

## Chapitre 10 : Conception de la base de données du Repérage Numérique (RN)

Cette partie est consacrée aux étapes fondamentales pour le développement de la base de données du Repérage Numérique à Madagascar. Pour la conception et la réalisation de cette application, on a choisi de modéliser la base de données avec MERISE.

### 10.1 Définition

#### **Base de données :**

Une base de données c'est un ensemble structuré des données enregistré avec le minimum de redondance sur des supports accessibles par l'ordinateur pour satisfaire simultanément plusieurs utilisateurs de façon sélective et en un temps opportun.

#### **Méthode MERISE :**

Merise est une méthode d'analyse, de conception et de gestion de projet intégrée, ce qui en constitue le principal atout.

### 10.2 Spécification des besoins

C'est une étape primordiale au début de chaque démarche de développement. Son but est de veiller à développer un logiciel adéquat, sa finalité est la description générale des fonctionnalités du système, en répondant à la question : Quelles sont les fonctions du système?

Le système de l'Application RN doit répondre aux exigences suivantes :

- Le système doit pouvoir récupérer des informations de chaque entité pour mettre à jour la base des données de l'application.
- L'insertion des valeurs dans les entités.
- Modification des informations contenues dans les entités.
- La suppression des informations contenues dans les entités.
- L'impression des plans.

### 10.3 Méthode d'analyse et de conception

En appliquant cette méthode, nous allons mettre trois niveaux de réflexion que ces soient au niveau des données que des traitements :

- ⊗ **Le niveau conceptuel** qui correspond à la définition des finalités de l'Application en expliquant sa raison d'être. Ce niveau traduit les objectifs et les contraintes qui pèsent sur l'Application. L'informatique doit les intégrer sans remise en cause. Ces finalités constituent généralement le niveau le plus stable.
- ⊗ **Le niveau organisationnel** permet de définir l'organisation qu'il est souhaitable de mettre en place dans le Service Topographique pour atteindre les objectifs visés. On parle alors de choix d'organisation, pour lesquels la marge de manœuvre est plus importante. Ce niveau précise les postes de travail, la chronologie des opérations, les choix d'automatisation, tout en intégrant les contraintes éventuelles.
- ⊗ **Le niveau technique** intègre les moyens techniques nécessaires au projet. Ils s'expriment en termes de matériels ou de logiciels et sont (par suite des progrès technologiques) les plus sujets à changement.

### 10.4 La mise en œuvre de la méthode choisie :

Pour mettre en œuvre cette méthode, on va parcourir les différents niveaux suivants :

- ➔ La création du dictionnaire des données.
- ➔ L'élaboration du Modèle Conceptuel de Données (MCD)
- ➔ Le Modèle Logique de Données (MLD)
- ➔ Le Modèle Physique de Données (MPD)

#### 10.4.1 Création du dictionnaire de données:

Avant toute modélisation nous avons procédé à des entretiens pour le recueil de l'information auprès des utilisateurs du système d'informations œuvrant sur le domaine foncier et l'intégration des nouvelles demandes. Il faut regarder alors l'ensemble des données utiles en fonction de l'attente des Services Topographique. Cet ensemble forme le dictionnaire des données.

Dans le Repérage Numérique (RN) on a utilisé une base de données géographiques ou spatiales (découpage administratif : limite régionale, limite district et limite communale, limite des terrains à divers statut et orthophoto numérique)

Ce dictionnaire est sous forme de tableau :

Champs retenus	Type des données	Commentaire
Id_region	Caractère(5)	Code géographique de la région
Nom_region	Caractère(50)	Toponymie de la région
Id_district	Caractère(5)	Code géographique du District
Nom_district	Caractère(50)	Toponymie du district
Id_commune	Caractère(5)	Code géographique de la Commune
Nom_commune	Caractère(50)	Toponymie de la commune
Id_fokontany	Caractère(5)	Code géographique du Fokontany
Nom_fokontany	Caractère(50)	Toponymie du fokontany
Id_localite	Caractère(5)	Code géographique de la Localité
Nom_localite	Caractère(50)	Toponymie de la localité
Id_canton	Caractère(5)	Identité du canton appartenant le cadastre
Nom_canton	Caractère(50)	Nom du canton appartenant le cadastre
Id_section	Caractère(5)	Identité de la section contenant le cadastre
Nom_section	Caractère(50)	Nom de la section
Indice_section	Caractère(5)	Indice de la section
Num_Parcelle_CA	Caractère(5)	Numéro parcelle du cadastre
Indice_num_parcelle	Caractère(5)	Indice du numéro de la parcelle cadastrale
Indice_section	Caractère(2)	Indice de section de la parcelle cadastrale
Nom_section	Caractère(50)	Nom de section de la parcelle cadastrale
Code_District_CA	Caractère(50)	Code District à qui appartient le cadastre
Contenance_CA	Caractère(50)	Contenance du cadastre
Observation_CA	Caractère(50)	Observation
Id_Consistance	Caractère(50)	Identité de la parcelle
Type_Consistance	Caractère(50)	Type de Consistance de la parcelle
Contenance_C	Caractère(50)	Surface de la parcelle cadastrale
Code_postale	Caractère(10)	Code postale du district appartenant le certificat foncier
Id_commune	Caractère(3)	Code communal par district à qui appartient le certificat foncier

Num_registre	Caractère(10)	Numéro du registre de travaux dans le BIF
Observation_C	Caractère(50)	Observation lors du traitement du dossier
Contenance_C	Caractère(50)	Surface de la parcelle certificat
Num_FN	Caractère(10)	Numéro obtenu lors du dépôt de dossier
Code_circonscription	Caractère(3)	Code de chaque circonscription appartenant le plan régulier
Num_parcelle_PR	Caractère(50)	Numéro de la parcelle du plan régulier
Date_FN	Caractère(10)	Date de dépôt de dossier plan régulier
Observation_PR	Caractère(50)	Observation lors du traitement du plan régulier
Contenance_PR	Caractère(50)	
Num_Titre	Caractère(10)	Numéro de la parcelle titré
Cod_district_T	Caractère(50)	Code District à qui appartient le Titre foncier
Num_parcelle_T	Caractère(10)	Numéro de la parcelle provenant d'une immatriculation individuelle
Nom_propriet_T	Caractère(50)	Nom de la propriété
Observation_T	Caractère(50)	observation
Contenance_T	Caractère(10)	Surface de la parcelle
Nom_Propriete_R	Caractère(50)	Nom de la propriété provenant d'une réquisition immatriculation
Num_Réquisition	Caractère(10)	Numéro de la réquisition
Cod_District_R	Caractère(50)	Code District appartenant la réquisition
Observation_R	Caractère(50)	observation
Contenance_R	Caractère(50)	Surface de la parcelle provenant d'une réquisition
Id_dp	Caractère(10)	Identité de la parcelle appartenant à un domaine public
Type_dp	Caractère(50)	Type du domaine public
Nom_ass	Caractère(50)	Nom de l'aire à statut spécifique
Type_ass	Caractère(50)	Type de l'aire à statut spécifique
Conntenance_ass	Caractère(50)	Surface de l'aire à statut spécifique

Tableau 1 : Dictionnaire de donnée

#### *10.4.2 L'élaboration du Modèle Conceptuel de Données (MCD) :*

Le MCD est un schéma montrant les tables reliées par des associations; pour cela, nous allons utiliser le logiciel « PowerAMC » pour le schématiser.

