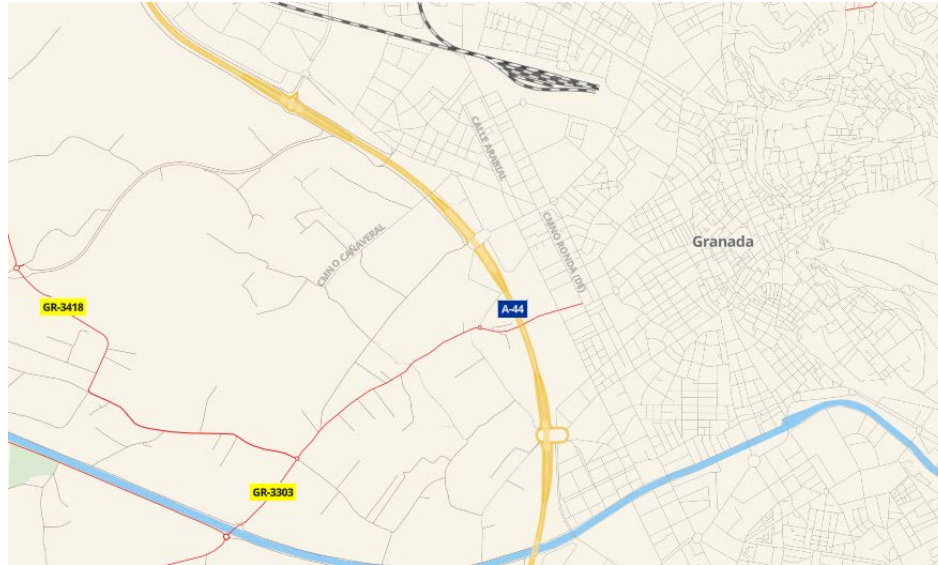# Exchange of experience
# of vector tiles service: pros and cons

Laura Alemany Gómez
laura.alemany@cnig.es
Centro Nacional de Información Geográfica (CNIG)
Spain

Exchange of experience
of vector tiles service: pros and cons

- Emilio López Romero
- Paloma Abad Power
- Laura Alemany Gómez
- Jose Maria García Malmierca
- Álvaro Bachiller

Exchange of experience
of vector tiles service: pros and cons

1) Introduction - What is Vector Tiles Service

2) Pros & cons VT

3) Creating a VT with Geoserver

Data: Download, simplify, PostGIS	Gridsets
What you need - Architecture	Layers
Data stores	Group layers
Styles	Tile layers

4) Comparing with other technologies

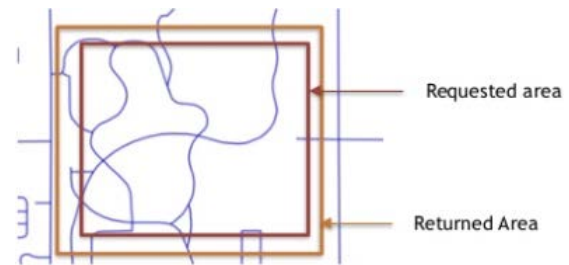**5) Client-side: Simbolizing VT (.json)**

6) Results

7) Visualization - Maputnik
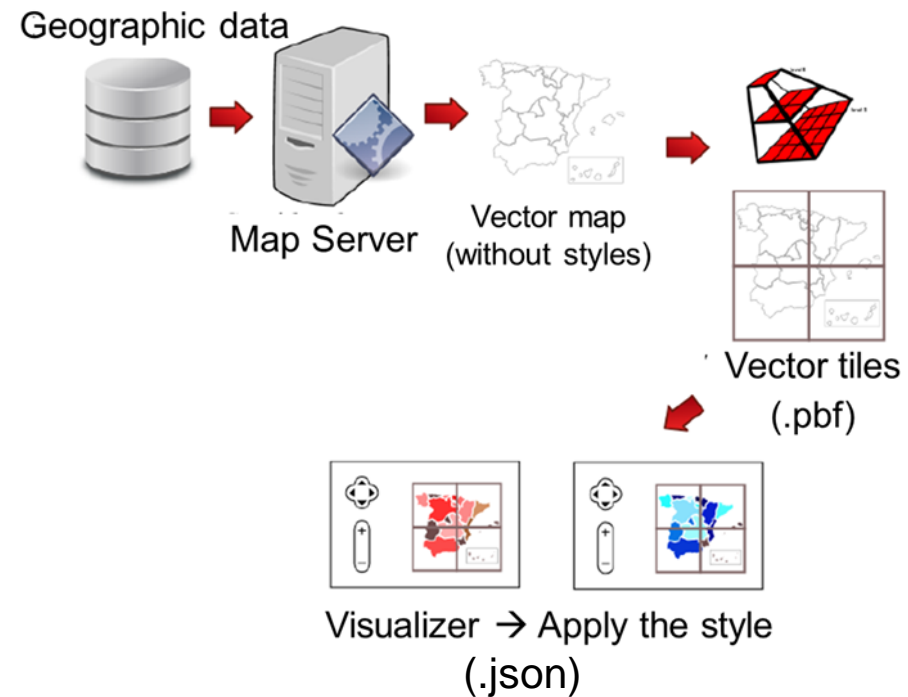
Inspire Helsinki 2019

