



**Ecole Nationale des Sciences Géographiques**

**Rapport de projet de recherche**

**Cycle: IT**

**Auteur**

David NIAMIEN

**Titre**

# **Développement d'un outil de visualisation de données de télédétection**

***Projet dirigé par:*** Adrien GRESSIN

Soutenu à Champs sur Marne le 28 Janvier 2015

Laboratoire : MATIS

73, Avenue de Paris, 94165 Saint Mandé Cedex (France)

## Abréviations et acronymes

**JS:** JavaScript

**MATIS:** Méthodes d'Analyses pour le Traitement d'Images et la Stéréorestitution

**OGC:** Open Geospatial Consortium

**PDO:** PHP Data Objects

**PHP:** Hypertext Preprocessor

**SGBD:** Système de Gestion de Base de Données

**UML:** Unified Modeling Language

**XML:** eXtensible Markup Language

## Sommaire

<b>Abréviations et acronymes.....</b>	<b>2</b>
<b>Introduction .....</b>	<b>5</b>
<b>I. Analyse.....</b>	<b>6</b>
1. Diagramme de cas d'utilisation .....	6
▪ Représentation textuelle des différents cas d'utilisation .....	7
2. Diagramme de classe .....	9
3. Analyse technique .....	10
▪ Extraction des données .....	10
▪ Système de Gestion de Base de Donnée.....	11
▪ Serveurs.....	12
▪ Affichage des données géographiques .....	14
<b>II. Implémentation .....</b>	<b>16</b>
1. Traitement des fichier XML .....	16
▪ Ecriture de scripts en python pour l'extraction et le stockage des métadonnées et de l'emprise géographique des images issues de fichier XML .....	16
2. Utilisation de la base de données avec PHP .....	16
▪ Serveur Web : Installation de WAMP server .....	16
▪ Langage PHP.....	17
▪ Etape 1 et 2 : Connexion au serveur de données .....	18
▪ Etape 3: Requête.....	18
▪ Etape 4: Exploitation des résultats.....	18
▪ Etape5 : Fermeture de la base.....	18
3. Solution côté client .....	19
▪ Technologies utilisées.....	19
<b>III. Résultats.....</b>	<b>19</b>
1. Affichage des fonds cartographiques.....	19
2. Affichage des images disponibles .....	19
▪ Requêtes sémantiques .....	19
<b>Conclusion et perspectives .....</b>	<b>20</b>
<b>Références.....</b>	<b>21</b>
<b>A. Annexes : guide d'utilisateur .....</b>	<b>21</b>
1. Présentation de l'outil .....	22
▪ Administration de données .....	22
▪ Administration du serveur.....	25
▪ Client web léger .....	26

## Table des figures

Figure 1: diagramme des cas d'utilisation.....	7
Figure 2: diagramme de classes .....	10
Figure 3: architecture générale de l'application .....	15
Figure 4: Architecture des drivers PDO (source : (Daspet & De Geyer, 2012)).....	17
Figure 5: utilisation de PHP pour accéder à la base de données .....	18
Figure 6: Affichage de fond de carte GeoPortail .....	19
Figure 6: Aperçu de la page qui permet de lister les images disponibles en fonction des critères de sélection .....	20

## Table des tableaux

Tableau 1: part du marché pour chaque serveur http .....	13
--	----

## Introduction

Dans le cadre de la formation du cycle Ingénieur de l'Ecole Nationale des Sciences Géographiques, un projet de recherche est réalisé en deuxième année par chaque étudiant afin de l'initier au monde de la recherche. C'est dans ce cadre, que le laboratoire de recherche en traitement d'image (MATIS) de l'Institut National de l'Information Géographique et Forestière a proposé un sujet de développement informatique pour la visualisation des données géographiques en particulier les données issues des capteurs de télédétection en différents modes d'acquisition.

L'objectif de ce projet est de mettre en place un outil de gestion et de visualisation des données géographiques disponibles au sein du laboratoire à travers une application Web.

En effet, le laboratoire dispose pour différents sites d'étude en France, des données hétérogènes issues de différents capteurs lidar, optique et radar. Il souhaite à travers une application web accéder à ces données, leurs métadonnées, leurs emprises au sol. Cet outil devant permettre aux équipes de recherche au sein du laboratoire d'avoir une connaissance rapide des données dont il dispose.

Après l'analyse du cahier de charge, le travail a été scindé en trois parties. La première traite l'analyse de l'étude. La seconde partie concerne l'implémentation de la solution retenue à la suite de l'analyse. Et enfin, la dernière partie présentera les résultats obtenus.

## I. Analyse

Le MATIS dispose de données diverses et variées issues des différents modes d'acquisition d'images de télédétection. Pour les données optiques, il dispose des images issues des capteurs Pléiades, SPOT, LandSat, WorldView, RapideEye et des caméras numériques de l'IGN. Des données radar du capteur TERRASAR-X sont également disponibles ainsi que des données issues des capteurs LIDAR.

Le commanditaire de l'étude souhaite avoir des informations détaillées sur ces données issues des capteurs précédemment cités à savoir les dates d'acquisition des différentes données pour mieux interpréter les traitements réalisés. Les caractéristiques des capteurs ayant acquis les données doivent être également connues ainsi que le format stockage de chaque donnée. Les données sont obtenues soit, par des acquisitions propres (cas des prises de vue aériennes) soit dans le cadre d'une collaboration avec des institutions partenaires ou dans le cadre d'un projet de recherche. Les investigateurs Principaux, co-investigateurs de chaque projet doivent être connus. Il doit également mettre à jour la base de données lorsque de nouvelles données sont disponibles.

Afin de mieux exprimer le besoin du commanditaire, nous passons par une analyse fonctionnelle et statique des activités touchées par notre étude. Pour cela, nous avons recours aux diagrammes UML.

En effet, ceux-ci permettent de définir une application selon plusieurs points de vue :

- Analyse fonctionnelle : diagramme de cas d'utilisation
- Analyse statique : diagrammes de classes

L'analyse dynamique (diagramme d'état, de séquence, d'activité) ne sera traitée dans notre étude ainsi que les diagrammes de composant et de déploiement.

Une analyse technique sera réalisée pour expliquer le choix des différents outils.

### **1. Diagramme de cas d'utilisation**

Le diagramme de cas d'utilisation « use cases » permet la description du comportement du système du point de vue de l'utilisateur sous forme d'actions et de réactions. Un acteur représente le rôle joué par un objet (personne ou chose) qui interagit avec le système (DEBRAUNWER & VAN DER HEYDE, 2008). Un objet peut jouer plusieurs rôles vis-à-vis du système.

Un diagramme de cas d'utilisation a été réalisé afin de mettre en évidence les relations entre chaque acteur et le système.

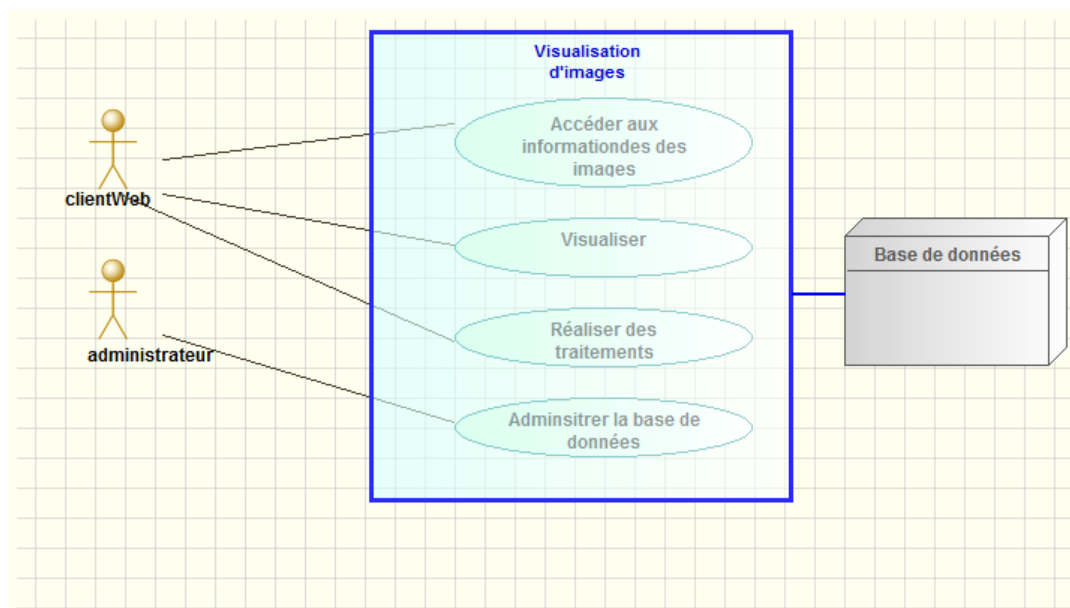


Figure 1: diagramme des cas d'utilisation

#### ■ Représentation textuelle des différents cas d'utilisation

Cette représentation sous forme textuelle des cas d'utilisation donne une description de leur comportement, de leurs actions et réactions.

