



PostGIS, a PostgreSQL module for spatial data

Vincent Picavet

PGDay.eu 2010

Who ?

Vincent Picavet

Former Makina Corpus employee (french FS SME)

Owner & Manager @ Oslandia (with Olivier Courtin)

OpenSource addict

PostGIS contributor



Oslandia, a free spatial SME

Services

Training
Support
Consulting
Development

Focus on spatial softwares

Spatial databases
Complex processes and analysis
routing, geocoding...
OGC and INSPIRE Web Services
Desktop client : QGIS

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



Presentation

Context

GIS

Examples

PostGIS

Project and actors

Spatial types

Spatial index

Spatial functions

Evolution

PostGIS in 2009

PostGIS in 2010

Questions



Context: GIS ?

80% of data is spatial

Geographical Information System

Capture

Store

Exchange

Analyze

Visualize



Originally : local government

Today : Public & Private sectors



Context : GIS architecture

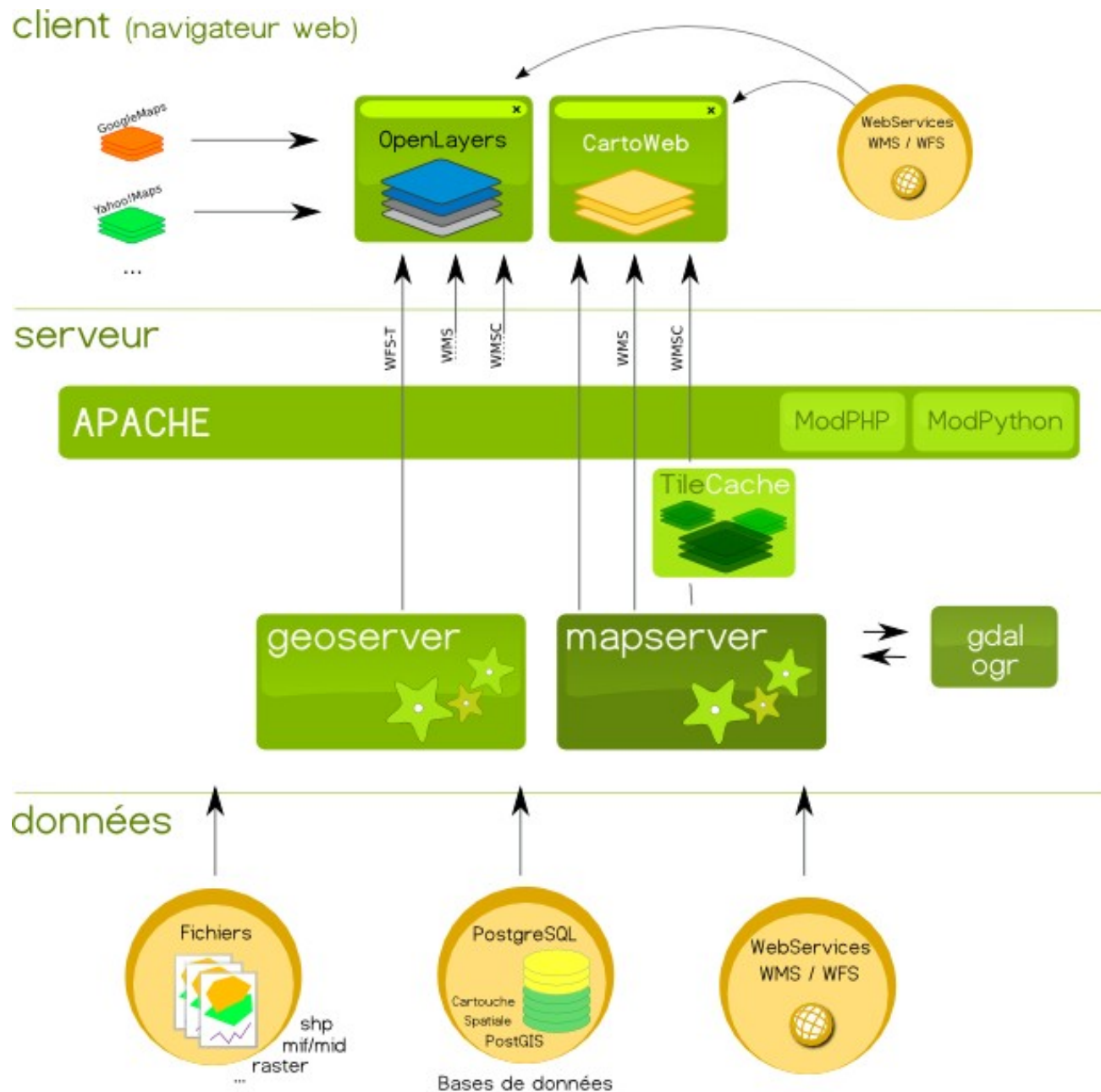
Distributed architecture

Clients

web
desktop

Data exchange

Standard WS



Context : A car-sharing service

BisonVert.net

Car-sharing free software

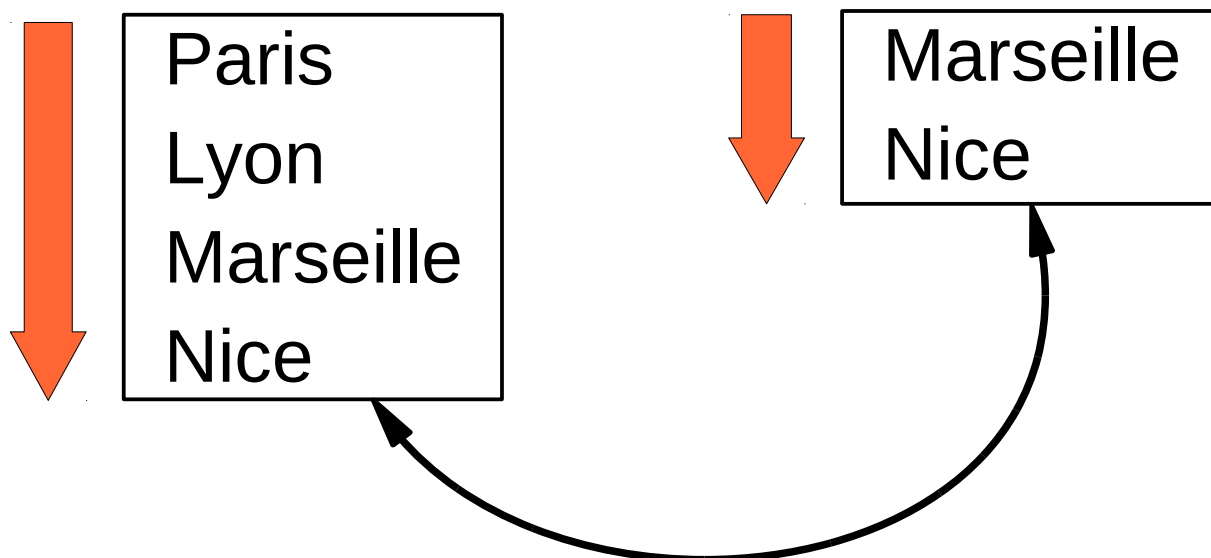


Goal :

Match people doing the same journey

Current method :

Match from/to/via names



► we can do better !



