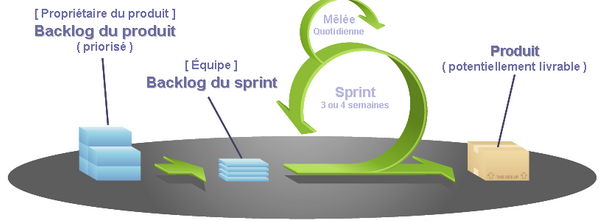
1. Présentation générale de l’application

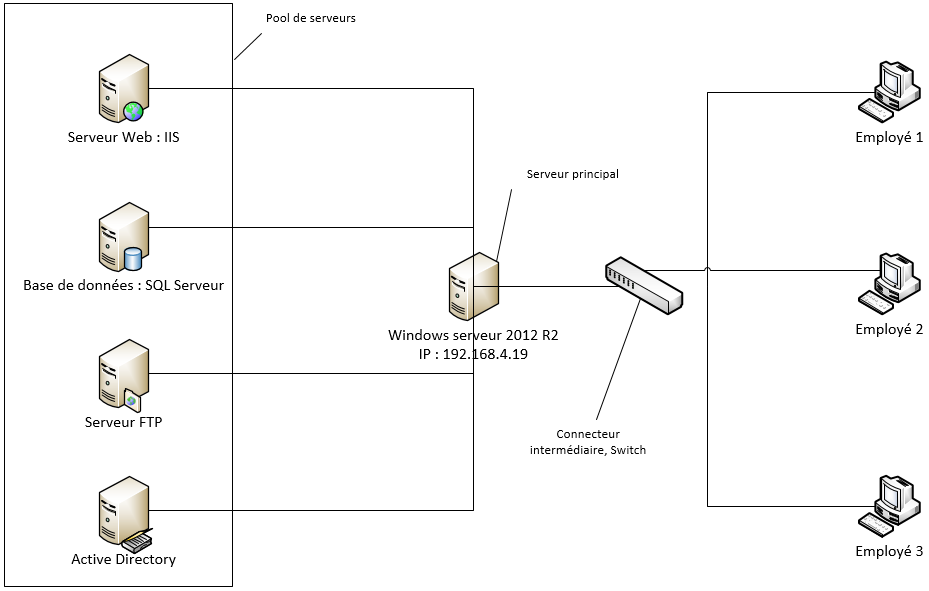
Notre application étant une solution assez complexe à mettre en œuvre et avec une période assez longue. Nous avons décidé avec le client, monsieur PATRICK LEFEVRE d’utiliser une méthode de communication pour nous guider vis-à-vis de ses besoins et ainsi lui offrir une solution à ces attentes de la meilleur qualité possible. La méthode utilisé s’appel, ‘La méthode AGILE’. Pourquoi AGILE ? L’agilité est un concept mis en œuvre pour recentrer le client au cœur du projet, elle permet une meilleure communication, une meilleure réactivité, une meilleure satisfaction, une meilleure visibilité sur le projet, une meilleure efficacité de la part des personnes en charge du projet, une meilleur efficacité et une meilleure adaptabilité.

Exemple de principe de la méthode agile :



Une fois l’agilité convenue avec le client, l’équipe m5f a décidé de segmenté le projet en trois parties pour une meilleure fluidité dans le travail et pour une meilleure organisation. Le faite de segmenter le projet, s’appel : les lots. Nous avons séparé le travail documentaire dans le premier lot, puis le commencement du développement dans le deuxième lot, enfin dans le dernier lot, se trouve la finalisation du développement ainsi que l’écriture de la thèse finale.

