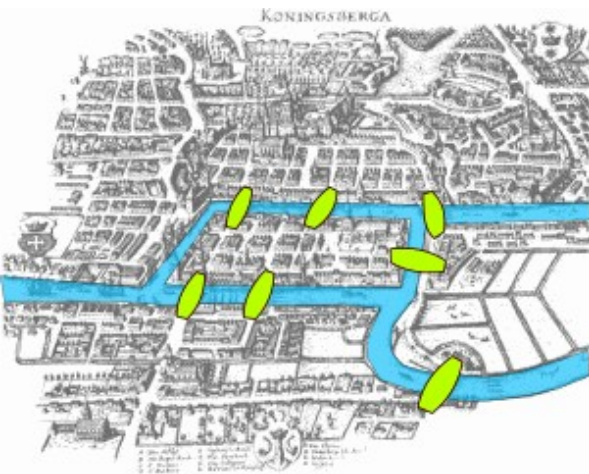
# Topologie



*Gare au monstre-spaghetti*

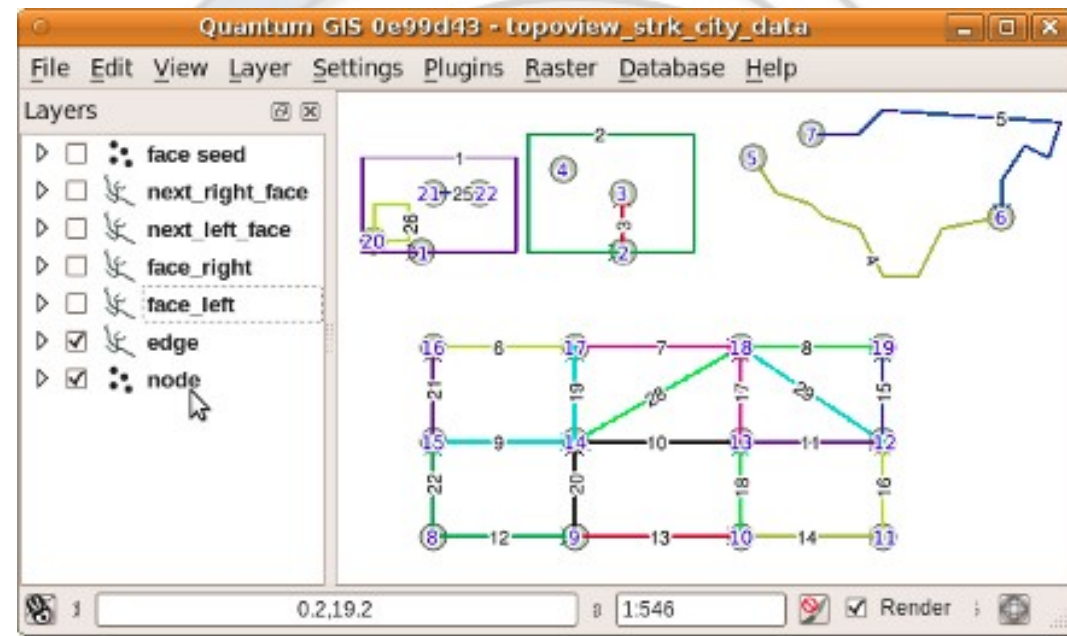
# Topologie - Graphes

- Relations explicites entre objets
- Représentation en graphe
- Différents modèles
  - Node / edge / face  $\leftrightarrow$  Noeud / Arc / Face



# Topologie PostGIS

- Type de donnée TopoGeometry
- Utilisation de schemas
  - «topology» pour les fonctions (et autres)
  - Chaque topologie dans son schema
- Support complet de SQL/MM
- Intégré dans la 2.0
  - Sandro Santilli
  - Region Toscane





```
-- Create a topology
SELECT topology.CreateTopology('hydro', 2154);
-- 1

-- we put the postgis topology features for hydro network in another table
CREATE TABLE tr_topo (gid integer);

-- Add a layer
SELECT topology.AddTopoGeometryColumn('hydro', 'public',
    'tr_topo', 'topogeom', 'MULTILINESTRING');
-- 1

-- Populate the layer and the topology from tr geometry features
INSERT into tr_topo (gid, topogeom)
    SELECT gid, topology.toTopoGeom(geom, 'hydro', 1) FROM tr;
```