Context : spatial car-sharing

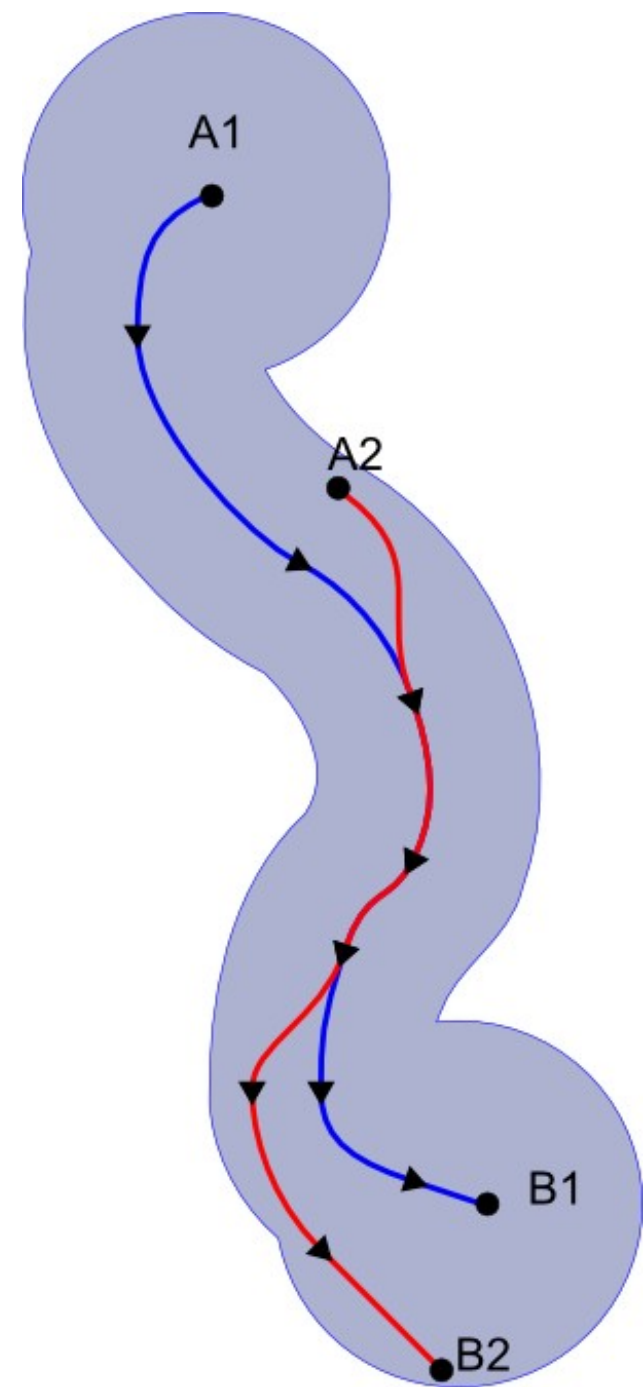
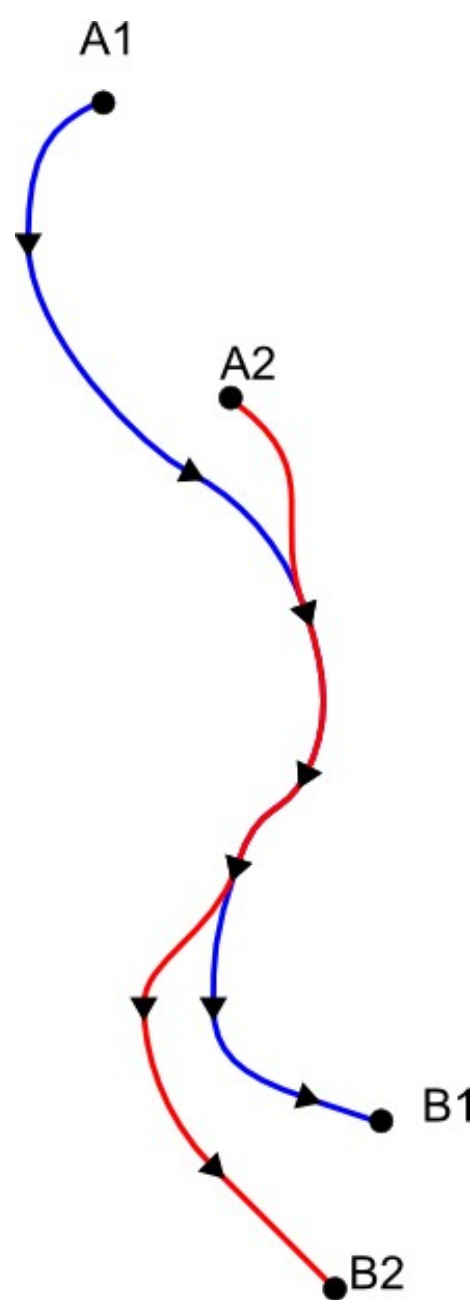
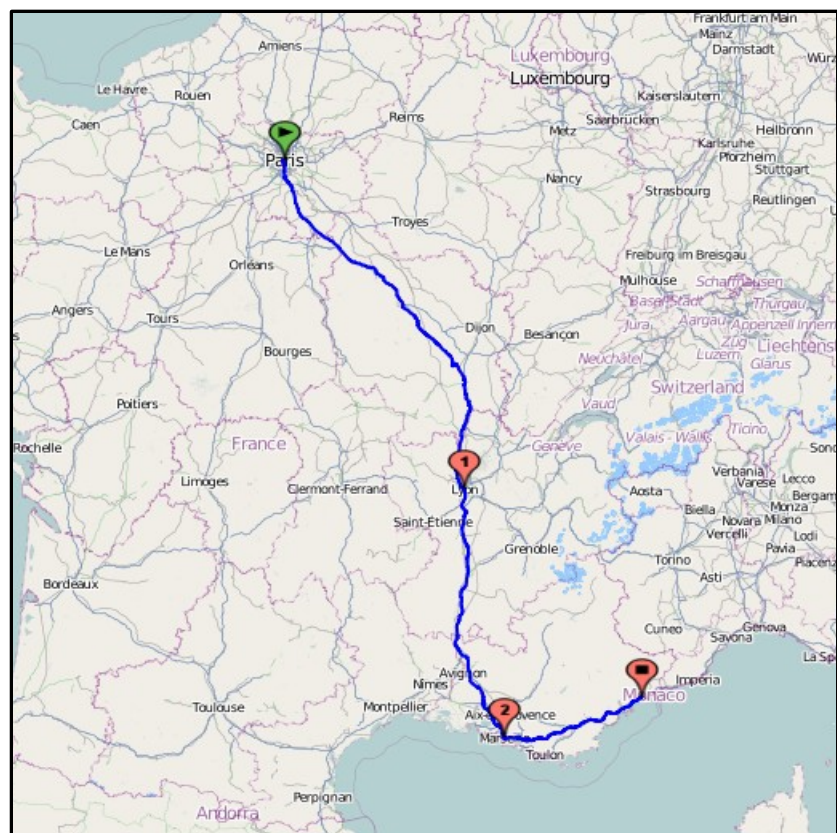
Solution :

Use real paths

1/ Compute path (routing)

2/ Match paths

(Spatial analysis)



Context : Main goals of a spatial DBMS

Geometry + attributes = «**feature**»

SQL Queries

- Filter on attribute part

- Filter on geometry part

Manage load

- Huge volumes of data

- Complex and long analysis and processes

Reasonable performance

Follow standards



Context : International standards

Specifications

OGC SFS (Simple Feature for SQL)
ISO SQL/MM part 3



What is specified

Supported geometry types
Spatial functions prototypes
Additional tables and referential integrity management

