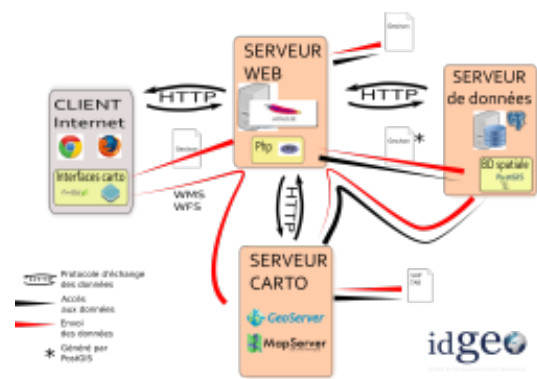


# Cartographie et internet



# Table des matières

|                                                      |    |
|------------------------------------------------------|----|
| Objectifs                                            | 3  |
| Introduction                                         | 4  |
| I - Fonctionnement d'Internet                        | 5  |
| 1. Transmission des données sur le web .....         | 5  |
| 2. Le HTML et le WWW .....                           | 7  |
| 3. Le XML (eXtensible Markup Language) .....         | 7  |
| 4. Le cloud .....                                    | 8  |
| 5. Contraintes et potentialités .....                | 9  |
| 6. Les réseaux web : terminologie .....              | 11 |
| 7. En résumé : les catégories de sites web .....     | 11 |
| II - Le webmapping                                   | 15 |
| 1. Le webmapping .....                               | 16 |
| 1.1. Qu'est-ce que le webmapping ? .....             | 16 |
| 1.2. Client cartographique .....                     | 18 |
| 2. Rendre disponible les données géographiques ..... | 19 |
| 2.1. Formats vecteurs VS formats rasters .....       | 19 |
| 2.2. HTML et API Javascript .....                    | 24 |
| 2.3. Les serveur web .....                           | 27 |
| 3. Les solutions en ligne .....                      | 36 |
| 3.1. Les solutions open-sources .....                | 37 |
| Conclusion                                           | 40 |

# Objectifs

Explorer les possibilités d'intégration d'une cartographie dynamique sur le web (le webmapping)

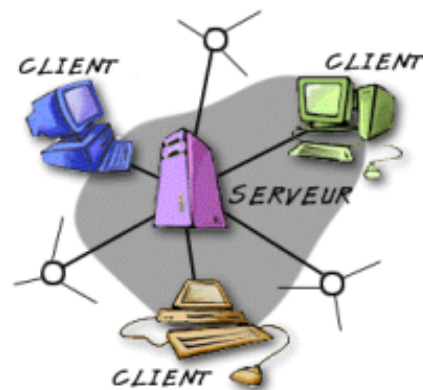
Permettre aux personnes souhaitant mettre leur SIG sur Internet de choisir entre les différentes technologies possibles

# Introduction

Les SIG en réseaux

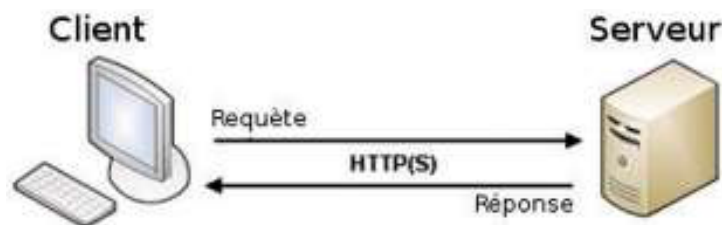
Intérêts des réseaux d'entreprises :

- Échanges des informations facilités ;
- Mutualisation des données ;
- Intégrités des données ;



Notion de client/serveur :

- Le client est une application qui va dialoguer avec le serveur (par exemple ArcGIS) ;
- Le serveur envoie une réponse au client en fonction de sa demande (par exemple ArcGIS Server envoie les données d'une couche PostGIS à ArcGIS pour qu'il la visualise).



Sur un réseau local, tous les ordinateurs sont connectés entre eux et avec le serveur par un réseau interne câblé. Ils ont le même système d'exploitation.

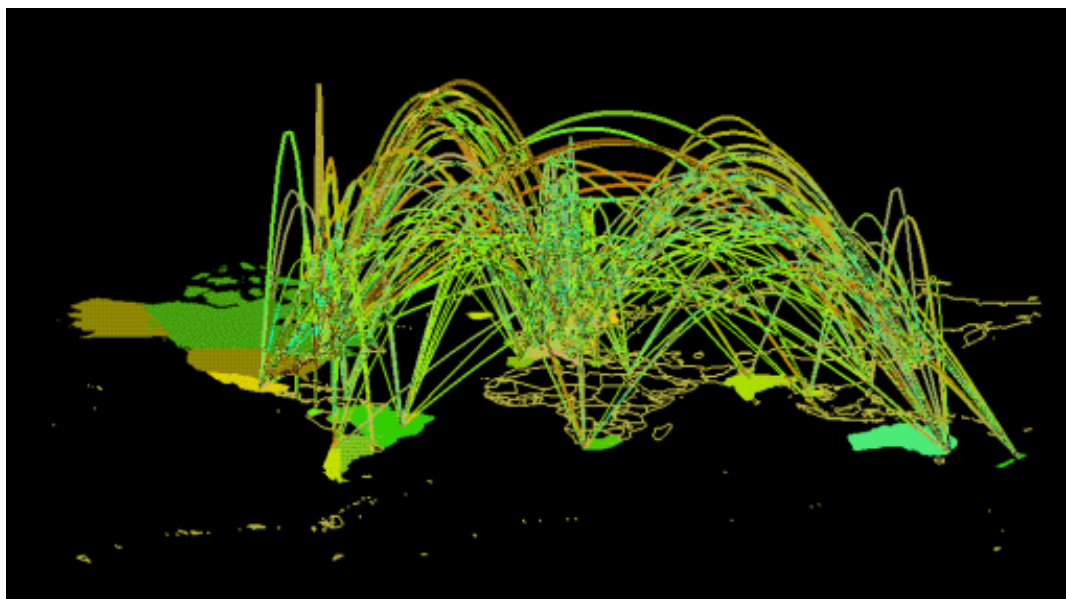
Sur Internet, c'est différent :

- Propres langages, méthodes d'affichage et gestion des informations ;
- Vitesse de transmission des données réduite ;
- Le nombre d'utilisateurs est difficile à connaître ;

Il faut adapter le SIG aux contraintes d'Internet...

# Fonctionnement d'Internet

La représentation des réseaux de l'Internet en géographie (cybergeographie) :



Atlas of Cyberspaces

L'Internet est donc un ensemble de réseaux d'ordinateur.

Nous allons voir :

- Comment les données sont transmises d'un ordinateur à l'autre ;
- Ce qu'est une page et un site web ;
- La notion de feuille de style et de XML ;
- Le Javascript et l'AJAX pour rendre dynamique une page en HTML ;
- Ce qu'est le Cloud ;

Nous finirons par une discussion sur les avantages et les inconvénients du web et sur la terminologie des différents réseaux utilisant la technologie web.

## 1. Transmission des données sur le web



Définition : Internet

Ensemble de réseaux d'ordinateurs interconnectés par l'intermédiaire de lignes spécialisées haut débit.



## La transmission des données sur le web

Les ordinateurs communiquent par l'intermédiaire de protocoles (= langage commun entre deux ordinateurs pour qu'ils puissent se comprendre) : TCP/IP.

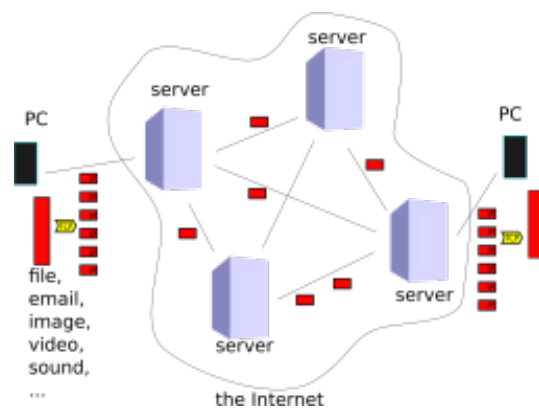
- Transmission Control Protocol :

Il découpe les données, qui vont transiter sur le réseau, en paquets.

- Internet Protocol :

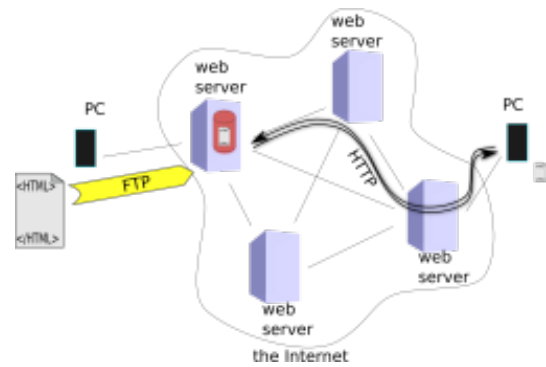
Donne l'adresse de l'envoyeur et du récepteur des paquets.

Les paquets passent par des routeurs (nœuds du réseau) qui dirigent au mieux les paquets vers leurs destinations.



Les autres protocoles :

- FTP (pour le transfert de fichiers),
- HTTP (pour les pages web),
- ...



Sur le web, une ressource est identifiée par son url (chaîne de caractères). Lorsque le client souhaite atteindre une ressource distante, il émet donc une requête HTTP mentionnant l'url de la ressource ; la requête comporte aussi dans certains cas une liste de paramètres que la ressource devra exploiter.

## 2. Le HTML et le WWW

Internet est surtout connu et utilisé grâce aux « pages Web » qui permettent de naviguer facilement sur le réseau par l'intermédiaire de liens hypertextes (World Wide Web) et du protocole HTTP.

Ces pages sont écrites dans un langage particulier, le HTML (Hyper Text Mark-up Language) que seuls les navigateurs (browser) peuvent comprendre.

Un site Web regroupe un ensemble de pages Web sur un serveur connecté en permanence à Internet (serveur web).



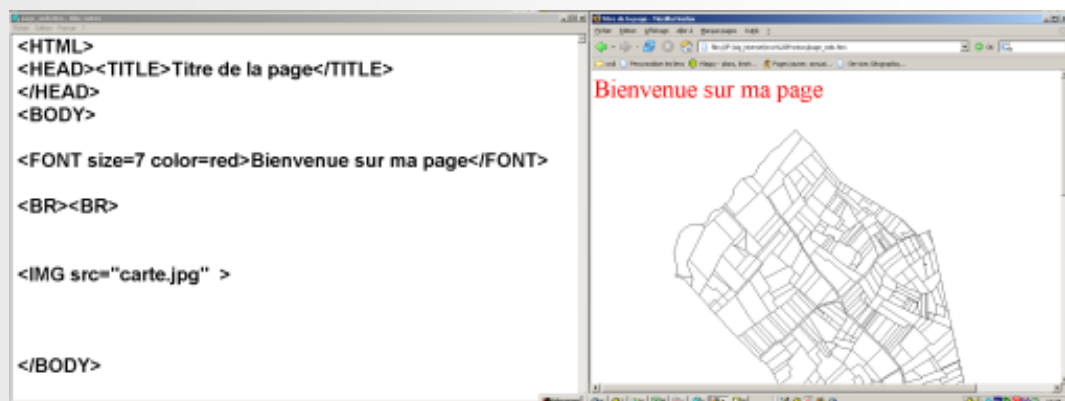
### Fondamental

Une page web :

- un fichier texte écrit dans un langage lu par les navigateurs : le HTML.

Principes du HTML :

- Des balises encadrent du texte et sont interprétées par le navigateur ;
- Le fichier HTML ne contient que du texte :
- Les images, les sons, les programmes « accompagnent » le fichier texte.



## 3. Le XML (eXtensible Markup Language)

- Il permet de créer ses propres balises, c'est un langage ouvert mais il est plus difficile à mettre en place que du HTML simple.

- Les documents sont adaptés aux échanges et aux applications de toute sorte (notamment utilisé pour les métadonnées).
- Il est extensible, et constitue la base de tous les nouveaux "dialectes" spécialisés, comme le SMIL (pour le multimédia), le SVG (pour le graphisme vectoriel), le GML (Geography Markup Language, cf. l'OpenGIS Consortium), ...

### Exemple : SVG

Le Scalable Vector Graphics (SVG) est un format de données conçu pour décrire des ensembles de graphiques vectoriels et basé sur XML

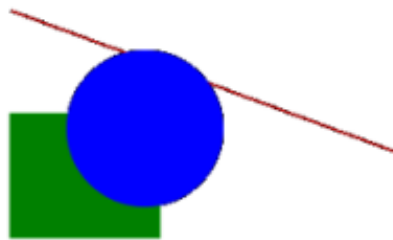
Les coordonnées, dimensions et structures des objets vectoriels : dans le document XML.

