

# Prodédures de stockage et d'organisation des données dans une base de données

---

*Deliverable number : D6.1*

*Nature:P*

*Contractual Date of Delivery: 14 November 1998*

*Task WP1.2 : Acquisition, stockage et pré-traitement des données*

*Nom du rédacteur Mouhamed Tidiane SECK*

*Alex CORENTIN*

*Institut*

*ESP-DAKAR*

*Adress*

*Dakar*

[seckm@ucad.sn](mailto:seckm@ucad.sn)

[Corenthin@ucad.sn](mailto:Corenthin@ucad.sn)

## Abstract:

Le présent rapport établit les différents mécanismes qui seront mis en place afin :

- D'établir des liens de transfert ou d'extraction avec les bases de données existantes contenant les données nécessaires aux indicateurs pertinents du projet, telles que décrites dans le livrable « Description fonctionnelle des données stockées », tout en tenant compte de l'hétérogénéité des bases de données et de l'évolutivité du système.
- D'organiser et de stocker ces données dans une base de données centralisée dont la copie maîtresse sera installée à l' E.S.P. de Dakar. Cette base de données, couplée à un serveur WEB permettra d'accéder aux informations à travers Internet. L'objectif du projet est de créer une interface WEB adaptée au type d'utilisateur et à la nature des requêtes soumises au système. Pour y parvenir, nous nous appuierons sur les travaux effectués par les autres équipes spécialisées notamment en matière de traitement d'images, systèmes multi-agents etc...
- De spécifier et de tester sur un cas concret un prototype permettant d'illustrer quelques fonctionnalités de base du futur système.

The present report establishes the various schemes which will be implemented :

- To define transfer and extraction links with the existing Data Bases containing the data related to the relevant indicators, such as the ones described in the deliverable « Functional description of the stored data », by taking into account the heterogeneousness of Data Bases and the evolutivity of the system.
- To organize and store these data in a centralized Data Base whose master copy will be installed at ESP dakar. This Data Base, linked to a Web server will allow to access to the data via Internet. The goal of the project is to create a Web interface which suits to potential expected users and requests. To achieve this goal, we will rely on work done by other teams in advanced data processing (Image processing, multi-agent systems...).
- To specify and to test on an actual case a prototype including some key functionalities of the forthcoming system.



# PROCEDURES DE STOCKAGE ET D'ORGANISATION DES DONNEES DANS UNE BASE DE DONNEES

## Contenu

|   |           |
|---|-----------|
| <b>AVANT PROPOS</b> .....   | <b>5</b>  |
| <b>1 - IMPORTATION DES DONNÉES EXISTANTES</b> .....                                 | <b>5</b>  |
| <b>2- STOCKAGE ET ORGANISATION DES DONNÉES</b> .....                                | <b>5</b>  |
| <b>3 - SCHÉMA DE PRINCIPE DU FUTUR SYSTÈME.</b> .....                               | <b>7</b>  |
| <b>4- CRITÈRES DE CHOIX DE LA PLATE-FORME LOGICIELLE.</b> .....                     | <b>10</b> |
| <b>5 - TYPES D'APPLICATION</b> .....  | <b>15</b> |
| <b>6 - DIFFÉRENTES ARCHITECTURES DE COUPLAGE BASES DE DONNÉES / WEB</b> .....       | <b>16</b> |
| <b>8 - ETUDE DE CAS</b> .....   | <b>19</b> |
| SOURCE DE LA PAGE HTML.....   | 20        |
| SOURCE DU FICHER DE CONNEXION BASE DE DONNÉES (IDC ASSOCIÉ À LA PAGE RÉSULTAT)..... | 23        |
| <b>9 - CONCLUSION</b> .....   | <b>25</b> |



## Avant propos

Pour l'essentiel, le projet SIMES puise ses données dans des sources diverses créées et maintenues par des structures autonomes. Ces données sont stockées dans des bases de données ou sont disponibles sous format numérique.

### 1 - Importation des données existantes

Le système doit prendre en compte l'hétérogénéité des sources de données, non seulement par rapport au type de donnée (fichiers texte, images cartographiques, etc.) mais aussi, par rapport au type de SGBD (Access, Oracle, Foxpro etc.).

Il s'agit, dans un premier temps, de choisir un SGBD pivot capable de stocker tous les types de données identifiés (alphanumériques, images, animations etc...). Ensuite on spécifie les procédures d'extraction d'informations en fonction des bases de données sources.

Pour cela :

- Les données de base doivent être disponibles au niveau du site central de l'ESP-Dakar. La situation optimale correspondrait à celle où toutes les bases de données seraient accessibles par Internet. Dans ces conditions, on pourrait envisager de véhiculer certaines requêtes vers des bases de données distantes ou d'effectuer des traitements sur des sites distants particulièrement adaptés. Cette solution est loin de pouvoir être réalisable à court terme car l'état actuel de la connectivité IP des sites participants au projet SIMES ne permet pas d'atteindre les performances requises pour une telle approche. Toutefois, il est néanmoins nécessaire d'avoir en ligne le plus grand nombre de bases de données.
- Les transferts de données de la source vers le site centralisé seront réalisés en utilisant des techniques standard :
  1. FTP pour les données distantes.
  2. Techniques d'importation pour les bases de données sources offrant une interface avec le SGBD pivot.
  3. Lien ODBC pour les bases de données sources disposant du driver associé avec le SGBD pivot.

Il faut noter l'existence de drivers ODBC entre tous les SGBD standard du marché.

En ce qui concerne les données qui ne sont pas en ligne, le transfert des informations vers la base de données centralisée à l'ESP-Dakar se fera à l'aide de disquettes, bandes, ou CDROM.

### 2- Stockage et organisation des données

Dans cette présentation nous partons de l'hypothèse que certaines données disponibles sur des sites en ligne ne sont pas dupliquées sur le site maître du projet.

L'ensemble des données disponibles devra être organisé de sorte à optimiser le temps de réponse aux requêtes de l'utilisateur.

Seront alors stockés sur le site centralisé :

- Les modèles de requêtes prédéfinis fournissant les résultats les plus représentatifs des données et de leurs traitements. Cela servirait par ailleurs de guide sur les potentialités du système aux utilisateurs désirant formuler des requêtes spécifiques.
- Les données de grande taille, comme les images fréquemment demandées par des requêtes différentes, seraient dupliquées sur le site central.
- Enfin, toutes les données dont les sources ne sont pas en ligne sur Internet.

Pour assurer la validité des résultats, il sera nécessaire de définir une fréquence de rafraîchissement des informations. En effet, toutes les données pertinentes du projet ne sont pas totalement centralisées, cela nécessiterait des grands moyens matériels (grande capacité de stockage) et des procédures lourdes pour assurer les mises à jour à partir des données réelles produites par les sites distants.

### 3 - Schéma de principe du futur système

L'accès aux informations du projet SIMES se fera à travers le réseau Internet via l'interface WEB.

Le modèle global de cette interface distingue trois niveaux :

- Le client WEB supporté par un navigateur standard (Internet explorer ou Netscape etc..).
- Le serveur WEB gérant les pages HTML
- La base de données contenant les données du système d'information.

Entre ces trois composantes on fait intervenir différentes passerelles et protocoles qui seront étudiés dans la suite du document.

