



HAL
open science

Migration et optimisation d'une base de données dans le cadre d'un logiciel de gestion de maintenance assistée par ordinateur

Julien Jammes

► **To cite this version:**

Julien Jammes. Migration et optimisation d'une base de données dans le cadre d'un logiciel de gestion de maintenance assistée par ordinateur. Base de données [cs.DB]. 2015. dumas-01291574

HAL Id: dumas-01291574

<https://dumas.ccsd.cnrs.fr/dumas-01291574>

Submitted on 21 Mar 2016

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

CONSERVATOIRE NATIONAL DES ARTS ET METIERS

CENTRE REGIONAL ASSOCIE DE HAUTE NORMANDIE

MEMOIRE

présenté en vue d'obtenir le DIPLOME d'INGENIEUR CNAM

SPECIALITE : Ingénieur Réseaux, Systèmes et Multimédia

OPTION : Réseaux et Systèmes Répartis

par

JAMMES Julien

**Migration et Optimisation d'une base de données dans le cadre d'un
logiciel de Gestion de Maintenance Assistée par Ordinateur**

Soutenu le 07/05/2015

JURY

PRESIDENT : Yann POLLET, Professeur

MEMBRES : Claude DUVALLET, Maître de conférences

Nicolas LARZUL, Ingénieur

Marc GORKA, Maître de conférences

Philippe WENDER, Ingénieur

REMERCIEMENTS

Je tiens à remercier Joël REVERT, directeur de l'agence Activités Nucléaires pour la confiance qu'il m'a accordée dès le départ du projet et pour le temps qu'il a consacré à m'accompagner lors de mes débuts au sein de l'agence Activités Nucléaires afin de bien comprendre le métier et le rôle du logiciel dans l'organisation de l'agence.

Je tiens également à remercier Patrice DE MARCONNAY, responsable grand projet informatique de COFATHEC, dont l'expérience et le recul sur les projets informatiques d'envergure a été d'un très grand secours, en particulier lors des premières étapes-clefs du projet.

Enfin, je tiens à remercier Claude DUVALLET, mon tuteur de stage, pour sa patience et ses conseils judicieux lors de la rédaction de ce mémoire.

ABREVIATIONS / GLOSSAIRE

Access : SGBDR édité par Microsoft, inclus dans la suite bureautique Office.

Active Directory : implémentation par Microsoft des services d'annuaires LDAP.

Application répartie : application dont l'architecture a été divisées en couches fonctionnelles séparées qui communiquent entre elles (exemple d'une application 3-tiers : une couche présentation pour l'affichage, une couche métier pour les traitements et une couche données pour le stockage des données persistantes).

Back-end : partie d'une application devant produire un résultat.

CFC : Chlorofluorocarbure : sous-classe de gaz fluorés, ininflammable, peu couteux et extrêmement stable, très utilisés notamment dans l'industrie du froid avant la découverte de leur impact sur la couche d'ozone et leur effet de serre et dont l'utilisation a été interdite à partir de 2000 suite au protocole de Montréal (1987).

Citrix : aujourd'hui Citrix XenApp : logiciel permettant d'accéder à distance à des applications, à partir de clients légers.

CNPE : Centre Nucléaire de Production d'Électricité (E.D.F.).

DSI : Direction des Systèmes d'Information.

Fluide frigorigène : fluide pur ou mélange de fluides purs sélectionnés principalement pour leur grande propriété d'absorption de chaleur (calories) lorsqu'ils passent de leur phase liquide à leur phase gazeuse et dont la principale propriété est de s'évaporer à une faible température sous pression atmosphérique.

Front-end : partie visible de l'application permettant les échanges entre l'utilisateur et le back-end.

GMAO : Gestion de Maintenance Assistée par Ordinateur (logiciel).

HCFC : Hydrochlorofluorocarbures : sous-classe de gaz fluorés utilisés en remplacement des CFC à partir de 2000. Moins stables que les CFC (donc moins destructeurs pour la couche d'ozone), ils conservent néanmoins un impact non négligeable sur la couche d'ozone et ainsi qu'un puissant effet de serre. Le protocole de Copenhague (1992) les interdit à l'horizon 2030 pour les pays industrialisés.

HFC : hydrofluorocarbures : sous-classe de gaz fluorés utilisés en remplacement des CFC et des HCFC suite à l'application du protocole de Montréal. Les HFC ne contiennent pas d'atomes de chlore responsable de la dégradation de la couche d'ozone. Ils restent néanmoins des puissants gaz à effet de serre et relèvent du protocole de Kyoto (1997) visant à réduire leur utilisation.

IHM : Interface Homme-Machine. Cf. Front-end.

LDAP : Lightweight Directory Access Protocol : protocole permettant l'interrogation et la modification des services d'annuaires.

SGBDR : Système de Gestion de Base de Données Relationnel : logiciel système destiné à stocker et à partager des informations dans une base de données relationnelle.

SQL Server : SGBDR édité par Microsoft.

SSII : Société de Service en Ingénierie Informatique : société de service spécialisée en génie informatique.

SOMMAIRE

REMERCIEMENTS.....	2
ABREVIATIONS / GLOSSAIRE.....	3
INTRODUCTION.....	7
1. LE CONTEXTE DU PROJET	9
1.1. L'agence « Activités Nucléaires »	9
1.2. La GMAO AssystMaint.....	13
1.2.1. Les logiciels de Gestion de Maintenance Assistée par Ordinateur (G.M.A.O)	13
1.2.2. La G.M.A.O. de l'agence : AssystMaint.....	14
2. LE PROJET	17
2.1. L'architecture de départ de l'application.....	18
2.1.1. Architecture logicielle	18
2.1.2. Architecture matérielle	20
2.2. Les contraintes du projet.....	22
2.2.1. Les contraintes techniques	22
2.2.2. Les contraintes fonctionnelles	24
2.2.3. Les contraintes organisationnelles.....	24
2.3. L'architecture cible de l'application	25
2.3.1. Architecture matérielle	25
2.3.2. Architecture logicielle	27

3. LE DEROULEMENT DU PROJET	29
3.1. Le travail préalable.....	29
3.1.1. Rédaction du cahier des charges et sélection du prestataire	29
3.1.2. La préparation de la migration des données.....	31
3.2. Le suivi des développements du prestataire.....	32
3.2.1. Le démarrage du projet.....	32
3.2.2. Les procédures de test	33
3.2.3. Les premières divergences avec le prestataire	34
3.2.4. Poursuite de la migration et retard du projet	35
3.3. La mise en production	36
4. L'APRES-PROJET.....	38
4.1. Un exemple concret d'évolutions de l'application : la gestion des mouvements de fluide.....	39
4.2. L'optimisation de la base de données SQL Server.....	43
4.2.1. L'optimisation d'un SGBDR.....	43
4.2.2. L'optimisation de la base de données SQL Server d'AssystMaint.....	49
CONCLUSION.....	55
Bibliographie.....	57
Liste des figures & tableaux	58
Résumé	59
Summary	59

INTRODUCTION

Au sein de l'agence Activités Nucléaires de COFELY, l'essentiel de l'activité est gérée par l'intermédiaire d'un logiciel développé en interne : AssystMaint, une application de Gestion de Maintenance Assistée par Ordinateur (GMAO).

Elle est utilisée au sein de l'agence Activités Nucléaires depuis le milieu des années 90. Elle a été développée initialement en autodidacte par Joël REVERT, directeur de l'agence, pour répondre aux besoins spécifiques de l'activité de l'agence.

Développée sous Access 97 au moment de mon arrivée, tant pour l'Interface Homme-Machine (IHM) que pour le Système de Gestion de Bases de Données (SGBD), l'application a ensuite été convertie au format Access 2003 (IHM + SGBD).

Elle a ensuite été transformée en application répartie (3 tiers) dans le cadre du projet de migration de l'application dont il est question ici.

L'impulsion d'origine du projet de migration de l'application sur SQL Server, une technologie robuste et facilement intégrable au SI de GDF-Suez, vient de la DSI qui voulait assurer la pérennité de l'application.

Nous verrons que tous ces éléments (origine de l'application, déclenchement du projet et choix technologiques) ont joué un rôle important dans le déroulement du projet.

C'est pourquoi nous verrons dans un premier temps le contexte du projet, en commençant par présenter l'Agence Activités Nucléaires et sa place dans le groupe GDF-Suez. Nous verrons également le rôle central de la GMAO AssystMaint dans l'activité de l'agence.

Nous présenterons ensuite le projet en lui-même : en premier lieu, l'architecture initiale de l'application puis les contraintes impactant le cahier des charges du projet et enfin l'architecture cible de l'application qui en découle.

Nous poursuivrons avec une étude du déroulement du projet : le travail préalable à la migration (sélection du prestataire et travail sur les données en prévision de la migration), le suivi du travail du prestataire et enfin la mise en production de l'application migrée sur SQL Server.

Nous terminerons enfin en présentant le travail réalisé depuis la migration de l'application. Nous prendrons en exemple le développement d'une fonctionnalité avancée de l'application et nous nous intéresserons enfin au travail d'optimisation de la base de données sous SQL Server qui n'avait pas été opéré par le prestataire de la migration des données.

1. LE CONTEXTE DU PROJET

Pour bien comprendre les différentes étapes du projet et les contraintes qui ont pesées, il est important de le resituer dans son contexte.

Nous présenterons donc d'abord l'agence Activités Nucléaires et son activité avant de nous intéresser à l'application en elle-même.

1.1. L'agence « Activités Nucléaires »

L'agence Activité Nucléaire est une agence spécialisée de la branche Énergie Services de GDF Suez située à Gonfreville l'Orcher.

