

**CONSERVATOIRE NATIONAL DES ARTS ET METIERS**

**PARIS**

---

**MEMOIRE**

**présenté en vue d'obtenir**

**le DIPLOME D'INGENIEUR CNAM**

**SPECIALITE : Informatique**

**OPTION : ISI (Ingénierie en Systèmes d'Information)**

**par**

**Rabih Saied**

---

**Intégration technique de l'ERP Dynamics AX dans le secteur public**

**Soutenu le : jeudi 11 juin 2015**

---

**JURY**

**PRESIDENT** : Elisabeth Métais, Professeur CNAM.

**MEMBRES** : Elisabeth Métais (CNAM), Nicolas Travers (CNAM), Marin Ferecatu (CNAM), Christophe Bodin (Gfi Informatique) et Zakaria Aazair (Gfi Informatique).

## Remerciements

Je tiens tout d'abord à remercier sincèrement Monsieur Christophe Bodin, qui, en tant que directeur du Pôle Intégration Technique (PIT), m'a permis de réaliser cette mission d'ingénieur.

Je remercie évidemment mon tuteur et collègue Monsieur Zakaria Aazair, pour sa disponibilité et tous ses précieux conseils tout au long de la réalisation de ces projets.

Je n'oublie pas aussi de remercier mon tuteur et maître de conférences au CNAM Monsieur Marin Ferecatu, pour son encadrement ainsi que pour sa disponibilité tout au long de la durée de ce travail.

Mes remerciements s'adressent également à toute l'équipe Gfi-SSP : les directeurs et chefs de projet, les consultants (en particulier Anne-Charlotte, Géraud, René et Suraj) et les développeurs (en particulier Sacha) pour leur accueil, l'aide et le temps qu'ils ont bien voulu me consacrer.

Enfin, je remercie mes parents, Simo, Najla et Ahmed. Mais, ce mémoire n'aurait surtout pas pu exister sans la patience et le soutien infaillible de mon épouse et de ma fille durant toutes ses nombreuses années passées au CNAM.

## Liste d'abréviations

**AD** : Active Directory.

**AOS** : Serveur d'objets d'application.

**BI** : Business Intelligence.

**CICEO** : Centre d'Intégration des Compétences et Conseil En Organisation.

**ESN** : Entreprise de Services du Numérique (nouveau nom donné au SSII).

**ERP** : Entreprise Ressource Planning ou PGI (Progiciel de Gestion Intégré).

**HA** : High Availability (Haute Disponibilité).

**IOPS** : Input/Output Operations Per Second (Opérations d'entrées-sorties par seconde).

**OLAP** : On-Line Analytical Processing (Traitement analytique en ligne).

**OLTP** : OnLine Transaction Processing (Traitement transactionnel en ligne).

**NLB** : Network Load Balancing (Equilibrage de Charge Réseau).

**RAID** : Redundant Array of Independent Disks, (Regroupement redondant de disques indépendants).

**RDS** : Remote Desktop Service.

**RPC** : Remote Procedure Call.

**PIT** : Pôle d'Intégration Technique.

**SAN** : Storage Area Network (Réseau de stockage).

**SSAS** : SQL Server Analysis Services.

**SSRS** : SQL Server Reporting Services.

**SSP** : Solution Secteur Public.

**WCF** : Windows Communication Foundation.

## Table des matières

<b>Remerciements</b> .....	2
<b>Liste d'abréviations</b> .....	3
<b>Table des matières</b> .....	4
<b>Synthèse</b> .....	6
<b>Abstract</b> .....	7
<b>Introduction</b> .....	8
<b>1<sup>ère</sup> Partie : Le contexte de la mission</b> .....	11
1. L'entreprise Gfi.....	12
1.1. Présentation et historique de la société Gfi.....	12
1.2. La branche Secteur Public .....	13
2. Gfi et l'intégration ERP dans le secteur public .....	15
2.1. Le vertical Gfi-SSP .....	15
2.2. L'équipe Software SSP .....	16
2.3. Les clients Gfi-SSP .....	19
3. L'ERP Microsoft Dynamics AX.....	21
3.1. Concept d'ERP ou PGI.....	21
3.2. Microsoft Dynamics AX.....	22
<b>2<sup>ème</sup> Partie : Un état de l'art sur l'intégration de Dynamics AX</b> .....	25
4. Un métier transverse.....	26
4.1. Les rôles et compétences du métier d'intégrateur ERP .....	26
4.2. Les charges de l'intégrateur .....	27
4.3. La méthodologie Sure Step .....	28
5. L'architecture de Dynamics AX .....	31
5.1. L'architecture logique de la solution .....	31
5.2. Les composants de la solution .....	32
6. Les problématiques posées par l'intégration de Dynamics AX .....	36
6.1. La virtualisation .....	36
6.2. Les besoins réseaux.....	37
6.3. Le mode de déploiement.....	38
6.4. Le stockage .....	39

6.5.	La sécurité .....	41
6.6.	La sauvegarde .....	43
6.7.	La gestion des impressions .....	44
6.8.	La haute disponibilité.....	45
6.9.	La supervision .....	46
7.	L'analyse des cas Gfi-SSP .....	48
<b>3<sup>ème</sup></b>	<b>Partie : Les études de cas Gfi-SSP</b> .....	<b>50</b>
8.	L'industrialisation des mises à jour des environnements.....	51
8.1.	Le contexte projet .....	51
8.2.	La mise en place des outils de travail .....	52
8.3.	La revue de l'organisation d'une livraison .....	56
8.4.	L'étude de l'organisation des scripts .....	58
8.5.	La nouvelle structure adoptée .....	61
8.6.	Le nouveau découpage en processus des scripts principaux .....	66
8.7.	La mise en œuvre chez les clients.....	70
9.	La gestion de la haute disponibilité dans Dynamics AX .....	75
9.1.	Le contexte projet .....	75
9.2.	La haute disponibilité par composants.....	76
9.3.	Une typologie de haute disponibilité .....	82
9.4.	Le déploiement chez un client en production .....	88
10.	La mise en place de la supervision de Dynamics AX .....	91
10.1.	Le contexte projet de la supervision.....	91
10.2.	L'étude de la supervision de Dynamics AX.....	92
10.3.	Superviser Dynamics AX dans Centreon.....	94
10.4.	La supervision de Dynamics AX en production .....	105
	<b>Conclusion</b> .....	<b>108</b>
	<b>Annexes</b> .....	<b>110</b>
	<b>Bibliographie</b> .....	<b>114</b>
	<b>Table des figures</b> .....	<b>115</b>
	<b>Liste des tableaux</b> .....	<b>116</b>
	<b>Résumé</b> .....	<b>117</b>

## Synthèse

Au sein de la branche SSP (Solutions Secteur Public) de Gfi Informatique, j'ai effectué ma mission d'ingénieur en tant qu'architecte sur le produit Microsoft Dynamics AX. C'est dans ce contexte que j'ai alors été confronté aux problématiques d'intégration de la solution financière Gfi-SSP basée sur cet ERP. Mais, ce travail d'ingénierie a surtout consisté à rechercher comment augmenter la qualité dans notre intégration. En effet, avant mon arrivée, certaines activités importantes de ce processus manquaient de maîtrise et certaines opérations n'étaient pas optimisées. Je me suis donc impliqué à apporter premièrement une amélioration dans la connaissance des problématiques d'intégration. En second lieu, mes travaux ont porté sur la nécessité de perfectionner l'industrialisation des mises à jour des environnements clients. Pour cela, j'ai donc réorganisé le cycle de livraison, mis en place un environnement de « build » chez les clients, installé de nouveaux outils de travail, restructuré l'arborescence des scripts en pensant en termes de processus et introduit la programmation orientée objet pour gérer la complexité des données environnements. Ensuite, j'ai apporté une connaissance approfondie sur la question du déploiement de la haute disponibilité (HA) dans Dynamics AX qui garantit maintenant la qualité de notre offre technique. Effectivement, j'ai créé une typologie de HA en fonction des composants critiques, de la difficulté de mise en œuvre et du coût. A partir de ce travail, j'ai pu alors déployer la HA avec succès sur une plateforme client en cours de démarrage en production. Enfin, suite à mon implication dans la supervision de Gfi-SSP, un pas majeur a été franchi permettant de contrôler les principaux composants de l'ERP. Après avoir rédigé un livrable sur la question, j'ai intégré à la demande d'un client des compteurs pertinents dans l'outil Centreon afin qu'il puisse superviser ses plateformes Dynamics AX. Par ailleurs, dans un second temps, ce travail m'a aidé à superviser pleinement le site d'un autre client en production. Par conséquent, l'ensemble de ces projets réalisés avec succès ont abouti à augmenter la valeur ajoutée de l'activité PIT (Pôle Intégration Technique) Dynamics AX au sein de Gfi.

**Mots-clés : ERP, Microsoft Dynamics AX, Gfi-SSP, intégration, industrialisation, supervision, haute disponibilité.**

## Abstract

This document describes the work I performed in the SSP (Public Sector Solutions) branch of Gfi Informatique company in view of obtaining my engineering title from Conservatoire National des Arts et Métiers (CNAM). My job there as a software architect was to assist and manage the deployment of the financial solution proposed by Gfi-SSP based on Microsoft Dynamics AX, period during which I had to solve several problems concerning the integration process. My contribution was fourfold.

First, I had to collect and analyze data, with a look on how to improve the quality of the integration. The main goal at this stage was to detect which activities lacked control and which were not optimized.

Second, I participated in perfecting the industrialization of updates on customer environments based on aforementioned analysis. To do this, I reorganized the delivery cycle, I implemented a build environment and installed new work tools. I restructured the scripts tree thinking in process terms and introduced object-oriented programming to manage the environments data complexity.

Third, I made a thorough study on the issue of deploying high availability (HA) in Dynamics AX which now guarantees the quality of our technical offer. Indeed, I have created a HA typology based on critical components, implementation difficulties and cost. From this work, I could then deploy HA successfully on a production platform client.

Finally, following my involvement in overseeing Gfi-SSP, a major step has been taken to control the main components of the ERP. After writing a document on this issue, I implemented relevant counters in the Centreon tool for a customer to monitor its Dynamics AX platforms. In a second time, this work helped me to supervise a production customer site fully.

As a consequence of my work, our technical offer can now be mastered and deployed more easily. Also, my work increased the overall added value of the PIT Dynamics AX business in my enterprise.

**Keywords : ERP, Microsoft Dynamics AX, Gfi-SSP, implementation, industrialization, monitoring, high availability.**

## Introduction

Depuis le début des années 2000, il existe un contexte favorable à l'intégration d'ERP (Entreprise Ressource Planning) dans le secteur public en France et cela pour deux raisons majeures.

Les organismes publics commencent premièrement à se tourner vers les solutions PGI (Progiciel de Gestion Intégré) pour des logiques économiques. A cause de la crise, ils sont d'abord confrontés à des contraintes financières de plus en plus croissantes. De plus, certaines réformes votées (réforme organisationnelle territoriale, réforme hospitalière,...) les poussent à des fusions entre entités.

Deuxièmement, la réforme GBCP (Gestion Budgétaire et Comptable Publique) initiée par l'Etat pousse tout le secteur public à s'orienter vers les ERP. En effet, ce décret de 2012 impose aux collectivités territoriales ainsi qu'aux opérateurs dont le financement est majoritairement public, la mise en place d'une nouvelle comptabilité budgétaire avant le 1<sup>er</sup> janvier 2016. Ainsi, ce sont près de 700 organismes publics qui sont concernés et la moitié d'entre eux dépendent d'un seul éditeur : Gfi Informatique. Fort de son expertise métier et d'un partenariat industriel avec Microsoft, Gfi a donc décidé de proposer l'ERP Microsoft Dynamics AX sur cette assurance GBCP avec son offre Gfi-SSP (Solutions Secteur Public).

C'est dans ce contexte, au sein de la branche SSP (Software Secteur Public) de cette entreprise que s'est déroulé durant six mois (septembre 2014 à mars 2015) mon stage de fin d'étude d'ingénieur. En poste depuis cinq ans dans cette société, j'ai eu ainsi l'opportunité de pouvoir intégrer à cette occasion le pôle transverse du PIT (Pôle Intégration Technique) en tant qu'architecte Dynamics AX. Avec cette nouvelle fonction, j'ai donc participé de manière active aux différents projets au sein des équipes Gfi-SSP constitués de chefs de projet, de consultants et de développeurs. Durant cette mission d'ingénierie, j'ai été alors confronté aux problématiques globales de l'intégration technique de Dynamics AX au sein des organismes publics.

Mais, l'objectif principal de cette longue mission a surtout été d'apporter des solutions d'amélioration au problème de la qualité du processus d'intégration technique de Dynamics

AX chez les clients Gfi SSP. Améliorer un processus consiste à permettre d'obtenir des résultats positivement évolutifs au regard par exemple d'exigences nouvelles, de dysfonctionnements repérés et analysés ou encore de suggestions internes. C'est un travail de fond qui concrétise tout le dispositif qualité mis en œuvre et qui conduit à la qualité proprement dite. Par conséquent, à partir des missions confiées, je me suis demandé de quelle manière optimiser l'industrialisation des mises à jour des environnements ? Comment est-il possible de mettre en place des systèmes de haute disponibilité (HA) pour la solution Gfi-SSP ? Enfin, comment peut-on superviser les plateformes Dynamics AX ? Voici notamment toutes les principales interrogations qui sont apparues et auxquelles je me suis impliqué à travailler dans ce mémoire d'ingénieur.

Pour y répondre, dans une première partie, je commencerai par présenter le contexte de la mission qui m'a été confiée. En effet, je ferai d'abord une présentation succincte de l'entreprise Gfi, puis, je soulignerai l'expérience de la société dans l'intégration d'ERP dans le secteur public et enfin, je terminerai par présenter l'ERP Microsoft Dynamics AX. Dans une deuxième partie, je dresserai un état de l'art sur l'intégration technique d'un ERP. Effectivement, après avoir montré les rôles et charges du métier d'intégrateur d'ERP, je m'arrêterai sur la complexité de l'architecture de Dynamics AX pour montrer au final les problématiques posées par le déploiement de cet ERP. Enfin, dans une dernière partie, à travers trois cas pratiques confiés, il s'agira d'augmenter la qualité du processus d'intégration technique de la solution Gfi-SSP. C'est donc, dans ce contexte, que j'aborderai, en premier lieu, les travaux suivants dans lesquels je me suis impliqué pour améliorer l'industrialisation des mises à jour des environnements clients :

- Mise en place d'un environnement « build » chez les clients ;
- Installation d'outils de travail (éditeurs de code source, client et hébergeur SVN,...) ;
- Réorganisation du cycle de livraison des mises à jour ;
- Réorganisation du travail dans l'équipe Infrastructure ;
- Réorganisation de la structure arborescente des scripts en pensant en termes de processus (copie d'environnement, gestion de la base modelstore,...) ;
- Conception d'un fichier de configuration XML par client avec leurs données environnements ;
- Conception de scripts PowerShell avec la programmation orientée objet ;

- Déploiement en contexte client (hors-production), tests et débogage des scripts ;
- Déploiement sur tous les environnements clients d'une version 1.0.

En second lieu, je présenterai les travaux effectués sur la gestion de la haute disponibilité (HA) dans Dynamics AX :

- Etude et rédaction d'un livrable reprenant les solutions de HA par composants ;
- Conception d'une typologie de haute disponibilité en fonction des critères suivants : composants critiques, coût et complexité de mise en œuvre ;
- Déploiement d'une solution HA chez un client en cours de démarrage en production.

Enfin, je détaillerai les tâches réalisées dans la supervision de notre produit Gfi-SSP :

- Etude et analyse des moyens de supervision, des seuils d'alerte et des outils existants ;
- Mise en place d'un environnement de test pour la supervision avec l'outil Centreon ;
- Tests de la supervision des composants dans cet environnement interne ;
- Rédaction d'un livrable à destination du client ;
- Réalisation de la supervision d'un environnement client en production ;

Pour réussir ces projets, j'ai suivi un découpage en quatre phases. Une première a été la compréhension de la documentation autour du sujet et des technologies à utiliser. A travers plusieurs sources documentaires (sites et ouvrages spécialisés), j'ai alors cherché à accumuler et analyser une masse d'informations sur des sujets comme la programmation orientée objet avec PowerShell, la haute disponibilité ou la supervision. La seconde a été une phase de spécification qui a été de mettre en adéquation mes recherches avec les besoins des clients et d'investiguer les problèmes. L'objectif a été de proposer des solutions alternatives (typologie de haute disponibilité possible, typologie de solutions de supervisions,...) avec les points forts et les points faibles. Une troisième a été une phase de conception avec les outils choisis (scripts de copie d'environnements ou de gestion de la base modelstore Dynamics AX, implémentation de compteurs de supervision dans l'outil Centreon, déploiement de la haute disponibilité dans une plateforme en production, rédaction de livrables pour le client,...). Enfin, la dernière phase a été le déploiement chez le client et l'intégration à une équipe projet sur le terrain.

## **1<sup>ère</sup> Partie : Le contexte de la mission**

L'objectif de cette première partie est d'expliquer le contexte dans lequel j'ai réalisé ma mission de fin d'études pour le diplôme d'ingénieur CNAM. En effet, il est tout d'abord important de comprendre les raisons du choix du sujet sur l'ERP Microsoft Dynamics AX dans le secteur public et les conditions dans lesquelles j'ai réalisé ce travail.

Pour cela, je vais ainsi commencer par présenter l'entreprise Gfi Informatique dans laquelle je travaille depuis maintenant six ans. Après cette présentation générale et un historique du groupe, je ciblerai ensuite un des secteurs importants pour la société : le secteur public.

Dans un second temps, je m'intéresserai sur les raisons qui ont poussées Gfi à vouloir investir dans l'intégration d'ERP dans le secteur public. Je présenterai ainsi le contexte national, l'offre proposée par la société (le vertical Gfi-SSP) et, pour terminer, l'équipe Software SSP (Solutions Secteur Public) chargée des projets de ce produit.

Enfin, dans un dernier point, je m'arrêterai sur le choix de l'ERP Microsoft Dynamics AX par l'entreprise pour bâtir sa solution. Après avoir défini le concept d'ERP, je ferai par conséquent une introduction sur la gamme Dynamics et je finirai par montrer le positionnement de Dynamics AX sur le marché.

# 1. L'entreprise Gfi

Ma mission s'est déroulée dans l'entreprise Gfi Informatique. En premier lieu, je vais donc commencer par une présentation générale et un bref historique de la société.

## 1.1. Présentation et historique de la société Gfi

Né il y a maintenant plus de quarante ans, le groupe Gfi s'est constitué au fil du temps en acquérant notamment de multiples entités.