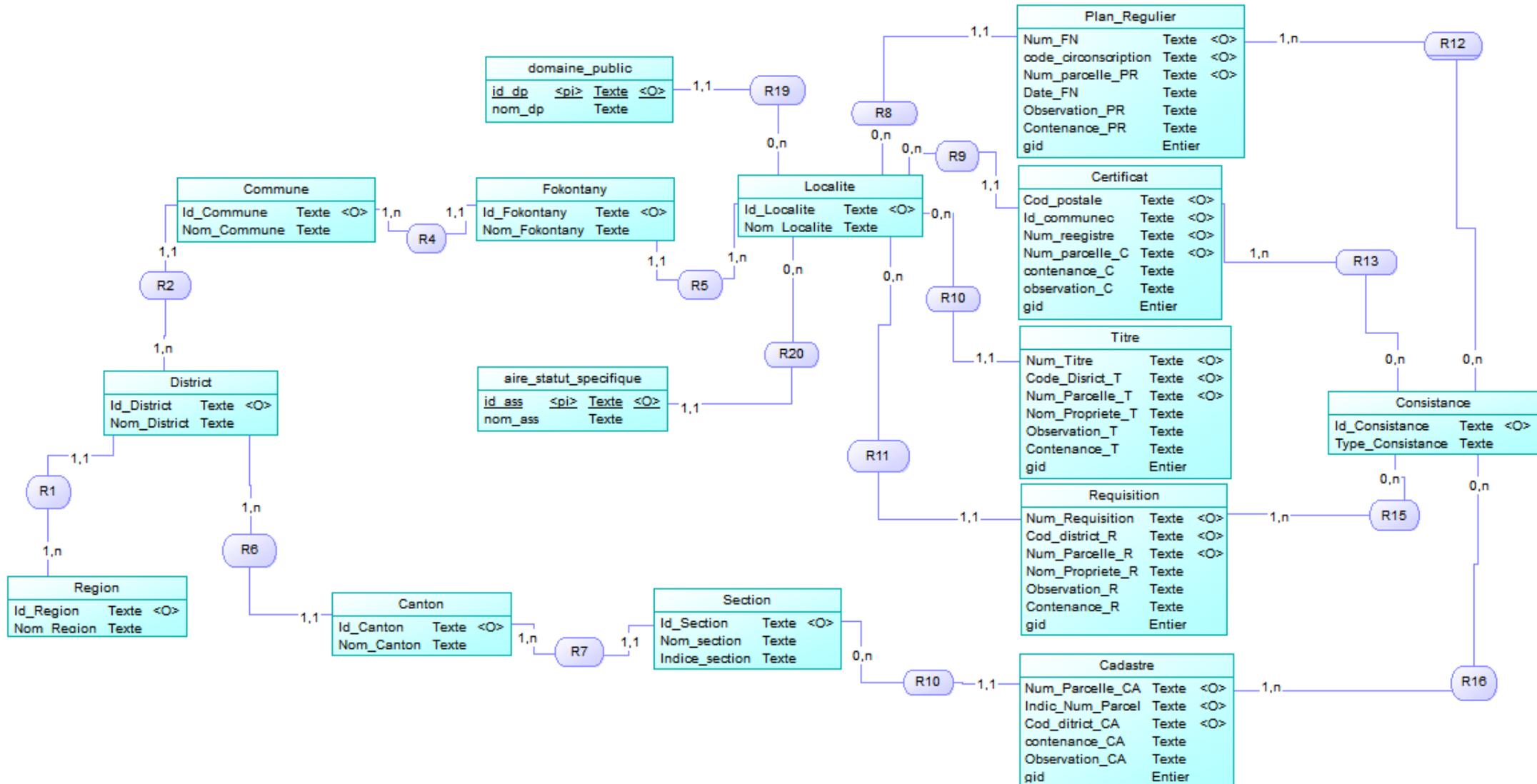


Figure 8: Modèle Conceptuel de Donnée

### 10.4.3 Le modèle logique des données (MLD) :

Le modèle logique constitue la dernière étape avant le passage sur un logiciel de SGBD. Il permet de déterminer les clés étrangères et les requêtes. Il consiste donc en deux étapes :

- ➊ Supprimer les relations complexes pour les transformer en table à part entière.
- ➋ Trouver les clés étrangères.

La clé étrangère est formée d'un ou plusieurs attributs permettant de relier la clé primaire d'une table.

**Region** (Id\_Region <pk>, Nom\_Region)

**District** (Id\_District <pk>, Id\_Region <fk>, Nom\_District)

**Commune** (Id\_commune <pk>, Id\_District <fk>, Nom\_commune )

**Fokontany** (Id\_fokontany <pk>, Nom\_fokontany <fk>, Id\_commune )

**Aire\_statut\_spécifique** (Id\_ass <pk>, Id\_localite <fk>, contenance\_ass, type\_ass )

**Plan\_regulier** (Num\_FN <pk>, Code\_circonscription <pk>, Num\_rapcelle\_PR <pk>, Date\_FN, Observation\_PR, Contenance\_PR, Gid)

**Cadastre** (Num\_Parcelle <pk>, Indice\_num\_parcelle <pk>, Indice\_section <pk>, Id\_section, Canton, Observation\_C, Contenance\_C, Gid)

**Certificat** (Code\_postale <pk>, Id\_Commune <pk>, Num\_Registre <pk>, Observation\_C, Id\_Localite <fk>, Id\_Consistance <fk>, contenance\_C, Gid)

**Titre** (Num\_Titre <pk>, Id\_District <pk, fk>, Num\_Parcelle <pk>, Nom\_Propriete, Observation\_T, contenance\_T, Id\_Localite <fk>, Id\_Consistance <fk>, Gid)

**Requisition** (Num\_Titre <pk>, Id\_District <pk, fk>, Num\_Parcelle <pk>, Nom\_Propriete, Observation\_R, Id\_Localite <fk>, Id\_Consistance <fk>, contenance\_R Gid)

#### *10.4.4 Modèle Physique des Données (MPD) :*

Dans la méthode Merise, le modèle physique des données consiste à implanter une base de données dans un SGBDR ; le langage utilisé pour ce type d'opération est le SQL.