- Cas d'utilisation : **Accéder aux informations**

**Résumé** : Ce cas d'utilisation permet d'accéder aux métadonnées des images et des capteurs.

Cas d'utilisation	Accéder aux informations des images
Acteur	clientWeb
Système	Outil de visualisation d'images
Intervenants	clientWeb
Niveau	Objectif utilisateur
Préconditions	Présence de données Autorisation d'accès
Opérations	
1	Connaître les nombres d'images dans une zone géographique en fonction de la date d'acquisition, niveau de traitement et le capteur qui a acquis les données
2	Effectuer une requête spatiale : soit en saisissant les coordonnées géographiques d'un endroit ou en dessinant un polygone ou rectangle englobant dans la zone souhaitée.

- Cas d'utilisation : **Visualiser**

**Résumé :** Ce cas d'utilisation permet de visualiser l'emprise géographique des différentes images disponibles dans la base de données.

Cas d'utilisation	Visualiser
Acteur	clientWeb
Système	Outil de visualisation d'images
Intervenants	clientWeb
Niveau	Objectif utilisateur
Préconditions	Présence de données Autorisation d'accès
Opérations	
1	Visualiser la trace des images
2	Visualiser les images

- Cas d'utilisation : **Réaliser des traitements**

**Résumé :** Ce cas d'utilisation permet de réaliser des traitements de base sur les images.

Cas d'utilisation	Réaliser des traitements
Acteur	clientWeb
Système	Outil de visualisation d'images
Intervenants	clientWeb
Niveau	Objectif utilisateur
Préconditions	Présence de données Autorisation d'accès
Opérations	
1	Réaliser une composition colorée
2	Réaliser un étalement d'histogramme
3	Réaliser des calculs statistiques de base : <ul style="list-style-type: none"> <li>- Calcul d'indices</li> <li>- Calcul de moyenne, d'écart-type, de coefficient de variation</li> </ul>

- Cas d'utilisation : **Administrer**

**Résumé :** Ce cas d'utilisation permet à l'administrateur d'administrer la base de données.

Cas d'utilisation	Administrer
Acteur	Administrateur
Système	Outil de visualisation d'images
Intervenants	administrateur
Niveau	Objectif utilisateur
Préconditions	
Opérations	
1	Créer la base de données
2	Mettre à jour la base
3	Créer des comptes des utilisateurs



## 2. Diagramme de classe

Plutôt que de répertorier les acteurs textuellement, on va réaliser un diagramme de classes dans lequel chaque acteur est relié par une association à une classe ce qui nous permettra de spécifier le nombre d'instances d'acteurs connectés au système à un moment donné. Notre diagramme de classes est constitué de six classes qui sont les suivantes :

- **image**: désigne l'ensemble des images disponibles. Elle est caractérisée par le nom de l'image, le type de codage, le mode de stockage, la date d'acquisition, la géométrie d'acquisition, la trace au sol, le format, le lieu d'acquisition, le système de coordonnées associé, l'emprise géographique, la version de l'image, le nombre de lignes et de colonnes...
- **projet** : désigne l'ensemble des différents projets. Il est caractérisé par le nom du projet, la durée du projet, l'investigateur principal et un champ pour les commentaires...
- **capteur** : il s'agit d'une classe abstraite c'est-à-dire une classe qui n'instancie pas directement. Elle nous permettra de stocker les informations communes de classes spécialisées **lidar**, **radar** et **optique**.
- **lidar** : il hérite de la classe **capteur** et caractérisé par le type, la densité de point, la description des échos...
- **radar** : il hérite de la classe **capteur** et caractérisé par le type, la résolution spatiale en géométrie radar, la résolution spatiale en géométrie sol, le mode d'observation, les modes polarimétriques, l'angle d'incidence, l'orientation de l'orbite, la direction de visée...
- **optique** : il hérite de la classe **capteur** et caractérisé par le type, la résolution spectrale, la résolution radiométrique, la résolution spatiale, résolution temporelle...

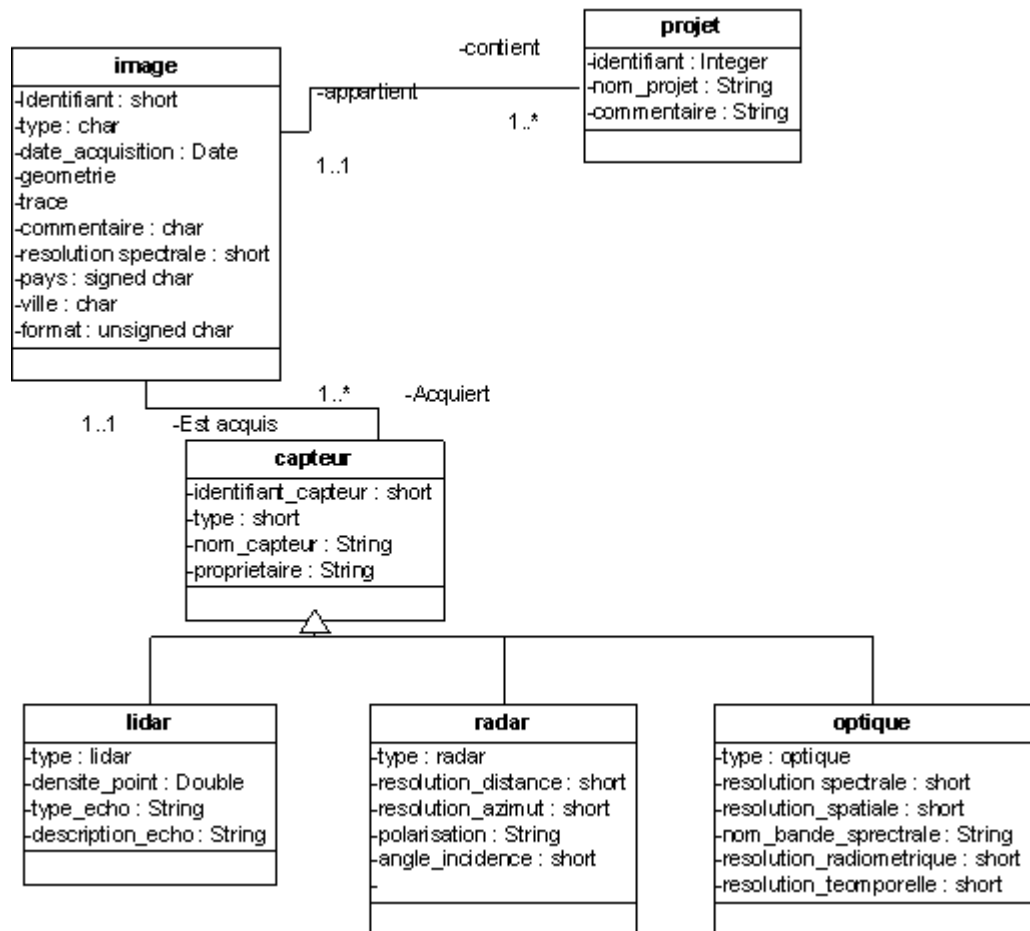


Figure 2: diagramme de classes

### 3. Analyse technique

Dans cette partie, un état de l'art des différentes méthodes et outils et leur choix seront traités.