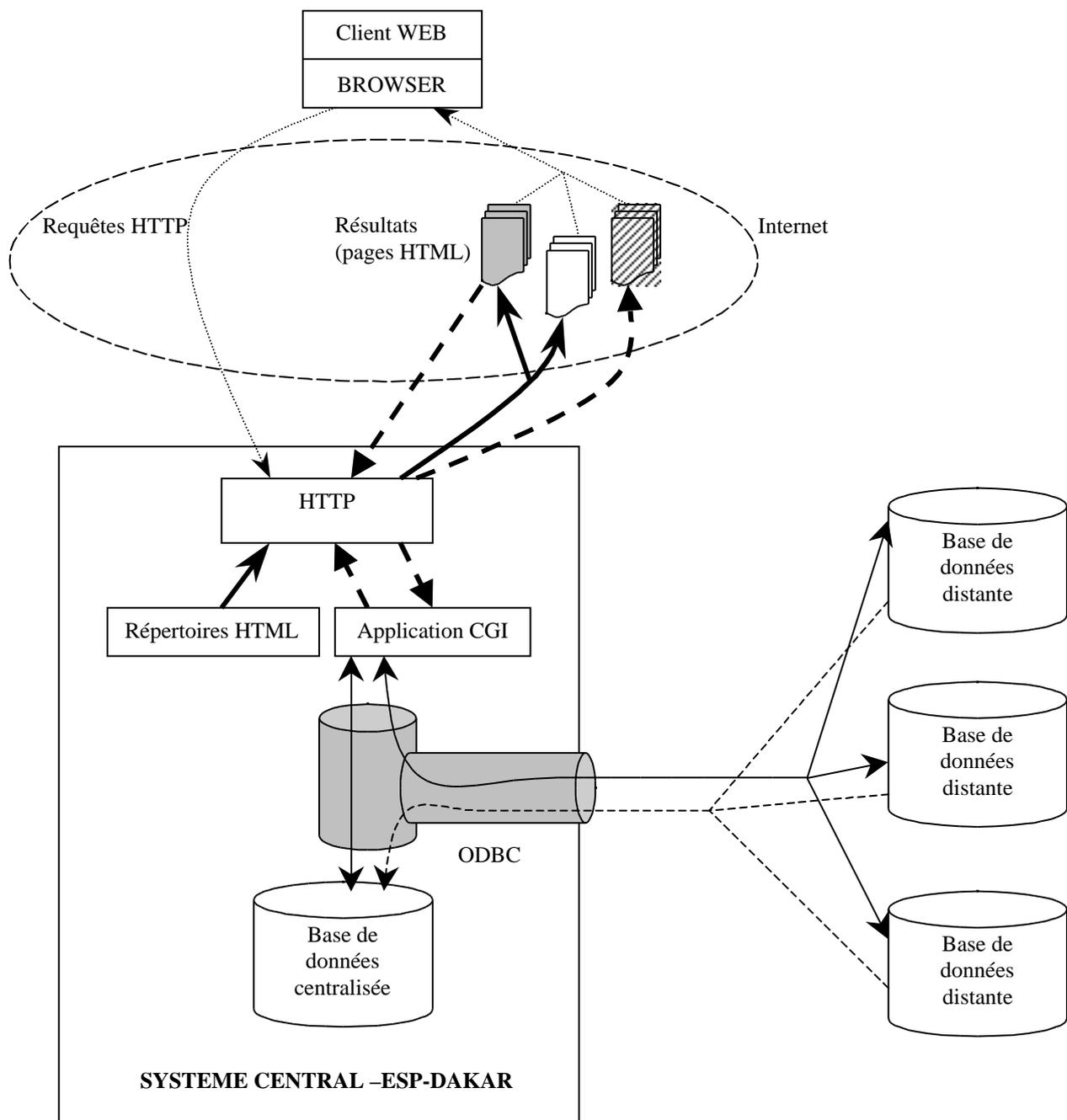
De façon classique c'est le protocole HTTP qui sert de passerelle entre le client et le serveur WEB, ce dernier joue deux rôles essentiels à savoir, d'une part, stocker les pages HTML en vue de les fournir à la demande aux clients WEB, et, d'autre part, appeler des applications via l'interface CGI pour générer des pages HTML dynamiques vers les clients WEB (cas des formulaires). C'est en particulier à travers l'interface CGI que le serveur WEB peut encapsuler des requêtes (SQL) vers les bases de données et générer comme précédemment des pages HTML dynamiques contenant les résultats souhaités.

Par exemple, pour une application CGI qui gère le traitement des requêtes via ODBC on distingue les étapes suivantes :

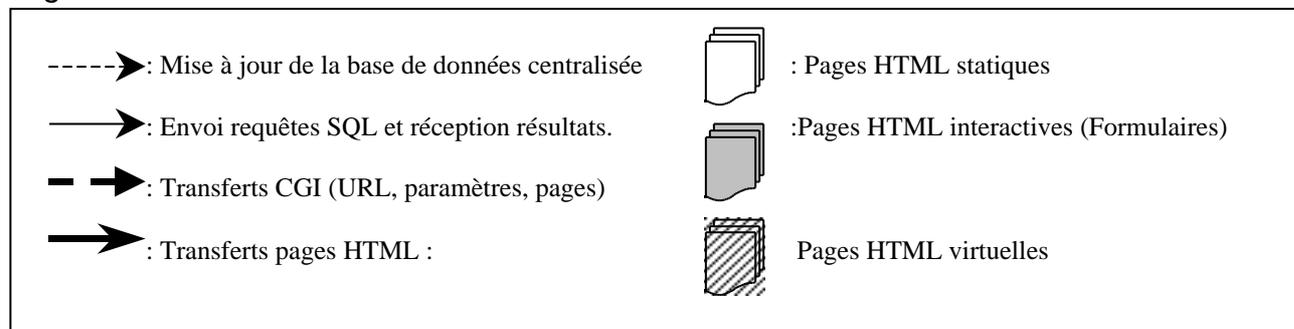
- Choix de la source de la donnée
- Connexion à la source
- Envoi de la requête sous forme d'instructions SQL
- Réception et traitement des résultats éventuels de la requête et/ou des erreurs obtenues.
- Validation ou annulation de la transaction
- Déconnexion de la base de données.

La figure 1 ci-dessous donne un aperçu global du système proposé.

On notera la présence de deux canaux ODBC, reliant l'application CGI avec la base de données centralisée d'une part et l'ensemble des bases de données sectorielles d'autre part.



**Légende**



**Fig.1 : Système futur : Organisation des données réparties.**

Le système présenté considère que la mise à jour de la base de données centralisée se fera via une liaison ODBC. Bien entendu, d'autres possibilités existent dont en particulier l'importation de données préalablement transférées sur le site central via FTP.

Les utilisateurs envoient leurs requêtes au serveur web en utilisant leur navigateur préféré, sur lequel seront également affichés les résultats de ces requêtes. Suivant la nature de la requête, le protocole HTTP pourra fournir à l'utilisateur les résultats suivants :

- Une page statique
- Un formulaire, dont les zones de saisie seront renseignées par l'utilisateur, informations qui seront transmises comme paramètres à l'application CGI.
- Une page virtuelle générée par l'application CGI.

Le serveur Web de l'ESP-Dakar aura donc à gérer trois types de pages HTML :

- Les pages statiques, par exemple la page d'accueil du site SIMES. Leur mise à jour ne peut se faire que par modification du fichier HTML correspondant.
- Les pages interactives, ou formulaires, qui permettent la saisie (dans des zones dédiées à cet effet) des paramètres de la requête de l'utilisateur. Ces paramètres seront par la suite traités par l'application CGI et les résultats seront recherchés dans les bases de données via ODBC.
- Les pages virtuelles, qui sont créées par l'application CGI pour répondre à la requête de l'utilisateur. Il s'agit d'un fichier HTML qui servira de modèle pour l'affichage des résultats de la requête. Cette application CGI est appelée par un utilisateur à travers une requête.

Evidemment, le serveur Web peut inclure des pages HTML hybrides, qui partagent les caractéristiques des pages décrites plus haut. Par exemple, à partir de la page virtuelle représentant le résultat de la demande d'une carte hydrographique de l'Afrique, on pourrait saisir comme paramètres les coordonnées de la région sur laquelle on veut un zoom, la page servant ainsi de formulaire. Ces paramètres pourront par la suite être utilisés par un script CGI afin d'effectuer le zoom de la zone demandée, puis d'afficher le résultat à partir d'une page virtuelle.