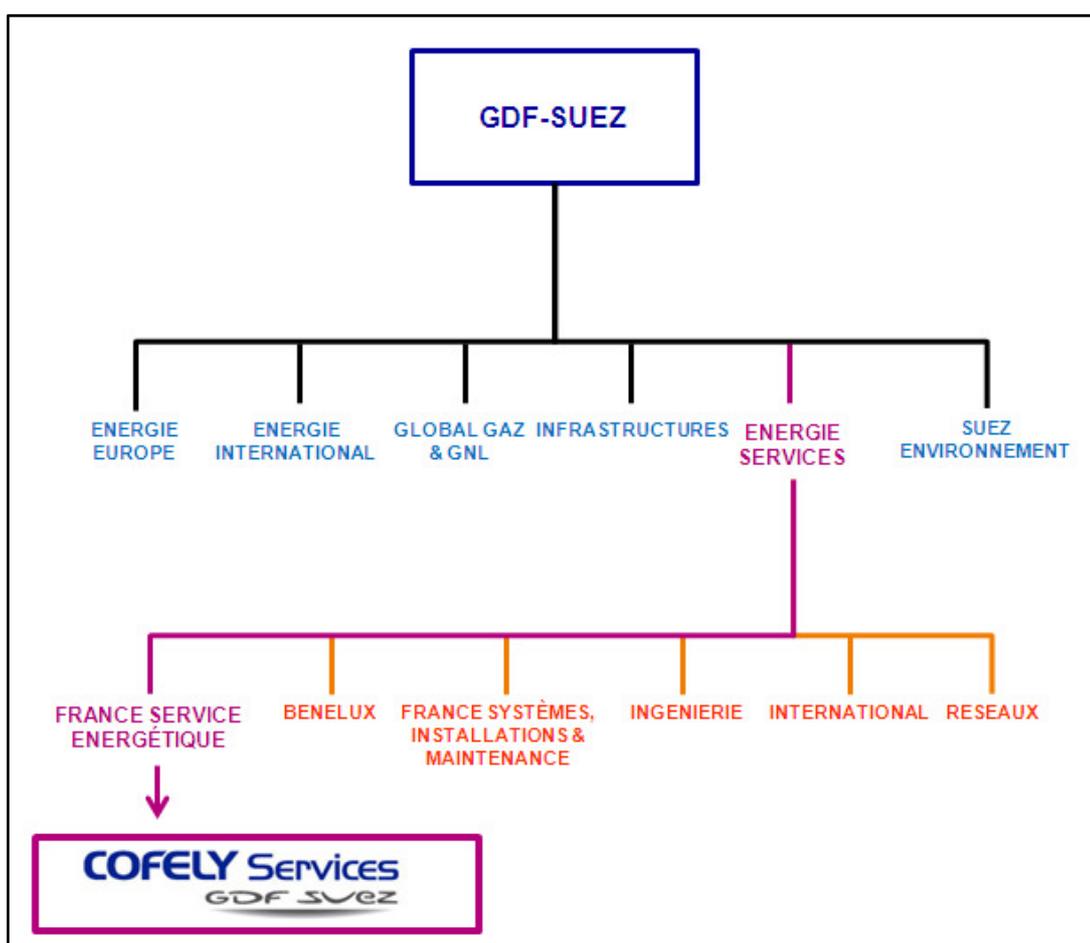


Figure 1. Place de COFELY Services au sein de GDF-Suez

Historiquement, l'Agence Activités Nucléaires était un département de la société Sochan. Elle s'appelait d'ailleurs à l'époque « Département MEC » pour « Maintenance en Environnement Contrôlé ».

Son cœur d'activité était déjà la maintenance Process industriel de groupes frigorifiques en milieu nucléaire très protégé : les CNPE, Centre Nucléaire de Production d'Électricité. E.D.F. était par conséquent le client quasi-exclusif du Département MEC.

En 1999, Sochan fusionna avec les sociétés Someth, Génèse, Cinerg et Pictet pour former, au sein de Gaz de France, la société COFATHEC. Le Département MEC a donc rejoint COFATHEC Services, la division spécialisée dans les services énergétiques de Gaz de France.

En 2008, Gaz de France fusionne avec Suez ; leurs filiales respectives devront également fusionner par la suite.

C'est donc en 2009 que COFATHEC Services, filiale d'ex-GDF, fusionne avec Elyo, filiale d'ex-Suez, pour donner naissance à COFELY et être intégré au sein de la branche spécialisée dans les services à l'énergie de GDF Suez : GDF Suez Énergie Services.

Si l'identité de l'agence a beaucoup évolué à travers le temps au gré des fusions de société, son activité et sa clientèle n'ont pas changé.

Aujourd'hui encore, EDF reste le client quasi-exclusif de l'Agence Activités Nucléaires et les équipes d'exploitation de l'Agence sont postés sur les différents CNPE de France. Le personnel administratif est quant à lui concentré au siège de l'agence, à Gonfreville l'Orcher.

L'activité de l'agence s'est ensuite étendue à des activités de maintenance et d'exploitation de réseaux de froid plus traditionnels (climatisation de bureau ou de

bâtiments semi-industriels) nécessitant les habilitations et autorisations requises pour intervenir sur les sites des CNPE (habilitations propres au domaine du nucléaire et à la radioprotection).

L'essentiel de l'activité de l'agence est donc d'assurer la maintenance du parc de groupes frigorifiques assurant le refroidissement et la stabilité des réacteurs nucléaires. Si ces équipements ne sont pas directement impliqués dans la réaction nucléaire, ils sont essentiels au bon fonctionnement et à la sûreté de l'installation. En cas de dysfonctionnement, un arrêt de tranche (arrêt d'une partie de la centrale) peut être décidé par le directeur de la centrale.

À noter que le récent accident nucléaire de Fukushima Daiichi est directement lié à une défaillance du système de refroidissement et des groupes électrogènes diesel de secours sensés les alimenter en cas de coupure d'urgence des réacteurs.

Le personnel de l'agence est donc réparti entre :

- Le personnel en agence : personnel administratif, management, services méthodes, qualité, etc.
- Le personnel posté se trouvant dans chaque centrale nucléaire : responsables de site et techniciens affectés à un CNPE.
- Les équipes « travaux » et « diesel » qui interviennent sur les différents CNPE dans le cadre de gros travaux (budgets pouvant aller jusqu'à plusieurs millions d'euros).

Tous accèdent à l'application et l'utilisent au quotidien pour l'essentiel de leur activité.

L'agence se doit également de suivre scrupuleusement la Note Technique 85/114. La NT 85/114 est un ensemble de prescriptions particulières, imposées par E.D.F. à tous les fournisseurs de service opérant dans les centrales nucléaires (norme du système qualité d'E.D.F.).

L'application est donc formatée pour suivre ces contraintes et fournir toute la documentation nécessaire à la réalisation de l'activité dans le respect de ces prescriptions particulières.

Nous allons à présent nous intéresser à l'application en elle-même, sa genèse, son développement et les limites qui ont nécessité la mise en place du projet de migration.

1.2. La GMAO AssystMaint

1.2.1. Les logiciels de Gestion de Maintenance Assistée par Ordinateur (G.M.A.O)

Une G.M.A.O. est un outil intégré au système d'informations d'une entreprise dont l'objectif est d'assister les entreprises dans leurs activités de maintenance.

En général, on retrouve systématiquement les fonctionnalités suivantes au sein d'une G.M.A.O. :

- gestion des équipements maintenus : inventaire, localisation chez le client et informations techniques,
- gestion de la maintenance : planification des interventions préventives et gestion des interventions correctives (avec un éventuel module de gestion de Demandes d'Interventions),
- gestion des stocks : pièces de rechange et consommables,
- gestion des devis : du devis proposé au devis final, accepté par le client,
- gestion des achats : fourniture de pièce, demande de sous-traitance, etc.,
- gestion du personnel : gestion de l'effectif et pointage des heures,
- gestion du budget : analyse des coûts de chaque chantier (main d'œuvre, pièces, sous-traitance, etc.),
- indicateurs clefs de performance : tableau de bord d'aide à la décision (technique ou financier).

L'emploi d'une G.M.A.O. permet en général une amélioration de la fiabilité et de la disponibilité des équipements maintenus, une meilleure planification des opérations de maintenance et des possibilités de Retour d'Expérience (REX) accrues grâce à l'historique des travaux réalisés.

Au niveau financier, elle permet également une meilleure gestion des coûts (rationalisation des achats, suivi des coûts de sous-traitance, etc.) et des stocks.

D'une manière générale, c'est une aide précieuse à la prise de décision, qu'elle soit technique (renouvellement de matériel, par exemple) ou financière (analyse budgétaire).

Enfin, la frontière avec un E.R.P. (Enterprise Resource Planning ou progiciel de gestion intégré) est souvent mince car les applications de G.M.A.O. sont souvent obligées de couvrir plusieurs domaines fonctionnels de l'entreprise (a minima : RH, finance et technique).

1.2.2. La G.M.A.O. de l'agence : AssystMaint

L'application AssystMaint a donc initialement été développée en autodidacte par le directeur de l'agence, Joël REVERT sous la forme d'une application Access, tant pour l'IHM que pour le SGBD. Cette technologie était alors adaptée au besoin de l'agence et au contexte d'utilisation : environnement restreint (une vingtaine d'utilisateurs distants et quelques utilisateurs en local) et facilité de développement (notamment pour la mise en place de documents adaptés aux contraintes du client principal, E.D.F.).

L'application reprenait toutes les fonctions habituellement échues aux logiciels de G.M.A.O. tout en tenant compte des spécificités de l'activité de l'Agence (essentiellement concernant la radioprotection et la démarche qualité E.D.F.).

Elle a donc été enrichie au fur et à mesure des besoins de l'agence depuis plus de 15 ans et a suivi l'évolution de l'agence :

- ajout de nouvelles fonctionnalités permettant à l'application de couvrir aujourd'hui l'intégralité de l'activité technique de l'agence,
- respect continu des procédures du client et de leurs évolutions,
- nombre d'utilisateurs en forte croissance (à l'heure actuelle, une trentaine d'utilisateurs à l'agence et plus d'une centaine d'utilisateurs distants répartis sur les différents sites postés au sein des CNPE, soit une progression de presque 500% du nombre d'utilisateurs).

Aujourd'hui, en plus des fonctions de gestion de l'activité, l'application couvre des fonctions de gestion administrative, de gestion du personnel et de gestion des habilitations.

Ces fonctions communiquent ensemble pour assurer un suivi efficace et pertinent de l'activité de l'agence.

Tous ces éléments ont rendu l'utilisation d'Access inadaptée aux enjeux de l'activité que l'application permet de gérer et ont poussé la DSI du groupe à mettre en place le projet de sécurisation et de pérennisation de l'application dont il est question ici.

En particulier :

- L'application contient des données à caractères sensibles, tant pour le client que pour l'agence : la sécurité de l'accès à ces données n'était pas garantie et aucune traçabilité d'accès n'était en place.
- Le schéma de la base de données s'est grandement complexifié : Access montrait ses limites en termes de performance.
- Le nombre d'utilisateurs simultanés a beaucoup augmenté : une fois encore Access montrait ses limites, cette fois, en termes de contrôle d'accès simultané.
- La pérennité de l'application était trop dépendante de la disponibilité et de la compétence de quelques rares personnes de l'agence, en particulier du directeur.
- La mise en production des nouvelles versions était particulièrement lourde : elle impliquait de passer physiquement sur les postes du personnel en agence et de mettre à jour la version utilisée via Citrix par les utilisateurs distants.
- Le serveur était hébergé au sein même de l'agence, sans processus de reprise d'activité préétabli en cas de sinistre.

Tous ces éléments ont donc forcé l'évolution de l'application vers des technologies plus adaptées, plus robustes, plus sûres.

Le contexte du projet étant défini, nous pouvons maintenant nous intéresser au projet en lui-même : ses contraintes et son contenu.

2. LE PROJET

Comme tout projet de ce type, sa finalité est un subtil équilibre entre les contraintes imposées par les directions du groupe et de l'agence et les objectifs ayant conduit à la mise en place du projet.

Il convient en plus d'assurer une compatibilité complète entre l'ancienne architecture de l'application et la nouvelle.

L'interruption de service devait être minimale et la migration devait être la plus transparente possible pour les utilisateurs.

Enfin, lors de l'étape de migration, l'opération devait s'effectuer à périmètre fonctionnel constant (les fonctionnalités métier de l'application devaient rester identiques).

Il convient donc de détailler dans un premier temps les contraintes qui ont pesé sur le projet avant de détailler l'architecture d'origine de l'application.

Ce sont ces deux éléments qui ont permis d'aboutir à l'architecture finale de l'application.

Nous présenterons ensuite l'architecture cible de l'application une fois la première étape du projet terminée (la migration des données).