#### ▪ Extraction des données

L'extraction des données est divisée en deux parties. La première traite l'analyse de fichier XML associé à chaque image. Il consiste à « parser » le fichier XML dans le but d'extraire les attributs répondant à notre diagramme de classe. La seconde étape consiste à insérer les attributs extraits dans la base de données. Bien qu'il existe de nombreux outils tels que PHP, Java pour manipuler du XML, nous allons nous intéresser aux possibilités qu'offre Python pour manipuler ce format de données. Nous utiliserons à cet effet, différentes bibliothèques dont **xml.etree.ElementTree** pour « parser » le fichier XML et **PsycoPG2** pour l'accès à la base de donnée et **daemons** pour la mise à jour automatique. Nous avons utilisé Python pour les raisons suivantes:

- le script se réalise en ligne de commande,
- prise en main facile avec un grand nombre de librairies,
- forums d'échange très actifs.

#### ▪ Système de Gestion de Base de Données

Une base de données est définie comme un ensemble structuré de données accessibles par l'ordinateur pour satisfaire simultanément plusieurs utilisateurs en un temps opportun.

Un Système de Gestion de Base de Données (SGBD) est quant à lui défini comme un logiciel qui permet d'interagir avec une base de données. Il permet à un utilisateur de définir les données, de consulter la base ou de la mettre à jour. Les Systèmes de Gestion de Base de Données se distinguent clairement des systèmes fichiers par le fait qu'ils permettent la description des données (nom, formats et caractéristiques..) de manière séparée de leur utilisation (mise à jour et recherche). Un SGBD répond aux objectifs :

- La **cohérence des données** : le SGBD doit permettre de garantir que certaines règles soient respectées lors de la manipulation des données (contraintes d'intégrité).
- L'**administration centralisée des données** : l'indépendance des données rend possible l'administration centralisée des données.
- La **souplesse d'accès aux données** : les utilisateurs (informaticiens et non-informaticiens) doivent pouvoir manipuler les données.
- La **sécurité des données** : le SGBD protège les données contre les accès non autorisés ou mal intentionnés (confidentialité).
- L'**accès concurrents sur les objets** : le SGBD doit permettre à plusieurs utilisateurs d'accéder simultanément aux bases de données comme s'ils étaient seuls.

Parmi les principaux SGBD du marché, nous pouvons les classer pour notre étude en deux groupes : les SGBD intégrant la cartouche géographique et les SGBD classiques.

- SGBD n'intégrant pas la cartouche spatiale : Microsoft SQL Server, Microsoft Access, dBase...
- SGBD intégrant la cartouche spatiale : PostgreSQL/PostGIS, Oracle Spatial, Mysql-Gis, Ingres...

Dans le cadre de l'étude, les SGBD spatiaux seront utilisés vu la nature des données manipulées. Pour le choix, on s'est appuyé sur l'étude réalisée par les sociétés LINAGORA et CampToCamp pour le compte du Centre National d'Etudes Spatiales (CNES) sur la comparaison des SGBDR à cartouches spatiales. Dans cette étude [1], il est rapporté que :

- MySQL est encore un projet jeune qui n'implémente pas toutes les recommandations de l'OGC. Il manque encore pas mal de fonctions de création de données géographiques. Mais ce dernier est prometteur et s'avère être le plus rapide sur les requêtes spatiales.

- PostgreSQL/PostGIS est très complet puisqu'il recouvre toutes les spécifications de l'OGC, plus des fonctions supplémentaires de création de géographie ou bien de requêtes complexes tout en étant moins complet qu'Oracle. Mais le projet est dynamique et en perpétuelle amélioration. Des modules externes comme Pgrouting permettent d'ajouter des fonctionnalités. Il s'interface avec de nombreux outils du monde du libre (Qgis, MézoGis, Mapserver, GeoServer, Python, PHP ...).

- Oracle Spatial est un projet non libre et payant (assez cher) qui offre une multitude de fonctionnalités (géocodage, réseaux, etc.). Il supporte toutes les fonctionnalités de l'OGC

mais ne respecte pas le nommage ce qui est problématique lors de la migration d'un système à un autre. Il pêche parfois sur ces performances et donnent de moins bons résultats que PostGIS sur certaines fonctions. Il s'interface avec un grand nombre de logiciels commerciaux. PostgreSQL est aujourd'hui un logiciel libre et ouvert et un serveur les plus avancés.

Les connaissances acquises pendant de la formation à l'école et la cohérence avec les outils de l'IGN ont motivé notre choix pour ce logiciel.

- **Serveurs**

La partie serveur est divisée en deux groupes : les serveurs http qui sont des logiciels qui prennent en compte les requêtes client-serveur du protocole http développée par le World Wide Web et les serveurs web qui sont eux, des plateformes de développement Web.

- **Serveurs http**

On définit le serveur http comme un logiciel prenant en compte les requêtes client-serveur du protocole http développée par le World Wide Web. Il en existe plusieurs dont les plus connus sont : Apache, IIS, Nginx, Apache Tomcat, Sun Java System Web Server...

- **Apache**

C'est un logiciel open-source et offre des possibilités de modification pratiquement illimitées. Apache est également un logiciel libre et prend en charge une vaste gamme de technologies web gratuits, y compris PHP, Perl, Python, Ruby et de nombreuses technologies propriétaires. Il est disponible sur les principaux systèmes d'exploitation et plates-formes matérielles, y compris Microsoft Windows, Linux, BSD et autres systèmes UNIX et OS /X. C'est également un logiciel très populaire.

- **IIS**

Le deuxième serveur le plus utilisé sur l'Internet est Internet Information Services (IIS). Il s'agit du serveur primaire utilisé dans le déploiement des technologies Microsoft telles que « asp.net » tout en soutenant diverses autres technologies Web offerts par des tiers. Il ne fonctionne que sur les systèmes d'exploitation serveur de Microsoft cependant, et est un logiciel propriétaire, nécessitant licence.

- **Apache Tomcat**

Le serveur Web Apache Tomcat est souvent utilisé avec Apache pour exécuter les servlets Java et le code JSP. Il peut être exécuté avec d'autres serveurs web, ou même comme un serveur web autonome lui-même quand le seul contenu pour être servi est basé sur les servlets Java et /ou JSP. Une partie du projet Apache Jakarta qui développe des logiciels open-source et gratuit pour la plate-forme Java de Sun, il prend également en charge la plupart des systèmes d'exploitation pris en charge par le serveur web Apache.

- **Sun Java System Web Server**

Ce serveur web propriétaire libre mais closed-source est développé par Sun Microsystems. Malgré son nom, il prend en charge les technologies Web non - Java, tels que PHP, Perl,

Python, Ruby et ASP avec les technologies Web basées sur Java, y compris les servlets Java et les JSP. Windows, Unix et Linux sont également supportés.

- **Nginx**

Nginx est un serveur asynchrone par opposition aux serveurs synchrones où chaque requête est traitée par un processus dédié. Il a été écrit par Igor Sysoev pour les besoins d'un site russe à très fort trafic. Nginx est reconnu pour ses hautes performances, sa stabilité, son ensemble de fonctionnalités, sa configuration simple ainsi que sa faible consommation de ressources. Le tableau ci-dessous montre la part du marché pour chaque serveur http.