### *1.1.1. Présentation générale du groupe*

Gfi est une ESN (Entreprise de Services du Numérique) française qui compte près de 10 000 collaborateurs dans le monde. Elle est implantée dans toute la France avec plus de 40 agences. A l'international, elle compte des filiales en Europe (Belgique, Luxembourg, Suisse, Espagne, Portugal) et au Maroc.

Gfi se définit « comme un acteur européen de référence des services informatiques à valeur ajoutée et des logiciels occupant un positionnement stratégique entre les opérateurs de taille mondiale et les acteurs de niche ». Elle a choisi de développer sa stratégie autour de six secteurs dont voici des exemples de clients :

- Banque Assurance : BNP Paribas, Société Générale, Caja de Madrid, ... ;
- Secteur Public : ministères, établissements publics, régions, départements, ... ;
- Telecom et Média : Orange, Telefonica, Radio-France, ... ;
- Transport-Services : SNCF, RATP, Air France, ... ;
- Industry-Retail : La Poste, Eurocontrol, ... ;
- Utilities-Energy-Chemicals : EDF, GDF-Suez, Total, Areva, ... .

Après cette rapide présentation générale, je vais maintenant donner quelques dates clés qui ont construit le groupe.

### *1.1.2. Historique du groupe*

Le Groupement Français d'Informatique (GFI) est créé en 1970. En 1991, la filiale française de la SSII (Société de Services en Ingénierie Informatique) américaine EDS (Electronic Data Systems) prend le contrôle de GFI. En 1995, Jacques Tordjman crée Gfi Informatique suite au rachat des activités d'ingénierie à EDS. A ce moment, 700 personnes travaillent pour cette nouvelle structure qui réalise un CA (Chiffre d'Affaire) de 45 millions d'euros. Un an plus tard sont créées de nouvelles filiales au Benelux et en Suisse. De 1995 à 1999, Gfi connaît alors une forte période de croissance qui lui permet de réaliser des acquisitions dans les domaines des progiciels et des nouvelles technologies (Netstar Technologies, AT&T, Sinorg,...).

En 2009, Vincent Rouaix prend la succession de Jacques Tordjman en tant que Président-Directeur Général. La Direction met alors en place un nouveau plan stratégique. En 2010, le CA s'élève alors à 658 millions d'euros et les effectifs à 8900 collaborateurs dont 5800 en France. Depuis 2011, Gfi Informatique poursuit cette stratégie de montée en valeur avec notamment de nombreuses acquisitions (Ares, Adix, pôle « Business Solutions » de Thales Services, Cognitis,...). Selon plusieurs sites internet, Gfi se classe désormais dans le Top 10 des ESN en termes de CA en France. Je vais à présent évoquer un des secteurs majeurs pour la société : le secteur public. Effectivement, Gfi Informatique est devenu au fil du temps un des partenaires incontournables des organismes publics.

## **1.2. La branche Secteur Public**

Dans ce secteur, l'entreprise est à 50% intégrateur de systèmes et 50% éditeur de logiciels. Gfi branche Software emploie 820 collaborateurs. Treize domaines métiers sont couverts dont la fiscalité, les ressources humaines, les transports, la santé ou encore les aides sociales. En 2012, le chiffre d'affaire représente alors les 80 millions d'euros.

Parmi les clients de cette branche, on dénombre 850 entreprises du secteur privé dont 520 du domaine de l'industrie, 210 du secteur des services, 60 clients banques finances assurances et 60 clients de l'énergie. Dans le secteur public, on compte plus de 2800

organismes clients. Parmi ces derniers, il y a 500 grandes collectivités, 1500 moyennes et petites collectivités territoriales, 600 établissements publics et 200 établissements hospitaliers. Ces chiffres montrent ainsi l'importante implantation de Gfi dans le secteur public.

Dans ce dernier, parmi les produits phares édités par la branche Software, on peut citer entre autres :

- Gfi-Astre GF pour les finances ;
- Gfi-Astre RH pour les ressources humaines ;
- Gestor pour la gestion des temps ;
- Gfi-Iodas pour l'action sociale.

A ces exemples de progiciels, il faut ajouter deux nouvelles solutions intégrées :

- Gfi-Public CRM (Customer Relationship Management) développé au cœur de Microsoft Dynamics CRM, pour les relations avec les citoyens et les usagers ;
- Gfi-SSP (Solution Secteur Public) pour la gestion financière, développé au centre de l'ERP Microsoft Dynamics AX.

Après avoir présenté l'entreprise Gfi et son histoire, puis dans un second temps, avoir souligné la forte pénétration de la société dans le secteur public, je vais maintenant détailler plus précisément l'offre Solutions Secteur Public (SSP).

## **2. Gfi et l'intégration ERP dans le secteur public**

Gfi est depuis longtemps considéré comme un grand intégrateur ERP sur le marché français (Tomas, 2007). En effet, l'entreprise a d'abord commencé à intégrer des solutions SAP, Oracle et Sage. En 2012, c'est autour des produits ERP Microsoft et depuis 2014, suite au rachat des activités d'iORGA Group, Gfi devient leader en France de l'intégration des solutions Oracle-JDE. Aujourd'hui, tous les secteurs sont concernés par cette intégration d'ERP et notamment le secteur public.

### **2.1. Le vertical Gfi-SSP**

Aujourd'hui, Gfi réalise 20% de son chiffre d'affaires dans les solutions et services pour le secteur public. Le Pôle Finances de la branche Software compte 150 collaborateurs spécialisés. 650 établissements publics et plus de 1000 collectivités territoriales utilisent les offres de ce pôle.

Fort de cet avantage, la société est ainsi devenue Microsoft Gold Certified Partner. En juillet 2009, Gfi a effectivement fait le choix d'utiliser l'ERP Microsoft Dynamics AX comme socle de renouvellement de certaines gammes de ses produits édités. En intégrant à Dynamics AX son vertical Secteur Public, Gfi offre alors sa longue expérience dans l'édition de solutions de gestion financière pour le secteur public.

Développée au cœur de l'ERP Microsoft Dynamics AX, la solution Gfi-SSP présente toute la convivialité, la performance, la flexibilité et l'évolutivité dont les acteurs financiers du secteur public peuvent avoir besoin.



**Figure 1 - Offres proposées par la solution Gfi-SSP**

D'après la proposition commerciale (Figure 1), les bénéfices de la mise en place de cette solution sont notamment :

- Ergonomie intuitive et familière (Microsoft Office) ;
- Workflow de validation intégré ;
- Sécurité et confidentialité éprouvées ;
- Reporting orienté processus métier ;
- Outils décisionnels (Business Intelligence) embarqués.

Grâce à cette solution Gfi SSP, le prix « 2014 ISV Microsoft Dynamics » pour la France a été décerné à Gfi Branche Software pour ses capacités à mettre en œuvre de belles réussites commerciales. Ces dernières ont été réalisées par l'équipe Software SSP que je vais évoquer dans le paragraphe suivant ainsi que mon intégration dans celle-ci.

## **2.2. L'équipe Software SSP**

L'équipe Software SSP est en charge des projets d'intégration de l'ERP Microsoft Dynamics AX dans le secteur public. Après avoir établi un organigramme de la structure, je dresserai un tableau de la situation des projets avec les différents clients Gfi-SSP.

D'après le Tableau I, la structure Software SSP qui travaille sur les projets du vertical Gfi-SSP se compose de trois principales équipes (pilotage, fonctionnelle, technique).

**Tableau I - Composition et missions des équipes en charge des projets vertical Gfi-SSP**

EQUIPE		NOMBRE	STRUCTURE	MSSIONS
<b>Pilotage</b>	<b>Direction / Pilotage de Projet</b>	4	CICEO	Suivi de projet (gestion planning, ordonnancement des tâches, affectation des ressources,...) ; Animer et coordonner les équipes projet ; Mesure des écarts ; Gestion des risques ; Assurer le respect des engagements contractuels
<b>Technique</b>	<b>Développement produit (R&amp;D)</b>	30	Activité Dynamics AX (Pôle Finances)	Développement du vertical Gfi-SSP conformément à la roadmap Gfi.
	<b>Développement projet (Intégration)</b>	4	Activité Dynamics AX (Pôle Finances)	Etude des besoins spécifiques ; Conception et réalisation du développement.
	<b>Intégration Technique</b>	2	PIT (Pôle Intégration Technique)	Définition de l'architecture technique ; Installation des environnements ; Déploiement ; Bascule en production ; Cadrage technique.
<b>Fonctionnelle</b>	<b>Service client (consultants fonctionnels)</b>	15	Activité Dynamics AX (Pôle Finances)	Expertise métiers ; Spécifications des besoins fonctionnels (cadrage fonctionnel) ; Formation.

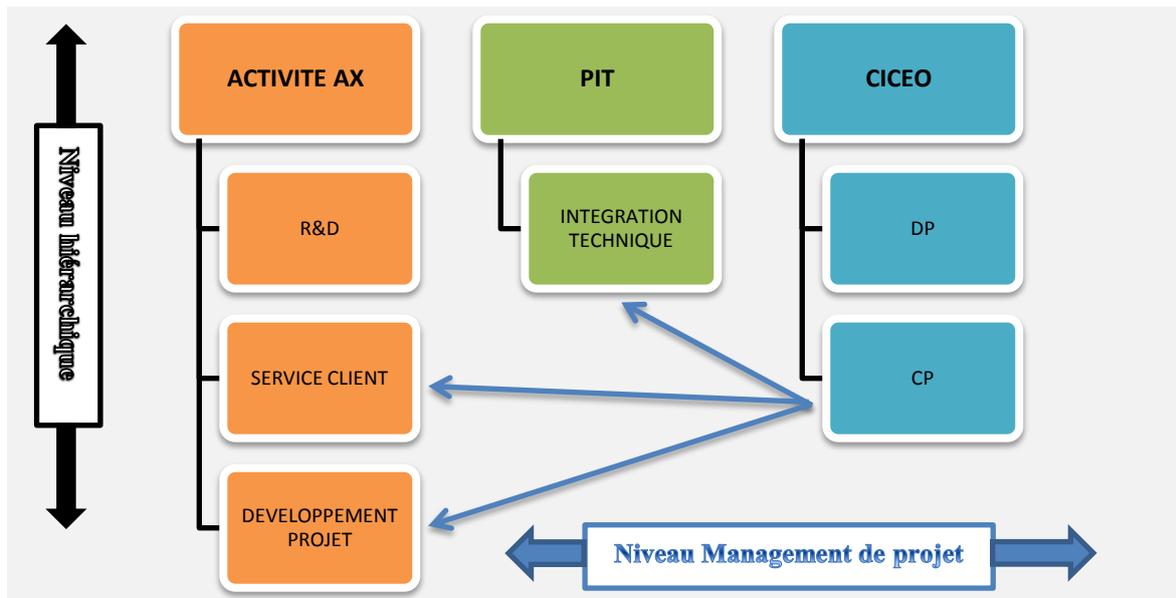
L'équipe de pilotage regroupe les directeurs de projets (DP) et les chefs de projets (CP). Ils appartiennent au Pôle transverse CICEO (Centre d'Intégration des Compétences et Conseil En Organisation). Le CICEO regroupe les activités transverses de direction de projet et de consulting progiciels en opérant sur un groupe de produits et de marchés dans plusieurs domaines regroupés au sein de la Branche Software. CICEO est responsable de la mise en œuvre du système de management de la qualité GMS (Global Management System) conforme à la norme ISO 9001 version 2008.

L'équipe fonctionnelle se compose elle exclusivement de consultants du Pôle Finances chargés de la spécification du besoin et du cadrage fonctionnel ainsi que des formations sur l'ERP Dynamics AX.

Enfin, l'équipe technique se divise en trois structures. La première est en charge du développement du produit Gfi-SSP. La seconde regroupe les développeurs responsables des personnalisations et développements spécifiques de l'application. La dernière regroupe les

intégrateurs techniques de l'ERP qui dépendent du pôle transverse PIT (Pôle Intégration Technique). Elle est chargée de l'intégration de l'ERP Dynamics AX chez les clients. Sous la responsabilité du directeur du PIT, cette équipe est actuellement constituée de deux architectes Dynamics AX.

Le management des équipes s'effectue sur deux niveaux (Figure 2) :



**Figure 2 - Organigramme du management de l'équipe Software SSP Dynamics AX**

Tout d'abord, il existe un premier niveau hiérarchique où :

- L'intégration technique dépend directement du responsable du PIT ;
- Les CP et DP dépendent du CICEO ;
- Enfin, la Recherche & Développement (R&D), le Service client et le Développement projet dépendent directement du responsable de l'Activité AX.

Dans un deuxième temps, il y a un niveau transverse de management où le CP manage les collaborateurs du service client, du développement projet et de l'intégration technique.

Ainsi, tous ses intervenants collaborent ensemble aux différents projets sur le vertical Gfi-SSP. Je vais donc à présent détailler ces divers projets en cours.

### **2.3. Les clients Gfi-SSP**

Actuellement, il existe huit organismes publics clients du vertical Gfi-SSP :

- Le CNFPT (Centre National de la Fonction Publique Territoriale) ;
- L'hôpital FOCH ;
- Le GPMD (Grand Port Maritime de Dunkerque) ;
- Les Ports de Paris ;
- L'ADEME (Agence de l'Environnement et de la Maîtrise de l'Energie) ;
- La Mairie de Monaco ;
- L'ANRU (Agence Nationale pour la Rénovation Urbaine) ;
- L'ENPC (Ecole Nationale des Ponts et Chaussées).

A partir de cette liste, j'ai établi un tableau qui récapitule les informations concernant l'évolution de ces projets en montrant notamment les objectifs du projet ainsi que leur degré de complexité (Tableau II) :

**Tableau II - Projets Gfi-SSP par clients classés date de démarrage prévue en production**

	UTILISATEURS	SITES	LIEU	OBJECTIF	NIVEAU	DEMARRAGE
<b>CNFPT</b>	1000	43	France métropolitaine et DOM-TOM	Mise en œuvre d'un système d'information financier.	Complexe	Janvier 2015
<b>GPMD</b>	< 100	1	Dunkerque (59)	Mise en œuvre d'un système d'information financier.	Moyen	Janvier 2016
<b>Mairie de Monaco</b>	< 100	1	Principauté de Monaco (98)	Mise en œuvre d'un système d'information financier.	Moyen	Janvier 2016
<b>ADEME</b>	1000	20	France métropolitaine et DOM-TOM	Mise en œuvre d'une solution pour la gestion des achats et de la gestion des comptabilités budgétaire et analytique.	Complexe	Janvier 2016
<b>Ports de Paris</b>	< 100	1	Paris	Mise en œuvre d'un système d'information financier.	Moyen	Janvier 2016
<b>ANRU</b>	< 50	1	Paris	Mise en œuvre d'un système d'information financier.	Simple	Janvier 2016
<b>ENPC</b>	< 100	1	Champs-sur-Marne (77)	Mise en œuvre d'un système d'information financier.	Simple	Janvier 2016
<b>FOCH</b>	< 100	1	Suresnes (92)	Mise en œuvre d'un système d'information financier.	Moyen	Avril 2016

Après avoir expliqué le vertical SSP, je vais à présent m'attacher à détailler le concept d'ERP et donner les raisons du choix de Dynamics AX pour la solution Gfi-SSP.

### **3. L'ERP Microsoft Dynamics AX**

Dans ce troisième point, il s'agit par conséquent d'aborder le concept d'ERP en apportant une définition. Puis, dans un second temps, présenter la gamme ERP de Microsoft Dynamics et son positionnement sur le marché mondial et national.

#### **3.1. Concept d'ERP ou PGI**

Conçu par un éditeur unique, l'ERP est une solution logicielle temps réelle adaptable composée de plusieurs modules interconnectés couvrant la plupart des fonctions-clés de l'entreprise. L'ERP est fondé sur l'utilisation d'une base de données relationnelle (un référentiel) alimentée et gérée par des modules applicatifs paramétrables. Selon Jean-Louis Lequeux (2008), pour désigner une solution comme étant un ERP, elle doit répondre à au moins trois critères précis :

- Comprendre une gestion de plusieurs domaines de l'entreprise par des modules intégrés ;
- Avoir un référentiel unique de données ;
- S'adapter rapidement aux règles de fonctionnement de l'entreprise (professionnelles, légales,...).

D'origine américaine, la notion d'ERP est apparue au début des années quatre-vingt-dix. Elle est de fait une généralisation de MPR (Manufacturing Resource Planning) prenant en charge toute la gestion de l'entreprise (ressources humaines, gestion comptable et financière, administration, production, logistique, ventes et achats,...). Le nom de ERP s'est imposé ensuite par rapport à d'autres, comme en France avec PGI (Progiciel de Gestion Intégré) moins « vendeur » selon Deixonne (2006).

Restés confinés jusqu'au début des années deux-mille aux grandes entreprises des secteurs industriels, commerciaux et administratifs, les éditeurs d'ERP s'intéressent maintenant aux PME (Petites et Moyennes Entreprises) / PMI (Petites et Moyennes

Industries) ainsi qu'aux organismes de tout secteur. Pour l'entreprise Microsoft, c'est devenu d'ailleurs le cœur de cible de sa gamme ERP Dynamics (Lequeux, 2008). En parallèle, les sociétés de service en informatique et en intégration de systèmes ont par conséquent développé des compétences dans l'intégration des progiciels du marché.

Après avoir abordé le concept d'ERP, je vais évoquer l'exemple de l'ERP Microsoft Dynamics AX choisie pour la solution Gfi-SSP.

### **3.2. Microsoft Dynamics AX**

Avant de voir le positionnement de Microsoft Dynamics AX sur le marché, je vais d'abord effectuer une présentation de la gamme Dynamics et un historique du produit AX.

#### *3.2.1. Présentation de la gamme Microsoft Dynamics*

La gamme Microsoft Dynamics intègre des solutions de gestion financière, de gestion de la chaîne logistique et de gestion de la relation client. Elle est destinée à cinq grands secteurs d'activité : l'industrie, la grande distribution, la finance, le commerce de détail et le secteur public. Microsoft Dynamics inclut notamment :

- Microsoft Dynamics AX (anciennement Axapta) qui englobe tout le business ;
- Microsoft Dynamics NAV (anciennement Navision) orienté pour des spécificités industrielles ;
- Microsoft Dynamics CRM : gestion de la relation client.

Microsoft Dynamics AX est la solution financière de planification des ressources d'entreprise (ERP) multilingue prenant en charge plusieurs devises. Elle possède un grand nombre de fonctionnalités de gestion d'entreprise à l'échelle mondiale pour ce qui est des finances, des ressources humaines et des opérations.

À l'origine, Microsoft Dynamics AX a été conçu par la société danoise Damgaard sous le nom d'Axapta en mars 1998. Lors de son lancement, le logiciel Axapta est

uniquement destiné aux marchés Danois et Américains. En 2000, Damgaard et la société Navision Software fusionnent pour donner naissance à Navision Damgaard.