Enfin, nous nous intéresserons aux acteurs du projet, en précisant leur rôle.

2.1. L'architecture de départ de l'application

2.1.1. Architecture logicielle

Dans sa configuration initiale, le logiciel était constitué des éléments suivants :

- Un fichier MDB Access, au format Access 97, contenant toutes les données de l'application,
- Plusieurs fichiers MDE Access, au format Access 2003, chaque fichier correspondant à un « module » de l'application.

Chaque module permet de gérer une partie de l'activité (matériels maintenus contrat par contrat, création des dossiers d'intervention, gestion du personnel, stockage de la documentation technique, etc.).

Les différents modules, au format Access 2003, communiquaient avec le fichier de base de données Access 97 par le système de « liaison » propre à Access.

L'accès à l'application s'opérait via un serveur Citrix pour les utilisateurs distants ; quant aux utilisateurs locaux, ils avaient une copie de chaque module sur leur poste. L'application, dans sa forme initiale, ne gérait que de façon très limitée la sécurité et le contrôle d'accès. Tout utilisateur identifié sur le réseau pouvait lancer l'application sans avoir à saisir d'identifiants propre à l'application. Seule la couche Citrix pour les utilisateurs distants nécessitait d'entrer des identifiants.

De plus, une fois connecté, l'utilisateur avait accès à tous les modules de l'application. Seuls quelques rares modules étaient protégés par un mot de passe « Access » à la fiabilité toute relative.

Pour finir, aucune traçabilité, tant sur la connexion à l'application que sur la modification des données, n'était en place.

Au démarrage du projet, 21 modules applicatifs étaient disponibles :

- Base de données « Matériels » : ce module permet de saisir les informations techniques de chaque matériel maintenu par l'Agence Activités Nucléaires.
- Plan de maintenance : ce module permet de préparer les différentes visites préventives qui seront réalisées dans le cadre de nos dossiers de réalisation de travaux.
- Documents de maintenance : ce module permet de gérer les gammes et les procès-verbaux d'intervention. Les gammes sont les procédures détaillées des différentes opérations de maintenances réalisées par nos équipes techniques.
- Dossier de Réalisation de Travaux (DRT) sur matériel : ce module permet de créer et d'éditer les dossiers d'intervention portant sur un matériel spécifique.
- Dossier de Réalisation de Travaux (DRT) sur local : ce module permet de créer et d'éditer les dossiers d'intervention portant sur un local (un local regroupe plusieurs matériels).
- Dossier de Réalisation de Travaux (DRT) sur bâtiment : ce module permet de créer et d'éditer les dossiers d'intervention portant sur un bâtiment donné (un bâtiment regroupe plusieurs locaux).
- Pointages : ce module permet le pointage des variables de paye et des heures de travail réalisés par les employés. Le pointage des heures a un double rôle : il permet de remonter les informations nécessaires au service paye du groupe et permet également de refacturer les interventions qui ne sont pas forfaitisées.
- Devis : ce module permet la création, la réactualisation et le suivi des devis créés par nos équipes d'exploitation.
- Dossiers « OT » SAP : ce module permet de suivre l'avancement financier des différents chantiers réalisés par nos équipes d'exploitation. Il permet également de monter les budgets d'une année sur l'autre.
- Demandes d'achat : ce module permet aux équipes d'exploitation d'émettre des demandes d'achats (de fourniture, de sous-traitance, de fluide frigorigène, etc.) qui, une fois validée par la chaîne hiérarchique, permettent au service Approvisionnement de commander les éléments requis.

- Personnel : ce module permet de gérer les informations relatives au personnel et notamment les différentes habilitations obligatoires pour intervenir en centrale nucléaire.
- Gestion étalonnage / outillage : ce module permet de gérer l'outillage et de suivre l'étalonnage des outils, dont l'utilisation est tracée lors des interventions.
- Menu général : ce module est le menu permettant de lancer les autres modules de l'application.
- Environnement : Ce module permet de réceptionner le fluide acheté par les équipes d'exploitation et de gérer les fiches de mouvements de fluide et d'huile.
- Gestion des FNC : Ce module permet de créer et suivre les Fiches de Non Conformités (document du système qualité EDF).
- Gestion des OTP : ce module permet de gérer les caractéristiques des différents contrats de l'agence.
- Gestion des analyses de risques : ce module permet la gestion des risques (avec les parades associées) présent dans la bibliothèque de risques disponibles pour la création des analyses de risque.
- Gestion des congés : ce module permet de tracer et de suivre les congés et RTT du personnel.
- Gestion des astreintes : ce module permet de créer les plannings d'astreintes des différentes zones d'astreintes de l'agence.
- Module « Méthodes » : ce module permet d'accéder à toutes les données non confidentielles de l'application (visualisation directe du contenu des tables) et la création de requêtes personnalisées.
- Gestion des DI Pilotimmo : ce module permet de créer, suivre et solder les Demandes d'Intervention des différentes directions immobilières d'EDF (« DI Pilotimmo : protocole propre à EDF).

2.1.2. Architecture matérielle

L'environnement matériel était hébergé au sein même de l'agence.

Le fichier de base de données (fichier au format Access 97) était stocké sur le serveur bureautique de l'agence (serveur de fichiers et DHCP de l'agence).

L'accès des utilisateurs distants était géré par un serveur Citrix, également présent à l'agence.

Quant aux utilisateurs locaux, une copie des fichiers Access des modules (fichiers MDE au format Access 2003) était présente sur chaque poste et communiquaient avec le fichier de base de données par l'entremise d'une liaison Access via le chemin UNC (Universal Naming Convention) du fichier de base de données.

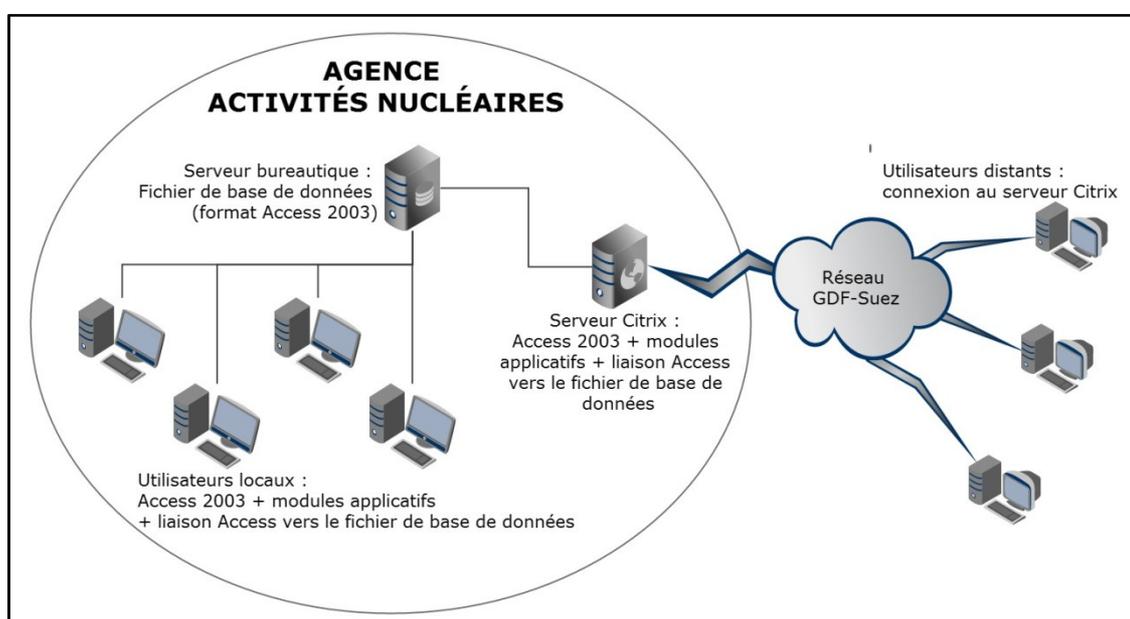


Figure 2. Architecture matérielle d'origine

Plusieurs inconvénients majeurs étaient liés à cette architecture :

- aucune sécurité sur l'infrastructure : pas de redondance, pas d'environnement protégé,
- des mises à jour fastidieuses : il fallait mettre à jour le serveur Citrix ainsi que la vingtaine de postes d'utilisateurs locaux à chaque modification de module,
- performances dégradées pour les utilisateurs distants : la connexion des utilisateurs distants était tributaire de la liaison ADSL de l'agence.

2.2. Les contraintes du projet

Plusieurs objectifs et contraintes ont été fixés conjointement par la DSI du groupe et par l'agence Activités Nucléaires afin de dessiner les contours de l'application une fois le projet terminé.

Des contraintes techniques, pour que l'application s'interface avec les technologies utilisées au sein du groupe et permettant à l'agence de garder le contrôle de certains éléments de l'application, des contraintes fonctionnelles, pour que les insuffisances de l'application remontées lors de l'étude soient corrigées et des contraintes organisationnelles pour que la pérennité de l'application soit assurée.

Nous verrons d'ailleurs que certaines contraintes organisationnelles, n'ont pas été totalement respectées car elles alourdissaient les processus d'évolutions de l'application au-delà des limites acceptées par la direction de l'agence Activités Nucléaires.

De plus, ces lourdeurs avaient un coût non négligeable pour l'agence qui venait s'ajouter au budget prévu initialement.

Tous ces éléments (manque de réactivité et coût important) ont poussé la direction de l'agence Activités Nucléaires à court-circuiter les procédures prévues initialement.

2.2.1. Les contraintes techniques

Les principales contraintes techniques ont été imposées par la DSI du groupe GDF-Suez. L'objectif étant d'intégrer parfaitement l'application au sein du SI, les technologies Microsoft employées par le SI ont donc été largement favorisées.

Ainsi l'identification des utilisateurs par l'application devait passer par le service d'annuaire LDAP (Lightweight Directory Access Protocol) de Microsoft : *Active Directory*. C'est par ce biais que chaque utilisateur du groupe GDF-Suez est

identifié dans le SI. Il permet de donner à chaque utilisateur l'accès ou non à l'application, ainsi que d'attribuer ses droits et permissions au sein de l'application.

Le pilier central de l'application étant sa base de données relationnelle, le SGBDR (Système de Gestion de Bases de Données Relationnelles) fut également imposé : *Microsoft SQL Server*.

La version 2008 de SQL Server était disponible depuis peu mais afin d'éviter l'emploi d'une version encore jeune, c'est la version 2005 qui fut retenue.

Elle présentait notamment l'avantage d'être connue des équipes techniques du SI et de faciliter sa mise en place.

Les interactions fortes entre *Active Directory* et *SQL Server* ont permis de mettre en place l'authentification des utilisateurs, la traçabilité des connexions ainsi qu'une gestion fine de profils utilisateurs reposant sur deux paramètres : fonctionnels et géographiques.

De plus, ces éléments ont pu être mis en place sans modifier la partie IHM du logiciel, en ne se reposant que sur les caractéristiques techniques d'*Active Directory* et de *SQL Server* (via l'authentification Windows et les vues SQL).

Enfin, pour garder le contrôle de l'interface et de la documentation extraite du logiciel, l'agence AAN a exigé de conserver Access pour la partie front-end.

Cela permet à l'agence de faire évoluer le logiciel et la documentation, notamment suite aux changements de législation en matière environnementale.

