



PostGIS, un module de PostgreSQL pour les données spatiales

Vincent Picavet

Rencontres SIG-LL 2010

Qui ?

Vincent Picavet

Anciennement Makina Corpus

Fondateur & Gérant Oslandia

Convaincu par les modèles Open Source

Contributeur PostGIS



Oslandia, une SSL spatiale

Services

Formation

Support

Conseil

Développement

Positionnement: SIG Côté serveur

Bases de données spatiales

Traitement et analyse complexe
routing, géocodage...

Web Services OGC et INSPIRE

PostGIS
MapServer OGC
PostgreSQL 3D
GraphServer PyWPS
Spatialite GEOS
PgRouting OSGeo
TinyOWS INSPIRE
GRASS



Plan de la présentation

Contexte

Les SIG

Cas concrets

PostGIS

Le projet et ses acteurs

Types spatiaux

Indexes spatiaux

Fonctions spatiales

Évolution

PostGIS en 2009

PostGIS en 2010

Questions



Contexte: SIG ?

80% de l'information est spatiale

Systeme d'Information Géographique

Capturer

Stocker

Échanger

Analyser

Représenter



Initialement : Collectivités territoriales

Aujourd'hui : Public & Privé

Contexte : Architecture SIG

Architecture distribuée

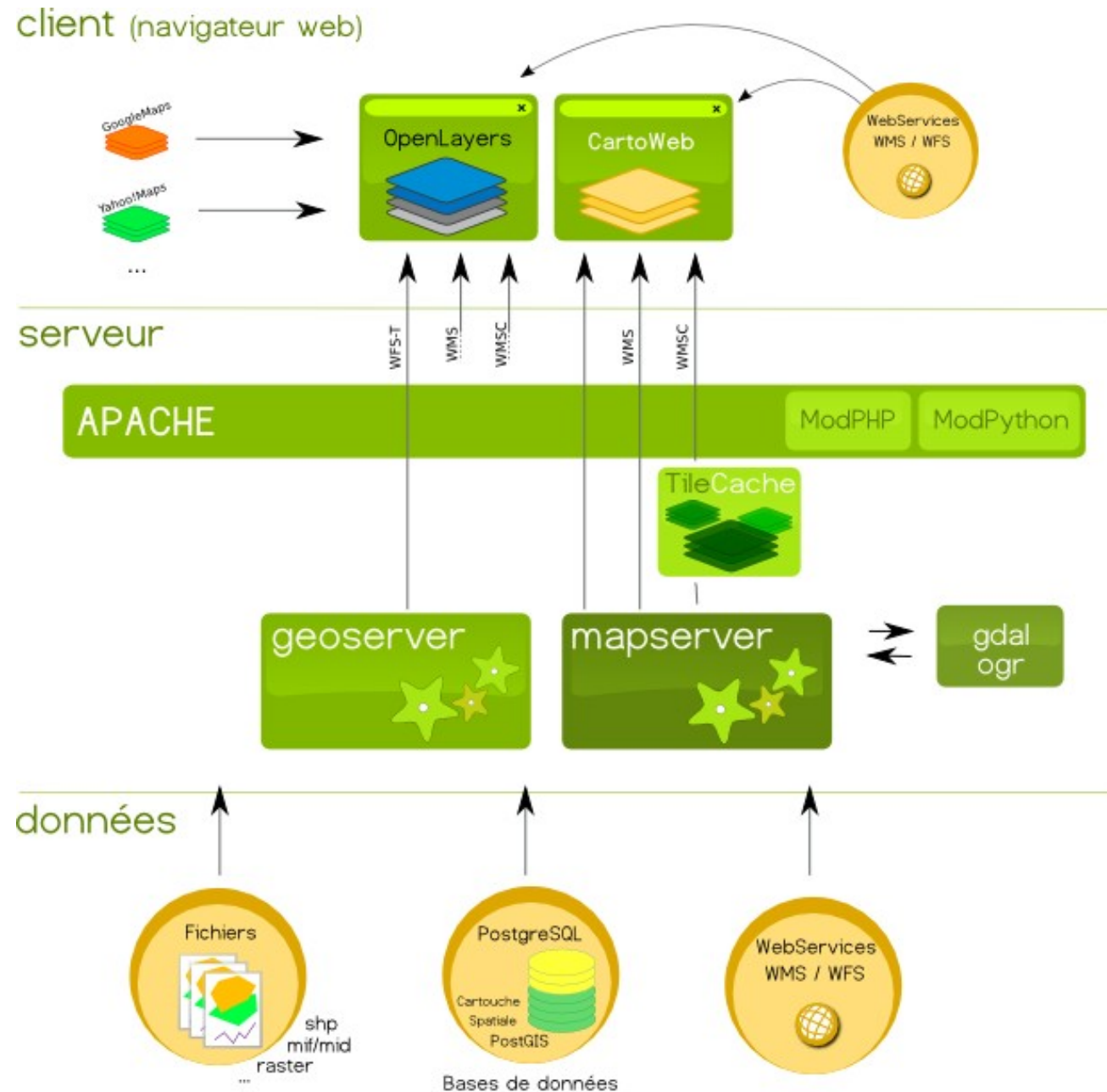
Clients

web

bureautique

Échanges

WS standardisés



Contexte : Problématique de covoiturage

BisonVert.net

Logiciel Libre de covoiturage

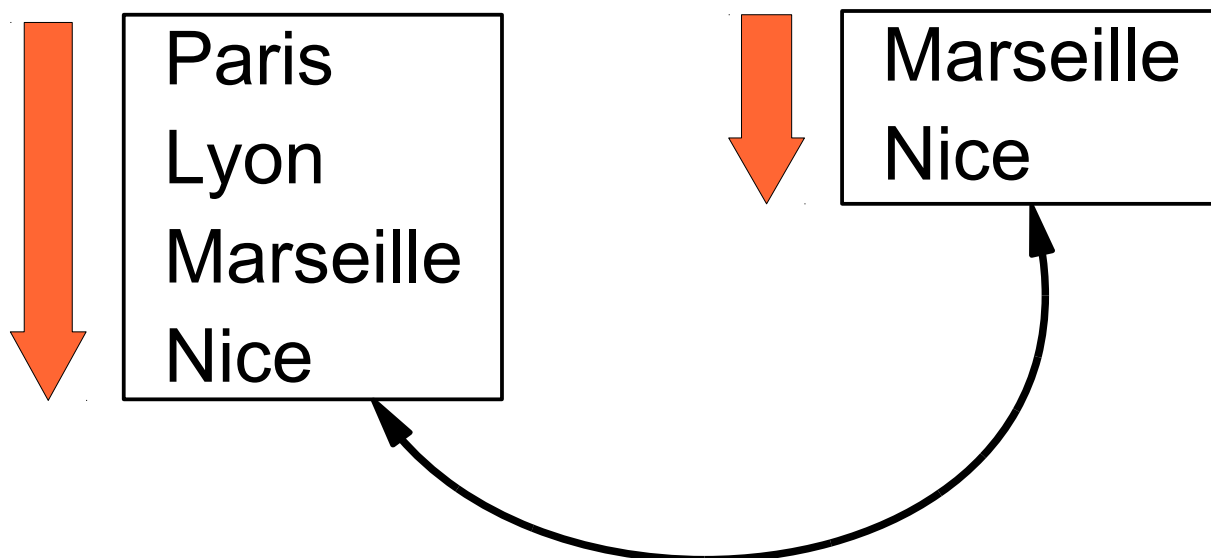


Objectif :

Associer les utilisateurs qui font le même trajet

Existant

Association avec le libellé des étapes



► Limité !