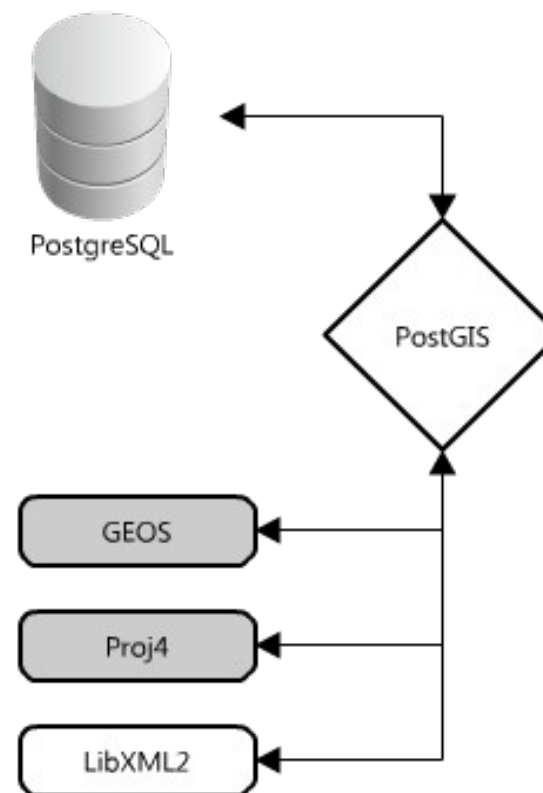


PostGIS : Principle and architecture

PostgreSQL plugin

Mainly written in C

Uses some external libraries :



Implements OGC SFS 1.1 (and part of ISO SQL/MM)

A lot of additional functions

PostGIS : Project history

2001

First alpha version

2003

Version 0.8 – Ready for production

2005

Version 1.0

Core rewriting and LWGEOM
Compatible with OGC SFS 1.1

2006

Version 1.2

Aims to ISO SQL/MM (Curves, ST_... prefixes)

2009

Version 1.4

PSC is born, OSGeo project

2010

Version 1.5

... release 2.0 ?



PostGIS : Community

Institutions

IGN : Institut Géographique National

IRSN : Institut de Radioprotection et de Sûreté Nucléaire

JRC : Joint Research Center – Union Européenne

...

French big companies

Mediapost

France Telecom

...

Community

Worldwide

Thousands of users

Very active postgis-users mailing list



PostGIS : PostGIS committers



LisaSoft

OpenGeo

Oslandia

CadCorp

Paragon Corporation

Refractions Research

Sandro Santilli

Sirius



PostGIS : Other spatial DBMS

Oracle Spatial (et Locator)



ESRI ArcSDE

IBM DB2

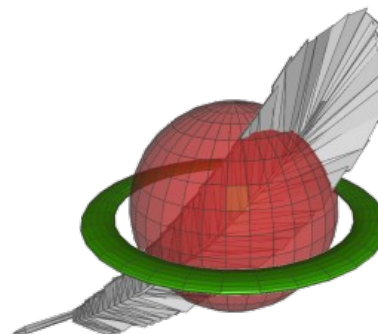


Microsoft SQLServer (> 2008)



Spatialite

Ingres



Sybase (last version)



Geometry : Representation and storage

Geometry (or HEWKB)

- Native database storage

- Binary format with hexadecimal encoding

WKT (Well Known Text)

- Textual representation

Dimensions

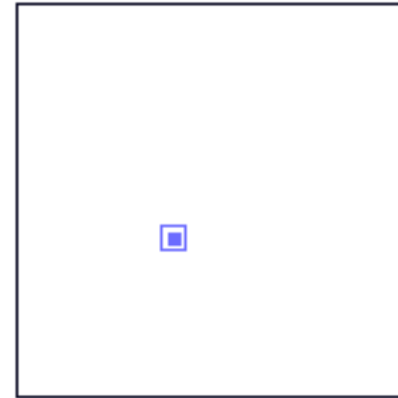
- 2D, 3D, or 4D

Projection system id (SRID)



Geometry : Point

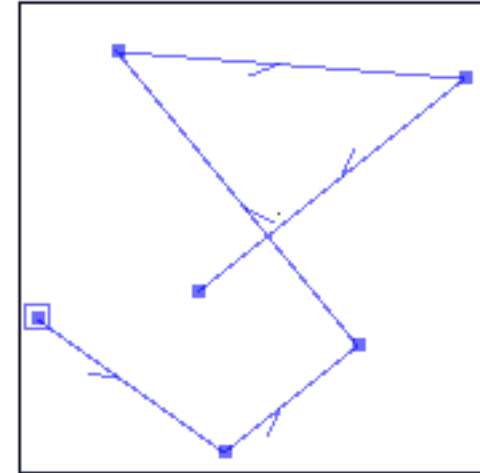
POINT (10 10)



Geometry : LineString

LINESTRING

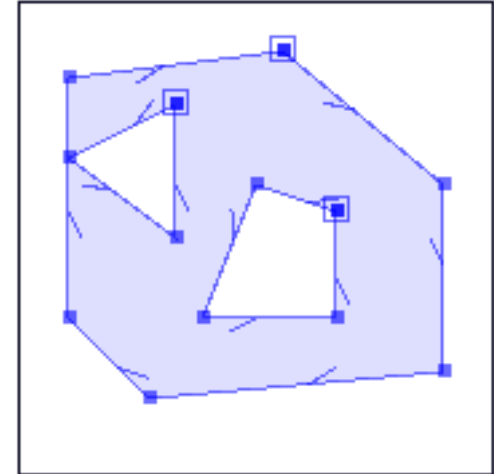
```
(  
  0 5, 5 1, 9 4, 2 14, 14 13, 4 4  
)
```