Style (CSS ou XSL) permettant d'indiquer les couleurs et les polices de caractères à utiliser.

Formes géométriques de base (rectangles, ellipses, etc.).

Le code suivant permet d'afficher un rectangle, un segment, un cercle et un texte :

```
1
2 <?xml version="1.0" encoding="utf-8"?>
3 <svg xmlns="http://www.w3.org/2000/svg" version="1.1" width="300" height="200">
4 <title>Exemple simple de figure SVG</title>
5 <desc> Cette figure est constituée d'un rectangle, d'un segment de droite et d'un cercle. </desc>
6 <rect width="100" height="80" x="0" y="70" fill="green" />
7 <line x1="5" y1="5" x2="250" y2="95" stroke="red" />
8 <circle cx="90" cy="80" r="50" fill="blue" />
```



### Remarque

Le XML est un format "verbeux" avantageusement remplacé par le JSON (Javascript Object Notation).

## 4. Le cloud

### Définition : Cloud computing

Manière de fournir et d'utiliser un parc de machines, d'équipement de réseau et de logiciels maintenu par un fournisseur, que les consommateurs peuvent utiliser via Internet en libre service.



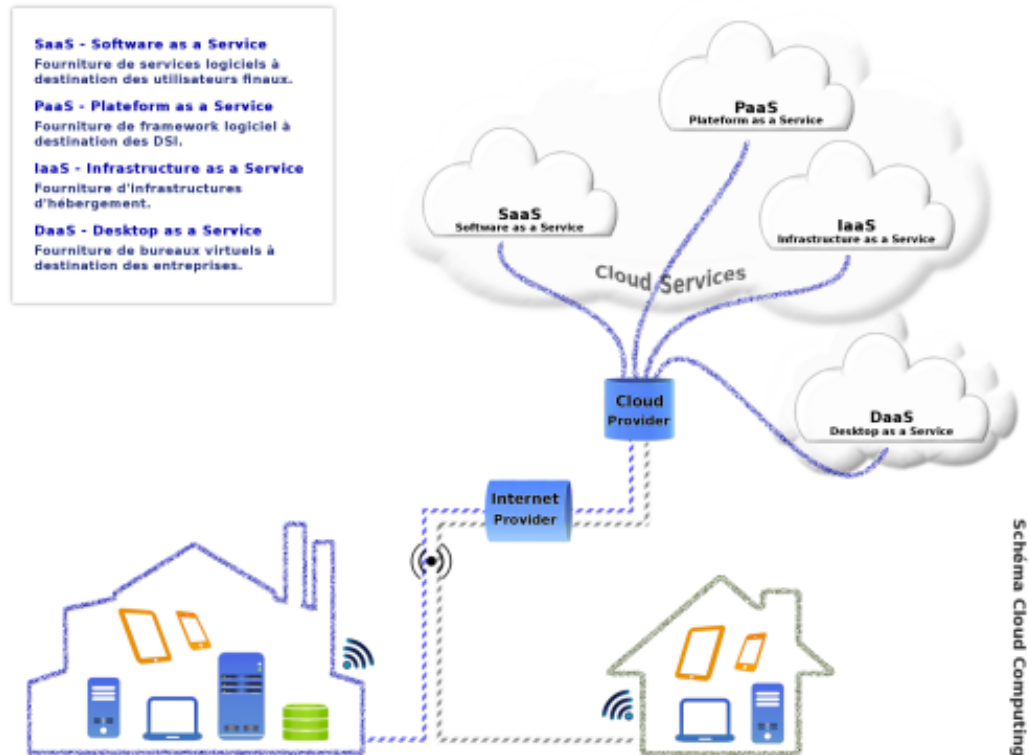


Schéma Cloud Computing. Image PNG 2048x1536 px. Produit par Mimmo D.DN - MDDN2014

## 5. Contraintes et potentialités

### Les contraintes

La vitesse :

Temps de réponse beaucoup plus long que sur un réseau interne d'ordinateurs (100 Mbps) :

- RTC (réseau Téléphonique Commuté) : 56 Kbit/s,
- Câble et fibre optique : actuellement entre 100 Mb/s et 8 Gb/s en téléchargement, et entre 100 et 700 Mb/s en transfert (upload),
- ADSL (Asymmetric Digital Subscribe Line) : vitesse compris entre 1 et 95 Mb/s en téléchargement et entre 1 et 30 Mb/s en transfert,
- Wi-Max (Worldwide Interoperability for Microwave Access) : 10 Mbit/s (théorique) sur 20 km,
- Satellite : 20 Mbit/s,
- 3G : entre 15 Mbit/s et 42 Mbit/s, 4G : entre 300 Mbit/s et 3 Gbit/s

Une vitesse entre 3 et 8 Mb/s est qualifiée de "moyen débit". En 2019, la valeur de 8 Mb/s correspond au seuil d'une connexion "haut débit". Il faut une vitesse d'au moins 30 Mb/s pour qualifier une connexion internet "très haut débit".

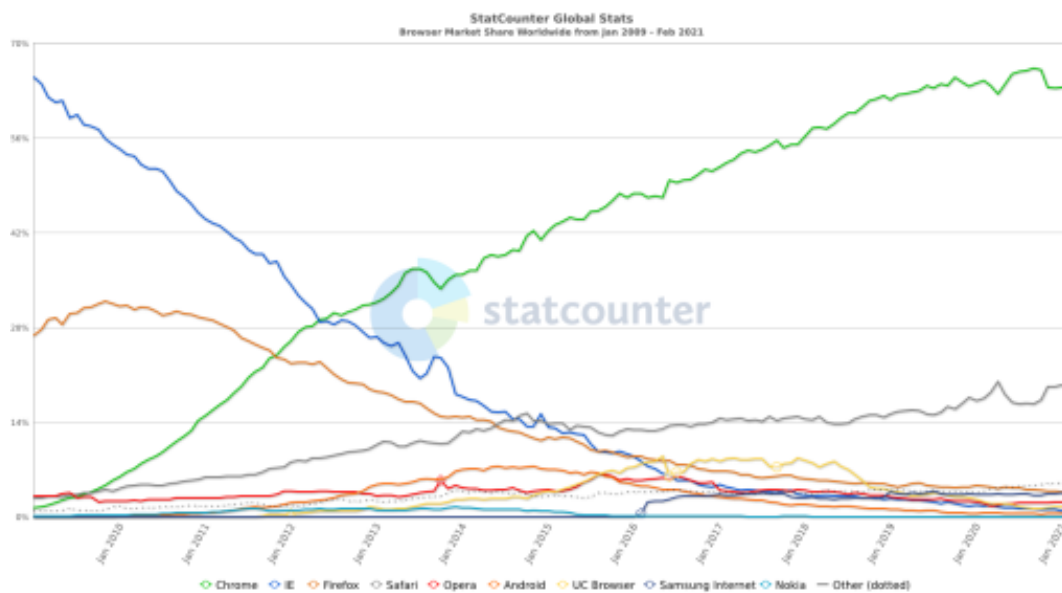
L'environnement hétérogène :

- L'utilisateur peut être sur un Mac, un PC ou sur station,
- La résolution de l'écran peut être différente d'un ordinateur à un autre,
- A chaque navigateur utilisé, un affichage différent, des possibilités d'extensions (plugins) plus ou moins limitées.



Le site Can I Use ?

Les navigateurs web par pays



Limitations techniques :

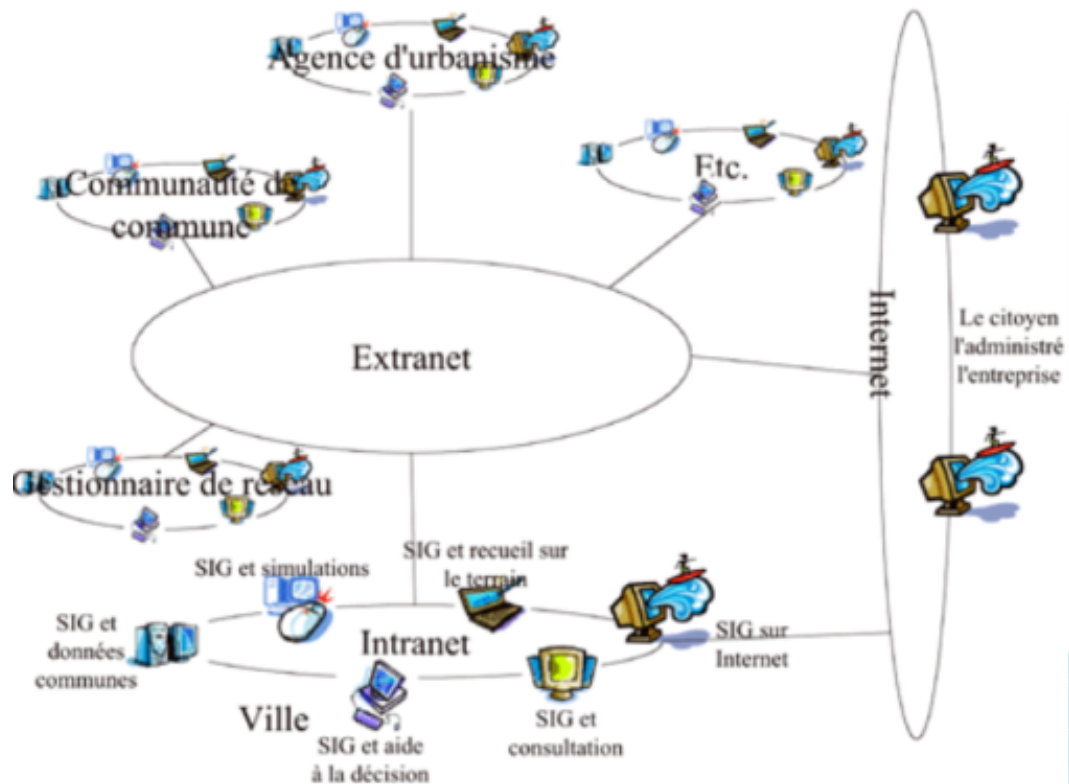
- Limitation du nombre de couleurs sur les navigateurs ;
- Nombreux « standards » : il faut parfois que le client (l'utilisateur) télécharge des "plugins" (programmes) pour visualiser des sites cartographiques

Les potentialités

- Interactivité,
- Disponibilité des informations à n'importe quelle heure du jour et de la nuit,
- Coût minime pour l'utilisateur final (un ordinateur et un modem),
- Simplification pour l'administrateur (une seule application à gérer sur le serveur).

## 6. Les réseaux web : terminologie

Internet, extranet, intranet



## 7. En résumé : les catégories de sites web

Les sites statiques

Ces sites se basent uniquement sur du HTML, il s'agit de visualisation de pages HTML statiques

La structure et le contenu des documents sont figés.

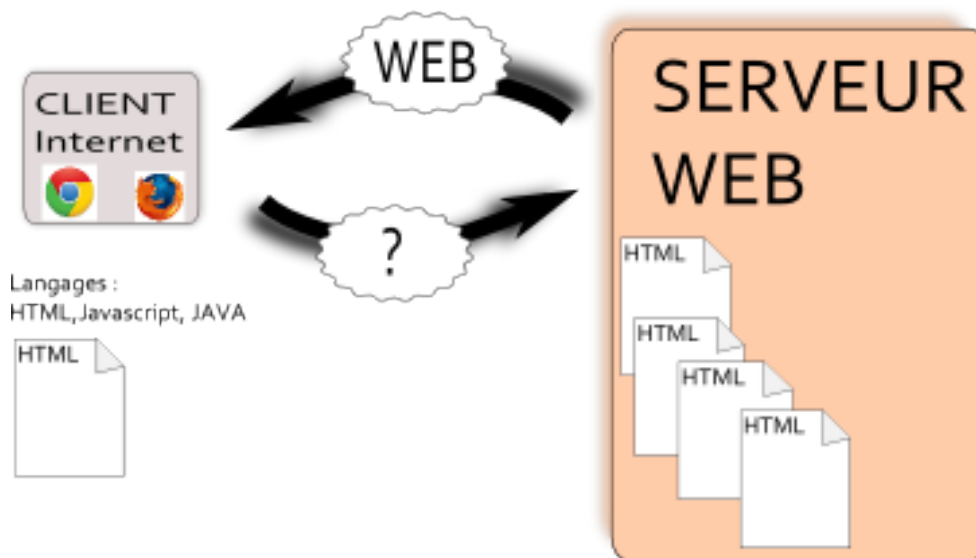
Un serveur web « de base » retransmet la page d'information correspondant à la demande du visiteur, sans traitement complémentaire.

Les pages sont « en dur » sur le serveur.

Les limites d'un tel site sont :

- maintenance difficile (modification manuelle de chaque page) ;
- pas de page personnalisée renvoyée à l'utilisateur.