2.2.2. Les contraintes fonctionnelles

Le projet devait garder le périmètre fonctionnel de l'application. Toute l'activité de l'agence étant déjà gérée par l'application, il n'était pas question de toucher à cet équilibre.

En revanche, deux éléments majeurs faisaient cruellement défauts à l'application : le contrôle de l'accès à l'application et à ses données ainsi que la traçabilité des connexions à l'application et des modifications faites sur les données.

Le premier était donc le contrôle de l'accès à l'application.

Le cahier des charges du projet incluait donc un système complet d'authentification devant permettre à l'application de reconnaître chaque utilisateur, de le faire appartenir à un groupe d'utilisateurs et d'appliquer à son groupe un ensemble de droits dans l'application.

Ces droits comportaient l'accès ou non à chaque module ainsi que les droits à l'intérieur du module concerné (lecture/modification/suppression des données).

L'autre spécification fonctionnelle ajoutée fut la traçabilité.

Les connexions à l'application sont tracées (qui se connecte, à quel module et à quel moment ?) ainsi que les modifications ou suppressions portant sur des données jugées critiques (qui a modifié quelle donnée et à quel moment ?).

2.2.3. Les contraintes organisationnelles

Outre les faiblesses techniques de l'application qui devaient être corrigées par les choix technologiques réalisés en amont et l'ajout de fonctions d'authentification et de traçabilité, le fonctionnement de la maintenance de l'application devait également être revu.

À l'origine, comme nous l'avons vu, l'application a été développée en autodidacte par le directeur de l'agence, Joël REVERT. Toute l'application et donc l'activité de l'agence reposait donc sur ses compétences et sa disponibilité en cas de problème.

Un des objectifs du projet était donc de corriger cette fragilité en confiant la maintenance et les évolutions futures du logiciel à une société de services en ingénierie informatique, après la première étape de migration des données vers SQL Server. L'agence ne devait conserver que la modification des états de sortie Access (documentation interne et documentation client).

2.3. L'architecture cible de l'application

L'objectif du projet était donc de sécuriser, fiabiliser et pérenniser l'application. Plusieurs grands axes ont permis de tendre vers ce triple objectif, notamment l'architecture même du logiciel.

2.3.1. Architecture matérielle

La première étape dans la sécurisation et la fiabilisation de l'application fut d'assurer l'intégrité physique du matériel.

L'hébergement de l'application a donc été transféré de l'agence à un datacenter du groupe GDF-Suez. Serveurs Citrix et serveur SQL Server ont donc rejoints les autres applications-clefs du groupe au sein du Data Center Parisien (DCP) de GDF-Suez.

L'application a donc profité de toutes les qualités inhérentes à un datacenter :

- Contrôle de l'environnement (climatisation, alimentation, risque incendie, etc.),
- Sécurité physique (accès restreint aux serveurs, télésurveillance, etc.),
- Performance du serveur et bande-passante accrue.

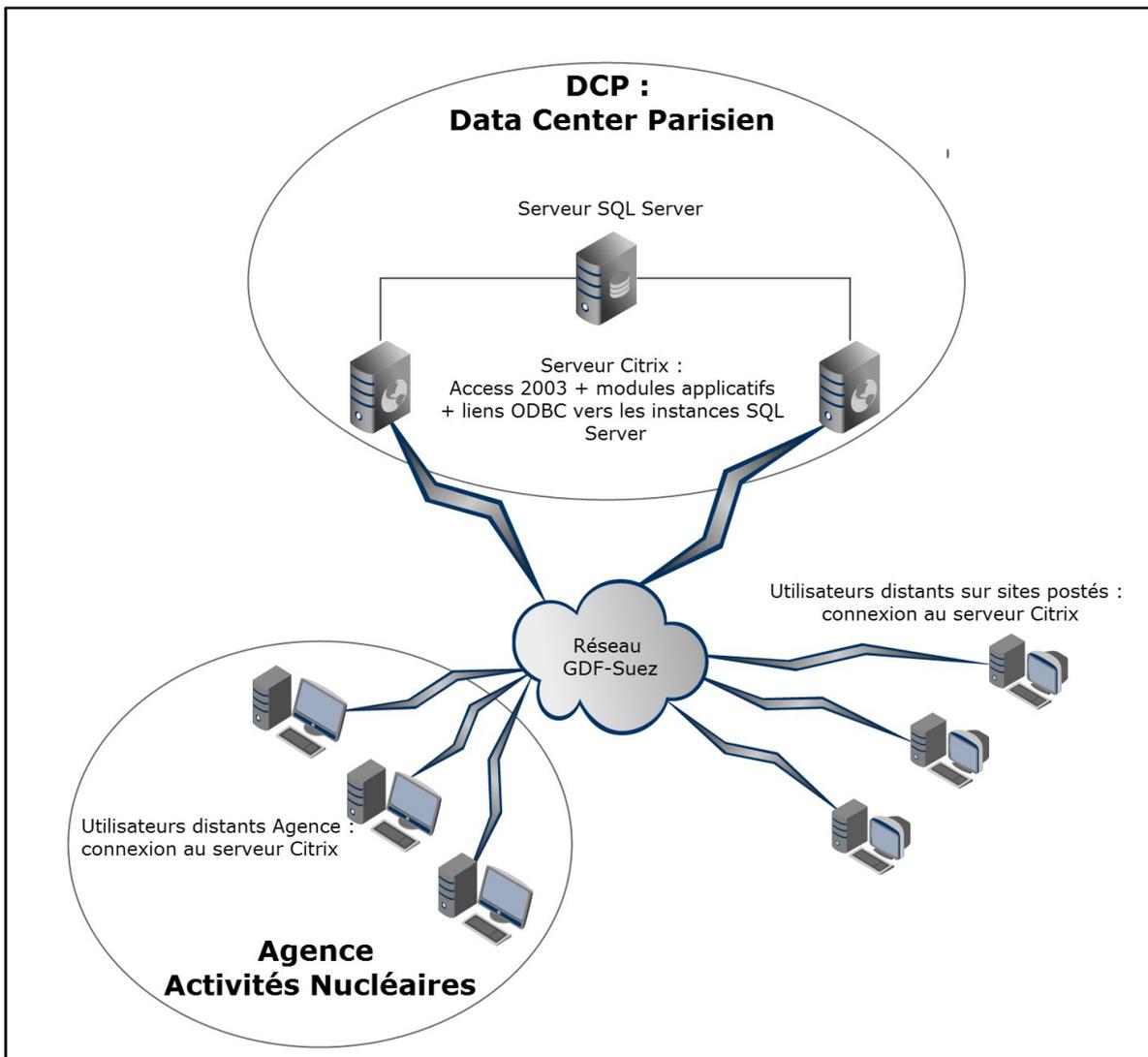


Figure 3. Architecture matérielle post-migration

L'utilisation de Citrix s'est donc généralisée à tous les utilisateurs de l'application. Concomitante à l'installation de la fibre optique au sein de l'agence AAN, le transfert n'a pas eu d'impact significatif sur le temps de réponse de l'application pour le personnel en agence.

En revanche, pour les utilisateurs postés sur les différents CNPE, le gain de performance fut notable.

En parallèle de ces modifications sur l'environnement de production, des environnements de développements et d'intégration à l'identique de l'environnement de production ont également été mis en place.

2.3.2. Architecture logicielle

La modification de l'architecture logicielle fut le principal levier pour assurer à l'application sa sécurisation et sa fiabilisation.

Le passage d'Access et de son moteur de SGBD « Jet » à SQL Server permis un bond significatif en termes de performance, mais surtout en termes de sécurité.

En effet, afin de limiter au maximum les modifications sur l'interface du logiciel, toute la gestion du contrôle d'accès à l'application, à ses différents modules et aux données devait être gérée par le SGBD par l'entremise de ses liens forts avec l'annuaire *Active Directory*. La solution retenue fut donc la mise en place de vues SQL.

Une vue SQL est une table virtuelle créée par le SGBD servant d'intermédiaire entre les modules de l'application et les tables de la base de données.

L'emploi de ces vues permet de régler finement les droits en lecture/modification/suppression des données en fonction du module d'où les données sont appelées et des droits attribués aux utilisateurs sur les différentes vues.

Les droits en questions sont donc appliqués aux vues et non aux tables elles-mêmes.

Concrètement, cela permet par exemple au service RH et Formation d'éditer les informations personnelles d'un employé par l'intermédiaire du module de gestion du personnel (droit en modification et suppression sur une vue de la table « Employés ») tout en permettant aux équipes opérationnelles d'accéder en lecture seule à une partie de ces informations (droit en lecture seule sur cette même table par l'intermédiaire d'une autre vue filtrant les informations confidentielles).

Elle permet également de filtrer certaines informations en fonction du niveau de confidentialité requis.

Le schéma de l'application s'est donc considérablement complexifié par rapport à son architecture d'origine.

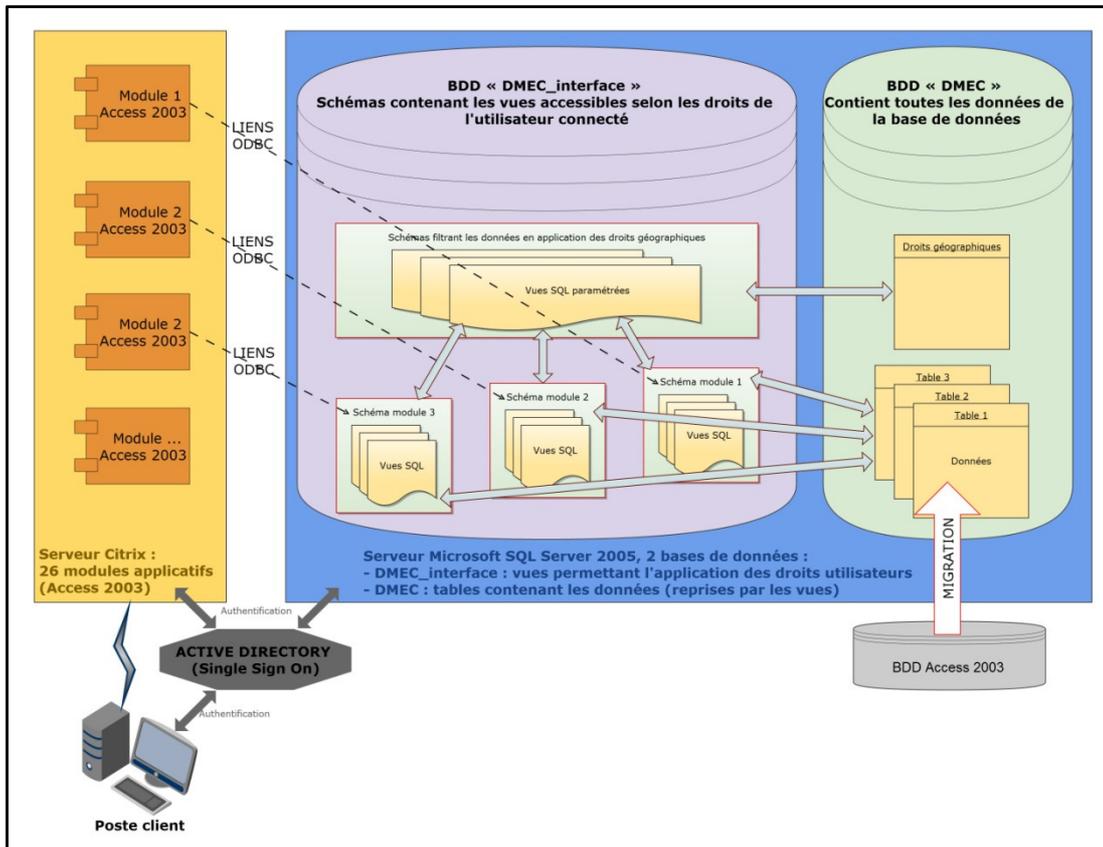


Figure 4. Architecture logicielle post-migration

3. LE DEROULEMENT DU PROJET

Nous avons donc vu le contexte ainsi que les contraintes qui ont défini ce projet, il est maintenant temps de voir le déroulement du projet de façon plus concrète.