Geometry : Polygon

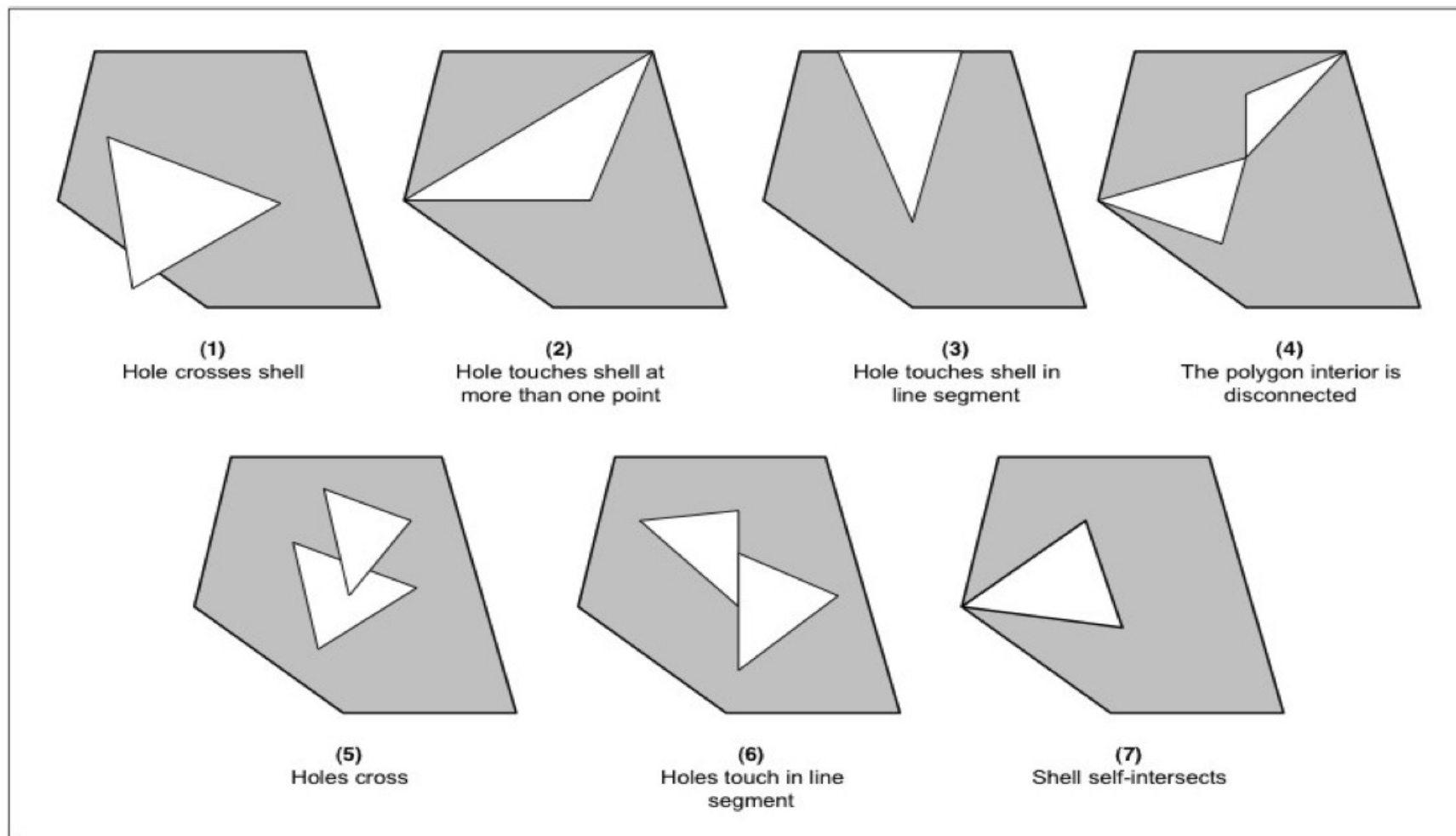
POLYGON

```
(  
  (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) Mandatory first ring is external ring
- 2) Rings coordinates must be closed

Geometry : Polygons and SFS validity



Invalid types (according to OGC SFS)

Schema from JTS website (vivid solutions.com)

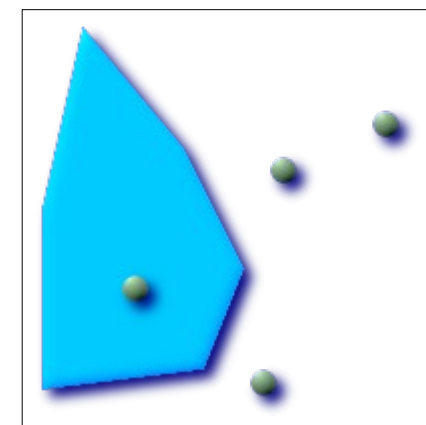
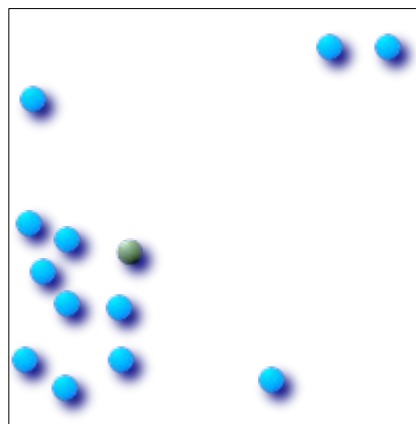
Geometry : Multiples and aggregates

MULTIPOINT

MULTILINESTRING

MULTIPOLYGON

GEOMETRYCOLLECTION



- 1) Different projection systems cannot be mixed
- 2) Neither can different dimensions

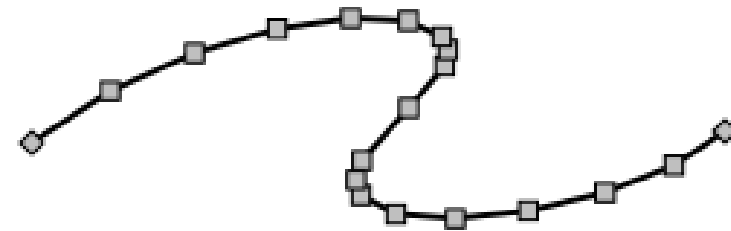
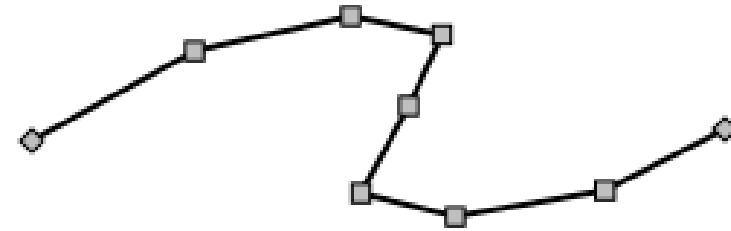
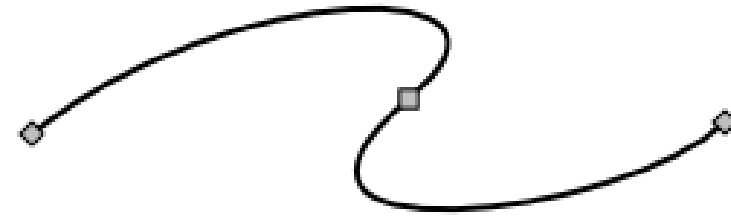
Geometry : curves

«curves» types :

CIRCULARSTRING

COMPOUNDCURVE

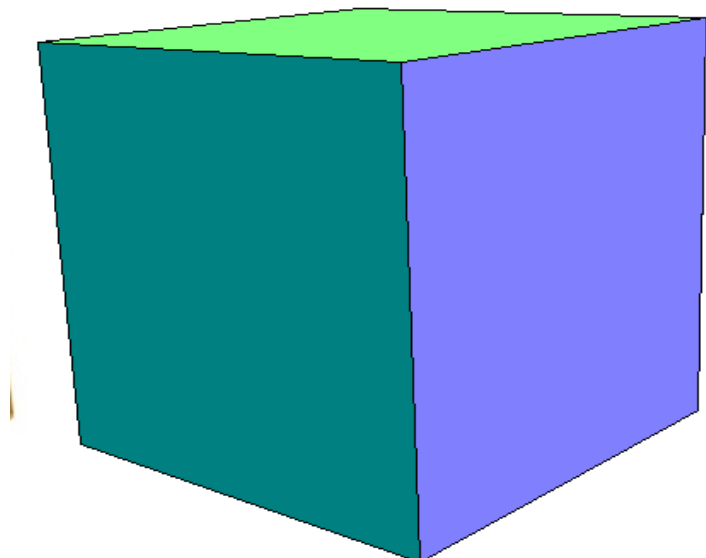
MULTISURFACE