3

Vector Tiles
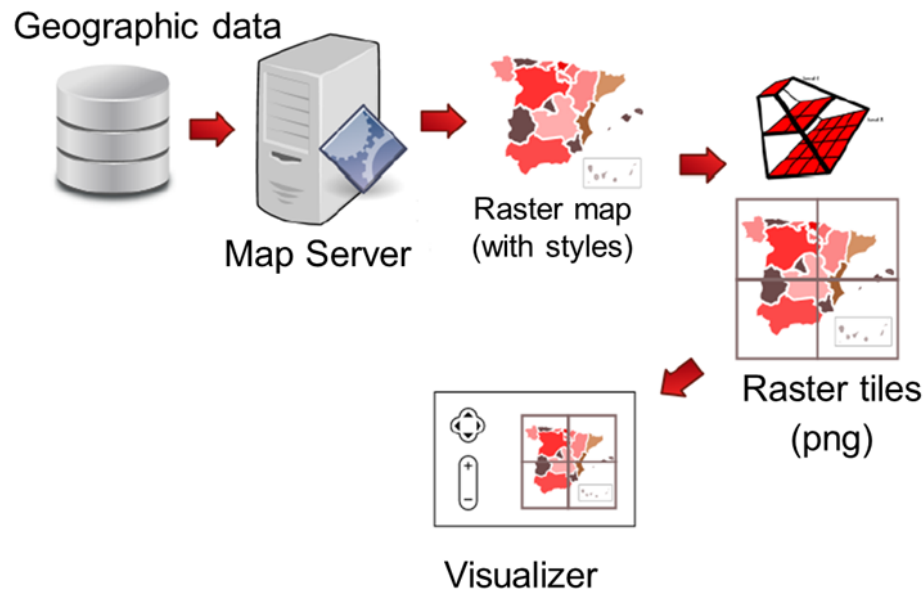
What is a Vector Tiles Service?

- A vector representation of geographic data covering a spatially contiguous and rectangular extent

Requested area

Returned Area

Vector Tiles

WMTS and Vector Tiles

Vector Tiles

WMTS and Vector Tiles Service

BOTH	
Belong to view services. Intended for visualization Based on **tiles** technology Can be served through the same interfaces	
WMTS	**Vector Tiles Service**
Deliver raster images	Deliver geographic data in a vector format split into squared tiles
Encoded in png, jpg	Geojson, topojson , pbf
+ satellite map	+ streets with labels available in many languages

VT – Pros & Cons

Vector Tiles Services

PROS:

According to Mapbox 75% less

- Smaller data size and therefore lower disk space requirement:
- The visualization is on Client side → More flexibility in application development
- **Client (not the server) decides on styling**: Font, if an element should be rendered…
- **Geometries** are delivered to the client.
- Only need to **tile data once** to have multiple styled maps
- Drawn vector can look better on high-resolution displays
        Allows overzoom without loosing resolution

CONS:

- The map is rendering on the **client side** and requires a bit **more powerful hardware**
        Nowadays this is not a problem
- Data is generalized and therefore **not suitable for direct edits**

Why Geoserver?

PROS

- Easy to learn and use. It has a graphic interface.
- No need of high knowledge of programming
- Most of our services use Geoserver and we know the technology
- It has a big community

CONS

- Less customisable and configurable than others.

Creating a Vector Tiles Service

0) DATA: Sources and conversions

1) Download Official Geographical Information

- Transport Network (CNIG)
- Hidrographic Network (CNIG)
- Buildings (D.G. Cadastre)
- Addresses(CNIG)
- Geographical names(CNIG)
- Administrative Units (CNIG, IHM)
- Forestal Map (Ministry of Environment)
- Protected Sites (Ministry of Environment)

Simplification

- Keep only fields needed to identify, classify and simbolize features.

    Such as name, class, ref, geometry…

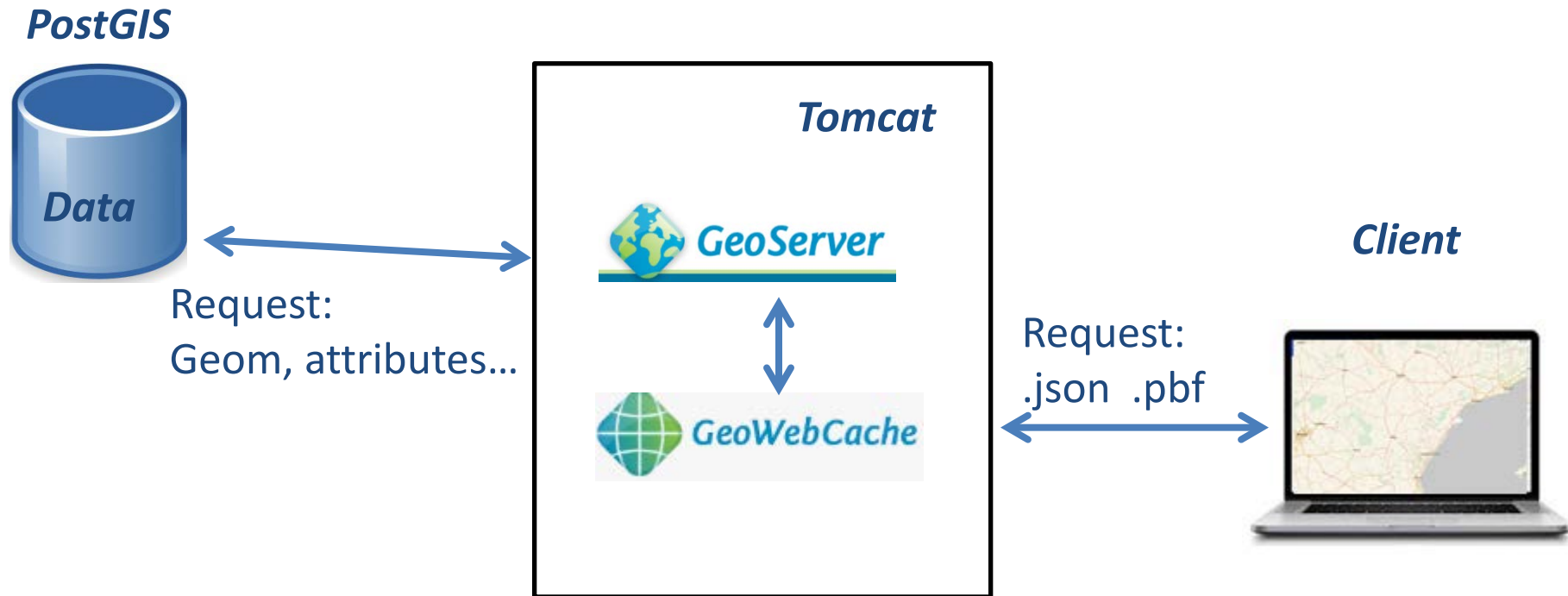## Import to PostGIS

- Easier management
- Higher performance

Creating a Vector Tiles Service

0) What you need

- Vector Tiles Extension for GeoServer ([Link](#))