Au cours de l'été 2002, Microsoft acquiert Navision. La société est tout d'abord renommée Microsoft Business Solutions Axapta puis Microsoft Dynamics AX à partir de la version 3.0 SP6. La version 4.0 construite par Microsoft apporte une mise à jour complète de l'interface ainsi qu'une meilleure intégration avec les technologies Microsoft. En 2011 sort la dernière version actuelle : Dynamics AX 2012 (AX 6.0). Elle inclut des améliorations supplémentaires au niveau de l'interface utilisateur mais surtout au niveau des applications orientées pour le commerce de détail, les divertissements et médias et le secteur public.

### *3.2.2. Positionnement du produit Dynamics AX sur le marché*

L'ERP Dynamics doit faire face à une forte concurrence sur le marché mondial. Il existe tout d'abord une offre open source avec un coût d'acquisition évidemment plus faible. Mais, il y a surtout la concurrence d'autres solutions propriétaires dont :

- SAP avec sa gamme SAP Business qui est le leader mondial. Utilisée surtout par de grandes entreprises, l'objectif affiché de SAP est de se développer maintenant sur le marché des petites et moyennes entreprises (inférieure à 250 salariés) en répondant avec sa suite BusinessOne.
- Oracle avec sa gamme Oracle Fusion Applications.
- ERP SAGE qui vise en particulier le marché des PME où l'éditeur est bien implanté. Les logiciels de SAGE visent en particulier les entreprises de moins de 500 salariés. Cependant, en novembre 2005, Sage rachète l'éditeur Adonix pour s'ouvrir le marché des PME/PMI de 500 à 2 000 salariés.

Au niveau mondial, selon un article du site [erp-infos.com \(http://www.erp-infos.com/info\\_article/m/1544/le-combat-des-chefs.html\)](http://www.erp-infos.com/info_article/m/1544/le-combat-des-chefs.html) paru en 2011, on constate qu'au final trois leaders se retrouvent le plus souvent dans la « short-list » des choix des entreprises : SAP dans 40 % des cas, Oracle dans 30 % et Microsoft Dynamics dans 20 %. Néanmoins, avec la gamme Dynamics AX, Microsoft affiche tout de même environ 20 000 clients dans le monde.

Au niveau national, en 2005, SAP et Oracle représentent près des 2/3 du marché en valeur (Tomas, 2007). Mais toujours selon le même auteur, la pénétration des très nombreuses PME/PMI doit permettre notamment à Microsoft de rattraper une partie de leur retard dans les prochaines années. Parallèlement, la tendance actuelle du marché des ERP s'oriente vers la verticalisation (ou approche sectorielle) de l'offre des éditeurs (Tomas, 2007). Le but est ainsi de rendre le déploiement d'un ERP plus rapide, plus simple, moins coûteux et plus accessible aux entités de taille moyenne et petite. C'est dans ce contexte qu'en France, en 2012, les intégrateurs Bull (acquis par Atos Origin en 2014) et Gfi Informatique signe un accord avec Microsoft pour adapter Dynamics CRM et AX pour le secteur public.

Cette première partie a eu donc comme objectif de montrer le contexte général dans lequel j'ai exercé ma mission de fin d'étude d'ingénieur dans la branche Software de l'entreprise Gfi Informatique.

En effet, j'ai commencé d'abord par présenter la société et son rôle historique d'intégrateur d'ERP. Puis, j'ai poursuivi sur sa forte implémentation sur le secteur public qui a alors mené sur un accord avec Microsoft pour implémenter son ERP Dynamics AX dans ce secteur.

J'ai présenté ensuite dans les grandes lignes les clients de la solution Gfi-SSP. Enfin, j'ai terminé par une approche du concept d'ERP, un historique rapide sur Dynamics AX et sa position sur le marché actuel.

Maintenant, dans la partie qui suit, je vais aborder un état de l'art du métier de l'intégration de Dynamics AX. Le but est en fait de soulever les problématiques essentielles que l'on peut rencontrer dans les différentes activités de ce processus complexe.

## **2<sup>ème</sup> Partie : Un état de l'art sur l'intégration de Dynamics AX**

L'objectif de cette seconde partie est de réaliser un état de l'art sur l'intégration technique de Dynamics AX. Il est effectivement primordial de réaliser cette étape pour pouvoir répondre à la problématique posée en introduction.

Par conséquent, je vais donc m'intéresser au métier d'intégrateur technique ERP. Je verrai ainsi les différents rôles et compétences de ce métier, les qualités requises, ses nombreuses missions et enfin, la nécessité de travailler en mode projet pour réussir les différentes tâches qui lui incombent.

Deuxièmement, je présenterai l'architecture du produit Dynamics AX. Après une présentation de l'architecture logique de la solution, je terminerai par celle des principaux composants.

Enfin, dans un dernier point, je mettrai en évidence les grandes problématiques que soulève l'intégration de la solution Gfi-SSP chez nos clients.

## **4. Un métier transverse**

Dans le premier point de cette deuxième partie, je commencerai donc par traiter la question du métier d'intégrateur technique ERP. En effet, étant amené maintenant à exercer ce métier, je mènerai une réflexion d'abord sur ses différents rôles, puis sur les qualités attendues pour ce travail, et enfin, sur la nécessité du mode projet pour réussir les tâches incombées.

### **4.1. Les rôles et compétences du métier d'intégrateur ERP**

L'équipe d'intégration technique doit sélectionner, produire, traduire, transposer les différents éléments de l'ERP et les assembler au système d'information en fonction de l'architecture retenue. L'équipe d'infrastructure technique est composée uniquement d'informaticiens pleinement dédiés au projet ERP (Tomas, 2007). Néanmoins, de façon ponctuelle, il est possible qu'elle fasse appel à des consultants techniques pour compléter son expertise.

Ainsi, l'équipe infrastructure technique possède plusieurs rôles qui soulignent sa transversalité. Tout d'abord, selon Tomas (2007), son rôle principal demeure à s'assurer que chacun des composants techniques est en place, compatible avec l'ensemble et fonctionne correctement dans le contexte du projet (durant l'implémentation, à la mise en production puis pendant toute la vie du système). Parmi ses composants techniques, on peut notamment lister les systèmes d'exploitation, les bases de données, le réseau, les protocoles de communication et les composants mêmes de l'ERP.

Elle a aussi un rôle de recommandation et participe aux choix des différents composants techniques nécessaires à la mise en place de l'ERP. Elle influence ainsi grandement les décisions majeures d'infrastructures grâce à son expertise. De plus, elle a la responsabilité du support technique du projet au sens large. Elle définit effectivement les différents paramètres ainsi que les procédures d'exploitation de l'environnement technique afin de permettre la viabilité de l'ERP. Elle participe enfin au déploiement de l'ERP qui selon Lequeux (2008) est l'opération la plus lourde et la plus délicate. C'est pourquoi, pour exercer ce métier, il est nécessaire d'avoir certaines qualités.

Ce métier demande tout d'abord une grande connaissance du système d'information et une assez grande expérience. La rigueur, la méticulosité et la logique sont des qualités majeures dans ce métier pour faire face aux différentes responsabilités. De même la souplesse, l'astuce et la rapidité permettent de contourner les difficultés qui peuvent survenir. En outre, le rôle central de l'intégrateur entre l'équipe de développement et l'équipe des consultants demande une certaine capacité au dialogue, une grande ouverture d'esprit, d'adaptabilité mais aussi de résistance à la pression. Ces qualités alors réunies, les charges de l'équipe d'intégration technique ERP peuvent être abordées plus aisément. Mais quelles sont ces différentes charges ?

#### **4.2. Les charges de l'intégrateur**

Le métier d'intégrateur technique ERP passe d'abord par une bonne compréhension du mode projet. En effet, l'intégration est d'abord un projet dans le projet qui se décompose lui-même en sous-projet et qui ont pour résultat des livrables. Selon Deixonne (2011), chaque sous-projet se décompose ensuite de phases identiques : lancement, conception de la solution, résolution de la solution, intégration de celle-ci et passage en production.

Les charges affectées à l'équipe d'infrastructure technique ERP sont multiples et réparties dans plusieurs phases du projet. Dans le Tableau III, j'ai réuni par exemple toutes les tâches possibles que notre équipe est amenée à réaliser en indiquant les rôles et les responsabilités des intervenants à partir de la matrice RACI (Responsible, Accountable Consulted Informed). Le R réalise l'action. Il doit y avoir au moins un R pour chaque action. Le A est celui qui doit rendre des comptes sur l'avancement de l'action. Il est totalement responsable d'une action. Il y a toujours un A (et un seul) pour chaque action. Les C sont les entités qui doivent être consultées. Les I sont les entités qui doivent être informées.

Grâce à ce tableau, il est ainsi possible de se rendre compte que les charges listées soulignent toute la complexité de mise en œuvre. C'est pourquoi, une planification et un bon enchaînement des tâches entre les différents acteurs du projet s'avère être crucial.

**Tableau III - Enchaînement des tâches de notre équipe infrastructure PIT Dynamics AX**

TACHE	LIBELLE	TACHES PRECEDENTES	R	A	C	I
<b>ETUDE ET ANALYSE</b>						
A1	Echanges préliminaires	--	Gfi	Gfi		
A2	Collecte et réunion des informations du périmètre technique	A1	Gfi	Gfi	Client	
A3	Analyse et définition de la structure du système cible	A2	Gfi	Gfi	Client	
A4	Définition et validation de l'architecture technique	A2	Gfi	Gfi	Client	
A5	Vérification des prérequis d'installation	A4	Client	Gfi	Gfi	
A6	Rédaction du Document d'Architecture Technique version 1.0	--	Gfi	Gfi	Client	
<b>INSTALLATION ENVIRONNEMENTS HORS PRODUCTION</b>						
B1	Préparation de la plate-forme hors production	A3, A4	Client	Client		Gfi
B2	Validation de la plate-forme hors production	B1	Gfi	Gfi	Client	
B3	Installation de l'environnement d'Intégration	B2	Gfi	Gfi		Client
B4	Tests fonctionnels et techniques sur Intégration	B3	Gfi	Gfi		Client
B5	Installation de l'environnement de Test	B1	Gfi	Gfi		Client
B6	Installation de l'environnement de préproduction	B1	Gfi	Gfi		Client
B7	Mise en place de la sauvegarde des environnements	B6	Client	Client	Gfi	
B8	Rédaction du Document d'Installation Technique version 1.0	B3, B5, B6	Gfi	Gfi	Client	
B9	Support technique des environnements hors production	B6	Gfi	Gfi		
B10	Déploiement des mises à jour	B6	Gfi	Gfi		Client
B11	Mise en place de la supervision des environnements	B6	Client	Client	Gfi	
<b>INSTALLATION ENVIRONNEMENTS PRODUCTION</b>						
C1	Préparation de la plate-forme production		Client	Client		Gfi
C2	Validation de la plate-forme production	C1	Gfi	Gfi	Client	
C3	Installation de l'environnement production	C2	Gfi	Gfi		Client
C4	Migration des données vers production	C3	Gfi	Gfi		Client
C5	Tests fonctionnels et techniques sur production	C4	Gfi	Gfi		Client
C6	Rédaction du Document d'Installation Technique version 2.0	C4	Gfi	Gfi	Client	
<b>DEMARRAGE EN PRODUCTION</b>						
D1	Assistance technique au démarrage	C5	Gfi	Gfi		Client
D2	Transfert de compétences	D1	Gfi	Gfi		Client
D3	Mise en place de l'optimisation technique	D1	Gfi	Gfi		Client
D4	Rédaction du Document d'Exploitation Technique version 1.0	D1	Gfi	Gfi	Client	

Ces tâches spécifiques à notre structure technique au sein de l'activité Gfi-SSP s'inspirent d'une méthodologie propre à Microsoft : Sure Step. Cette dernière s'applique spécialement aux déploiements des projets Dynamics que je vais présenter maintenant.

### 4.3. La méthodologie Sure Step

Comme le montre la figure suivante (Figure 3), elle se constitue de six phases principales (Diagnostic, Analyse, Conception, Développement, Déploiement, Exploitation)

qui présentent les étapes d'avancement du projet et de deux phases additionnelles post-projet (Optimisation et Mise à jour).



**Figure 3 - Schéma des phases du projet Dynamics définies selon Sure Step adaptée de Shankar et Bellefroid (2011)**

En outre, il faut savoir que Sure Step définit une vingtaine de rôles des différents intervenants en indiquant les compétences nécessaires et les tâches types (pour de plus amples détails, voir en [Annexe](#) les rôles de l'architecte et du consultant technique).

Cette méthodologie emprunte beaucoup d'éléments au référentiel ITIL (Information Technology Infrastructure Library) et à la méthodologie de gestion de projet PMI (Project Management Institute). Néanmoins, certains processus ne sont décrits que de manière superficielle comme les tests unitaires ou les tests de validation. Une grande partie des livrables proposés restent assez superficiels et ne sont que de simples documents Word. Enfin, une certification et une formation sur cette méthodologie peuvent s'avérer très coûteuse pour une entreprise. Toutefois, pour notre équipe infrastructure, Sure Step nous sert de référentiel pour construire et suivre nos projets d'implémentation technique chez nos clients.

Le métier d'intégrateur technique ERP est donc incontestablement un métier central et transverse par les différents rôles qu'il occupe. En outre, les charges qui lui incombent soulignent la nécessité d'avoir des qualités spécifiques pour pouvoir les réaliser dans les multiples projets plus ou moins complexes auxquels il prend part. Je vais, à présent, dans le point suivant aborder la complexité de l'ERP Dynamics AX à travers son architecture technique.

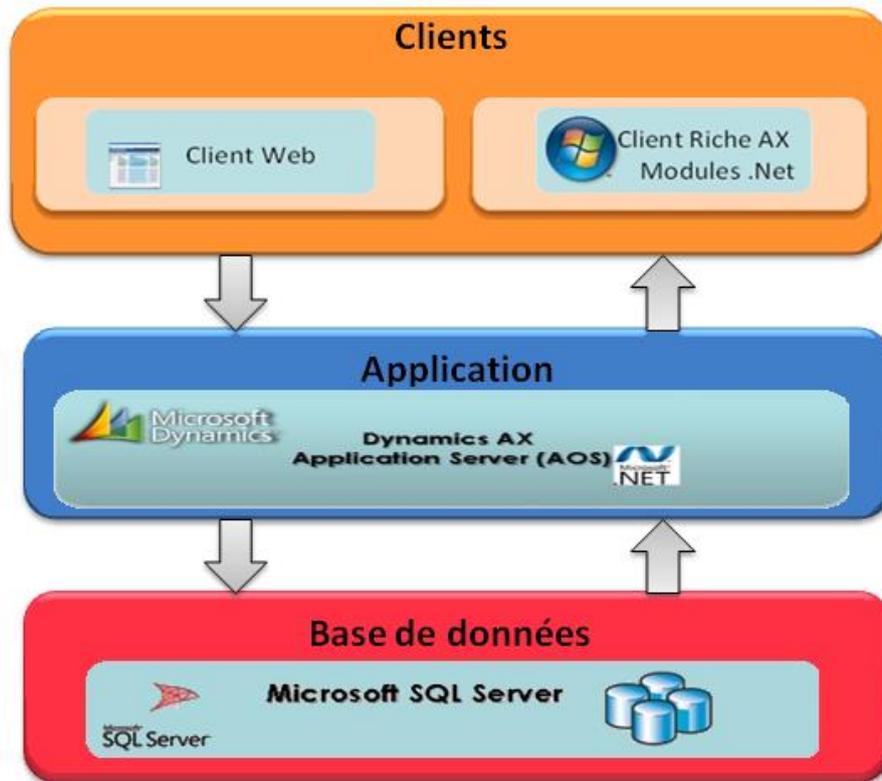
## 5. L'architecture de Dynamics AX

Dans ce paragraphe, je vais maintenant présenter l'architecture et les composants principaux de Dynamics AX dans le but de montrer la grande complexité de son intégration technique.

### 5.1. L'architecture logique de la solution

Dynamics AX possède une architecture basée sur un modèle 3-tiers (<https://msdn.microsoft.com/en-us/library/aa660629.aspx>) :

- Comme le montre la Figure 4, il y a d'abord le tiers base de données. La base de données Microsoft SQL Server héberge les données saisies qui sont utilisées par l'application Dynamics AX. Elle stocke également le magasin de modèles qui contient les métadonnées ;
- Au niveau intermédiaire, le tiers application avec le serveur d'objets d'application (AOS) qui est un service qui exécute en continue la plus grande partie de la logique du système ;
- Enfin, le tiers client qui représente le moyen par lequel l'utilisateur accède à Dynamics AX (client riche, client web,...). Le client sert d'interface avec les données et les fonctionnalités Dynamics AX.

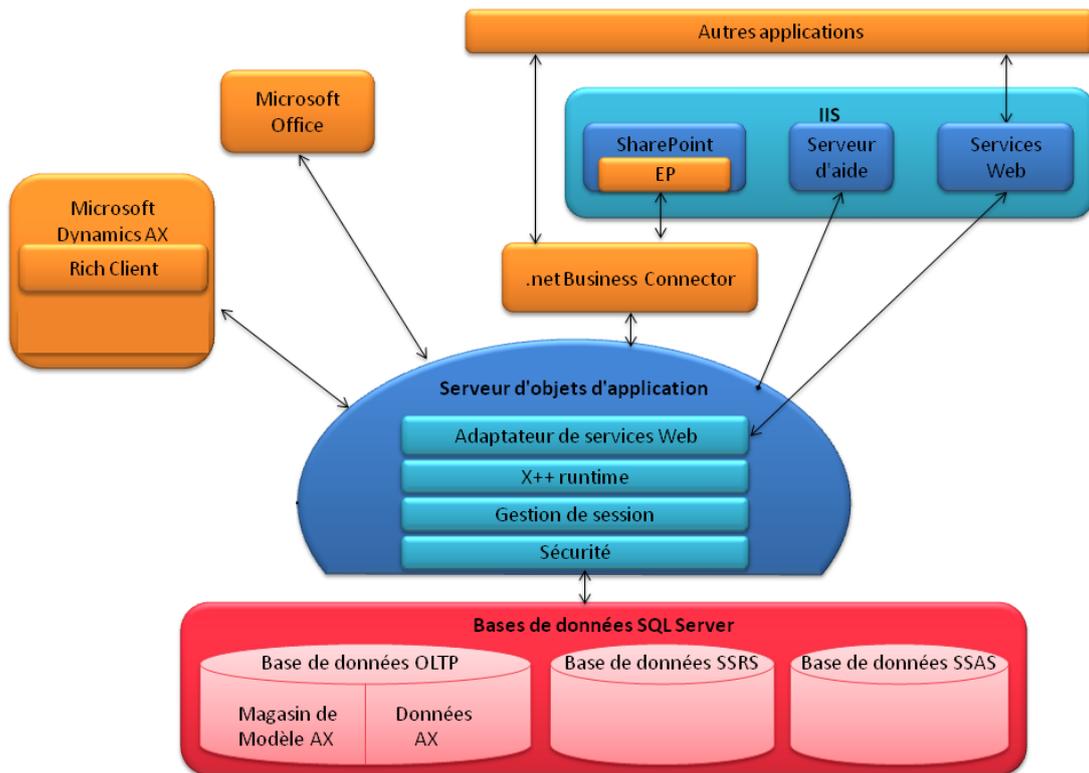


**Figure 4 - Modèle 3-tiers de l'architecture Microsoft Dynamics AX**

Outre ces trois briques essentielles, d'autres peuvent s'ajouter pour enrichir la pile des technologies couvrant d'autres besoins en termes de Business Intelligence ou de Reporting par exemple. Ce schéma de l'architecture de la solution Dynamics AX me permet ainsi d'effectuer une transition sur les composants que je vais expliciter plus en détail dans le point suivant.