Geometry : 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 in database : additional tables

geometry_columns : spatial fields catalog

	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: projection systems catalog

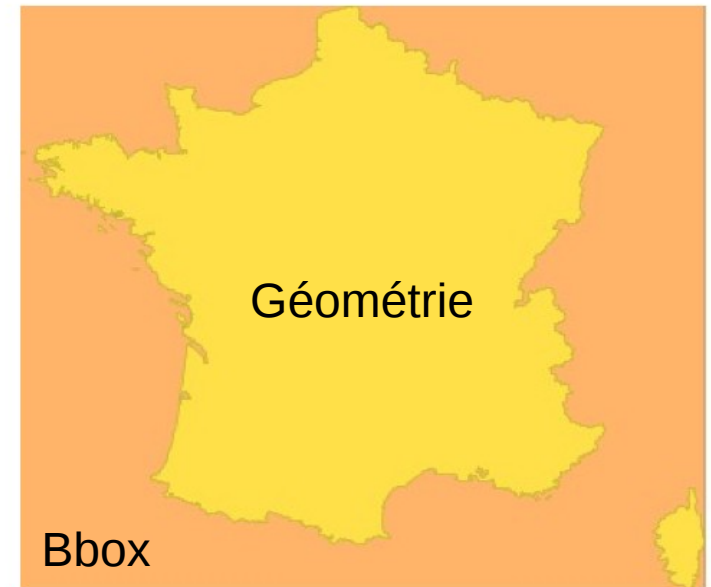
	srid [PK] integer	auth_name character var	auth_srid integer	srttext character varying(2048)	proj4text character varying(2048)
1	2000	EPSG	2000	PROJCS["Anguilla 1957 / British We	+proj=tmerc +lat_0=0 +lon_0=-62 +k=0.999
2	2001	EPSG	2001	PROJCS["Antigua 1943 / British We	+proj=tmerc +lat_0=0 +lon_0=-62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

Hint : can be interesting to store these tables in a different schema

Spatial index : principle & creation

Better spatial filter performance

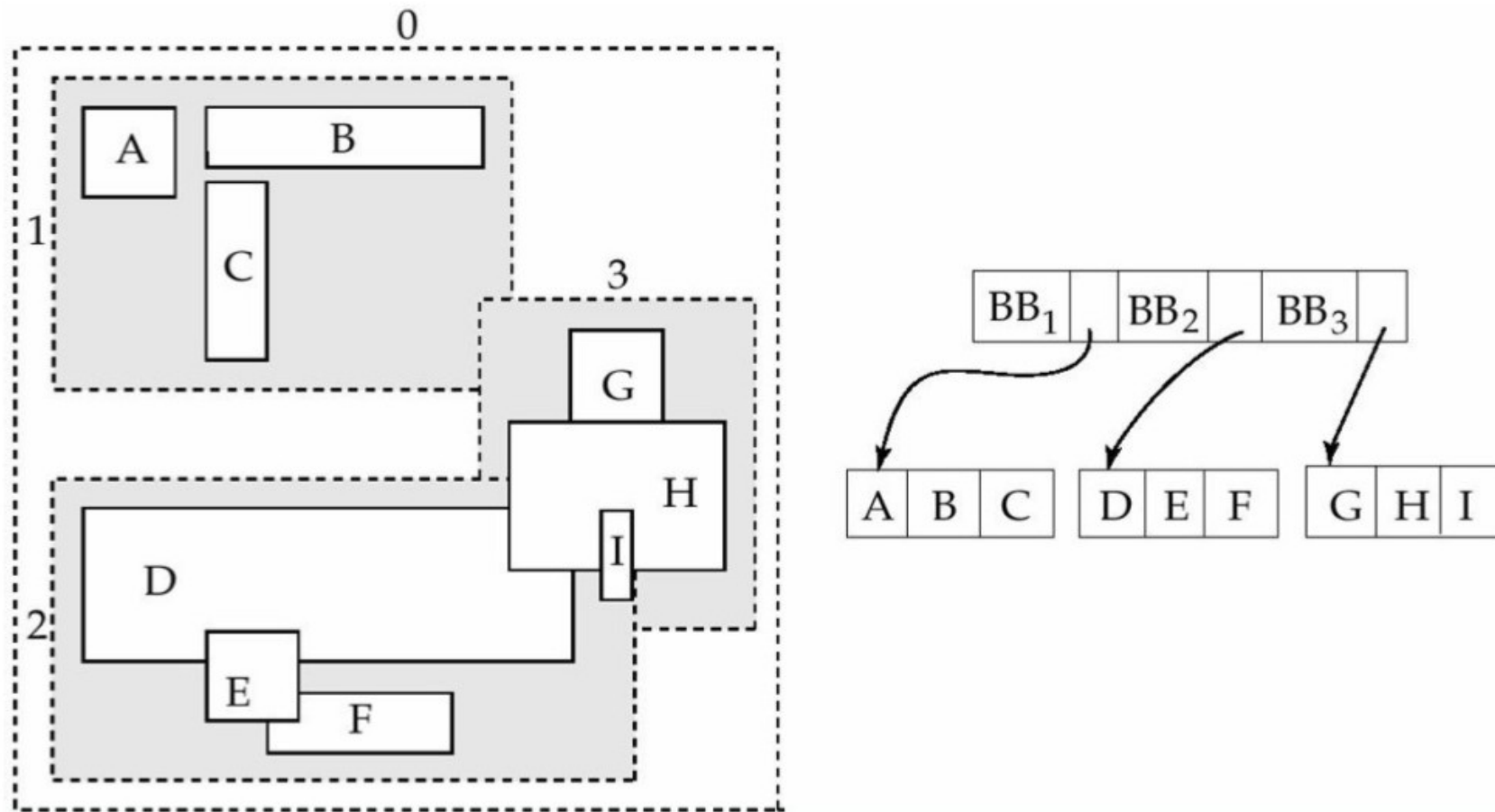
Geometry approximation with bbox



Spatial index creation :

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