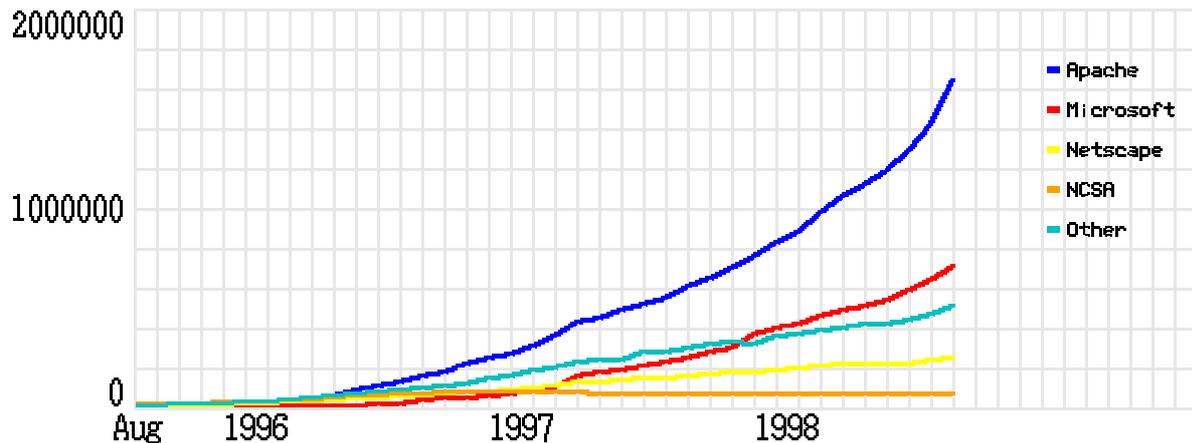
Il serait aussi intéressant de produire des résultats de requêtes en fonction du profil de l'utilisateur. Les avantages de ces pages dynamiques ne sont pas négligeables : outre une meilleure convivialité, les ressources du serveur sont optimisées car certaines opérations comme des mises en forme ou des tris sont alors totalement exécutées sur l'ordinateur du client. Le serveur web se trouverait ainsi libéré de certaines tâches encombrantes et/ou répétitives.

Sur tous ces aspects concernant les pages HTML, il reste à résoudre le problème du maintien de leur cohérence. Les investigations que nous avons déjà effectuées sur le plan bibliographique nous conduisent à penser que ce thème peut constituer un des thèmes de recherche que nous pouvons prendre en charge.

## 4- Critères de choix de la plate-forme logicielle

Chaque partenaire du projet SIMES a bien évidemment acquis une expérience sur une plate-forme particulière.

Il serait ainsi illusoire de mettre tout le monde sous le même moule. Néanmoins, il nous a paru utile de donner quelques indications sur les tendances au niveau international.



*Fig 2 : Croissance des serveurs Web d'Août 1995 à Septembre 1998 (en nombre de serveurs) Source :Netcraft WebServer Survey <http://www.netcraft.co.uk/survey/>*

Nous nous sommes référés à l'étude du Netcraft WebServer Survey qui prend en considération plusieurs critères comme la rapidité et la sécurité du serveur. Dans cet extrait nous avons retenu les serveurs parmi les plus connus du marché. L'enquête porte sur plus de trois millions de serveurs.

L'analyse de cette courbe montre que le serveur Apache est le plus utilisé, suivi de celui de Microsoft IIS fourni avec Windows NT et en troisième position la suite Netscape. Tous les autres, dont Oracle, sont regroupés dans « Other ».

Un premier choix aurait donc penché vers l'un de ces trois serveurs. Mais nous devons prendre en compte l'utilisation massive des bases de données parallèlement à l'utilisation des pages Web, seul domaine pris en compte dans l'enquête citée plus haut.

Dans ces conditions, la solution Oracle intégrant à la fois un serveur WEB et un SGBD performant ou des solutions comme O2 qui fournissent des solutions intégrées basées sur l'approche objet, ne sont pas à écarter.

Le tableau ci-dessous établit un comparatif de quatre serveurs WEB candidats. Les indications portent essentiellement sur les caractéristiques générales, les prix, les systèmes d'exploitations supportés, la sécurité, l'ouverture aux interfaces standard.

|                                    | <b>ORACLE</b>   | <b>APACHE</b>  | <b>MICROSOFT</b>   | <b>NETSCAPE</b>   |
|------------------------------------|---|--|--|---|
| <b>Nom du Serveur</b>              | Oracle Web Application Server   | Apache   | Internet Information Server  | Netscape Server   |
| <b>Version</b>                     | 3.01  | 1.3  | 4.0  | 3.5.1   |
| <b>Vendeur</b>                     | Oracle Corp.  | The Apache Group   | Microsoft Corp.  | Netscape Communications Corp  |
| <b>Meilleures caractéristiques</b> | Environnement pour le développement d'applications  | Rapide, support public pour le développement   | Pages ASP, compatible Microsoft APIs et driver ODBC  | Java run-time (JDK 1.1).<br>Convertit PDF en HTML.<br>Compatible LDAP, Oracle et Informix   |
| <b>Prix</b>                        | appeler Oracle  | Gratuit  | Gratuit avec NT 4.0 option pack  | \$1,295   |
| <b>Système d'exploitation</b>      | HPUX<br>Windows NT<br>Windows 95<br>Solaris   | NetBSD<br>Digital UNIX<br>BSDI<br>AIX<br>OS/2<br>SCO<br>HPUX<br>Windows NT<br>Linux<br>FreeBSD<br>IRIX<br>Solaris  | Windows NT   | Digital UNIX<br>AIX<br>HPUX<br>Windows NT<br>IRIX   |
| <b>Démarrage et connexion</b>      | <ul style="list-style-type: none"> <li>- Peut écrire des connexions multiples</li> <li>- Les fichiers de connexion peuvent être automatiquement recyclés ou archivés</li> <li>- Le serveur peut générer des commentaires</li> <li>- Connexions de mesure de performances</li> <li>- Les scripts CGI peuvent créer leur propre connexion CERN/NCSA</li> <li>- Format de connexion commun</li> <li>- S'exécute comme un service et/ou application Windows NT</li> <li>- Peut s'exécuter à partir de inetd (pour les systèmes Unix et OS/2)</li> <li>- Peut écouter à</li> </ul> | <ul style="list-style-type: none"> <li>- Peut écrire des connexions multiples</li> <li>- Les fichiers de connexion peuvent être automatiquement recyclés ou archivés.</li> <li>- Le serveur peut générer des commentaires</li> <li>Les scripts CGI peuvent créer leur propre connexion</li> <li>Peut servir des répertoires racine différents pour des adresses IP différentes</li> <li>CERN/NCSA</li> <li>- Format de connexion commun</li> <li>- S'exécute comme un service et/ou application Windows NT</li> <li>- Peut s'exécuter à partir de inetd (pour les systèmes Unix</li> </ul> | <ul style="list-style-type: none"> <li>- Peut écrire des connexions multiples</li> <li>- Les fichiers de connexion peuvent être automatiquement recyclés ou archivés.</li> <li>- Le serveur peut générer des commentaires.</li> <li>Les scripts CGI peuvent créer leur propre connexion</li> <li>Peut servir des répertoires racine différents pour des adresses IP différentes</li> <li>CERN/NCSA</li> <li>Format de connexion commun</li> <li>S'exécute comme un service et/ou application Windows NT</li> <li>Peut écouter à des adresses et ports multiples</li> </ul> | <ul style="list-style-type: none"> <li>- Peut écrire des connexions multiples</li> <li>Les fichiers de connexion peuvent être automatiquement recyclés ou archivés</li> <li>- Le serveur peut générer des commentaires</li> <li>- Connexions de mesure de performance</li> <li>Les scripts CGI peuvent créer leur propre connexion</li> <li>Peut servir des répertoires racine différents pour des adresses IP différentes</li> <li>CERN/NCSA</li> <li>Format de connexion commun</li> <li>S'exécute comme un service et/ou application Windows NT</li> <li>Peut s'exécuter à partir de inetd (pour les systèmes Unix et</li> </ul> |