Part de marché des serveurs web en juin 2014						
Serveur	Licence	Mai-2014	Part de marché	Juin-2014	Part de marché	Evolution
Apache	libre	366 262 346	37.56%	353 672 431	36.50%	-1.05
Microsoft IIS	propriétaire	325 854 054	33.41%	352 208 487	36.35%	2.94
Nginx	libre	142 426 538	14.60%	133 763 494	13.81%	-0.8
Google	propriétaire	20 685 165	14.60%	20 192 595	2.08%	-0.04

Source : Netcraft

Tableau 1: part du marché pour chaque serveur http

Le serveur http Apache a été utilisé dans le cadre de ce travail. Ce choix s'explique par le fait qu'Apache est un projet Open Source. IL peut fonctionner sur des systèmes d'exploitation libre comme Linux et a une configuration standard et simple. Apache supporte les modules chargeables, ce qui rend capable d'exécuter presque n'importe quel langage de script dynamique disponible, y compris Perl, Python, PHP et plus encore. Comme le montre le tableau ci-dessus Apache est un logiciel très populaire.

- **Serveurs Web**

- **WamServer**

WampServer (WAMP signifie Windows Apache MySql PhpMyAdmin) est une plate-forme de développement Web libre sous Windows pour des applications Web dynamiques à l'aide du serveur Apache2, du langage de scripts PHP et d'une base de données MySQL. Il possède également PHPMyAdmin pour gérer plus facilement vos bases de données. L'équivalent sous Linux est LAMP (Linux Apache MySql PhpMyAdmin).

- **Xampp**

XAMPP est un pack d'outils, disponible en OpenSource et écrit en PHP, regroupant MySQL, PHP et Perl qui permet de mettre facilement en place un serveur web. Ce projet a été réalisé dans le but de simplifier au mieux l'installation et la configuration d'un serveur Web Apache. Il est disponible sous Linux, Mac OS et Solaris.

- **EasyPHP**

Il s'agit d'une plateforme de développement Web libre, permettant de faire fonctionner localement (sans se connecter à un serveur externe) des scripts PHP. EasyPHP n'est pas en soi

un logiciel, mais un environnement comprenant deux serveurs (un serveur web Apache et un serveur de bases de données MySQL), un interpréteur de script (PHP), ainsi qu'une administration SQL phpMyAdmin.

Wamp a été utilisé dans le cadre de ce travail car la configuration et la prise en main sont simples.

- **Affichage des données géographiques**

Trois principales technologies sont disponibles pour la réalisation d'API Cartographique Web [2]:

- La première à avoir été utilisée est le Flex une technologie d'Adobe qui nécessite un plug-in dans les navigateurs web.
- Puis est apparue le Silverlight une technologie de Microsoft demandant également un plug-in pour fonctionner dans les navigateurs. Cette technologie a rencontré un franc succès aux Etats-Unis.
- Enfin, la dernière solution apparue est la technologie JavaScript, aujourd'hui c'est elle qui a la faveur de tous les développeurs. Mis en avant avec l'adoption du "Web 2.0" et du HTML5. Elle s'est démarquée et est désormais la seule technologie viable pour l'ensemble des développeurs.

Nous étudions ici deux interfaces de programmations (API) développées en JavaScript pour l'affichage des données géographiques: OpenLayer et Leaflet.

- **Leaflet**

Leaflet est une API Cartographique OpenSource développée en JavaScript par Vladimir Agafonkin et de nombreux contributeurs. Les sources étant disponibles sur GitHub (<https://github.com/Leaflet/Leaflet>).

Leaflet propose des fonctionnalités basiques et une première approche très simple pour les développeurs. Il est sous licence BSD 2.

- **OpenLayer**

OpenLayer est une API libre Open Source Geospatial Foundation (OSGeo) permet d'intégrer une carte dynamique dans une page web grâce à JavaScript. Il fournit un grand nombre de fonctions permettant de manipuler les cartes et d'ajouter du contenu. Il est également sous licence BSD 2.

Les deux API permettent d'accéder aux fonds de carte de l'IGN à travers l'obtention d'une clef de développement de l'API de Géoportail et aussi aux données issues d'OpenStreetMap, de CloudMade.

Pour le projet, nous utiliserons Leaflet pour sa facilité de prise en main.

L'analyse des différentes technologies existantes ont permis de mettre en place le schème ci-dessous.

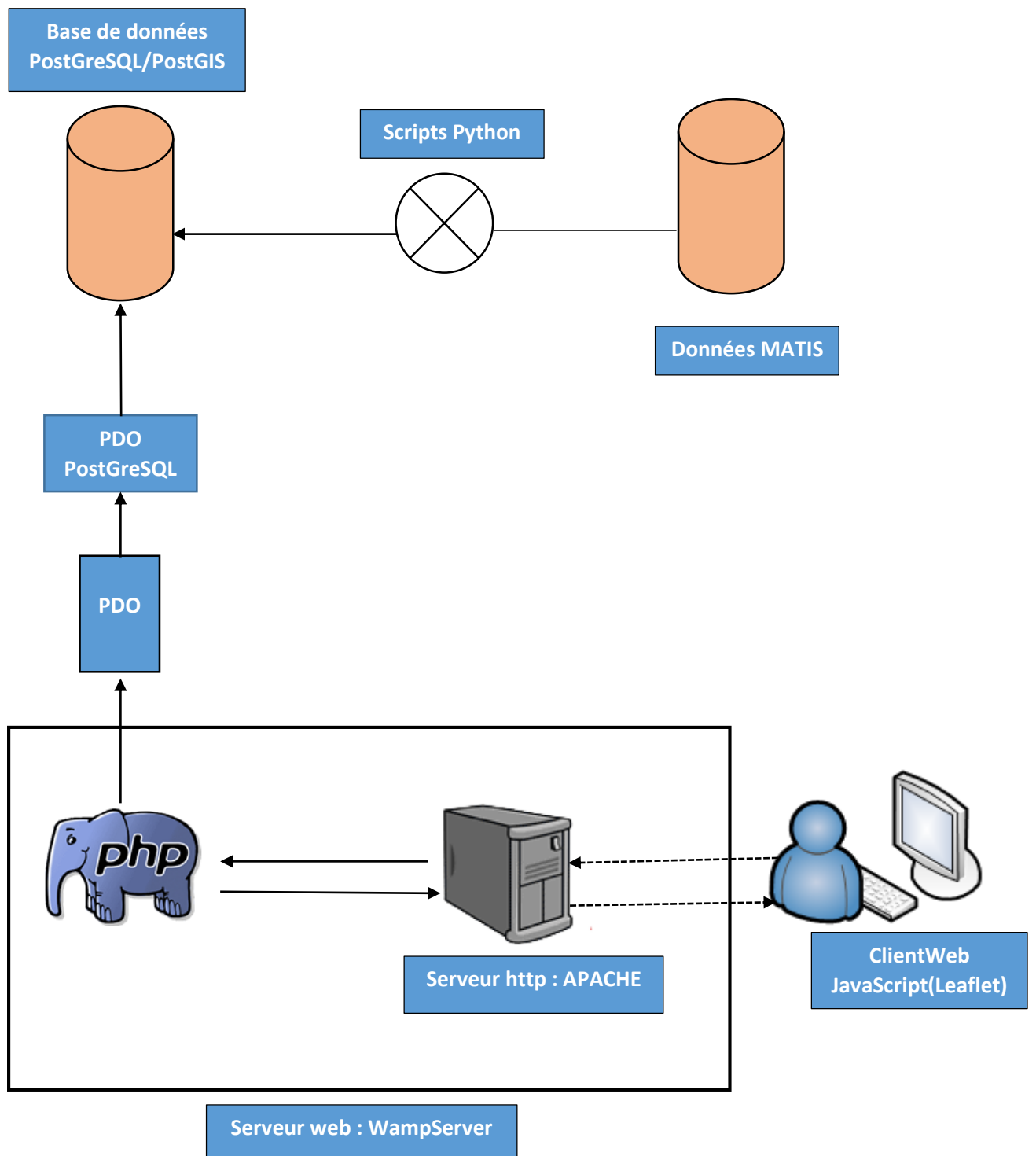


Figure 3: architecture générale de l'application

## II. Implémentation

### 1. Traitement des fichiers XML

- Ecriture de scripts en python pour l'extraction et le stockage des métadonnées des images issues de fichier XML

Les données à manipuler sont au format XML. Il s'agit d'un format standardisé, hiérarchique et manipulable par presque tous les langages de programmation et la majorité des logiciels. Cette partie est divisée en deux étapes.