- [-] Schémas (3)
  - [-] hydro
    - Collationnements (0)
    - Domaines (0)
    - Configurations FTS (0)
    - Dictionnaires FTS (0)
    - Analyseurs FTS (0)
    - Modèles FTS (0)
    - Fonctions (0)
    - Séquences (5)
    - Tables (4)
      - edge\_data
      - face
      - node
      - relation
    - Fonctions trigger (0)
    - Types (0)
    - Vues (1)
      - edge

```
select * from hydro.edge limit 10;
```

neau sortie

ortie de données

Expliquer (Explain)

Messages

Historique

	edge_id integer	start_node integer	end_node integer	next_left_edge integer	next_right_edge integer	left_face integer	right_face integer	geom geometry(LineString)
1	175256	190369	190361	175230	-175243	0	0	01020000206A080
2	167356	183762	181917	166725	167356	0	0	01020000206A080

```
select * from tr_topo limit 10;
```

eau sortie

ortie de données

Expliquer (Explain)

Messages

gid integer	topogeom topology.topogeometry
116768	(1,1,163704,2)
116767	(1,1,163705,2)

# ... SQL power ! ...

create table

rec\_res2 as

with recursive

search\_graph(edge\_id, start\_node, depth, path, length, cycle) as (

select

1

g.edge\_id, g.start\_node, 1 as depth, ARRAY[g.edge\_id] as path  
, st\_length(g.geom) as length, false as cycle

from

hydro.edge as g

where

edge\_id = 173832

union all

select

2

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

)

select

sg.\*  
, edge.geom as geom

from

search\_graph as sg

join

hydro.edge as edge

on






sg.edge\_id = edge.edge\_id

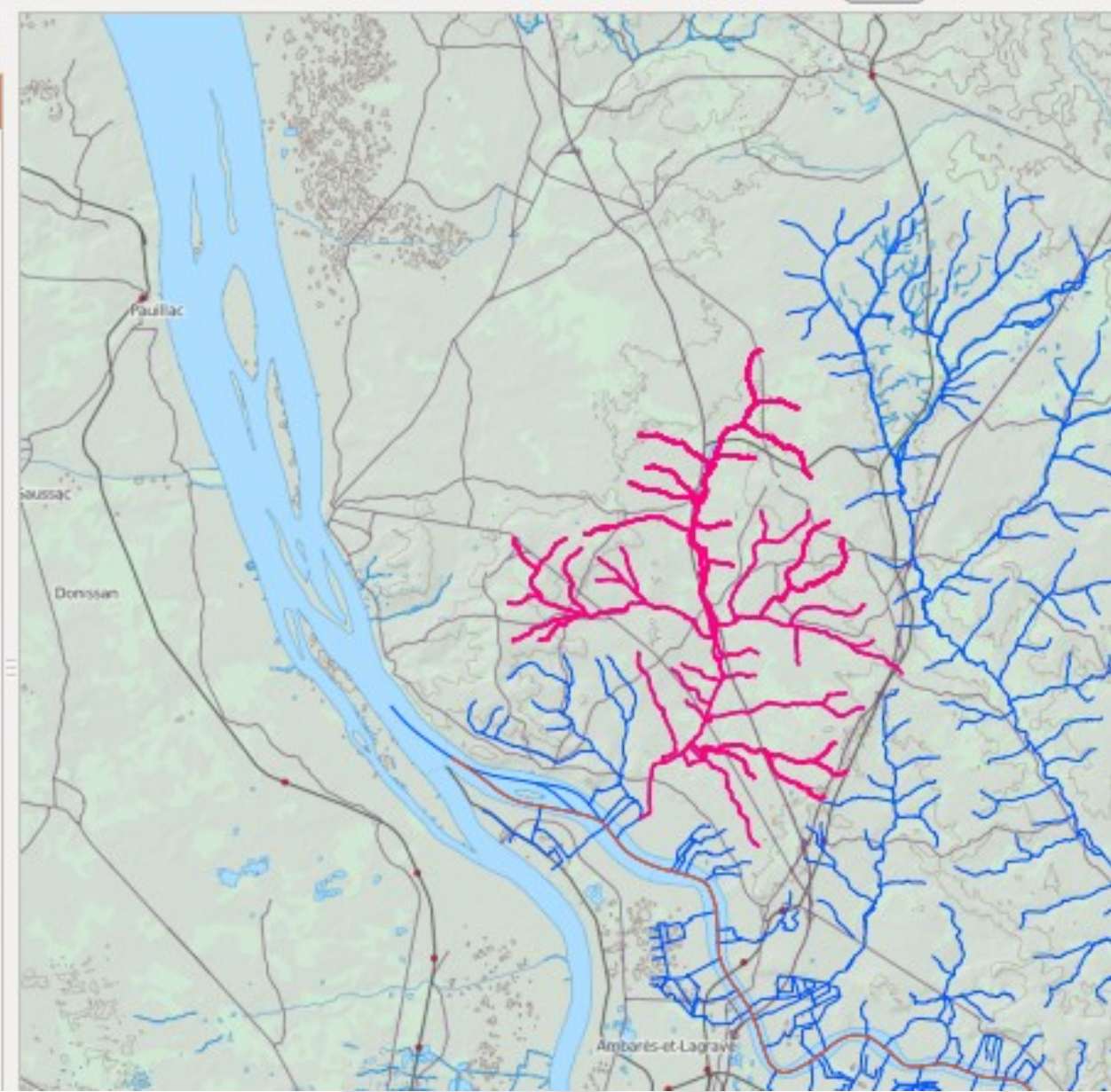
limit 1000;