Un système de gestion de base de données ou SGBD représente un ensemble coordonné de logiciel permettant à un utilisateur de communiquer avec une base de données pour décrire et organiser les données, rechercher, sélectionner et modifier les données, mémoriser, manipuler, interroger, traiter les données ; un SGBD offre la possibilité à l'utilisateur de manipuler les représentations abstraites des données indépendamment de leur organisation et de leur implantation sur les supports physiques, on peut considérer un SGBD comme un interpréteur d'un langage de programmation de haut niveau qui dans le cas idéal permet à l'utilisateur de décrire précisément ce qu'il veut obtenir et non comment l'obtenir.

Voici le modèle physique de données conçus à partir du MLD :

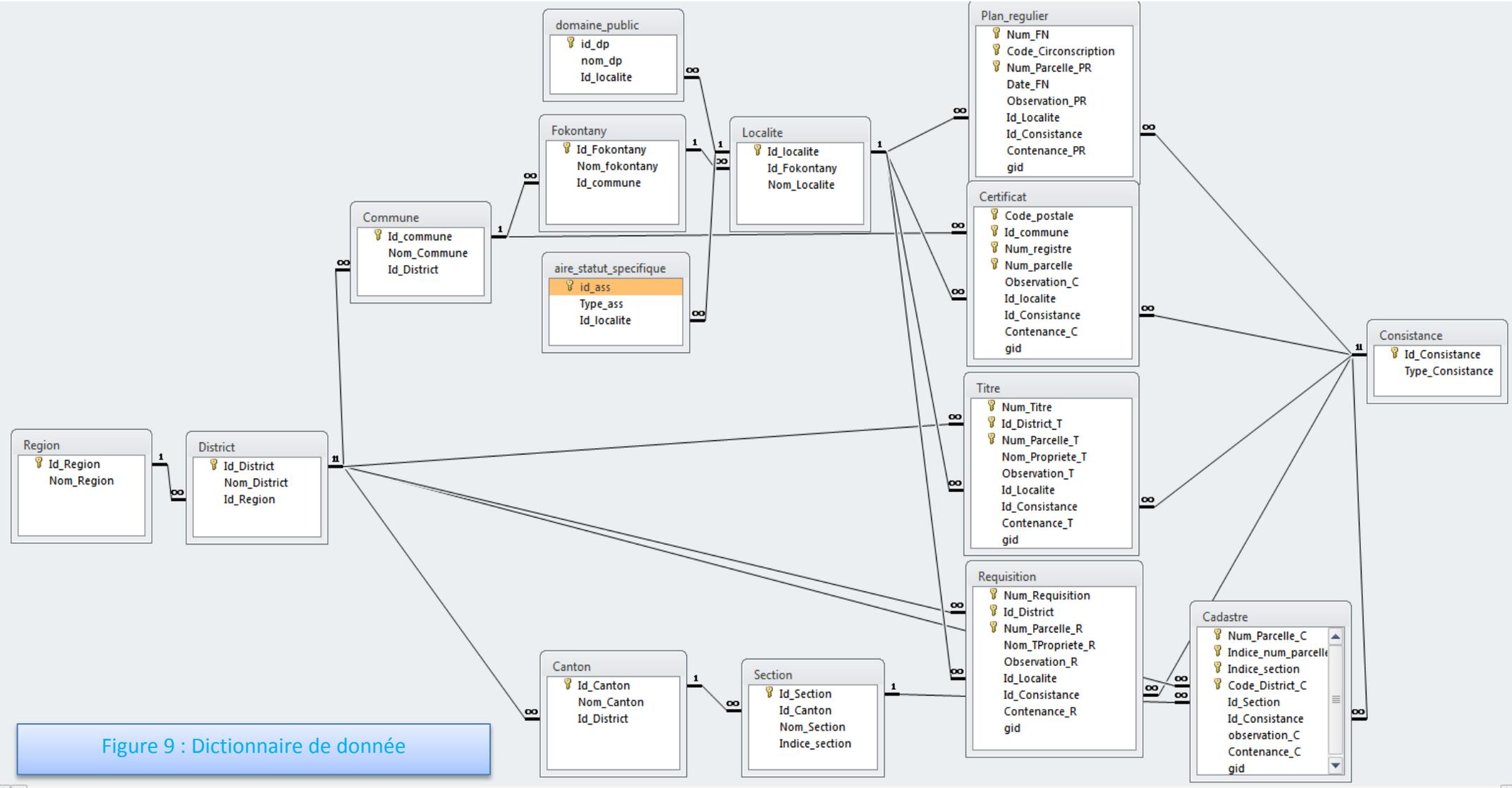


Figure 9 : Dictionnaire de donnée

## Chapitre 11 : Outils utilisés

### 11.1 Mapserver

Issu de milieux universitaires (Université du MINNESOTA USA) et amélioré par des communautés de développeurs, MapServer est un serveur cartographique Open Source (à code ouvert) permettant de réaliser des applications de Webmapping. Le Webmapping étant à la fois le processus de génération des cartes ainsi que leur diffusion sur Internet et leur visualisation dans un navigateur web. Mapserver respecte les spécificités de l'OGC. Actuellement, la version 4.4 de Mapserver s'adapte quasiment à tout type d'environnement. Il peut être facilement étendu afin de supporter de nouveaux formats de données, environnements de développement, systèmes d'exploitation ou serveurs Web. En entrée, il accepte une multitude de formats de données géographiques. En sortie, il produit des cartes interactives à destination d'Internet sur plusieurs formats.

#### 11.1.1 Mapserver et Apache

Apache est un programme permettant d'implémenter un ordinateur en serveur Web. Il utilise le module PHP pour interpréter les scripts. Il est installé directement lors de l'installation de Mapserver. C'est donc un programme capable d'interpréter les requêtes HTTP arrivant sur le port associé au protocole HTTP (par défaut le port 80), et de fournir une réponse avec ce même protocole. Mapserver peut être utilisé en CGI. Il faut rappeler qu'un serveur web est un logiciel permettant à des clients d'accéder à des pages web, c'est-à-dire des fichiers au format HTML à partir d'un navigateur (aussi appelé browser) installé sur leur ordinateur. Un des principaux intérêts de l'utilisation de CGI est la possibilité de fournir des pages dynamiques.