## **5.2. Les composants de la solution**

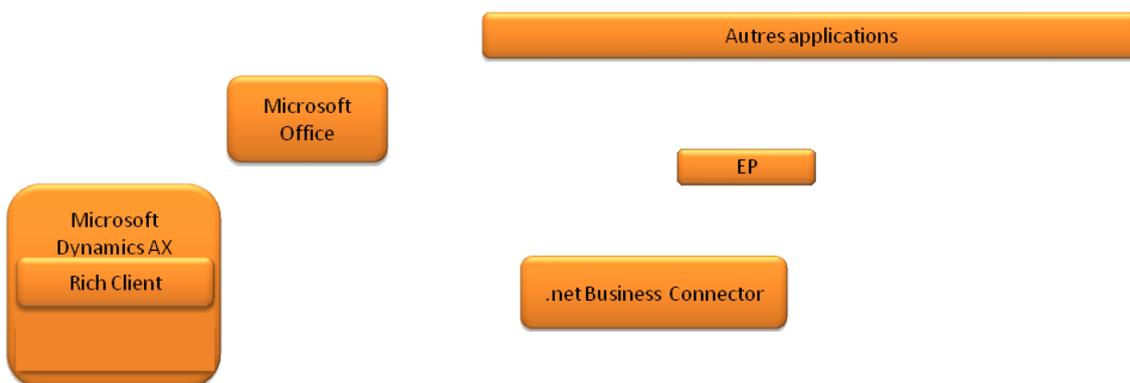
En effet, la solution Dynamics AX se décompose en plusieurs composants qui interagissent entre eux. Je vais donc expliquer les principaux d'entre eux suivant leur couche d'appartenance. Le schéma ci-dessous (Figure 5) permet de donner une description plus détaillée concernant les types de technologies pouvant être inclus dans chaque couche.



**Figure 5 - Schéma des composants applicatifs de la solution Microsoft Dynamics AX**

### 5.2.1. La couche client

Les différents types de clients permettant d'accéder aux fonctionnalités et données de l'ERP sont les suivants :



**Figure 6 - Composants de la couche client**

- Le client riche (Microsoft Dynamics AX Rich Client) permet d'accéder de manière exhaustive à l'ensemble des fonctionnalités du produit ;

- Les clients Web offre aux utilisateurs la capacité d'accéder aux fonctionnalités de Dynamics AX au sein du composant Enterprise Portal ;
- Des applications externes peuvent être employées comme client de l'application. L'interaction avec Dynamics AX se fait à travers l'Application Integration Framework (AIF) qui fournit un cadre pour les échanges dans le cadre d'un EAI (Entreprise Application Integration) basé sur XML.
- .NET Business Connector est un client sans interface utilisateur.

### 5.2.2. *La couche application*

La couche application se compose d'un ou de plusieurs des éléments ou rôles suivants :

- Contrôleur de domaine Windows Active Directory : Dynamics AX utilise l'authentification intégrée de Windows pour authentifier les utilisateurs Active Directory ;
- Serveur d'objets d'application (AOS) : il contrôle la communication entre les clients, les bases de données et les applications Dynamics AX. Le serveur d'application de Dynamics AX fournit les fonctionnalités suivantes : traitements métiers, AIF et batch. Le serveur AOS effectue beaucoup de tâches dont l'exécution du code (X++), la sécurité ou encore la gestion des sessions.
- Portail d'entreprise : Dynamics AX fournit un ensemble de sites Web donnant l'accès aux données de l'application. Ces sites sont appelés collectivement Enterprise Portal.
- Reporting (Microsoft SQL Server Reporting Services – SSRS) : permet aux utilisateurs de créer et de consulter des rapports sur papier ainsi que des rapports interactifs. Pour intégrer Dynamics AX et Reporting Services, il faut installer les extensions de production de rapports sur un serveur exécutant Reporting Services.
- Business Intelligence (Microsoft SQL Server Analysis Services – SSAS) : fournit les fonctionnalités décisionnelles à travers les cubes analytiques. L'intégration de Dynamics AX et Analysis Services se fait à travers l'installation d'extensions d'analyse sur un serveur exécutant Analysis Services.

### 5.2.3. La couche base de données

Dynamics AX nécessite SQL Server pour la base données. L'instance SQL Server héberge les bases des différents composants suivants :



**Figure 7 – Couche base de données**

- La base de données transactionnelle Dynamics AX : c'est la base de données de traitement transactionnel en ligne (OLTP) qui enregistre toutes les données transactionnelles ;
- La base de données modelstore : le magasin de modèle est une base de données où sont stockés tous les éléments d'application Dynamics AX y compris les personnalisations. La couche et le modèle d'informations font partie intégrante du magasin. L'AOS a accès à la boutique de modèle, gère la gestion des couches et fournit des données de modèle à tous les sous-systèmes de Dynamics AX tels que formulaires, rapports et code X++.
- La base de données SSAS (nécessaire pour les cubes OLAP) : la base de données SSAS ou Business Intelligence (BI) est répliquée à partir de la base de données OLTP dans un format optimisé pour accélérer la génération d'états et l'analyse ;
- La base de données SSRS utile au serveur de rapports ;
- Les bases de données SharePoint (contenu et configuration) ;

La présence de ces nombreux composants montre par conséquent la complexité de l'architecture de Dynamics AX. C'est pourquoi, afin de réussir au mieux l'implémentation technique, il est nécessaire de connaître les différentes problématiques posées par l'intégration de cet ERP. C'est l'objet de ma présentation dans le point suivant.

## **6. Les problématiques posées par l'intégration de Dynamics AX**

L'intégration d'un ERP génère en effet de multiples problématiques. Il est donc évident qu'il faut les déterminer en amont afin de les cerner pour prévenir toutes difficultés. De plus, il est fort possible que l'on rencontre les mêmes problématiques dans tous les projets.

### **6.1. La virtualisation**

Une des premières problématiques qui est soulevée par l'implémentation de Dynamics AX chez nos clients est la question de la virtualisation. En effet, les raisons de la virtualisation sont évidemment multiples et intéressantes à plus d'un titre pour les clients (réalisation d'économies, réduction du coût de la maintenance,...). Ensuite, il est à souligner que la solution Dynamics AX s'avère rapidement coûteuse en machine physique, en ressources mémoire et en stockage. C'est pourquoi, la virtualisation semble être pour les clients la solution la plus intéressante.

Cependant, il existe des préconisations en termes de virtualisation définies par Microsoft. Dynamics AX est ainsi certifié pour fonctionner sur une plateforme Windows Server avec la solution de virtualisation de Microsoft nommée Hyper-V. L'ERP est aussi supportée sur d'autres produits comme VMware selon la politique de support Microsoft (Microsoft Support Lifecycle). Dans celle-ci, Microsoft fournit une liste exhaustive et actualisée des hyperviseurs certifiés SVVP (Server Virtualization Validation Program).

Toutefois, selon une étude de Microsoft, le scénario de virtualisation coïncide avec une perte des performances allant jusqu'à 20% même avec un paramétrage optimisée. C'est pourquoi, s'agissant du serveur de données et/ou du serveur AOS, il est possible de les déployer plutôt sur des machines physiques. Sinon, il est également plausible de compenser cette perte de performance en augmentant de 20% les ressources physiques. Dans tous les cas, il est donc évident qu'il faut prendre en considération le contexte client (taille, budget, ressources,...).

Enfin, je terminerai ce sujet en évoquant que tous nos clients ont opté pour la solution complète de virtualisation (six avec le produit VMware et deux avec Hyper V). En outre, depuis le premier trimestre de 2014, en collaboration avec des partenaires certifiés comme Gfi, Microsoft propose Dynamics AX 2012 R3 en hébergement virtualisé sur Windows Azure (nom de la plateforme cloud de Microsoft). Ce service IaaS (Infrastructure comme service) permet ainsi d'avoir le contrôle des machines virtuelles et de la configuration réseau sans se soucier du matériel.

## 6.2. Les besoins réseaux

La problématique suivante concerne le débit et la qualité du réseau. Mesurée en mégabits par seconde (Mbps), la bande passante est donc la quantité de données qui peuvent passer de la source à la destination par unité de temps, en moyenne sur de nombreuses transactions. Mais, la latence (la vitesse) a aussi son importance. Mesurée en milliseconde (ms), la latence est le temps qu'il faut pour une seule transaction de données pour s'exécuter : le temps entre l'envoi des données sur l'extrémité de la source et sa réception à l'extrémité distante. La latence mesure donc le temps minimum possible pour le transfert d'un paquet sur la connexion.

La bande passante et la latence réseau sont donc des facteurs importants à prendre en compte dans la performance de Dynamics AX à cause de la nature du trafic réseau entre le client, l'AOS et SQL Server. Le client AX communique effectivement en utilisant des petits paquets de données entre lui et le serveur AOS. En retour, ce dernier fait des petites requêtes similaires vers SQL Server au nom de tous les clients. Microsoft a répertorié les besoins réseaux minimaux des connexions du client vers le serveur d'application (AOS), et du serveur AOS vers le serveur de base de données (Tableau IV).

**Tableau IV - Tableau des prérequis réseau par Microsoft pour Dynamics AX**

	CLIENT VERS AOS	AOS VERS BASE DE DONNEES
<b>Bande passante (b)</b>	100 Mbps (Mégabits par seconde)	100 Mbps (Mégabits par seconde)
<b>Latence (l)</b>	< 5 ms	< 5 ms

Par conséquent, une faible bande passante (inférieure à 100 Mbps) ou une latence élevée (supérieure à 5 ms) peuvent entraîner des temps de réponse lents sur le réseau.

L'enjeu des besoins réseau montre ainsi la nécessité de bien évaluer le contexte réseau client afin de déceler des risques éventuels de baisse de performance avec Dynamics AX. Une fois les résultats analysés, il faut donc faire au client des préconisations afin de résoudre ou contourner ses problèmes réseau. Après cette étape, on peut évoquer avec le client la question du mode de déploiement à privilégier.

### **6.3. Le mode de déploiement**

Lié au problème réseau, le mode de déploiement constitue un autre enjeu à traiter lors de l'analyse et la définition de la structure cible du client. Mais qu'est-ce que couvre la notion de déploiement ? Ce thème définit la manière dont va être implémenté Dynamics AX du côté serveur et du côté client. Evidemment, le mode de déploiement reste spécifique au contexte client (répartition géographique des sites, type de réseau,...). Microsoft préconise de tenir compte de deux types de connexions :

- Dans le cas d'un réseau local (LAN), il faut se connecter directement via un client Microsoft Dynamics AX (client riche). Le déploiement peut alors se faire à l'aide d'une stratégie GPO (Group Policies Object) lançant une installation silencieuse sur le poste utilisateur.
- Dans le cas d'un réseau étendu (WAN), il faut se connecter à l'aide d'une solution de connexion à distance. Ainsi, l'exécution de la solution Dynamics AX se comporte comme si elle était exécutée sur l'ordinateur local de l'utilisateur final (client léger).

La question du mode de déploiement chez le client évoquée, il est ensuite possible de se pencher sur le problème du stockage. A partir des préconisations Microsoft, quelles sont en effet les solutions qu'il faut proposer pour garantir les meilleurs temps de réponse en I/O (Input/Output) ou E/S (Entrées/Sorties).

## 6.4. Le stockage

Le stockage demeure incontestablement un sujet sensible dans l'intégration technique de Dynamics AX. Toutes erreurs dans la mise en œuvre du stockage risquent d'avoir des répercussions sur la performance et peuvent entraîner des coûts.

L'ERP de Microsoft utilise le stockage de manière intense. Outre les requêtes, il utilise la fonctionnalité « Read Committed Snapshot Isolation » (Isolement de capture instantanée) qui améliore l'accès concurrentiel aux applications OLTP (Traitement transactionnel en ligne) et peut avoir des tables temporaires stockées dans TempDB de SQL Server. La taille de TempDB peut donc influencer sur les performances du système. Si elle est trop faible, la capacité de traitement du système peut être trop occupée à faire croître automatiquement la base de données pour qu'elle prenne en charge les impératifs de la charge de travail à chaque démarrage de SQL Server.

En général, dans le cadre de de la solution Dynamics AX, on recommande l'utilisation de disques en RAID 10, RAID 0+1 ou le cas échéant en RAID 1 (Tableau V). RAID désigne un groupe de deux disques minimums gérés sous la forme d'une seule unité, qui sert à enregistrer les données accompagnées d'informations supplémentaires ou redondantes afin d'assurer une récupération dans le cas d'une défaillance de disque. Dans un système RAID, un disque défaillant peut normalement être remplacé alors que le serveur fonctionne encore.

**Tableau V - Avantages et inconvénients des systèmes RAID retenus pour Dynamics AX**

NOM	DESCRIPTION	AVANTAGES	INCONVENIENTS
<b>RAID 10 ou 1+0</b>	Il permet de combiner, le niveau de RAID 1 et le niveau de RAID 0. En premier lieu, on crée deux grappes RAID 1, qui constitueront elles-mêmes la grappe RAID 0.	<ul style="list-style-type: none"> <li>- Gain en performance : amélioration en I/O sans risque de perte de données.</li> <li>- Fiabilité assez grande et reconstruction assez performante.</li> </ul>	<ul style="list-style-type: none"> <li>- 50% seulement de l'espace disque utile est disponible comme le RAID 1</li> <li>- Solution la plus coûteuse (réservée à des applications ayant des exigences particulières en termes de performance et de fiabilité).</li> </ul>
<b>RAID 0+1</b>	Système à deux niveaux : RAID 1 (la mise en miroir) et RAID 0 (l'agrégation).	<ul style="list-style-type: none"> <li>- Equilibre entre la disponibilité des données et l'augmentation des performances de lecture.</li> </ul>	<ul style="list-style-type: none"> <li>- Tolérance aux pannes moins élevée que RAID 10.</li> <li>- Immobilisation d'au moins 4 disques durs.</li> </ul>
<b>RAID 1</b>	Il consiste en l'utilisation de n disques redondants, chaque disque de la grappe contenant à tout moment exactement les mêmes données, d'où l'utilisation du mot « miroir ».	<ul style="list-style-type: none"> <li>- Excellent niveau de protection des données.</li> <li>- Grande tolérance aux pannes.</li> </ul>	<ul style="list-style-type: none"> <li>- Pas de gain de performance.</li> <li>- Coût élevé : 50% seulement de l'espace disque utile est disponible, l'autre moitié dédiée à la sauvegarde.</li> </ul>

Les bases de données SQL Server doivent être localisées sur des disques physiques entièrement dédiés pour SQL Server. Pour de bonnes performances, il est recommandé d'éviter le partage avec d'autres applications permettant de mieux gérer le SAN (Storage Area Network). Le SAN est un réseau spécialisé permettant de mutualiser des ressources de stockage. Les recommandations du fournisseur du stockage SAN sont donc à prendre en compte. Pour la taille du secteur disque, il est préconisé d'avoir une valeur entre 8 Ko et 64 Ko (ou toute valeur recommandée par le fournisseur de stockage SAN). Le nombre doit être un multiple de 8 car SQL Server travaille avec des pages de 8 Ko de taille.

Afin de garantir les meilleurs temps de réponse, il est par conséquent nécessaire de fournir les meilleures capacités I/O. Pour donner une idée générale d'une configuration typique de l'espace requis pour l'installation de Dynamics AX, le tableau suivant (Tableau VI) précise l'IOPS (Opérations d'entrées-sorties par seconde) nécessaire pour les données SQL, le journal de log et la base TempDB :

**Tableau VI - IOPS nécessaire pour le stockage de Dynamics AX**

<b>STOCKAGE</b>	<b>IOPS RECOMMANDE</b>	<b>RAID RECOMMANDE</b>
Données	~750	RAID 1+0
Journal de log	~1000	RAID 1
TempDB	~1000	RAID 1

Concernant la taille pour le stockage, les besoins réels varient en fonction de la configuration du système et sur la base des applications et fonctionnalités qui ont été décidé d'être installé. De nombreux facteurs influent évidemment sur le choix du matériel approprié pour une installation de Dynamics AX comme le nombre de transactions par heure, le nombre d'utilisateurs simultanés ou encore le nombre de connexions à distance. De plus, de l'espace disque supplémentaire est nécessaire pour le reporting (SSRS) et la Business Intelligence (SSAS). Il faut penser alors à ajouter environ 10 à 20% de la taille de la base de production pour les cubes et les données temporaires.

Enfin, en cas de virtualisation du stockage, il faut noter que cela peut générer des problèmes au niveau de la performance. Effectivement, en environnement virtualisé, le stockage se présente comme un volume logique homogène partagé. De plus, comme le nombre d'accès concurrents devient plus important, des goulots d'étranglement peuvent alors

apparaître. C'est pourquoi, il faut choisir une bonne solution de virtualisation de stockage en mettant en place par exemple des mécanismes de basculement afin d'obtenir une bonne répartition de la charge.

En conclusion, les recommandations principales de stockage à retenir sont : être conforme aux bonnes pratiques sur les entrées-sorties (I/O), utiliser si possible le RAID 10 et accorder autant d'importance à la performance de TempDB qu'à celle de la base de données principale. Il est temps maintenant de voir une autre problématique : la sécurité.

## **6.5. La sécurité**

La sécurité pour les clients est un enjeu qu'il ne souhaite pas négliger. Quels sont donc les préconisations par composants dans ce domaine ?