3



Couches

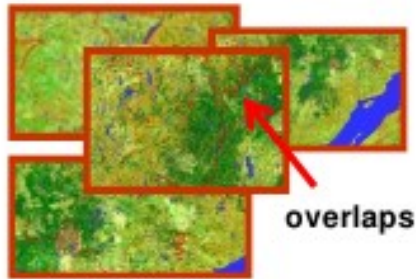
- ▶ ☒  recursive\_upstream
- ▶ ☒  shortest\_path\_topology
- ▶ ☐  shortest\_path\_pgrouting
- ▶ ☒  hydro network
- ☒  background



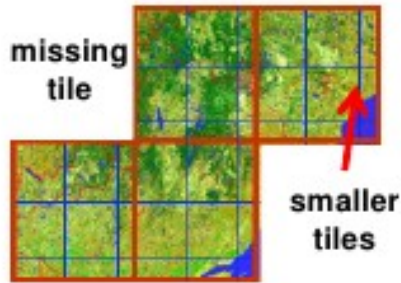
# PostGIS Raster

- Analyse Raster / vecteur
- Nouveau type de données
  - Ressemble à geometry
  - ... mais pour les rasters
- Multiresolution, multibande, couverture de tuiles
- Import/export (GDAL)
- Fonctions
  - Statistiques, reprojection, edition, calculs
- Fonctions Vecteur/raster

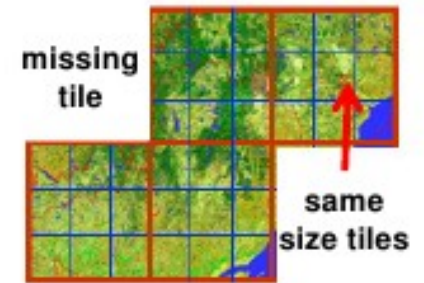
# PostGIS Raster



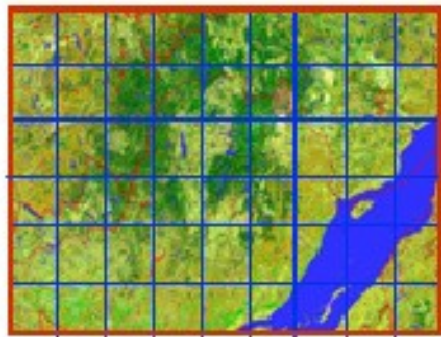
a) warehouse of untiled and unrelated images (4 images)



b) irregularly tiled raster coverage (36 tiles)



c) regularly tiled raster coverage (36 tiles)



d) rectangular regularly tiled raster coverage (54 tiles)

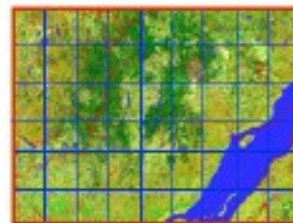


Table 1

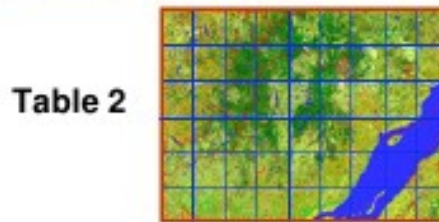
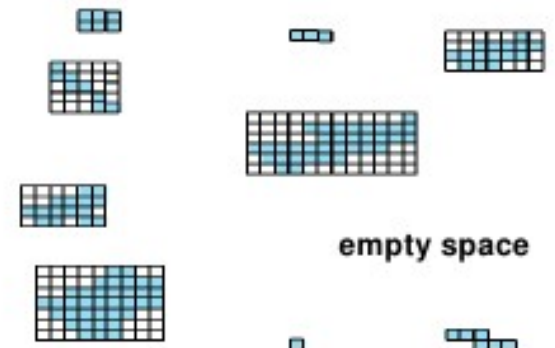