1. Architecture réseau
   1. Plan réseau



* 1. Explication du réseau mis en place

Le réseau mis en place par nos soins, contribue fortement au fonctionnement de l’application intranet. Pour ce faire, un minimum devait être réalisé, à savoir : un serveur DNS (qui sert à convertir l’adresse IP de l’application en nom de domaine, pour faciliter l’accès au utilisateur. Ce nom de domaine est : ‘www.knowledgemanagement.fr’), un serveur IIS (qui contient l’application demandé par le client, à savoir une base de connaissance) et un Active Directory (qui contient tous les compte utilisateurs de la société. Grâce à ce système les utilisateurs ne pourront accéder à l’application que si ils sont inscrit dans l’active directory). De plus, une liaison permanente est en place entre le serveur principal et les différents serveurs du parc informatique. Les employées de l’entreprise pourront se connecter à l’application web de façon permanente.

* + 1. DNS

Un serveur DNS est système informatique qui permet de convertir une adresse IP en nom de domaine (exemple : IP 🡺 192.168.1.19 sera convertir en ‘www.knowledgemanagement.fr’). Ce procédé a était mis en place car les utilisateurs du web extranet ou du web intranet, retienne plus facilement une phrase plutôt qu’une série de chiffres. Pour ce projet, le serveur DNS est installé sur le serveur principal ‘Windows serveur 2012 R2’ est fournis un nom de domaine pour l’application.

* + 1. IIS

Un serveur IIS est un système informatique qui permet la lecture de site internet. ‘IIS’ est certifié Microsoft est permet l’hébergement de plusieurs site web. Ce serveur web marche avec un langage prioritaire qui est : ASP.NET (Propriétaire Microsoft). Pour ce projet, nous avons choisi d’utiliser le serveur web IIS qui héberge non application web pour l’entreprise ainsi que le serveur FTP qui est en coalition avec le serveur IIS.

* + 1. Active Directory

Active Directory est un service d’annuaire LDAP créé par Microsoft. Son objectif principal est de centraliser les identifications et authentifications de systèmes Windows. Il permet de répertorier les éléments d’un réseau administré comme les comptes utilisateurs, les serveurs, les postes de travail, les dossiers partagés, les imprimantes… Active Directory est devenu indispensable pour une entreprise à fort personnel. En effet, une gestion des salarié est quasiment obligatoire arrivé à un certain seuil. Nous avons mis en place ce procédé pour gérer les utilisateurs de la société de monsieur PATRICK LEFEVRE. Ce procédé permet de séparer les administrateurs, des salariés. Le but de ce choix nous permet au niveau applicatif de gérer les personnes qui se connectent à l’application interne. De plus, nous avons effectué une synchronisation automatique entre l’Active Directory et la base de données SQL Serveur pour récupérer les comptes utilisateurs.

* 1. Pourquoi cette solution ?

Nous avons choisi une solution Microsoft pour la fluidité de ces serveurs. En effet, cette solution nous permet de gérer au mieux l’infrastructure des différents serveurs mis en place. Nous avons décidé de mettre en place un ‘Active Directory’, or ce système n’est utilisé que par Microsoft car ils en sont propriétaire. Grâce au ‘Gestionnaire de serveur’ intégré nous pouvons voir tous les problèmes liée à l’utilisation, le fonctionnement de chaque services, de chaque serveurs…

De plus, l’administration est simplifiée grâce à l’interface graphique qu’elle délivre. Il est facile à déployer, à administrer et à utiliser. Contrairement à son concurrent principal ‘Linux’, cette solution propose un panel d’outils que l’on peut installer via l’interface graphique, ce qui nous garantit un gain de temps dans la mise en place de l’architecture. Comme les Windows serveur ont le même noyau que les Windows classique vendu sur le marché, il est très facile de mettre à jour nos différents serveurs. Le choix sait porter sur un la version 2012 R2, pourquoi ? Non pas parce que cette dernière est plus facile d’utilisation mais pour avoir les dernières mises à jour de sécurité. Un autre point essentiel de la solution que nous avons choisie est que physiquement sur nos serveurs, nous n’avons pas besoins de mettre X carte réseaux pour pouvoir les administrer. Un seul suffit pour pouvoir accéder à l’ensemble du parc.