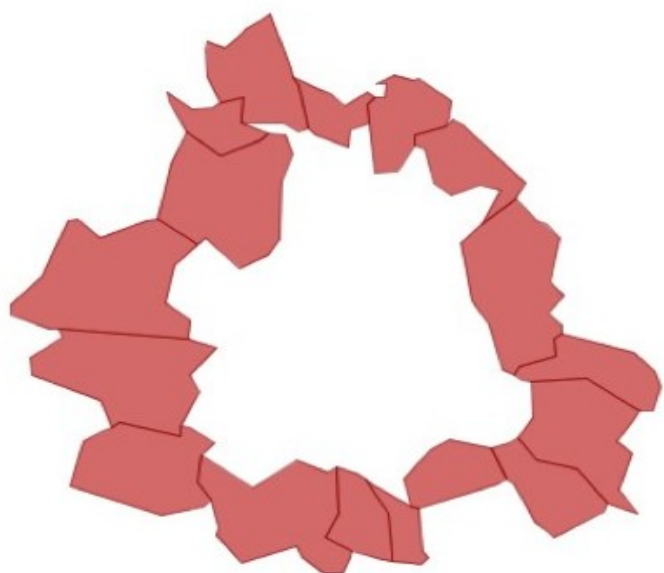
Spatial index : R-tree



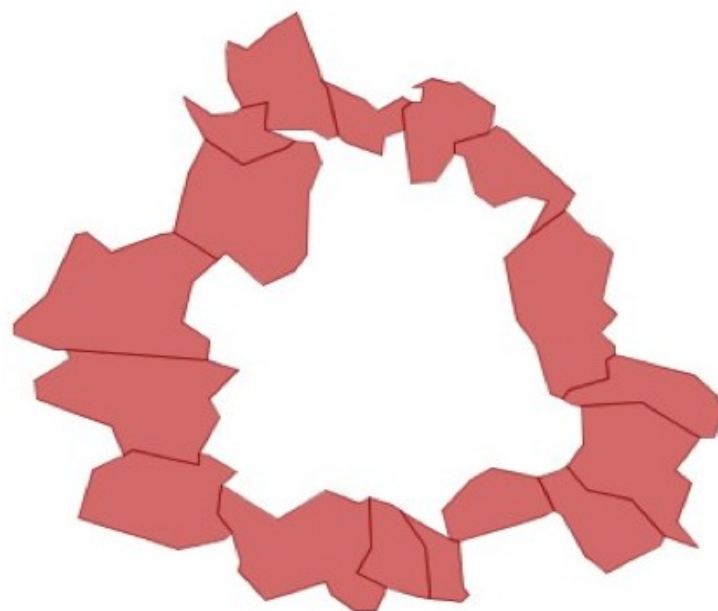
Bounding boxes are grouped in regions of the index

Spatial index

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



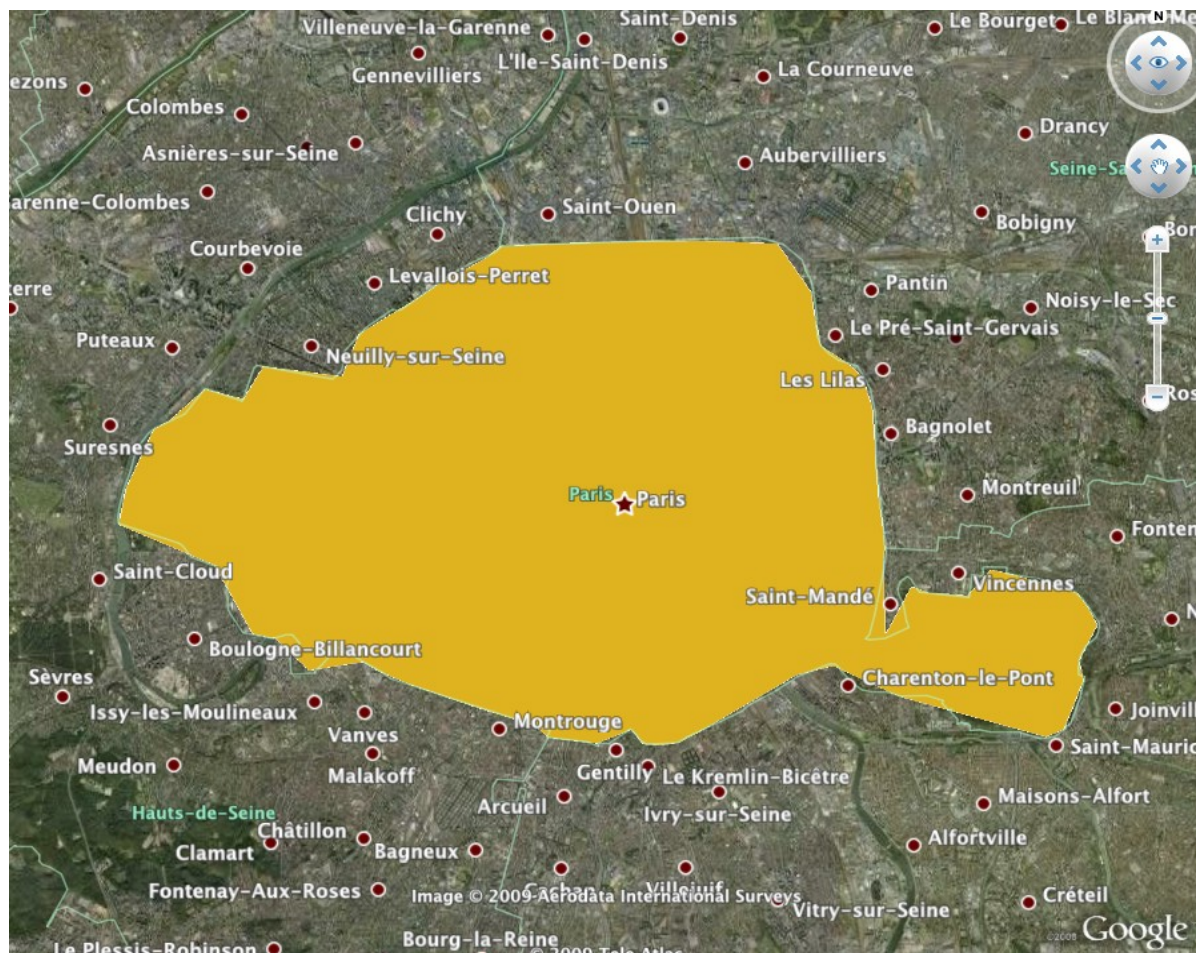
Sans index: temps = 150 ms



Avec index: temps = 30 ms

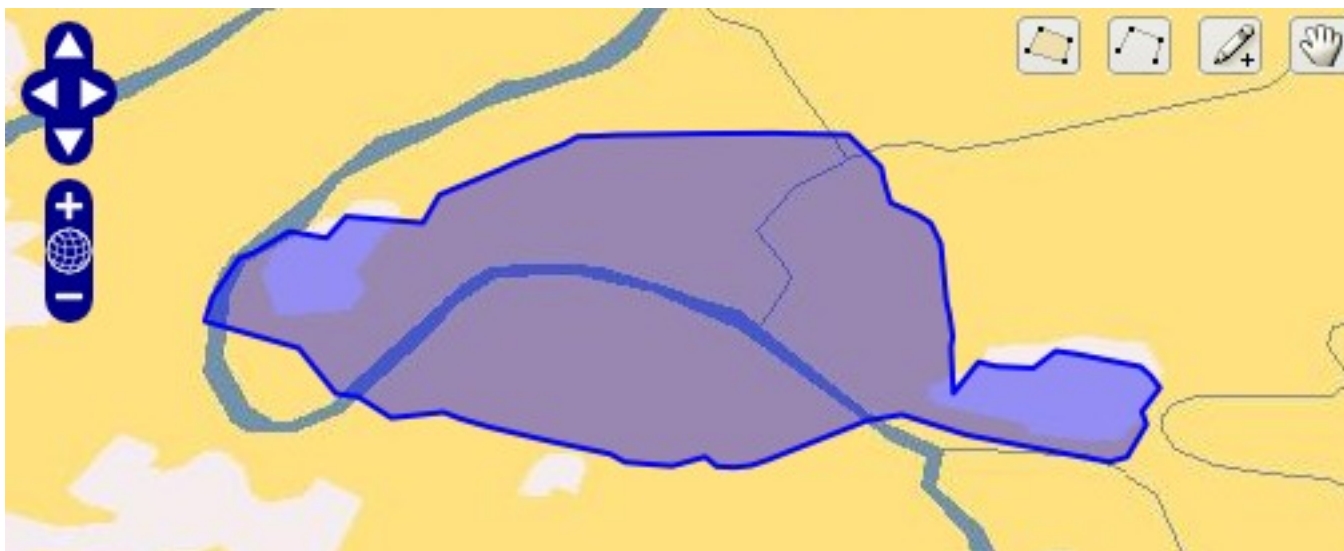
Postgis functions : KML export

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



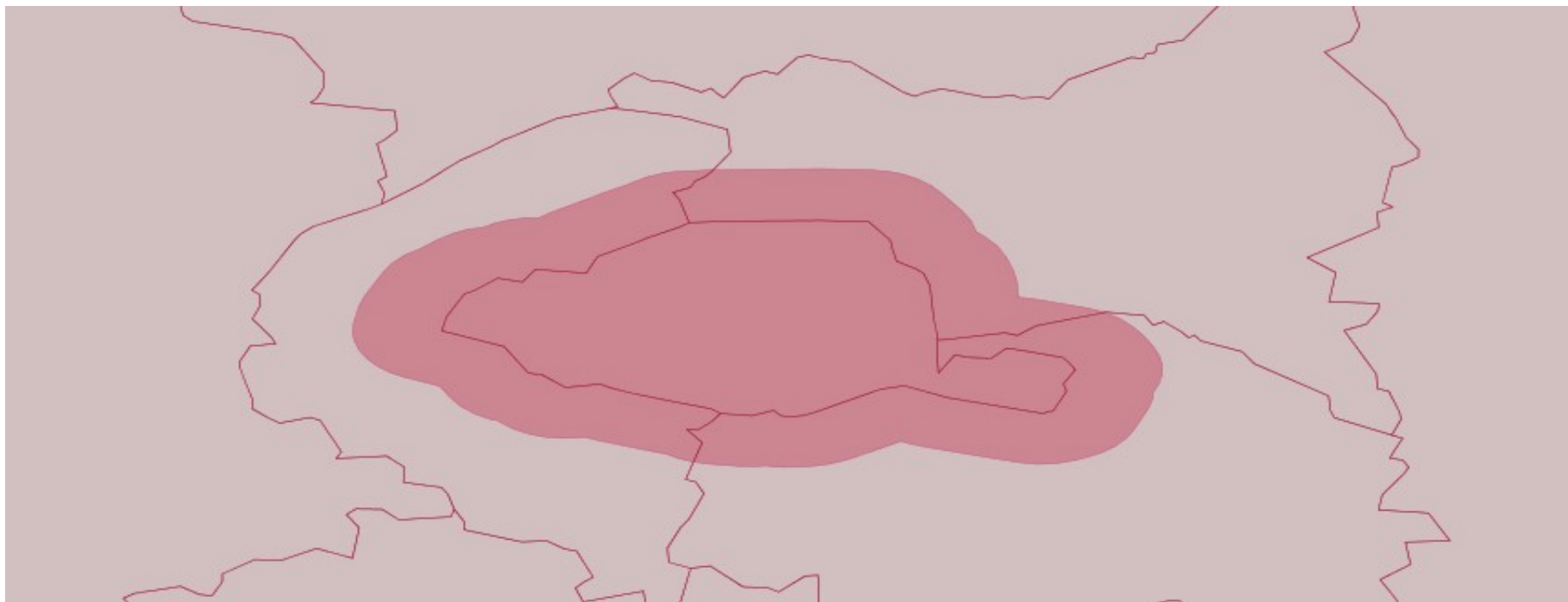
PostGIS functions : GeoJSON export

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



PostGIS functions : buffer

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



PostGIS functions : geometry aggregate



Les communes de France



Les communes de France fusionnées par département

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

PostGIS functions : intersection, subquery