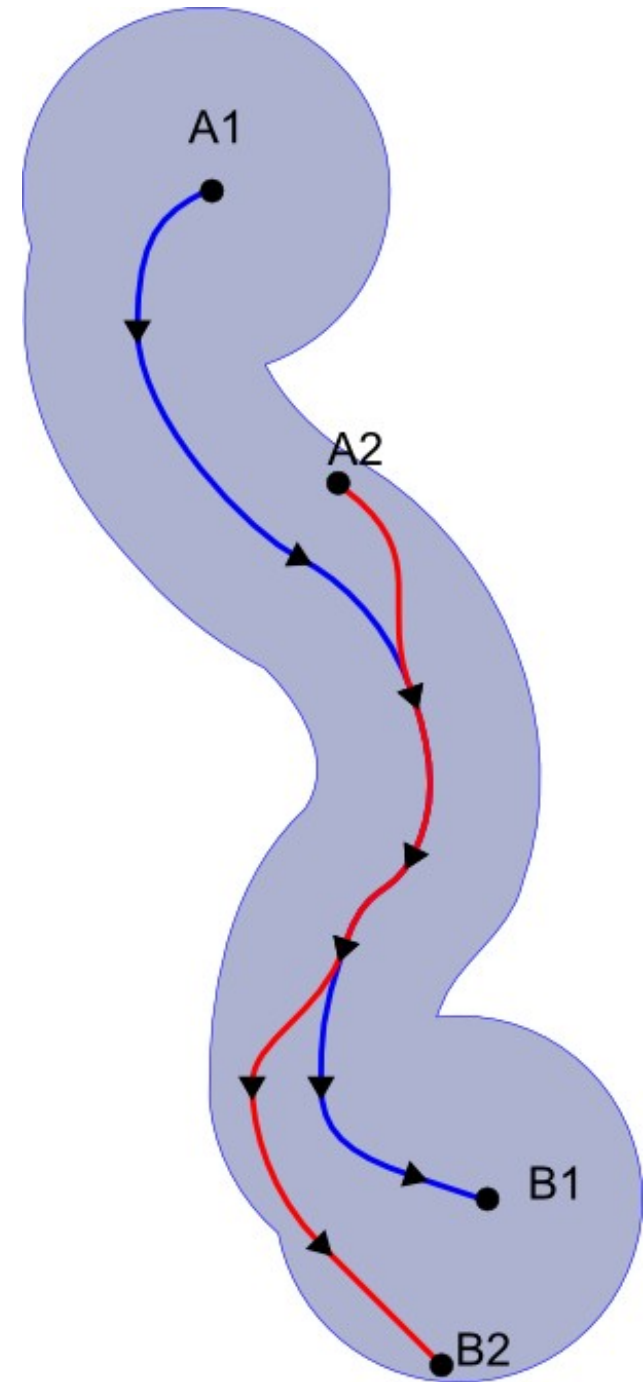
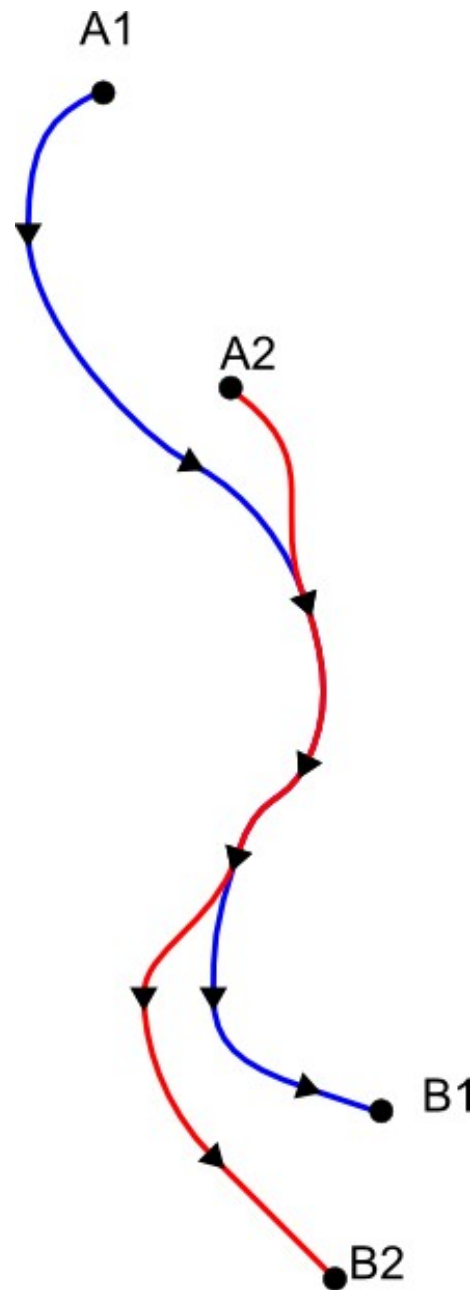
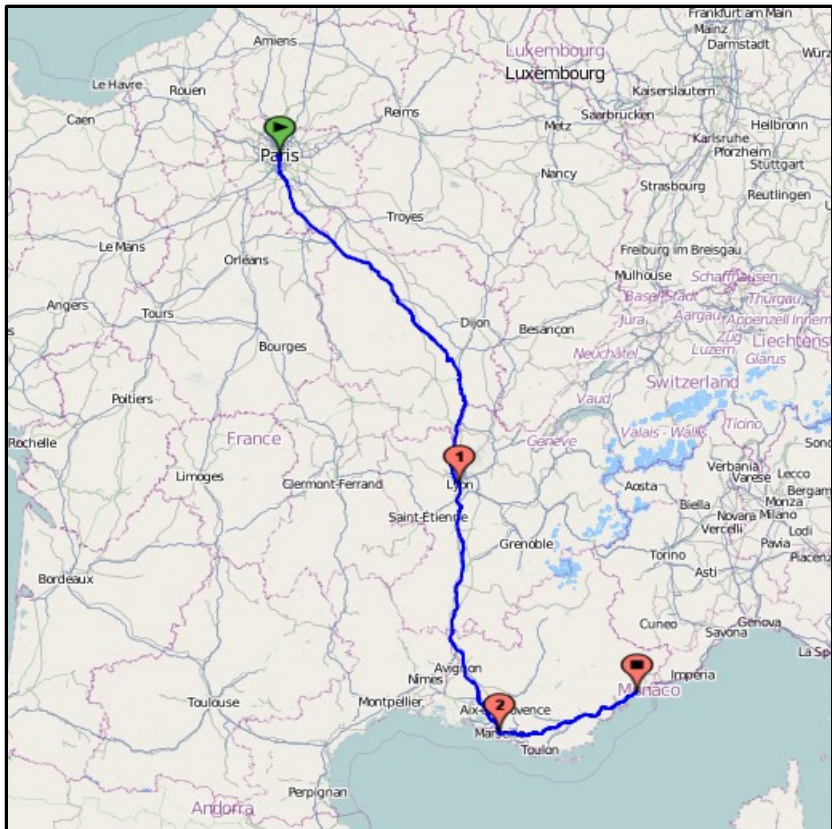


Contexte : covoiturage spatial

Solution :

Utiliser les trajets réels

- 1/ Calcul du trajet (routing)
- 2/ Comparaison des trajets (analyse spatiale)



Contexte : Principaux enjeux d'un SGBD spatial

Géométrie + attributs = «**feature**»

Requêtage SQL

- Filtres sur partie attributaire

- Filtres sur partie spatiale

Tenir la charge

- Volumes de données importants

- Traitements longs et complexes

Performances acceptables

Respect des standards existants



Contexte : Standards Internationaux

Les spécifications:

OGC SFS (Simple Feature for SQL)

ISO SQL/MM part 3



Ce qui est spécifié:

Types de géométries supportés

Prototypes des fonctions spatiales

Table et traitements additionnels pour intégrité référentielle

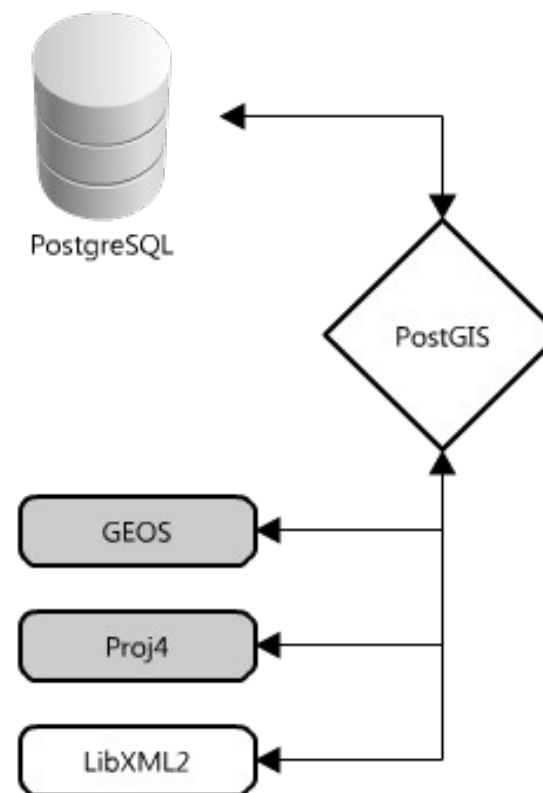


PostGIS : Principe et architecture

Plugin de PostgreSQL

Écrit principalement en C

Fait appel a des librairies tierces :