Dans une première étape, nous « parsons » le document XML c'est-à-dire analyser les fichiers XML correspondant à chaque image et obtenir l'arbre correspondant en vue d'obtenir les attributs des différents nœuds. Ces nœuds serviront par la suite à caractériser les différentes tables dans la base de données. On utilisera la librairie **xml.etree.ElementTree**.

Avant l'analyse des différents fichiers XML, une configuration de notre environnement de travail est nécessaire pour un meilleur suivi des processus de traitement et de maintenance. Le fichier de configuration contient les différents répertoires pour l'accès aux données et également les paramètres de connexion à la base. Huit répertoires de stockage ont été créés : PLEIADE\_REPERTOIRE, ORTHO\_REPERTOIRE, SPOT\_REPERTOIRE, WORLDVIEW\_REPERTOIRE, RAPIDEYE\_REPERTOIRE, LANDSAT\_REPERTOIRE, LIDAR\_REPERTOIRE, TERRASARX\_REPERTOIRE.

Avec un algorithme de recherche récursive, on parcourt le répertoire concerné par le traitement pour analyser les fichiers XML de chaque image.

La seconde étape a pour objectif de charger les différents attributs extraits précédemment des différents répertoires de stockage. On a utilisé la librairie **PsycoPG2**. Différentes étapes sont indispensables :

- Ouverture de la connexion

On appelle alors le fichier de configuration.

- Exécution de la requête

Pour exécuter la requête, il faut un objet connexion et un curseur qui va permettre l'échange entre la base de données PostgreSQL/PostGIS.

- Récupération du résultat et validation pour le stockage en base
- Fermeture de connexion

Il faut fermer la connexion une fois, le traitement terminé pour libérer la mémoire.

On procédera ainsi pour tous les répertoires qu'on veut traiter.

### 2. Utilisation de la base de données avec PHP

- Serveur Web : Installation de WAMP server

On a installé WampServer 2.5 sur la machine pour travailler en local. Par défaut, WampServer s'installe et se configure avec le SGBDR MySQL.



- Langage PHP

Pour l'échange entre la base de données et le serveur, le langage PHP a été maintenu. L'extension PDO (*PHP Data Objects*) disponible sur PHP 5.1 est utilisée dans le cadre de de travail car il a l'avantage de travailler de façon unifiée, quel que soit le Système de Gestion de Base de Données (SGBD). Il permet aux développeurs de PHP de concevoir plus rapidement des connecteurs vers de nouvelles bases de données. La figure ci-dessous montre l'architecture des drivers.

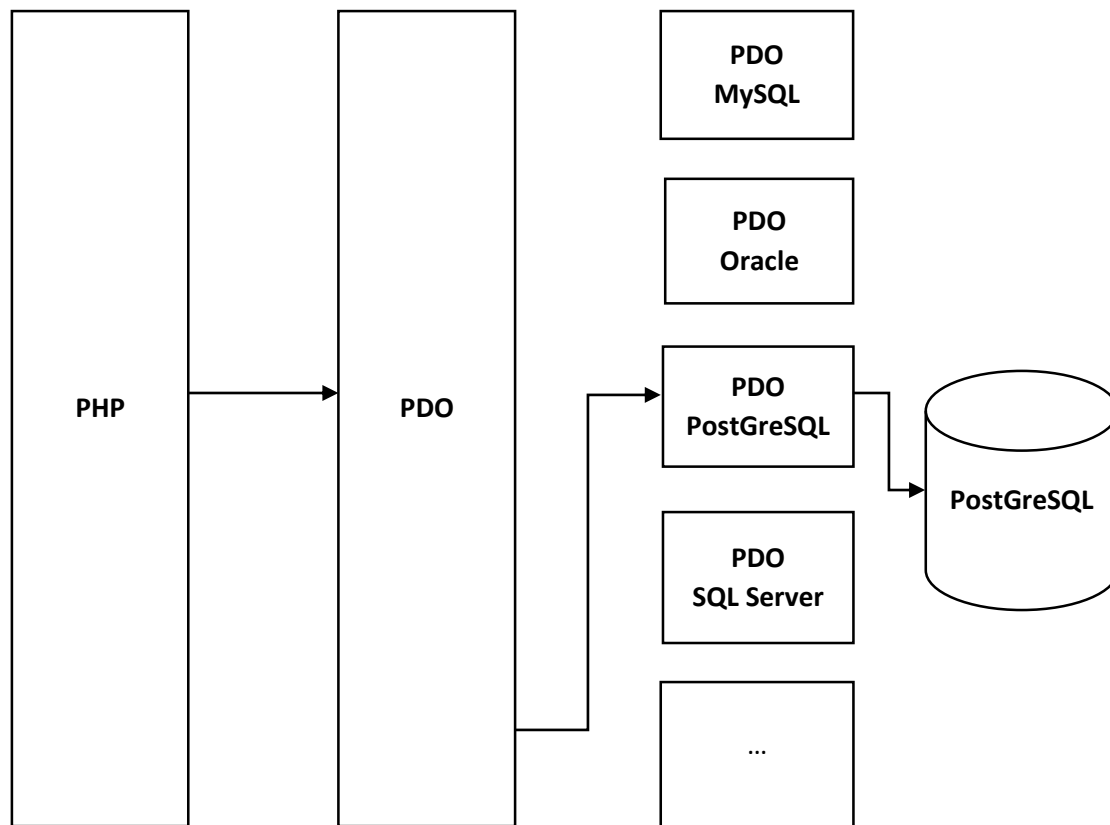


Figure 4: Architecture des drivers PDO (source : (Daspet & De Geyer, 2012))

L'utilisation de la base de données avec PHP s'effectue en cinq principales étapes (Daspet & De Geyer, 2012).

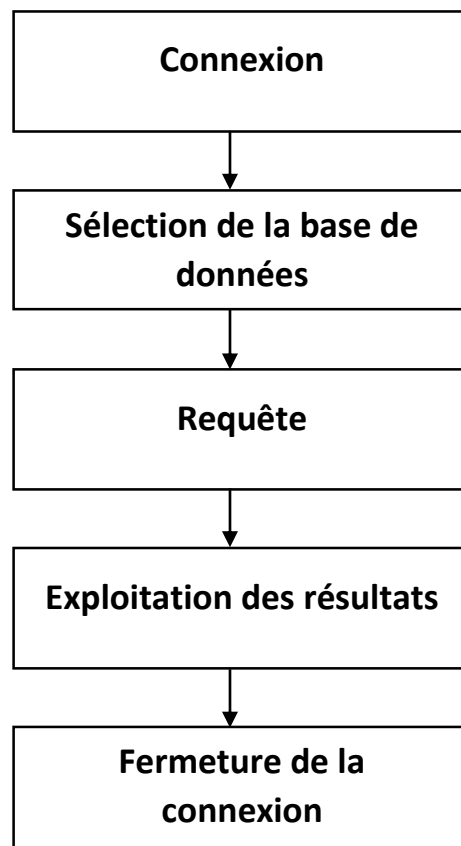


Figure 5: utilisation de PHP pour accéder à la base de données

- **Etape 1 et 2: Connexion au serveur de données**

Nous allons régulièrement nous connecter à la base de données de notre application, pour ne pas chaque fois saisir un mot de passe et un login, un fichier de configuration sera nécessaire. Le fichier va contenir les paramètres de connexion et l'instanciation de la classe PDO.

- **Etape 3: Requête**

Une fois, la connexion ouverte, on utilisera le langage SQL pour lire, modifier ou supprimer les données. Deux méthodes sont disponibles pour utiliser l'objet PDO : `exec()` et `query()`.

- Méthode `exec()` pour la définition des données (INSERT, UPDATE, DELETE).
- Méthode `query()` pour la manipulation des données (SELECT, EXPLAIN, SHOW).

