

**PARTIE III : MISE EN ŒUVRE DU SYSTEME D'INFORMATION LOCALISEE
(SIL)**

Chapitre7 : **IMPLEMENTATION DU SIL**

7.1. **Architecture informatique du SIL :**

7.1.1. **La topologie du système :**

La topologie du système informatique de l'OAT comporte plusieurs postes de travail reliés par des réseaux locaux et le réseau Internet. A l'intérieur d'un site, les postes de travail sont reliés par un réseau local tandis qu'une liaison Internet relie les postes de travail éloignés géographiquement. On regroupe les utilisateurs du système informatique en trois catégories :

- les utilisateurs locaux : ce sont les utilisateurs qui se trouvent dans le même site que la base de données centrale et qui peuvent accéder à ce dernier via le réseau local. Ils ont la possibilité de :

- **Insérer des données**
- **Mettre à jour des données**
- **Consulter des données**

- les administrateurs locaux : ce sont les utilisateurs locaux qui ont le privilège d'administrer la base de données centrale. Ce sont eux qui peuvent intégrer les données envoyées par les partenaires éloignés dans la base de données centrale en contrôlant la validité et la conformité des données.

- les utilisateurs éloignés : ce sont les partenaires qui peuvent accéder à la base de données centrale via le réseau global (Internet) et ce en consultation ou pour la transmission des données. Ils ont le droit de :

- **Insérer des données**
- **Extraire des données**
- **Mettre à jour des données**

Nous trouvons ci-dessous une figure montrant cette topologie

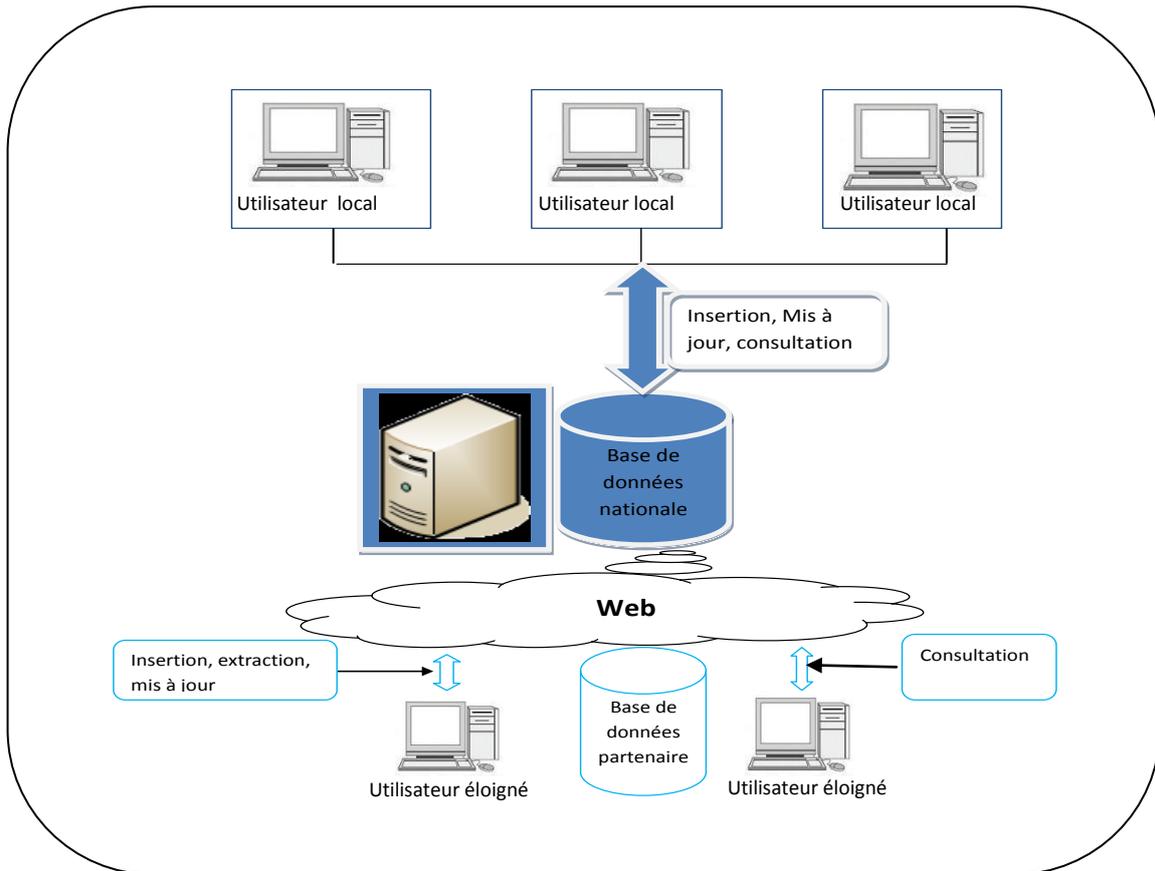


Figure III-1 : Topologie du système d'information de l'OAT

7.1.2. Présentation de l'architecture informatique du SIL :

L'architecture informatique du SIL comprend :

- une base de données alphanumérique et spatiale
- une interface autonome ;
- une interface en ligne ;

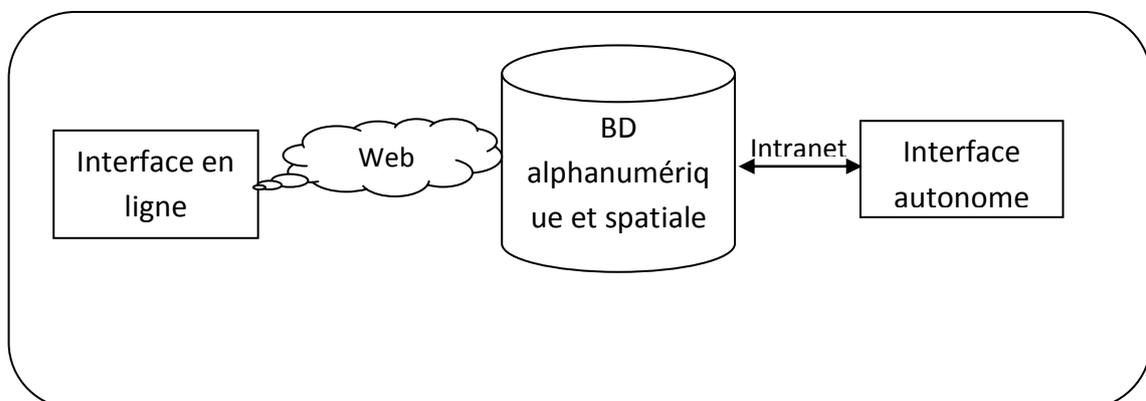


Figure III-2 : Architecture informatique du SIL

❖ la base de données alphanumérique et spatiale (base de données) :

Elle stocke toutes les données socio-économiques et démographiques ainsi que toutes les données qui ont une référence spatiale. C'est une composante du SIL gérée à l'aide d'un logiciel de SGBD avec une cartouche spatiale comme Oracle/Oracle spatial, PostgreSQL/PostGIS, MySQL/MyGIS, etc. La base de données nationale est localisée au sein de la cellule OAT.