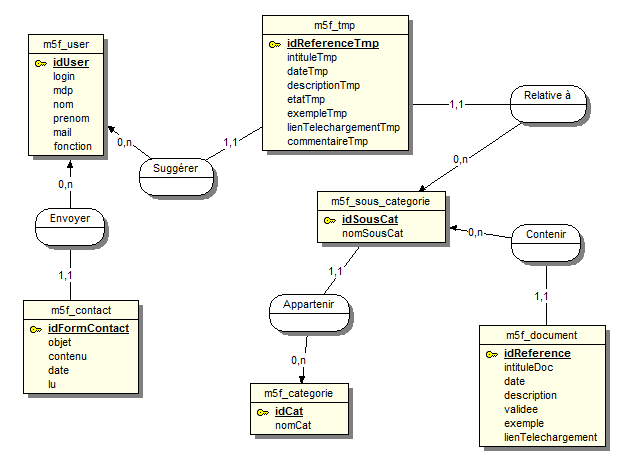
* 1. Base de données
     1. Outils de SGBD

Notre application web est système informatique dynamique, ce qui veut dire que cette dernière est forcément administrée dynamiquement avec une base de données. Nous avions le choix entre la solution ‘MySQL’ et la solution ‘SQL Serveur’. Notre choix sait portée sur SQL Serveur pour plusieurs raisons. Notre architecture réseau utilise des serveurs Microsoft et pour rester dans le même cadre, cette

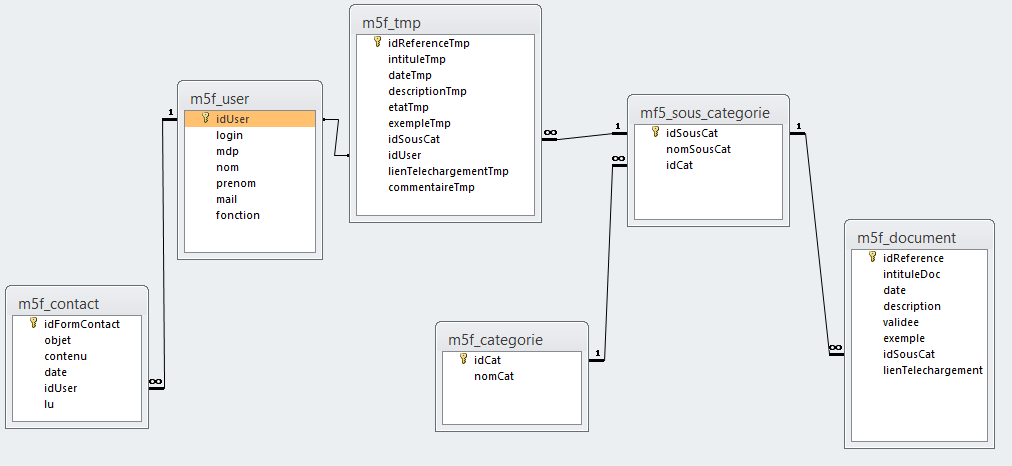
1. Dessin d’application
   * + 1. Dictionnaire de données