Pour notre étude, nous utiliserons la méthode qui renvoie des résultats `query()`.

- **Etape 4: Exploitation des résultats**

Après l'exécution d'une requête de sélection, les données ne sont pas affichées, elles sont mises en mémoire. Il faut donc les récupérer et les afficher.

- **Etape 5: Fermeture de la base**

### 3. Solution côté client

#### ▪ Technologies utilisées

L'interface graphique pour la visualisation des données est une page web. Afin que la page soit dynamique, on a utilisé HTML et Javascript. Elle peut ainsi interroger la base de données. Leaflet a été utilisé pour l'affichage des fonds cartographiques de GeoPortail et OpenStreetMap.

Pour répondre au cahier de charge, une page web en PHP a été développée. Cette page permet de lister les images qui répondent aux requêtes faites par l'utilisateur en fonction des capteurs ayant acquis les données et leurs dates d'acquisition.

### III. Résultats

#### 1. Affichage des fonds cartographiques

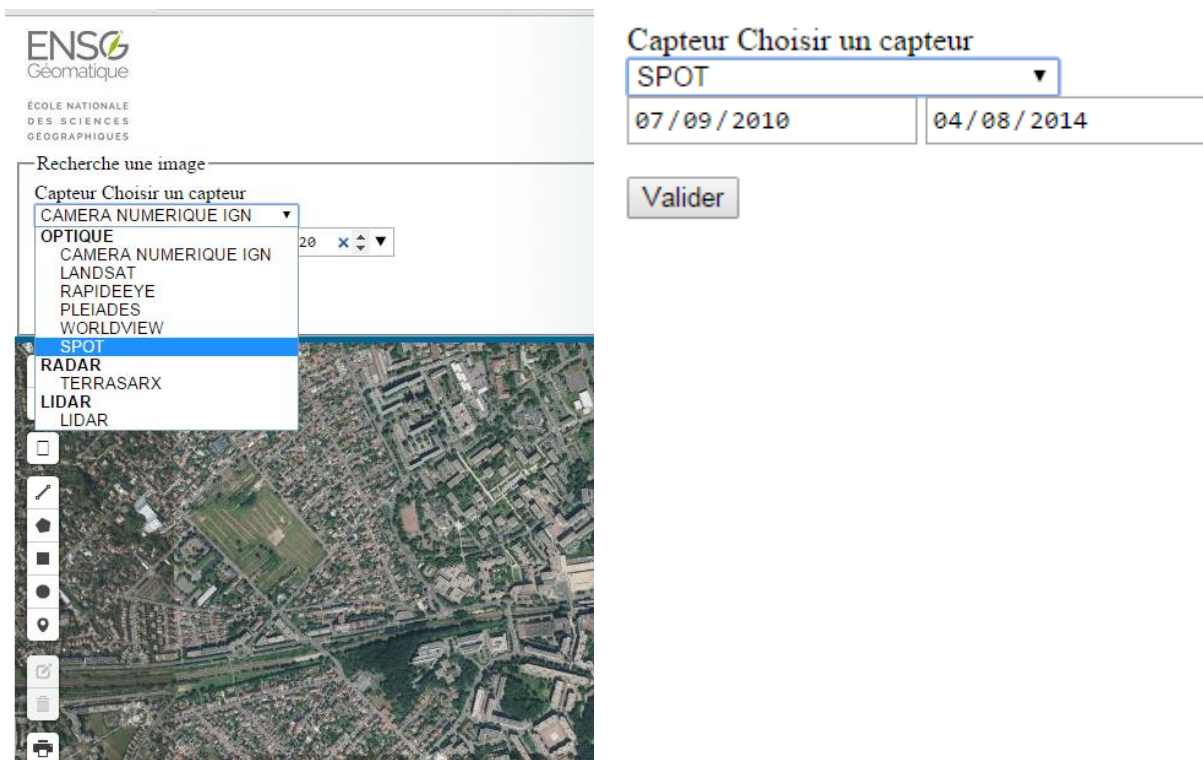


Figure 6: Affichage de fond de carte GeoPortail

#### 2. Affichage des images disponibles

##### ▪ Requêtes sémantiques

Nous effectuons un affichage des images disponibles sur notre serveur en fonction de la date d'acquisition et du capteur ayant acquis les données.



Caracteristiques des images																	
Id	Nom de l'image	Lieu d'acquisition	Format	Date acquisition	Niveau de traitement	Version de l'image	Nombre de ligne	Nombre de colonne	Système de représentation cartographique	Resolution spatiale en X (m)	Resolution spatiale en Y (m)	Resolution spectrale	Resolution radiométrique	Etat du satellite	Nombre de senseurs	Mode d'observation	Direction de visée
194			GeoTIFF	2010-09-20	Stereo 1B	22.2	6051	8820		0.574	0.574		16	1		FullSwath	Forward
195			GeoTIFF	2010-09-20	Stereo 1B	22.2	24204	35180		0.574	0.574		16	1		FullSwath	Forward

Figure 7: Aperçu de la page qui permet de lister les images disponibles en fonction des critères de sélection

## Conclusion et perspectives

Ce projet répond aux fonctionnalités principales du système, celles d'afficher les informations sur les images disponibles sur le serveur de données du MATIS en fonction d'un certain nombre de critères.

Sur le plan de compétences acquises, Il m'a permis de mettre en pratique les cours reçus au durant de la formation et d'acquérir des bases en développement informatique. La prise en main des différents outils informatiques (Python, PHP) a pris la majeure partie du temps ce qui n'a pas permis de remplir toutes les fonctionnalités souhaitées par le commanditaire.

En perspective, il faudrait intégrer l'affichage de l'emprise géographique de chaque image au niveau du client. Il faut aussi améliorer l'esthétique de l'interface du côté client.

## Références bibliographiques

### Livres:

Daspet, E., & De Geyer, C. (2012). *PHP 5 avancé*. Eyrolles.

DEBRAUNWER, L., & VAN DER HEYDE, F. (2008). *UML2: Initiation, exemples et exercices corrigés*.: ENI.

### Sites Internet

[1] : [http://www.memoireonline.com/07/08/1287/m\\_mise-en-place-d-une-plate-forme-de-cartographie-dynamique8.html](http://www.memoireonline.com/07/08/1287/m_mise-en-place-d-une-plate-forme-de-cartographie-dynamique8.html)

[2] : <http://igm.univ-mlv.fr/~dr/XPOSE2013/APICARTO/market.html>

<http://leafletjs.com/>

<http://damien.pobel.fr/post/cartographie-api-geoportail-ign-leaflet>

<https://docs.python.org/2/library/xml.etree.elementtree.html>

<http://initd.org/psycopg/docs/>

## A. Annexes : guide d'utilisateur

## 1. Présentation de l'outil

Cet outil permet de gérer les données disponibles au sein du laboratoire MATIS. La gestion concerne l'administration des données et leur visualisation. Pour réaliser ces fonctionnalités, l'utilisateur doit réaliser un certain nombre d'opérations.

### ■ Administration de données

Partie administration des données, l'outil a été développé sous Python. Chaque image doit posséder un fichier XML répondant aux normes W3C (World Wide Consortium).