Ces sites peuvent contenir des éléments animés et des scripts

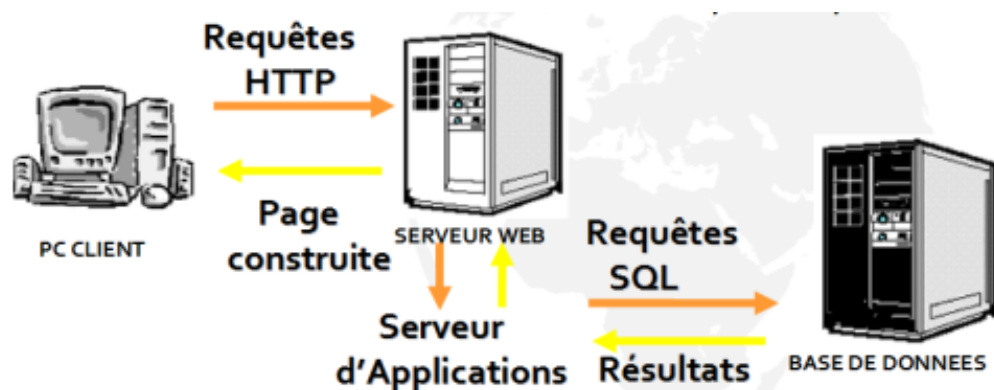


## Les sites dynamiques

Les sites web sont couplés à des serveurs d'applications qui permettent d'exécuter des opérations à la demande et d'adapter leur réponse aux besoins d'un visiteur.

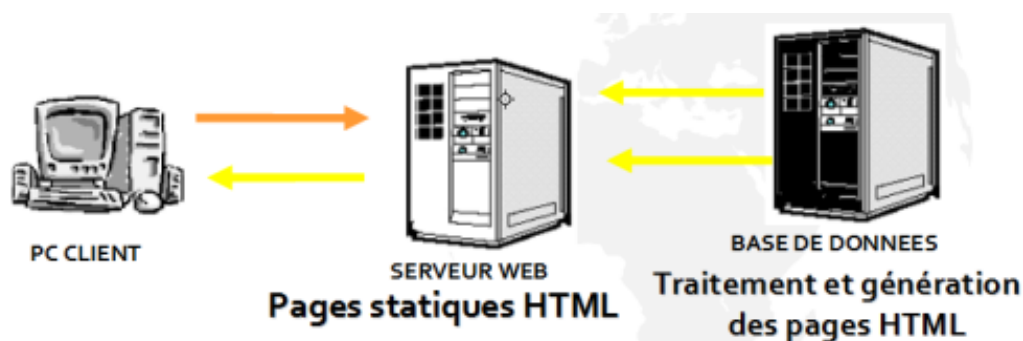
La page est construite à la « volée » par le serveur.

Il faut utiliser un langage de programmation côté serveur (ASP, PHP, JAVA, ...).



## Les sites semi-dynamiques

Les pages HTML sont générées automatiquement à partir d'une base de données en local. Puis les pages sont mises en ligne. Lors des modifications des données le programme de génération des pages est relancé.



Cf. Les tuiles d'images qui sont mises en cache sur un serveur (Google Map par exemple)

## Les applications clientes

Le traitement des données se fait chez le client (avec ou sans serveur) ;

Il faut étendre les possibilités du navigateur Web par l'ajout de plugs-in, applet .. qui permettent la gestion d'autres formats de données ;

Le client récupère du serveur programme et données.



\* \*

\*

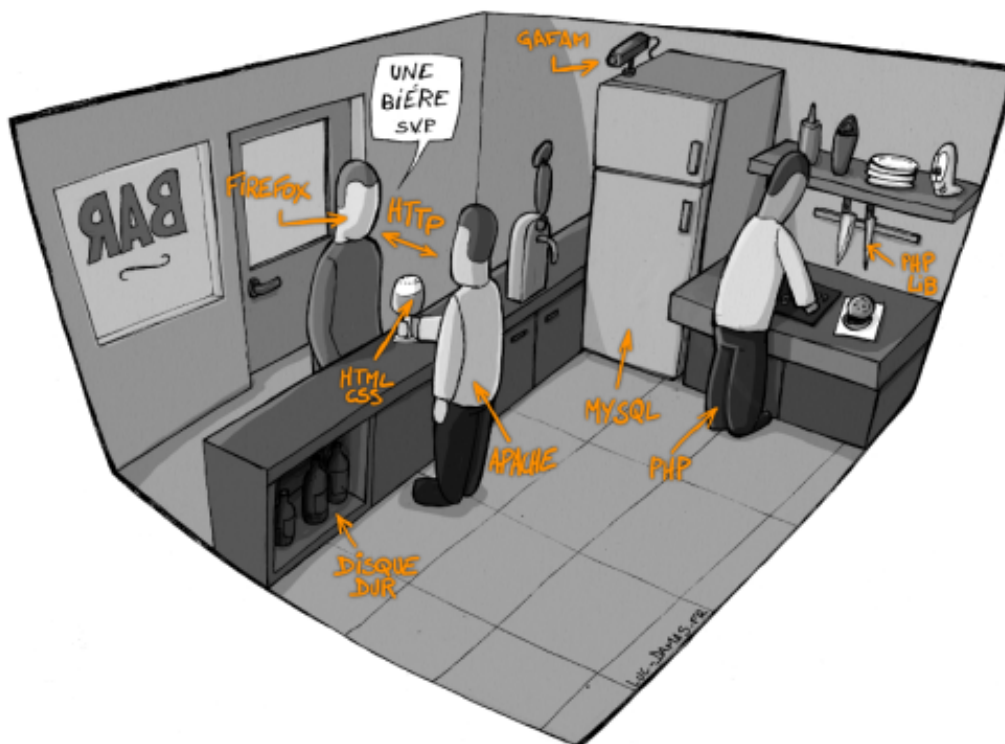
Pour visualiser des sites cartographiques, il faut :

- Un ordinateur (PC, Mac ou station) ;
- Une connexion Internet par le biais d'un modem ;
- Un navigateur (Internet Explorer, Netscape, FireFox, Chrome,...) ;
- Des modules (plugin) ajoutés au navigateur pour pouvoir consulter certains sites.



Pour proposer ses propre données sur internet, il faut :

- Un hébergeur qui va accueillir vos pages web et qui possède un serveur web afin de diffuser vos pages et de répondre aux demandes des internautes (requêtes) ;





# Le webmapping

# 1. Le webmapping

## 1.1. Qu'est-ce que le webmapping ?

### Définition

---

C'est la diffusion par le biais d'un site web de données cartographiques ;

L'accès à l'information est dynamique : ce qui s'affiche doit être le résultat d'un traitement déclenché à la demande de l'utilisateur ;

### Petite histoire du Webmapping

Le webmapping va bientôt fêter ses 20 ans. Le développement est concomitant du développement du web et de l'accès amélioré et évolutif du web (développement des infrastructures, donc des développements et donc des usages).

Ainsi, nous considérons être dans la deuxième génération des outils de webmapping.

- La première génération a permis de mettre en ligne des données sur des cartes. Les données étaient gérées par le concepteur, stockées sur le serveur pour être diffusées de manière "statique" ou peu évolutive.
- La deuxième génération permet l'interaction entre la donnée et le ou les utilisateurs. Les données peuvent être croisées et les fonctionnalités tentent de se rapprocher des capacités d'un SIG bureautique.

Le webmapping participe de l'effort de l'ouverture et du partage des données. On parle d'interopérabilité et de travail collaboratif.

Open GIS Consortium, OGC, est une association mondiale née en 1994 qui s'attache à harmoniser l'information géographique et qui a imposé des standards d'interopérabilité : <https://www.opengeospatial.org/standards/>

### Fonctionnalités d'une application webmapping

Au minima :

- Cartographier des données géographiques avec choix des couches et de l'emprise dans un navigateur ;

Souvent :

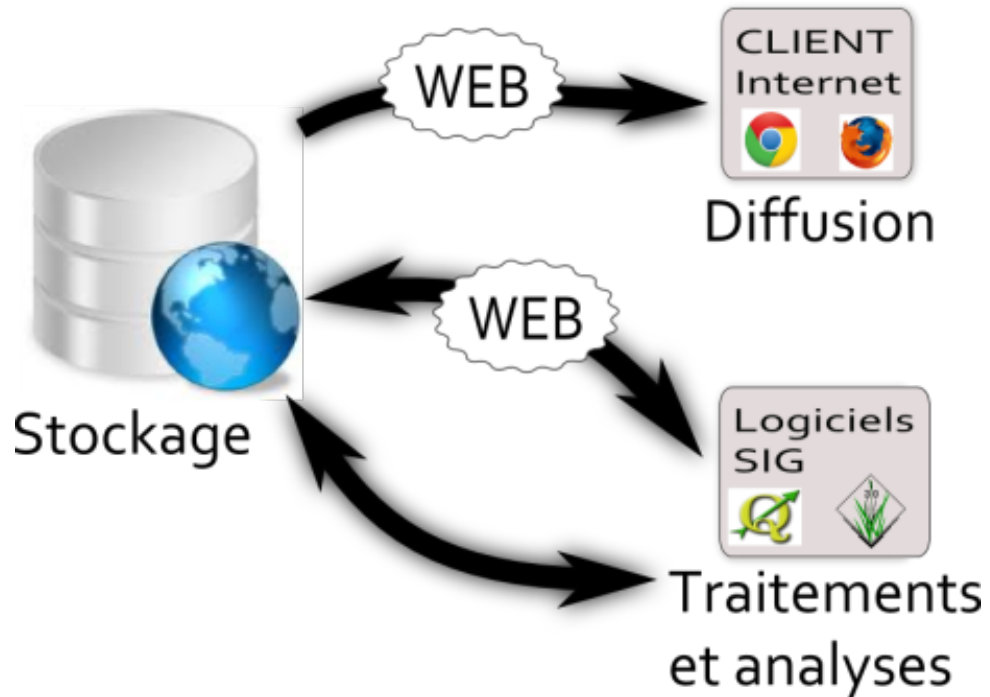
- Effectuer des mesures, interroger les données ;
- Localiser une adresse, faire des requêtes ;

Plus rarement :

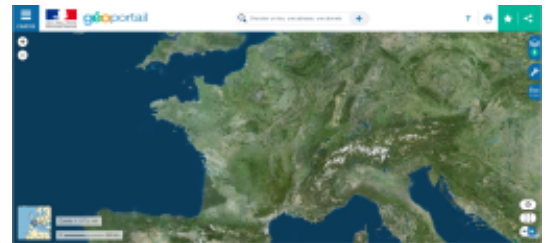
- Saisie et modification des données, requêtes spatiales.

Les données SIG doivent pouvoir être accessibles pour la diffusion et la mise à jour.





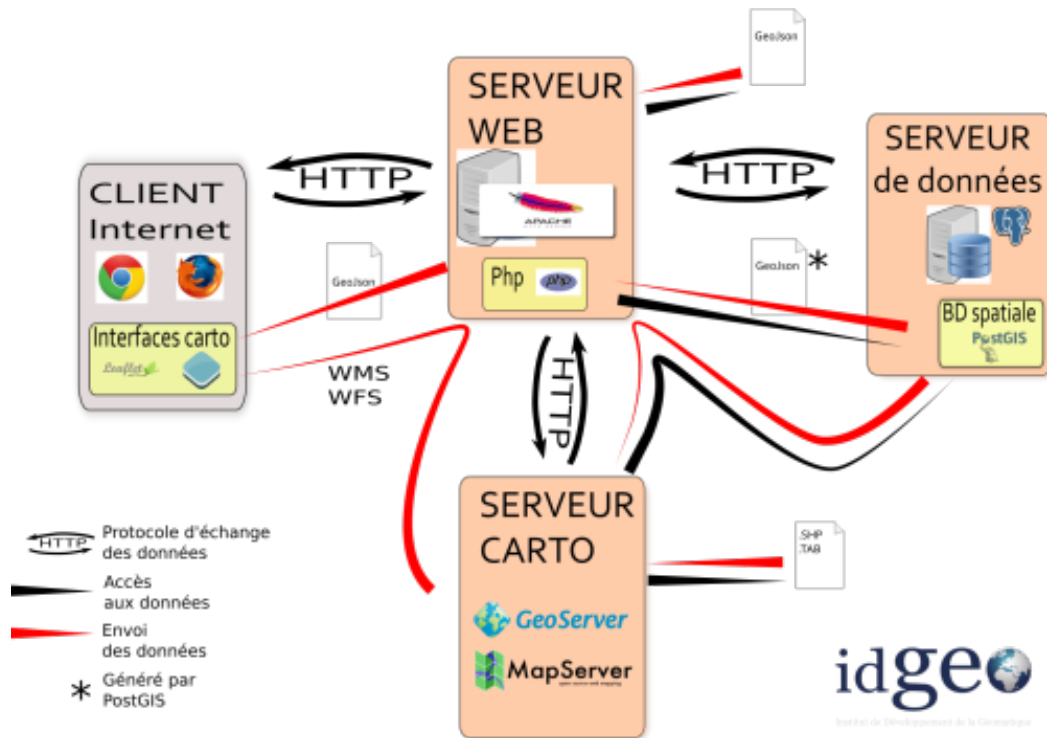
Testons le géoportail <http://www.geoportail.gouv.fr>  
 Les fonction de base ( naviguer, sélectionner des couches,  
 consulter la légende, l'échelle ...) => consulter des  
 données !