|                 |  |   |  |   |
|-----------------|--|---|--|---|
|                 | <p>des adresses et ports multiples</p> <ul style="list-style-type: none"> <li>- Les connexions peuvent être personnalisées</li> <li>- Peut chercher un utilisateur dans une connexion</li> <li>- Connexion avec syslog (Unix) ou Event Log (Windows NT)</li> <li>- Peut générer des connexions pour des browsers</li> </ul>  | <p>et OS/2)</p> <ul style="list-style-type: none"> <li>- Peut écouter à des adresses et ports multiples</li> <li>- Les connexions peuvent être personnalisées</li> <li>- Connexion avec syslog (Unix) ou Event Log (Windows NT)</li> <li>- Peut générer des connexions pour des browsers</li> </ul>   | <ul style="list-style-type: none"> <li>- Les connexions peuvent être personnalisées.</li> <li>- Connexion avec syslog (Unix) ou Event Log (Windows NT)</li> <li>- Peut générer des connexions pour des browsers</li> </ul>   | <p>OS/2)</p> <ul style="list-style-type: none"> <li>- Peut écouter à des adresses et ports multiples</li> <li>- Les connexions peuvent être personnalisées</li> <li>- Peut chercher un utilisateur dans une connexion</li> <li>- Connexion avec syslog (Unix) ou Event Log (Windows NT)</li> <li>- Peut générer des connexions pour des browsers</li> </ul>   |
| <b>Sécurité</b> | <ul style="list-style-type: none"> <li>- Serveur certifié intégré</li> <li>- Interdit l'accès par le nom du domaine</li> <li>- Exécution CGI par UID</li> <li>- Interdit l'accès par adresse IP</li> <li>- Interdit l'accès par utilisateur et groupe</li> <li>- Compatible S-HTTP</li> <li>- Peut changer la liste de contrôle d'accès de l'utilisateur sans redémarrer le serveur</li> <li>- Permissions hiérarchiques pour les documents basés sur le répertoire</li> <li>- Interdit l'accès par répertoire et fichier</li> <li>- Groupes utilisateurs configurables (pas seulement une liste d'utilisateur unique)</li> <li>- Compatible PCT, SSL v. 2, SSL v. 3, Set</li> <li>- Peut nécessiter mot de passe</li> <li>- Les règles de sécurité peuvent se baser d'URLs</li> </ul> | <p>Interdit l'accès par le nom du domaine</p> <p>Exécution CGI par UID</p> <p>Interdit l'accès par adresse IP</p> <p>Interdit l'accès par utilisateur et groupe</p> <p>Peut changer la liste de contrôle d'accès de l'utilisateur sans redémarrer le serveur</p> <p>Permissions hiérarchiques pour les documents basés sur le répertoire</p> <p>Interdit l'accès par répertoire et fichier</p> <p>Groupes utilisateurs configurables (pas seulement une liste d'utilisateur unique)</p> <p>Peut cacher une partie d'un document suivant des règles de sécurité</p> <p>Compatible SSL v. 2, SSL v. 3</p> <p>Peut nécessiter mot de passe</p> <p>Les règles de sécurité peuvent se baser d'URLs</p> | <p>Serveur certifié intégré</p> <p>Interdit l'accès par le nom du domaine</p> <p>Exécution CGI par UID</p> <p>Interdit l'accès par adresse IP</p> <p>Interdit l'accès par utilisateur et groupe</p> <p>Compatible S-HTTP</p> <p>Peut changer la liste de contrôle d'accès de l'utilisateur sans redémarrer le serveur</p> <p>Permissions hiérarchiques pour les documents basés sur le répertoire</p> <p>Interdit l'accès par répertoire et fichier</p> <p>Groupes utilisateurs configurables (pas seulement une liste d'utilisateur unique)</p> <p>Peut cacher une partie d'un document suivant des règles de sécurité</p> <p>Compatible SSL v.2, SSL v. 3, Set</p> <p>Peut nécessiter mot de passe</p> <p>Les règles de sécurité peuvent se baser d'URLs</p> | <p>Serveur certifié intégré</p> <p>Interdit l'accès par le nom du domaine</p> <p>Interdit l'accès par adresse IP</p> <p>Interdit l'accès par utilisateur et groupe</p> <p>Compatible S-HTTP</p> <p>Peut changer la liste de contrôle d'accès de l'utilisateur sans redémarrer le serveur</p> <p>Permissions hiérarchiques pour les documents basés sur le répertoire</p> <p>Groupes utilisateurs configurables (pas seulement une liste d'utilisateur unique)</p> <p>Peut cacher une partie d'un document suivant des règles de sécurité</p> <p>Compatible SSL v.2, SSL v. 3, Set</p> <p>Peut nécessiter mot de passe</p> <p>Les règles de sécurité peuvent se baser d'URLs</p> |

|                                |  |  |   |   |
|--------------------------------|--|--|---|---|
| <b>Autres caractéristiques</b> | Agit aussi comme serveur proxy HTTP<br>Outils interactifs inclus<br>Accès direct (sans-CGI) au SGBD<br>Moteur de recherche<br>Maintenance à distance | le code source complet du serveur inclus<br>Agit aussi comme serveur proxy HTTP<br>Outils interactifs inclus<br>Accès direct (sans-CGI) au SGBD<br>Maintenance à distance<br>Moteur de recherche | Sert aussi d'autres protocoles TCP<br>Outils interactifs inclus<br>Accès direct (sans-CGI) au SGBD<br>Maintenance à distance<br>Moteur de recherche | Outils interactifs inclus<br>Accès direct (sans-CGI) au SGBD<br>Maintenance à distance<br>Moteur de recherche |
|--------------------------------|--|--|---|---|

Tab 1 : Comparaison détaillée entre les serveurs Web Oracle, IIS, Apache et Netscape. Source : WebServer Directory <http://Webserver.internet.com>

Il apparaît, en conclusion de cette comparaison que ces quatre serveurs offrent à peu de choses près les mêmes fonctionnalités pour la gestion de pages HTML, la sécurité et la conformité par rapport aux standards.