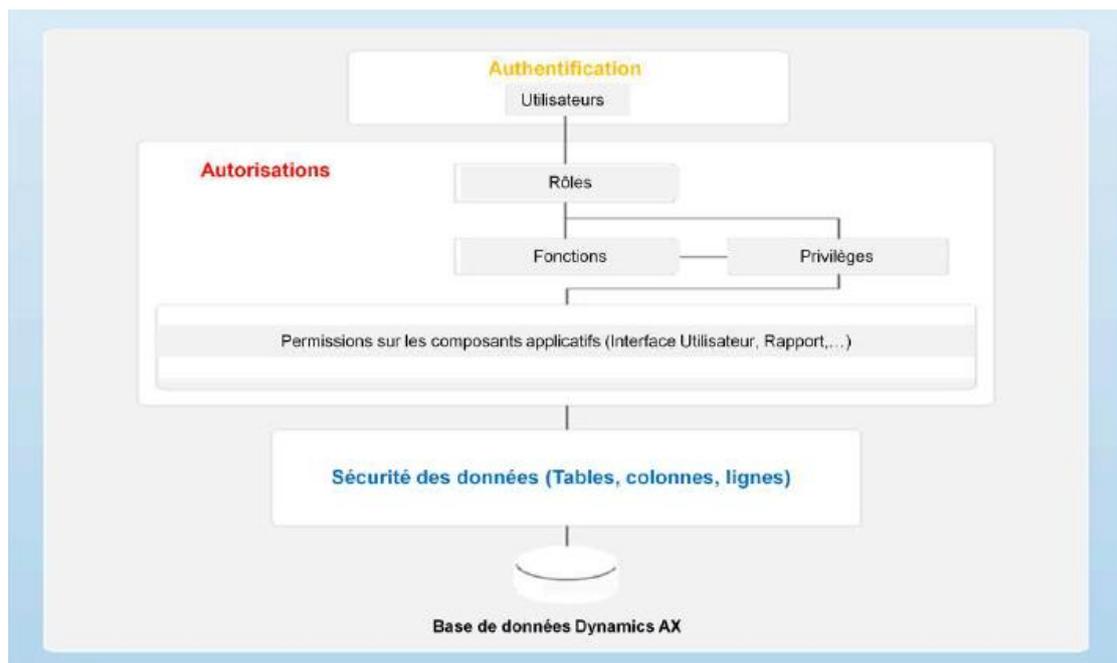
Tout d'abord au niveau client, le déploiement sur client léger est évidemment plus sécurisé que celui en local. En effet, aucune donnée n'est traitée, mise en cache, stockée et transmise sur le réseau. Pour améliorer la sécurité sur les ordinateurs clients, il existe plusieurs recommandations telles que le déploiement d'une stratégie de groupe (GPO), la mise en place de l'option Encryption File System (EFS) en local,.... Enfin, l'authentification des utilisateurs est sécurisée au travers du serveur de domaine Active Directory.

Pour le composant AOS, la sécurité passe par l'utilisation d'un compte de service domaine dédié avec des droits restreints. Il est possible également de modifier les numéros des ports de communications TCP/IP, WSDL (Web Services Description Language) et NET-TCP utilisés par défaut par AOS. Enfin, il reste la possibilité d'activer le protocole IPsec (Internet Protocol Security) qui aide à protéger les réseaux contre les attaques.

Au niveau SQL Server, en plus des bonnes pratiques générales de sécurité (chiffrement des données sensibles et sécurisation des logs), il existe la sécurité des enregistrements sur la base des rôles. En outre, l'accès aux données par TPF (Table Permissions Framework) dans Dynamics AX qui peut permettre d'ajouter un niveau supplémentaire de sécurité aux tables qui stockent des données sensibles. Pour le composant

reporting, des paramètres de sécurité assignent des utilisateurs et des groupes au rôle « DynamicsAXBrowser » dans Report Manager, de restreindre l'accès à des répertoires et à des reports précis. Pour le composant analyse des cubes, la sécurité est indépendante de Dynamics AX. Il faut définir les utilisateurs aux rôles de la base de données directement dans Microsoft SQL Server Analysis. Des personnalisations sont également possibles pour limiter l'accès d'un rôle à des dimensions et cellules spécifiques dans un cube.

Enfin, au niveau de l'application Dynamics AX (Figure 8), seuls les utilisateurs authentifiés qui possèdent des droits dans l'application peuvent se connecter. Les rôles de sécurité sont utilisés afin de contrôler l'accès aux éléments individuels de l'application (menus, bouton d'action, reports, contrôles web, champs,..). Enfin, un dernier niveau de sécurité peut être utilisé pour refuser l'accès aux tables, champs et lignes de la base de données.



**Figure 8 - Architecture de la sécurité dans l'ERP Dynamics AX**

Le sujet de la sécurité dans Dynamics AX est par conséquent assez complet pour permettre de bien orienter le client dans des choix de mise en œuvre. Je vais à présent traiter une autre problématique aussi cruciale : la sauvegarde des données.

## 6.6. La sauvegarde

Une stratégie de sauvegarde et de récupération repose sur la détermination du RPO (Objectif de point de récupération). RPO désigne la durée maximum d'enregistrement des données qu'il est acceptable de perdre lors d'une panne. Pour répondre à cet objectif, il faut établir avec le client un plan en trois temps.

Dans un premier temps, il est d'abord nécessaire de commencer à délimiter le périmètre de la sauvegarde. Il faut alors spécifier les bases de données et les données du système de fichiers (données applicatives, données utilisateurs et configuration des serveurs). L'avantage offert par la solution Dynamics AX est d'avoir le code applicatif stocké dans une base de données (le magasin de modèle). Une sauvegarde de la base de données permet donc de faire aussi une sauvegarde des modèles associés à l'application.

Dans un second temps, il s'agit de réaliser la planification des opérations de maintenance SQL Server. Un plan de maintenance peut-être défini de la manière suivante (Tableau VII) :

**Tableau VII - Exemple de planning des sauvegardes des bases chez un client**

	<b>L</b>	<b>M</b>	<b>Me</b>	<b>J</b>	<b>V</b>	<b>S</b>	<b>D</b>
<b>1. Vérification intégrité</b>	21h00	21h00	21h00	21h00	21h00	21h00	21h00
<b>2. Réorganisation des index</b>	21h30		21h30		21h30		
<b>3. Mise à jour des statistiques</b>	22h30	22h30	22h30	22h30	22h30	22h30	22h30
<b>4. Backup complet des bases AX</b>	00h00				00h00		
<b>4. Backup différentiel (optionnel)</b>		00h00	00h00	00h00		00h00	00h00
<b>5. Backup du journal de transaction</b>	<b>tous les 4h de 9h à 18h (7/7)</b>						

Dans ce tableau, l'ordre dans lequel s'effectuent les opérations est très important car cela permet d'obtenir une sauvegarde parfaite des données. La sauvegarde des bases SQL Server sont réalisées à chaud. La sauvegarde du journal de transaction est faite sur une fréquence de 4 heures afin de répondre à l'exigence d'un RPO spécifié à 4 heures. Dans le cas d'un client qui possède des sites hors de la France métropolitaine, la problématique des plages horaires est alors à revoir pour adapter en conséquence la stratégie.

Enfin, le dernier temps du plan concerne le test des procédures de maintenance. Celles de sauvegarde doivent être testées au moins une fois par mois afin de garantir le bon état des

sauvegardes. Concernant la restauration des bases, cela peut permettre au client de savoir combien de temps peut-elle prendre. Il est enfin recommandé aux clients de bien documenter leurs procédures et d'en garder une copie dans leur manuel d'exploitation.

La sauvegarde des données Dynamics AX est donc incontestablement une problématique stratégique. Seule une bonne planification construite sur les contraintes du client peut donc permettre d'obtenir des sauvegardes fiables. Après avoir discuté de cette thématique, je vais aborder dans le point suivant la problématique de la gestion d'impression sous Dynamics AX.

### **6.7. La gestion des impressions**

La gestion des impressions peut apparaître au premier abord simple à traiter avec Dynamics AX. Mais, cette problématique des impressions devient complexe lorsque le contexte client cumule l'utilisation d'un accès à distance type RDS, une multitude d'imprimantes existantes et par conséquent de nombreux drivers d'impression. Ainsi, pour les impressions sous RDS, on peut distinguer deux cas de figure selon le mode d'accès choisi : serveur ou client.

Depuis le serveur, l'imprimante réseau est accessible depuis un nom de partage (*\\NomServeur\NomPartage*). L'avantage de ce mode est de garantir de bonnes performances si le serveur d'impression est sur le même LAN que l'imprimante. L'inconvénient est que le paramétrage de l'imprimante se fait selon le client et doit prendre en considération la contrainte LAN/WAN. L'installation des pilotes d'imprimante sur les terminaux fonctionne bien dans les scénarios où les dispositifs d'impression, serveurs d'impression, et les terminaux sont tous situés sur le même réseau local. Cependant, attacher une imprimante directement au serveur n'est pas recommandé dans le cas de répartition multi-site surtout pour des réseaux WAN. En effet, la vitesse d'impression peut être impactée fortement dans le cas d'une latence importante du réseau.

Depuis le client (principe de redirection d'imprimante), l'imprimante est directement liée au poste de travail. Même s'il paraît plus simple d'activer la redirection des imprimantes

locales dans la session RDS, il est important de bien s'assurer que le même pilote est présent sur le client et le serveur. La complexité de ce processus a donc conduit Microsoft à créer un mode d'impression (Easy Print) à partir de Windows Server 2008. Ainsi, par défaut, le serveur RDS tente d'utiliser le driver Easy Print si le client le supporte, sinon il trouve un pilote d'impression approprié dans sa liste de drivers.

La gestion des impressions dépasse évidemment le seul cadre de Dynamics AX. Mais, il est primordial de l'aborder avec le client qui choisit le déploiement RDS afin de lui montrer les avantages et les inconvénients sur le sujet. A présent, je vais traiter un tout autre thème : celui de la mise en place de la haute disponibilité sur les composants Dynamics AX.

## **6.8. La haute disponibilité**

La Haute Disponibilité (HA – High Availability) est un sujet critique car l'indisponibilité des services informatiques peut avoir un coût considérable pour les clients. Pour Dynamics AX, la HA signifie mettre en place un système pour ses composants conçu de manière à garantir un fonctionnement au plus près des 100% du temps à un niveau acceptable de performance. Mais comment se quantifie l'indisponibilité ?

Il y a d'abord le RTO (Objectif de temps de récupération) qui représente la durée maximale d'interruption acceptable pendant lequel une ressource informatique peut ne pas être fonctionnelle. Exprimé en unité de temps, ce chiffre est très important car c'est lui qui va piloter le choix des techniques de haute disponibilité à mettre en œuvre. Comme son nom l'indique c'est un objectif, c'est donc au niveau contractuel dans le SLA (Service Level Agreement) que tout se règle en général. Ce chiffre découle du précédent, mais sur une base annualisée, on prend le temps d'indisponibilité total sur le temps total annuel.

Voici par exemple des pourcentages théoriques (Tableau VIII) selon l'hypothèse d'un service disponible 365 jours, 7/7 et 24/24 (c'est-à-dire au total 525 600 minutes) :

**Tableau VIII - Taux de disponibilité théorique selon le type adapté du site**  
<http://technet.microsoft.com/en-us/library/cc748824%28v=office.15%29.aspx>

TYPE DE DISPONIBILITE	DISPONIBILITE	INDISPONIBILITE PAR AN
Non géré	90%	52560 mn (36,5 jours)
Géré	99%	5256 mn (3,5 jours)
Bien géré	99,9 %	500 mn (8 heures)
Tolérant les fautes	99,99%	50 mn (1h20)
Haute Disponibilité	99,999%	5 mn
Très Haute Disponibilité	99,9999%	0,5 mn (30 sec)
Ultra Haute Disponibilité	99,99999%	0,05 mn (3 sec)

Ce pourcentage de disponibilité se calcule à l'aide de la formule suivante (où  $x$  est le pourcentage,  $n$  le nombre total de minutes dans un mois donné et  $y$  le nombre total de minutes pendant lesquelles un système ou un service est indisponible) :

$$x = (n - y) * \frac{100}{n}$$

Mon travail d'architecte est donc ici de proposer aux clients un système extensible proche de la haute disponibilité en fonction de leurs moyens. La notion d'extensibilité (« scalability ») s'avère alors nécessaire. En effet, un service est « scalable » s'il peut répondre à une demande plus forte sans remettre en cause son architecture. On distingue donc deux types d'évolutions d'une plateforme : « Scale up » (mise à niveau de la configuration matérielle des serveurs) et « Scale out » (ajout de nouveaux serveurs d'un rôle existant).

Toutefois, la mise en place de la haute disponibilité possède indéniablement un coût (matériel, logiciel, licence, maintenance,...). Par conséquent, le challenge est ici de bien assimiler toutes les options possibles de HA par composant Dynamics AX. Puis, dans un second temps, il faut pouvoir être capable de proposer des architectures d'implémentations de HA en intégrant les contraintes des clients.

## 6.9. La supervision

Il est en effet nécessaire de mettre en place une solution permettant de gérer entièrement la solution Dynamics AX chez les clients. La supervision consiste à pouvoir surveiller le bon fonctionnement d'un système ou d'une activité par l'intermédiaire de

compteurs pertinents mettant en évidence des lenteurs ou des dysfonctionnements. Néanmoins, superviser Dynamics AX peut rapidement devenir un sujet complexe.

Effectivement, la complexité se manifeste d'abord par la diversité des sources supervisées. L'ERP est un « assemblage » de plusieurs composants. Par conséquent, pour chacun d'eux, il existe de nombreux compteurs de performance. De plus, il est possible de surveiller les journaux des événements Windows. Ces derniers sont des fichiers spéciaux qui enregistrent les événements significatifs sur la machine. Par défaut, Windows propose les principaux types d'événements suivants : application, sécurité, système et installation. Dynamics AX propose en plus la surveillance du journal des événements du Workflow (système qui fournit des fonctionnalités pour créer des flux de travaux individuels ou des processus business). Enfin, il reste la possibilité de superviser aussi les services pour savoir s'ils sont bien en exécution (AOS, SQL Server,...).

La deuxième difficulté se situe ensuite dans le choix d'une solution de supervision parmi la multitude des produits existants. Il existe des outils de base dans Windows comme Windows Performance Monitor (PerfMon) ou Ressource Monitor. SQL Server possède également des outils natifs tels que SQL Server Profiler. Dynamics AX en fournit aussi comme Performance Analyser et Trace Parser. Mais, la plus complète des solutions proposées par Microsoft reste SCOM (System Center Operations Manager) pour la supervision de Dynamics AX. Enfin, des alternatives libres basées sur Nagios sont plébiscitées par les clients.

Enfin, la dernière complexité concerne les alertes à mettre en place lorsqu'un compteur de performance atteint un seuil prédéfini. Ainsi, par exemple, on peut décider d'être averti par email quand on atteint des seuils critiques. Mais, lorsqu'un service s'arrête, on peut aussi imaginer de mettre en place la possibilité d'un redémarrage automatique. Comme on le voit, le chantier de la supervision occupe une dimension importante dans le bon fonctionnement de l'implémentation de Dynamics AX. Il est donc primordial de ne pas négliger ce projet dans le cadre général de l'intégration.

## 7. L'analyse des cas Gfi-SSP

Cette deuxième partie a donc permis de réaliser un état de l'art sur l'intégration technique de Dynamics AX. En effet, j'ai commencé par souligner la transversalité du métier d'intégrateur technique ERP à travers les différents rôles qu'il peut porter et ses diverses missions. Pour réussir celles-ci, des qualités sont idéalement requises ainsi qu'un fonctionnement en mode projet. Dans un second temps, j'ai exposé l'architecture technique de Dynamics AX afin de bien montrer la diversité des composants et la complexité de déploiement qui peut se poser. Puis, j'ai terminé cet état de l'art en m'arrêtant sur les grandes problématiques que soulève l'intégration de cette solution. Effectivement, ces différents sujets doivent être connus et maîtrisés par l'intégrateur afin de pouvoir maîtriser la qualité de son processus. Enfin, dans le cas des missions Gfi-SSP que j'ai effectuées, cet état de l'art m'a donné des éléments afin de pouvoir mener à bien la phase d'analyse.

En effet, du point de vue du contact avec les clients, l'apport de ce travail a permis de bien préparer l'approche des sujets que j'ai eu à traiter. Ces contacts s'établissaient de la manière suivante :

- Etape 1 : Prise de contact par courriel de la part du client pour fixer un rendez-vous téléphonique ou chez le client ;
- Etape 2 : Préparation de la réunion avec une fiche de collecte de données ;
- Etape 3 : Réunion avec le client afin de détailler le besoin souhaité ;
- Etape 4 : Rédaction d'un cahier des charges avec les informations collectées ;
- Etape 5 : Analyse du besoin.

D'un point de vue de la gestion de projets, l'approche Sure Step (Shankar et Bellefroid, 2011) a été un fil conducteur dans l'analyse des cas. Je me suis effectivement appuyé sur cette méthodologie pour créer par exemple la fiche standard de collecte de données à partir de plusieurs modèles proposées. Cette fiche était utilisée pour préparer les réunions avec les clients et de bien cadrer le périmètre des réunions. Les types de données récoltés concernaient les configurations réseaux, matérielles et logicielles avec l'objectif de vérifier la compatibilité avec l'ERP Dynamics AX. Ensuite, certaines informations étaient

intégrées ou mises à jours dans nos principaux livrables (Document d'Architecture Technique et/ou Document d'Exploitation Technique).

Au niveau des problématiques d'intégration, cette contribution m'a donné conscience de la complexité du processus d'implémentation et m'a apporté de la profondeur dans l'étape d'analyse des cas Gfi-SSP. Concrètement, cela m'a inspiré l'idée de construire une procédure standard d'analyse afin de bien appréhender les problèmes auxquels j'ai été confronté. Celle-ci se constitue des étapes suivantes :

- Etape 1 : Se documenter sur la problématique (site Microsoft Technet, blogs, ouvrages spécialisées sur la thématique,...) ;
- Etape 2 : Effectuer une synthèse sur le sujet en mettant en avant les avantages, les inconvénients, le coût et le niveau de complexité de mise en œuvre pour chaque solution ;
- Etape 3 : Corréler les informations de la synthèse avec les données clients pour présélectionner les solutions possibles dans son périmètre.

Enfin, au niveau des types de cas d'études traités, cet état de l'art m'a permis ainsi d'aborder de manière sereine d'abord le problème de la haute disponibilité et de la supervision des composants Dynamics AX. En effet, la première problématique m'a été confiée suite au futur démarrage d'un client en production. Le second a fait l'objet d'une demande client qui souhaitait pouvoir superviser la solution Gfi-SSP depuis un outil en sa possession. Le dernier type de problème diffère des deux premiers car il est parti d'un constat interne au cours des nombreux échanges avec mon tuteur sur le nombre croissant de clients Gfi-SSP et par conséquent du nombre d'environnements techniques à tenir à jour. Il était donc devenu indispensable de réfléchir comment passer de l'étape « artisanale » à une étape d'industrialisation. Les sections suivantes sont ainsi consacrées à présenter les solutions proposées et leur justification fondé par l'état de l'art présenté.

