



GéoInf'2015

Open Webmapping

Pr. Olivier Ertz
HEIG-VD, COMEM+



Roadmap

- Contexte et introduction
- Éléments de base
- Workshop OpenLayers (1)
- Workshop PostGIS
- Workshop OpenLayers (2)
- Exercice final
- Sujet exploratoire



C'est quoi
le plan ... ?

A large, hand-drawn style speech bubble with a thick black outline, containing the text 'C'est quoi le plan ... ?' in a simple, black, sans-serif font.

Une histoire de convergence

- GIS
- WebGIS
- Webmapping
- GeoWeb
- Neogeography
- Neocartography
- SDI, etc.



https://fr.wikipedia.org/wiki/Les_Maitres_cartographes



Du GIS tool au WebGIS

- Run QGIS
- Add layer
- Et hop tout est prêt !
- Une boîte à outils intégrée pour tout faire
- Modéliser, acquérir, analyser, visualiser, ...

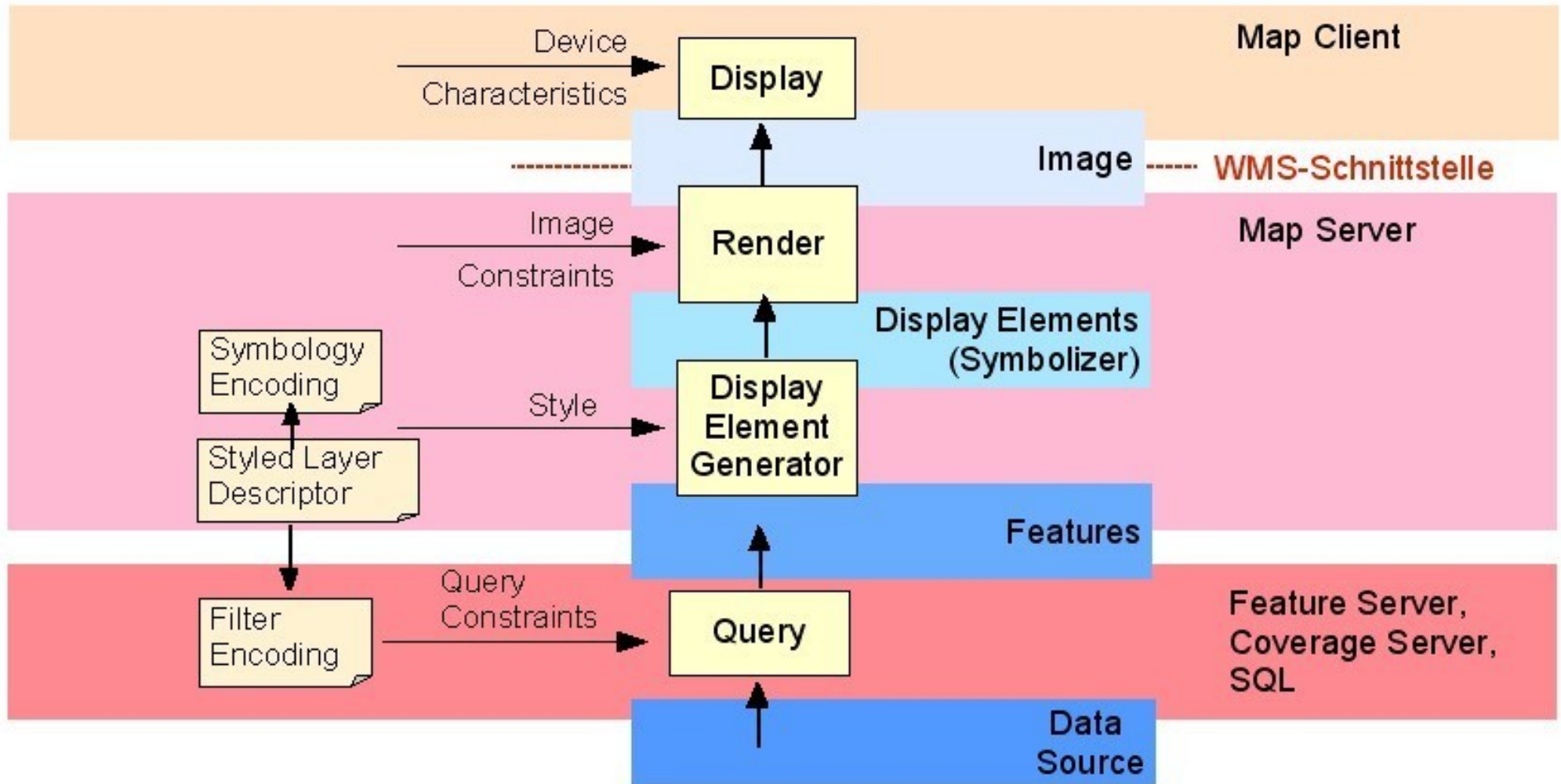
→ Le mythe du WebGIS ?