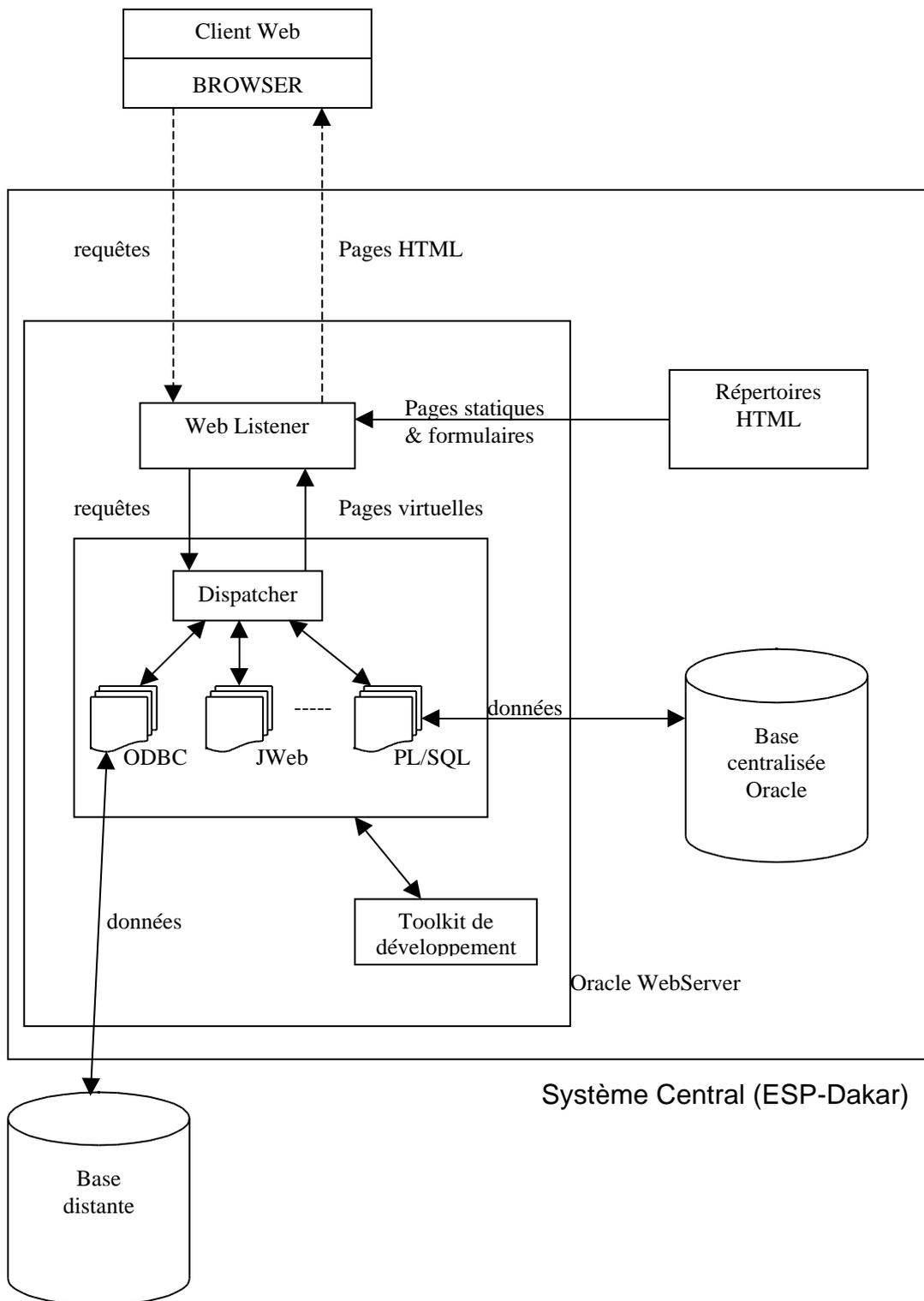
Nous ajoutons quelques commentaires sur la manière dont ces différents serveurs gèrent des accès aux bases de données, locales comme distantes.

On retrouve ici la dichotomie entre le monde Unix, majoritairement occupé par les serveurs Appache et Netscape. Les accès bases de données sont réalisés par des applications CGI généralement écrites en PERL et dans une moindre mesure en Java, C ou C++.

En revanche, dans le monde Microsoft on trouve majoritairement le serveur IIS couplé à des bases de données comme SQL server, Oracle ou Access pour les petites applications. La connexion à la base de données se fait par l'intermédiaire d'ODBC et les applications CGI sont écrites en Vbscript, Javascript ou par des techniques propriétaires comme IDC (Internet database connector) ou ASP (active server pages).

- La méthode IDC utilise des fichiers d'extension .idc qui contiennent des instructions SQL. L'URL envoyé par le navigateur n'est plus le nom d'une page HTML mais le nom d'un fichier .idc avec les paramètres nécessaires. Une fois ces instructions SQL exécutées au niveau de la base de données, le résultat est envoyé au navigateur en utilisant comme modèle un fichier d'extension .htx.
- Plus puissante, la méthode ASP consiste à incorporer des scripts (en VBScript, JavaScript ou PERL) dans les pages HTML, ce qui donne plus de facilités au programmeur. Toutefois, l'accès aux bases de données est ici plus complexe : Les pages ASP se connectent aux bases de données via des composants ActiveX.

Oracle Web Application Server, quant à lui, se connecte aux bases de données en suivant le schéma de la figure 3 ci-dessous :



*Fig.3 : Organisation des données réparties sous Oracle Web Application Server*

Le Web Listener est un « démon » HTTP, qui reçoit les requêtes de l'utilisateur sous forme d'adresses URL. Ce listener peut être un serveur web quelconque Apache, IIS ou le listener fourni par défaut avec Oracle Web Application Server.

Le web Listener cherche ensuite les pages HTML et formulaires demandés par la requête de l'utilisateur. Quand une requête nécessite des accès aux bases de données, le Web Listener l'adresse au Web Request Broker (WRB).

Suivant la nature de la requête reçue, le WRB se chargera d'accéder à la base de données puis de générer une page virtuelle contenant les résultats.

Pour cela, le WRB utilise des modules de routines livrées avec le système.

- Le module PL/SQL, qui exécute des procédures PL/SQL dans des bases de données.
- Le module Jweb, qui permet d'exécuter des procédures Java ou se connecter à des bases de données, en utilisant JDBC.
- Le module ODBC, permettant d'exécuter des requêtes SQL vers des bases de données offrant cette connexion.
- Le module C, pour exécuter des modules C.
- Le module LiveHTML, permettant d'interpréter et générer des pages dynamiques contenant des scripts Perl .
- Le module Perl, pour exécuter des modules Perl.

## 5 - Types d'applications

Comme nous l'avons souligné plus haut, les applications web dynamiques fournissent une information actualisée. Dans ce cas, il existe une interaction entre le navigateur et la base de données. Mais l'examen de la nature de cette interaction peut révéler une certaine complexité qui requiert de faire une analyse préalable sur la nature de l'applicatif (transactionnelle ou interactive).

### Cas d'une application transactionnelle

Lorsqu'il s'agit d'accéder à une base de données en temps réel, on se heurte au concept de transaction, le web devient véritablement dynamique, et les applications fonctionnent comme des applications Client-Serveur. Une telle architecture nécessite d'interfacer le serveur web avec un moniteur transactionnel chargé de dialoguer avec le serveur de bases de données. Pour ce type d'application, variante du Client-Serveur de présentation, le développement s'effectue depuis un environnement utilisant un langage de quatrième génération (L4G). C'est le cas d'Oracle qui permet de générer des applications transactionnelles dont les interfaces clientes sont directement accessibles à partir d'un navigateur Web.

### Cas d'une application interactive

Il s'agit d'applications pour lesquelles le temps de réponse n'est pas un facteur critique. Pour cela, il existe des automates d'interfaçage Web/SGBD. Ces outils permettent au concepteur de disposer de composants préalablement définis dans un environnement visuel. Microsoft propose notamment l'IDC que nous détaillerons plus

loin. Les autres produits se composent d'au moins deux modules : l'un pour élaborer les pages web et les formulaires, l'autre est le programme CGI appelé au niveau du serveur qui assume la connexion avec le serveur de bases de données à l'aide de pilotes natifs ou ODBC.

## **6 - Différentes architectures de couplage bases de données / Web**

Parmi les différents modèles d'architecture Web couplés aux bases de données, on distingue les modèles suivants : CGI, API, JAVA, ASP et IDC.

### **L'accès CGI (Common Gateway Interface)**

Cet accès est le plus ancien, il consiste à utiliser un appel CGI pour exécuter des programmes contenant des requêtes SQL vers une base de données. Cette méthode consomme beaucoup de ressources systèmes car chaque requête provenant d'un client web provoque au niveau du serveur HTTP un appel à un programme externe.

### **L'accès API (Application Programming Interface)**

Plus moderne que le CGI, il consiste à utiliser une API existante entre le serveur HTTP et la base de données. Il existe deux API auxquels se conforment les principales bases de données : NSAPI de Netscape et ISAPI de Microsoft. Ces API permettent de s'affranchir de codages de programmes en incluant dans les pages HTML les codes d'accès aux bases de données. Cet accès est très à la mode mais constitue un handicap au portage des applications, aussi bien par rapport aux serveurs HTTP qu'aux systèmes de gestion de bases de données.