### **3<sup>ème</sup> Partie : Les études de cas Gfi-SSP**

Dans cette dernière partie, il s'agit maintenant de confronter certaines problématiques vues dans la deuxième partie dans le contexte client. Les projets suivants ont été réalisés lors de ma mission avec toujours comme ligne directrice la volonté d'augmenter la qualité du processus d'intégration de l'ERP Dynamics AX.

Par conséquent, je vais d'abord présenter ma première mission qui a consisté à optimiser l'industrialisation des mises à jour des environnements clients. Après la brève présentation du contexte de ce projet, je détaillerai les tâches que j'ai réalisées et les résultats obtenus. Je terminerai enfin par présenter la mise en place du livrable chez le client.

Ensuite, je parlerai du projet du déploiement de la haute disponibilité (HA) dans Dynamics. J'évoquerai tout d'abord les méthodes choisies pour traiter cette problématique. Ensuite, j'expliquerai comment j'ai réalisé une typologie HA pour Dynamics AX. Enfin, je mettrai en avant mon expérience d'une mise en œuvre de la haute disponibilité chez un client.

Pour terminer, j'aborderai la dernière mission que j'ai réalisée sur le sujet de la supervision de Dynamics AX. Après une présentation du projet, j'étudierai en détails les moyens et les solutions pour superviser cet ERP. Puis, dans un second temps, j'exposerai les résultats obtenus de la supervision d'une plateforme Dynamics AX en production.

## 8. L'industrialisation des mises à jour des environnements

Ce premier projet concerne les mises à jour des livrables Gfi-SSP et la copie des environnements Dynamics AX. Après la présentation du contexte, je déroulerai les tâches de ce projet jusqu'au déploiement dans les environnements clients.

### 8.1. Le contexte projet

Ce projet est né suite au nombre croissant des environnements Dynamics AX déployés chez nos clients. En effet, il existe en moyenne trois à quatre environnements par client qui sont à maintenir à jour tout au long des projets menés. Partant de ce constat, j'ai alors décidé de lister les problèmes existants dans :

- Le cycle de livraison : une mise à jour complète des livraisons Gfi-SSP a une durée assez longue (environ six heures) à cause essentiellement des étapes de compilation et de synchronisation dans Dynamics AX après l'importation des fichiers modèles.
- L'automatisation des outils : la procédure de mise à jour n'est que partiellement automatisée. Les scripts existants ne permettent pas d'enchaîner les opérations automatiquement. Par conséquent, la mise à jour demande la mobilisation d'une personne pour les exécuter.
- La méthodologie : les livraisons Gfi-SSP s'exécutent à l'heure actuelle directement dans les environnements utilisés par les équipes projets. Cela pose deux difficultés :
  - ✓ Tout d'abord, ce processus ne peut commencer qu'en soirée après 18h00 et donc se terminer assez tard ;
  - ✓ Ensuite, cela signifie que si la mise à jour se déroule mal (erreurs de compilation ou de synchronisation), il faut alors revenir en arrière en restaurant la base de données afin de ne pas bloquer les environnements.

Par conséquent, l'objectif qui m'a été confié est de mener un travail d'ingénierie permettant d'améliorer l'industrialisation des mises à jour. Pour y arriver, j'ai donc élaboré un planning d'ordonnancement de mes tâches à accomplir (Tableau IX). Le projet a duré en totalité un mois et demi environ réparti entre la mi-novembre 2014 et la mi-mars 2015.

**Tableau IX - Planification des tâches pour le projet d'industrialisation**

TACHE	LIBELLE	TACHES PRECEDENTES	DUREE EN JOURS
<b>ANALYSE ET OUTILS</b>			
<b>T1</b>	Mise en place des nouveaux outils de travail	--	1
<b>T2</b>	Analyse de l'existant et modélisation	--	3
<b>T3</b>	Mise en place d'un environnement de build	--	0,5
<b>CONCEPTION, DEVELOPPEMENTS ET TESTS</b>			
<b>T4</b>	Conception d'une nouvelle arborescence	T1, T2	5
<b>T5</b>	Conception du fichier de configuration client standard	T4	4
<b>T6</b>	Conception et optimisation des scripts de copie d'environnement	T4	9
<b>T7</b>	Tests et débogage des scripts de copie	T5, T6	3
<b>T8</b>	Conception et optimisation des scripts de livraison	T4	12
<b>T9</b>	Tests et débogage des scripts de livraison	T8	5
<b>DEPLOIEMENT EN CONTEXTE CLIENT</b>			
<b>T10</b>	Mise en place d'un environnement de build chez le client	T7, T9	0,5
<b>T11</b>	Déploiement du package V1.0 et adaptation au contexte d'un client	T10	0,5
<b>T12</b>	Tests et débogage du package V1.0	T11	1,5
			<b>45 jours</b>

Ce planning souligne évidemment que la majorité du temps a été consacré à l'écriture/réécriture des scripts et aux tests. Le contexte du projet présenté, je vais maintenant évoquer le premier travail que j'ai effectué : l'analyse et la compréhension de l'existant.

## 8.2. La mise en place des outils de travail

Avant de m'occuper de la conception des scripts, je me suis donc impliqué tout d'abord à mettre en place l'environnement technique permettant le développement de ce projet.

Ainsi, le premier outil qu'il a fallu mettre en place est un gestionnaire de source. En effet, il était nécessaire de gérer la complexité de la gestion des versions de fichiers développés au fur et à mesure de leur conception. Nous étions maintenant deux personnes dans l'équipe Infrastructure et il fallait donc pouvoir réorganiser notre travail pour que chaque membre de l'équipe puisse participer à l'amélioration des scripts en ne perdant pas les mises à jour réalisées par chacun.

C'est pourquoi, j'ai d'abord décidé de rechercher un logiciel capable d'effectuer ce type de travail collaboratif. J'ai donc choisi le logiciel TortoiseSVN. Les raisons du choix de ce client SVN (SubVersion) s'explique tout d'abord par son utilisation sous licence GPL (Licence Publique Générale). Grâce à sa gestion de versions décentralisée (existence de plusieurs dépôts), il permet aussi à chacun de travailler à son rythme, de façon désynchronisée des autres, puis d'offrir un moyen d'échanger nos travaux respectifs. En s'intégrant dans l'explorateur de Windows, il offre, de plus, une interface graphique permettant de réaliser la plupart des tâches de gestion de source SVN en ligne de commande (Figure 9). En outre, il gère plusieurs protocoles réseaux (http, https, svn, ssh,...) par rapport à d'autres concurrents. Autre avantage par rapport aux autres clients, c'est un des seuls logiciel multilingues (plus d'une quarantaine dont le français). Enfin, grâce à sa facilité d'installation et d'utilisation toutes personnes peuvent le prendre rapidement en main.

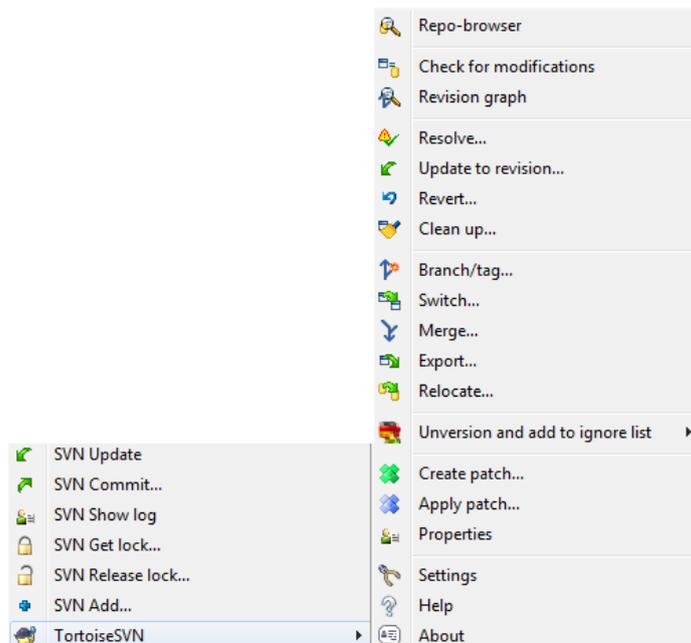


Figure 9 - Exemples de menu contextuel proposé par TortoiseSVN

Après avoir choisi TortoiseSVN, j'ai ensuite mené une étude sur les solutions d'hébergement SVN. Effectivement, pour utiliser un client SVN, on a besoin d'un hébergement pour les dépôts de nos sources. A partir du site <http://www.svnhostingcomparison.com/>, j'ai donc pu comparer plusieurs solutions référencées (Figure 10). Grâce à une liste de critères proposés (coût, nombre de développeurs, protocole, capacité de stockage,...), j'ai pu sélectionner des hébergeurs par rapport à nos besoins.

**Subversion (SVN) Hosting Comparison**

Monthly Cost:  any  free only  \$10 or less  \$11-50  \$50+

# of Modules:  any  1  2-9  10+  unlimited

# of Developers:  any  1-5  6-20  21+  unlimited

Disk Quota:  any  0-100 MB  101-1000 MB  1001+ MB  unlimited

Support:  yes  no

SSL:  yes  no

Issue Mgmt:  any  no

WebSVN:  any  no

Trac:  any  no

Extras:  Web Hosting  Project Mgmt  Blog

OSS Only:  any  yes  no

Name	Monthly Cost (\$)	# of Modules	# of Developers	Disk Quota (MB)	Support	SSL	Issue Mgmt	WebSVN	Trac	Extras	OSS Only
Google	0	Infinity	Infinity	Infinity	None	Yes	Custom	Similar	No	Project Mgmt, Web Hosting, Post Commit Hooks	Yes
Berlios_Developer	0	Infinity	Infinity	Infinity	Email	Yes	Custom	No	No	Project Mgmt	Yes
Gna	0	Infinity	Infinity	Infinity	Email	Yes	Custom	No	No	Project Mgmt	Yes
SourceForge	0	Infinity	Infinity	Infinity	Email	Yes	Custom	No	No	Project Mgmt	Yes
Bounty_Source	0	Infinity	Infinity	Infinity	Email	Yes	Custom	No	No	Project Mgmt	Yes
CloudForge 30 Day Free Trial	0	Infinity	Infinity	2048	Community	Yes	Bugzilla	Yes	Yes	TeamForge Project, Lots of Integration	No
Assembla	0	Infinity	Infinity	1000	Email	Yes	None	Similar	No	Project Mgmt	No
XP-Dev	0	2	Infinity	200	Email	No	Custom	Similar	Yes	Lots of Integration	No

**Figure 10 - Site comparatif d'hébergement « Subversion (SVN) Hosting Comparison »**

Finalement, j'ai décidé de proposer l'hébergeur XP-Dev (<https://xp-dev.com/>). La motivation de ce choix s'explique d'abord à cause des contraintes fixées en amont. Ce site d'hébergement présente effectivement une offre de base gratuite suffisante pour nos besoins, une possibilité d'ajouter un nombre illimité de membres dans chaque projet créé. Mais, il offre surtout une capacité à créer des projets privés (à la différence de l'offre de Google) et donc de conserver la confidentialité des sources. C'est ce dernier point qui fait que j'ai pu convaincre mon équipe de choisir ce site.

Cette étape terminée, je me suis chargé de mettre en place les outils sur nos postes. Effectivement, pour l'écriture des scripts, je me suis appuyé sur l'éditeur de source Notepad++ et Windows PowerShell ISE. Notepad++ est un éditeur de code source convivial qui prend en charge plusieurs langages. Fondé sur la programmation orientée objet, PowerShell lui est un environnement d'écriture de scripts intégré (ISE) dans lequel on peut exécuter des commandes, écrire, tester et déboguer des scripts via une interface utilisateur graphique Windows (Figure 11). Powershell est le principal langage de développement que j'ai décidé d'utiliser. L'extension de ses fichiers est *ps1*.

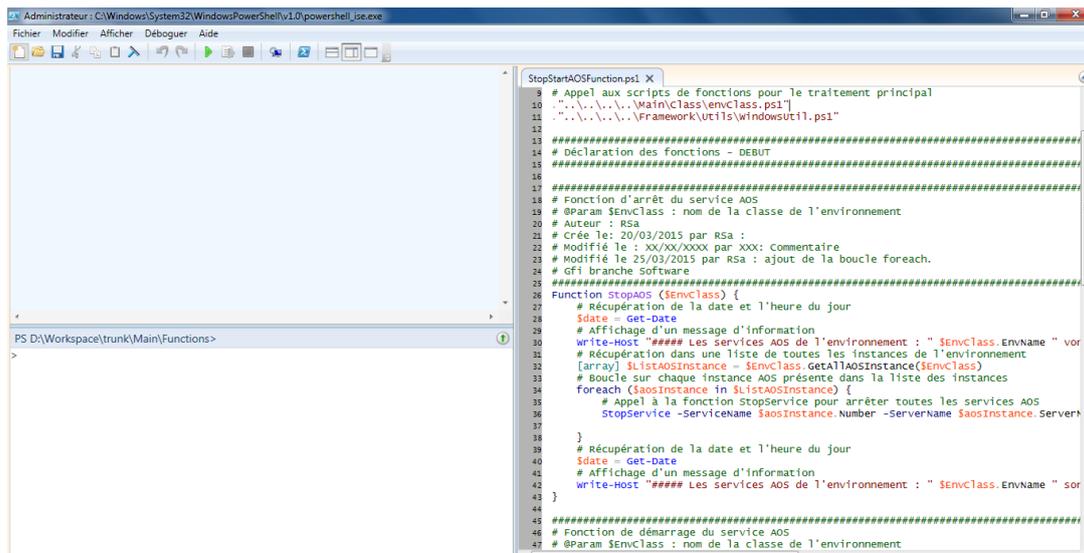


Figure 11 - Interface de Windows PowerShell ISE

La raison du choix de ce langage pour développer les scripts s'explique sur le fait que Microsoft a développé des commandes références avec PowerShell pour la gestion de ses produits et notamment de son ERP Dynamics AX. En effet, il existe des commandes prédéfinies qui permettent par exemple l'importation des modèles (*Install-AXModel*), celle des magasins de modèles (*Install-AXModelStore*), l'exportation des modèles (*Export-AXModel*), celle des magasins de modèles (*Export-AXModelStore*) ou encore la publication des reporting (*Publish-AXReport*). Ainsi, il est plus facile pour nous de construire des scripts autour de ces commandes fondamentales dans l'utilisation quotidienne de l'exploitation de Dynamics AX. Enfin, par rapport aux autres langages ou procédés pour l'écriture de scripts, PowerShell est orienté POO (Programmation Orientée Objet) qui lui confère un avantage non négligeable. Effectivement, la création de scripts en objet nous permet de maîtriser la complexité de la gestion du développement. L'environnement technique mis en place par mes soins, je suis passé à l'étape suivante qui est la revue de l'organisation d'une livraison dans un environnement Dynamics AX.

### 8.3. La revue de l'organisation d'une livraison

Avant mon intervention, l'organisation d'une livraison s'articulait de la manière suivante (Figure 12) :

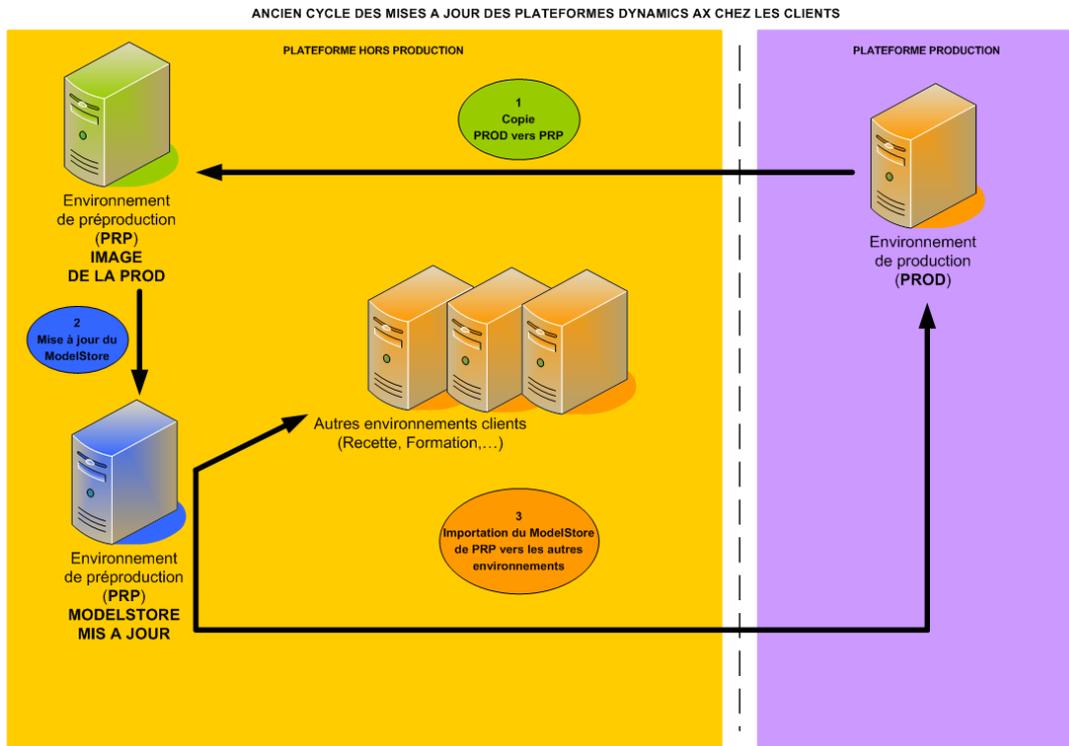


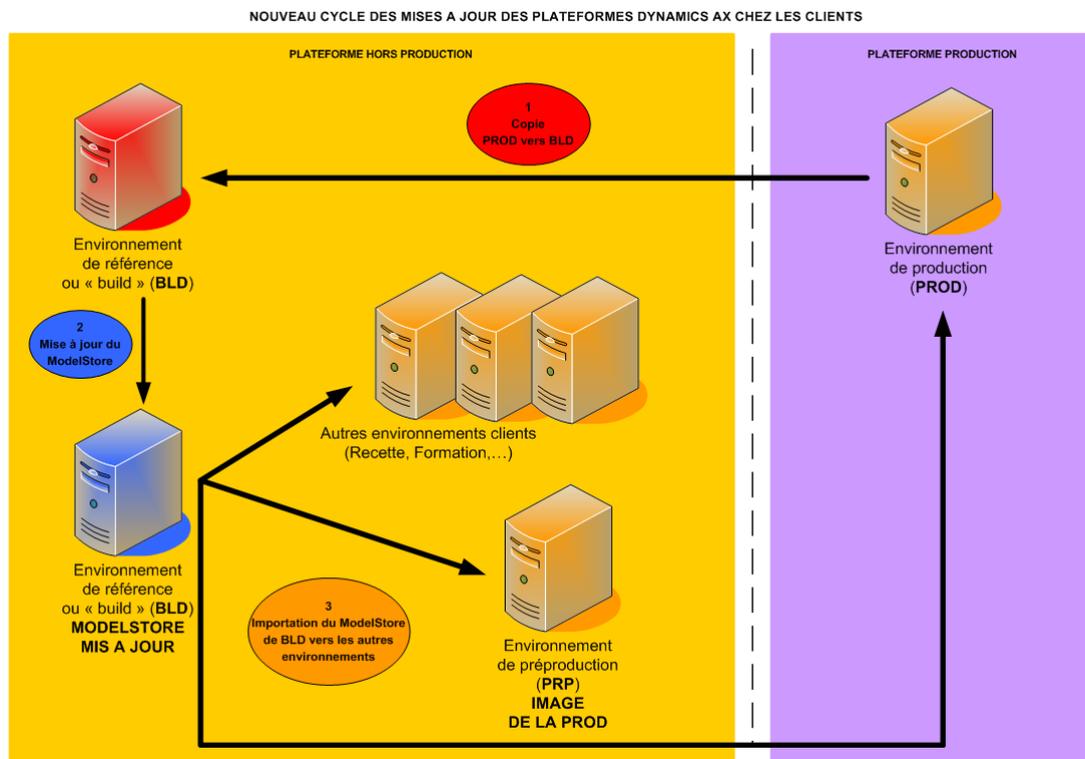
Figure 12 - Schéma de l'ancien cycle de mise à jour des environnements AX

- Etape 1 : Copier l'environnement de production vers l'environnement de préproduction (PRP) ;
- Etape 2 : Mettre à jour PRP :
  - ✓ Importer les fichiers de livraisons Gfi-SSP ;
  - ✓ Compiler et synchroniser l'environnement PRP ;
  - ✓ Exporter le magasin de modèle de l'environnement PRP ;
- Etape 3 : Importer le magasin de modèle de PRP dans tous les environnements.