```
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 : 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;
```

Results:

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



PostGIS Functions : create geometry

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

Result : PARIS



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

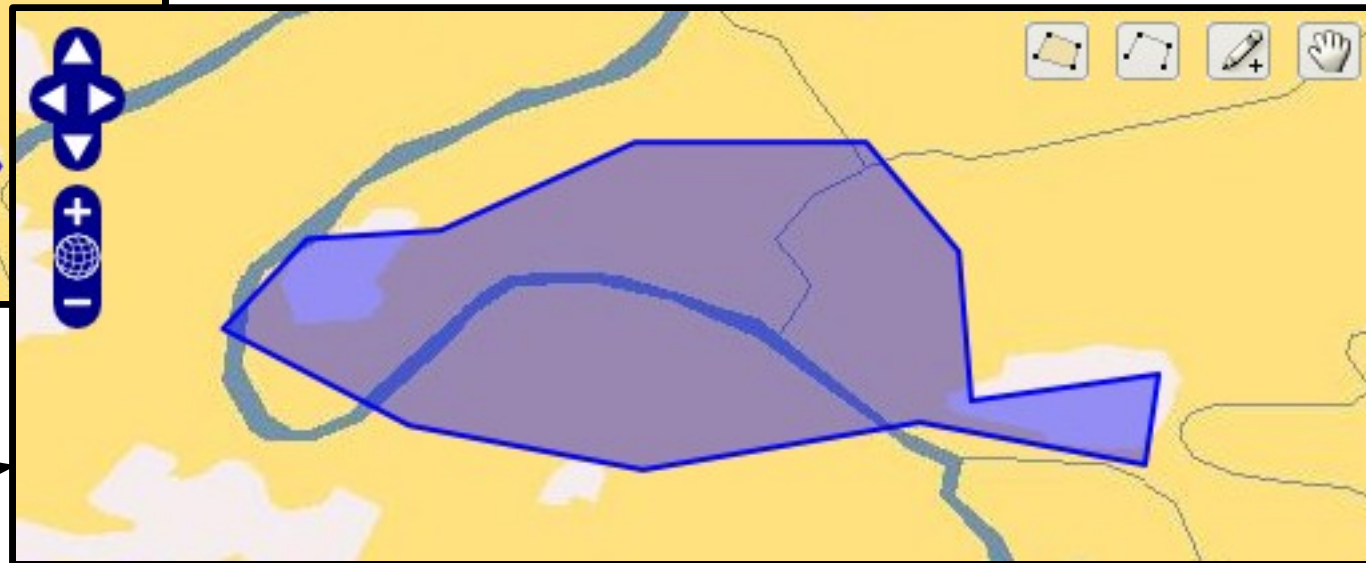
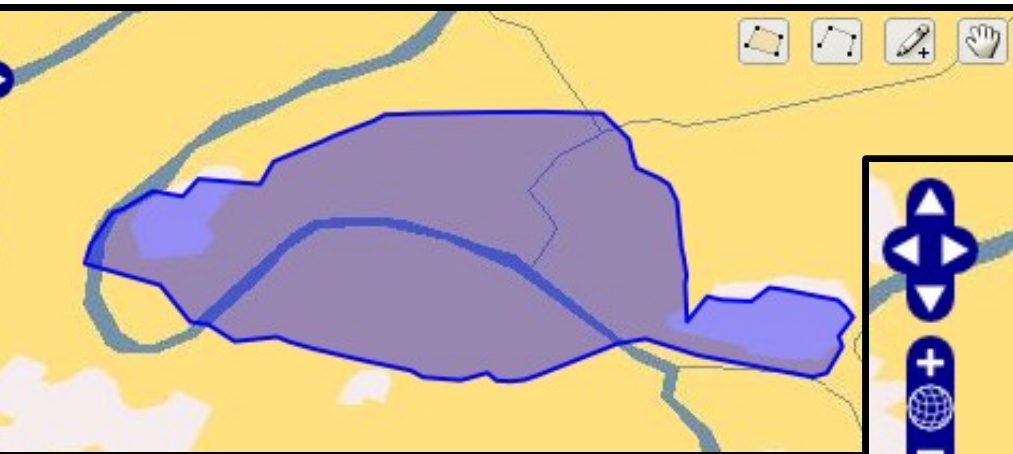
Result: POINT(600440 2428686)



PostGIS functions : Generalization via ST_Simplify

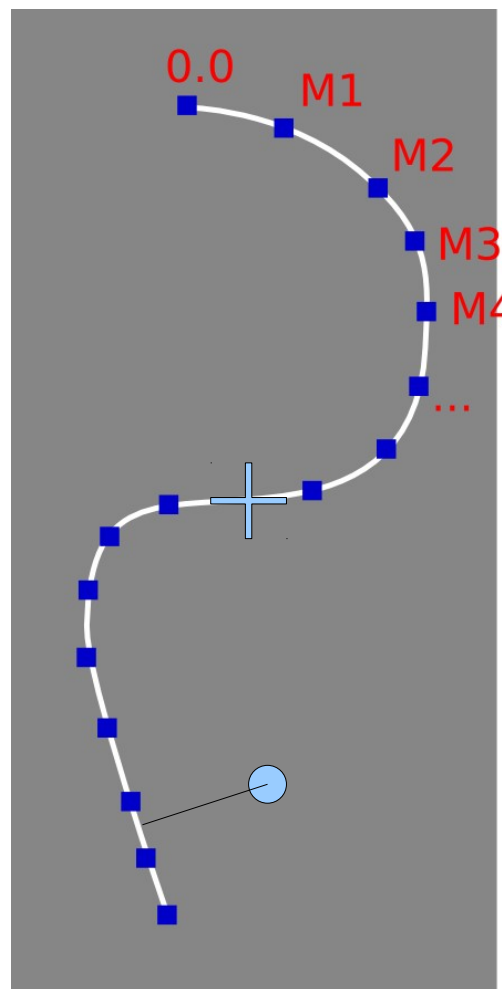
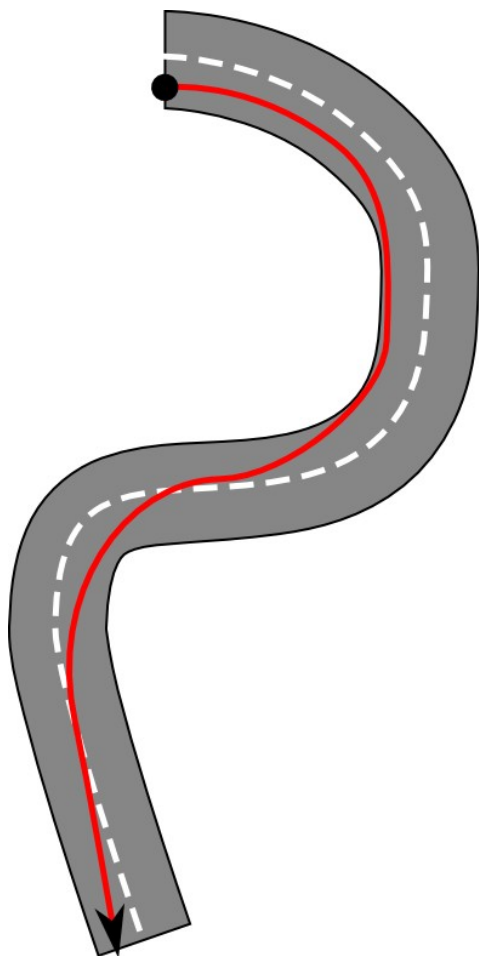
Algorithm : Douglas-Peucker

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



PostGIS functions : linear referencing

Functions for linear referencing (Road network for example)



`ST_line_interpolate_point(linestring, location)`

`ST_line_substring(linestring, start, end)`

`ST_line_locate_point(LineString, Point)`

`ST_locate_along_measure(geometry, float8)`

PostGIS Functions : pgRouting

PgRouting, an additional module for graph routing



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 in 2009 : 1.4 and 1.5

1.4 released end of july 2009

1.5 released in february 2010

1.5.2 released in september 2010 (minor fixes)

Performance improvement

Code refactoring

- Reusable library (liblwgeom)

New functions

Bugfixes



PostGIS 1.4 : ST_Union performances

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

Version 1.3

Time: **10865 ms**

Version 1.4

Time: **2391 ms**



PostGIS 1.4 : Manual improvement

Documentation :
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

Keeps metadata table 'geometry_columns' up to date