### **L'accès distribué : JAVA**

Dernière née des technologies, celle-ci vise à fournir au client le logiciel lui permettant de faire lui même la connexion avec la base de données. Ceci est un des enjeux de JAVA qui notamment avec son extension JDBC permet de fournir une « applet » chargée de se connecter au serveur par une connexion ODBC.

JAVA est un langage de développement bien adapté au Web pour deux raisons : d'abord l'exécution d'une applet s'effectue sur une machine virtuelle Java, donc indépendamment de la plate-forme; ensuite, les applets sont compactes, donc transitent facilement sur le réseau. Son avantage par rapport à ODBC est qu'il ne nécessite pas de pilotes sur le poste client. Le téléchargement dynamique de l'applet permet d'interroger plus aisément des bases de données hétérogènes, quelle que soit leur localisation. Toutefois, le poste client doit assumer une partie des traitements tandis que le propre du web est de concentrer le traitement sur le serveur.

Toutes ces méthodes, bien que possédant un certain nombre de qualités et d'avantages, présentent aussi de nombreuses contraintes, notamment au niveau de la compatibilité du navigateur client ; mais encore et surtout un manque de facilité et de flexibilité quant à leur mise en œuvre. C'est pourquoi nous nous sommes tournés

vers les accès ASP (Active Server Pages) et IDC (Internet Database Connector) pour réaliser nos premiers tests.

### **L'accès ASP**

Les ASP (en français Pages de Serveur Actives) sont des scripts (applications dynamiques performantes et totalement interactives) exécutés depuis un serveur web intégrant la technologie ASP. Ici, le problème de compatibilité du Browser Client en fonction du langage utilisé pour script ASP ne se pose plus.

Il convient néanmoins de préciser qu'un script ASP est du code à cheval entre HTML et les langages de programmation tels que JavaScript, VBScript et JAVA. Le code HTML étant généralement utilisé pour la mise en forme et les liens hypertextes, tandis que les langages de programmation sont utilisés pour donner aux ordinateurs une série d'instructions complexes.

L'intérêt que nous portons à l'ASP réside dans sa simplicité et sa flexibilité d'intégration de langages tels que VBScript, Jscript, REXX, PERL, JAVA et les composants ActiveX, dans un même document. Les résultats satisfaisants déjà obtenus avec la technologie IDC/HTX nous ont poussés à effectuer nos premiers développements.

### **L'accès IDC/HTX**

IDC est un élément d'Internet Information Server de Microsoft. Il correspond à un fichier DLL (httpodbc.dll) qui s'appuie, comme son nom l'indique, sur le standard ODBC (Open DataBase Connectivity). L'intérêt de cette approche est que cette passerelle fonctionne avec n'importe quel SGBD possédant un driver ODBC, ce qui en fait une solution générique adaptable à différents contextes d'application (bases SQL Server, Oracle, ACCESS...).

Les différents composants de cette solution s'interfaçent de la façon suivante : deux types de fichiers sont utilisés par httpodbc.dll pour transmettre une requête à la base de données et permettre l'affichage des résultats. Les fichiers Internet Database Connector (.idc) permettent l'accès à la base de données et l'exécution des requêtes, et les fichiers modèles d'extension HTML (.htx) assurent la présentation des résultats sous forme de pages HTML.

Le fichier IDC indique la source ODBC à laquelle on veut accéder, les informations nécessaires à l'identification d'un utilisateur, la requête à soumettre au SGBD, et le fichier qui contient la présentation HTML à respecter pour visualiser le résultat de la requête.

#### **Exemple de Fichier IDC :**

```
Datasource: Soucre  
Username : nom  
PassWord : mot_de_passe  
Template: fichier.htx  
SQLStatement:  
+ SELECT nom , prenom FROM utilisateur WHERE pays = 'Sénégal'
```

Ceci est un exemple simple de consultation. Le même principe peut être appliqué pour des consultations plus complexes ou la mise à jour d'informations via les procédures stockées de SQL Server.

Dans ce cas, on peut créer un formulaire HTML qui référence un fichier IDC auquel les valeurs saisies sont passées en paramètre. La commande SQL du fichier IDC passera à son tour les paramètres à la procédure stockée.

Les procédures stockées permettent ainsi de développer des applications plus complexes. Elles assurent une vérification des valeurs saisies, l'insertion/la mise à jour/la suppression de lignes sur plusieurs tables en une seule transaction, et elles améliorent les sécurités d'accès en limitant par exemple les permissions accordées au compte utilisateur Internet.

Un fichier HTX contient la présentation HTML à respecter pour visualiser le résultat d'une requête. C'est donc une page HTML contenant des balises standard, complétées de zones spécifiques qui seront remplacées par les informations issues de la requête (les zones spécifiques sont matérialisées par <% ... %>).

### **Exemple de Fichier HTX :**

```
<HTML>
<B><%nom%></B> <%prenom%>
</HTML>
```

Les mots-clefs entre les balises <% et %> sont interprétés et remplacés par les informations trouvées dans la base sous forme de listes.

En reprenant le schéma d'interaction des composants d'Internet Database Connector, nous positionnons les fichiers HTX et IDC de la façon suivante : cette offre de connexion entre le serveur Web et un SGBD est un exemple d'extension ISAPI d'Internet Information Server, extensions propriétaires qui remplacent le standard CGI.

## 8 - Etude de cas

Le Centre de Suivi Ecologique ( CSE ), partenaire privilégié du projet SIMES sur l'Opération pilote « Vallée du fleuve Sénégal » nous a sollicité pour l'aider à développer un site WEB couplé à une base de données Access. C'est pourquoi l'ensemble des applications ont été réalisées et testés sur un projet du CSE où il a été question de la "Mise en place d'un système d'interrogation à distance d'une base de Méta Données pour le Système d'Information sur la Désertification".