De la prise en main du logiciel à mon arrivée au sein de l'agence à la mise en production de la nouvelle mouture du logiciel, de nombreuses étapes se sont succédées.

Nous verrons donc le travail préalable qui a été nécessaire, la phase de sélection puis de collaboration avec le prestataire choisi pour la migration et la tierce maintenance applicative, les tests avant mise en production et la mise en production le jour J.

3.1. Le travail préalable

3.1.1. Rédaction du cahier des charges et sélection du prestataire

Comme nous l'avons vu précédemment, une des contraintes imposées par la DSI du groupe était de confier la migration des données à un prestataire extérieur.

La première étape a donc été de formaliser un cahier des charges complet afin de lancer un appel d'offre à destination des SSII voulant répondre à notre demande.

Deux points principaux figuraient au sein de ce cahier des charges :

- la migration des données d'Access vers SQL Server 2005,
- la proposition d'une solution de sécurisation de l'application avec identification fine des utilisateurs.

De plus, l'appel d'offre ouvrait la possibilité pour le prestataire qui effectuerait la migration d'obtenir par la suite la maintenance du logiciel.

L'objectif de la sécurisation de l'application répondait à deux besoins majeurs :

- utiliser l'annuaire Active Directory pour paramétrer l'accès à l'application et son déploiement sur les postes des utilisateurs,
- attribuer des droits en lecture/modification/suppression, selon le profil de chaque utilisateur et le module utilisé.

Trois prestataires ont répondu à notre offre : Cap Gemini, Steria et Activ'Alo.

Cap Gemini et Steria sont deux acteurs majeurs dans le domaine de l'informatique en France et Activ'Alo était la société de service qui fournissait à l'agence l'ancien serveur hébergeant l'application.

Si Activ'Alo fut rapidement écarté (le contenu de l'appel d'offre dépassait leurs compétences), les réponses de Cap Gemini et de Steria étaient toutes deux différentes et pertinentes.

Cap Gemini proposa de reprendre l'IHM de l'application et d'intégrer au sein de l'IHM les mécanismes de contrôles d'accès à l'application et à ses différents modules.

Steria proposa d'utiliser SQL Server et ses vues SQL pour faire reposer la sécurisation sur les mécanismes propres à SQL Server.

La solution de Cap Gemini était plus fine et permettait une grande flexibilité dans l'attribution des droits aux utilisateurs mais reposait sur des développements VBA intégrés à l'IHM de l'application.

La solution de Steria proposait une flexibilité moindre mais reposait quasi intégralement sur SQL Server. De plus, le temps de développement de leur solution était largement inférieur à celui de Cap Gemini, tout comme le tarif de leur prestation.

C'est donc la solution de Steria qui fut retenue.

Son atout majeur était de faire reposer les mécanismes de sécurisation sur SQL Server et non sur Access, ouvrant la possibilité de conserver le travail réalisé sur la base de données même en cas de changement d'outil pour la partie IHM de l'application.

Une fois le prestataire sélectionné, le travail en collaboration avec le prestataire a pu commencer.

3.1.2. La préparation de la migration des données

Un des objectifs principaux du projet fut la migration des données depuis Access vers SQL Server 2005. Access étant beaucoup plus permissifs sur les contraintes d'intégrité, il a fallu au préalable corriger les lacunes du schéma de base de données de l'application.

Access et SQL Server étant deux produits Microsoft, la tâche fut facilitée par l'existence d'un assistant de migration intégré au sein de SQL Server et permettant de générer le schéma et d'importer les données du fichier MDB Access vers une base de données SQL Server nouvellement créée.

Cet assistant relevait régulièrement des incohérences de schéma qu'il fallait corriger, à la source, sur le fichier MDB accueillant les données de production de l'application avant la migration.

La première tâche concrète du projet fut donc de réaliser la migration des données depuis Access vers SQL Server ; elle s'est concrétisée par des échanges entre notre prestataire, Steria et l'agence Activités Nucléaires.

Steria nous indiquait les éléments du schéma de base de données qui étaient refusés par SQL Server et l'agence appliquait en amont, sur la base Access, les corrections nécessaires.

L'essentiel fut la suppression et la mise en place de relations d'intégrité conformes aux standards SQL ; en particulier la suppression des références circulaires et la correction de données provoquant des doublons sur des index ou sur des clefs étrangères.

3.2. Le suivi des développements du prestataire

Une fois la base Access corrigée, afin de se conformer aux standards SQL mis en œuvre dans SQL Server, Steria a pu commencer à développer sa solution de sécurisation.

3.2.1. Le démarrage du projet

La première étape du travail de Steria fut la mise en place de leur solution de sécurisation au sein de SQL Server :

- Création des différents groupes d'utilisateurs en conformité avec le cahier des charges,
- Création des schémas de base de données sur lequel les droits des différents groupes utilisateurs seront appliqués,
- Création des vues qui sont ensuite rattachées aux différents schémas,
- Mises en place des droits relatifs aux groupes utilisateurs et aux schémas.

Une fois cette première étape terminée, Steria a pu commencer à s'intéresser aux différents modules d'IHM.

L'application étant découpée en modules fonctionnels correspondant chacun à un fichier Access d'IHM, Steria a commencé la modification en l'appliquant à un premier module.

Les modules devaient être modifiés pour permettre leur connectivité à la base de données SQL Server, la notion de table liée propre à Access a donc été remplacée par des liens ODBC pointant sur les vues SQL.

Une fois cette seconde étape réalisée, Steria nous a fait son premier retour sous la forme d'une base de données SQL Server reprenant les données du fichier Access de base de données originel et un module applicatif modifié pour communiquer avec cette nouvelle base.

C'est avec ces premiers éléments que nous avons pu lancer les procédures de test.

3.2.2. Les procédures de test

Sur la base de ce premier module et de la base de données SQL Server migrée, nous avons donc écrit puis effectué des tests pour valider le fonctionnement des mécanismes d'identification et d'attribution des droits.

Il a fallu également s'assurer de toute la mécanique de distribution de l'application : gestion des droits dans l'Active Directory et serveur Citrix.

C'est d'abord la configuration de production du serveur Citrix qui a été mise en place.

Pour que l'application fonctionne, le serveur Citrix devait avoir Access installé ainsi qu'une connexion ODBC vers le serveur SQL Server.

Des utilisateurs test ont ensuite été créés dans l'Active Directory afin de valider le déploiement automatique de l'application sur le bureau des utilisateurs.

Les membres du groupe de déploiement de l'application devaient recevoir directement sur leur bureau l'icône de lancement, correctement paramétrée.

Une fois ces points validés, nous avons pu commencer à tester la solution de sécurisation de l'accès aux données de Steria sur le serveur d'intégration.

Ces tests consistaient en des tentatives de connexions à l'application en utilisant les différents profils d'utilisateurs prévus dans l'application.

Chaque profil ayant des droits et accès différents, il a fallu tester les accès à chaque module (21 modules) pour chaque profil (10 profils différents).

Ces tests représentaient donc 210 tentatives de connexion à réaliser, accompagnées de quelques actions clés par module pour vérifier le bon fonctionnement des verrous en lecture/ajout/modification.

Dans un premier temps, les tests ont été réalisés sur un premier module afin de valider le fonctionnement général de la solution de sécurisation mise en place par Steria.

Une fois le fonctionnement de leur solution de sécurisation validé sur un premier module, Steria a pu déployer leur solution sur l'ensemble des modules de l'application et les tests se sont poursuivis.

Ces tests étaient assez longs et fastidieux. Ils ont donc été conduits par un personnel intérimaire, embauché pour l'occasion.

C'est à ce moment-là que sont apparues des divergences entre Steria et l'agence.

3.2.3. Les premières divergences avec le prestataire

Le fonctionnement prévu avec Steria lors de la phase de migration prévoyait que les modules soient figés, une fois livrés à Steria pour l'implémentation de leur solution de sécurisation.

L'activité de l'agence étant en évolution constante, que ce soit pour suivre les évolutions réglementaires, les desideratas du client ou de la direction de l'agence, ce système de cristallisation des modules a très vite posé des problèmes à la direction de l'agence Activités Nucléaires.

Certains modules étaient en développement chez Steria pendant plusieurs semaines avant qu'un premier retour soit disponible à l'agence.

S'agissant d'un logiciel développé en interne, l'agence avait toujours été habituée à une très grande réactivité d'évolution des modules et des documents de sortie. D'autant que la migration de l'application a eu lieu pendant la période de fusion de COFATHEC et d'ELYO, période propice à de nombreuses modifications sur le logiciel (notamment sur toute la partie documents de sortie). De plus, pendant qu'un module était en cours de modification par Steria, il était impossible pour l'agence de travailler en parallèle sur les évolutions requises. Pour finir, lorsque certains modules entraient dans ce cas de figure, Steria facturait un supplément d'intervention à l'agence Activités Nucléaires au prix fort.

Tous ces éléments ont concouru à détériorer les relations entre l'agence et son prestataire ; chaque modification donnant lieu à d'âpres négociations sur la prise en charge financière des modifications à impacter, y compris en cas de modification triviale ne nécessitant aucune intervention concrète de notre prestataire.

3.2.4. Poursuite de la migration et retard du projet

En dépit de ces désaccords récurrents sur la prise en charge des évolutions nécessaires de l'application, la migration de l'application se poursuit.

Le planning initialement prévu pâtit de ces allers-retours entre l'agence et Steria mais la principale cause de ce retard fut l'immaturation du logiciel.

En effet, si ma présence à l'agence devait servir à suivre le projet de migration, la direction de l'agence Activités Nucléaires a mis mes compétences à contribution pour améliorer très largement le logiciel.

Ainsi, de nombreux modules ont été entièrement refondus par mes soins avant d'être confiés à Steria dans le cadre du projet de migration.

Tous ces développements de « dernière minute » n'ont pas facilité les échanges avec Steria.

Néanmoins, la migration se poursuit jusqu'à la mise en production avec cependant presque une année de retard sur le planning initialement prévu.

3.3. La mise en production

La mise en production était prévue initialement le 20 mai 2009 mais a finalement eu lieu le 15 février 2010.

L'objectif de la mise en production était de réduire au maximum la durée d'indisponibilité de l'application.