❖ l'interface autonome :

C'est une application informatique développée avec un langage de développement et qui interagit avec la base de données nationale ou son extrait au niveau de chaque entité et au niveau des services déconcentrés. L'interface autonome permet de :

- saisir et mettre à jour les données sociales, démographiques et économiques
- administrer la base de données alphanumériques (contrôle de conformité, contrôle qualité avant intégration des données, etc.)

❖ l'interface en ligne :

C'est une application Web dynamique qui permet de consulter et mettre à jour la base de données alphanumérique pour les partenaires et les décideurs éloignés et disposant d'un accès à l'Intranet/Extranet de la cellule OAT. L'interface en ligne permet de réaliser les traitements suivants :

- saisir et mettre à jour en ligne les données socio-économiques ;
- transmettre les données via la messagerie électronique ou télécharger ces dernières à partir d'un autre serveur (auprès des partenaires) ;
- consulter en ligne les indicateurs du tableau de bord national (TBN).

7.1.3. Alternative d'implémentation du SIL :

Trois cas peuvent être adoptés comme architecture technique de l'OAT qui doit permettre l'évolution et la progression du système. Cette évolution concerne aussi bien l'application que la base de données. L'architecture proposée est conçue à pouvoir s'évoluer en fonction des technologies disponibles, de l'évolution des besoins, et de la montée en charge de l'application. Néanmoins, le choix de l'architecture doit être effectué de façon à faciliter la migration aussi bien de l'application que des données sans occasionner l'arrêt du système pour une longue durée.

a. Architecture client –serveur :

La première architecture probable pour l'OAT est une architecture client/serveur dont une base de données SIG sous Mapinfo ou ArcView couplée à un SGBD. Les données socio-économiques de l'OAT sont gérées via un SGBDR de type MS-Access. Une liaison ODBC sera établie entre l'application SIG et le SGBDR pour assurer leur intégration. L'application cliente sera constituée d'un Viewer (MapinfoProViewer ou Arcexplorer suivant qu'on adopte Mapinfo ou ArcView). Des modules applicatifs écrits en Mapbasic ou en Avenue seront greffés à l'application pour automatiser les tâches de consultation et d'édition afin qu'un utilisateur non initié en SIG puisse utiliser le système.

b. Architecture client-serveur avec un serveur de base de données géographique :

Nous pouvons envisager également l'achat d'un logiciel serveur de base de données géographique comme SpatialWare de Mapinfo ou la cartouche spatiale d'Oracle (Oracle spatial data cartridge). L'utilisation d'un serveur de données géographiques permet de s'affranchir des problèmes liés à l'intégrité et la sécurité des données (contrôle de l'accès). L'utilisation d'un serveur de données géographiques comme Oracle spatial permet de gérer au sein d'un même serveur de base de données les données géographiques et les données alphanumériques ce qui permettra de résoudre les problèmes liés à l'intégration des 2 systèmes (SIG et SGBD).

Les avantages de la généralisation d'une telle solution sont nombreux :

- L'administration des données géographiques serait réalisée par l'administrateur de base de données Oracle et ne nécessiteraient plus de compétence spécifique;
- Les données géographiques bénéficieraient de la qualité de service des bases Oracle (sauvegarde, reprise sur panne, gestion des transactions, etc.) ;
- Les données géographiques pourraient être utilisées par des applications non-géographiques;
- La gestion des données géographiques ne nécessiterait plus de serveurs dédiés;
- La solution Oracle est supportée par un grand nombre de fournisseurs de produits SIG, ce qui faciliterait le partage de données géographiques entre outils.

c. Architecture multi-niveaux (serveur de base de données géographiques, serveur d'application, serveur Web, client Web) :

Le dernier scénario pour l'architecture technique du SIL sera : l'architecture multi-niveaux ou multi-tiers. C'est l'architecture des systèmes d'information d'entreprise qui commence à substituer à l'architecture client/serveur classique.

L'architecture multi-tiers d'entreprise consiste en la dissociation de manière logique :

- du stockage des données (tiers données : serveur de base de données) ;
- de la logique applicative (tiers traitement : serveur d'application) ;
- et de la logique d'affichage (tiers présentation : serveur d'affichage).

Cette dissociation permet d'assurer une facilité de déploiement et de maintenance du système. Elle permet aussi de garantir une sécurité maximale de ce dernier. En effet, l'accès au serveur de base de données n'est possible que via le serveur d'application qui comporte tout un mécanisme de gestion de la sécurité et de la transaction.

De plus, l'architecture multi-tiers ne requiert une connexion permanente entre le serveur et les postes clients. Les traitements étant assurés par les serveurs locaux contrairement à l'architecture client-serveur classique, ceci permet de réduire considérablement le temps d'utilisation des lignes de télécommunication.

On peut envisager à terme la mise en place d'un mécanisme d'entrepôt de données (ou DATAWAREHOUSE) qui permet de réaliser une analyse multidimensionnelle (dimension temporelle, dimension spatiale) à partir de la base de données nationale. On produit ainsi des données agrégées et synthétisées qui sera facilement consultable en ligne via le réseau Internet et surtout pour les décideurs.

7.1.4. Architecture matériel et logiciel du SIL :

L'architecture multi-niveaux est la solution la plus adaptée pour mettre en œuvre une application Intranet/Extranet basée sur le protocole http. Elle est la mieux adaptée aux besoins et exigences (sécurité de l'application, intégrité des données, contrôle d'accès, etc.) du SIL. L'application comportera trois niveaux :

- serveur de base de données : logiciel de système de gestion de base de données (SGBD) pouvant gérer l'accès simultané de plusieurs utilisateurs ;
- serveur http : Apache Web Server ou Microsoft IIS ou autres ;

- client HTML : logiciel de navigation Internet comme Netscape Navigator, Microsoft Internet Explorer, Mozilla Firefox, etc.

Pour assurer l'intégration de la base de données centrale avec la base de données pour SIG, il faut que le logiciel de SGBD et le logiciel de SIG puisse se communiquer via une liaison de type ODBC.

Nous proposons ci-après une esquisse de configuration logicielle permettant de mettre en œuvre une architecture multi-niveau basée sur le standard J2EE pour le SIL :

- serveur de base de données : PostgreSQL avec une extension spatiale PostGIS pour contenir les données alphanumériques et géographiques ;

- serveur http : Apache Web Server ;

- Serveur cartographique : Mapserver

- Système d'exploitation pour la machine serveur : Microsoft Windows XP ;

- langage de développement des applications Web dynamiques (Interface en ligne) : PHP, PhpMapscript, JavaScript

- logiciel de SIG Client : QGIS pour le client lourd et Pmapper pour le client léger

Cette configuration est basée sur des logiciels Microsoft ce qui permet d'assurer leur compatibilité et son déploiement.