#### 11.1.2 Caractéristiques

MapServer est performant en termes de vitesse d'affichage des cartes.

- ☉ Il est très fiable car il peut faire face à plus de 150000 connections simultanées.
- ☉ En terme d'adaptabilité et d'évolutivité, il s'accommode quasiment à tous types d'environnement. Il peut être facilement étendu afin de supporter de nouveaux formats de données, environnement de développement, système d'exploitation ou serveur web.
- ☉ Il peut facilement intégrer différents types d'éléments cartographiques dans une application tel que l'échelle, la légende, la visibilité des couches dépendant de l'échelle, système de prévisualisation sophistiqué.

### 11.1.3 Fonctionnalités

Mapserver est installé sur la machine serveur.

- Il est capable de stocker et gérer des données localisées. Le stockage des données attributaires passe par un Système de Gestion de Bases de Données Relationnelles.
- Il permet de mettre à jour des données graphiques et attributaires.
- le serveur de carte doit être couplé avec un SGBDR capable de gérer la spatialité.
- Il retourne les informations sous forme adaptée aux navigateurs Web ;
- Il doit permettre le croisement de données, le calcul et la mesure de longueur et de superficie.

### 11.1.4 Avantages

- Le principal avantage de MapServer est le « prix » car il est totalement gratuit.
- Il incorpore plusieurs langages de programmation.
- Il fonctionne avec un maximum d'explorateur client.
- Il utilise des formats ouverts ou des formats propriétaires (shapefile, geotiff, tab).

### 11.1.5 Inconvénient

Comme la plupart des logiciels Open source :

- Mapserver nécessite un personnel motivé, et formé;
- Il présente les difficultés d'installation ;
- Il nécessite un temps d'investissement humain important.

## 11.2 PostgreSQL

Notre sujet s'inscrit dans un cadre Open source, c'est-à-dire logiciels dont les sources sont **accessibles** et **modifiables**. Dans ce contexte, plusieurs SGBD nous sont proposés. Les plus connus sont MySQL et PostgreSQL. D'une manière générale PostgreSQL est un Système de Gestion de Base de Données Relationnelles (SGBDR) développé au département d'Informatique de l'université de Californie et fonctionnant sur des systèmes de type UNIX ou WINDOWS. Son architecture est de type client/serveur. Il est ainsi constitué d'une partie serveur, dont le programme est postmaster, traitant les requêtes des clients et d'une partie client permettant d'accéder aux données. PostgreSQL supporte une grande partie du standard SQL tout en offrant de nombreuses fonctionnalités modernes :

- requêtes complexes ;
- clés étrangères ;
- déclencheurs (triggers) ;
- vues ;
- intégrité des transactions ;
- contrôle des accès simultanés (MVCC ou multiversion concurrency control).

De plus, PostgreSQL apporte une puissance additionnelle substantielle en incorporant les quatre concepts de base ci-après afin que les utilisateurs puissent facilement étendre le système. Il s'agit des concepts de classes, héritage, types, fonctions.

D'autres fonctionnalités accroissent la puissance et la souplesse : Ce sont les méthodes d'indexation, opérateurs, contraintes et les fonctions d'agrégat.

Ces fonctionnalités placent PostgreSQL dans la catégorie des bases de données objets relationnelles. Ainsi, bien que PostgreSQL possède certaines fonctionnalités orientées objets, il appartient avant tout au monde des SGBDR.

### *11.2.1 Caractéristiques de PostgreSQL*

PostgreSQL possède de nombreuses caractéristiques faisant de lui un SGBDR robuste et puissant digne des SGBDR commerciaux. PostgreSQL dispose :

- des interfaces graphiques Windows et DOS nécessaire pour gérer les bases de données ;
- des bibliothèques écrites en plusieurs langages ;
- PostgreSQL peut-être employé comme base de données spatiale principale pour les Systèmes d'Information Géographique.
- une API ODBC permettant à n'importe quelle application supportant ce type d'interface d'accéder à des bases de données de type PostgreSQL.

### *11.2.2 Fonctionnement de PostgreSQL*

PostgreSQL fonctionne selon une architecture client/serveur. Pour qu'une application cliente accède à une base de données, elle se connecte via le réseau ou localement à un postmaster en cours d'exécution. Ensuite, le Postmaster déclenche un processus serveur séparé pour gérer la connexion. Lorsqu'une requête est faite, le processus Postmaster lance un nouveau processus fils appelé postgres qui va établir la connexion entre le client et le serveur PostgreSQL.

Une fois la connexion établie, le processus client peut envoyer une requête au serveur. La requête est transmise en texte simple, c'est-à-dire qu'aucune analyse n'est réalisée au niveau de l'interface client. Le serveur analyse la requête, crée un plan d'exécution, exécute le plan et renvoie les lignes trouvées au client par la connexion établie. Ainsi PostgreSQL peut gérer plusieurs connexions à partir des processus fils qu'il génère.

## **11.3 Postgis**

PostGIS est un module d'extension de PostgreSQL, permettant d'offrir des fonctionnalités pour les objets géoréférencés et géométriques. PostGIS est développé par Réfractons Research inc., comme projet de recherche spatial de technologie de base de données. Réfractons Research inc. est une compagnie de consultation de base de données se spécialisant dans l'intégration de données et les logiciels personnalisés de développement. PostGIS est une prolongation du système de gestion de base de données PostgreSQL.

PostGIS utilise deux bibliothèques principales GEOS et Proj4. Geos est une bibliothèque qui enrichit ou complète la panoplie des fonctions spatiales de PostGIS tandis que Proj4 est une bibliothèque permettant la re-projection dans les divers systèmes de projection connus. Un système de projection étant une transcription sur un plan d'une surface courbe sans trop altérer ou provoquer des déformations.

### Caractéristiques

PostGIS est implémenté conformément aux spécifications SQL Standard de l'OGC ;

PostGIS est actuellement à sa version 1.5.0, couplable avec GEOS lui apportant des fonctionnalités bien utiles telles que Within(), Disjoint(), Touches(), GeomUnion(), Intersection(), Buffer()...

PostGIS permet de créer des bases de données afin de stocker et de traiter les données géométriques ;

PostGIS interagit avec plusieurs autres langages tel que le C, le PHP, le java ...

L'architecture des applications de Webmapping est illustrée par le schéma suivant.