Tile Image Formats
- ☑ application/json;type=geojson
- ☑ application/json;type=topojson
- ☑ application/x-protobuf;type=mapbox-vector
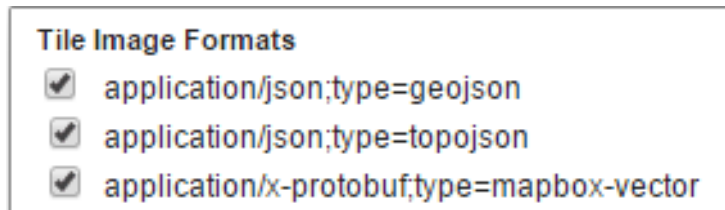
- [GeoWebCache](#) (Java)

  a) Integrated with GeoServer

  b) Also available as a standalone product

  – Tiling Server

  – Proxy between a map client and a map server
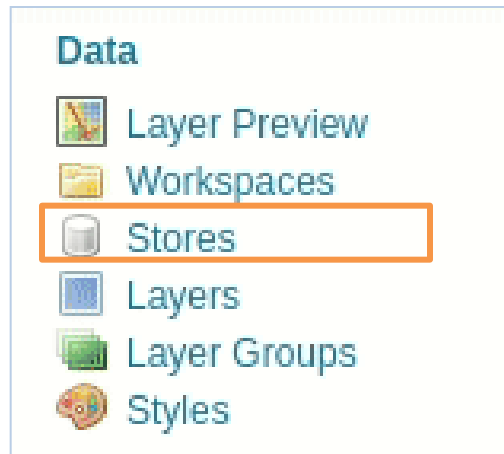
  – Caching and storing tiles as they are requested

Workflow - Architecture

PostGIS

Data

Tomcat

GeoServer

GeoWebCache

Client

Request:
Geom, attributes…

Request:
.json .pbf

If GWC has the tile→ It gives to the viewer
If not → Ask Geoserver, save it and give it to the viewer

Creating a Vector Tiles Service

1) Data Stores

Data

- Layer Preview
- Workspaces
- **Stores**
- Layers
- Layer Groups
- Styles

Origenes de datos vectoriales

- Application Schema DataAccess - Application Schema DataStore allows mapping of FeatureTypes to extern
- Directory of spatial files (shapefiles) - Takes a directory of shapefiles and exposes it as a data store
- Generalizing data store - Data store supporting generalized geometries
- **PostGIS - PostGIS Database**
- PostGIS (JNDI) - PostGIS Database (JNDI)
- Properties - Allows access to Java Property files containing Feature information
- Shapefile - ESRI(tm) Shapefiles (*.shp)
- Web Feature Server (NG) - Provides access to the Features published a Web Feature Service, and the abili

Origenes de datos raster

Max open prepared statements

50

- ☐ encode functions
- ☐ Support on the fly geometry simplification
- ☐ create database

create database params

[   ]

[Guardar] [Cancelar]

Uncheck "Support on the fly geometry simplification" because we want **raw data**.

If you don't uncheck it, Geoserver simplify the geometry on his own

2) Data Styles
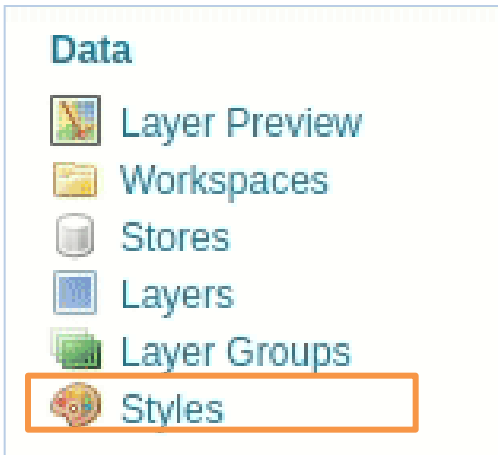
2) *Data Styles* → 1 per layer

SLD
Styles defined in Geoserver are **meaningless because they change with the client-side styles** (defined in the **.json file**)

We can use it for WMTS
You can raster vector tiles using **Mapnik**
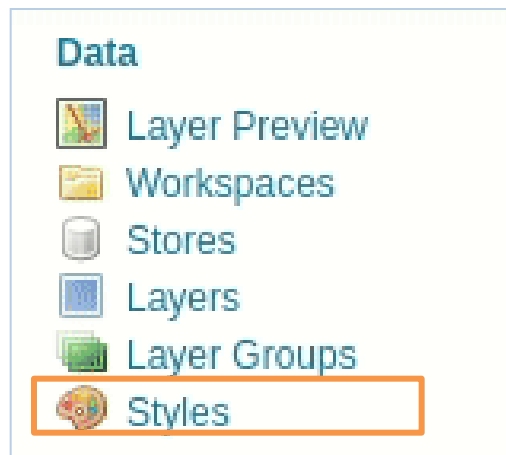The same style for WMTS

Why is it so important?
GeoWebCache uses SLD filters for cahing tiles
The visualizer will process all data unless you filter it in the style you define in Geoserver

Creating a Vector Tiles Service

GeoServer

2) *Data Styles* → 1 per layer

	Nombre del estilo
☐	rt0-autopista_lin
☐	rt0-autovia_lin
☐	rt0-carretera_nacional_lin

Example: Tunnel

Data
- 📊 Layer Preview
- 📁 Workspaces
- 🛢 Stores
- 🗺 Layers
- 🖼 Layer Groups
- 🎨 Styles