Le premier problème de ce cycle réside dans la longue indisponibilité de l'environnement de préproduction en cours de mise à jour. Pour pallier ce problème, les opérations sont menées de nuit. Après des échanges avec mon tuteur, j'ai donc émis l'idée qu'il était nécessaire de mettre en place un nouvel environnement (nommé BLD pour

« Build ») chez nos clients afin de ne plus bloquer les autres environnements. Cet environnement spécialement dédié à notre équipe infrastructure technique, nous permet désormais de lancer les opérations de livraison sans bloquer un environnement utilisé par les équipes projet. La nouvelle organisation d'une livraison avec ce nouvel environnement permet alors d'optimiser les étapes qui deviennent (Figure 13) :

- Etape 1 : Copier l'environnement PROD vers BLD ;
- Etape 2 : Mettre à jour BLD :
  - ✓ Importer les fichiers de livraisons Gfi-SSP ;
  - ✓ Compiler et synchroniser l'environnement BLD ;
  - ✓ Exporter le magasin de modèle de l'environnement BLD ;
- Etape 3 : Importer le magasin de modèle de BLD dans tous les environnements.



**Figure 13 - Schéma du nouveau cycle de mise à jour des environnements AX**

Ainsi, l'environnement BLD dans le processus de mise à jour joue un rôle d'intermédiaire qui résout les difficultés soulevées précédemment. Les avantages de cette solution sont entre autres que :

- Les opérations peuvent être menées avec un temps d'indisponibilité beaucoup plus réduit ;
- Les opérations de mise à jour sont centralisées sur l'environnement BLD ;
- Les opérations de « post-update » (compilation, synchronisation dans Dynamics AX) sont exécutées principalement sur cet environnement. Le temps d'indisponibilité pour maintenance sur les autres environnements est réduit considérablement (d'environ six heures, on passe à une heure).
- En cas d'erreur, l'impact est minimisé et permet de donner du temps aux développeurs pour corriger.

Initialement, le responsable de la réalisation de toutes les tâches de mise à jour chez un client était mon collègue et tuteur. Maintenant, avec mon arrivée, je suis devenu aussi responsable de réaliser ces travaux en fonction des priorités des livraisons prévues par le CICEO. Le travail est ensuite réparti par mon tuteur pour savoir qui s'occupera de telles livraisons pour tels clients. Après avoir fixé les bases de cette nouvelle organisation et créé un nouvel environnement de « build », il était nécessaire de retravailler les scripts servant à cette opération.

#### 8.4. L'étude de l'organisation des scripts

En partant de l'existant chez les clients, j'ai donc d'abord étudié la structure générale des dossiers de livraisons. Il s'organisait alors de la manière suivante :

- Un dossier *copyenv* : contenant les scripts de copie d'environnement avec :
  - ✓ Un répertoire *COMMON* contenant trois scripts fonctions pour la copie d'environnement (Figure 14) :

Nom	Modifié le	Type	Taille
 BackupRestore.ps1	15/02/2014 00:33	Fichier PS1	18 Ko
 CopySourceDest.ps1	25/05/2014 17:42	Fichier PS1	27 Ko
 ModifyDatabaseSecurity.ps1	05/05/2014 12:46	Fichier PS1	9 Ko

**Figure 14 - Contenu du sous-dossier *COMMON***

- ✓ Un dossier au nom d'environnements (ex : RCT pour environnement de Recette, FRM pour environnement de Formation, PRP pour environnement de Préproduction,...) contenant des fichiers et dossiers dédiés à l'environnement (Figure 15) :

Nom	Modifié le	Type	Taille
BackupTemp	03/06/2014 11:11	Dossier de fichiers	
bak	03/06/2014 11:12	Dossier de fichiers	
ExportParams	03/06/2014 11:12	Dossier de fichiers	
logs	03/06/2014 11:12	Dossier de fichiers	
CopyDBRctForm.ps1	05/06/2014 11:45	Fichier PS1	21 Ko
Export_Params.ps1	23/05/2014 16:17	Fichier PS1	4 Ko

**Figure 15 - Contenu du dossier FRM-RCT**

Le fichier *CopyDBRctForm.ps1* (un descriptif complet du script est en Annexe : Figure 41) est le script exécuté pour la copie de l'environnement RCT (Recette) vers celui de FRM (Formation). Il exécute les scripts fonctions du répertoire COMMON (*BackupRestore.ps1*, *CopySourceDest.ps1*,...).

- Un dossier par nom d'environnement où selon la Figure 16, les scripts sont classés par thématique (exploitation et livraison) :

Nom	Modifié le	Type	Taille
2 - Exploitation	06/08/2014 23:13	Dossier de fichiers	
Livraison	08/08/2014 12:00	Dossier de fichiers	
ReportAxmodel.ps1	06/08/2014 16:14	Fichier PS1	1 Ko

**Figure 16 - Contenu du dossier FRM**

- ✓ Le dossier *2-Exploitation* contient les scripts des opérations autour de l'exploitation des environnements (Figure 17). Basé sur des commandes natives PowerShell, ce sont des fichiers fonctions exécutant des opérations du cycle de mise à jour (importation, exportation,...).

Nom	Modifié le	Type	Taille
0-ExportModelStore.ps1	06/08/2014 22:45	Fichier PS1	5 Ko
1-ExportModel.ps1	05/06/2014 09:51	Fichier PS1	5 Ko
1-ImportModel.ps1	05/06/2014 09:51	Fichier PS1	7 Ko
2-ImportModelStore.ps1	08/08/2014 00:18	Fichier PS1	12 Ko
2-ImportModelStore.ps1.log	06/08/2014 23:42	Document texte	1 Ko
3-UninstallLayer.ps1	05/06/2014 09:51	Fichier PS1	7 Ko
3-UninstallModel.ps1	05/06/2014 09:51	Fichier PS1	8 Ko
4-start_stopAOS.ps1	05/06/2014 09:52	Fichier PS1	14 Ko
5-CompileAX.ps1	05/06/2014 09:51	Fichier PS1	21 Ko
6-Synchronize.ps1	05/06/2014 09:51	Fichier PS1	18 Ko
7-CleanBackupModelStore.ps1	14/02/2014 14:59	Fichier PS1	4 Ko
8-PublishEP.ps1	14/02/2014 14:59	Fichier PS1	5 Ko
9-PublishReports.ps1	05/06/2014 09:50	Fichier PS1	4 Ko
10-BackupModels.ps1	06/08/2014 16:16	Fichier PS1	3 Ko
10-SSRSReports.ps1	05/06/2014 09:50	Fichier PS1	8 Ko
10-SSRSReports2.ps1	14/02/2014 14:59	Fichier PS1	9 Ko
11-PublishBI.ps1	14/02/2014 14:59	Fichier PS1	5 Ko
12-SET-DatasourceSSAS.ps1	05/06/2014 09:50	Fichier PS1	1 Ko
13-UserInfoWML.ps1	05/06/2014 13:33	Fichier PS1	2 Ko

**Figure 17 - Contenu du dossier 2-Exploitation**

- ✓ Le dossier *Livraison* (Figure 18) contient les fichiers liés aux opérations après la mise à jour des environnements telles que les compilations, synchronisation ou encore le déploiement des reports.

Nom	Modifié le	Type	Taille
1-ax_synchro_FRM.log	27/08/2014 03:59	Document texte	1 Ko
2-ax_compilecil_FRM.log	27/08/2014 03:37	Document texte	5 Ko
0-ax_compile_FRM.log	27/08/2014 03:14	Document texte	70 Ko
5-SSRSReports.ps1FRM.log	08/08/2014 12:41	Document texte	1 672 Ko
5-SSRSReports.ps1	08/08/2014 12:00	Fichier PS1	9 Ko
0-Compile.cmd	07/08/2014 11:55	Script de comman...	1 Ko
2-CompileCIL.cmd	17/06/2014 20:14	Script de comman...	1 Ko
1-Synchronize.cmd	17/06/2014 20:14	Script de comman...	1 Ko
FIN.log	17/06/2014 20:04	Document texte	1 Ko
END.cmd	17/06/2014 20:04	Script de comman...	1 Ko
PostUpdate_ADEME_FRM.xml	17/06/2014 19:54	Document XML	4 Ko
2-CompileCIL.xml	17/06/2014 19:24	Document XML	1 Ko
1-Synchronize.xml	17/06/2014 19:23	Document XML	1 Ko
0-Compile.xml	17/06/2014 19:23	Document XML	1 Ko
2-ax_compilecil_test.log	08/06/2014 10:58	Document texte	4 Ko
3-UpdateGFILabels_TEST.log	16/03/2014 00:39	Document texte	1 547 Ko
4-UpdateDataJobs_TEST.log	15/03/2014 23:58	Document texte	2 Ko
4-UpdateDataJobs.xml	15/03/2014 23:58	Document XML	1 Ko
4-UpdateDataJobs.bat	15/03/2014 23:56	Fichier de comma...	1 Ko
3-UpdateGFILabels.xml	14/03/2014 18:49	Document XML	1 Ko
3-UpdateGFILabels.bat	14/03/2014 18:41	Fichier de comma...	1 Ko
OldPostUpdate_FOCH_TEST.xml	14/03/2014 18:10	Document XML	4 Ko

**Figure 18 - Contenu du dossier Livraison**

Concernant cette organisation des répertoires, j'ai ensuite noté plusieurs remarques. Cette vision pensée par environnement atteint rapidement ses limites. En effet, on se trouve

face à une duplication des scripts et donc un gros problème de maintenance sur ces fichiers. De plus, il manque une notion de visibilité dans l'enchaînement des scripts. On ne connaît pas l'ordre dans lequel il faut exécuter ces fichiers. Enfin, les scripts sont tous mis à un même niveau hiérarchique. Cette étude montre donc qu'il était nécessaire de réfléchir sur une nouvelle structure afin d'apporter de la facilité dans l'utilisation des scripts et de la simplicité dans leur maintenance.

## **8.5. La nouvelle structure adoptée**

Une première version (V1.0) d'un package pour la mise à jour des environnements a ainsi été réalisée à partir d'étapes successives de réflexion et de développement. En effet, j'ai d'abord pensé qu'il était plus judicieux de raisonner non plus en termes d'environnement installé chez le client mais plutôt par processus utilisés pour la mise à jour des environnements. Ce premier niveau de raisonnement permet alors de supprimer la redondance de fichiers et de montrer clairement notre maîtrise des opérations. Ensuite, il m'est apparu évident qu'il fallait séparer les scripts fonctions des scripts utilisés pour l'exécution des traitements d'une livraison. Ce second niveau permet la réutilisation des processus développés. Un troisième niveau concerne la création de fichiers principaux qui centralise uniquement l'enchaînement des opérations. Enfin, un quatrième niveau concerne la modélisation orientée objet de ces processus. Cette dernière ouvre la perspective de réduire la complexité de l'écriture des scripts et de renforcer la vision processus. Les fichiers font désormais appels à des classes d'objets permettant de modéliser un environnement Dynamics AX.

### *8.5.1. Conception d'un fichier de configuration client*

Concrètement, j'ai d'abord recherché et placé pour chaque client toutes les informations contextuelles des environnements Dynamics AX dans un fichier XML (Figure 19). Ce fichier nommé *config.xml* (*Main\Ressources*) est pour le moment généré manuellement. Il se base sur les informations récupérées à partir des bases de données en exécutant un script qui existait déjà : *Export\_Params.ps1*.

```

<CONFIG custName="" storageScripts="">
  <Environnements name="HPR">
    <Environnement name="REC">
      <Components>
        <ClusterAIF clusterName="aifREC">
        <ClusterAOS clusterName="aosREC">
          <Instance
            account=""
            batch=""
            number=""
            port=""
            instance=""
            isAOSMaint=""
            name=""
            serverName="" />
        </ClusterAOS>
        <ClusterDMF clusterName="dmfREC">
        <ClusterHELP clusterName="helpREC">
        <ClusterSHAREPOINT clusterName="sharepointREC">
        <ClusterSMTP clusterName="smtpREC">
        <ClusterSGBD clusterName="sgbdREC">
          <Instance
            account=""
            name=""
            port=""
            serverName=""
            SQLInstance="">
        </ClusterSGBD>
        <ClusterSSAS clusterName="ssasREC">
        <ClusterSSRS clusterName="ssrsREC">
      </Components>
      <Services>
    </Environnement>
    <Environnement name="TEST">
    <Environnement name="PRP">
  </Environnements>
</CONFIG>

```

Figure 19 - Extrait du config.xml standard

### 8.5.1. Modélisation d'un environnement Dynamics AX

Afin de gérer la complexité de la récupération des informations du config.xml dans les scripts, j'ai commencé à modéliser dans un diagramme de classe un environnement Dynamics AX (Figure 20). Ainsi, j'ai créé une classe environnement qui se compose elle-même d'un ou plusieurs clusters (AOSCLUSTER, SSRSCLUSTER,...) qui hérite des caractéristiques de la classe CLUSTER. Un cluster se compose d'un ou plusieurs mêmes composants (AOSINSTANCE pour le cluster AOS, SSRSINSTANCE pour le cluster SSRS,...) qui hérite des caractéristiques de la classe COMPOSANT.

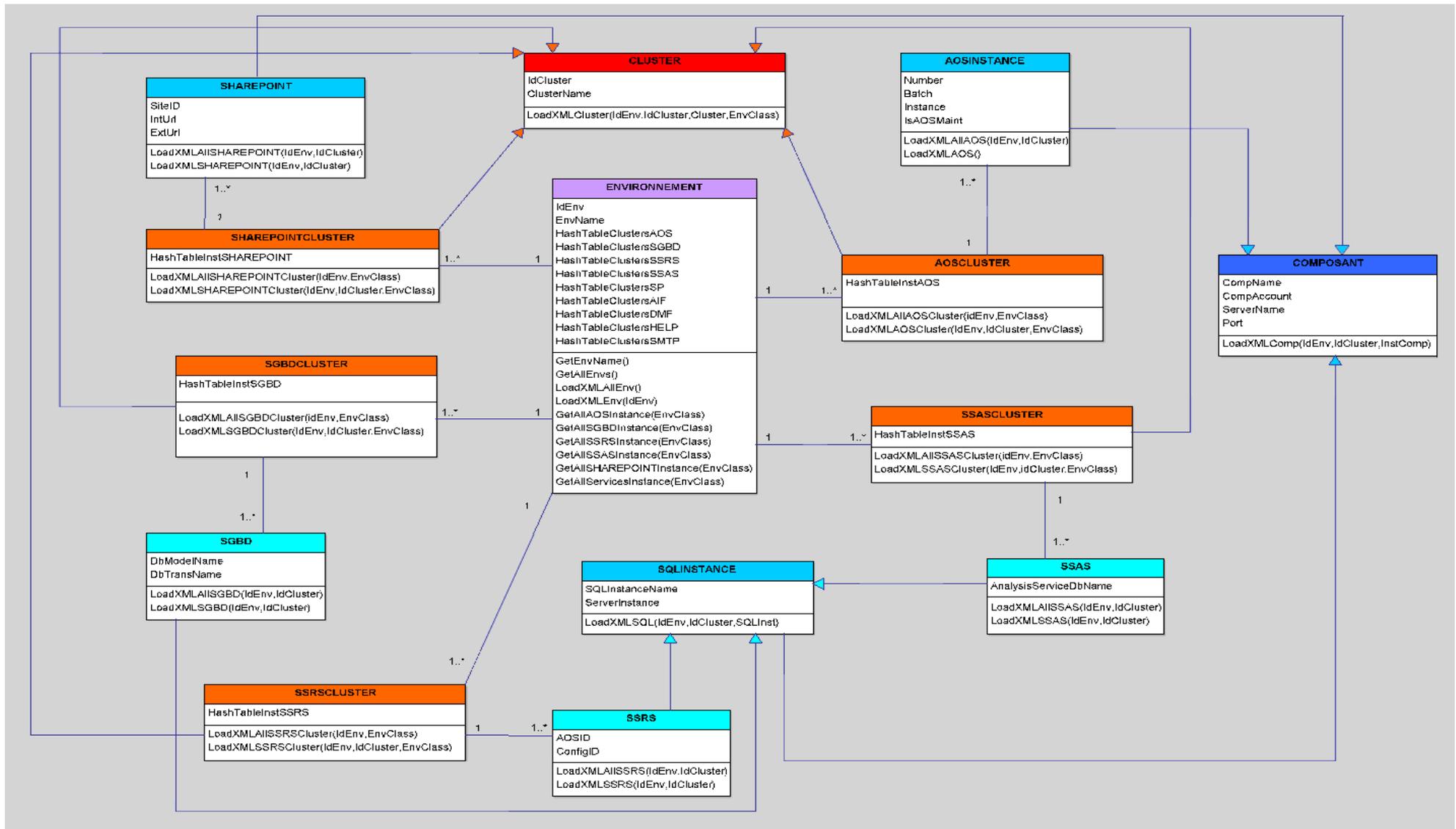
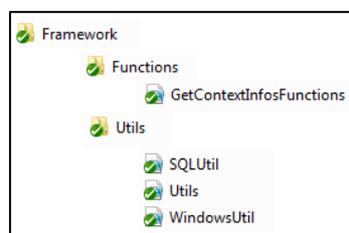


Figure 20 - Extrait du diagramme de classe modélisant l'objet environnement

Grâce à des méthodes spécifiques (exemples : *LoadXMLAOS()* pour une instance AOS, *LoadXMLAllAOS()* pour toutes les instances AOS, *LoadXMLAOSCluster()* pour un cluster AOS, *LoadXMLAllAOSCluster()* pour tous les cluster AOS d'un environnement, *LoadXMLEnv()* pour toutes les informations d'un environnement, *LoadXMLAllEnv()* pour tous les environnements d'un client,...), je charge tout d'abord les informations issues du fichier XML config.xml. Ensuite, grâce à des méthodes de la classe environnement (exemples : *GetAllAOSInstance()* pour les instances AOS d'un environnement), je récupère les valeurs des propriétés des classes. En conclusion, cela me permet de m'affranchir de la structure du config.xml afin de ne raisonner qu'en termes de composants.