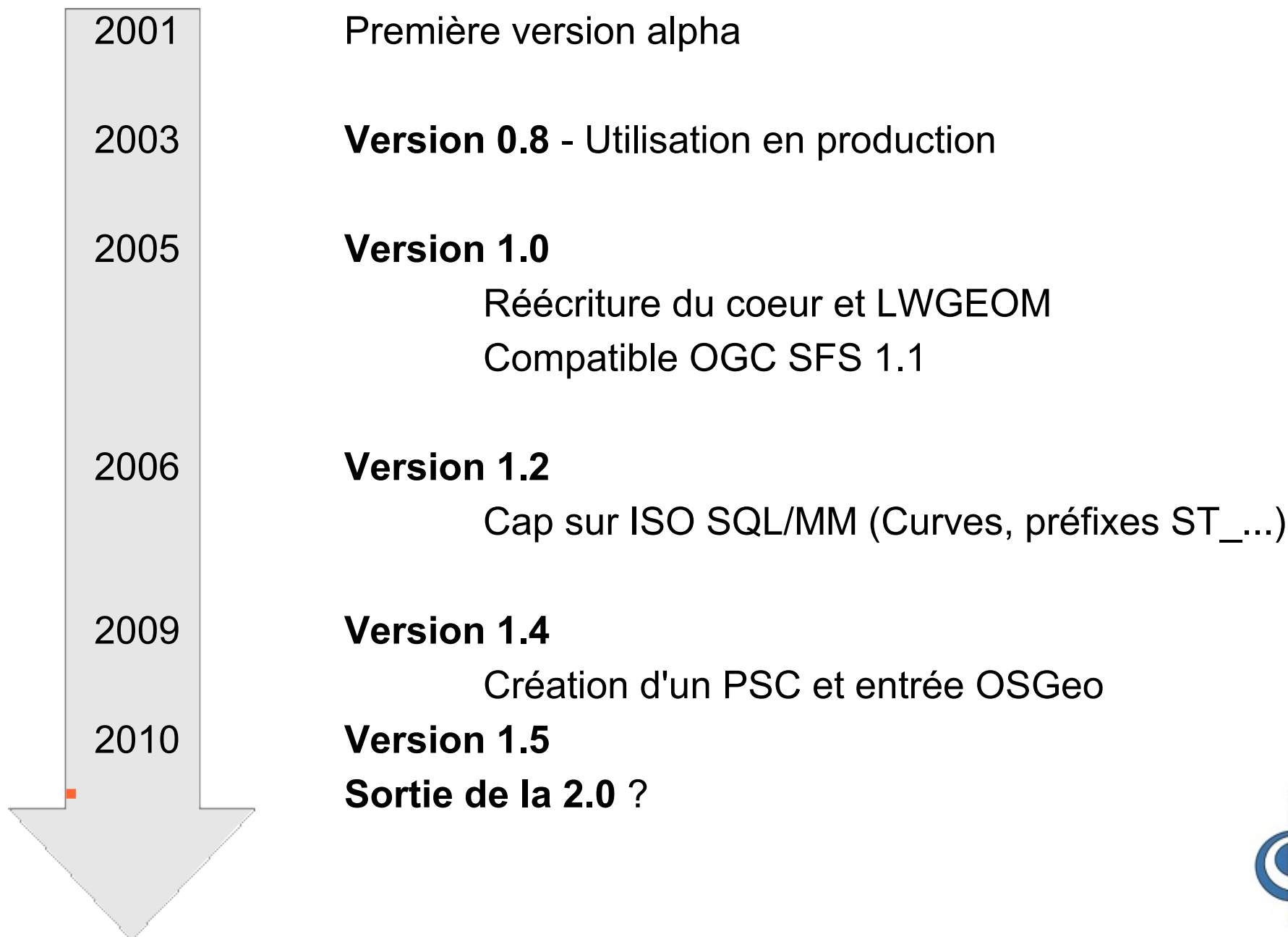


Implémente OGC SFS 1.1 (et une partie de ISO SQL/MM)

Nombreuses fonctions additionnelles



PostGIS : Rappel historique



PostGIS : Communauté

Institutionnels

IGN : Institut Géographique National

IRSN : Institut de Radioprotection et de Sûreté Nucléaire

JRC : Joint Research Center – Union Européenne

...

Entreprises françaises

Mediapost

France Telecom

...

Communauté

Mondiale

Plusieurs milliers d'utilisateurs

Mailing-list postgis-users très active



PostGIS : Committers PostGIS



LisaSoft

OpenGeo

Oslandia

CadCorp

Paragon Corporation

Refractions Research

Sandro Santilli

Sirius



PostGIS : Autres SGBD spatiaux

Oracle Spatial (et Locator)



ESRI ArcSDE

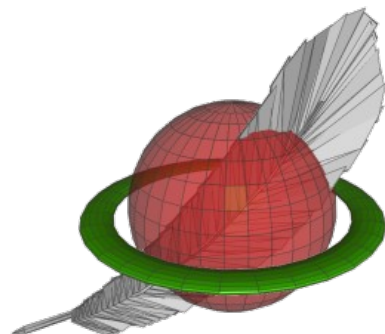
IBM DB2



Microsoft SQLServer 2008



Spatialite



Géométries : Représentation et stockage

Geometry (ou HEWKB)

Stockage natif en base

Binaire avec encodage hexa

WKT (Well Known Text)

Représentation textuelle

Dimensions

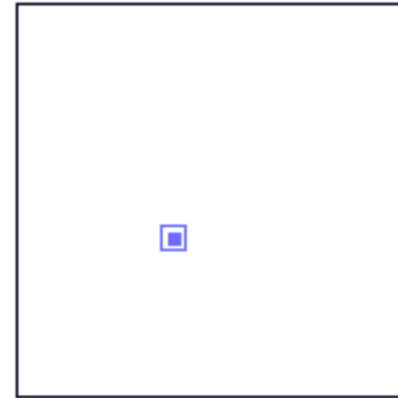
2D, 3D, ou 4D

Identifiant d'un système de projection (SRID)



Géométries: Point

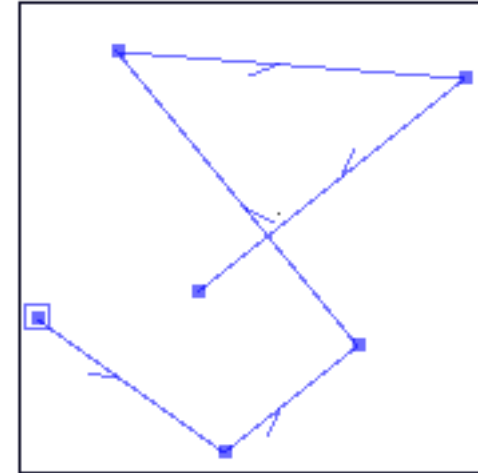
POINT (10 10)



Géométries: LineString

LINESTRING

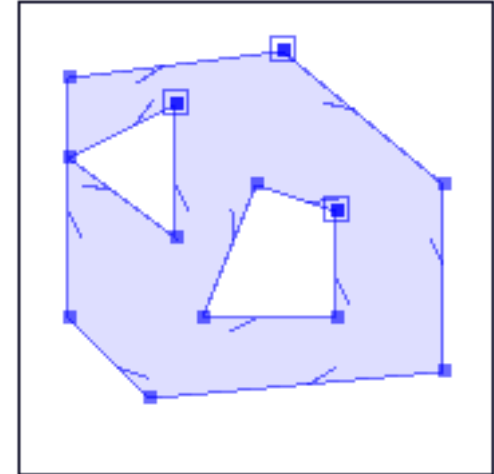
```
(  
  0 5, 5 1, 9 4, 2 14, 14 13, 4 4  
)
```