Les effectifs concernés de la DSI, de l'agence et de Steria ont donc été mobilisés lors de la semaine précédant la migration pour effectuer les ultimes tests de la partie Citrix et Active Directory de l'application : configuration du serveur Citrix, création des groupes et des droits utilisateurs allouant l'accès à l'application et déploiement automatique de l'application sur des utilisateurs test.

Une fois ces étapes validées, toutes les équipes de Cofely et de Steria étaient prêtes pour la mise en production.

Afin de répondre à l'objectif d'une indisponibilité minimum de l'application, il a été décidé de procéder à une mise en production de type « Big Bang ».

La notion de mise en production « Big Bang » est une appellation ITIL par opposition à des mises en production « par lot » ou « par phase ».

L'idée est de passer de l'ancienne version du logiciel à la nouvelle en une seule fois.

L'application a donc été coupée pendant 2 jours, du lundi 0h au mardi minuit.

Le premier jour a mobilisé les équipes de Steria, de l'agence AAN et de la DSI du groupe GDF-Suez.

Pendant ce premier jour, le paramétrage final de l'application a été complété, les liens entre le nouveau serveur Citrix, le nouveau serveur SQL et le nouveau

serveur de fichiers hébergeant les modules (fichiers au format Access 2003) ont été montés.

Une fois cette étape terminée, le reste du temps d'indisponibilité de l'application a servi à effectuer d'ultimes tests sur la plate-forme de production.

Le mercredi matin, l'application était active et fonctionnelle, fruit d'un travail de presque 3 années.

4. L'APRES-PROJET

Le fonctionnement prévu initialement dans le contrat que nous avons avec Steria prévoyait que les modules, une fois migrés sur la nouvelle architecture, étaient entièrement pris en charge par Steria, titulaire du contrat de maintenance de l'application.

La relation de confiance avec le prestataire s'étant progressivement étiolée, c'est d'un commun accord entre l'agence et Steria que le contrat de maintenance de l'application a été rompu.

Cette décision fut le seul échec de ce projet qui devait se terminer par une prise en charge externalisée de l'application.

À l'usage, ce mode de fonctionnement s'est avéré incompatible avec les besoins de l'agence Activités Nucléaires.

Outre les surcoûts systématiques de la part de notre prestataire que générait ce mode de fonctionnement, c'est bien l'inertie inhérente à une prise en charge externalisée de l'application lors des évolutions de l'application qui a poussé l'agence AAN à rompre le contrat de maintenance.

Cette décision fut pour moi l'occasion de développer des solutions applicatives qui faisaient cruellement défaut à l'application.

Il est également apparu que Steria n'avait pas optimisé la base de données SQL Server, se contentant de la migration des données.

Cette tâche, capitale pour profiter pleinement des possibilités offertes par SQL Server, permet de gros gains de performance sur les modules les plus complexes.

Nous allons donc voir un exemple d'évolution importante de l'application, la gestion des stocks de fluide et des mouvements de fluides et nous intéresser ensuite à l'optimisation de la base SQL Server.

4.1. Un exemple concret d'évolutions de l'application : la gestion des mouvements de fluides

Plusieurs évolutions majeures ont vu le jour au sein de l'application après la phase de migration des données.

Afin d'illustrer ces évolutions, nous allons étudier la mise en place du système de gestion des stocks et des mouvements de fluide sur les installations prises en charge par les équipes techniques de l'agence Activités Nucléaires.

Cet exemple a l'avantage de bien représenter les différentes contraintes qui pèsent sur l'application : contraintes réglementaires, contraintes liées à la réalité du métier, contraintes liées aux exigences du client et contraintes fixées par la direction de l'agence.

La gestion des fluides frigorigènes est au cœur de l'activité de l'agence.

Il existe de nombreux fluides frigorigènes et ces fluides sont plus ou moins toxiques et plus ou moins dangereux pour l'environnement.

Au début du XX^e siècle, les fluides frigorigènes les plus employés étaient de la famille des CFC (Chlorofluorocarbures). Ils présentaient l'avantage d'être inertes, stables, ininflammables et non toxiques pour l'Homme.

Moins toxiques et plus stables que les fluides les précédant, ils furent néanmoins interdits en raison de leur impact majeur sur la détérioration de la couche d'ozone (Protocole de Montréal, 1987).

Ils furent remplacés par d'autres types de fluides notamment les HFC (Hydrofluorocarbures) qui présentaient l'avantage de ne pas avoir les effets dévastateurs des CFC sur la couche d'ozone. Ces HFC favorisent néanmoins l'effet de serre et ont également été interdits, cette fois lors du Protocole de Kyoto, en application depuis 2005.

Ce récapitulatif très succinct permet d'illustrer les différentes contraintes réglementaires qui pèsent sur l'activité de l'agence.

Les équipements dont nous assurons la maintenance ont utilisé tour à tour ces différentes catégories de fluides et légalement, c'est à l'opérateur de maintenance de s'assurer que la législation en vigueur est bien appliquée.

De plus, l'éthique du métier nous incite à minimiser les impacts sur l'environnement de notre activité.

Afin de contrôler les opérateurs habilités à manipuler ces fluides, le Ministère de l'Environnement demande des statistiques très précises quant à l'achat, l'utilisation, le recyclage et la destruction de ces différents fluides.

Ces données sont recoupées avec les ventes réalisées par les fournisseurs de fluide frigorigène.

Si des discordances dans ces données sont relevées au niveau du Ministère de l'Environnement, l'habilitation permettant à l'agence et à ses 250 employés d'effectuer leur travail de manipulateurs de fluides frigorigènes peut être suspendu.

Ces données permettent également de quantifier les quantités de fluides frigorigènes relâchées dans l'atmosphère afin de vérifier que l'état français respecte bien ses engagements vis-à-vis des différents protocoles relatifs à l'environnement.

Parallèlement à ces contraintes réglementaires, l'application doit également tenir compte de la réalité du métier et des exigences du client.

Notre principal client (EDF) exige de notre part que ces pertes de fluides soient quantifiées.

Ces pertes sont liées la plupart du temps à des fuites dans les circuits frigorifiques et un des objectifs de la maintenance de ses installations est de prévenir l'apparition de ces fuites.

Le principal problème est que cette exigence du client est difficile à évaluer dans les faits.

La seule méthode valable pour quantifier avec précision les quantités de fluide relâchées dans l'atmosphère est de réaliser un suivi précis des quantités de fluide

ajoutées au circuit de chaque matériel et d'aller peser à intervalle régulier la quantité restante dans le matériel.

Cette méthode est relativement lourde à mettre en place et la plupart des frigoristes se contentaient d'estimations plus ou moins précises.

De plus, il est impossible, de prime abord, de distinguer la part de fluide relâchée dans l'atmosphère à cause d'un défaut sur le matériel de celle due à une erreur de manipulation de l'opérateur.

Enfin, la dernière exigence émane de la direction de l'agence elle-même : il s'agissait de trouver une solution logicielle simple et intuitive permettant d'effectuer ce suivi de façon efficace sans sacrifier pour autant les exigences réglementaires et les exigences client. Il était également important de responsabiliser les chefs d'équipe quant au suivi de leur stock de fluides.

À noter également que le logiciel n'implémentait à l'origine aucune gestion des stocks des différents fluides employés par nos équipes.

Cet élément a été très handicapant lorsque la législation à ce sujet fut renforcée suite à l'application du Protocole de Kyoto.

La solution développée fut donc la suivante :

- Chaque équipe possède désormais un stock de fluides clairement quantifié dans la base de données et suivi de près par le responsable d'équipe,
- Le stock de fluides est incrémenté par le biais des demandes d'achat de fluide, réalisées dans la GMAO AssystMaint,
- Chaque appoint de fluide réalisé vient décrémenter le stock du fluide concerné de l'équipe concernée,
- Les quantités de fluide envoyées en destruction ou en recyclage sont également renseignées,
- Les quantités de fluide présente dans chaque matériel à un instant T sont également conservées en base (permettant de fournir à tout instant une information très précieuse à nos clients),
- La notion de « stock négatif » a été bannie.

Cette solution complète de gestion et de suivi des fluides frigorigènes est en place depuis maintenant deux ans et permet à l'agence une précision inédite sur toutes ces données.

Elle permet également d'éditer tous les documents définis par la réglementation en vigueur (Fiches de Mouvements de Fluide et Certificat de Contrôle d'Étanchéité).

De plus, le système mis en place permet d'éditer très rapidement le bilan réglementaire annuel que l'Agence est tenue de fournir chaque année au Ministère de l'Environnement.

L'agence a d'ailleurs été auditée sur son suivi des manipulations de fluide depuis la mise en place de ce système et a reçu des appréciations très élogieuses quant à la rigueur et à la qualité de ce suivi.

Pour finir, le travail de l'agence et de la Direction Régionale de Cofely Centre Ouest (dont nous dépendons) à ce sujet est remontée jusqu'au Ministère de l'Environnement et certaines de nos procédures ont été reprises afin de servir de base pour les exigences ministérielles à venir, notamment pour la partie documentaire. Ainsi, le futur document CERFA qui sera imposé par le Ministère de l'Environnement à tous les manipulateurs de fluides frigorigènes sera très largement inspiré du document produit par Cofely.

Nous avons donc vu un exemple concret de développement logiciel à forte valeur ajoutée pour le bon fonctionnement de l'agence et qui a nécessité un gros travail préparatoire tenant compte de toutes les contraintes qui pèsent sur le logiciel (réglementaires, métiers et hiérarchiques).

Nous allons maintenant voir une deuxième tâche importante, conséquence du projet de migration : l'optimisation de la base de données.

4.2. L'optimisation de la base de données SQL Server

Suite à l'abandon de notre prestataire de service et à ma formation sur la technologie SQL Server, il apparût que leur travail sur notre base de données fut plus qu'incomplet.

Deux éléments majeurs étaient préjudiciables : l'absence d'optimisation de la base de données et des mécaniques de journalisation des événements peu judicieuse.

Il a donc fallu reprendre la base de données, table par table, afin de faire les ajustements nécessaires et réécrire la mécanique de journalisation des événements.

Pour bien comprendre l'importance de l'optimisation de la base de données, nous allons décrire très succinctement les éléments qui constituent un SGBDR et la façon dont ils interagissent avec le matériel avant de décrire concrètement les éléments d'optimisation de la base de données mis en place.

4.2.1. L'optimisation d'un SGBDR

Un SGBDR est constitué des éléments principaux suivants :

- un fichier de données : il enregistre les données de la base sur le disque,
- un fichier de journal : il enregistre les différentes transactions envoyées au serveur par ordre chronologique ainsi que leur état d'achèvement,
- un moteur relationnel : il exécute les requêtes en mémoire,
- un moteur de stockage : il charge en mémoire les données pour l'exécution des requêtes (si ces données ne sont pas déjà présentes en mémoire) et écrit sur le disque les modifications de données des transactions achevées correctement.

Ce qu'il faut retenir de ces éléments, c'est l'importance de la mémoire.

En effet, le moteur relationnel ne travaille que sur les données présentes dans la mémoire vive (Random Access Memory : RAM) du serveur.

L'écriture des données modifiées par des transactions dans le fichier de données est, quant à elle, différée afin d'optimiser les opérations d'écriture (les écritures sont regroupées par lots).