### Remarque

Les technologies utilisées sont avant tout celles du web :

- architecture client-serveur :
  - le client est un navigateur capable d'interpréter du code HTML ;
  - le serveur est à la fois une machine qui stocke des documents et des programmes susceptibles de les générer à la demande et un programme permettant le dialogue avec le client.
- protocole HTTP pour les échanges entre le client et le serveur
- stockage des données dans des fichiers et des bases relationnelles (PostgreSQL/GIS, SQLite, SpatialLite Géopackage) interrogeables à l'aide du standard SQL.
- les standards HTML, JavaScript, AJAX, XML, SVG élaborés par le World Wide Web Consortium (W3C) pour le développement de l'application à proprement parler.



## 1.2. Client cartographique

### 🔑 Définition

Le client cartographique : une interface de consultation de données cartographiques situées sur un serveur

- Visualisation de données images et vecteurs géolocalisées ;
- Interactions avec l'utilisateur (zoom, déplacement, choix des couches à afficher) ;
- Création ou modification d'objets vecteur.

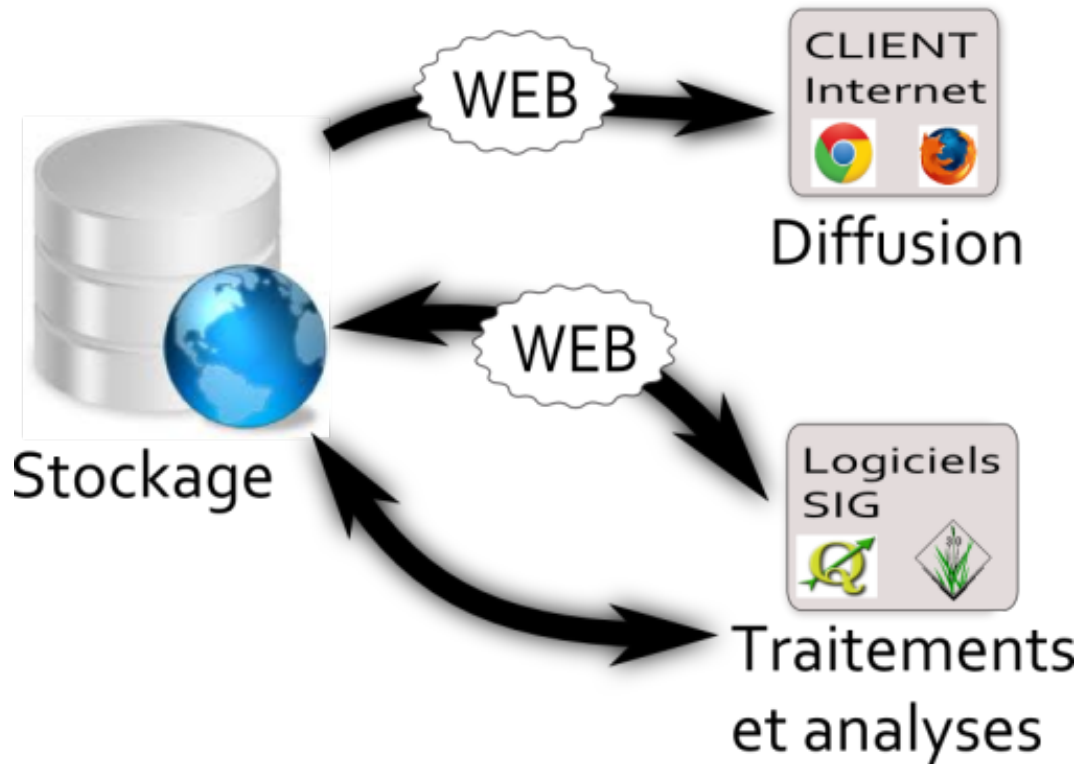
### Deux types de clients cartographique

Les clients web « légers » plutôt orientés consultation :

- interfaces web (navigateur web) ;
  - OpenLayers, Leaflet, LizMap, ArcGIS on Line...

Les clients « lourds » pour le traitement et l'analyse :

- logiciels SIG.
  - QGIS, MapInfo, ArcGIS



## 2. Rendre disponible les données géographiques

Comment rendre les données de votre système d'information géographiques accessibles pour les personnes d'un service, l'ensemble des communes d'une collectivité ou pour un public élargi ?

Les interfaces cartographiques permettent de représenter la données sur une page web, les serveurs cartographiques vont pouvoir mettre à disposition cette données sous forme de flux qui vont être utilisés par ces interfaces de manière dynamique, c'est-à-dire que les données visualisées sur la page web correspondent aux dernières données à jour dans le SIG.

Comment représenter une information géographique dans un navigateur internet ?

### 2.1. Formats vecteurs VS formats rasters

Les navigateurs web peuvent lire une image raster depuis longtemps mais le format vecteur n'était pas prévu pour être lu dans une page web.

Nous allons voir les différents formats raster et vecteur disponibles sur les navigateurs les plus récents pour afficher des données cartographiques.

#### 2.1.1. Format raster

Les formats d'images bitmap pour Internet :

- JPEG (Joint Photographic Experts Group) :
  - 16 M de couleurs, compression avec dégradation ;

- Gif (Graphics Interchange Format) :
  - 256 couleurs, faible compression sans perte, transparence, animations ;
- PNG (Portable Network Graphics) :
  - 16 M de couleurs, compression sans perte, transparence.

Les images lourdes (photographies aériennes, images satellites) nécessiteront une compression plus importante (cf ECW).

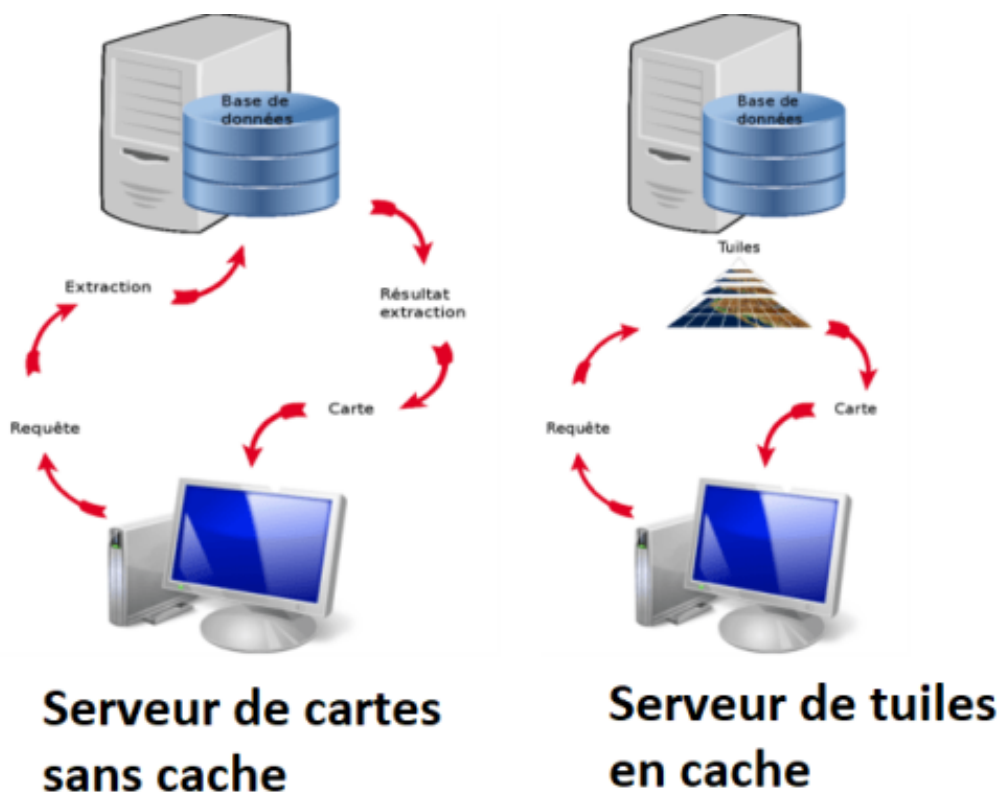
#### Remarque : Le tuilage

De plus en plus, la technologie du tuilage est utilisée pour réduire le temps de chargement et l'utilisation du serveur.

### 2.1.2. Web Map Tile Service (WMTS)

#### Définition

WMTS : standard décrivant la manière de diffuser des données cartographiques sous forme de tuiles prédéfinies à partir d'un serveur de données cartographique distant.

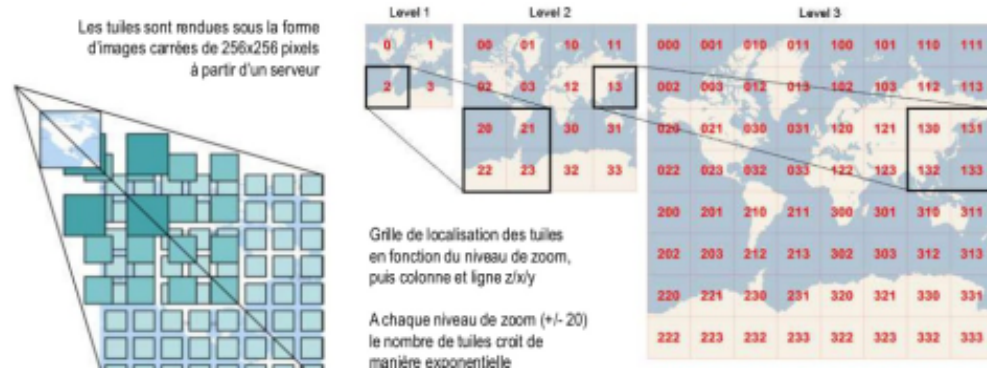


le WMTS met l'accent sur la performance et permet de récupérer des images pré calculées (mises en cache) appartenant à des dallages prédéfinis (tuiles).

Néanmoins, les tuiles peuvent être stockées à la volée, c'est-à-dire générées par la demande des utilisateurs => si l'image n'existe pas sur le serveur de tuile (ou de stockage), la requête demande une image (ou des images, suivant l'emprise (extent) et la taille de l'écran) qui suivant la configuration sera stockée en dur ou mémoire cache (effacée régulièrement)

ATTENTION => un serveur de tuile avec un cache peut devenir gourmand en espace de stockage !

Le principe du tuilage d'une image raster :



## 🔑 Définition : Comment réaliser une pyramide d'images

Il existe quelques solutions intégrées aux solutions du marché :

- Mapserver => MapCache (actuellement 1.10)
- GeoServer => GeoWebCache
- Qgis => utilitaire de tuilage dans les menus contextuel, ou par le geoprocessing Gdal2Tiles
- ArcGis => utilitaire intégré : Gérer le cache des tuiles
- Gdal (évidemment)
- IGN => Librairie Be4 / Rock4

WMTS supporte 3 requêtes :

GetCapabilities : renseigner sur le contenu du service WMTS ;

GetTile : permet à un client WMTS de demander une tuile ;

GetFeatureInfo : permet de récupérer les informations à l'endroit du clic.

Les 2 premières sont obligatoires, alors que GetFeatureInfo est optionnelle.

## 👉 Exemple : Le GetCapabilities du geoportail !

<https://wxs.ign.fr/choisirgeoportail/geoportail/wmts?SERVICE=WMTS&REQUEST=GetCapabilities>

### GetTile

**SERVICE** : le nom du service, en l'occurrence WMTS. Paramètre obligatoire.

**REQUEST** : le nom de la requête, en l'occurrence GetTile. Paramètre obligatoire.

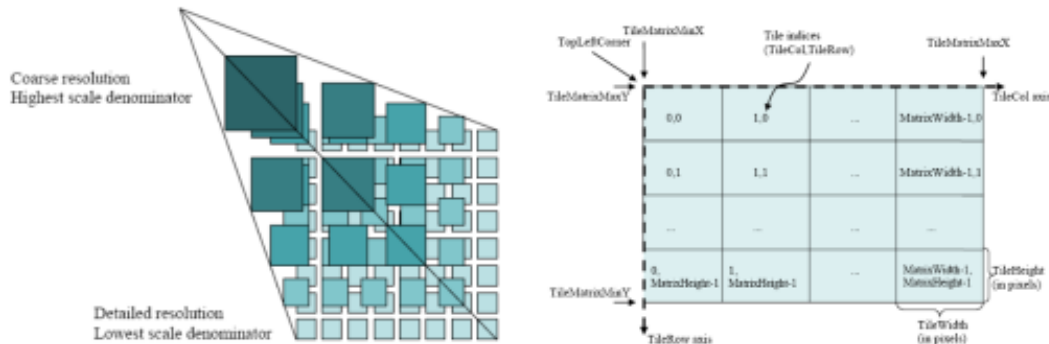
**VERSION** : le numéro de version. Actuellement, la version 1.0.0 est valable. Paramètre obligatoire.