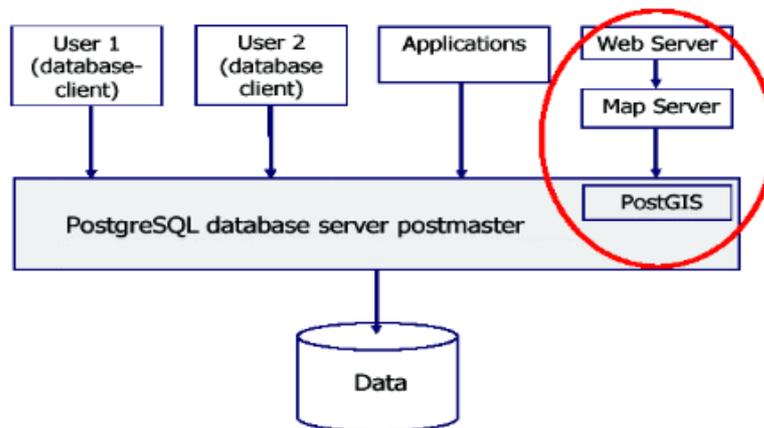


Figure 10 : Mise en évidence des applications Web et PostGIS/PostgreSQL

## Chapitre 12 : L'implémentation

### 12.1 Langages utilisés

- Le **Xhtml**, (eXtended Hypertext Markup Language) langage de balisage hypertexte, est le langage informatique créé et utilisé pour écrire les pages Web. Il sert à écrire des pages pour le World Wide Web. Conçu à l'origine comme le successeur de HTML, XHTML se fonde sur la syntaxe définie par XML, plus récente, mais plus simple que celle définie par SGML sur laquelle repose HTML. Il s'agissait en effet à l'époque de tirer parti des bénéfices techniques attendus de la simplification offerte par XML.

- ⊗ Le **SQL**, (Structured Query Language) langage de requête interprété par le SGBDR. C'est un langage informatique normalisé servant à exploiter des bases de données relationnelles. La partie langage de manipulation des données de SQL permet de rechercher, d'ajouter, de modifier ou de supprimer des données dans les bases de données relationnelles.
- ⊗ Le **PHP**, langage interprété (un langage de script) exécuté du côté serveur. Plus connu sous son sigle PHP (acronyme récursif), c'est un langage de programmation libre4 principalement utilisé pour produire des pages Web dynamiques via un serveur HTTP3, mais pouvant également fonctionner comme n'importe quel langage interprété de façon locale. PHP est un langage impératif orienté-objet.
- ⊗ Le **JavaScript**, langage de script incorporé dans un document HTML. Il est principalement utilisé dans les pages web interactives mais aussi côté serveur. C'est un langage orienté objet à prototype, c'est-à-dire que les bases du langage et ses principales interfaces sont fournies par des objets qui ne sont pas des instances de classes, mais qui sont chacun équipés de constructeurs permettant de créer leurs propriétés, et notamment une propriété de prototypage qui permet d'en créer des objets héritiers personnalisés. En outre, les fonctions sont des objets de première classe.

## 12.2 Données mises à disposition

Les données mises à disposition par le Repérage Numérique sont vectorielles, attributaires et raster.

Les données vectorielles sont au format SHP d'ESRI. Ils correspondent au Domaine Public, Limite des parcelles des terrains, aux limites Régionales, limite District, Limite communale, et limite fokontany provenant de la base de données de FTM (Foiben-Taosarintanin'i Madagasikara).

Les données attributaires proviennent des fichiers de formes (.dbf) renseignant les différentes couches géographiques.

Les données raster sont au format (.tif) qui correspond au fond orthophoto de Madagascar.

## 12.3 Installation des logiciels

Les sources disponibles sur le Web peuvent être téléchargées facilement à partir des sites web (voir annexes). Cependant, leur installation et leur configuration ne sont pas toujours aisées. Le suivi des instructions est d'une importance capitale pour le bon fonctionnement de ces programmes. Les diverses installations effectuées concernent PostgreSQL 9.1 et PostGIS 1.5.3 (Annexe 1), Apache et PMAPPER (Annexe 2), et QuantumGIS (Annexe 4).

## 12.4 Configuration de la base de données

### 12.4.1 Création de la base de données spatiales

La création d'une base de données spatiales se fait en PostgreSQL.

Cliquez droit sur Base de données/ Ajouter une base de données. Entrez le nom « mems » de la base, puis dans l'onglet définition choisissons le modèle «Template Postgis ».

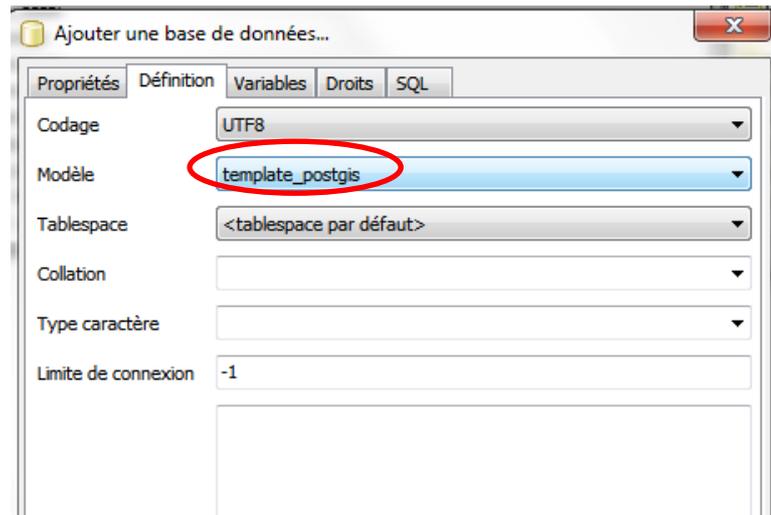


Figure 11 : création base de données spatiale

La commande **psql** est le moniteur interactif de PostgreSQL. Elle permet en outre de faire des requêtes adressées directement à un serveur PostgreSQL. On se connecte à la base de données « mems » en faisant :

```
psql mems
```

La création d'une table « titre » se fait comme suit :

```
CREATE TABLE "titre"
(
  "Num_Titre" character varying(50) NOT NULL,
  "Id_District" character varying(50) NOT NULL,
  "Num_Parcelle" character varying(2) NOT NULL,
  "Nom_Propriete" character varying(50),
  "Observation" character varying(50),
  "Localite" character varying(100),
  "Consistance" character varying(100),
  "Contenance" character varying(50),
  gid integer,
  the_geom geometry,
  code_circonscription "char",
  CONSTRAINT "Titre_pkey" PRIMARY KEY ("Num_Titre" , "Id_District" , "Num_Parcelle" ),
  CONSTRAINT "DistrictTitre" FOREIGN KEY ("Id_District")
)
```

Nous lui avons ajouté une troisième colonne où seront stockées nos données géométriques. Ceci peut se faire directement ou en utilisant la fonction AddGeometryColumn() dont la syntaxe est la suivante :

```
AddGeometryColumn ([Table], [ nom_Colonne_Geometrique ], [SRID], [ Type ], [Dimension ]);
```

On a donc:

```
SELECT AddGeometryColumn ( 'titre ', 'geom ', 29702, 'POLYGON', 2 );
```

#### 12.4.2 Création de vue dans PostgreSQL

Pour faciliter l'affichage et la recherche dans PMAPPER, il nous faut faire une requête. Cette requête consiste la concaténation de certains champs et remplacement d'un code par sa vraie valeur.