SLD

What do you have to define?
1) Attribute filter (Also in BD)
2) Max - Min zoom / level

```
<sld:Rule><!--Autovia E 1:150K SUB-->
  <sld:Name>Autovia</sld:Name>
  <sld:Title>Autovia E 1:150K</sld:Title>
  <sld:Abstract>Línea gris #CDC0B7 con ancho de 2 y borde 1</sld:
<ogc:Filter>
  <ogc:PropertyIsEqualTo>
      <ogc:PropertyName>tunel</ogc:PropertyName>
      <ogc:Literal>T</ogc:Literal>          T: true
  </ogc:PropertyIsEqualTo>
</ogc:Filter>
<sld:MinScaleDenominator>30000</sld:MinScaleDenominator>
<sld:MaxScaleDenominator>150000</sld:MaxScaleDenominator>
<sld:LineSymbolizer>
  <sld:Stroke>
    <sld:CssParameter name="stroke">#B5A195</sld:CssParameter>
    <sld:CssParameter name="stroke-width">3</sld:CssParameter>
  </sld:Stroke>
</sld:LineSymbolizer>
<sld:LineSymbolizer>
  <sld:Stroke>
    <sld:CssParameter name="stroke">#CDC0B7</sld:CssParameter>
    <sld:CssParameter name="stroke-width">2</sld:CssParameter>
  </sld:Stroke>
</sld:Stroke>
```

3) Gridsets

Tile Caching
- Tile Layers
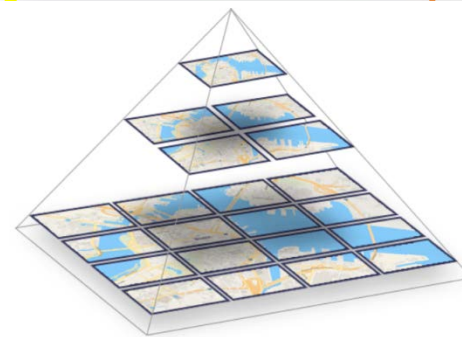- Caching Defaults
- Gridsets
- Disk Quota
- BlobStores

The bigger…
    Tile Dimensions   ⟶   The bigger
The better   The size of the tile
    Resolution
    Accuracy

	Gridset	CRS	Tile Dimensions	Zoom levels
☐	VT-EPSG:3857 - 512	EPSG:3857	512 x 512	
☐	VT-EPSG:3857 - 256	EPSG:3857	256 x 256	
☐	VT-EPSG:3857 - 2048	EPSG:3857	2048 x 2048	
☐	VT-EPSG:3857 - 1024	EPSG:3857	1024 x 1024	

0   256x256
…   …
…   …
…   …
…   …
…   …
21   2048x2048

Creating a Vector Tiles Service   GeoServer

3) Gridsets

Tile Caching

- Tile Layers
- Caching Defaults
- Gridsets
- Disk Quota
- BlobStores

	Gridset	CRS	Tile Dimensions
☐	VT-EPSG:3857 - 512	EPSG:3857	512 x 512

https://epsg.io/3857
WGS 84 / Pseudo-Mercator -- Google Maps, OSM, Bing, ArcGIS
MapBox is done for visualizing tiles in EPSG:3857

Tile Matrix Set

Define las cuadrículas a partir de: ⦿ Resolutions ○ Scale denominators

Nivel	Tamaño de píxel	Escala	Nombre	Celdillas	
0	156.543,03390625	1: 559.082.263,9508929	EPSG:3857 - 256:0	1 x 1	⊖
1	78.271,516953125	1: 279.541.131,97544646	EPSG:3857 - 256:1	2 x 2	⊖
2	39.135,7584765625	1: 139.770.565,98772323	EPSG:3857 - 256:2	4 x 4	⊖
3	19.567,87923828125	1: 69.885.282,99386162	EPSG:3857 - 256:3	8 x 8	⊖
4	9.783,939619140625	1: 34.942.641,49693081	EPSG:3857 - 256:4	16 x 16	⊖

Añadir nivel de zoom

Guardar   **Cancelar**

You can add as much zoom levels as you consider

4) Layers

GeoServer

4) *Layers*

SLD
Apply style for each layer

Server defined legends
are meaningless with
vector tiles because they
change with the client-
side styles

5) Layer groups

Data
- Layer Preview
- Workspaces
- Stores
- Layers
- **Layer Groups**
- Styles

Datos | Publicación | Cacheado de Teselas

Nombre

rt0

Título

Transport Network v0

Capas

- Agregar capa...
- Agregar grupo de capas...

	Orden de dibujo	Capa	Estilo por defecto	Estilo	Elimin
1	↓	vt_rt:aerodromo_pol	☐	rt0-aerodromos_pol	⊖
2	⇑ ↓	vt_rt:aerodromo_pto	☐	rt0-aerodromos_pto	⊖
3	⇑ ↓	vt_rt:autopista_lin	☐	rt0-autopista_lin	⊖
4	⇑ ↓	vt_rt:autovia_lin	☑	rt0-autovia_lin	⊖
5	⇑ ↓	vt_rt:carretera_nacional_lin	☐	rt0-carretera_nacional_lin	⊖
6	⇑ ↓	vt_rt:ferrocarril_lin	☐	rt0-ferrocarril_lin	⊖
7	⇑ ↓	vt_rt:ferrocarril_pk_pto	☐	rt0-ferrocarril_pk_pto	⊖
8	⇑ ↓	vt_rt:ferrocarril_pol	☐	rt0-ferrocarril_pol	⊖
9	⇑ ↓	vt_rt:carretera_primaria_lin	☐	rt0-carretera_primaria_lin	⊖
10	⇑ ↓	vt_rt:carretera_secundaria_lin	☐	rt0-carretera_secundaria_lin	⊖

ORDER OF LAYERS
- **WMTS** → Very important
- **Vector Tiles Service** → Don't mind
  .json defines the order