7.2. Mise en œuvre de la solution adoptée :

7.2.1. Description de l'architecture multi-tiers :

L'architecture multi-tiers plus précisément à trois tiers ou à trois niveaux que nous allons adopter est très bien adaptée pour notre application intranet/extranet.

Il s'agit d'un modèle logique d'architecture applicative qui vise à modéliser une application comme un empilement de trois couches logicielles dont le rôle est clairement défini par :

- La présentation des données : correspondant à l'affichage, la restitution sur le poste de travail, le dialogue avec l'utilisateur ;
- Le traitement métier des données : correspondant à la mise en œuvre de l'ensemble des règles de gestion et de la logique applicative ;
- Et enfin l'accès aux données persistantes: correspondant aux données qui sont destinées à être conservées sur la durée, voire de manière définitive.

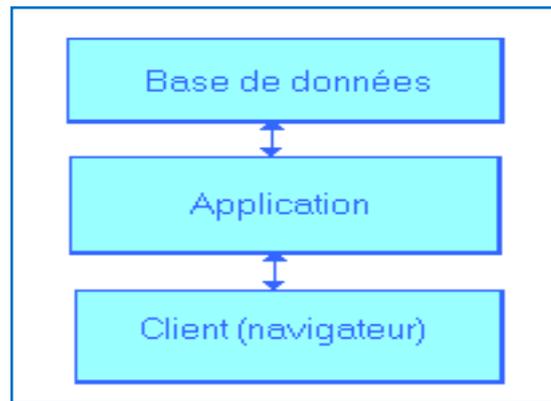


Figure III-3 : Architecture à trois niveaux

L'utilisation de l'architecture 3-tiers va nous permettre :

- Allègement du poste de travail client
- Prise en compte de l'hétérogénéité des plates-formes (serveurs, clients, langages, etc.) ;
- Introduction de clients « légers » (plus liée aux technologies Intranet/HTML) ;
- Amélioration de la sécurité des données, en supprimant le lien entre le client et les données. Le serveur a pour tâche, en plus des traitements purement métiers, de vérifier l'intégrité et la validité des données avant de les envoyer dans la couche de données.
- Rupture du lien de propriété exclusive entre application et données. Dans ce modèle, la base de données peut être plus facilement normalisée et intégrée à un entrepôt de données.
- Et enfin, meilleure répartition de la charge entre différents serveurs d'application.

7.2.2. Fonctionnement de l'architecture 3-tiers :

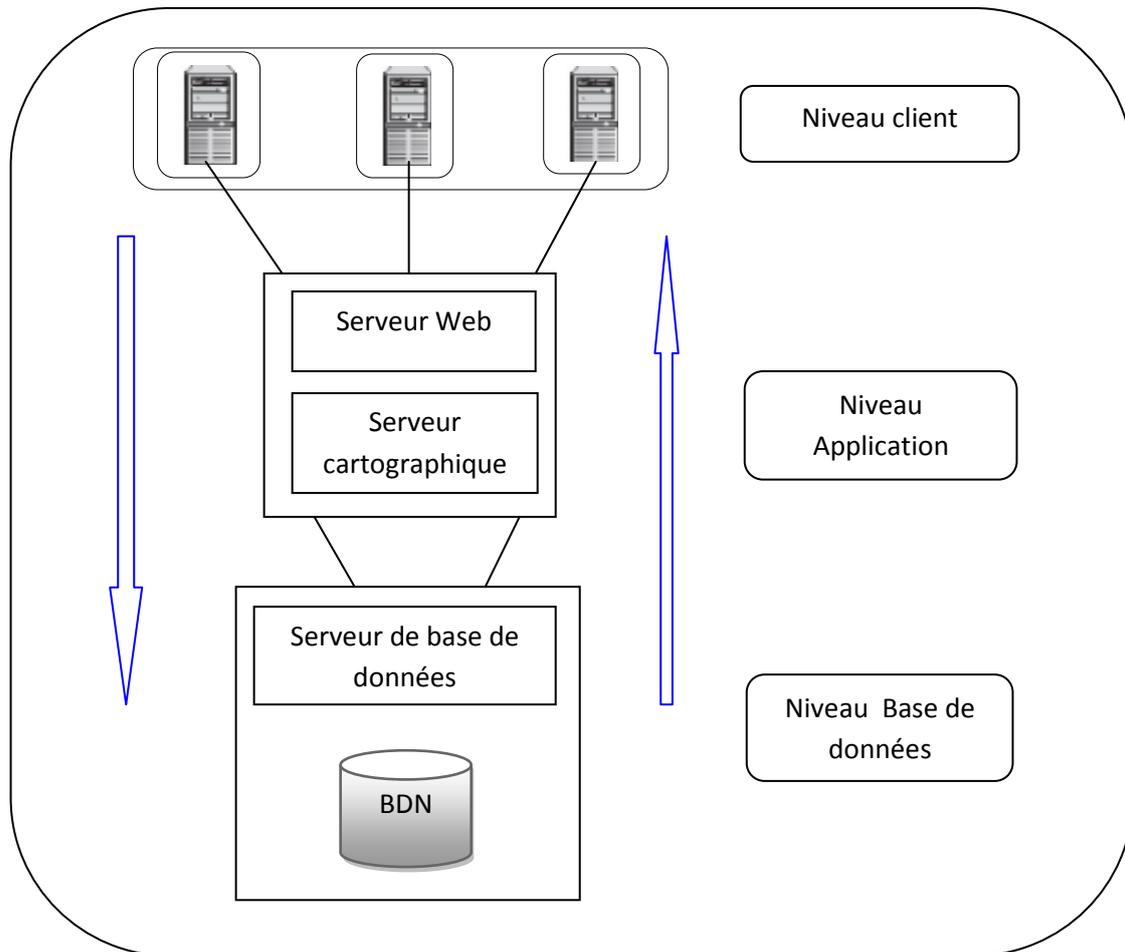


Figure III-4 : Fonctionnement de l'architecture 3-tiers

a. Niveau client :

Le client est la machine et/ou par extension le logiciel qui peut se connecter au serveur, recevoir et exploiter les données diffusées par le serveur.

Les clients peuvent être regroupés en deux types :

- Client léger
- Client lourd

L'utilisateur effectue des requêtes à travers son logiciel client pour afficher des cartes spécifiques ou recevoir des informations géographiques.

a.1. Client léger :

Le terme **client léger** (aussi appelé par un anglicisme *thin*) désigne parfois des éléments matériels et parfois des éléments logiciels.

➤ Au sens matériel, un client léger est un ordinateur qui, dans une architecture client-serveur, n'a presque pas de logique d'application. Il dépend donc surtout du serveur central pour le traitement.

➤ Au sens logiciel, L'usage veut qu'une application en client léger n'impose à l'utilisateur que d'avoir un navigateur Web, même s'il doit être équipé de plugins ou de machines virtuelles.