#### - Etape 1 : Installation

Pour exécuter le script, il faut installer Python et importer les bibliothèques : `xml.etree.ElementTree` et `psycpg2`

#### - Etape 2 : Création du fichier config.ini

Dans ce fichier, on stocke le chemin d'accès aux répertoires des différentes données. On y intègre aussi les paramètres de connexion d'accès à la base de données pour ne pas chaque fois être amené les saisir. Il se présente ainsi dans notre cas :

[CONFIGURATION]

PLEIADE\_REPERTOIRE=E:\Cours\_ENSG\IT2\projetRecherche\Donnees\PLEIADES

ORTHOPHOTO\_REPERTOIRE=E:\Cours\_ENSG\IT2\projetRecherche\Donnees\ORTHOPHOTO

SPOT\_REPERTOIRE=E:\Cours\_ENSG\IT2\projetRecherche\Donnees\SPOT

WORLDVIEW\_REPERTOIRE=E:\Cours\_ENSG\IT2\projetRecherche\Donnees\Worldview

RAPIDEEYE\_REPERTOIRE=E:\Cours\_ENSG\IT2\projetRecherche\Donnees\RAPIDEEYE

LANDSAT\_REPERTOIRE=E:\Cours\_ENSG\IT2\projetRecherche\Donnees\Landsat

LIDAR\_REPERTOIRE=E:\Cours\_ENSG\IT2\projetRecherche\Donnees\LIDAR

TERRASAR\_REPERTOIRE=E:\Cours\_ENSG\IT2\projetRecherche\Donnees\TERRASAR

dbname=metadata

user=postgres

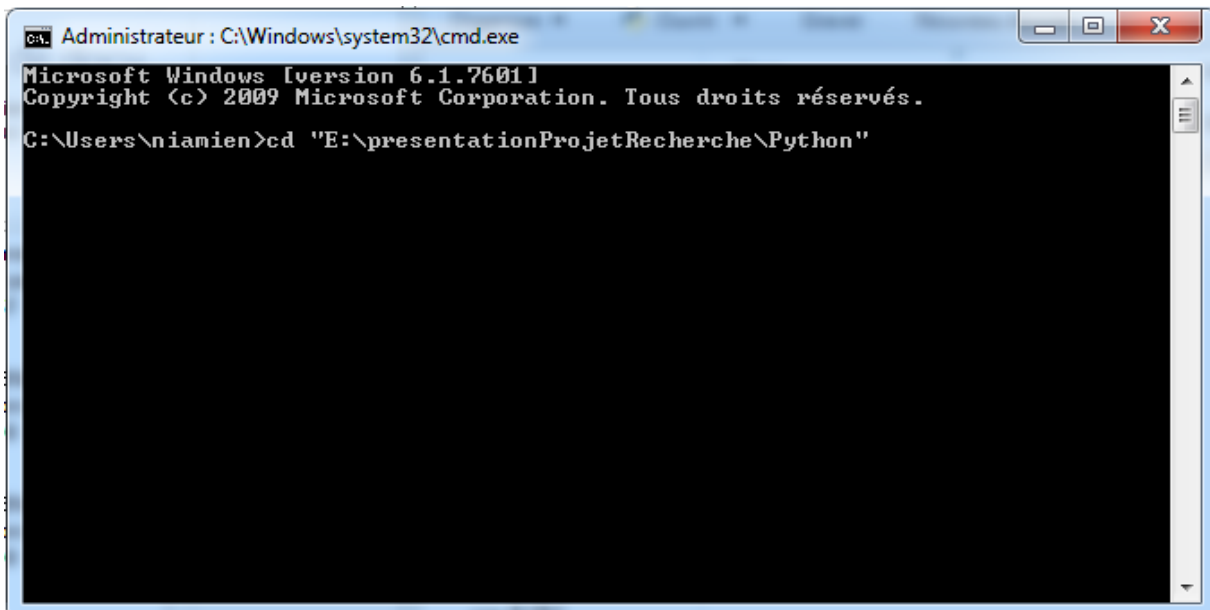
password=xxxx

host=localhost

port=5432

- Etape 3 : Lancement

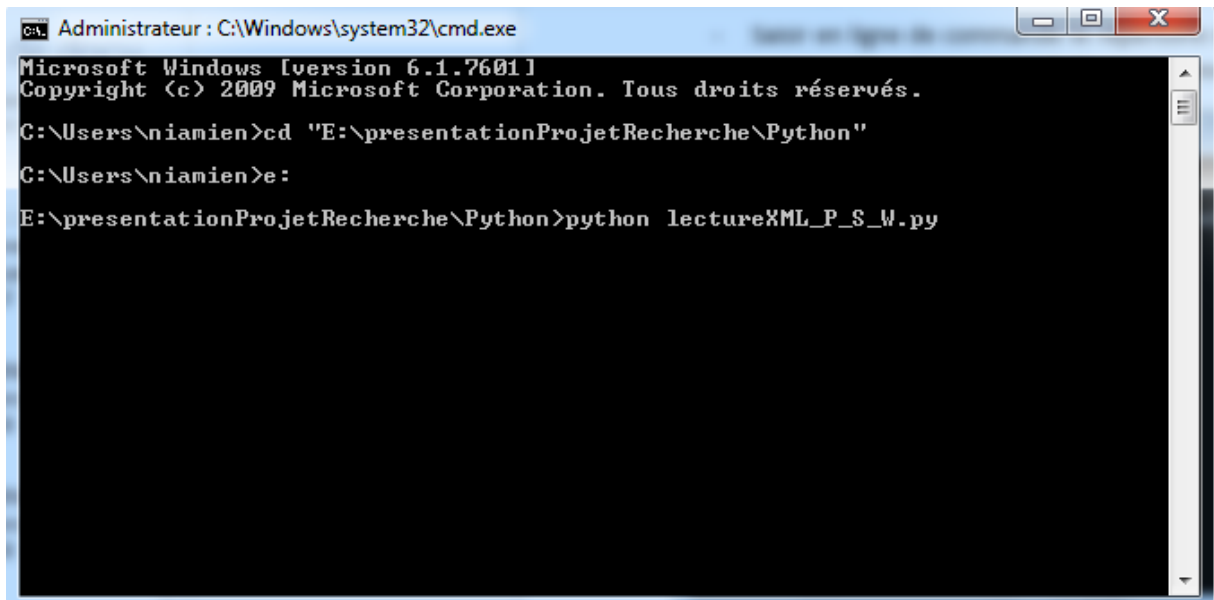
Saisir en ligne de commande le répertoire d'accès au fichier de configuration et aux scripts python. Notre script Python est nommé « *lectureXML\_P\_S\_W.py* »



```
Administrateur : C:\Windows\system32\cmd.exe
Microsoft Windows [version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. Tous droits réservés.
C:\Users\niamien>cd "E:\presentationProjetRecherche\Python"
```

- Etape 4 : Appeler le script Python

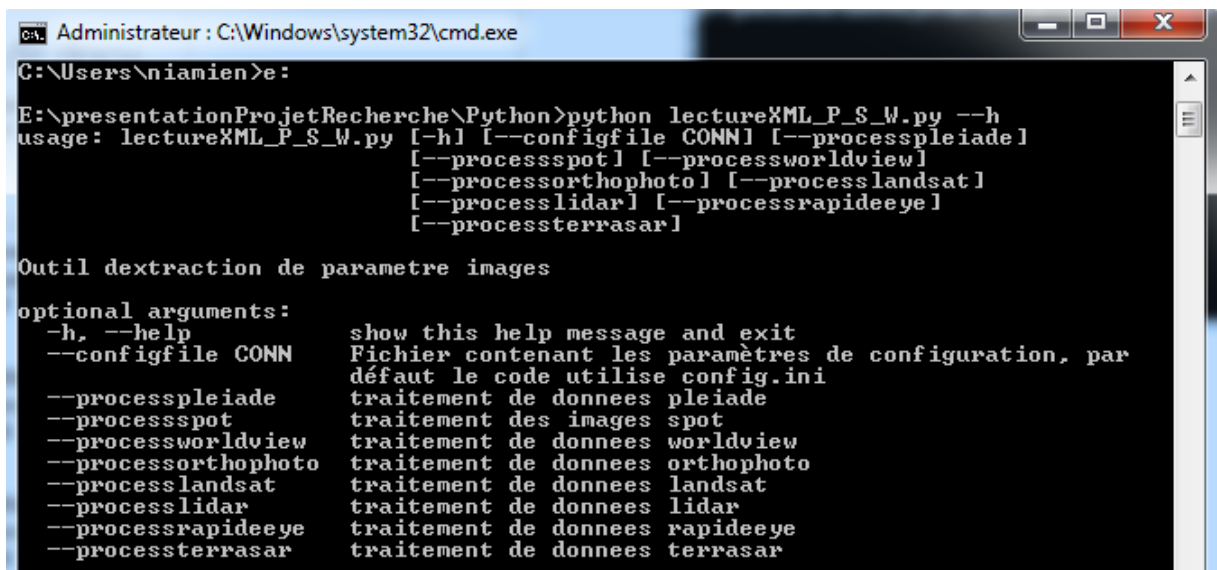
Notre script Python est nommé « *lectureXML\_P\_S\_W.py* »



```
Administrateur : C:\Windows\system32\cmd.exe
Microsoft Windows [version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. Tous droits réservés.
C:\Users\niamien>cd "E:\presentationProjetRecherche\Python"
C:\Users\niamien>e:
E:\presentationProjetRecherche\Python>python lectureXML_P_S_W.py
```