WebGIS, Webmapping, GeoWeb

- Du sur mesure versus GIS
- Webmapping = view+navigation+query
- Autres “modules” : analysis (carto, routing, ...), editing, search and geocoding services, etc
- Web = distribué, interconnecté, interopérable, ...
- Webmapping = cartographie en ligne
- ➔ **Et plus** : services carto., géoservices, API, ...

Portrayal model





QgsMapRenderer.py

```
from PyQt4.QtCore import *
from PyQt4.QtGui import *

img = QImage(QSize(800,600), QImage.Format_ARGB32_Premultiplied)
img.fill(QColor(255,255,255).rgb())
p = QPainter()
p.begin(img)
p.setRenderHint(QPainter.Antialiasing)
render = QgsMapRenderer()
aLayer = qgis.utils.iface.activeLayer()
render.setLayerSet([ aLayer.id() ])
aLayer.loadSldStyle('/tmp/test.sld')
rect = QgsRectangle(render.fullExtent())
render.setExtent(rect)
render.setOutputSize(img.size(), 96)
render.render(p)
p.end()
img.save("/tmp/render.png", "png")
```



Webmapping basics

- Client/serveur
- Serveur cartographique
- Client de navigation cartographique
- Standard OGC Web Map Service



[http://demo.boundlessgeo.com/geoserver/wms?
service=WMS&version=1.1.0&request=GetMap&layers=ne_10m_admin_0_countries&styles=&bbox=-180,-90,180,90&width=800&height=400&srs=EPSG:4326&format=image/png](http://demo.boundlessgeo.com/geoserver/wms?service=WMS&version=1.1.0&request=GetMap&layers=ne_10m_admin_0_countries&styles=&bbox=-180,-90,180,90&width=800&height=400&srs=EPSG:4326&format=image/png)





Server-side: GetMap

- **URL du service** <http://demo.boundlessgeo.com/geoserver/wms>
 - `request=GetMap` : la "méthode" WMS
 - `version=1.1.0` : la version WMS "parlée"
 - `bbox=-180,-90,180,90` : l'emprise géographique
 - `srs=EPSG:4326` : le syst. de coord.
 - `layers=ne_10m_admin_0_countries` : le(s) couche(s)
 - `styles=` : le style de rendu (voir ci-dessous)
 - `format=image/png` : le format de sortie
 - `width=800&height=400` : les dimensions de l'image
- **Style à choix | personnalisable** (OGC SLD/SE)

<http://geoserver-heig.rhcloud.com/wms?>

`service=WMS&version=1.1.0&request=GetMap&layers=world_simple&styles=green&bbox=-180,-90,180,83&width=684&height=330&srs=EPSG:4326&format=image/png`

Client-side: my 1st webmap app

Exercice Ex0a :

- On veut construire une IHM minimaliste pour naviguer dans la couche `ne_10m_admin_0_countries`
- GetCapabilities, GetMap, ...
- Zoom in/out +
- Pan west/east/north/south
- *A l'ancienne* avec 6 boutons





Ex0b: overlay & control

- On veut superposer la couche des lacs `ne_10m_lakes` disponible du même serveur ...
- ... et laisser l'utilisateur contrôler sa visibilité (ex. avec une checkbox)
- On a donc les notions de :
 - Baselayer : une couche de référence
 - Overlay : plurs couches en superposition



INFO: Sudan
25,946,220 inhab.

Ex0c: feature info

- Utiliser la méthode WMS GetFeatureInfo permettant d'interroger les entités de la couche `ne_10m_admin_0_countries` par un clic sur la carte
 - GetFeatureInfo = les paramètres GetMap + ...
 - `query_layers`: le nom de(s) la couche(s) à interroger
 - `x,y`: les coordonnées graphiques de clic
 - `info_format`: le mime-type du résultat (ex. `text/plain`, voir le `GetCapabilities`)
 - Du résultat, extraire puis afficher le nom et la population du pays
 - Ce sont les propriétés `admin` et `pop_est`
(voir les détails avec la méthode `DescribeFeatureType`)

<http://demo.boundlessgeo.com/geoserver/ne/wfs?request=DescribeFeatureType>



Et avec un framework JS cartographique

- ... C'est plus facile :
 - OpenLayers, Leaflet : découplé/ compatible avec plrs types de services cartographiques (...)
 - Bing Maps, Google Maps : fortement couplé à son propre service cartographique et autres (...)
 - Et tous les autres, GeoAdmin API, ArcGIS API for JavaScript (...)

- Exemples :

<http://www.geo.vd.ch> <https://map.geo.admin.ch> <http://sitn.ne.ch>

<http://www.atlas.bfs.admin.ch> <http://sitn.ne.ch/geoclip6/cartto.php>

<http://www.camptocamp.org/map> <http://urbangene.heig-vd.ch> etc.