```
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

= «geodetic support»

Functions for this type

Area, distance, indexation...

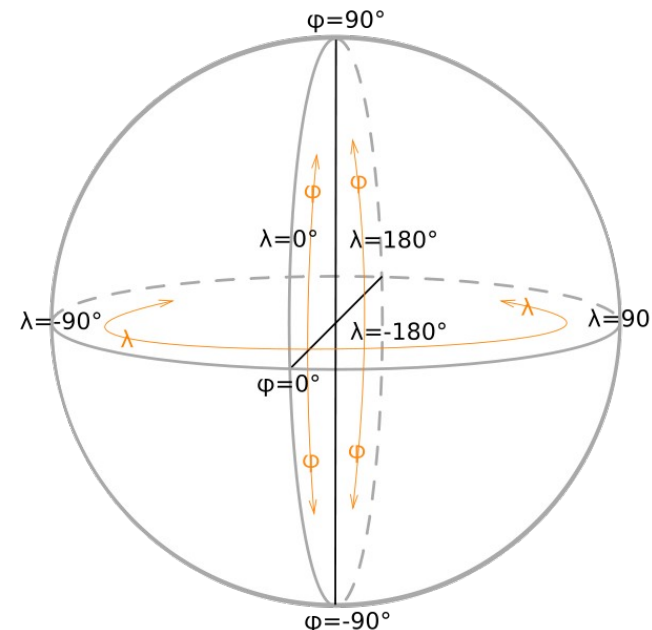
Import and export functions

GML

KML

GeoJSON

Bugfixes



What's coming : 2.0

Probably out around spring 2011

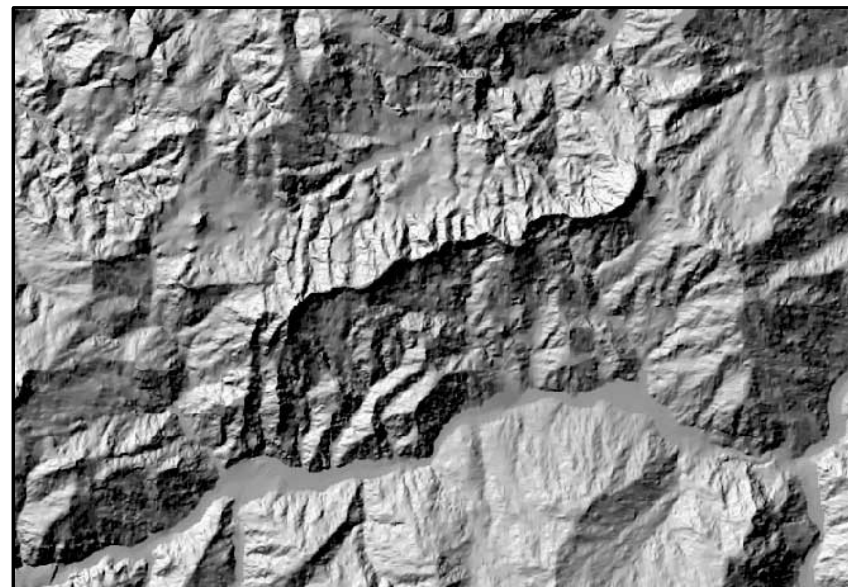
On-going work :

WKTRaster

Raster in database manipulation

3D storage and primitives

topology and graphs



End – Questions ?

Thanks for your attention

Questions, informations :
infos@oslandia.com

Oslandia :
www.oslandia.com

PGDay.eu feedback :
[Http://2010.pgday.eu/feedback](http://2010.pgday.eu/feedback)