5) Layer groups
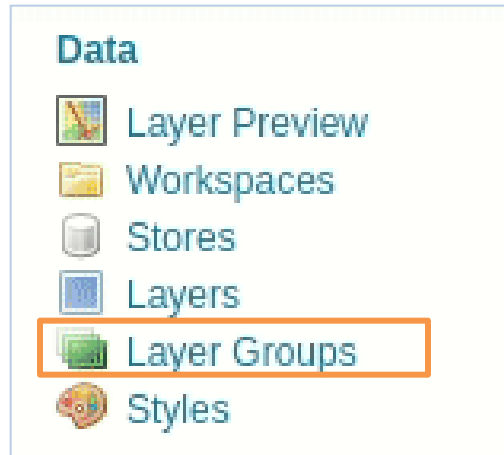
Add the GridSets you want the service has associated

GWC can only cache these GridSets

5) Layer groups

Data
- Layer Preview
- Workspaces
- Stores
- Layers
- Layer Groups
- Styles

| Datos | Publicación | **Cacheado de Teselas** |

Cache image formats
- ☐ application/json;type=geojson
- ☐ application/json;type=topojson
- ☐ application/json;type=utfgrid
- ☑ application/x-protobuf;type=mapbox-vector
- ☐ image/gif   .pbf → Optimized
- ☐ image/jpeg   for web

What is the content of a Tile (pbf)?

.pbf is a binary format

.pbf → .json

```
{
    "type": "FeatureCollection",
    "features": [
        {
            "type": "Feature",
            "geometry": {
                "type": "Polygon",
                "coordinates": [
                    [
                        [
                            2.0841461420059204,
                            41.3572947924726
                        ],
                        [
                            2.08415687084198,
                            41.35727667298437
                        ],
                    ]
                ]
            },
            "properties": {
                "class": "lake",
                "vt_layer": "water"
            },
            "id": 2
        },
        {
            "type": "Feature",
            "geometry": {
                "type": "Polygon",
                "coordinates": [
                    [
                        [
                            2.086804211139679,
                            41.3596522961968
                        ],
                        [
                            2.087574005126953,
                            41.35787662669159
                        ],
                        [
                            2.087574005126953,
                            41.35706527858204
                        ],
```

Geometries

Inspire Helsinki 2019

5) Layer groups

Data
- Layer Preview
- Workspaces
- Stores
- Layers
- Layer Groups
- Styles

| Datos | Publicación | **Cacheado de Teselas** |

Cache image formats
- ☐ application/json;type=geojson
- ☐ application/json;type=topojson
- ☐ application/json;type=utfgrid
- ☑ application/x-protobuf;type=mapbox-vector
- ☐ image/gif
- ☐ image/jpeg

What is the content of a Tile (pbf)?

You can see it:

http://geojson.io/

Geometries

```
{
    "type": "FeatureCollection",
    "features": [
        {
            "type": "Feature",
            "geometry": {
                "type": "Polygon",
                "coordinates": [
                    [
                        [
                            2.0841461420059204,
                            41.3572947924726
                        ],
                        [
                            2.08415687084198,
                            41.35727667298437
                        ],
                    ]
                ]
            },
            "properties": {
                "class": "lake",
                "vt_layer": "water"
            },
            "id": 2
        },
        {
            "type": "Feature",
            "geometry": {
                "type": "Polygon",
                "coordinates": [
                    [
                        [
                            2.086804211139679,
                            41.3596522961968
                        ],
                        [
                            2.087574005126953,
                            41.35787662669159
                        ],
                        [
                            2.087574005126953,
                            41.35706527858204
                        ],
```
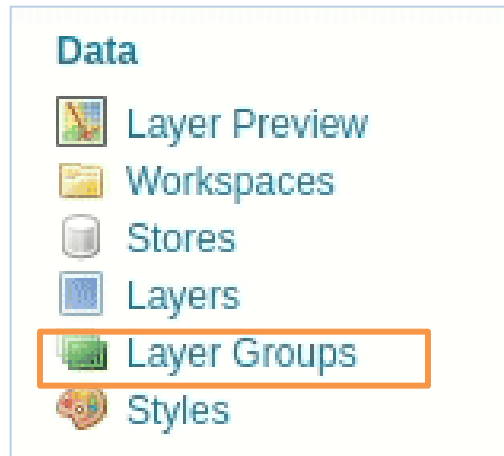
Inspire Helsinki 2019

Creating a Vector Tiles Service

GeoServer

6) Tile Layers (pre-caching)

Tile Caching

- Tile Layers
- Caching
- Gridsets
- Disk Quota
- BlobStore

Create a new task:

Number of tasks to use:	01 ▾
Type of operation:	Seed - generate missing tiles ▾
Grid Set:	EPSG:3857 ▾
Format:	application/x-protobuf;type=mapbox-vector ▾
Zoom start:	00 ▾
Zoom stop:	10 ▾
Bounding box:	☐ ☐ ☐ ☐

These are optional, approximate values are fine.

Bounding box
Default: All World
Better specify the BBOX (of Spain)
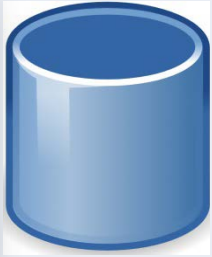
Spanish Transport Network
≈ 12 hours (8 cores, RAM 32GB, magnetic hard disc)