Code SQL pour la création d'une vue du couche Titre :

```
(  
SELECT DISTINCT p."gid", x.info_titre,  
p."Num_Parcelle",p."Nom_Propriete",p."Observation",p."Id_Localite",p."Id_Consistance",p."Co  
ntenance",p."the_geom"  
FROM "Titre" p, "Localite" l, (select concat (p."Num_Titre",'',p."code_circonscription",'  
'd."Nom_District") info_titre, p."gid" y from "Titre" p, "District" d where  
p."Id_District"=d."Id_District") x  
WHERE x.y=p."gid";  
)
```

### Chapitre 13 : La présentation du prototype

Le modèle de page web conçu est le suivant :

#### 13.1 Page d'accueil

La page d'accueil comprend deux principales parties à savoir le corps et la zone de connexion.

- La première partie permet à l'utilisateur de savoir la généralité sur le contenu et l'objectif de la conception du site web
- la zone "connexion" permet à un utilisateur, ayant un compte utilisateur, de se connecter pour accéder à la page web contenant la carte dynamique présentant la Repérage Numérique à Madagascar



Figure 12 : Page d'accueil du Repérage Numérique

## 13.2 Page pour la carte dynamique

### 13.2.1 Interface

Voici un aperçu de l'interface de **RN** :

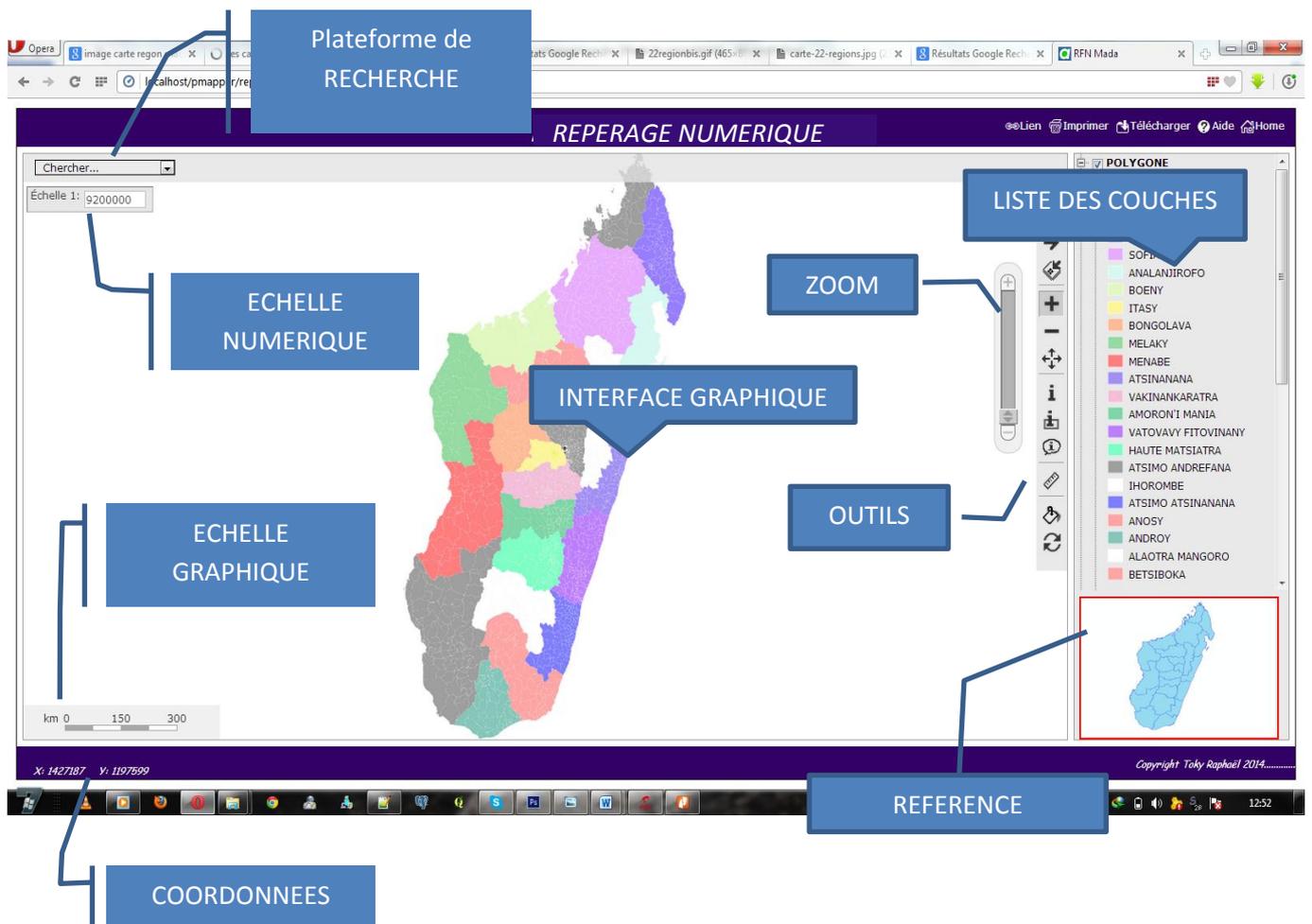


Figure 13 : Interface du Repérage Numérique

Il est constitué des éléments suivants :

**Une barre de titre** : elle spécifie le titre et le sujet traité par l'application

**Une barre d'outils**: elle contient les outils de manipulation immédiate de la carte. Ce sont les outils :

- Zoom In  qui permet de faire un rapprochement immédiat d'une zone ciblée sur la carte
- Zoom out  qui permet de s'éloigner d'une zone ciblée sur la carte
- Zoom all qui permet de voir l'ensemble des données
- Zoom qui définit les différentes fonctionnalités de manipulation par zoom sur la carte (Zoom In, Zoom Out)
- « Refresh » pour réinitialiser la carte.