Géométries: Polygon

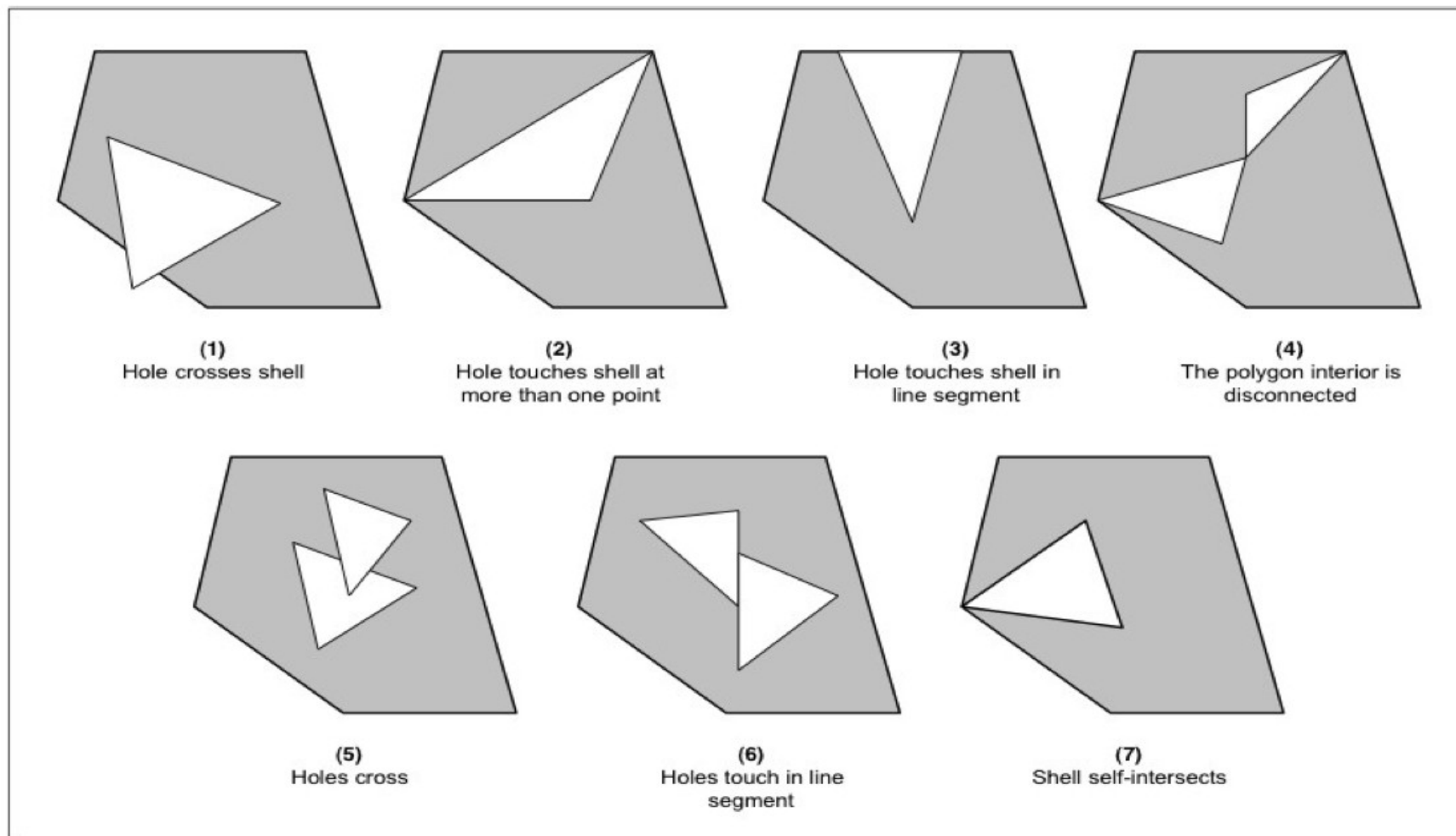
POLYGON

(
(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

Géométries: Polygon et validité SFS



Types invalides (au sens OGC SFS)

Schema extrait du site de JTS ([vividsolutions.com](http://www.vividsolutions.com))

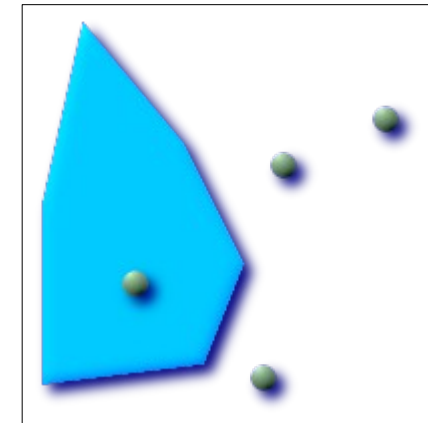
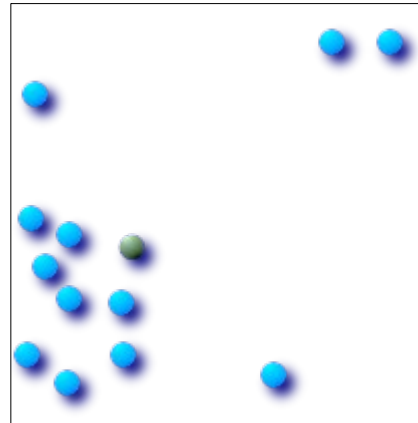
Géométries: Les types multiples et agrégats

MULTIPOINT

MULTILINESTRING

MULTIPOLYGON

GEOMETRYCOLLECTION



- 1) On ne peut pas mixer différents systèmes de projection
- 2) Idem pour les dimensions

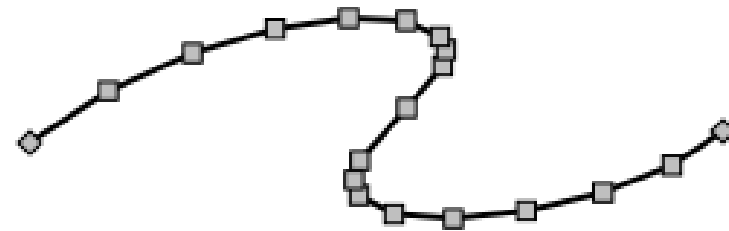
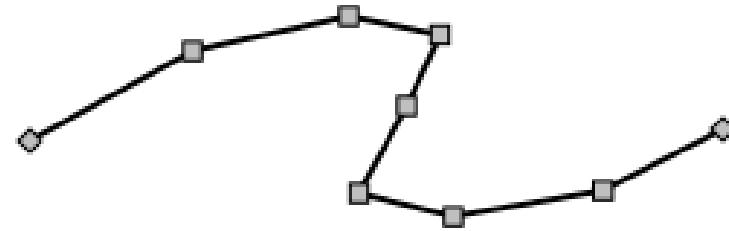
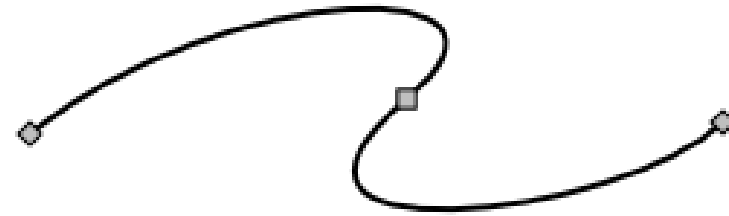
Géométries: Les courbes

Les types dits «curves» :

CIRCULARSTRING

COMPOUNDCURVE

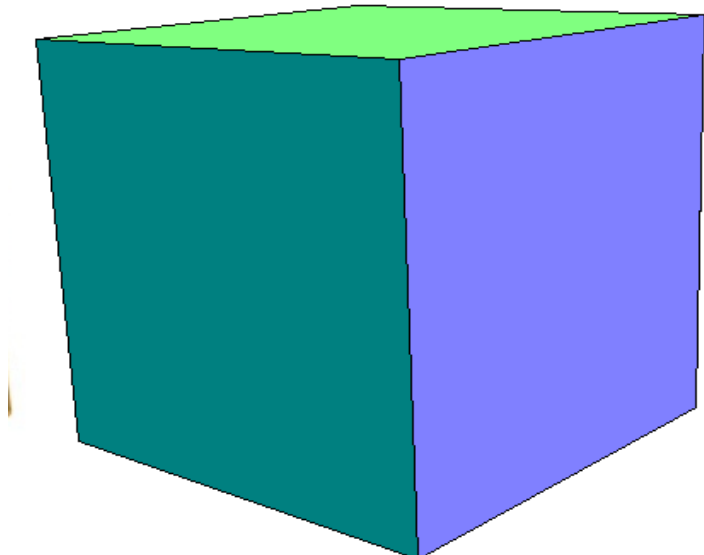
MULTISURFACE



Géométries: Polyhedral Surface

(PostGIS 2.0)

```
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)))
```



PostGIS en base : Les tables additionnelles

geometry_columns : catalogue des colonnes spatiales

	oid	f_table_catalog [PK] character va	f_table_schema [PK] character v	f_table_name [PK] character v	f_geometry_column [PK] character varyii	coord_dimension integer	srid integer	type character varying(30)
1	709958	"	public	dept	the_geom	2	27582	MULTIPOLYGON
2	709957	"	public	world	the_geom	2	4326	MULTIPOLYGON