L'utilisation des clients légers va donc permettre aux utilisateurs de ne rien installer sur son machine qu'un navigateur web pour accéder à l'information géographique du serveur de base de données. Voici quelques catalogues de clients légers.

Tableau 4 : Catalogue des logiciels client-léger

Logiciels clients légers	Descriptions
P-mapper http://sourceforge.net/projects/pmapper/files/	C'est un client léger qui respecte les normes WMS de l'OGC. Il peut se connecter à tous les serveurs cartographiques respectant ces normes
Cartoweb http://www.cartoweb.org/	C'est une des solutions les plus prometteuses. Ce SIG par internet se connecte à Mapserver et comporte de nombreuses fonctions de navigation et de gestion des couches. Il dispose aussi d'un module d'impression des cartes au format pdf et peut interagir avec un serveur PostGIS pour mettre à jour ou créer à distance des objets géographiques. Son architecture est relativement complexe pour les néophytes mais permet des infrastructures évoluées.
Ka-Map http://ka-map.maptools.org/	Toujours destiné à se connecter à Mapserver, c'est une solution simple mais complète en termes de navigation et de gestion des couches. Elle est légère et relativement aisée en termes d'installation. Elle utilise un mode de décalage comparable à celui de GoogleMap.
Intermap http://sourceforge.net/projects/intermap/	C'est un client générique qui respecte les normes WMS de l'OGC. Il peut donc se connecter à tous les types de serveur cartographique respectant ces normes. c'est le client cartographique utilisé par

	GeoNetwork, un outil de catalogage de géodonnées respectant la norme ISO19115.
OpenLayers http://www.openlayers.org/	C'est une librairie écrite en Javascript AJAX permettant de créer des sites cartographiques et pouvant se connecter à tout serveur respectant les normes WMS et WFS. Par ailleurs, OpenLayers permet de se connecter à plusieurs serveurs commerciaux tels que MSN, Yahoo, etc.

Notre choix s'incline sur la **P-mapper version 4.2.0** à cause de ses fonctionnalités riches en fonctions SIG. En outre, il permet la connexion à PostGIS pour récupérer les données de base. Sa mise en œuvre est un peu difficile mais abordable.

a.2. Client lourd :

Les « clients lourds » sont des SIG locaux classiques. Ils comportent toutes les fonctions qu'un SIG se doit d'avoir de nos jours :

- Affichage de fichiers vecteurs et rasters ;
- Outils complets de navigation;
- Création, ajout, modification et suppression d'entités géographiques ;
- Calculs géométriques (distances, surface, etc.);
- Analyses thématiques (classification) et analyses spatiales (intersection, inclusion).

Cependant, ils se distinguent de leurs congénères par des fonctions de connexion à des serveurs cartographiques ainsi qu'à des bases de données spatiales.

Tableau 5 : Catalogue des logiciels clients-lourds

Logiciels clients lourds	Descriptions
<p>Open Jump http://openjump.org/</p>	<p>Open Jump est un SIG libre issu de jump-Project, une fondation canadienne faisant collaborer des chercheurs et développeurs pour produire des outils SIG libres et orientés métiers.</p> <ul style="list-style-type: none"> - Connexion WMS et WFS - Types supportés : Shape, GML, et OGR génériques - Possibilité de connexion à PostGIS - Relativement complet en analyse spatiale - Plugins raster, mise en page, etc.
<p>Quantum GIS http://qgis.org/</p>	<p>C'est un client lourd très prometteur. Autrefois relativement léger, il s'est enrichi du moteur GRASS ainsi que de ses fonctionnalités les plus utiles :</p> <ul style="list-style-type: none"> - Connexion WMS - Connexion PostGIS : modification attributaire, géométrique - Nombre abondants de types supportés : shp, mif/mid, tab - Moins complet que GRASS en termes d'analyse spatiale
<p>Udig http://udig.refractions.net/</p>	<p>C'est un SIG local totalement orienté pour l'accès aux données par internet.</p> <ul style="list-style-type: none"> - Très complet en termes d'accès aux données par internet - Connexion PostGIS : modification attributaire, géométrique - Pas d'accès à des fichiers SIG classiques
<p>Grass http://grass.itc.it/</p>	<p>C'est la référence dans le domaine du SIG Open Source. Il permet des analyses extrêmement poussées aussi bien sur des fichiers vecteurs que rasters. De plus, il respecte les normes OGC et du fait de l'implémentation en son sein des bibliothèques GDAL et OGR, il peut lire un nombre de formats de fichier énorme. Enfin, il dispose d'un moteur 3D ainsi que la simulation de modèle. Concernant l'accès aux données par internet il est assez limité mais permet néanmoins la connexion vers des bases</p>

	<p>PostGIS</p> <ul style="list-style-type: none"> - Nombreux types supportés - Connexion PostGIS (import/export) - Complet : analyse vecteur et raster - Assez difficile à utiliser et installation fastidieuse sous windows
--	--

Notre choix s'incline vers le QuantumGIS vu ses caractéristiques, on l'a pris comme une solution fiable pour réaliser ce projet. Malgré son état par rapport à GRASS en termes d'analyse spatiale, le logiciel QGIS est doté d'une interface conviviale et intuitive qui permet d'attaquer facilement la base de données externe Post GIS. Il peut supporter également de nombreux types de format comme shp, mif/mid et tab. Il constitue alors un des SIG dits « Open Source » capable de gérer les bases de données à distance. Sur ce fait, nous allons utiliser le Quantum GIS 1.7.4.

b. Niveau traitement :

Ce niveau assure la logique applicative, il est composé par le serveur web et le serveur d'application.

- **Le serveur web** adopté est l' « Apache http server » qui est directement installé avec le package « ms4w », ce serveur a pour fonction de :

- Ecouter les requêtes des clients et cacher le contenu,
- Equilibrer les charges sur le serveur d'application,
- Renvoyer les pages XML et HTML générées aux clients.

- **Le serveur d'application ou le serveur cartographique** adopté est le « Mapserver » qui sera installé également avec le package « ms4w ». Le serveur assurera :

- La gestion de la logique du travail ;
- Le formatage des documents générés ;
- Le développement des pages web, forms, report ;
- La gestion de la sécurité.

c. Niveau Base de données :

Ce niveau est constitué par le serveur de base de données. C'est un composant logiciel permettant le stockage, le traitement des données alphanumériques et spatiales. En outre, en tant que serveur, il permet ainsi de partager ces informations vers d'autres logiciels. Ces

informations seront protégées par un login et un mot de passe qui sera très indispensable dans un projet webmapping.

Nous utilisons comme serveur de base de données le couple : PostgreSQL/PostGIS.

7.2.3. Etablissement de la connexion sur tous les niveaux :

a. Installation des logiciels nécessaires :

La réalisation de cette connexion nécessite la mise à disposition des logiciels que nous venons de choisir et de les installer : Mapserver, Apache http server, P-mapper, QuantumGIS, PostgreSQL/PostGIS. Tous ces logiciels sont libres (OpenSource), il suffit alors de trouver leur site officiel et de procéder aux téléchargements (cf. annexe).