**Une zone d'affichage de la carte** : elle affiche les cartes présentes.

**Une zone d'affichage des couches** : formée de 3 groupes tels que: Ligne, Polygone et Raster.

Elle est constituée par :

- Une légende qui explique les symboles visibles sur la carte
- Un panel couche qui spécifie les couches actives sur la carte

**Une zone de référence:** elle contient :

Une carte de localisation qui permet de situer géographiquement la zone ciblée sur la carte

**Une zone de recherche :** Elle permet de faire une recherche spécifique à une couche et de faire apparaître les données attributaires correspondantes.

Voici un aperçu d'un résultat de recherche :

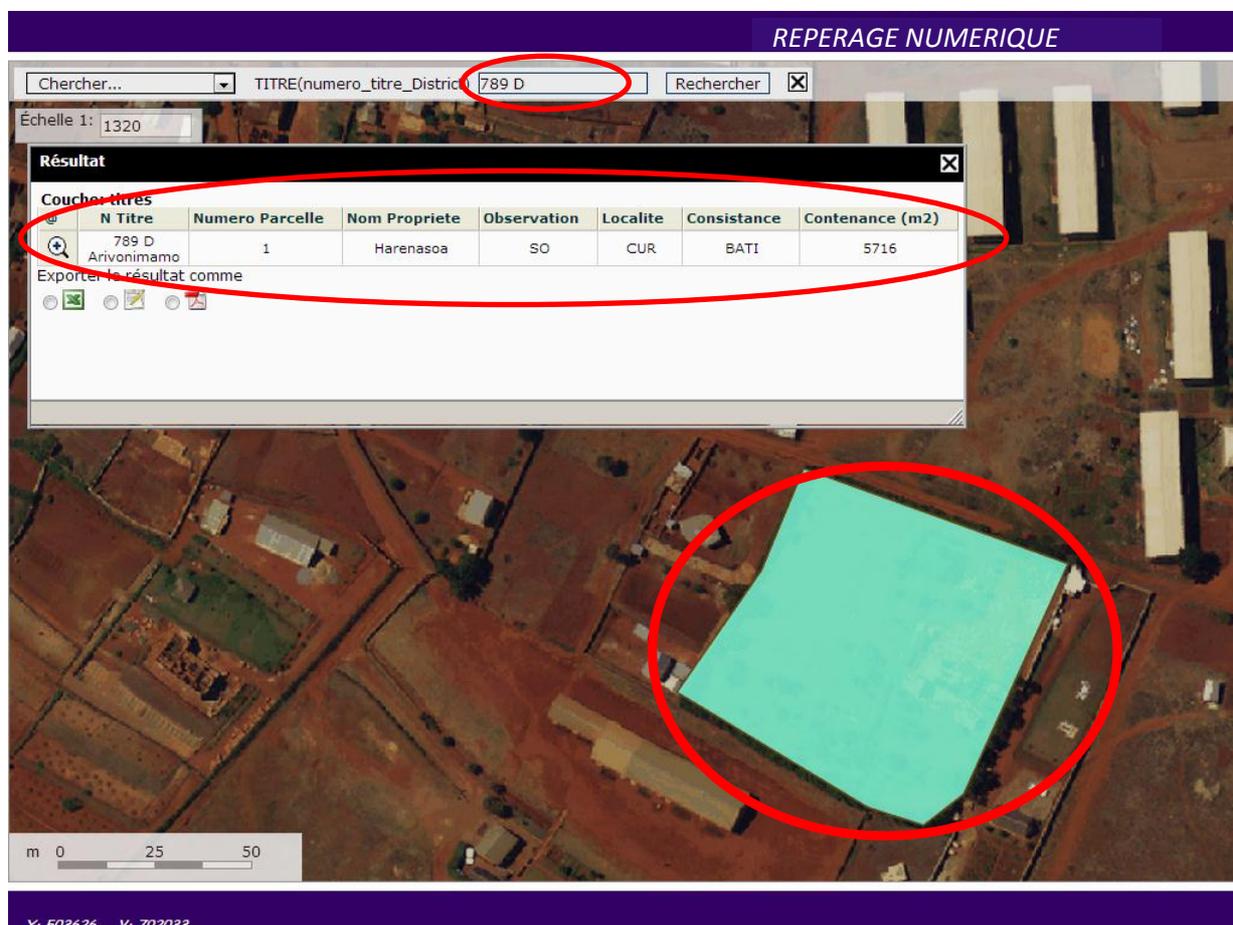


Figure 14 : Résultat de recherche sur RN

Aperçu d'une impression dans le RN :

Configuration de l'impression :

**Paramètres d'impression** [X]

**Paramètres d'impression** [Printer Icon]

Échelle 1: 1400

Avec carte d'aide à la localisation

Créer un document PDF

Standard

Titre de l'impression  
HARENASOA T 789D

Format | A3 ▼

Orientation | Paysage ▼

Légende | En bas, en lignes ▼

Créer la page d'impression

Figure 15 : configuration d'impression

Page d'impression :