spatial_ref_sys: catalogue des systèmes de projection

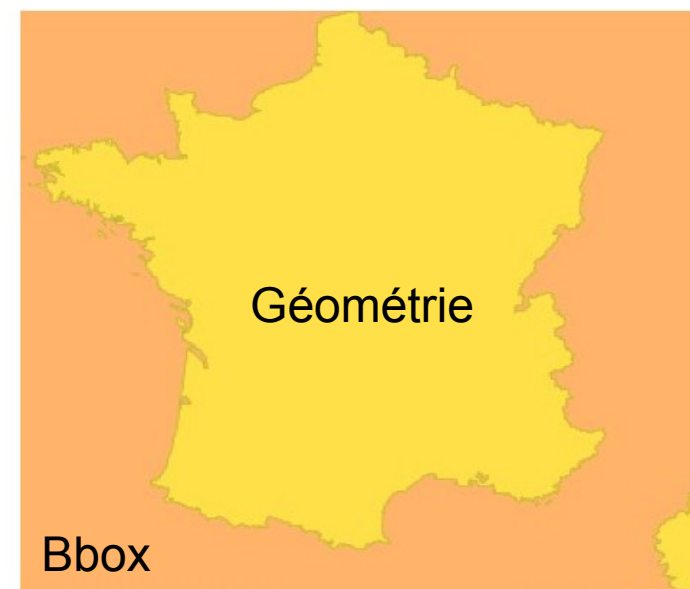
	srid [PK] integer	auth_name character var	auth_srid integer	srttext character varying(2048)	proj4text character varying(2048)
1	2000	EPSG	2000	PROJCS["Anguilla 1957 / British We	+proj=tmerc +lat_0=0 +lon_0=-62 +k=0.999
2	2001	EPSG	2001	PROJCS["Antigua 1943 / British We	+proj=tmerc +lat_0=0 +lon_0=-62 +k=0.999
3	2002	EPSG	2002	PROJCS["Dominica 1945 / British W	+proj=tmerc +lat_0=0 +lon_0=-62 +k=0.999

Astuce : placer ces tables dans un schéma à part

Index spatiaux: Principe et création

Améliorer performances sur filtrage

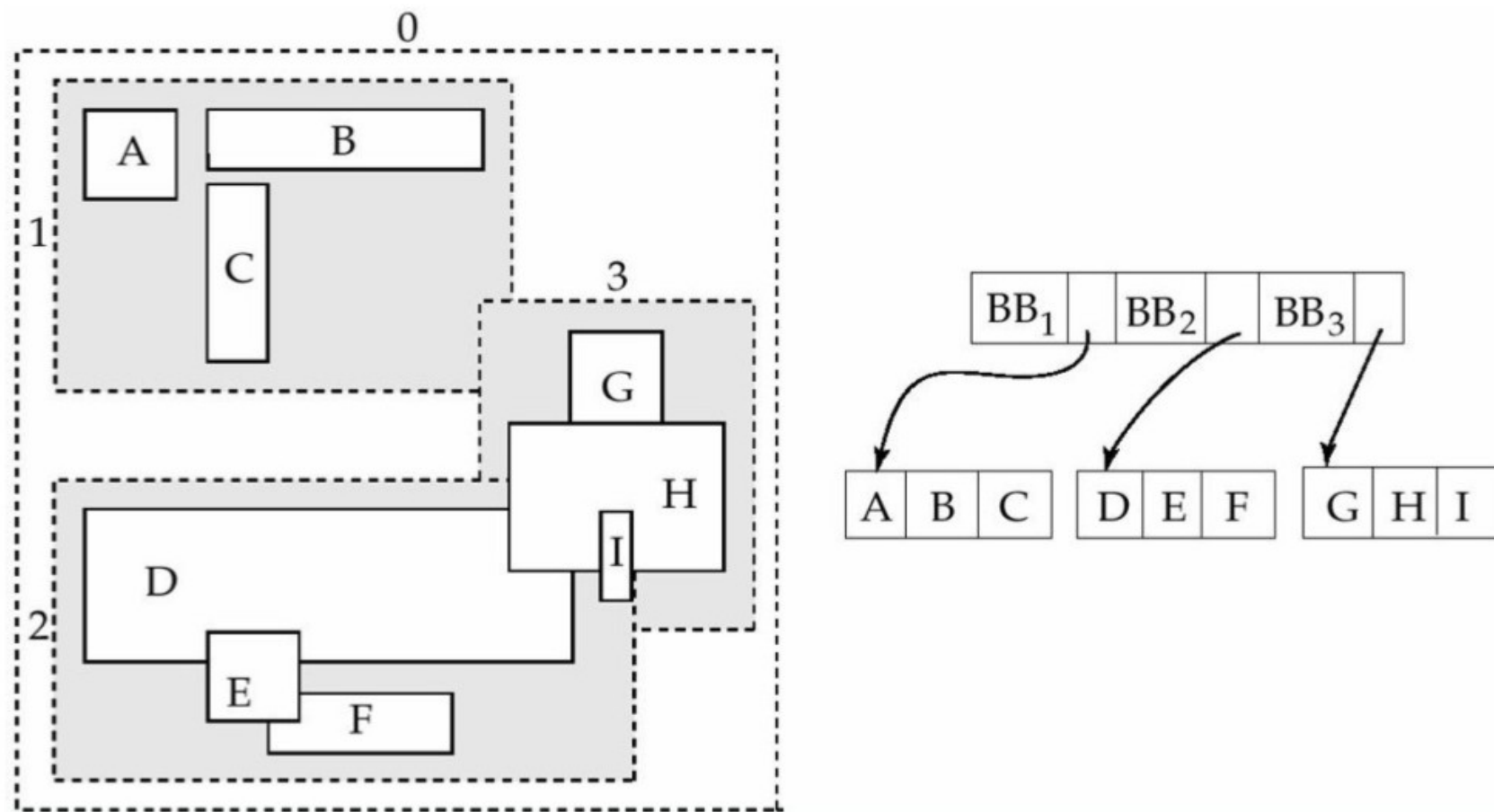
Approxime les géométries: Bbox



Création d'un index spatial:

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

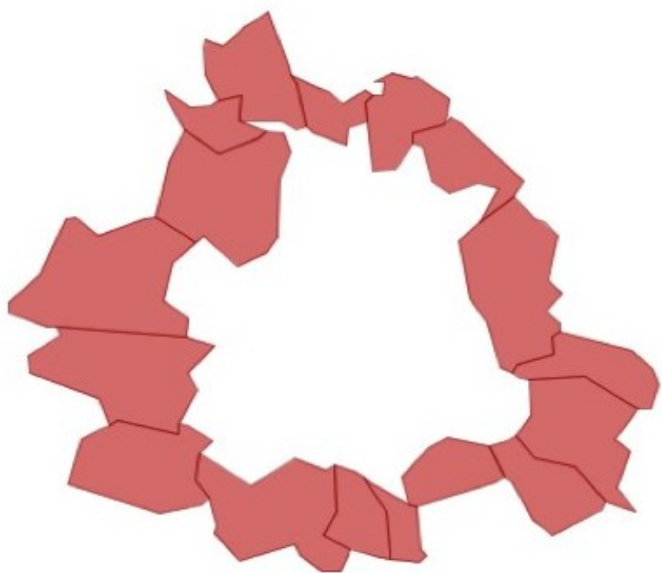
Index spatiaux: R-Tree



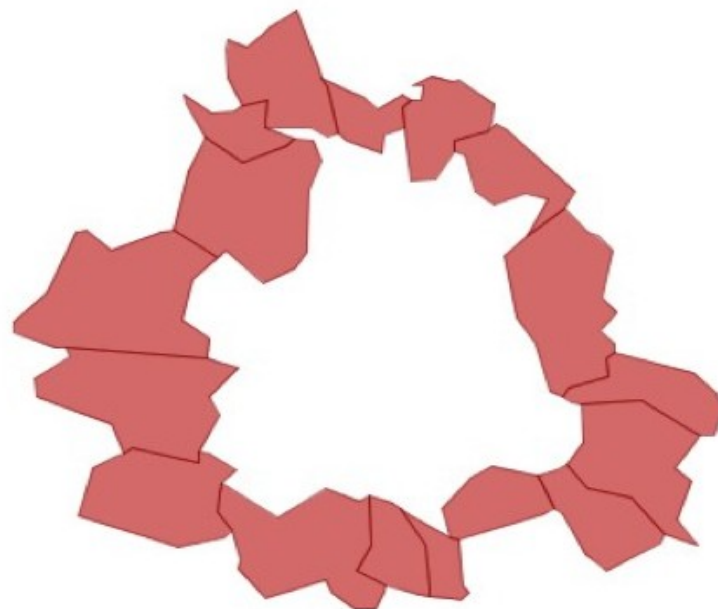
Regroupement des Bbox dans des régions de l'index