De plus, à l'instar des données dernièrement utilisées, les requêtes les plus récemment utilisées sont également conservées en mémoire.

Lorsque l'on dit d'une requête ou d'une transaction, qu'elle est gardée en mémoire, il s'agit du texte de la requête ainsi que de son plan d'exécution.

Le plan d'exécution correspond aux opérations logiques permettant au moteur relationnel de fournir le résultat d'une requête au client.

Enfin, la gestion de la mémoire obéit à la règle du Last Recent Use, c'est-à-dire que les données et requêtes les plus récemment utilisées sont placées en mémoire et lorsque le moteur relationnel a besoin de nouvelles données, les plus anciennes sont sorties de la mémoire.

La conséquence de ce fonctionnement est qu'une requête dont le plan d'exécution et les données sont encore présentes en mémoire sera très rapide à exécuter.

Une fois que la mémoire est saturée, les performances du serveur chutent significativement car le moteur relationnel doit attendre que les données dont il a besoin soient placées en mémoire.

Le principal axe d'optimisation d'une base de données est donc de s'assurer que les données et requêtes les plus couramment demandées soient toujours stockées en mémoire.

A contrario, la mémoire n'étant pas extensible à l'infini, il faut s'assurer que les données les moins demandées par les clients ne viennent pas encombrer inutilement la mémoire du serveur.

Le premier axe d'optimisation de la mémoire (et donc des performances du serveur SQL) est donc le choix des types de données affectées aux champs des tables.

Un choix de type inapproprié peut avoir des effets pervers sur le traitement des données (comme l'utilisation de nombre à virgule flottante pour des données comptables) ou simplement une surcharge inutile de la mémoire (utilisation du type « nvarchar » au lieu de « varchar » pour des données texte qui n'exploiteront jamais les caractères supplémentaires supportés par le type « nvarchar »).

Une utilisation juste des types de données permet également de limiter le nombre d'entrées/sorties qui seront nécessaires pour traiter une requête. Ces mêmes entrées/sorties qui forment un des principaux goulots d'étranglements dans les performances d'un SGBD.

Pour rappel, les données sont regroupées par « page » dans un SGBD.

Une page va contenir un ou plusieurs enregistrements d'une table selon la taille de chaque enregistrement.

Lorsqu'une requête est exécutée, le SGBD va charger en mémoire les pages contenant les enregistrements dont il a besoin.

Ainsi, plus une page contiendra d'enregistrements, moins le SGBD aura besoin de charger de pages différentes pour exécuter une requête et plus la requête s'exécutera rapidement.

Le deuxième outil permettant d'assurer une utilisation optimale de la mémoire est la création d'index judicieux sur les tables.

Un index est une structure de données redondante qui permet d'ordonner les données indexées afin d'y accéder le plus rapidement possible.

Redondante, car SQL Server va recopier dans des « pages d'index » les données de la table que l'on a choisi d'indexer. Il va ensuite ordonner le contenu de ces pages d'index selon les critères appliqués à l'index.

A cette liste ordonnée, SQL Server va ajouter une colonne contenant un pointeur vers la ligne complète de la table. Le pointeur est en fait une référence au fichier

de données, au numéro de page et au numéro de ligne qui contient l'ensemble des données de la ligne de la table.

Lorsqu'une requête va interroger la table indexée, le moteur d'SQL Server va donc pouvoir parcourir rapidement les pages d'index correspondantes et récupérer ensuite, grâce au pointeur, l'emplacement des données demandées pour aller les lire.

Dans SQL Server, les index sont organisés sous forme « d'arbre B ».

Un arbre B est une structure de données en « arbre équilibré ».

La propriété « équilibrée » d'un arbre B permet d'accéder à tous les éléments de l'index en un nombre d'étapes constant.

Il assure également un grossissement logarithmique de la profondeur de l'arbre (et donc de l'index).

Cela signifie que la profondeur de l'arbre grossit très lentement par rapport au nombre d'éléments indexés : même avec des millions d'enregistrements indexés, la profondeur d'arbre ne sera que de quatre ou cinq.

Il ne faudra donc que 4 ou 5 étapes pour trouver l'élément indexé.

Au final, les index permettent un gain significatif de performance en échange d'une utilisation d'espace mémoire supplémentaire.

À cette définition générale de la notion d'index, nous pouvons ajouter une précision quant à la façon dont SQL Server ordonne physiquement les données en mémoire (ou sur le disque).

En effet, deux méthodes d'organisation des pages de données existent, selon que la table concernée possède un index « ordonné en clusters » ou non.

Avec un index clusters, on « fusionne » le concept d'index avec la table elle-même : les lignes de données de la table sont physiquement stockées dans l'ordre défini par l'index clusters.

Par exemple, si la table des employés possède un index clusters basé sur le nom et le prénom de chaque employé, les données de la table seront physiquement ordonnées alphabétiquement selon le nom puis le prénom de chaque employé.

De plus, chaque page de données fait référence à la page qui la précède et à la page qui la suit. Ce système, appelé « liste à double lien », permet d'insérer facilement une nouvelle page en ne modifiant que les liens de page, sur les pages entre lesquelles la nouvelle page est insérée.

Il permet également, lorsqu'une page est pleine et qu'une donnée nouvelle doit être insérée, de scinder l'ancienne page en deux nouvelles pages, réorganisée via la modification des liens de page.

À noter que la création d'une clef primaire sur une table implique forcément la constitution d'un index clusters, basé sur la clef primaire.

Sans index cluster, les données sont « organisées en tas » (on parle de *heap*, qui signifie « tas » en anglais) : les lignes de données ne sont pas stockées dans un ordre particulier et les pages de données ne sont pas chaînées entre elles par le système de liste à double lien.

Lorsqu'une requête va chercher des informations dans une « table heap », le moteur du SGBDR est obligé de parcourir l'ensemble des lignes de la table : on parle de « table scan ».

Un « table scan » est évidemment bien plus long à réaliser qu'un « index scan » (nous avons vu qu'en 4 ou 5 étapes, un index est capable de mettre à disposition du moteur relationnel une donnée indexée parmi plusieurs millions d'autres données).

Une fois ce fonctionnement assimilé, les mécanismes sous-jacents au concept d'optimisation d'une base de données par les index apparaissent plus clairement.

L'objectif est bien de trouver le meilleur compromis dans la quantité et la qualité des index à implémenter.

Les données testées régulièrement dans le cadre de conditions WHERE devraient systématiquement faire l'objet d'un index accélérant drastiquement l'exécution des requêtes.

Cependant, il ne faut pas non plus tomber dans l'excès inverse qui consisterait à indexer l'ensemble des données de l'ensemble des tables.

Le moteur relationnel d'SQL Server gagnerait certes en efficacité mais la mémoire disponible n'étant pas extensible à l'infini, il passerait également son temps à attendre que le moteur de stockage vienne placer en mémoire les index récemment consultés (ces derniers venant prendre la place des index les plus anciennement consultés, d'après le principe du *Last Recent Use* expliqué précédemment).

Le résultat final serait catastrophique en termes de performance : le principal goulot d'étranglement d'une base de données étant les entrées/sorties des données entre le disque et la mémoire.

Le dernier élément ayant un impact sur le bon fonctionnement d'un SGBDR relève moins de l'utilisation de la mémoire que du processeur : il s'agit de l'usage de procédures stockées.

Les procédures stockées sont de différentes natures mais d'une façon générale, il s'agit de suites d'instructions manipulant les données de la base de données.

Dans SQL Server, il existe deux types majeurs de procédures stockées :

- les fonctions (avec ou sans paramètres d'entrées) qui retournent une valeur ou une table : ce sont des procédures stockées qui n'exécutent une suite d'instruction que lorsqu'elles sont appelées,
- les triggers (connus également sous le nom de déclencheurs) qui sont des procédures stockées particulières se déclenchant automatiquement après un SELECT, un UPDATE, un INSERT ou un DELETE.

Les triggers en particulier peuvent avoir un impact négatif important sur les performances du serveur dans la mesure où des triggers mal conçus vont venir

alourdir les opérations de base qu'un serveur SQL exécute en permanence (SELECT/UPDATE/INSERT/DELETE).

Nous allons donc à présent détailler succinctement ce travail d'optimisation qui a été effectué sur l'ensemble de la base de données.

4.2.2. L'optimisation de la base de données SQL Server d'AssystMaint

Ce travail d'optimisation aurait dû être fait par notre prestataire qui s'était chargé de la migration des données afin de rendre la migration réellement pertinente.

De ce point de vue, leur prestation fut très incomplète et la base de données qu'ils nous ont livrée présentait plusieurs défauts et même de sérieuses lacunes concernant les types de données.

Nous présenterons donc succinctement mais concrètement le travail d'optimisation qui a été nécessaire afin de profiter pleinement des avantages de SQL Server : le travail sur les types de données et le travail sur les index.

a. Le travail sur les types

Comme nous l'avons vu précédemment, le choix des types de données peut avoir un impact sur les performances mais également sur les fonctionnalités de l'application.

Sur la base de données SQL Server que nous a livrée Steria, deux erreurs majeures ont été commises : sur les types *nvarchar* (qui auraient du être typés en *varchar*) et sur les types *float* (qui auraient du être typés en *decimal*).

La première concerne toutes les données de texte contenues dans les différentes tables.

SQL Server permet de stocker les données texte dans 4 types de données différentes dont voici les caractéristiques :

Type de champ	Caractères Unicode	Longueur variable
<i>nchar</i>	Oui	Non
<i>char</i>	Non	Non
<i>nvarchar</i>	Oui	Oui
<i>varchar</i>	Non	Oui

Tableau I. Types de données « texte » dans SQL Server

Unicode est un standard permettant de gérer les échanges de textes dans différentes langues au niveau mondial, il permet notamment de gérer des caractères inutilisés en langue française (idéogrammes asiatiques par exemple).

Ce type de données texte foisonne au sein de la base de données et lors de la migration, toutes les données textes ont été converties en *nvarchar* au lieu d'être converties au format *varchar*.

Toutes les données textes de la base de données occupent donc deux fois plus d'espace que nécessaire, rendant la très grande majorité des requêtes de l'application plus longue à l'exécution.

Si cette première erreur « se contente » de grever les performances de l'application, la deuxième erreur est plus grave car elle induit des comportements erronés dans l'application.

Cette deuxième erreur porte sur les données numériques à virgule.

Il existe deux façons de stocker les données numériques à virgule dans une base de données : en tant que données numériques à virgule flottante (type « float »

sur SQL Server) ou en tant que données numériques à virgule fixe (type « decimal » ou « money » sur SQL Server).

Les virgules flottantes sont des approximations très proches utilisées lorsque des comportements numériques exacts ne sont pas requis ou sont impossibles (notamment en mathématiques).

Au sein d'AssystMaint, les données numériques à virgule utilisés sont exclusivement des prix ou des quantités.

Ces données nécessitent des comportements numériques exacts pour ne pas engendrer de problèmes d'arrondis qui seraient problématiques dans ces domaines (création de budget déséquilibré, calcul de quantité de fluide imprécise, etc.).

Steria aurait donc dû convertir l'ensemble de ces données au format « decimal » ou « money » (application particulière du « decimal » pour les valeurs marchandes).

b. Le travail sur les index

Nous avons expliqué précédemment que des index pertinents peuvent impacter drastiquement les performances d'SQL Server.