Inspire Helsinki 2019

Comparing technologies

DATA: Transport Network	TILES GENERATION		SERVICE	
	Software	Result		
geojson – seq files *Allows tippecanoe to distribute tasks. Uses all available cores*	Tippecanoe (.ini)	MBTiles (SQLite + .pbf) 1 per zoom level	node.js own development	
DB Postgis 	GeoServer GeoWebCache	.pbf	GeoServer GeoWebCache	...
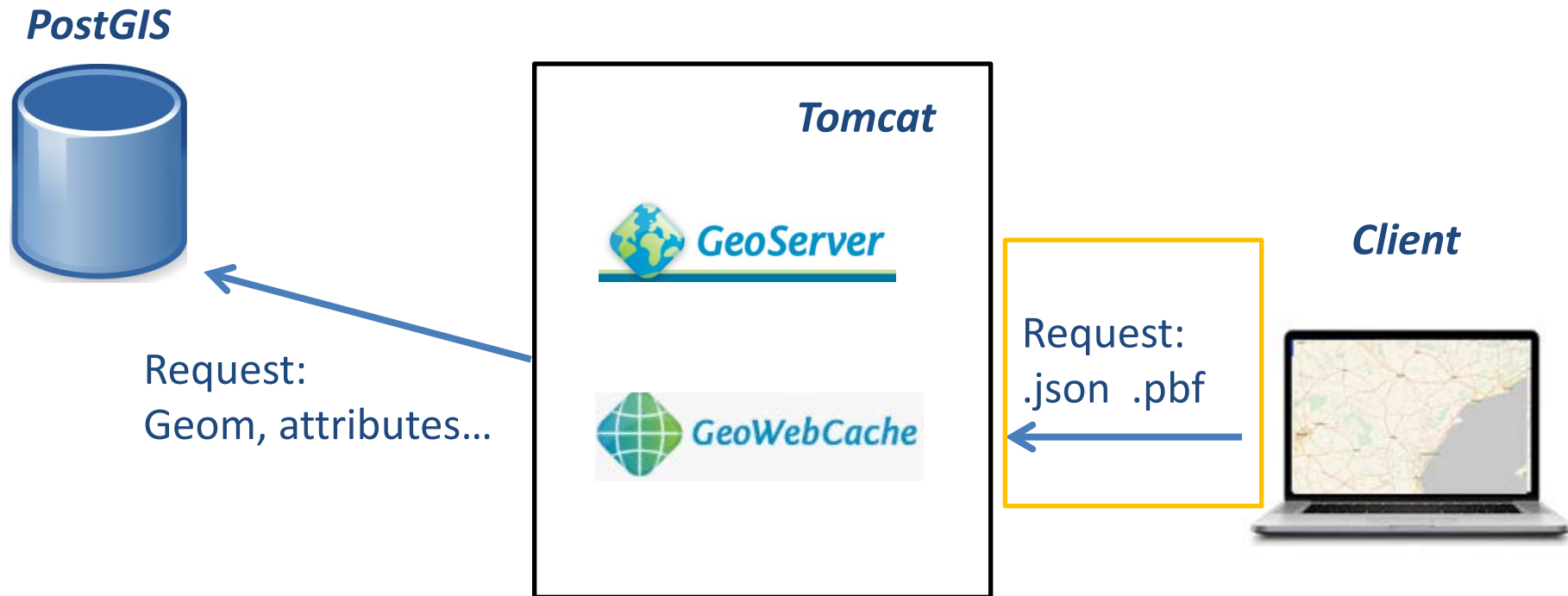
https://tms-redes-transporte.idee.es/demoVectorTiles/comparador/comparador.html#4.8/36.555/-6.927

Workflow - architecture

PostGIS

Tomcat

GeoServer

GeoWebCache

Request:
Geom, attributes…

Request:
.json .pbf

Client

If GWC has the tile→ It gives to the viewer
If not → Ask Geoserver, save it and give it to the viewer

Simbolizyng Vector Tiles Service

.json

```json
{
    "version": 8,          → Mapbox style specification
    "name": "Geoserver",   → style name
    "metadata": {
        "mapbox:autocomposite": false,
        "mapbox:type": "template",
        "maputnik:renderer": "mbgljs",
        "openmaptiles:version": "3.x"
    },
    "sources": {
        "rt0-1024": {       → "rt0-512", "rt0-1024"...
            "type": "vector",
            "scheme": "tms",   → GeoWebCache ZXY
            "tiles": [
                "https://tms-redes-transporte.idee.es/rt/vt_rt:rt0@VT-EPSG:3857-
                1024@pbf/{z}/{x}/{y}.pbf"
            ],
            "minzoom": 06,
            "maxzoom": 18
    },
    "sprite": "mapbox://sprites/mapbox/basic-v9",
    "glyphs": "mapbox://fonts/mapbox/{fontstack}/{range}.pbf",
```
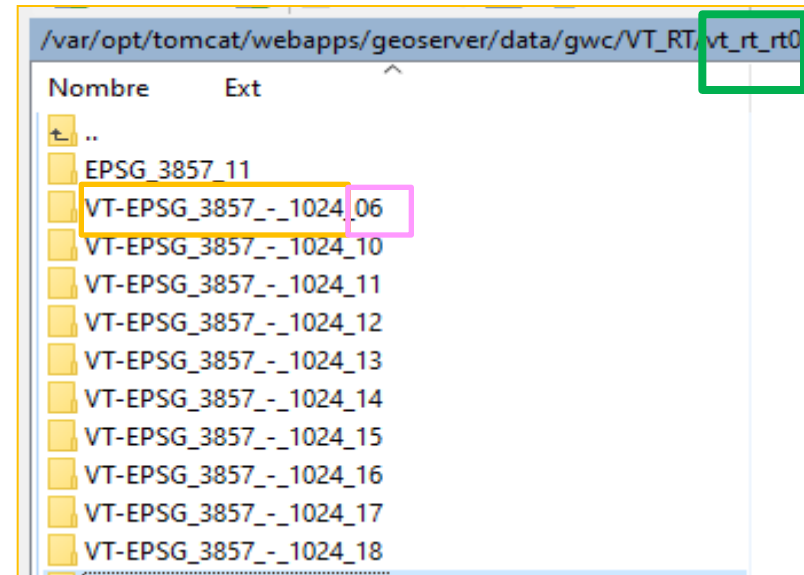
Template for VT

Workspace:layer group

GridSet

/var/opt/tomcat/webapps/geoserver/data/gwc/VT_RT/vt_rt_rt0

Nombre	Ext
..	
EPSG_3857_11	
VT-EPSG_3857_-_1024_06	
VT-EPSG_3857_-_1024_10	
VT-EPSG_3857_-_1024_11	
VT-EPSG_3857_-_1024_12	
VT-EPSG_3857_-_1024_13	
VT-EPSG_3857_-_1024_14	
VT-EPSG_3857_-_1024_15	
VT-EPSG_3857_-_1024_16	
VT-EPSG_3857_-_1024_17	
VT-EPSG_3857_-_1024_18	