Le serveur de base de données PostgreSQL/PostGIS est déjà installé précédemment.

L'installation du QGIS 1.7.4 ne pose pas de problème après le téléchargement sur son site officiel.

a.1. Installation du serveur cartographique et du serveur web :

Mapserver, apache http server seront installés après le téléchargement de la package « ms4w ».

L'installation de Mapserver grâce au package « ms4w » ne pose aucun problème. Dans cette partie, nous allons expliquer comment se fait le téléchargement du package, l'installation des logiciels packagés et le premier lancement du mapserver avec un navigateur internet.

Etape1 : Téléchargement

En visitant le site officiel de « Maptools », nous sommes en face de même genre d'interface que celle du dessous.



Figure III-5 : Page d'accueil de Maptools

Descendons en bas de la page et cliquons sur le lien « MS4W downloads page ».



Figure III-6 : Page de téléchargement du package ms4w

Cliquons sur la version la plus récente du package pour démarrer le téléchargement.

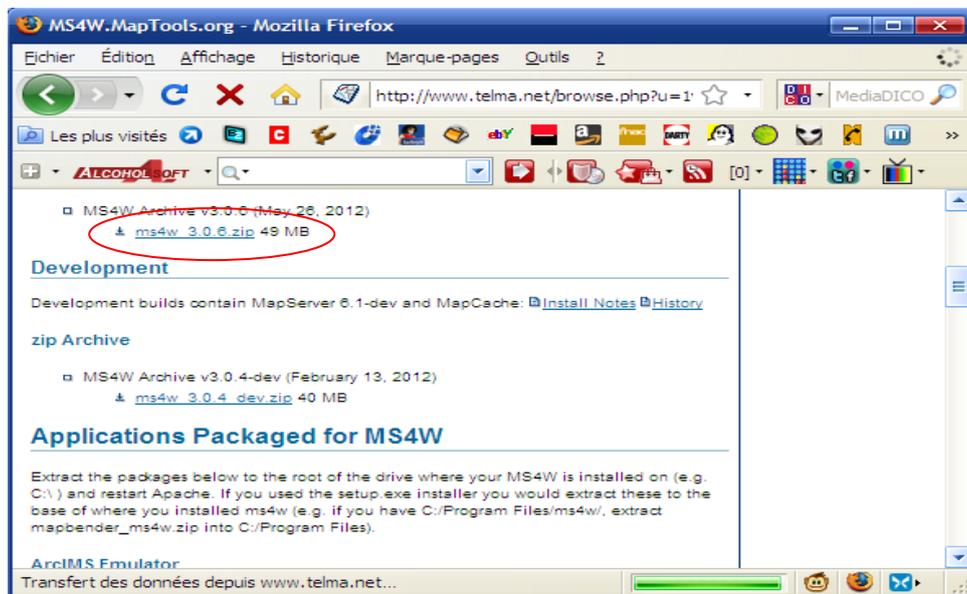


Figure III-7 : Choix de la version à télécharger

Cliquons sur « Enregistrer ».

Choisissons le répertoire où va stocker le fichier puis nous cliquons sur « Enregistrer ».

Etape 2 : Installation

Maintenant qu'on a récupéré tous les fichiers nécessaires, il suffit de les décompresser l'un après l'autre dans la racine de notre disque dur. Ici, nous avons choisi la racine C:\.

Pour installer le serveur apache, nous devons joindre le répertoire C:\ms4w\ et double cliquer sur « apache-instal.bat »

A titre de vérification, nous devons lancer un navigateur de notre choix en indiquant l'adresse suivante : <http://localhost/> ou <http://127.0.0.1/>. La capture d'écran suivante sera le résultat :

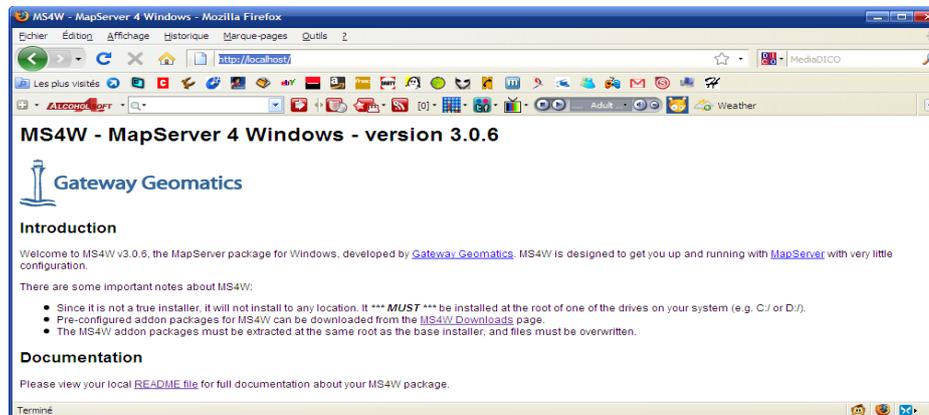


Figure III-8 : Page de vérification de l'installation du serveur web (Apache)

Ce résultat nous confirme que Apache et Mapserver a été bien installé.

a.2. Installation du client léger : P-mapper

L'installation de P-mapper ne pose vraiment aucun problème, lorsque nous obtenons le fichier *.zip



Il suffit de le décompresser dans le répertoire : C:\ms4w\ que nous venons d'installer Apache et Mapserver.

La figure ci-dessous se présente en bas de la page, lorsqu'on lance notre navigateur avec la même adresse précédente. Ceci montre que P-mapper est bien installé.

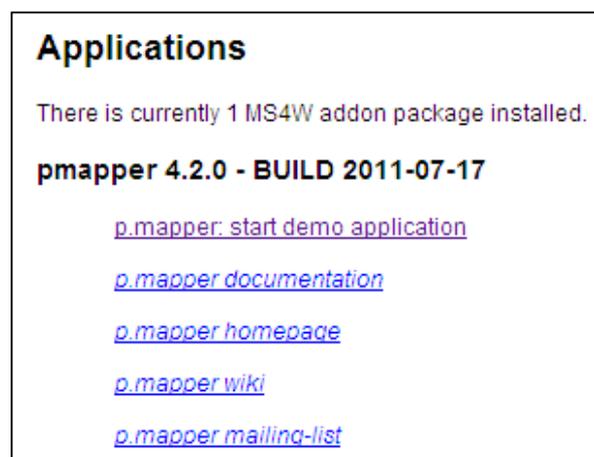
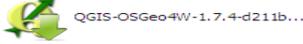


Figure III-9 : Page de vérification de l'installation du client léger P-mapper

a.3. Installation du client lourd : Quantum GIS

Son installation est simple, après avoir téléchargé la version 1.7.4 que nous avons choisi ; on double clic sur le fichier exécutable 

Que nous avons stocké dans un répertoire de notre disque dur.

b. Etablissement de la connexion serveur web et serveur cartographique :

La connexion entre Mapserver et apache est déjà établie lorsqu' on a lancé « *apache-install.bat* » et confirmé par le résultat de l'URL <http://localhost/>.