Index spatiaux

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



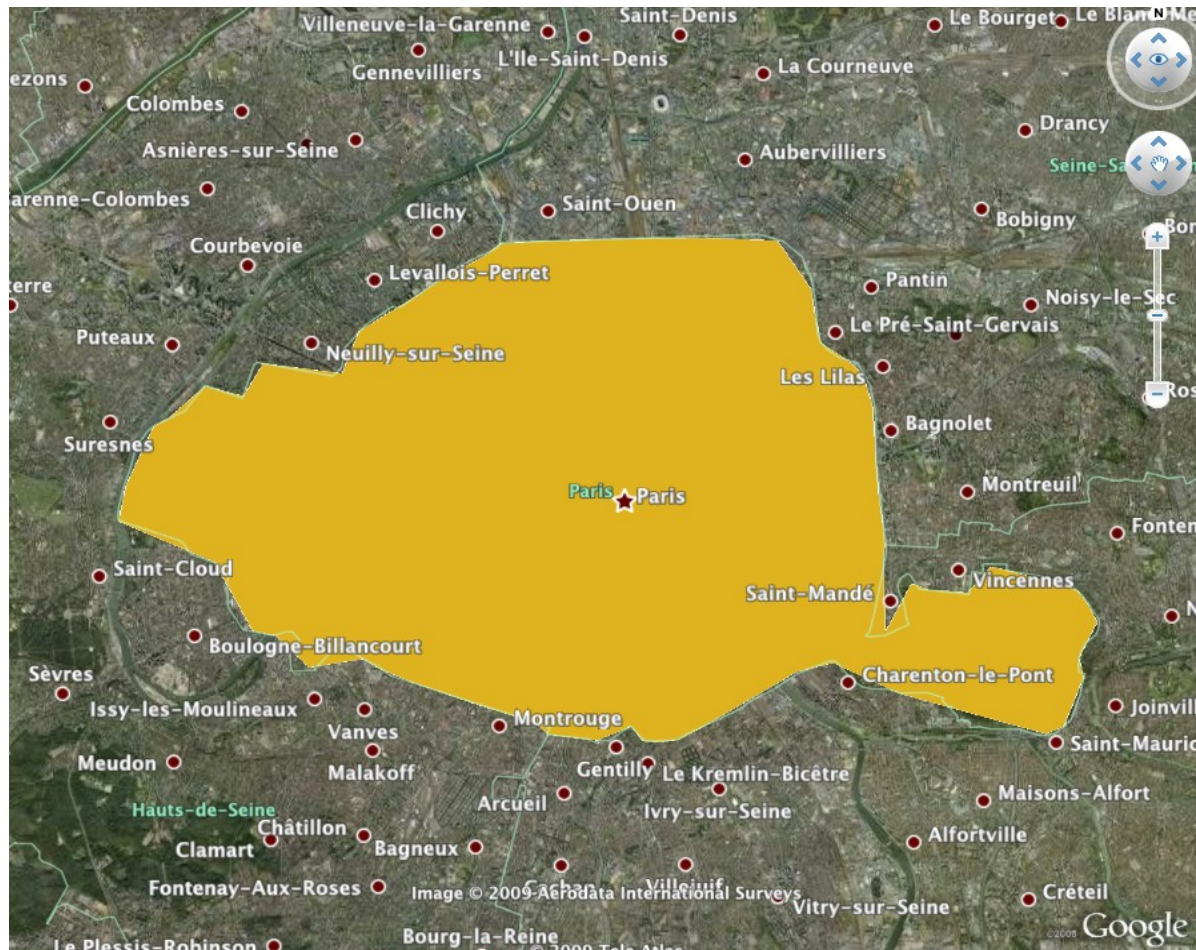
Sans index: temps = 150 ms



Avec index: temps = 30 ms

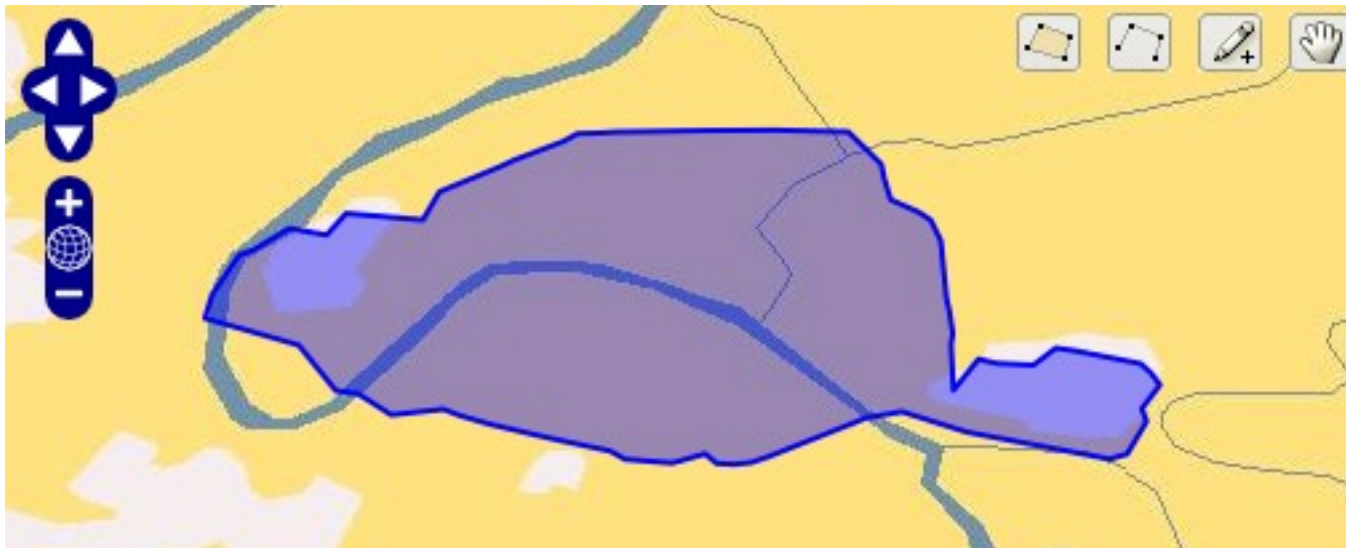
Fonctionnalités : Export KML

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



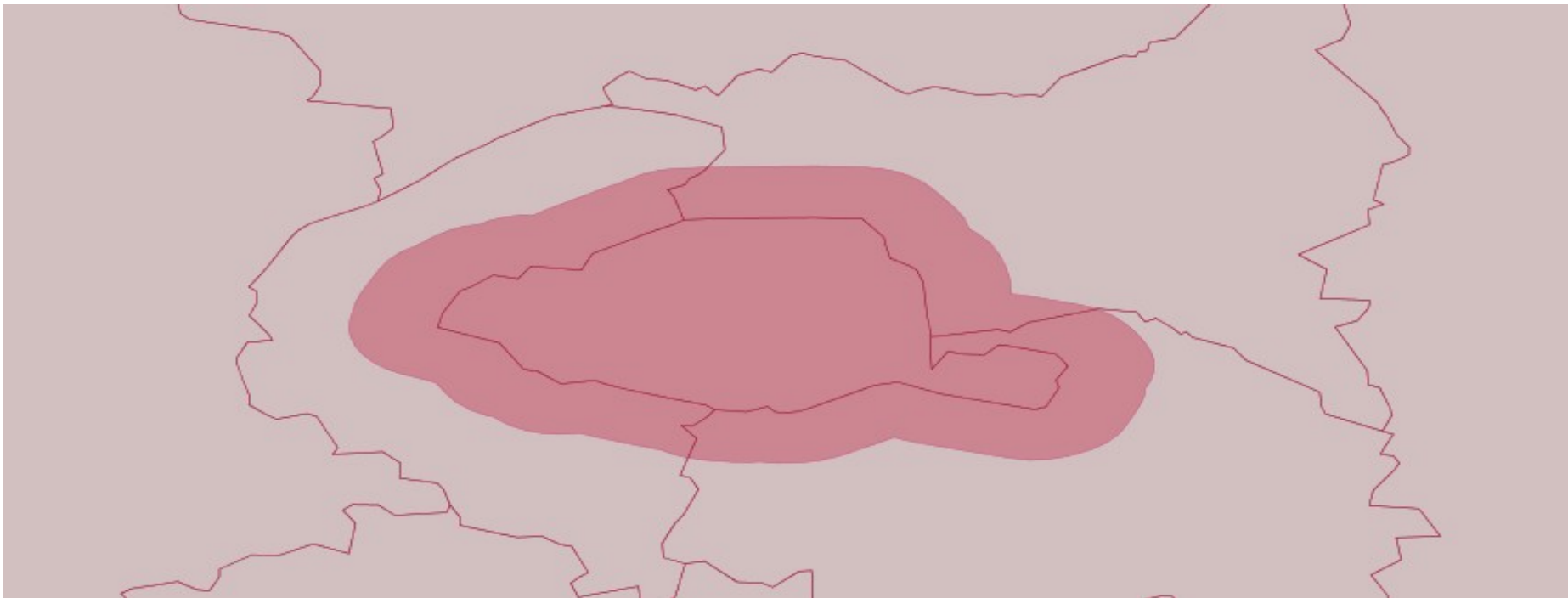
Fonctionnalités : Export GeoJSON

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



Fonctionnalités : ST_Buffer

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



Fonctionnalités : Aggrégation de géométries



Les communes de France



Les communes de France fusionnées par département

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

Fonctionnalités : **ST_Intersects** et sous requête

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

Résultat:

```
PARIS
HAUTS-DE-SEINE
SEINE-SAINT-DENIS
VAL-DE-MARNE
```



Fonctionnalités : ST_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;
```

Résultats:

```
75 | 0  
92 | 7  
93 | 12  
94 | 13
```



Fonctionnalités : Création de géométries

```
SELECT nom_dept
FROM dept
WHERE St_Within(
    GeometryFromText ( ' POINT(600440 2428685) ' , 27572) ,
    the_geom) ;
```

Resultat: PARIS



```
SELECT ST_AsText (  
    ST_GeomFromGML (  
        '<gml:Point srsName="EPSG:27572">  
            <gml:pos srsDimension="2">  