Simbolizyng Vector Tiles Service

.json

SPRITE:
`"sprite": "mapbox://sprites/mapbox/basic-v9",`

An individual image that has all the icons included in a style.
For Mapbox: + **.png** with images
                 + **.json** with name and relative cordinates

Advantages:
- 1 image for all icons → Less requests → Higher performance

GLYPHS:
`"glyphs": "mapbox://fonts/mapbox/{fontstack}/{range}.pbf",`

For texts   -   fontstack: Open Sans Bold

Simbolizyng Vector Tiles Service

.json

```json
"layers": [
    { "id": "area_de_servicio_pol-fill-14-->18",
      "type": "fill",
      "source": "rt0-512",
      "source-layer": "area_de_servicio_pol",
      "paint": {
        "fill-color": "#DCDCDC",
        "fill-opacity": 1.0,
        "fill-outline-color": "#cccccc"
      },
      "minzoom": 14
    },
    { "id": "via_servicio_lin_0-line-14-->18",
      "type": "line",
      "filter":[
        "all",
        ["==", "tunel", "F"]
      ],
      "source": "rt0-512",
      "source-layer": "via_servicio_lin",
```

Simbolizyng Vector Tiles Service
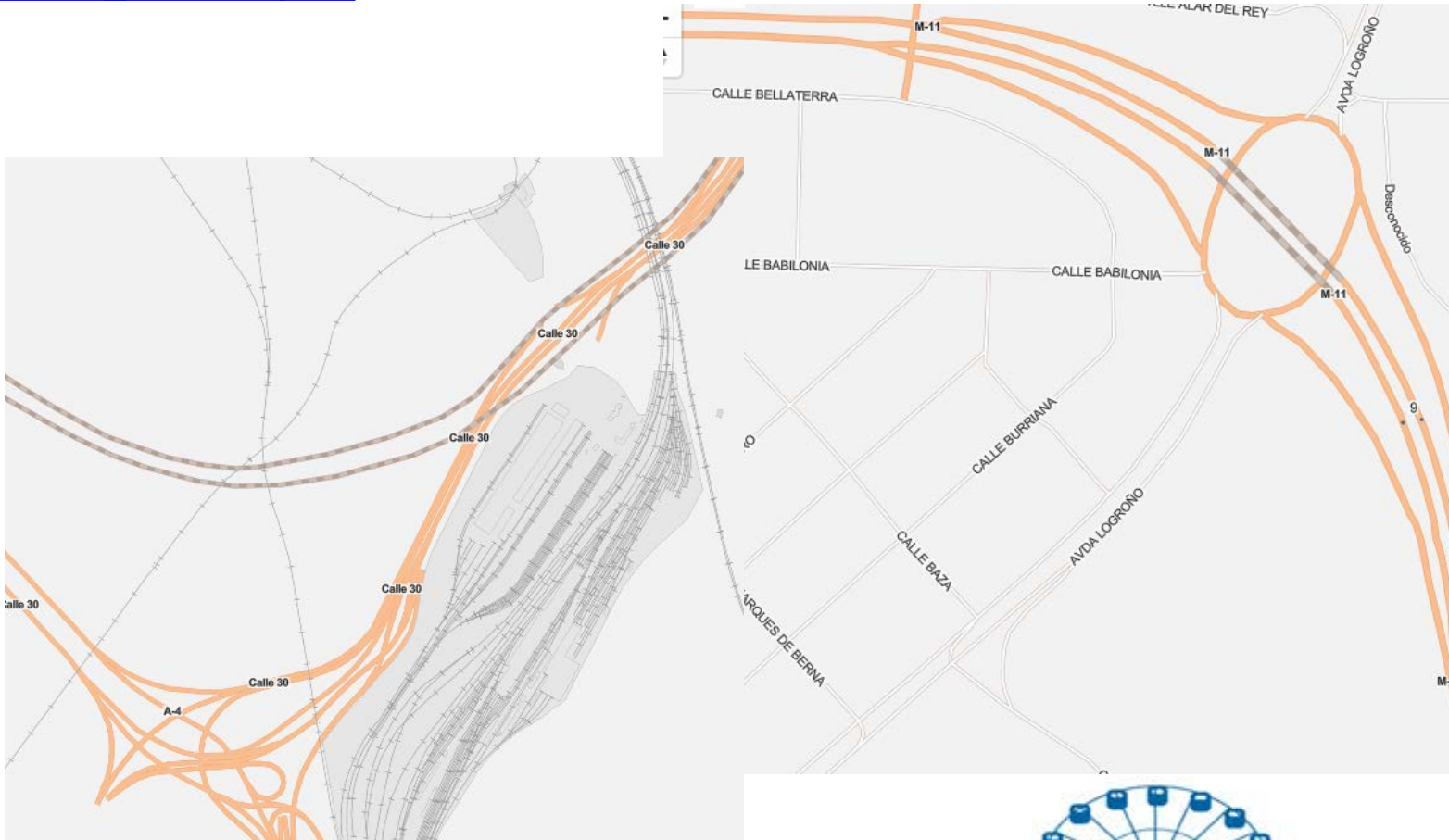
.json

,

```json
{
    "id": "portal_pto-symbol-17-->18",
    "type": "symbol",
    "source": "rt0-256",
    "source-layer": "portal_pto",
    "layout": {
        "text-field": "{nombre}",
        "text-size": 8,
        "text-padding": 10,
        "text-font": [
            "Lato Light"
        ]
    },
    "paint": {
        "text-color": "#666565",
        "text-opacity": 1.0,
        "text-halo-color": "#ffffff",
        "text-halo-width": 2.0,
        "text-halo-blur": 1.0
    },
```

Results

Transport Network JSON :
https://tms-redes-transporte.idee.es/demoVectorTiles/style/style-full-geoserver_vtRT-GS.json