Figure 4 : le modèle relationnel

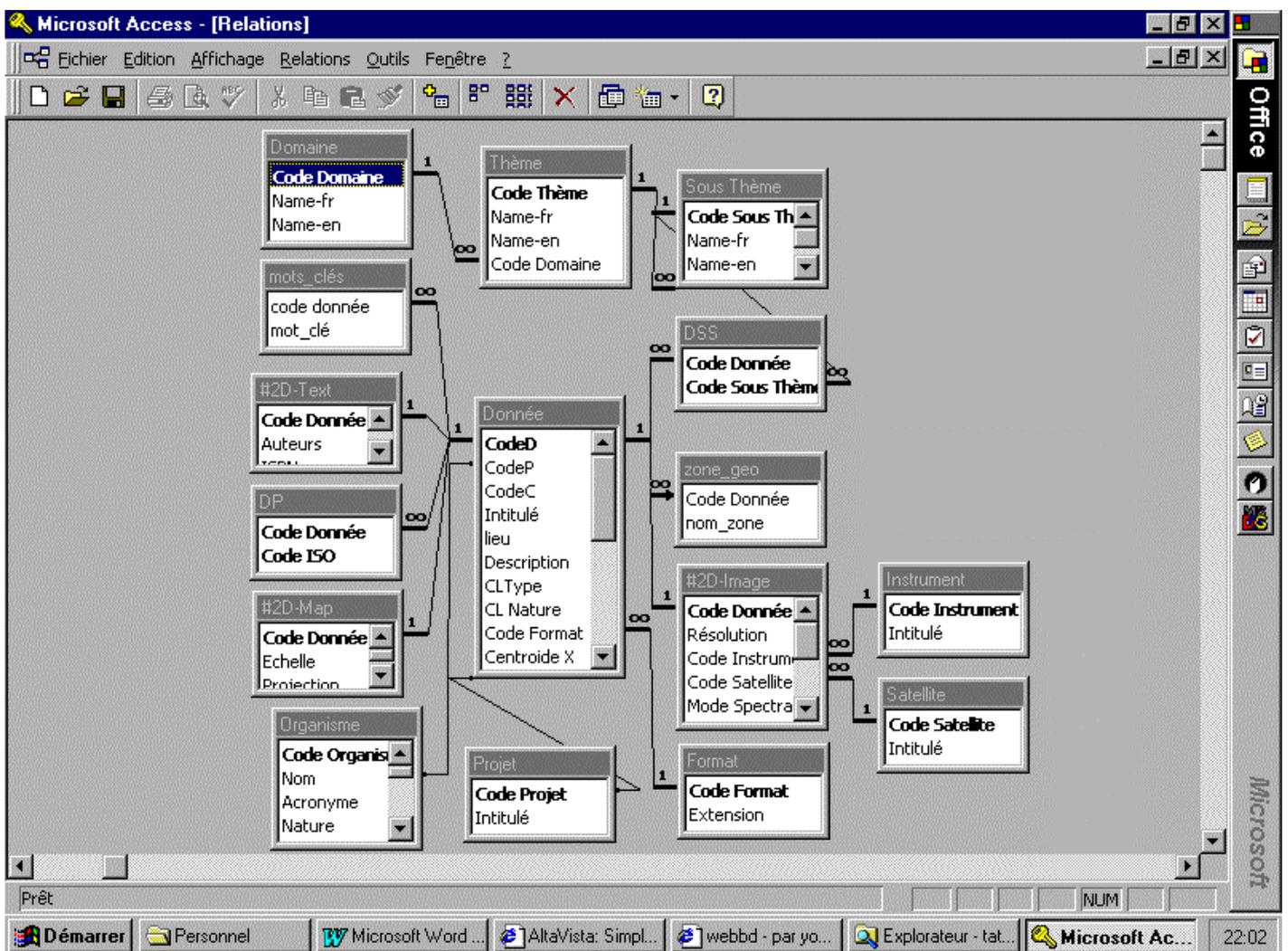
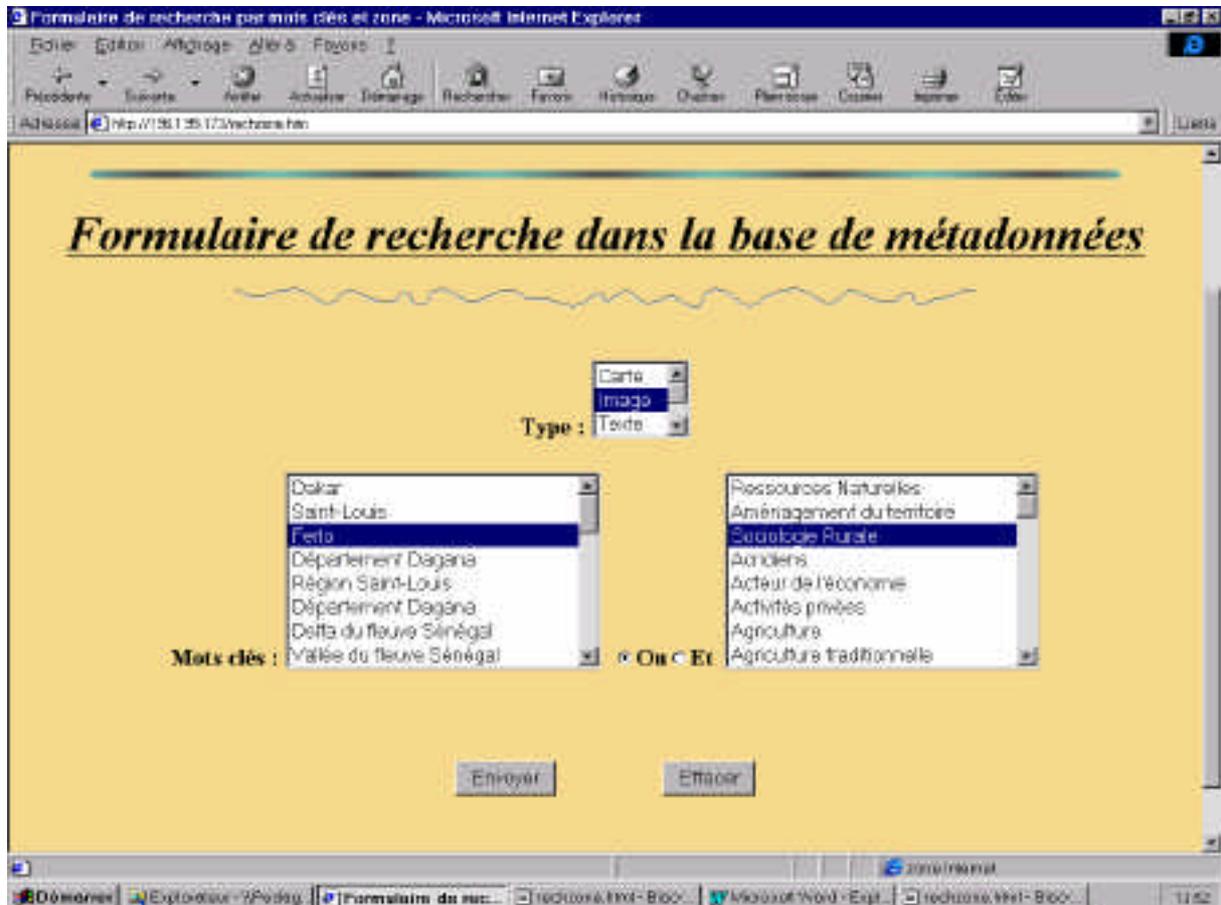


Figure 5 : Exemple de requête envoyée vers la base de données via un formulaire sur le web



### Source de la page HTML

```
<!DOCTYPE HTML PUBLIC "-//IETF/DTD HTML//EN">
<html>

<head>
<meta http-equiv="Content-Type"
content="text/html; charset=iso-8859-1">
<meta name="GENERATOR" content="Microsoft FrontPage 2.0">
<title>Formulaire de recherche par mots clés et zone</title>
</head>

<body bgcolor="#F0DC9F">

<p align="center"><font color="#008000" size="7"><em><strong><u>Système
d'Informations sur la Désertification </u></strong></em><em><strong>(SID)</strong></em></font></p>

<p align="center"></p>

<p align="center"><font size="6"><em><strong><u>Formulaire de
recherche dans la base de métadonnées </u></strong></em></font></p>
```

```
<p align="center"><font size="5"><strong></strong></font></p>
```

```
<form action="rechzone.idc" method="POST">  
  <blockquote>  
    <blockquote>  
      <div align="center"><center><table border="0"  
        cellpadding="9" cellspacing="11">  
        <tr>  
          <td><strong>Type : </strong><font size="5"><select  
            name="type" multiple size="3">  
              <option value="1">Carte</option>  
              <option value="2">Image</option>  
              <option value="3">Texte</option>  
              <option value="4">Tableau</option>  
              <option value="5">Autres</option>  
            </select></font></td>  
          </tr>  
        </table>  
      </div><div align="center"><center><table  
        border="0" cellpadding="2" cellspacing="5">  
        <tr>  
          <td><strong>Mots clés : </strong><select  
            name="zone" multiple size="8" tabindex="2">  
              <option>Dakar</option>  
              <option>Saint-Louis</option>  
              <option>Ferlo</option>  
              <option>Département Dagana</option>  
              <option>Région Saint-Louis</option>  
              <option>Département Dagana</option>  
              <option>Delta du fleuve Sénégal</option>  
            </select></td>  
          <td valign="bottom"><input type="radio"  
            name="choix" value="or"><strong>Ou</strong><input  
            type="radio" checked name="choix" value="And"><strong>Et</strong></td>  
          <td><select name="motcle1" multiple size="8"  
            tabindex="2">  
            <option value="Ressource Naturelle">Ressources Naturelles</option>  
            <option  
            value="Aménagement du territoire">Aménagement du territoire</option>  
            <option value="Sociologie Rurale">Sociologie Rurale</option>  
            <option value="Acridiens">Acridiens</option>  
            <option value="Acteur de l'économie">Acteur de l'économie</option>  
            <option value="Activités privées">Activités privées</option>  
            <option value="Agriculture">Agriculture</option>  
          </select></td>  
        </tr>  
      </table>  
    </center></div><p align="center">  
  </blockquote>  
  <p align="center"><input type="submit" name="B1"  
    value="Envoyer">  
  <input type="reset" name="B2" value="Effacer"></p>  
</form>  
</body>  
</html>
```