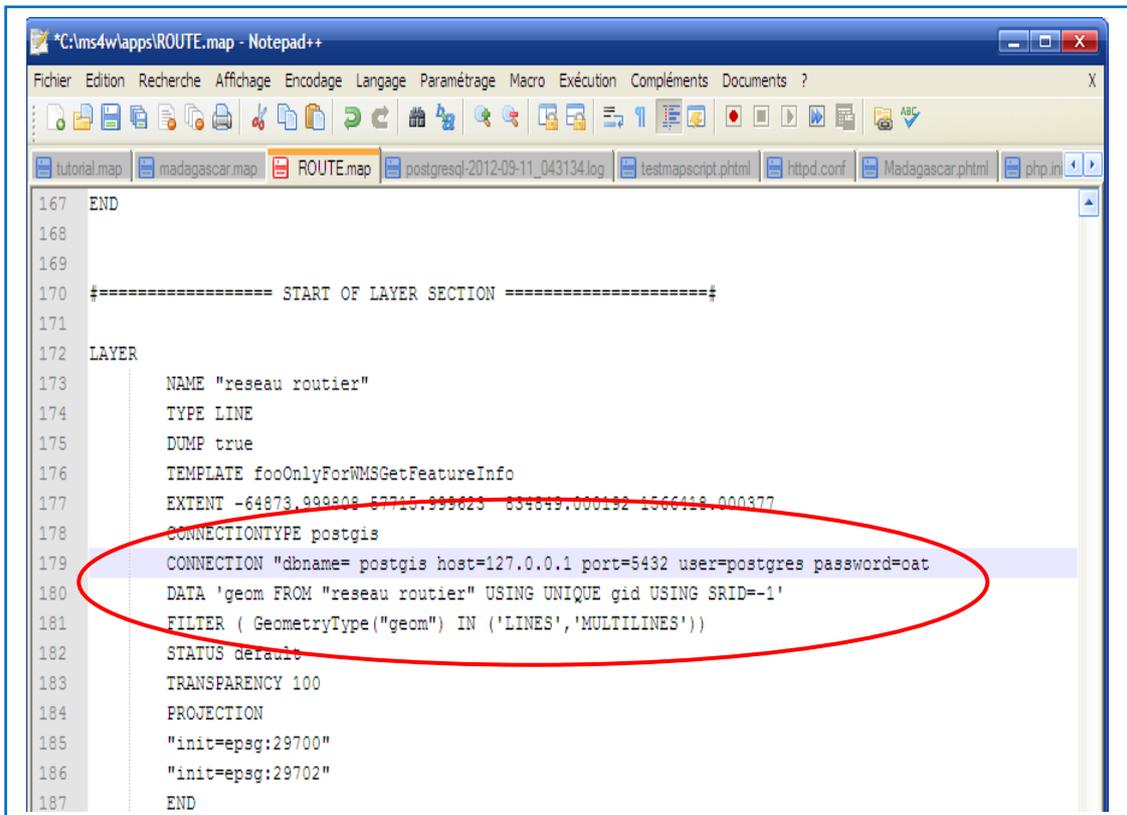
c. Etablissement de la connexion Mapserver-PostGIS :

La connexion entre le serveur cartographique et le serveur de base de données est établie dans le bloc « **mapfile** »

Mapfile :

C'est un fichier texte qui va contenir tous les paramètres nécessaires à Mapserver pour la génération d'un document cartographique statique ou dynamique.

La description du mapfile sera détaillée à l'annexe2.



```

167 END
168
169
170 #===== START OF LAYER SECTION =====#
171
172 LAYER
173     NAME "reseau routier"
174     TYPE LINE
175     DUMP true
176     TEMPLATE fooOnlyForWMSGetFeatureInfo
177     EXTENT -64873.999808 57715.999823 834849.000192 1566418.000377
178     CONNECTIONTYPE postgis
179     CONNECTION "dbname= postgis host=127.0.0.1 port=5432 user=postgres password=oat"
180     DATA 'geom FROM "reseau routier" USING UNIQUE gid USING SRID=-1'
181     FILTER ( GeometryType("geom") IN ('LINES','MULTILINE'))
182     STATUS default
183     TRANSPARENCY 100
184     PROJECTION
185     "init=epsg:29700"
186     "init=epsg:29702"
187     END

```

Figure III-10 : Exemple d'un fichier Mapfile

d. Etablissement de la connexion QGIS-Post GIS :

Pour cela, lancer Quantum GIS 1.7.4 en double-cliquant sur son raccourci ou en allant dans le menu « Démarrer » \ tous les programmes.

Pour réaliser la connexion entre le client lourd et le serveur de base de données, nous devons créer une connexion enregistrée en cliquant sur le bouton « Ajouter une couche PostGIS ». 

de la barre d'outils de l'interface principale :

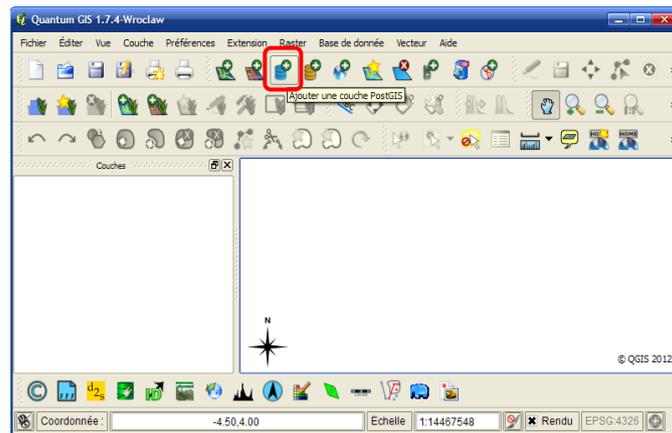


Figure III-11 : Fenêtre principale du client lourd QGIS

La fenêtre « **Ajouter une ou plusieurs tables Post GIS** » apparaît et nous devons cliquer sur « **Nouveau** » pour accéder au gestionnaire de connexion

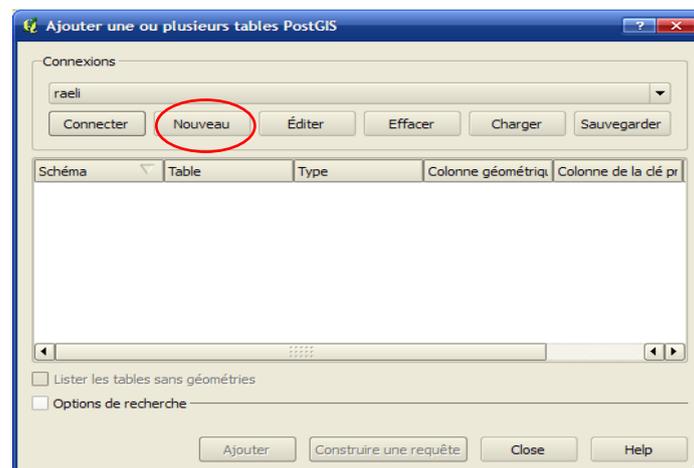


Figure III-12 : Ajout d'une table PostGIS

Lorsque la fenêtre « Créer une nouvelle connexion PostGIS » apparaîtra, nous procédons au remplissage des paramètres de connexion à PostGIS.