Ce travail ne peut s'effectuer qu'après un temps d'utilisation de la base de données, afin de permettre à SQL Server de générer des statistiques d'utilisation (ou de non utilisation) des index.

Toutes ces statistiques peuvent être obtenues à l'aide des Dynamic Management Views de SQL Server (ou DMV).

Les DMV sont des vues systèmes générées par SQL Server qui permettent d'analyser les performances d'un serveur SQL, de diagnostiquer des problèmes de

performance et plus généralement, d'optimiser le bon fonctionnement d'un serveur SQL.

Par exemple, lorsqu'SQL Server traite une requête, il enregistre également les index qu'il tente d'utiliser.

Si l'index en question n'est pas trouvé, SQL Server enregistre les paramètres de l'index qui lui a fait défaut.

Dans notre travail sur les index, la première étape consiste donc à compter le nombre d'index manquants à l'aide de ces DMV.

Cette étape est réalisable grâce à la requête suivante :

```
SELECT DB = DB_NAME(database_id), [Index manquants] = count(*)
FROM sys.dm_db_missing_index_details
GROUP BY DB_NAME(database_id)
ORDER BY 2 DESC;
```

Lors de la première exécution de cette requête pendant la phase d'optimisation des index, SQL Server nous avait signalé 265 tentatives d'appel à des index inexistantes.

Tous n'ont pas automatiquement été implémentés. Il a fallu peser pour chaque index, le coût de l'index (en stockage et en procédure de mise à jour de l'index) et le gain de l'index afin de déterminer si son implémentation était nécessaire.

La deuxième étape consiste ensuite à identifier ces index manquants, en favorisant en priorité les index qui auraient l'impact positif le plus important sur les performances de l'application.

Pour cela, nous utilisons les DMV « dm_db_missing_index_group_stats » et « dm_db_missing_index_details » afin d'évaluer le gain de performances que chaque index manquant permettrait de gagner ainsi que les champs à ajouter à l'index manquant.

Voici la requête correspondante :

```
SELECT TOP 20
    [Total Cost] = ROUND(avg_total_user_cost *
                        avg_user_impact *
                        (user_seeks + user_scans), 0),
    avg_user_impact,
    TableName = statement,
    [EqualityUsage] = equality_columns,
    [InequalityUsage] = inequality_columns,
    [Include Cloumns] = included_columns
FROM sys.dm_db_missing_index_groups g
INNER JOIN sys.dm_db_missing_index_group_stats s ON s.group_handle =
g.index_group_handle
INNER JOIN sys.dm_db_missing_index_details d ON d.index_handle =
g.index_handle
ORDER BY [Total Cost] DESC;
```

Le coût total de chaque index manquant est évalué assez logiquement par le produit des données suivantes :

- avg_total_user_cost : coût moyen actuel des requêtes qui serait impacté par le nouvel index,
- avg_user_impact : gains estimés (en pourcentage) par SQL Server sur les requêtes concernées par l'index manquant, si l'index manquant était implémenté.
- user_seeks : nombre de fois que les requêtes utilisateurs utiliseraient l'index pour interroger l'information contenue dans l'index.
- user_scans : nombre de fois que les requêtes utilisateurs balayeraient l'intégralité de l'index pour trouver un enregistrement complet à l'aide de celui-ci.

La DMV permet également de préciser les champs qui devraient constituer l'index et dans quelles conditions ces champs seraient utilisés au sein des requêtes (conditions WHERE d'égalité ou d'inégalité).

Enfin, elle indique les éventuels champs qu'il pourrait être judicieux d'inclure dans l'index.

Les champs inclus ne font pas partie de l'index même (pas de recherche possible sur ces champs via cet index) mais sont copiés au dernier niveau de l'index (dans les *feuilles* de l'index) afin que les requêtes puissent accéder directement à

cette information sans que le moteur relationnel ait besoin d'interroger la table suite à l'utilisation de l'index.

La dernière étape relative à l'optimisation des index consiste à traquer les index existants inutilisés.

Comme nous l'avons vu précédemment, les index consomment de l'espace de stockage et du temps processeurs lors de la mise à jour des informations contenues dans les index.

Un index inutilisé portant sur une donnée fréquemment modifiée va induire de nombreux mécanismes de mises à jour et ralentir inutilement le serveur.

Pour réaliser cette étape, nous allons faire appel à la DMV « dm_db_index_usage_stats ».

Cette DMV permet de récupérer des statistiques quant à l'usage de chaque index, nous allons donc pouvoir lister les index qui ne sont pas utilisés pour récupérer des données (0 user_seek, 0 user_scans) mais qui ont été mis à jour suite à des modifications de leur table de rattachement (user_updates >0).

Voici la requête correspondante :

```
SELECT TOP 100
    DatabaseName = DB_NAME(),
    TableName = OBJECT_NAME(s.[object_id]),
    IndexName = i.name,
    user_updates,
    system_updates
FROM
    sys.dm_db_index_usage_stats s
INNER JOIN
    sys.indexes i
    ON s.[object_id] = i.[object_id]
    AND s.index_id = i.index_id
WHERE
    s.database_id = DB_ID()
AND
    OBJECTPROPERTY(s.[object_id], 'IsMsShipped') = 0
AND
    user_seeks = 0
AND
    user_scans = 0
AND
    user_lookups = 0
AND
    i.name IS NOT NULL
ORDER BY
    user_updates DESC
```

SQL Server fournit donc une série de statistiques qui peuvent se révéler extrêmement précieuses lors des phases d'optimisations de la base de données.

L'application de ces méthodes d'optimisation a permis des gains de performances drastiques sur les modules les plus complexes de l'application.

CONCLUSION

Le projet de migration de l'application AssystMaint a donc constitué un travail de longue haleine extrêmement complet et découpé en différentes étapes.

En premier lieu, des tâches d'études tenant compte des contraintes inhérentes au fonctionnement et au métier de l'agence Activités Nucléaires ainsi qu'au fonctionnement du groupe GDF Suez et de son système d'information ont été nécessaires.

Dans un deuxième temps, c'est un véritable travail de chef de projet et de maîtrise d'œuvre qui a été requis : rédaction d'un cahier des charges, sélection d'un prestataire, contrôle de son travail et validation des surfacturations.

Une fois la migration terminée, les relations détériorées entre l'agence Activités Nucléaires et son prestataire de service en ingénierie informatique ont conduit à l'abandon du prestataire et à la fin de la maintenance applicative de la solution logicielle par ce dernier.

Cette situation m'a ouvert la possibilité de poursuivre ma participation au sein de l'agence Activités Nucléaires afin de prendre en charge le développement et l'amélioration quotidienne de l'application.

De plus, l'optimisation de la base de données n'ayant pas été réalisée par notre prestataire en temps utile, ce travail passionnant fut l'occasion pour moi de monter en compétence sur SQL Server, me permettant d'être un professionnel aujourd'hui confirmé sur cette technologie.

Pour conclure, ce projet déjà bien complet ouvre encore aujourd'hui de nouvelles opportunités.

SQL Server 2005 arrive en fin de vie et une migration sur SQL Server 2012 permettrait également l'ajout de fonctionnalités pertinentes à une application déjà très complète d'un point de vue métier tout en offrant un challenge technique intéressant.

Enfin, l'agence a été contactée une première fois pour étudier la possibilité d'ouvrir l'utilisation d'AssystMaint à d'autres entités du groupe GDF Suez, consacrant l'intérêt et la robustesse de cette solution logicielle.

Bibliographie

- BROUARD F., BRUCHEZ R., SOUTOU C., 2010. *SQL*. Pearson, Paris, 307p.
- LOW G., ALBERT H., BARKER C., HIONS M., 2012. *Developing MS SQL Server 2012 databases*. Microsoft Corporation, Redmond, 422p.
- BROUARD F., 2003. *Optimisation des SGBDR et du SQL*.
<http://sqlpro.developpez.com/cours/optimiser/>
- BROUARD F., 2003. *Optimisation des bases de données MS SQL Server*.
<http://sqlpro.developpez.com/sqlserver/optimisation/intro/>
- STIRK I, 2008. *Uncover hidden data to optimize application performance*.
<http://msdn.microsoft.com/en-us/magazine/cc135978.aspx>
- WIKIPEDIA, 2014. Vulgarisation sur les thèmes suivants :
 - CFC : <http://fr.wikipedia.org/wiki/Chlorofluorocarbone>
 - GMAO : <http://fr.wikipedia.org/wiki/GMAO>
 - HCFC : <http://fr.wikipedia.org/wiki/Hydrochlorofluorocarbone>
 - HFC : <http://fr.wikipedia.org/wiki/Hydrofluorocarbone>
 - LDAP : <http://fr.wikipedia.org/wiki/LDAP>

Liste des figures & tableaux

Figures :

1. Place de COFELY Services au sein de GDF-Suez	9
2. Architecture matérielle d'origine	21
3. Architecture matérielle post-migration	26
4. Architecture logicielle post-migration	28

Tableaux :

I. Types de données « texte » dans SQL Server	50
---	----

Résumé

L'agence COFELY Services Activités Nucléaires est spécialisée dans le froid industriel. Sa principale activité est d'assurer la maintenance des systèmes de refroidissement des centrales nucléaires françaises.

L'agence utilise un logiciel de maintenance assistée par ordinateur nommé AssystMaint. Documentation remise au client, documentation réglementaire obligatoire, gestion des matériels maintenus et du personnel : AssystMaint permet de piloter l'intégralité de l'activité de l'agence.

Considérant l'importance majeure d'AssystMaint, la DSI du groupe GDF Suez a donc décidé de fiabiliser cette solution logicielle en confiant la migration de la base de données de l'application d'Access 2003 vers MS SQL Server 2005 à un prestataire de service.

C'est d'abord en tant que chef de projet, puis en tant que développeur et administrateur de base de données que Julien JAMMES est intervenu sur ce projet.

Dans sa première partie, ce projet fut l'occasion d'écrire un cahier des charges spécifique, de sélectionner un prestataire puis d'encadrer et contrôler son travail.

Une fois le prestataire abandonné, ce projet permit d'approfondir le domaine des bases de données, en particulier de SQL Server et de mettre en place des processus d'optimisation avancée.

Mots-clefs : Base de données, Migration, Access, SQL Server, Administration, Optimisation.

Summary

Cofely Services Nuclear Activities Agency specializes in industrial cooling. Its main activity is to ensure the maintenance of cooling systems of French nuclear power plants.

The agency uses a computerized maintenance management system software called AssystMaint. Client documentation, mandatory regulatory documentation, maintained equipment management and staff management: AssystMaint is used to manage the entire agency's activity.

Considering the major importance of AssystMaint, the IT Department of the GDF Suez group decided to entrust a service provider with the migration of the software's database from Access 2003 to MS SQL Server 2005.

It is primarily as a project manager, then as a developer and database administrator that Julien JAMMES intervenes on this project.

The first part of this project offered the opportunity to write specifications, to select a service provider and to supervise and monitor his work.

Once the service provider discarded, this project allowed deepening the field of databases, especially SQL Server, and implementing advanced optimization process.

Key-words: Database, Migration, Access, SQL Server, Administration, Optimization.