Figure 6 : modèle du fichier HTX – résultat



Code HTML associé à la page Résultat (figure 6)

```

<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML//EN">
<html>

<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
<meta name="GENERATOR" content="Microsoft FrontPage 3.0">
<title>Résultat de la requete</title>
</head>

<body bgcolor="#F0DC9F">

<p>&nbsp;</p>

<p align="center"><font size="7"><em><strong><u>Résultat de la
requête</u></strong></em></font></p>
<div align="center"><center>

<table border="2">
  <tr>
    <td align="center"><p align="center"><strong>Mot clé</strong></td>
    <td align="center"><strong>Disponible à</strong></td>
    <td align="center"><strong>Acronyme</strong></td>
    <td align="center"><strong>Type</strong></td>
    <td align="center"><strong>Titre </strong></td>
    <td align="center"><strong>URL</strong></td>
    <td align="center"><font size="3"><strong>Nom de la zone</strong></font></td>
  </tr>
  <%begindetail%> <tr>
    <td align="center"><%mot_clé%>
  </td>
    <td align="center"><%Nom%>
  </td>
    <td align="center"><%Acronyme%>
  </td>
    <td align="center"><%CLType%>
  </td>
    <td align="center"><%Intitulé%>
  </td>
  </tr>

```

```

        <td align="center"><a href="http://196.1.95.173/<%Chemin%>"><%Chemin%></a></td>
        <td><%nom_zone%>
    </td>
</tr>
</enddetail%></table>
</center></div>
</body>
</html>

```

### Source du fichier de connexion base de données (IDC associé à la page résultat)

Datasource: Source  
 Template: rechzone.htx  
 SQLStatement: SELECT mots\_clés.mot\_clé, Organisme.Nom, Organisme.Acronyme, Donnée.Chemin, Donnée.Intitulé, zone\_geo.nom\_zone, Donnée.CLType  
 +FROM (Organisme INNER JOIN (Donnée LEFT JOIN zone\_geo ON Donnée.CodeD = zone\_geo.[Code Donnée]) ON (Organisme.Nom = Donnée.CodeC) AND (Organisme.Nom = Donnée.CodeP)) INNER JOIN mots\_clés ON Donnée.CodeD = mots\_clés.[code donnée]  
 +WHERE (((mots\_clés.mot\_clé) In ('%motcle1%')) AND ((Donnée.CLType) In (%type%))) %choix% ((zone\_geo.nom\_zone) In ('%zone%'));

Figure 7 : Réponse à la requête envoyée (Fichier HTX Résultat)

| Mot clé               | Disponible à                              | Acronyme | Type | Titre   | URL                               | Nom de la zone     |
|-----------------------|---|----------|------|---|-----------------------------------|--------------------|
| ressources naturelles | Institut Sénégalais de Recherche Agricole | ISRA-SL  | 1    | Etude sur la gestion communautaire des ressources naturelles au Sénégal: l'étude de cas de Mbane, département de Dagana, région de Saint-Louis du 24 août-3septembre 1992 | <a href="#">IMAGES/carte1.gif</a> | Département Dagana |
| ressources agricoles  | Institut Sénégalais de Recherche Agricole | ISRA-SL  | 1    | Etude sur la gestion communautaire des ressources naturelles au Sénégal: l'étude de cas de Mbane, département de Dagana, région de Saint-Louis du 24 août-3septembre 1992 | <a href="#">IMAGES/carte1.gif</a> | Département Dagana |
| environnement         | Institut Sénégalais de Recherche          | ISRA-SL  | 1    | Etude sur la gestion communautaire des ressources naturelles au Sénégal: l'étude de cas de Mbane, département de Dagana. région de  | <a href="#">IMAGES/carte1.gif</a> | Département Dagana |

A la sélection de l'URL le document associé est visualisé



```

<td align="center"><strong>Disponible à</strong></td>
<td align="center"><strong>Acronyme</strong></td>
<td align="center"><strong>Type</strong></td>
<td align="center"><strong>Titre </strong></td>
<td align="center"><strong>URL</strong></td>
<td align="center"><font size="3"><strong>Nom de la zone</strong></font></td>
</tr>
<tr>
<td align="center">bilharziose</td>
<td align="center">Institut Français de Recherche Scientifique pour le Développement en
Coopération</td>
<td align="center">ORSTOM-FR</td>
<td align="center">3</td>
<td align="center">Représentation des maladies et recours thérapeutiques chez les peulh et la walo-walo de
Richard Toll et des environs du Lac de Guiers: maladies sexuellement transmises, maladies associées à
l'eau</td>
<td align="center"><a
href="http://196.1.95.173/"></a></td>
<td>Ferlo</td>
</tr>
</table>
</center></div>
</body>
</html>

```

## 9 - Conclusion

Malgré la simplicité de la solution IDC/HTX, nous avons relevé de nombreuses insuffisances telles que :

- son incompatibilité avec UNIX puisque liée à Windows NT;
- son manque d'ouverture pour réaliser des traitements spécifiques autres que les requêtes SQL.
- Le manque de richesse fonctionnelle inhérent au modèle HTX de visualisation des résultats.

En plus de cela, lorsque la structure de la base de données devient complexe, la complexité des requêtes s'accroît et par conséquent les contrôles deviennent plus fins et moins facilement interprétables par le serveur. C'est pourquoi, nous avons déjà entamé la migration vers des outils de développement plus puissants et multi plates-formes (ASP et JAVA).

Les technologies évoquées dans ce rapport sont en pleine évolution ; par conséquent proposer un choix unique pleinement justifié paraît actuellement prématuré.

C'est la raison pour laquelle nous proposons de tester trois solutions qui semblent se dégager.

- Une solution entièrement basée sur LINUX avec l'association du serveur WEB Apache de la base de donnée Postgres et du langage Java.

Cette solution offre le double avantage de la quasi gratuité de la plate-forme logicielle et de sa portabilité. En revanche les techniques sous-jacentes ne sont qu'au début de leur développement au sein de l'équipe de Dakar.

- Une solution entièrement basée sur Windows NT avec les technologies IDC et ASP. Cette solution est aujourd'hui bien maîtrisée par l'équipe de Dakar et c'est la raison pour laquelle l'étude de cas que nous avons présenté ici repose sur la technique IDC. Il faut quand même préciser l'abandon de la technique IDC et son remplacement par ASP.
- Une solution basée sur l'offre Oracle V8 et Oracle Web server. L'équipe de Dakar possède déjà l'intégralité du logiciel offert gracieusement par la société Oracle dans le cadre d'un appui aux activités pédagogiques et de recherche. Le déploiement de cette solution est également en cours.

En tant qu'élément fédérateur de toutes ces solutions, Java apparaît comme l'outil de développement qui permettrait à terme de garantir la portabilité et donc de s'affranchir de la contrainte de l'unicité de la plate-forme matérielle et logicielle.

#### **Bibliographie :**

- [1] - Le Micro Bulletin CNRS N° 74 mai/juin 1998.
- [2] - Visual Interdev, Joseph O'Neil Orsborn/Mc Graw Hill.
- [3] - Programmation CGI, ShiShir Gundavaram , Editions O' Reilly.