                600440 2428686  
            </gml:pos>  
        </gml:Point>'  
    )  
);
```

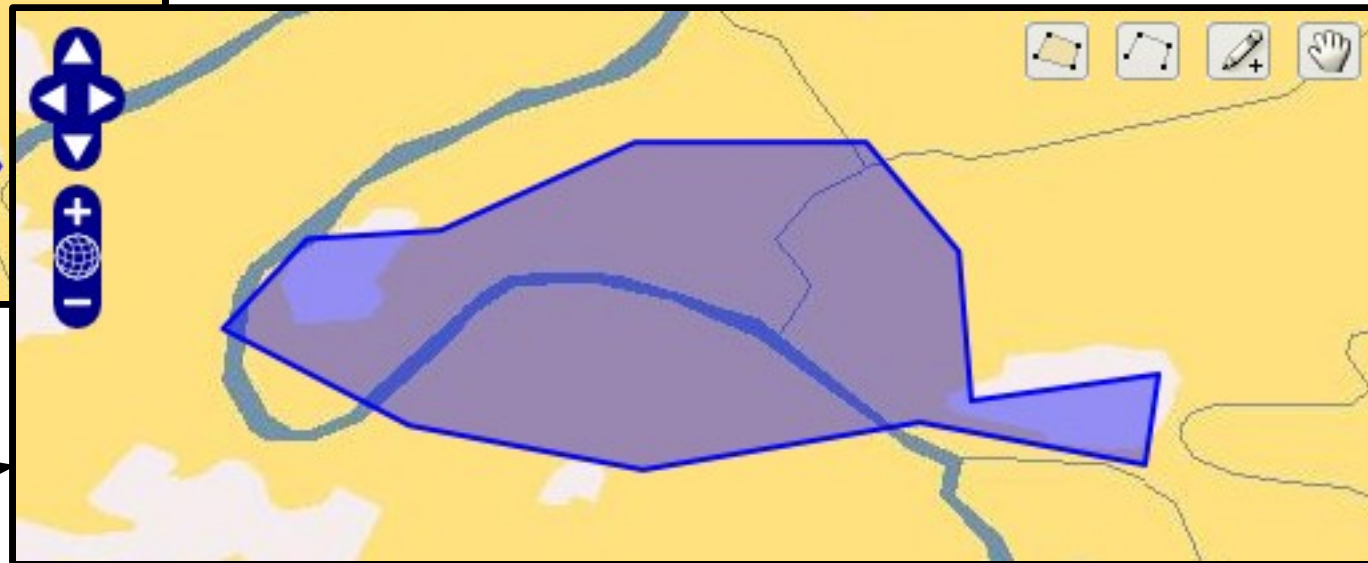
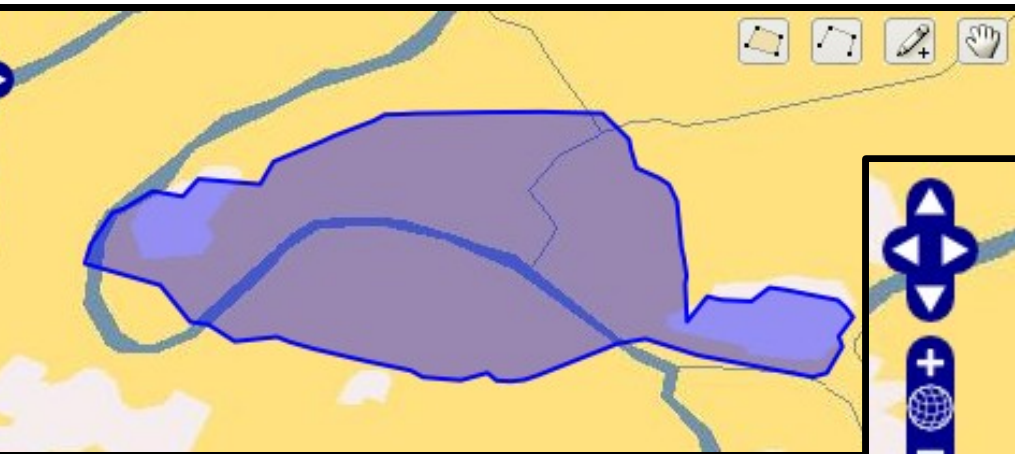
Résultat: POINT(600440 2428686)



Fonctionnalités : Généralisation via ST_Simplify

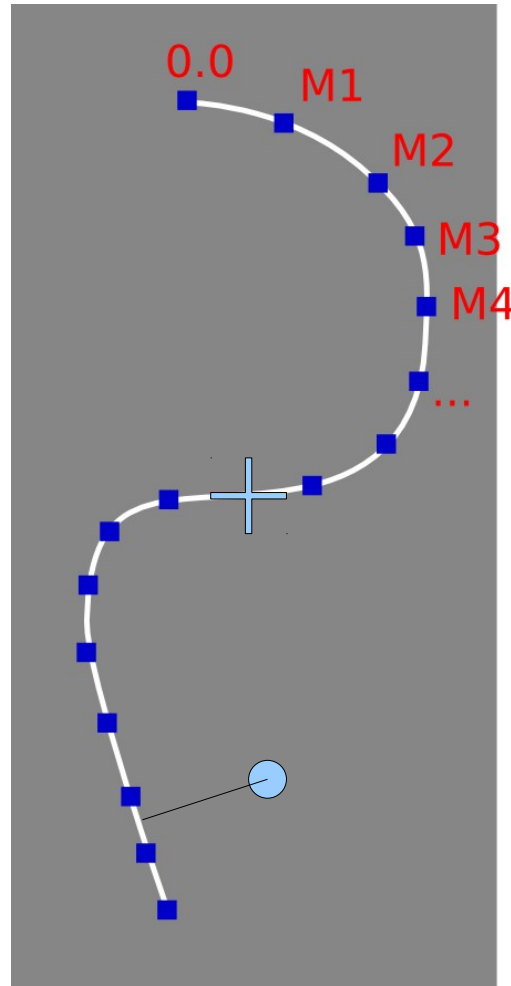
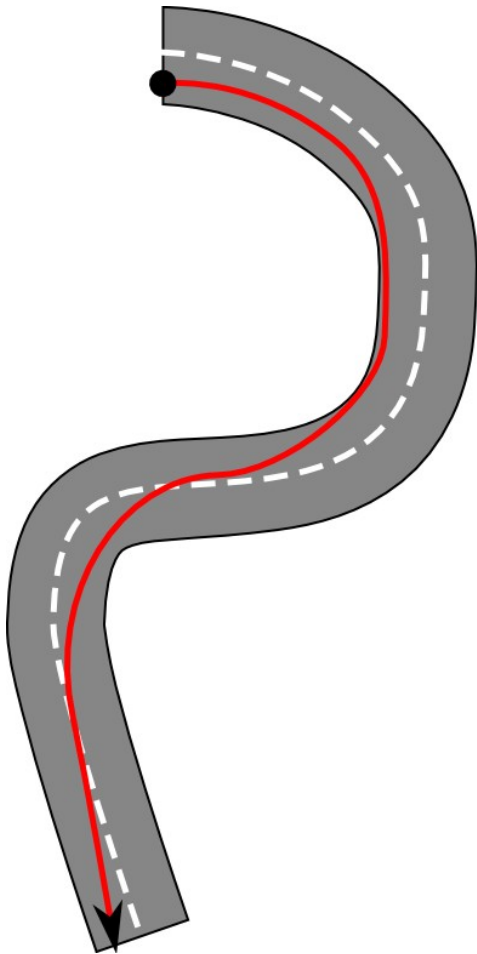
Algorithme Douglas-Peucker

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



Fonctionnalités : référencement linéaire

Fonctions de référencement Linéaire (cas de tronçons routiers)



`ST_line_interpolate_point(linestring, location)`

`ST_line_substring(linestring, start, end)`

`ST_line_locate_point(LineString, Point)`

`ST_locate_along_measure(geometry, float8)`

Fonctionnalités : pgRouting

PgRouting, un module additionnel de parcours de graphe



Yokohama (Japan)

Select Routing Method

Shortest Path A Star - undirected

- Add START point
- Add FINAL point

Route

Reverse

Reset

Geocode Address

No data available

Example: 神奈川県横浜市中区海岸通1-2

Geocode

Reset

Isoline (Driving Distance)

10000 [m] around START point

Isoline

Reset

PostGIS en 2009 : 1.4 et 1.5

1.4 Sortie fin Juillet 2009

1.5 Finalisée en 2010, sortie février 2010

Amélioration des performances

Séparation du code

 bibliothèque réutilisable (liblwgeom)

Nouvelles fonctionnalités

Bugfixes



PostGIS 1.4 : Performance de ST_Union