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Données | Type | Longueur | Description | Remarque |
| ***m5f\_user*** | idUser | INT |  | Identifiant de l’utilisateur |  |
|  | login | VARCHAR | 70 | Login de connexion |  |
|  | mdp | VARCHAR | 255 | Mot de passe de connexion |  |
|  | nom | VARCHAR | 70 | Nom de l’utilisateur |  |
|  | prenom | VARCHAR | 70 | Prénom de l’utilisateur |  |
|  | mail | VARCHAR | 255 | Adresse e-mail |  |
|  | fonction | VARCHAR | 255 | Fonction | Accesseur, Administrateur ou Contributeur |
| ***m5f\_sous\_categorie*** | idSousCat | INT |  | Identifiant de la sous-catégorie |  |
|  | nomSousCat | VARCHAR | 70 | Nom de la sous-catégorie |  |
| ***m5f\_contact*** | idFormContact | INT | 11 | Identifiant du contact |  |
|  | objet | VARCHAR | 255 | Objet du message de contact |  |
|  | contenu | TEXT |  | Contenu du message de contact |  |
|  | lu | BIT |  | Indique si le message a été lu par l’administrateur |  |
|  | date | DATETIME |  | Date d’envoi du message |  |
| ***m5f\_document*** | idReference | VARCHAR | 32 | Identifiant du document |  |
|  | intituleDoc | VARCHAR | 255 | Intitulé du document |  |
|  | date | DATETIME |  | Date d’édition du document |  |
|  | description | TEXT |  | Description du document |  |
|  | validee | BIT |  | Etat du document | Oui / Non |
|  | exemple | TEXT |  | Exemple du document |  |
|  | lienTelechargement | VARCHAR | 255 | Lien pour télécharger le .exe/.zip |  |
| ***m5f\_categorie*** | idCat | INT |  | Identifiant de la catégorie |  |
|  | nomCat | VARCHAR | 70 | Nom de la catégorie |  |
| ***m5f\_tmp*** | idReferenceTmp | VARCHAR | 32 | Identifiant du document temporaire |  |
|  | intituleTmp | VARCHAR | 255 | Intitule du document temporaire |  |
|  | descriptionTmp | TEXT |  | Description du document temporaire |  |
|  | dateTmp | DATE |  | Date du document temporaire |  |
|  | etatTmp | VARCHAR | 32 | Etat du document temporaire | Oui / Non |
|  | exempleTmp | TEXT |  | Exemple du document temporaire |  |
|  | lienTelechargementTmp | VARCHAR | 255 |  |  |
|  | commentaireTmp | TEXT |  | Commentaire si refusé |  |

* + - 1. MCD



* + - 1. MLD



* + - 1. MLR

**m5f\_user** (idUser, login, mdp, nom, prenom, mail, fonction)

**m5f\_categorie** (idCat, nomCat)

**m5f\_sous\_categorie** (idSousCat, nomSousCat, #idCat)

**m5f\_document** (idReference, intituleDoc, date, description, validee, exemple,#idSousCat, lienTelechargement)

**m5f\_tmp** (idReferenceTmp, intituleTmp, dateTmp, descriptionTmp, etatTmp, exempleTmp, #idSousCat, lienTelechargementTmp, #idUser, commentaireTmp)

**m5f\_contact** (idFormContact, objet, contenu, date, lu, #idUser)

1. Liaison Active Directory & BDD

Nous avons mis en place une liaison entre la base de données et l’Active Directory de l’entreprise qui nous permet de récupérer tous les comptes utilisateurs pour permettent à ces dernier de pouvoir se connecter en tant que membres de l’entreprise. Pour ce faire nous avons dû exécuter des scripts pour garantir une liaison permanente entre ces deux entités.

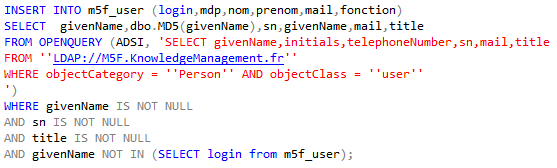
Ce script permet d’établir une liaison entre la base de données et l’Active Directory en créant un objet ADSI.



L’étape suivante consiste à lier un identifiant SQL Serveur avec un identifiant Active Directory. Pour ce faire, il faut exécuter la commande suivante.



Ensuite, une fois la liaison effectuée, un script d’insertion en base qui va permettre d’aller chercher les utilisateurs dans l’Active Directory et les insérer dans la base de données.



Enfin, il est possible de rompre la liaison entre SQL Serveur et l’Active Directory, en exécutant ce script.