Table 2

e) tiled images (2 tables of 54 tiles)



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





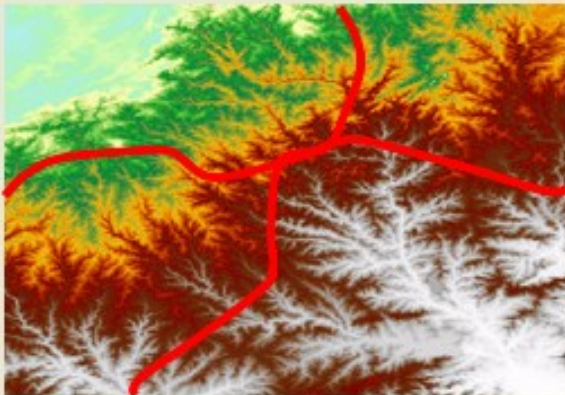
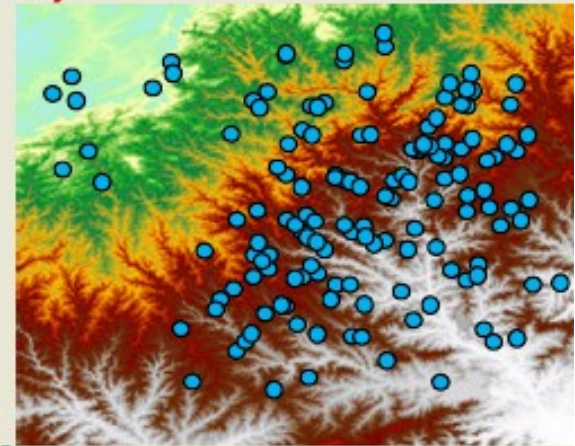
# PostGIS 2.0 : PostGIS Raster

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 2.0 : plus proche voisins

- Recherche KNN-GIST dans PostgreSQL 9.1
- Utilise les indexes !
- Plus proche voisins spatiaux
  - `SELECT name, gid FROM geonames  
ORDER BY  
geom <-> st_setsrid(st_makepoint(-90,40),4326)  
LIMIT 10;`
- Opérateurs de distance
  - `<->` or `<#>` : centre ou bbox
  - Besoin de refiltrer pour les géométries `<>` points

# Travaux actuels

- PostgreSQL 9.2 → SP-Gist
  - SP comme SPatial !
  - Plus rapide à créer (→ 3x) / plus rapide à lire  
→ PostGIS 2.1
- Améliorations
  - Raster, topology, 3D
  - Divers nouvelles fonctions

# PostGIS 3.0 ?

- Codesprint et barcamp à Paris Mai 2012
- Directions pour le futur...
  - Git, build system (mainly Windows)
  - Geometry backend (GEOS vs BGL vs ?)
  - Raster improvement
  - 3D topology & processing (CGAL?)
  - Point clouds
  - Performance, performance, performance



# ... PostgreSQL 9.2 et après...

- Type range
- Stocker du JSON (~Mongo in PG)
- Scans index-only
- Cascading, Multi-master replication
- Performances
  - Multicore read & write
  - Parallel bulk load
- FDW (shapefiles...)

GPU

# C'est tout ...

## Questions ?

**[vincent.picavet@oslandia.com](mailto:vincent.picavet@oslandia.com)**

**Twitter : @vpicavet**

**<https://github.com/Oslandia/presentations>**

**<http://www.oslandia.com>**

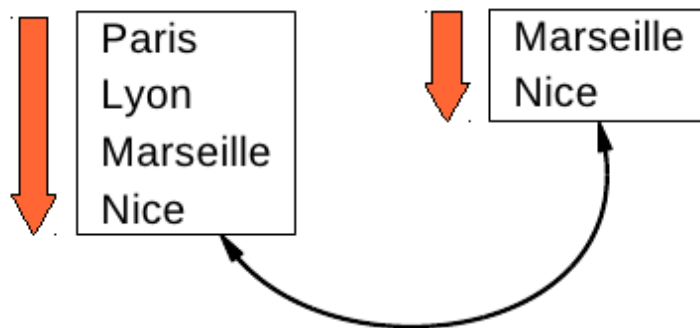
# Free bonus slides



# Cas d'utilisation SIG



- Bisonvert.net
  - Covoiturage opensource
- Objectif
  - Relier les personnes faisant un même trajet
- Méthode courante
  - Utiliser les noms des étapes



# Cas d'utilisation SIG

Solution :

Use real paths

1/ Compute path (routing)

2/ Match paths

(Spatial analysis)