**LAYER** : le nom de la couche concernée. Paramètre obligatoire.

**STYLE** : un nom de style prédéfini mis à disposition par le service WMTS. Paramètre obligatoire.

**FORMAT** : le format d'image à renvoyé. Paramètre obligatoire.

**tileMatrixSet** : un jeu de tuile pour la couche prédéfinie. Il s'agit d'une référence à l'entier de la pyramide. Paramètre obligatoire.



### Exemple : Image satellite d'un glacier

17 images tif de 1,7 Go en moyenne visualisées sur Internet : <http://www.icelegacy.com/my-product/st-elias/>

On peut apprécier la fluidité de l'affichage malgré le poids des images d'origines !

Trouver une URL d'une tuile à partir de site (console de développement (F12), onglet network ou réseau) :

[http://icelegac.cluster013.ovh.net/wp.icelegacy.com/try\\_map/Tuile/tuile\\_img\\_2016\\_6m/13/800/5837.png](http://icelegac.cluster013.ovh.net/wp.icelegacy.com/try_map/Tuile/tuile_img_2016_6m/13/800/5837.png)

### 2.1.3. Format vecteur

#### Définition

Des objets individualisés et redessinés en fonction du zoom :

- nécessite obligatoirement une technologie supplémentaire (plugin, viewer,...) .

Plusieurs formats vecteurs :

- SVG, GML : dérivé du XML ;
- GeoJSON (Geographic Javascript Object Notation) : Format ouvert d'encodage des objets et données attributaires ;
- KML (Keyhole Markup Language): Dérivé du XML, standard d'échange des données sur le web (utilise avec Google Earth) ;

#### Exemple : SVG

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<svg width="296.926mm" height="209.973mm"
viewBox="0 0 3507 2480"
xmlns="http://www.w3.org/2000/svg"   xmlns:xlink="http://www.w3.org/1999/xlink"   version="1.2"
baseProfile="tiny">
<g fill="#f0b37d" fill-opacity="1" stroke="#000000" stroke-opacity="1" stroke-width="3.07087" stroke-
linecap="square" stroke-linejoin="bevel" transform="matrix(0.999751,0,0,0.999751,35.4242,23.6162)"
font-family="MS Shell Dlg 2" font-size="34.375" font-weight="400" font-style="normal" >
<path vector-effect="none" fill-rule="evenodd" d="M1336.63,1389.22 L1337.46,1388.29 L1337.24,1387.03
L1338.66,1386.69 L1340.1,1383.39 L1341.42,1383.72 L1343.02,1383.66 L1345.7,1382.06 L1347.03,1380.11
L1346.98,1378.57 L1345.78,1377.58 L1348.91,1372.14 L1348.27,1370.75 L1348.97,1369.48
...
```

#### Exemple : KML

```
<?xml version="1.0" encoding="utf-8" ?>
<kml xmlns="http://www.opengis.net/kml/2.2">
<Document id="root_doc">
```

```

<Schema name="depts" id="depts">
  <SimpleField name="nom_dept" type="string"></SimpleField>
  ...
</Schema>
<Folder><name>depts</name>
  <Placemark>
    <Style><LineStyle><color>ff0000ff</color></LineStyle><PolyStyle><fill>0</fill></PolyStyle></Style>
    <ExtendedData><SchemaData schemaUrl="#depts">
      <SimpleData name="nom_dept">DORDOGNE</SimpleData>
    </SchemaData></ExtendedData>
    <MultiGeometry><Polygon><altitudeMode>relativeToGround</altitudeMode><outerBoundaryIs><LinearRing><coordinates>0.4461050343983,45.514636974487146
0.450892985582893,45.518646844986087 0.449305073488189,45.523917317570792
0.457778691226885,45.525559978184873 0.465736720243763,45.539623253832943

```

### Exemple : GML

```

<?xml version="1.0" encoding="utf-8" ?>
<ogr:FeatureCollection
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation=""
  xmlns:ogr="http://ogr.maptools.org/"
  xmlns:gml="http://www.opengis.net/gml">
  <gml:boundedBy>
    <gml:Box><gml:coord><gml:X>-572072.7873922574</gml:X><gml:Y>5065996.44665878</gml:Y>
    </gml:coord><gml:coord><gml:X>1064194.663851942</gml:X><gml:Y>6637051.972159549</gml:Y></gml:coord> </gml:Box></gml:boundedBy>
    <gml:featureMember>
      <ogr:depts fid="depts.0">
        <ogr:geometryProperty>
          <gml:MultiPolygon srsName="EPSG:3857">
            <gml:polygonMember><gml:Polygon><gml:outerBoundaryIs><gml:LinearRing><gml:coordinates>49660.
185269534551,5702907.7015151987 50193.177557346542,5703544.7436526949
50016.411991539571,5704382.1248145504

```

### Exemple : GeoJSON

```

{
  "type": "FeatureCollection",
  "crs": { "type": "name", "properties": { "name": "urn:ogc:def:crs:EPSG::3857" } },
  "features": [
    { "type": "Feature", "properties": { "gid": 1, "code_dept": "24", "nom_dept": "DORDOGNE" }, "geometry": {
      "type": "MultiPolygon", "coordinates": [ [ [ [ 49660.19, 5702907.7 ], [ 50193.18, 5703544.74 ], [ 50016.41,
5704382.12 ], [ 50959.69, 5704643.13 ], [ 51845.57, 5706877.97 ], [ 52733.73, 5706684.6 ], [ 53794.5,
5706761.55 ], [ 55542.88, 5707886.97 ], [ 56388.79, 5709217.06 ], [ 56323.55, 5710246.27 ], [ 55501.25,
5710877.66 ], [ 57476.89, 5714579.58 ], [ 57016.58, 5715500.9 ], [ 57456.8, 5716359.04 ], [ 57141.61,
5717342.48 ], [ 55820.11, 5718922.1 ], [ 55704.64, 5719592.35 ], [ 56242.08, 5719454.55 ], [ 56597.28,
5719884.51 ], [ 57068.67, 5721485.53 ], [ 58535.45, 5723129.64 ], [ 59602.44, 5723300.84 ],

```



### 2.1.4. Les tuiles vecteurs

#### Principe

Le tuilage vecteur est un mode de diffusion des données analogue au WMTS (de la tuile “image”), mais qui transmet des objets plutôt que des pixels.

Les spécifications du format vector tile a été mis en place par Mapbox, entreprise privée qui développe des applications de webmapping et des services de fond de cartes (<https://www.mapbox.com/about/company/>)

"Comme pour un service WMTS, les données sont pré-découpées par le serveur et diffusées sous forme de dalles vecteur adaptées à l'échelle de visualisation, ce qui permet à l'utilisateur de ne charger que les données sur la zone concernée, d'avoir des données suffisamment généralisées pour être affichées aux petites échelles et de conserver les dalles calculées dans un cache."

[https://geoservices.ign.fr/documentation/services\\_betas/vecteur-tuile.html](https://geoservices.ign.fr/documentation/services_betas/vecteur-tuile.html)

#### Production

ArcGIS Online, Geoserver, Mapserver

#### Utilisations

Avec une interface web carto comme OpenLayers ou Mapbox GL JS ;

Plugin-in QGIS vector tiles reader

#### Avantages

Les dalles vecteurs sont plus légères que les dalles rasters et le choix du style se fait au niveau du client, pas du serveur (personnalisation de l'affichage).

#### Inconvénients

Technologie encore jeune ; pas conçu pour afficher l'emprise complète d'une données vecteur lourde ; perte de précision géométrique pour optimiser la taille des tuiles ; nécessite un PC récent.

## 2.2. HTML et API Javascript

Afin d'élaborer des interfaces de visualisation des données géographiques fluides et dynamiques, le langage de programmation Javascript est utilisé et va permettre d'ajouter plus d'interactivité dans les pages web qu'une simple image de carte ou un fichier pdf.

Le Javascript est un langage de programmation exécuté chez le client.

Vous pouvez créer vous-même vos programmes ou les récupérer par le biais des bibliothèques Javascript (les API).



### 🔑 Définition : Les API

API : Application Programming Interface

C'est un ensemble de fonctions permettant des créer des interfaces homme machine (IHM) facilement ;

- Elle se base sur du web service (flux de données).

Exemple d'API :



- LeafLet,
- OpenLayers,
- GoogleMap,
- LizMap

## 2.2.1. Les interfaces cartographiques Javascript

### LeafLet

Leaflet est un ensemble de fonctions javascript permettant de visualiser simplement des données cartographiques dans une page web et sur un mobile.

Cette bibliothèque utilise le format GeoJSON (qui permet de décrire des données de type point, ligne, chaîne de caractères et polygone dans un format manipulable par le Javascript) qu'elle transforme en SVG pour l'afficher sur le web (mais Canvas peut être utilisé).



### Exemple

```
1 <HTML><HEAD>
2 <link rel="stylesheet" href="http://cdn.leafletjs.com/leaflet-0.7.3/leaflet.css" />
3 <script src="http://cdn.leafletjs.com/leaflet-0.7.3/leaflet.js"></script>
4 </HEAD>
5 <BODY>
6 <div id="carte" style="height:400px;"></div>
7 <SCRIPT>
8 var map = L.map('carte').setView([43.566,1.383], 13);
9 L.tileLayer('http://{s}.tile.osm.org/{z}/{x}/{y}.png').addTo(map);
10 L.marker([43.566182,1.38367]).addTo(map);
11 </SCRIPT>
```

[cf. Exemple de l'API Leaflet]

### OpenLayers

Bibliothèque de fonctions JavaScript permettant la mise en place d'applications cartographiques fluides.

Permet d'afficher des fonds cartographiques tuilés ainsi que des marqueurs provenant d'une grande variété de sources de données.

Elle utilise CANVAS pour afficher les données sur la page HTML



### Exemple

```
1 <HTML><HEAD>
2 <script src="/scripts/ol_v5/ol.js" type="text/javascript"></script>
3 <link rel="stylesheet" href="/scripts/ol_v5/ol/ol.css" type="text/css">
4 <div id="carte"></div>
5 <script >
6 var map = new ol.Map({
7   layers: [ new ol.layer.Tile({ source: new ol.source.OSM() }) ],
8   target: 'carte',
9   view: new ol.View({ center: [0, 0], zoom: 2 })
10 });
11 </script >
12 </BODY>
13 </HTML>
```

[cf. Exemple de l'API OpenLayers 4]

## Google Map

Google Maps est un service de carte géographique et de plan en ligne.

- Le service a été créé par Google. Il s'agit d'une forme de géoportail. Lancé en 2004 aux États-Unis ;

Caractéristiques :

- à partir de l'échelle d'un pays, on peut zoomer jusqu'à l'échelle d'une rue.
- Deux types de plan sont disponibles : un plan classique, avec nom des rues, quartier, villes et un plan en image satellite, qui couvre aujourd'hui le monde entier.

### Exemple

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <title>Simple Map</title>
5   <meta name="viewport" content="initial-scale=1.0">
6   <meta charset="utf-8">
7   <style>
8
9   #map {
10     height: 100%;
11   }
12
13   html, body {
14     height: 100%;
15     margin: 0;
16     padding: 0;
17   }
18 </style>
19 </head>
20 <body>
21   <div id="map"></div>
22   <script>
23     var map;
24     function initMap() {
25       map = new google.maps.Map(document.getElementById('map'), {
26         center: {lat: 43.566182, lng: 1.38367},
27         zoom: 8
28       });
29     }
30   </script>
31   <script src="https://maps.googleapis.com/maps/api/js?key=CODE_API_GOOGLE&callback=initMap"
32     async defer></script>
33 </body>
34 </html>
```

[cf. Exemple de l'API Google Map]

### Remarque

La politique tarifaire de Google pour son service de cartographie Google Map a été modifiée en juillet 2018 : gratuit à sa création en 2004, le service est devenu progressivement payant.

Depuis 2015, le quota gratuit était de 25 000 appels par jour, contre 1000 par jour depuis juillet 2018. Dans tous les cas, Google Maps impose aux entreprises utilisant ses services de renseigner un numéro de carte bancaire, même dans le cas d'une utilisation gratuite.