Results

Transport Network JSON :
https://tms-redes-transporte.idee.es/demoVectorTiles/style/style-full-geoserver_vtRT-GS.json

JSON for Transport Network, Buildings, Hydrography, etc:
- https://vts.larioja.org/style/mapa-base-igo-v1.json (Official information)
- https://vts.larioja.org/style/mapa-base-vector-v1.json (OSM)
- https://vts.larioja.org/style/mapa-oscuro-vector-v1.json
- https://vts.larioja.org/style/mapa-hibrido-v1.json

WORKLOAD

- Preparing Data 30%
- Creating the Service 20%
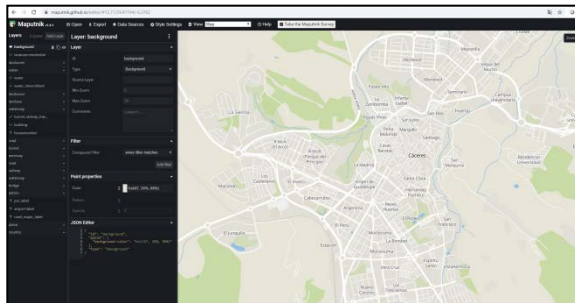- Publishing 50%

Inspire Helsinki 2019

Visualization - Client

TOOLS, SOFTWARE

- MapBoxGL JS
- Open Layers (Libraries)
- Leaflet (Libraries)

SEE RESULTS & MODIFY THE STYLE
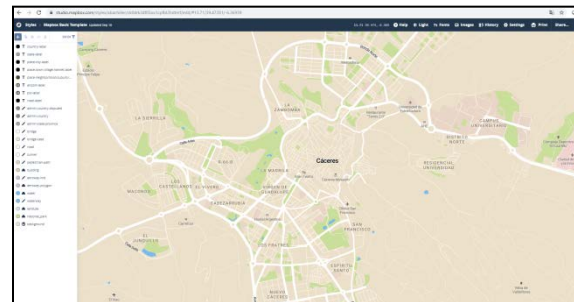
Maputnik
 https://maputnik.github.io/editor/#13.78/40.48196/-3.66936

Mapbox Studio: https://studio.mapbox.com/

Example with Maputnik

Give shadow to our buildings

1) Open Maputnik:

https://maputnik.github.io/editor/#16.32/40.454378/-3.689535

2) Open the following .json in Maputnik

JSON for Transport Network, Buildings, Hydrography, etc:

- https://vts.larioja.org/style/mapa-base-igo-v1.json (Official information)

3) Copy the layer of Buildings and translate it

1 layer: 2 Styles
No need more resources
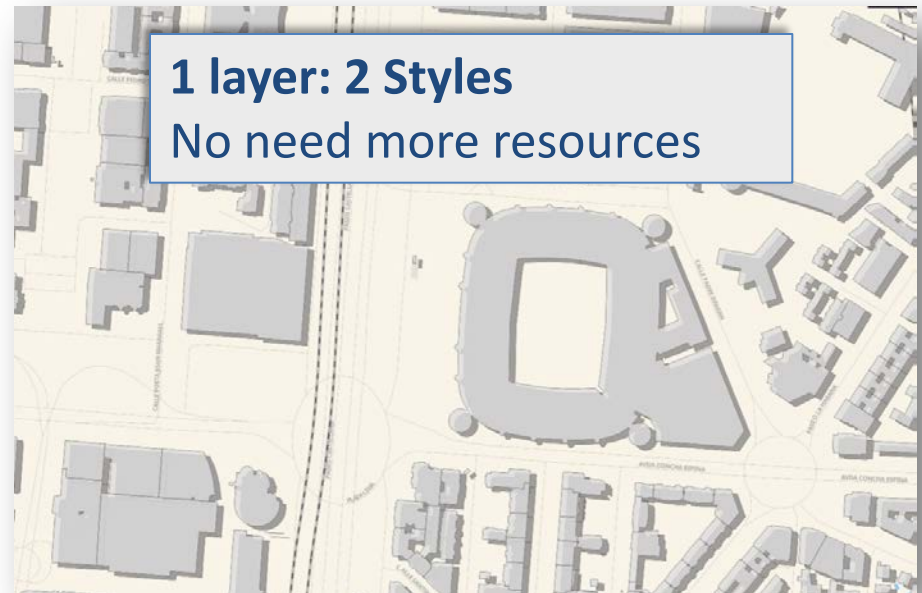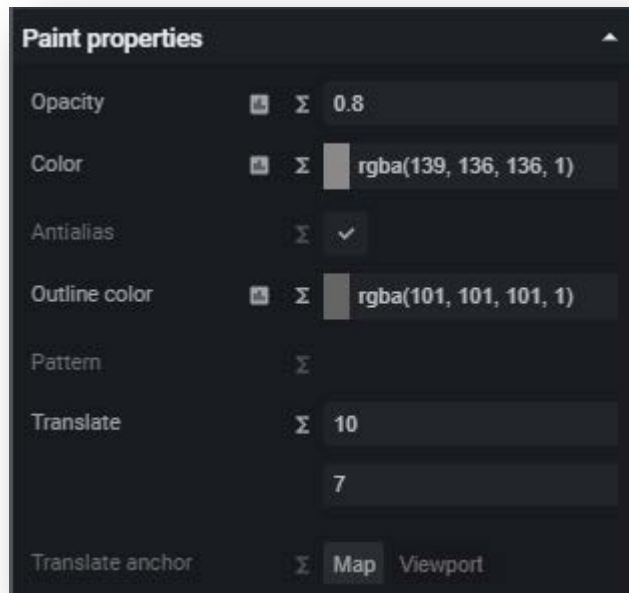
Exchange of experience
of vector tile service: pros and cons

Thank you

Centro Nacional de Información Geográfica (CNIG)
Spain

Laura Alemany Gómez
laura.alemany@cnig.es
Jose Maria García Malmierca
jmgmalmierca@fomento.es
Álvaro Bachiller
alvaro.bachiller@cnig.es

Inspire Helsinki 2019