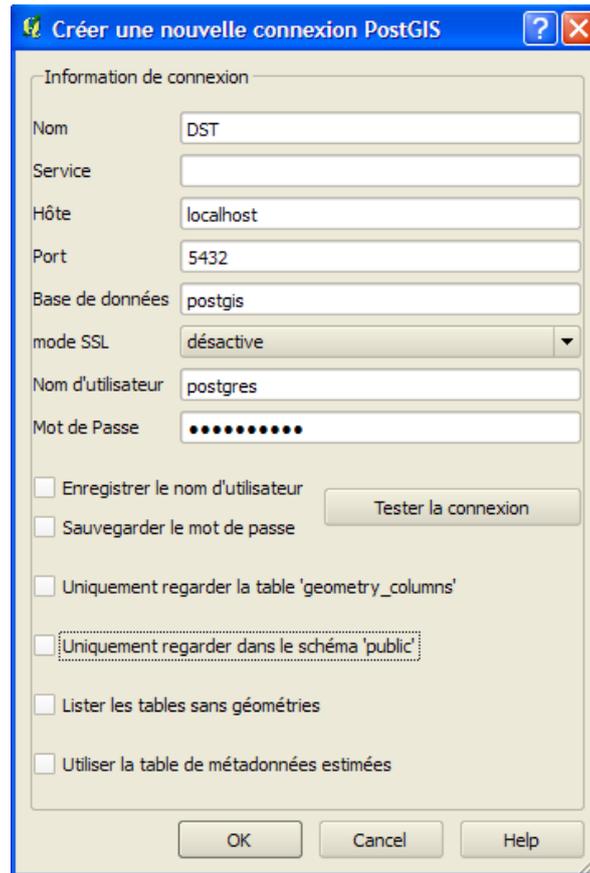


Figure III-13 : Création d'une connexion PostGIS

Le tableau ci-dessous explique la manière dont nous procédons au remplissage :

Tableau 6 : Paramètre de connexion au serveur de BDD PostGIS

Paramètres de connexion PostGIS	
Paramètre	Description
Nom	Un nom pour cette connexion. Il peut être identique à Base de données.
Hôte	Nom pour l'hôte de la base de données. Il doit s'agir d'un nom existant, car il sera utilisé pour ouvrir une connexion Telnet ou interroger l'hôte. Si la base de données est sur le même ordinateur que QGIS, mettons simplement local host.
Base de données	Nom de la base de données.
Port	Numéro de port que le serveur de base de données PostgreSQL écoute. Le port par défaut est 5432.
Nom d'utilisateur	Nom d'utilisateur utilisé pour se connecter à la base de données.

Mot de passe	Mot de passe utilisé avec le Nom d'utilisateur pour se connecter à la base de données.
Mode SSL	<p>Comment sera négociée la connexion SSL avec le serveur. Voici-les options :</p> <ul style="list-style-type: none"> – désactiver : essayer une connexion SSL non cryptée uniquement – permettre : essayer une connexion non-SSL. Si cela échoue, essayer une connexion SSL ; – préférer (par défaut) : essayer une connexion SSL. Si cela échoue une connexion non-SSL ; – requiert : essayer seulement une connexion SSL <p>Il faut noter qu'une accélération massive du rendu des couches Post GIS peut être obtenue en désactivant le SSL dans l'éditeur de connexion.</p>

Une fois que tous les paramètres et les options sont définis, nous pouvons tester la connexion en cliquant sur le bouton « **Tester la connexion** » afin d'assurer l'établissement de la connexion.

7.2.4. Génération des cartes statiques ou dynamiques sur internet :

Pour générer une carte sur internet, nous avons deux possibilités : l'utilisation du mode CGI (mapserve.exe) de Mapserver et la fonction dl () de PHPMapscript.

Quoi qu'il en soit la création d'un fichier mapfile est nécessaire.

a. Description de la structure de mapfile :

a.1. Définition :

Le « mapfile » est le cœur de MapServer. Il définit les relations entre les objets, indique à MapServer où se trouvent les données et définit la façon dont elles seront dessinées. Ce sont des fichiers textes qui vont contenir tous les paramètres nécessaires à Mapserver pour la génération d'un document cartographique, statique ou dynamique. Ils sont organisés en blocs ou objets hiérarchiques, imbriqués, notés NOM_DU_BLOC...END. Le bloc racine est le bloc MAP.

Ainsi, tout mapfile est donc compris entre balise MAP et la balise END.

a.2. Caractéristiques :

- Les mapfiles sont insensibles à la casse (minuscules/majuscules), sauf pour les noms des champs attributaires, notés entre [crochets], et naturellement les valeurs de type chaîne, notées entre guillemets ou apostrophes ;
- Un mapfile contient au maximum 50 couches mais on peut modifier cette valeur dans le code source et recompiler mapserver ;
- Les chemins peuvent être indiqués de manière absolue (à partir de la racine du système), ou relativement à l'emplacement du mapfile. Les chemins vers les données sont relatifs au paramètre SHAPEPATH et en l'absence de ce dernier les données seront recherchées dans le même répertoire que le mapfile.
- Les commentaires sont indiqués par le caractère #
- Les attributs sont désignés en utilisant la syntaxe suivante : [ATTRIBUTENAME]... Notons que le nom de l'attribut entre les crochets *EST* sensible à la casse. Généralement les fichiers shape ESRI ont leurs propres attributs (noms de colonnes .dbf) en majuscules, et pour Post GIS, les minuscules sont *TOUJOURS* utilisées.

a.3. Descriptions :

Le « mapfile » est constitué par l'assemblage de plusieurs blocs : le bloc MAP, le bloc WEB, le bloc REFERENCE, le bloc LEGENDE, le bloc PROJECTION, le bloc SCALEBAR, le bloc OUTPUTFORMAT et le bloc LAYER.