```
SELECT ST_Union(the_geom) from dept;
```

Version 1.3 Time: **10865 ms**

Version 1.4 Time: **2391 ms**



PostGIS 1.4 : Refonte du manuel

Rédacteurs:
Regina Obe et
Kevin Neufeld

Name

`ST_Simplify` — Returns a "simplified" version of the given geometry using the Douglas–Peucker algorithm.

Synopsis

```
geometry ST_Simplify(geometry geomA, float tolerance);
```

Description

Returns a "simplified" version of the given geometry using the Douglas–Peucker algorithm. Will actually do something only with (multi)lines and (multi)polygons but you can safely call it with any kind of geometry. Since simplification occurs on a object-by-object basis you can also feed a GeometryCollection to this function.

Note!

Note that returned geometry might loose its simplicity (see [ST_IsSimple](#))

Note!

Note topology may not be preserved and may result in invalid geometries. Use (see [ST_SimplifyPreserveTopology](#)) to preserve topology.

Performed by the GEOS module.

Availability: 1.2.2

Examples

A circle simplified too much becomes a triangle, medium an octagon,

```
SELECT ST_Npoints(the_geom) As np_before, ST_NPoints(ST_Simplify(the_geom,0.1)) As np01_notbadcircle, ST_NPoints(ST_Sir
ST_NPoints(ST_Simplify(the_geom,1)) As np1_octagon, ST_NPoints(ST_Simplify(the_geom,10)) As np10_triangle,
(ST_Simplify(the_geom,100) is null) As np100_geometrygoesaway
FROM (SELECT ST_Buffer('POINT(1 3)', 10,12) As the_geom) As foo;
--result
np_before | np01_notbadcircle | np05_notquitecircle | np1_octagon | np10_triangle | np100_geometrygoesaway
-----+-----+-----+-----+-----+-----
         49 |                33 |                17 |                9 |                4 | t
```



PostGIS 1.4 : populate_geometry_columns

Synchro entre les colonnes géométriques
et la table 'geometry_columns'

```
SELECT * FROM geometry_columns;
public | dept | the_geom | 2 | 27572 | MULTIPOLYGON
```

```
ALTER TABLE dept RENAME TO foo;
SELECT populate_geometry_columns();
```

```
SELECT * FROM geometry_columns;
public | foo | the_geom | 2 | 27572 | MULTIPOLYGON
```



PostGIS 1.4 : populate_geometry_columns II

```
CREATE VIEW myview AS
  SELECT gid,
         ST_Buffer(the_geom, 2500) AS the_geom
  FROM france
  WHERE code_dept='75';
```

```
SELECT populate_geometry_columns();
```

```
SELECT * FROM geometry_columns
WHERE f_table_name='myview';
```

```
|public|myview|the_geom|2|27572|POLYGON
```



PostGIS 1.5

Type 'geography' : latitude, longitude

= «support géodésique»

Fonctionnalités pour ce type

Surface, distance, indexation...

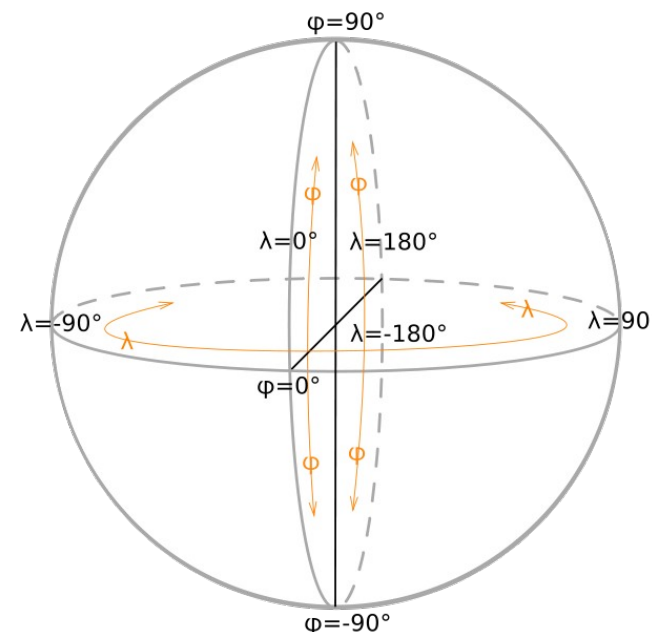
Fonctions d'import

GML

KML

GeoJSON

Bugfixes



Nouveautés : la 2.0

Sortie envisagée fin 2010

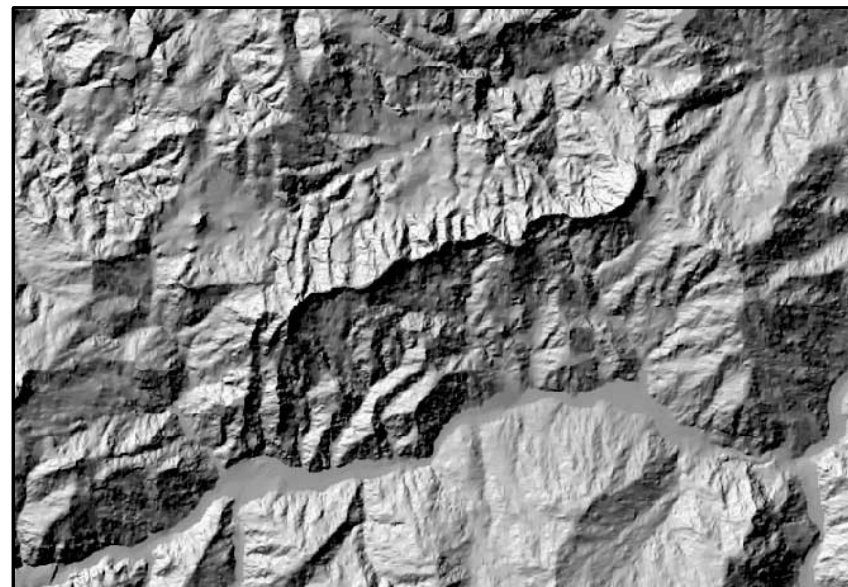
Chantiers en cours ou prévus :

WKTRaster

images manipulables depuis la base

Primitives 3D

Fonctionnalités de
topologie et graphes



Fin – Questions ?

Merci de votre attention

Questions, informations :
infos@oslandia.com

Oslandia :
www.oslandia.com