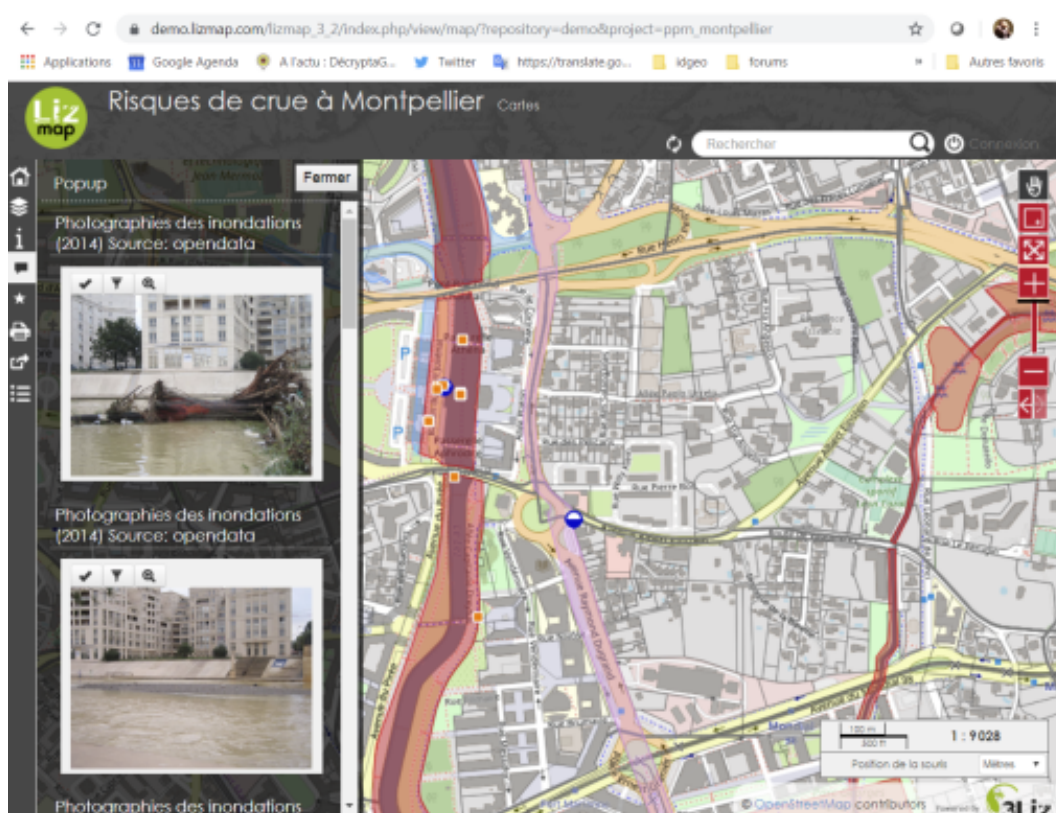
Voir la grille tarifaire

## LizMap

LizMap se base sur Openlayers 2 et sur un ensemble de bibliothèques afin de fournir une interface complète sans avoir à développer.

Exemple :

[https://demo.lizmap.com/lizmap\\_3\\_2/index.php/view/map/?repository=demo&project=pprn\\_montpellier](https://demo.lizmap.com/lizmap_3_2/index.php/view/map/?repository=demo&project=pprn_montpellier)



### 2.3. Les serveur web

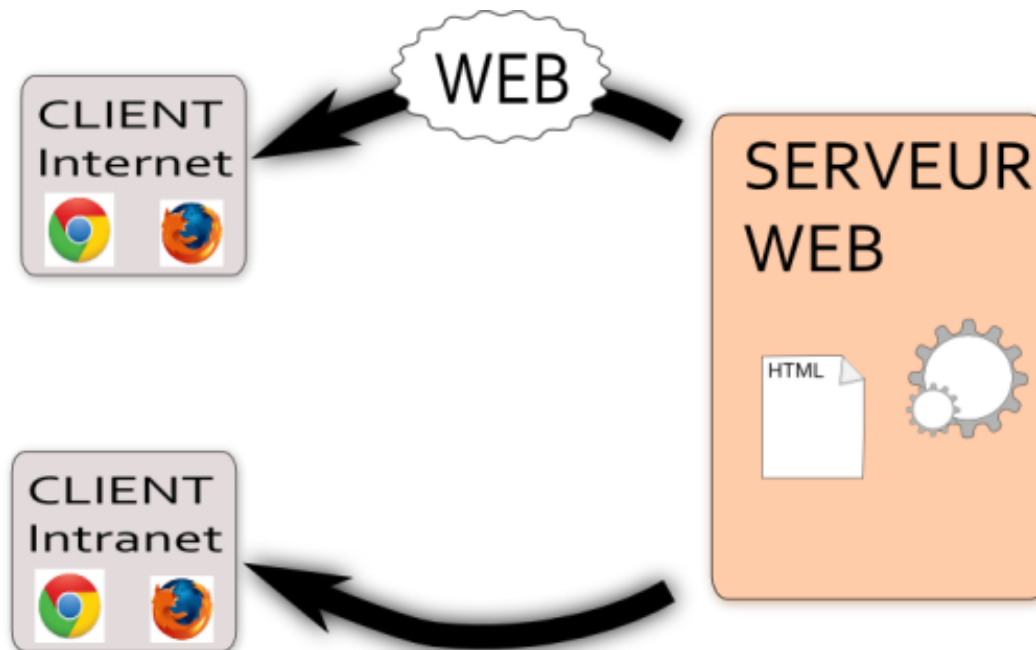
#### Objectifs

Pour publier des données cartographiques sur le web, on peut récupérer un flux de données par les web services : il faut un serveur cartographique.

L'objectif de cette partie est de comprendre le fonctionnement d'un serveur cartographique et des flux de données qu'il peut fournir.

Les serveurs web :

- Un serveur web accueille les sites écrits en HTML.
- Il permet de rendre ces sites interactifs par l'intermédiaire de programmes (CGI) qui créent des pages HTML à la volée suivant la demande de l'utilisateur.



Intérêts de l'utilisation de CGI :

- Fournir des pages dynamiques, c'est-à-dire des pages pouvant être différentes selon un choix ou une saisie de l'utilisateur.

L'application la plus fréquente :

- L'utilisation de formulaires HTML dont les données entrées vont permettre de générer une page personnalisée par le CGI

Exemple de programmes CGI : Perl, C, C++, PHP, ASP

### 2.3.1. Le serveur web carto

#### Définition

Un serveur cartographique est un programme qui prend les données du SIG pour les mettre à disposition du client internet.

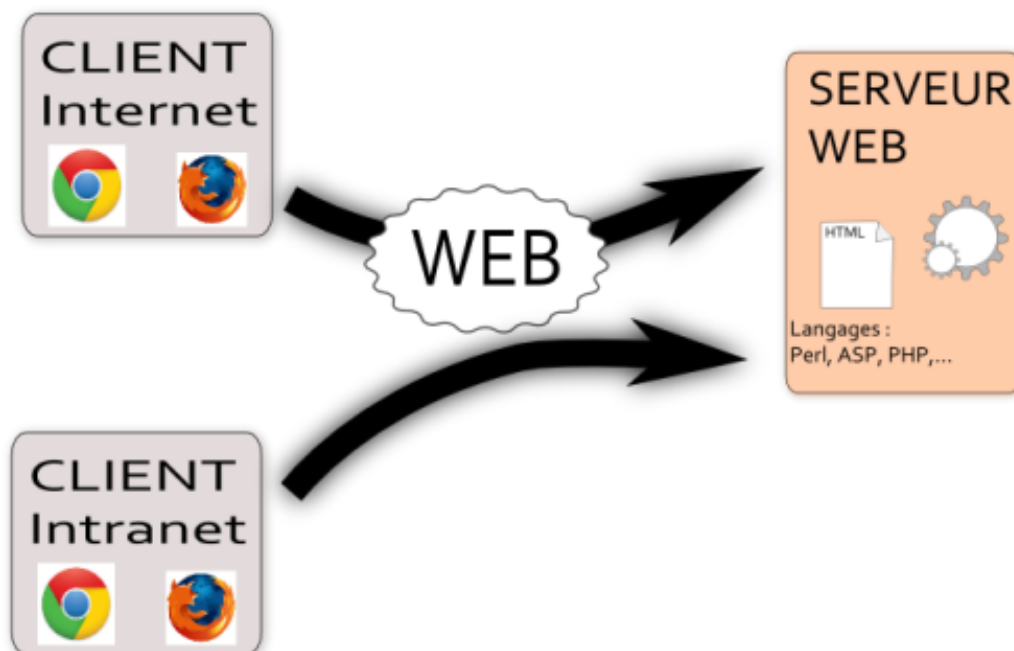
Le serveur Web réceptionne les requêtes des clients et les envoie au serveur Web carto qui traite les requêtes.

Le résultat du traitement est renvoyé au serveur web qui les retourne au client.

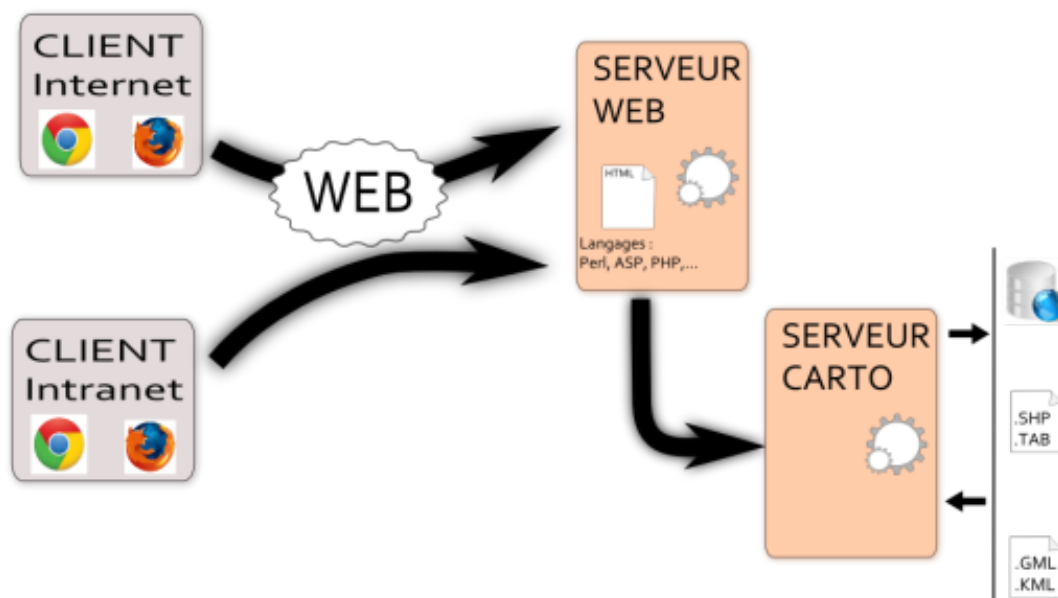
Exemples de serveurs web carto :

-  ArcGIS Server (ESRI) ;
- MapServer, Geoserver, QGIS server (libre) ;
- GeoConcept Internet Server ;
- ...

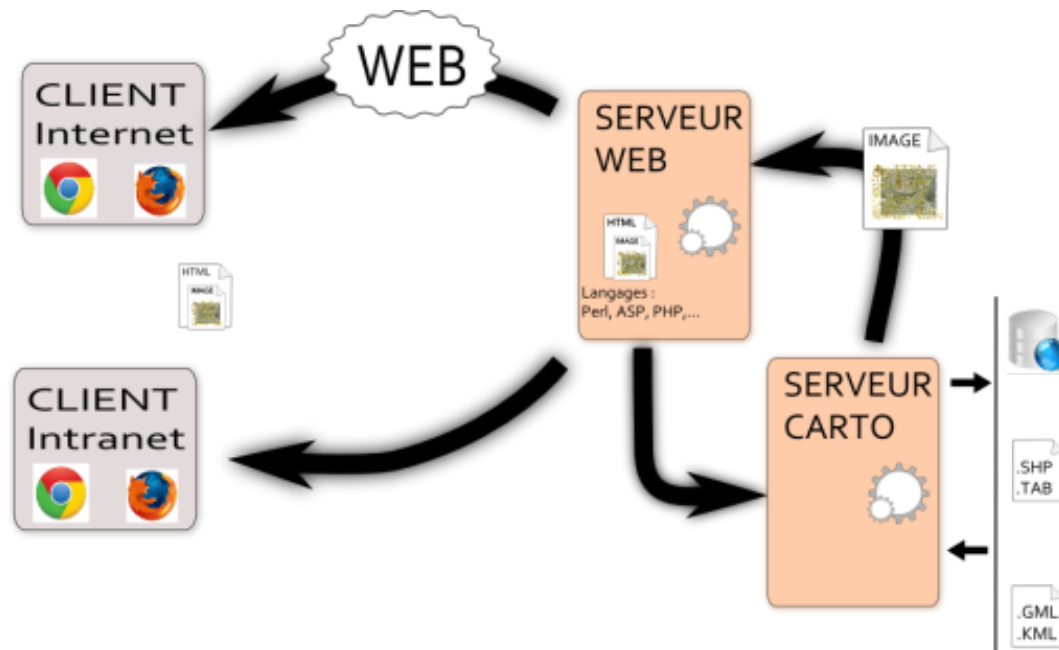
La requête spatiale est envoyée au serveur web



Le serveur WEB passe le relais au serveur carto



Le serveur carto, à partir des données, crée une image ou un flux de données que le serveur web transmet au client



### Remarque

On parle de flux de données car, à chaque requête du client (déplacement sur la carte, zoom avant ou arrière), de nouvelles données sont générées et envoyées au client par le serveur.