La description détaillée du mapfile se trouve à l'annexe2.

b. Affichage d'une carte statique par Mapserver CGI 6.0.3 :

Dans ce cas, nous allons utiliser la couche « **region.shp** » de la BD500 de la FTM que nous servirons également comme fond géométrique de notre Base de données (BDN). Lorsque notre mapfile a été bien articulé et configuré, nous pouvons afficher notre carte sur le web. Voici une partie du mapfile permettant la génération de la carte.

```

LAYER
  NAME "REGION"
  TYPE POLYGON
  DUMP true
  TEMPLATE fooOnlyForWMSGetFeatureInfo
  EXTENT 64873.999808 57715.999623 834849.000192 1566418.000377
  CONNECTIONTYPE postgis
  CONNECTION "dbname= 'postgis' host=127.0.0.1 port=5432 user=postgres password=cat"
  DATA 'geom FROM "REGION" USING UNIQUE Id_region USING srid=-1'
  FILTER ( GeometryType("geom") IN ('POLYGON','MULTIPOLYGON'))
  METADATA
    'ows_title' 'REGION'
    "description" "REGION"
    "result_fields" "Id_region,nom_region,nb_fiv,pop93,mca,surface,province,geom"
    "result_headers" "Id_region,nom_region,nb_fiv,pop93,mca,surface,province,geom"
    "legendicon" "images/legend/REGION.png"
  END # metadata
  STATUS default
  TRANSPARENCY 100
  PROJECTION
    "init=epsg:29700"
    "init=epsg:29702"
  END
  CLASS
    Name 'REGION'
    STYLE
    SYMBOL 0
    SIZE 10.5
    OUTLINECOLOR 170 85 255
    ##COLOR 170 85 255
    END
  END # CLASS
END # LAYER

END #Map

```

Figure III-14 : le mapfile de génération de la carte des 22 régions

Nous avons alors besoin de deux machines, le serveur et le client mais pour ce travail de mémoire, nous allons simuler nos résultats sur une seule machine. La machine se comportera donc en tant que client et serveur : l'adresse de la machine serveur sera donc <http://localhost/> ou <http://127.0.0.1/>.

Pour cela, nous lançons notre navigateur et de taper l'adresse : <http://localhost/cgi-bin/mapserv.exe?map=C:/ms4w/apps/region.map&mode=map>.

La figure ci-dessous sera le résultat :

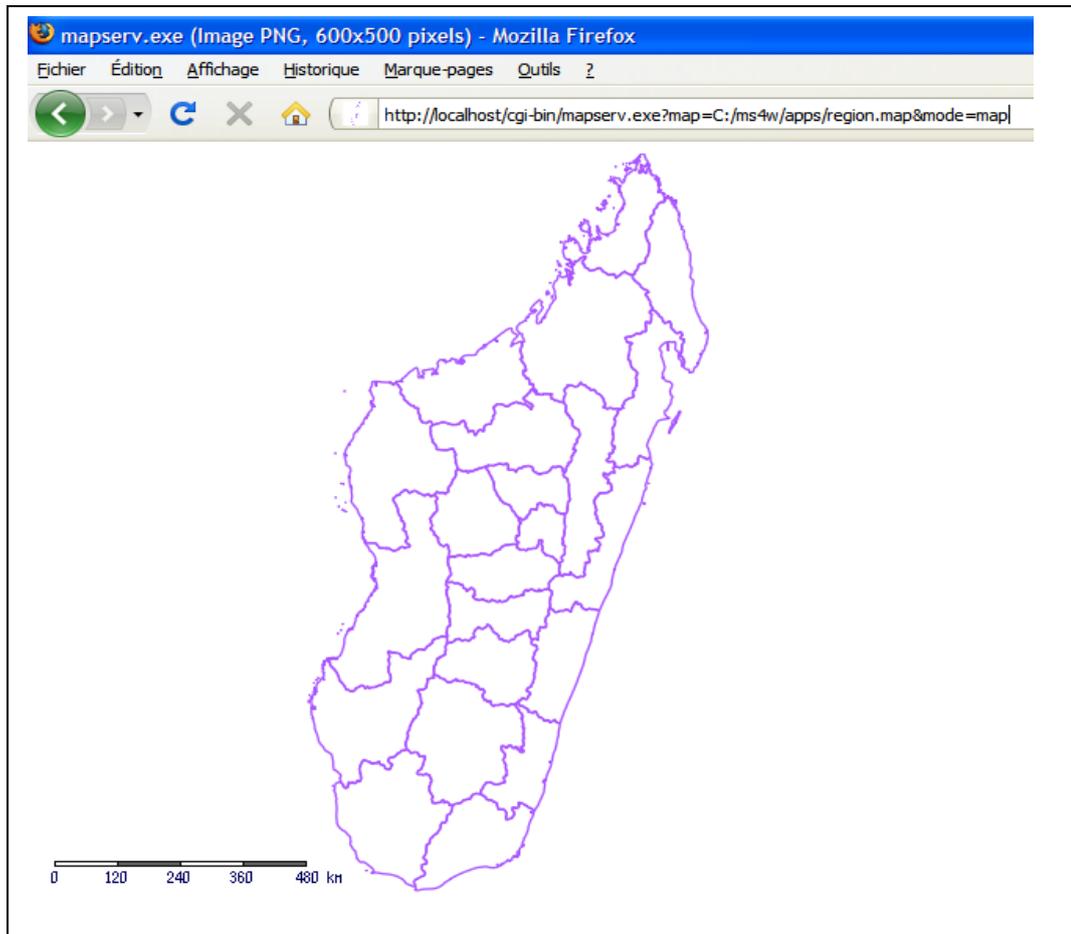


Figure III-15 : Carte statique générée par mapserver CGI

c. Génération d'une carte dynamique sur le web :

Pour cela, nous allons utiliser le langage PHP/Mapscript et le client léger Pmapper 4.2. A part la configuration du fichier mapfile, nous devons modifier certaines configurations par défaut du logiciel client Pmapper. En effet, quand nous avons notre mapfile (region.map), dans pmapper nous allons dans le dossier pmapper/config ensuite nous créons un nouveau dossier (avec le nom que nous avons choisi: « region »).

A l'intérieur de ce dossier nous plaçons notre mapfile ainsi que les trois autres fichiers qu'il y a déjà dans le dossier situé: pmapper/config/default (js_config.php, custom.php_notused, custom).

Ensuite, dans pmapper/config nous créons un nouveau fichier appelé config_region.xml (nous allons copier et coller le config_default.xml) et nous modifions à l'intérieur le paramètre amenant à notre dossier carte.

```
</pmapper>
<config>
  <pm_config_location>region</pm_config_location>
  <pm_javascript_location>javascript</pm_javascript_location>
  <pm_print_configfile>common/print1.xml</pm_print_configfile>
  <pm_search_configfile>inline</pm_search_configfile>
</config>
<map>
  <mapFile>region.map</mapFile>
  <tplMapFile>common/template.map</tplMapFile>
  <categories>
    <category name="Madagascar">
      <group>REGION</group>
    </category>
  </categories>
</map>
```

Figure III-16 : Modification du code de configuration de Pmapper

Quand nous avons fait tout cela, nous allons dans un navigateur web (MozillaFirefox), nous tapons l'adresse URL qui nous amène à la visualisation du fichier de démo par défaut dans Pmapper (http://localhost/pmapper/map_default.phtml) et nous rajoutons à la fin de cette URL: ? Config=region.