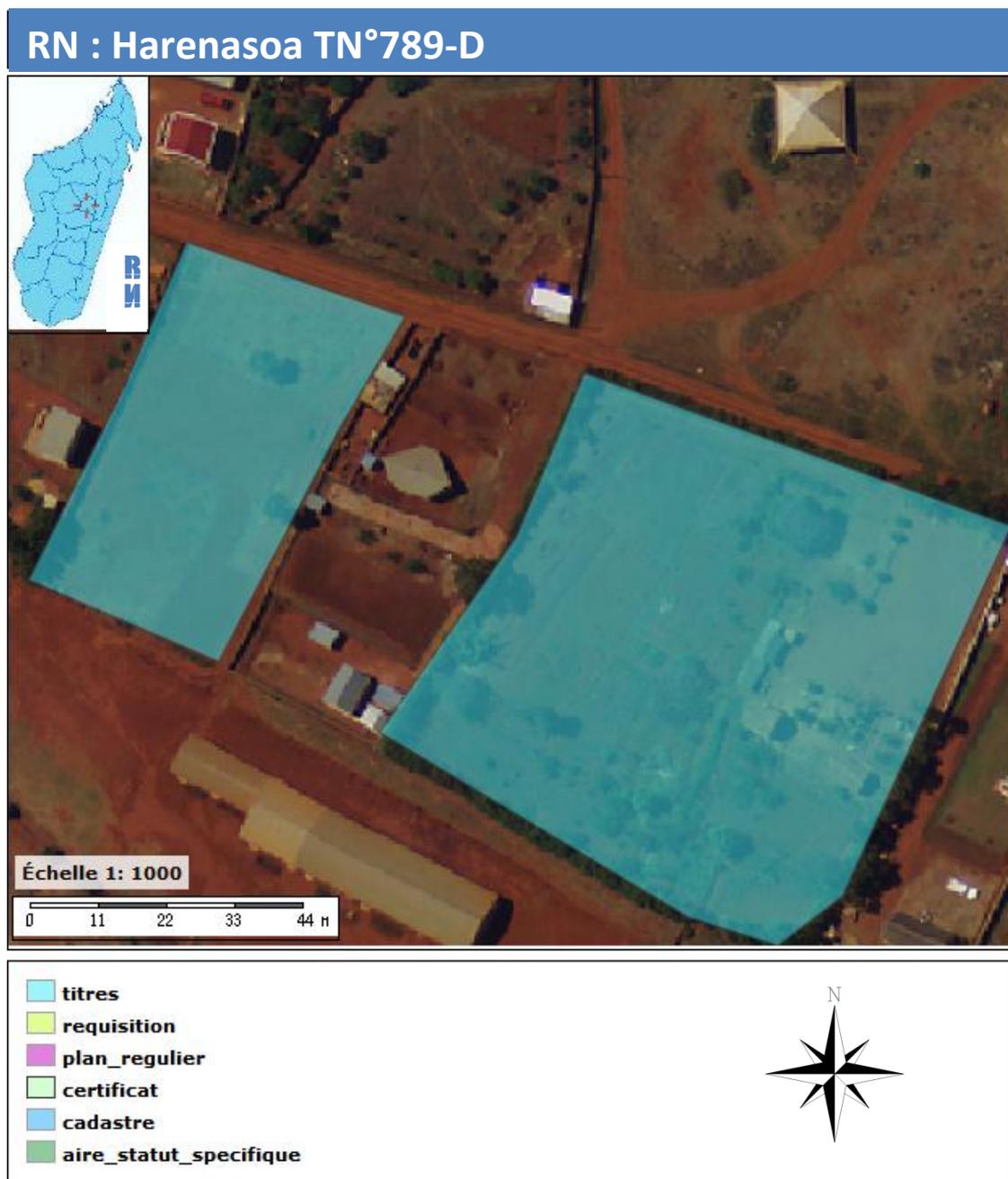


Figure 16: Impression dans le RN

### 13.2.2 Fonction

Le cadre RN est destiné à offrir une large fonctionnalité et de multiples configurations afin de faciliter l'utilisation. Les fonctions incluses sont:

- L'interface DHTML (DOM) zoom / pan (pas d'utilisation de trames)
- Zoom / Pan aussi via les touches du clavier, molette de la souris, la carte de référence, curseur
- Configuration facile de présentation et le comportement avec le fichier de configuration XML

- Fonctions de requête (identifier, sélectionner, recherche)
- Recherche entièrement sélectionnée d'attribut, y compris suggérer, cochez les cases, etc
- Mise en page flexible de résultats de la requête par l'intermédiaire de modèles JavaScript
- Résultats de la recherche s'affichent avec base de données joint et hyperliens
- Interface utilisateur multilingue prédéfini: Français FR, Anglais EN
- Entièrement XHTML 1.0 Strict conforme
- légendes HTML et différents styles d'affichage de la légende et des couches
- Fonctions d'impression: HTML et PDF
- Pop-up identifié lorsque vous déplacez avec la souris sur la carte
- Lancer la carte de mesure de zoom prédéfini: par mesure explicite ou fonction de couche
- Divers plugins: la transparence de la couche, résultat de la requête à l'exportation, et plus

### 13.2.3 Fonction supplémentaire

Grace au fichier ini/config/config.ini, cette application a les fonctions supplémentaires suivantes :

- `allGroups`  
définit la façon dont les groupes / couches sont affichés dans la table des matières. Si une couche appartient à un groupe ajouter le nom du groupe ici, sinon utiliser le nom de la couche. Entrée obligatoire, afin d'adapter en fonction de vos paramètres.
- `defGroups`  
définit quels groupes / couches sont activés par défaut lors du démarrage de l'application. Si une couche appartient à un groupe d'ajouter le nom du groupe, sinon utiliser le nom de la couche. S'il n'est pas défini, toutes les couches sont sous tension.
- `limitResult`  
Limite les résultats de la sélection avec l'outil de sélection ou de recherche. Utile pour éviter la charge du serveur lourde quand l'utilisateur sélectionne très beaucoup de fonctionnalités.
- `highlightColor`  
Sélectionnez la couleur de sélection / recherche. valeur RVB.
- `highlightSelected`  
Définit si la fonction SELECT provoque fonction surbrillance.
- `AutoZoom`  
Réglage de zoom-in automatiquement après les résultats de sélection ou la recherche sont affichés.
- `zoomAll`  
Ajoute un bouton "Zoom à toutes les fonctions" entraîner table pour sélectionner ou la recherche.

- `tocStyle`  
Définit le style de la table des matières:
  - `arbre`: vue de style arbre; comme l'affichage des fichiers dans l'Explorateur Windows (utilise une classe de JS de Geir Landro)
  - `plat`: vue Appartement de couches, basculé entre les couches et la légende
  - `combi`: vue Appartement de couches combinées avec la légende
- `scaleLayers`  
Liste des couches (TOC) mise à jour automatiquement en fonction de l'échelle
- `layerAutoRefresh`  
Actualiser automatiquement la carte lorsque la sélection des couches changé
- `iCow / CIST`  
largeur de l'icône / hauteur en pixels.
- `imgFormat`  
Format d'image pour les icônes de la légende
- `INFOWIN`  
Style de l'affichage de la fenêtre de résultats de la requête (identifier / de recherche)
  - `fenêtre`: Dans la fenêtre séparée
  - `cadre`: dans infoframe de la fenêtre principale de cadres
- `alignQueryResults`  
Permet l'alignement automatique des colonnes dans les tableaux de résultats pour les requêtes (identifier, sélectionner, recherche), selon estimée type de données (chaîne, nombre, monnaie, etc.)
- `sliderMax / sliderMin`  
Min et Max valeur pour le zoom avec curseur. Doit être adapté à mesure spécifique et étendue de l'échelle des ensembles de données.
- `toolBarOrientation`  
Orientation de la barre d'outils. Barre d'outils sera automatiquement écrit dans le `toolFrame`. Définissez ici si leur orientation doit être horizontale ou verticale, en fonction de votre mise en page de cadres.
- `enableRightMousePan`  
Active la fonction PAN avec le bouton droit de la souris enfoncé, même si un autre outil est actuellement actif.