### a) serveurs web Carto libres

MapServer :

- Génération d'images et de services web.

GeoServer :

- Service web transactionnel .

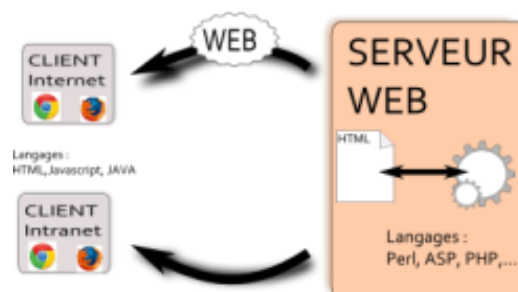
QGIS Server :

- Service web (WMS).

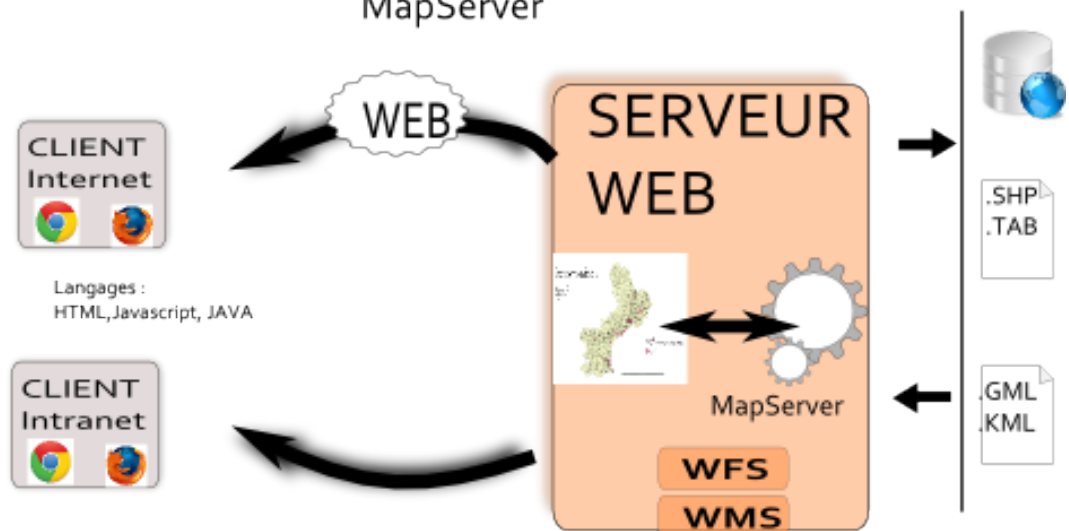
### i MapServer/QGIS server :

Génération d'images et de services web

Fonctionnement d'un serveur web  
avec programmes CGI



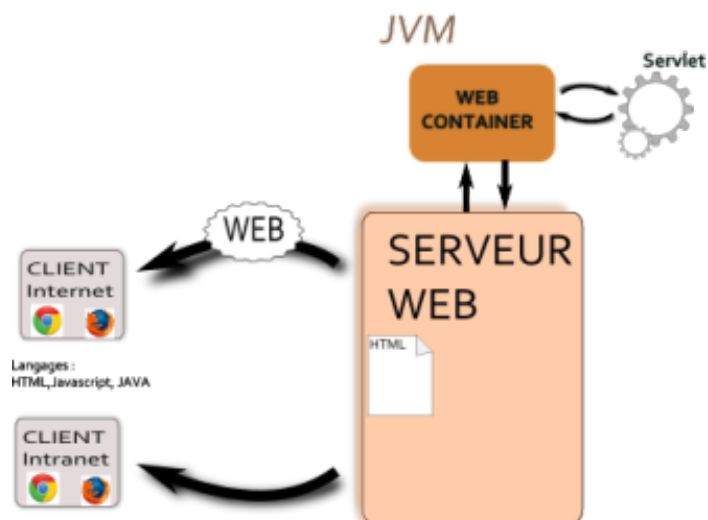
## Fonctionnement d'un serveur Carto MapServer



### ii GeoServer

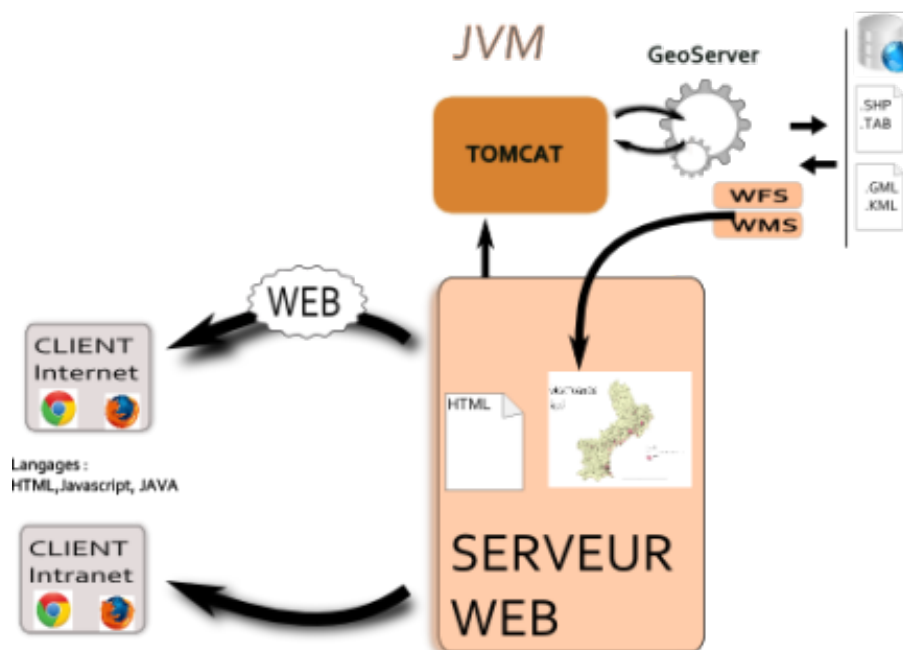
Service web transactionnel

## Fonctionnement d'un serveur web avec servlets JAVA





## Fonctionnement d'un serveur Carto GeoServer



### 2.3.2. Les webservice

#### 🔑 Définition : Services WEB :

Un "web service" est un programme informatique permettant la communication et l'échange de données entre applications et systèmes hétérogènes sur internet ou intranet. Il regroupe tout un ensemble de fonctionnalités accessibles sur le web par et pour des applications ou machines, sans intervention humaine et en temps réel.

En géomatique un web service propose un service qui va permettre d'accéder à des données distantes géoréférencées, soit pour la consultation, c'est-à-dire l'affichage simple de carte (WMS, WMTS), soit pour du stockage de données (WCS et WFS) soit pour du traitement distant pour éviter d'utiliser du temps d'utilisation du processeur et de la mémoire (WPS), soit pour accéder aux méta-données (CS-W).

L'Open GeoSpatial Consortium (OGC)

Organisme fondé en 1994 qui développe de manière collaborative des standards d'interfaces ouvertes (une récente GeoNode) et des standards d'encodages associés (Web Services) pour faciliter l'échange de l'information (interopérabilité).

Défini des protocoles d'échanges de données géographiques :

- WMS : Web Map Services => Renvoi une image
- WFS : Web Feature Services => pour les données vecteurs ;
- WMTS : Web Map Tilling Services => pour accélérer le chargement des données, mise en cache et tuilage des images;
- TMS : Tile Map Service => service de tuilage stockées sous la forme z/x/y
- WFS-T : Web Feature Services - Transactionnel => pour modifier les données vecteurs ;
- WCS : Web Coverage Service => pour récupérer des données raster sur une emprise donnée;
- CSW : Catalog Service Web : accéder aux catalogues de ressources (métadonnées - Geonetwork) ;
- WPS (WCPS) : Web Processing Service : fourni des service pour faire des traitements par le web.

Le Vecteur Tuilé - Spécification de MapBox

<https://docs.mapbox.com/vector-tiles/specification/>

Les avantages :

- rapidité d'accès
- possibilité de modifier le style, les données sont côté client au format vecteur





## Remarque : Les serveurs web carto

Ce sont donc les serveurs cartographiques, connectés en permanence aux données du SIG qui vont fournir ces flux de données.

### 2.3.3. Web Map Services

Produit dynamiquement des cartes sous forme d'images bitmap (JPEG, PNG et GIF) à partir d'un serveur de données géoréférencées ;

Visualisation via un client qui est capable d'envoyer des requête au serveur cartographique :

- Directement dans le logiciel SIG ;
- Dans une page Web.



## Définition

La spécification WMS normalise la manière selon laquelle des applications clientes doivent demander une carte et la manière selon laquelle les serveurs doivent décrire les données qu'ils sont capables de fournir.

Un serveur WMS doit répondre à 3 requêtes types :

- GetCapabilities
- GetMap
- GetFeatureInfo

### GetCapabilities : métadonnées

Connaître le nom des couches de données, les systèmes de coordonnées dans lesquels on peut les obtenir, les styles disponibles.

La réponse est un fichier XML.

<https://wxs.ign.fr/choisirgeoportail/geoportail/r/wms?SERVICE=WMS&REQUEST=GetCapabilities>

### GetMap : visualisation

Elle permet au client d'obtenir la carte disponible sur le serveur suivant des caractéristiques définies à l'aide de paramètres.

### GetFeatureInfo : consultation

Obtenir les informations attributaires portées par le (ou les) objet(s) localisé(s) là où on a cliqué sur la carte.



## Complément : SLD

Défini par l'OGC, le Styled Layer Descriptor (SLD) est un format de description (en XML) permettant la mise en forme de données géographique provenant d'un flux WMS.

Il joue le même rôle qu'un fichier CSS pour une page HTML

```

1 C'est un fichier XML qui va être associé à un flux WMS :
2 <StyledLayerDescriptor version="1.0.0" xsi:schemaLocation="http://www.opengis.net/sld StyledLayerDescriptor.xsd"
  xmlns="http://www.opengis.net/sld" xmlns:ogc="http://www.opengis.net/ogc" xmlns:xlink="http://www.w3.org/1999
  /xlink" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
3   <NamedLayer>
4     <Name>depts</Name>
5     <UserStyle>
6       <Title>style des départements</Title>
7       <FeatureTypeStyle>
8         <PolygonSymbolizer>
9           <Fill><CssParameter name="fill">#FF7171</CssParameter></Fill>

```

```

10      <Stroke><CssParameter name="stroke">#FF7171</CssParameter><CssParameter name="stroke-width">2.0
      </CssParameter></Stroke>
11      </PolygonSymbolizer>
12    </FeatureTypeStyle>
13  </UserStyle>
14 </NamedLayer>
15 </StyledLayerDescriptor>

```

Les serveurs WMS : MapServer, GeoServer, Qgis Server

### 2.3.4. Web Feature Services

Permet à un client d'utiliser des objets géographiques chargés depuis un serveur cartographique distant au moyen de requêtes. Il pourra ainsi de réaliser des manipulations sur ces objets :

- Modifier l'apparence des objets dans un logiciel SIG ou dans une interface web carto ;
- Créer, modifier ou supprimer un nouvel objet géographique situé sur un serveur distant ;



#### Définition

Un serveur WFS doit répondre à 5 types de requêtes :

- GetCapabilities
- DescribeFeatureType
- GetFeature
- LockFeature
- Transaction



#### Complément : Les versions

La version 1.0.0 de la spécification WFS nécessite l'utilisation de la version 2.1 de GML, tandis que la version 1.1.0 de la spécification WFS requiert l'utilisation de la version 3.1.1 de GML.

Pour les deux versions de la spécification WFS, un nombre arbitraire d'autres codages peut également être défini, en plus du format GML 2.1.2 ou 3.1.1 requis (pour 1.0.0 et 1.1.0 respectivement).

GML 2.1.2 prend en charge le codage des 'primitives' géométriques de base: points, lignes, polygones, etc.

GML 3.1.1 prend en charge le codage pour des représentations géométriques plus avancées: courbes, surfaces, multi-dimensions (heure, élévation, imagerie multi-bandes).

De plus, GML 3.1.1 inclut la prise en charge du codage pour les jeux de données intégrés de manière topologique.

#### GetCapabilities

Connaître le nom des couches de données, les systèmes de coordonnées dans lesquels on peut les obtenir, la sémantique disponible.