- Saisir -- h pour accéder à l'aide

Cette aide vous décrit les différents traitements possibles

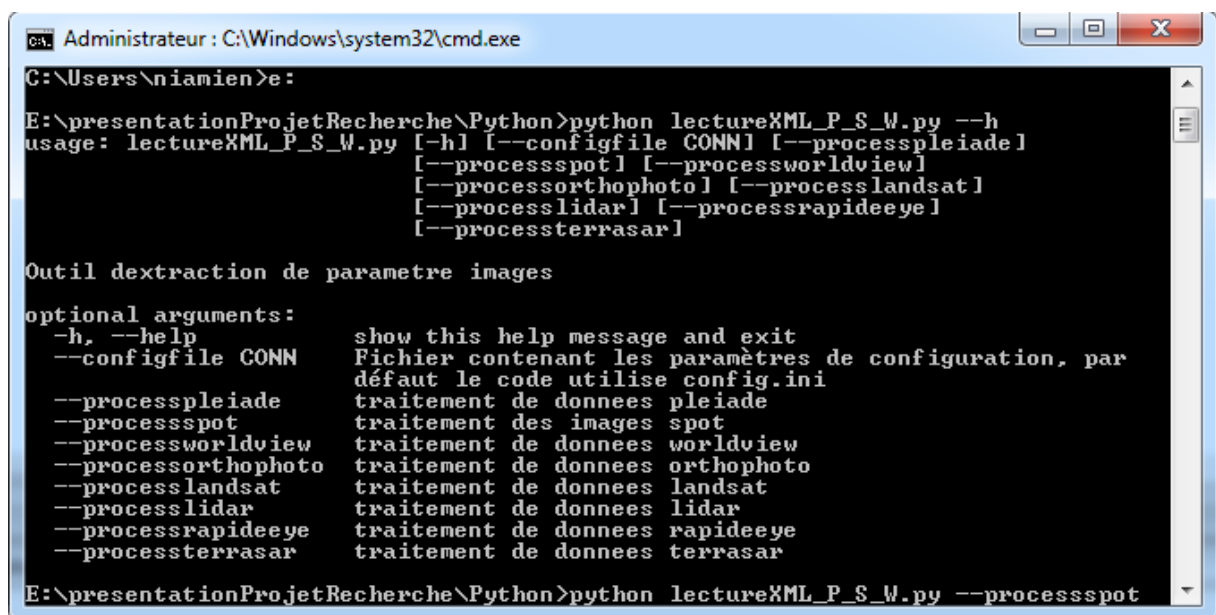


```
C:\Users\niamien>E:\presentationProjetRecherche\Python>python lectureXML_P_S_W.py --h
usage: lectureXML_P_S_W.py [-h] [--configfile CONN] [--processpleiade]
                           [--processspot] [--processworldview]
                           [--processorthophoto] [--processlandsat]
                           [--processlidar] [--processrapideeye]
                           [--processterrasar]

Outil d'extraction de paramètres images

optional arguments:
  -h, --help            show this help message and exit
  --configfile CONN     Fichier contenant les paramètres de configuration, par
                        défaut le code utilise config.ini
  --processpleiade       traitement de données pleiade
  --processspot          traitement des images spot
  --processworldview     traitement de données worldview
  --processorthophoto    traitement de données orthophoto
  --processlandsat       traitement de données landsat
  --processlidar         traitement de données lidar
  --processrapideeye     traitement de données rapideeye
  --processterrasar      traitement de données terrasars
```

- Choisir le capteur à traiter  
Nous avons décidé dans l'exemple de traiter le capteur SPOT



```
C:\Users\niamien>E:\presentationProjetRecherche\Python>python lectureXML_P_S_W.py --h
usage: lectureXML_P_S_W.py [-h] [--configfile CONN] [--processpleiade]
                           [--processspot] [--processworldview]
                           [--processorthophoto] [--processlandsat]
                           [--processlidar] [--processrapideeye]
                           [--processterrasar]

Outil d'extraction de paramètres images

optional arguments:
  -h, --help            show this help message and exit
  --configfile CONN     Fichier contenant les paramètres de configuration, par
                        défaut le code utilise config.ini
  --processpleiade       traitement de données pleiade
  --processspot          traitement des images spot
  --processworldview     traitement de données worldview
  --processorthophoto    traitement de données orthophoto
  --processlandsat       traitement de données landsat
  --processlidar         traitement de données lidar
  --processrapideeye     traitement de données rapideeye
  --processterrasar      traitement de données terrasars

E:\presentationProjetRecherche\Python>python lectureXML_P_S_W.py --processspot
```

- Affichage du résultat à l'écran  
Nous montrons ici les résultats pour vous aider à comprendre le traitement. Dans la version finale, elle ne sera plus visible. Vous y accéderez en vous connectant à la base de données qui a été créée.
- Accéder à la base de données PostgreSQL pour accéder aux attributs extraits.



- Administration du serveur

- Etape 1 : Création d'un fichier de configuration

On crée un fichier de configuration en PHP pour y stocker tous les paramètres de configuration. Ce fichier à cette forme :

```
<?php

define("DB_HOST", "localhost");

define("DB_NAME", "metadata");

define("DB_USER", "postgres");

define("DB_PASSWORD", "xxxx");

define("DB_PORT", "5432");

define("DSN",
"pgsql:dbname=".DB_NAME.";host=".DB_HOST.";user=".DB_USER.";password=".DB_PASSW
ORD.";port=".DB_PORT);

//Connexion

try
{
    $PgsqConnection = new PDO(DSN);
}

catch(PDOException $e)
{
    echo "Erreur Connexion : ".$e->getMessage();
    die();
}

?>
```

- Etape 2 : Création d'un fichier de fonctions en PHP

Ce fichier contient les requêtes qu'on veut envoyer au serveur.

- Etape 3: Création de la page web en PHP

Ce fichier contient notre page web.

- Client web léger

Le client va accéder aux informations liées aux données en effectuant une requête en fonction du capteur et de la date d'acquisition des images.

The screenshot shows the ENSG (Ecole Nationale des Sciences Géographiques) web interface. On the left, there is a sidebar with the ENSG logo and a search section titled 'Recherche une image'. This section includes a 'Capteur Choisir un capteur' dropdown menu with a list of options: CAMERA NUMERIQUE IGN, OPTIQUE, LANDSAT, RAPIDEYE, PLEIADES, WORLDVIEW, SPOT (highlighted), RADAR, TERRASARX, and LIDAR. Below the menu is a map of a city area. On the right, there is a search form with the same 'Capteur Choisir un capteur' dropdown menu set to 'SPOT'. Below the dropdown are two date input fields: '07/09/2010' and '04/08/2014'. A 'Valider' button is located below the date fields.

ENSG  
Géomatique  
ÉCOLE NATIONALE  
DES SCIENCES  
GÉOGRAPHIQUES

Recherche une image

Capteur Choisir un capteur

CAMERA NUMERIQUE IGN  
OPTIQUE  
LANDSAT  
RAPIDEYE  
PLEIADES  
WORLDVIEW  
SPOT  
RADAR  
TERRASARX  
LIDAR

07/09/2010 04/08/2014

Valider