#### Exemple : Le GetCap de l'IGN

<https://wxs.ign.fr/choisirgeoportail/geoportail/wfs?SERVICE=WFS&REQUEST=GetCapabilities>

#### DescribeFeatureType

Connaître le genre graphique des couches de données, l'emprise.

#### GetFeature

Elle permet au client d'obtenir les objets vectoriels sous forme de flux XML (GML).

Les données transmises sont codées avec le langage GML( Geography Markup Language ), dérivé du XML pour gérer les données spatiales.

sources : <https://georezo.net/wiki/main/standards/wfs>

### Remarque : Deux types de Web Feature Service

- WFS basique : seulement les trois premières opérations (GetCapabilities, DescribeFeatureType et GetFeature). Il peut être considéré comme un service WFS "en lecture seule".
- WFS avec transactions ou transactionnel : service WFS qui supporte toutes les opérations d'un WFS basique et en plus implémente les opérations de transaction.

Les trois types de transaction possibles avec WFS sont INSERT (création), UPDATE (mise à jour) et DELETE (suppression). Cependant seule la suppression est considérée pertinente avec le binding HTTP GET / KVP. En effet l'envoi de données vecteur encodées en KVP n'est pas pertinent.

Les serveurs WFS transactionnels offrent de grandes possibilités :

ils permettent à quiconque dispose d'une connexion internet et des droits appropriés de mettre à jour à distance, éventuellement en situation de mobilité, des données partagées.

Cette technologie est utilisée par tous les portails collaboratifs, dont le plus connu d'entre eux OpenStreetMap.

Serveurs WFS-T : Geoserver, Qgis Server

### LockFeature

Permet de bloquer des objets lors d'une transaction.

### Transaction

Permet de modifier l'objet (création, mise à jour, effacer).

### 2.3.5. Les autres web services de l'OGC

#### a) Le Web Coverage Service (WCS)

Permet de recevoir l'information géographique sous forme de grille de données (coverage);

Similaire au WMS mais :

- Permet de retourner toutes les informations sur l'image brute ;
- Le résultat peut être utilisé dans des processus de traitements complexes.

#### Définition

---

- GetCapabilities : liste des données, des opérations et des paramètres ;
- DescribeCoverage : Document XML qui décrit la couverture demandée ;
- GetCoverage : Retourne la données dans différents formats :
  - Images : JPEG,GIF,PNG,TIFF,BMP ;
  - Données géoréférencées : GeoTiff, Gtopo30, ArcGrid

#### b) Le Catalog Service for the Web (CSW) :

Fonctions de gestion et d'accès à des ressources spatialisées par le web;

- Le catalogue regroupe l'ensemble des descriptions des ressources que le gestionnaire du catalogue (qui peut être directement le producteur de données ou un acteur tiers) souhaite référencer ;

#### Définition

---

3 types de requêtes :

- GetCapabilities : Permet de retourner les métadonnées du service CSW (description du service, opérations mises en œuvre, critères de recherches possibles,...) ;
- GetRecords: Permet de réaliser une recherche de métadonnées dans le catalogue et de récupérer les informations (flux XML);
- GetRecordById : Permet d'accéder à une métadonnée précise à partir de son identifiant ;

#### c) Le Web Processing Service (WPS)

Normaliser les appels et les réponses de services de traitements de données SIG sur le web.

#### Définition

---

3 types de requêtes :

- GetCapabilities : liste les traitements, les opérations et méthodes supportées ;
- DescribeProcess : description d'une procédure (paramètres d'entrée et réponses) ;
- Execute : exécuter une procédure avec des paramètres d'entrée (DataInputs) et de sortie (RawdataOutput ou ResponseDocument)

## 3. Les solutions en ligne

Les solutions en ligne ont pour caractéristiques de prendre vos données pour les mettre sur un serveur web où une interface cartographique, plus ou moins paramétrable, permet de les visualiser.

Les solutions propriétaires :

- Les sociétés de services qui développent, hébergent votre site (exemple : NetAGiS, GeoClip);
- Les outils spécifiques :

- ArcGISOnline, CartoDB, ...

Les solutions libres :

- Umap, QGIS Cloud, ...

### 3.1. Les solutions open-sources

#### Umap

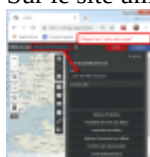
Permet de créer des cartes personnalisées sur des fonds Open Street Map

<http://umap.openstreetmap.fr/fr/>



**Exemple : Créer une carte des emplacements pour Personne à Mobilité Réduite (PMR) sur Toulouse Métropole**

1. Récupérer les données sur <https://data.toulouse-metropole.fr/explore/dataset/pmr/export/> au format GeoJSON ;
2. Sur le site umap, créer une carte et modifier son nom :



3. Définir l'emprise initiale :



4. Importer les données GeoJSON :



5. Modifier les propriétés de la forme par défaut :  
prendre par exemple <http://pochoirs-signalisation.e-monsite.com/medias/images/picto-marquage-handicape.jpg> ;



6. Enregistrer et sortir de l'édition :



#### QGIS Cloud:

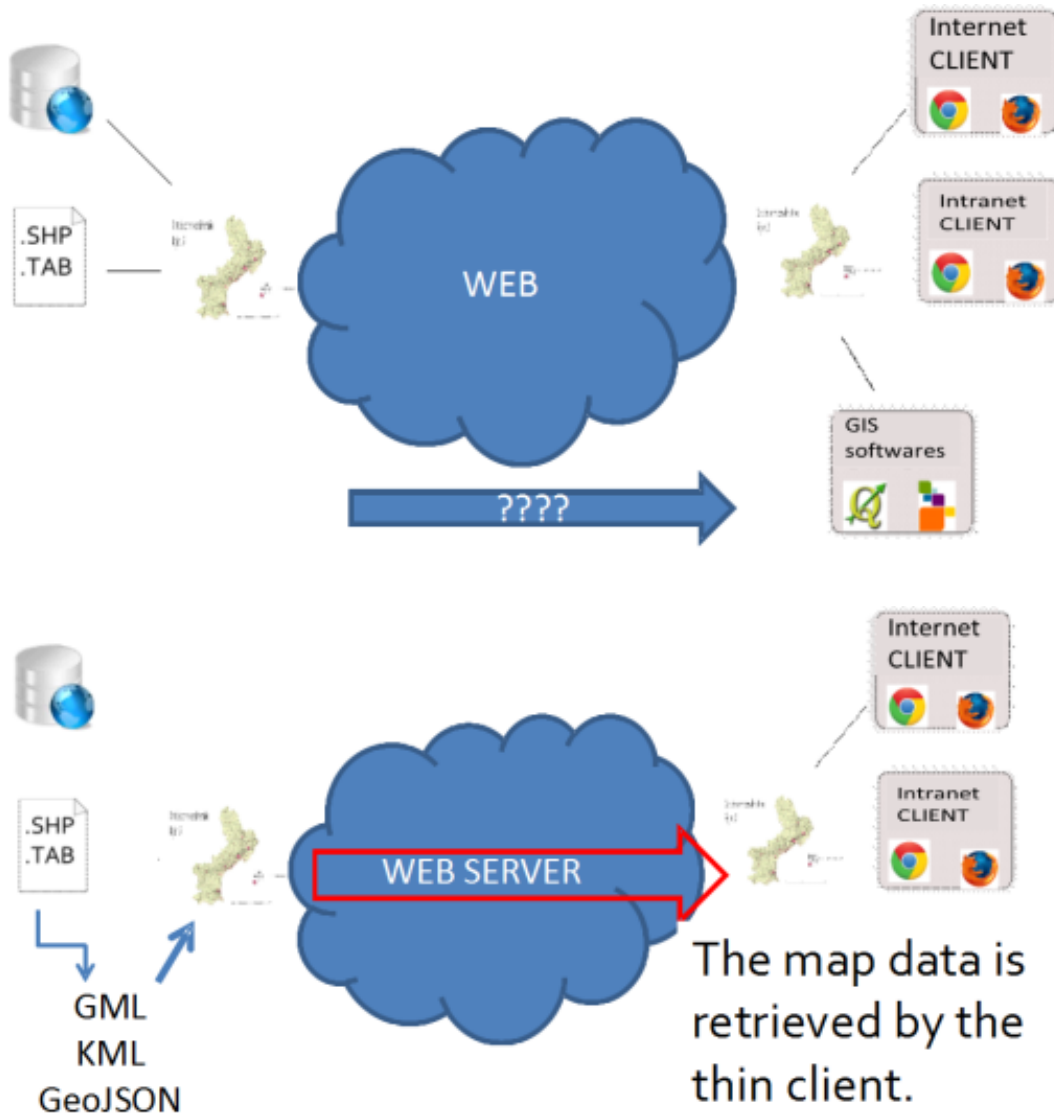
Propose une solution en ligne pour publier ses cartes sur Internet.

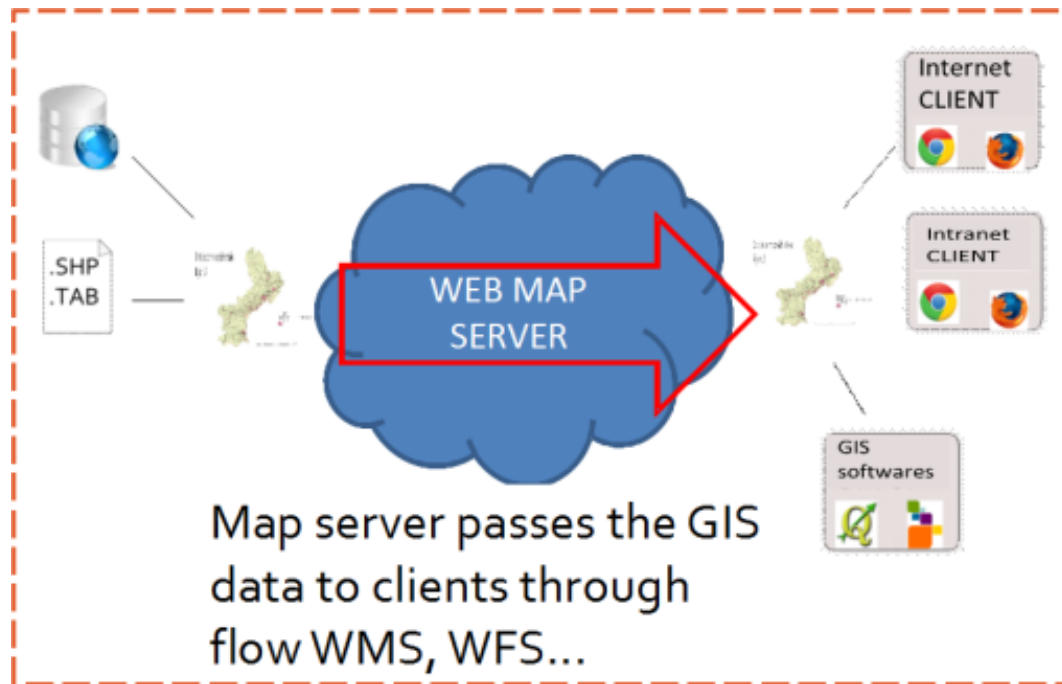
- Créer un compte sur <https://qgiscloud.com> ;
- Installer le plugin dans QGIS ;
- Définir une base de données en ligne ;
- Charger vos données ;
- Créer une carte et publier la carte ;

QGIS private Cloud virtuel Server (pour l'Intranet)

\* \*  
\*

## Les objectifs du webmapping





Les questions qu'il faut se poser :

- Intranet et Internet augmentent l'auditoire de façon considérable et de ce fait aussi la diversité des attentes et des cultures :
  - Quel est le public visé, quelles sont ses attentes ?
  - Quelles contraintes imposer au niveau du poste client ?
    - Serveur web carto et/ou plug'in navigateur...
  - Quelles contraintes imposer au serveur web ?
    - Intégrer la solution cartographique sur le serveur...
  - Quelle option technique retenir ?
    - Cartes statiques, interactions simples, cartographie dynamique, mises à jour en ligne, mobilité, ...

Le MAP Markup Language : l'avenir du webmapping ?

- Des balises pour gérer les cartes et leur affichage ;
- un projet complexe qui doit ajouter les fonctionnalités que l'on retrouve avec le javascript dans la syntaxe HTML !

"So, the big idea is this: MapML resolves the Web Platform gap by combining map and map data semantics into a hypermedia format that is syntactically and architecturally compatible with and derived from HTML."

Peter Rushforth, <https://www.w3.org/community/maps4html/2019/12/09/the-design-of-mapml/>

# Conclusion

Le webmapping est une formidable opportunité de diffuser l'information géographique au plus grand nombre.

L'utilisation des serveurs cartographiques permettent de mettre à disposition des flux de données SIG sous forme d'images ou d'objets vecteurs.

Les interfaces cartographiques permettent de visualiser ces données sur un navigateur web en utilisant des bibliothèques de fonctions écrites en Javascript.