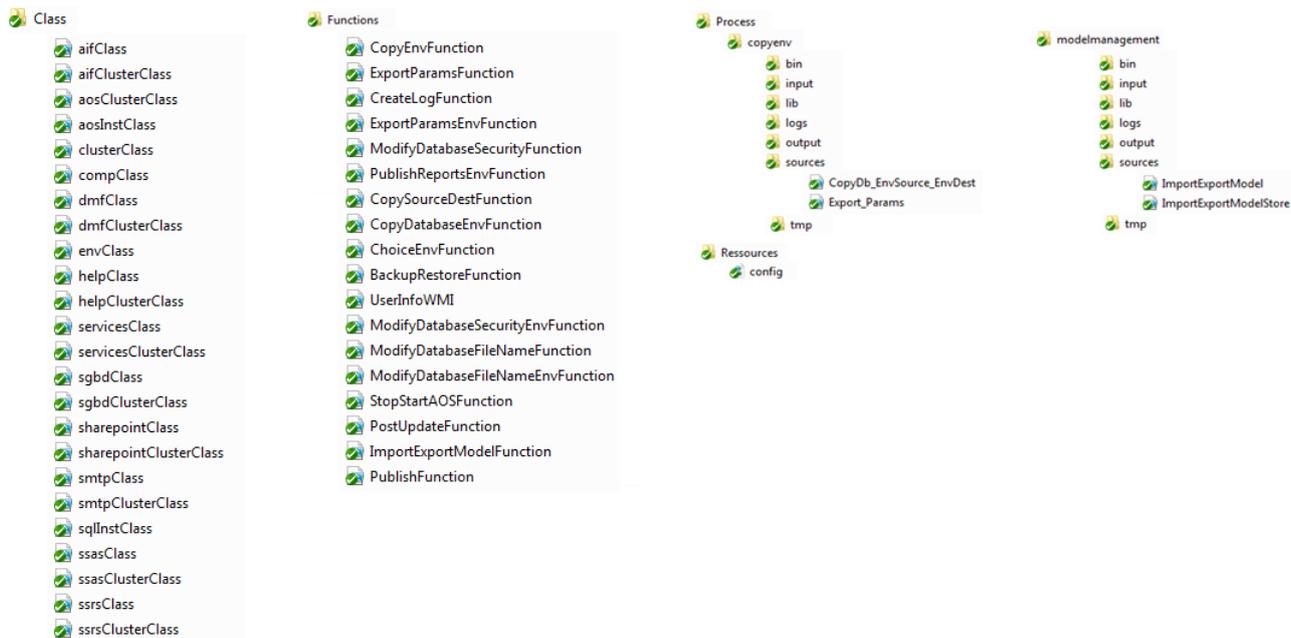
#### 8.5.1. Conception de l'arborescence des répertoires

Après ce travail de modélisation, je me suis occupé à créer un premier répertoire racine nommé *KitGfiEnv* dans lequel il existe maintenant deux dossiers : l'un nommé *Framework* et l'autre *Main*. Selon la Figure 21, le répertoire *Framework* se compose de fonctions génériques pour l'environnement Windows (dossier *Utils*) mais aussi celles qui permettent de récupérer les informations contextuelles des environnements clients (dossier *Functions*). Le but de ce *Framework* est de pouvoir ensuite réutiliser ces fonctions dans d'autres contextes.



**Figure 21 - Détails du répertoire *Framework***

Le dossier *Main* (Figure 22) regroupe lui les classes (dossier *Class*), les fonctions communes aux scripts (dossier *Function*), les ressources spécifiques (dossier *Ressources*) et les processus d'exploitation (dossier *Process*).



**Figure 22 - Détails du dossier *Main***

Pour une meilleure lisibilité, les fichiers de fonctions portent tous désormais dans leur nom la mention de « *Function* » et ils sont répartis dans le dossier *Main\Functions* pour bien les séparer des fichiers d'exécution. De cette façon, il est possible de les réutiliser plus aisément dans les scripts de traitement. De plus, un travail assidu a été mené par mes soins dans le « refactoring » du code. En effet, j'ai retravaillé le code des fonctions de façon à en améliorer la lisibilité, le rendre plus générique et par conséquent améliorer la maintenance du code. Enfin, j'ai créé la possibilité de générer automatiquement pour chaque traitement un fichier log qui reprend le nom, la date et l'heure du traitement. Les logs sont d'ailleurs classés par nom d'environnement dans un dossier « *Logs* » par type de processus.

Dans le répertoire *Process*, pour chaque processus (*copyenv*, *modelmanagement*,...), on retrouve alors une arborescence similaire :

- *Bin* : binaires qui résultent d'une compilation de nos scripts ;
- *Input* : fichiers en entrée du script (exemple : fichiers *.axmodel* à importer) ;
- *Logs* : résultats logs d'exécution des scripts lié au process courant ;
- *Output* : résultats d'exécution (exemple : fichiers *.modelstore* à exporter) ;
- *Sources* : fichiers source (ps1 ou cmd,...) ;

- *Tmp* : fichiers temporaires générés lors de l'exécution du script et qui peuvent être ensuite supprimés sans perte de données (exemple : fichiers *.bak* créés,...).

Ce découpage s'inspire de la définition d'un processus qui est un ensemble d'activités corrélées qui transforme des éléments d'entrée (dossier *Input*) en éléments de sortie (dossier *Output*).

Ainsi, l'écriture des scripts s'est simplifiée en devenant que de simples enchaînements d'actions sur les composants Dynamics AX. Cette nouvelle arborescence ne s'est pas fait évidemment du premier coup. Elle résulte d'une démarche itérative dans laquelle j'ai souvent échangé avec mon tuteur. Enfin, ce travail a permis d'avoir des résultats directs sur une partie du travail de l'équipe avec :

- Réduction des tâches de maintenance sur la gestion des scripts ;
- Utilisation facilitée des scripts.

Je vais maintenant voir plus en détails les deux principaux processus retenus à travers cette nouvelle organisation.

## **8.6. Le nouveau découpage en processus des scripts principaux**

Le nouveau découpage en processus des scripts se compose à l'heure actuelle de la copie d'environnement (*Main\Process\copyenv*) et de la gestion de la base modèle (*Main\Process\modelmanagement*). Ce sont les deux opérations que je vais aborder tout de suite.

### *8.6.1. Conception du script de copie d'environnement*

La copie d'environnement est l'étape qui consiste à récupérer les données d'un environnement source vers un environnement cible. Cette opération s'exécute en lançant le fichier *CopyDb\_EnvSource\_EnvDest.ps1* (*Main\Process\copyenv\sources*). A l'origine, c'est un fichier qui existait dans lequel on trouvait les variables en dur ainsi que des fonctions. En m'appuyant sur cet existant, j'ai totalement modifié sa structure en raisonnant en termes de suites d'opérations sur les composants (Figure 23) :

```

# Etape 0 - Choix des environnements Source et Destination
# Env source
$EnvSource = NewEnvClass
$EnvSource = ChoiceEnvSource

# Env de destination
$EnvDestination = NewEnvClass
$EnvDestination = ChoiceEnvDest

# Etape 1 - Arrêt des services AOS
StopAOS -EnvClass $EnvDestination *>(CreateLog -EnvSource $EnvSource -EnvDest $EnvDestination -LogName "StopAOS" -Process "copyenv")

# Etape 2 - Copie de la base des modèles de l'environnement Source (S) vers l'environnement Destination (D)
CopyModelDatabase -EnvClassSource $EnvSource -EnvClassDest $EnvDestination *>(CreateLog -EnvSource $EnvSource -EnvDest $EnvDestination

# Etape 3 - Copie de la base transactionnelle de l'environnement Source (S) vers l'environnement Destination (D)
CopyTransDatabase -EnvClassSource $EnvSource -EnvClassDest $EnvDestination *>(CreateLog -EnvSource $EnvSource -EnvDest $EnvDestination

# Etape 4 - Modification de la sécurité après la restauration de l'environnement Destination (D)
ModifyDatabaseSecurityEnv -EnvClass $EnvDestination *>(CreateLog -EnvSource $EnvSource -EnvDest $EnvDestination -LogName "ModifyDatabas
...
# Etape 5 - Mise à jour d'informations de paramétrages de la base transac de l'environnement Destination (D)
CopyEnvSourceEnvDest -EnvClassSource $EnvSource -EnvClassDest $EnvDestination *>(CreateLog -EnvSource $EnvSource -EnvDest $EnvDestinati

# Etape 6 - Modification des noms logiques des fichiers des bases de données de l'environnement Destination
ModifyDatabaseFileNameEnv -EnvClassSource $EnvSource -EnvClassDest $EnvDestination *>(CreateLog -EnvSource $EnvSource -EnvDest $EnvDest

# Etape 7 - Démarrage des services AOS
StartAOS -EnvClass $EnvDestination *>(CreateLog -EnvSource $EnvSource -EnvDest $EnvDestination -LogName "StartAOS" -Process "copyenv")

# Etape 8 - Déploiement des reports sur l'environnement de Destination
PublishAllReportsEnv($EnvDestination) *>(CreateLog -EnvSource $EnvSource -EnvDest $EnvDestination -LogName "PublishReport")

```

### Figure 23 - Extrait de l'enchaînement actuel des opérations de copie d'environnement

En plus de l'adoption des principes déjà cités, les principales améliorations apportées dans ce script par mes soins sont notamment :

- Choix des environnements source et destination ;
- Mise en place de commentaires afin d'améliorer la lisibilité et de décrire les différentes étapes effectuées par le script ;
- Mise en place d'invite question-réponse afin de rendre l'utilisation du script plus intuitive ;
- Les opérations d'arrêt et de redémarrage du service AOS s'effectue maintenant automatiquement à distance.

Ainsi, l'opération de copie d'environnement est désormais plus claire et plus rapide à exécuter. A présent, je vais présenter les modifications apportées aux scripts de gestion des modèles.

### 8.6.2. Conception de scripts pour la gestion de la base *modelstore*

Après la copie d'environnement, la gestion de la base de données *modelstore* est la seconde grande étape du cycle de mise à jour des environnements. Elle se divise en deux opérations : dans un premier temps, la livraison des fichiers Gfi-SSP dans l'environnement de référence (BLD), puis dans un second temps, l'importation du *modelstore* de BLD vers les autres environnements.

La livraison des fichiers Gfi-SSP s'effectue en lançant le fichier que j'ai nommé *ImportExportModel.ps1* (*Main\Process\modelmanagement\sources*). Initialement, il existait plusieurs fichiers qui contenaient ensemble des fonctions et des traitements. Pour effectuer cette étape, il fallait donc lancer à la suite plusieurs fichiers. En mettant les fonctions dans des scripts séparés, j'ai donc construit un unique fichier avec les améliorations suivantes :

- Choix de l'environnement dans lequel importé les fichiers Gfi-SSP ;
- Mise en place de commentaires afin d'améliorer la lisibilité et de décrire les différentes étapes effectuées par le script ;
- Mise en place d'invite question-réponse afin de rendre l'utilisation du script plus intuitive ;
- Un menu et sous-menu ont été ajoutés afin de permettre de lancer soit en une fois l'ensemble des traitements soit un traitement en particulier ;
- Les grandes opérations automatisées de ce processus sont maintenant dans l'ordre :
  - ✓ L'arrêt du service AOS à distance ;
  - ✓ La sauvegarde complète des bases de l'environnement ;
  - ✓ L'import des fichiers modèles en sélectionnant par invite les fichiers concernés (*.axmodel*) stockés dans le dossier *Common\Livrables* ;
  - ✓ Le redémarrage du service AOS à distance ;
  - ✓ Le lancement à distance de la tâche de planification nommé *PostUpdate\_NomClient\_NomEnv* qui effectue les traitements suivants sur l'application après import du fichier modèle : la compilation, la compilation CIL et la synchronisation ;
  - ✓ L'export du magasin de modèle de l'environnement dans un fichier (*.axmodelstore*).

Par conséquent, toutes ces améliorations permettent maintenant de mieux industrialiser cette étape récurrente. En effet, l'automatisation des opérations offrent la possibilité de ne plus monopoliser une ressource. Enfin, à présent, ce fichier permet d'avoir une compréhension très claire des traitements effectués dans le fichier qui permet à toute personne de pouvoir exécuter l'opération d'import des fichiers modèles.

Le deuxième script important concerne l'import/export du magasin de modèle de BLD vers tous les autres environnements du client. Elle s'effectue en exécuter le script *ImportExportModelStore.ps1* (*Main\Process\modelmanagement\sources*). Comme pour le fichier précédent, il existait les mêmes défauts. J'ai alors construit un nouveau fichier avec les mêmes avantages que le fichier *ImportExportModel.ps1*. Pour ce fichier, les grandes opérations automatisées sont maintenant dans l'ordre :

- La sauvegarde complète des bases de l'environnement ;
- La sauvegarde du fichier log de la base des modèles ;
- La réduction de la taille de la base des modèles et du fichier log ;
- L'import du magasin de modèle en sélectionnant par invite le fichier d'export (*.axmodelstore*) stocké dans *Main\Process\modelmanagement\output* ;
- Le lancement à distance de la tâche de planification nommé *PostUpdate\_NomClient\_NomEnv* qui effectue les traitements suivants sur l'application après import du fichier modèle : la compilation CIL, la synchronisation et la publication des reports sur la base de données SSRS ;

En conclusion, cette nouvelle organisation renforce la volonté de mettre en perspective une vision des scripts par type de processus avec un clair enchaînement d'opération sur les composants AX. Il est en effet intéressant de raisonner ainsi car cela permet de structurer les éléments pas grande opération de maintenance que l'on est amenée à exécuter sur les environnements clients. De plus, cette structure permet même de connaître l'ordre d'enchaînement des étapes de mises à jour d'un environnement. Enfin, cela permet de bien identifier les éléments nécessaires en entrée (dossier *input*) et les résultats en sortie (répertoire *output*) pour chacun des processus. C'est un raisonnement qui montre donc incontestablement la maturité de la maîtrise de nos processus d'industrialisation. Une fois l'écriture des scripts

réalisée, l'étape suivante a été ensuite de déployer en contexte client ces nouveaux scripts afin de les tester.

## **8.7. La mise en œuvre chez les clients**

Les scripts ont été mis en place chez un client dans une plateforme hors production et ensuite pour un autre client dans son environnement en production. Au préalable, un environnement BLD a été donc déployé chez ses clients afin de réduire les temps de maintenance sur les autres environnements de travail.

### *8.7.1. Les résultats en environnement hors production*

Premièrement, j'ai mis en place le nouveau package chez les environnements hors production d'un client. J'ai donc d'abord testé la copie des environnements qui a montré la possibilité de pouvoir dupliquer n'importe quel environnement assez aisément. Ensuite, j'ai mis en exécution les scripts pour le déploiement des livraisons. L'automatisation des opérations m'a permis de pouvoir me concentrer sur d'autres tâches en parallèle. Cependant, en fonction de la version de PowerShell installée chez le client, certaines opérations ont dû être réécrites afin de pouvoir tenir compte du contexte.

La mise en place des scripts a permis d'accélérer et automatiser les tâches des livraisons régulières dans les environnements hors production. Nous sommes ainsi passés à une nouvelle ère dans notre processus d'industrialisation. Le cycle de vie des environnements est sans aucun doute amélioré et permet aux équipes sur place de disposer plus rapidement d'environnements à jour. En outre, ce premier travail a permis de mettre au second plan ce processus de déploiement des livraisons pour nous concentrer sur d'autres tâches.

### *8.7.2. Les résultats en environnement production*

Après la réussite en environnement hors production, il a été question de déployer les scripts dans un environnement client en production. J'ai eu alors l'occasion pendant trois semaines d'intégrer sur un site une équipe projet Gfi-SSP chez un client en phase de démarrage en production. Le premier besoin a été alors de s'occuper rapidement des livraisons critiques dans des temps très courts (tous les trois jours). Ici aussi, l'automatisation

des scripts nous a donc permis de pouvoir réduire le délai de mise en production de ces livraisons urgentes.

Au passage, il a été nécessaire de réfléchir sur une mise en place d'une nouvelle organisation des tâches entre les équipes participantes dans ce processus de livraisons. En effet, il était d'abord nécessaire que les nouvelles corrections apportées par les développeurs soient centralisées dans un seul et même environnement sinon, avec le nouveau cycle que j'avais mis en place, elles seraient automatiquement écrasées lors de chaque mise à jour. De plus, il était évident qu'elles soient testées et validées avant de les intégrer directement en production.

Ainsi, comme le montre la Figure 24, il a été défini lors d'une réunion interne que les anomalies soient remontées par les consultants Gfi auprès des développeurs. Ces derniers apportent les corrections dans un environnement propre aux développements. Une fois les correctifs effectués et testés par les consultants, j'ai proposé lors d'une réunion avec les consultants que les consultants répertorient dans un fichier type tableur les corrections des développeurs et de donner un numéro de version à la livraison critique afin de conserver l'historique des actions menées. Ensuite, les consultants nous livrent par courriel ce fichier contenant la liste des projets à livrer. Puis, c'est à notre structure technique de s'occuper de la livraison en suivant le nouveau cycle de livraison mis en place décrit plus haut (Figure 13). Si l'on rencontre des difficultés, il est alors possible de contacter les développeurs afin de corriger le projet problématique. Une fois la livraison terminée, nous envoyons un courriel à toutes les équipes pour leur informer de la réussite de la mise à jour des environnements du client.

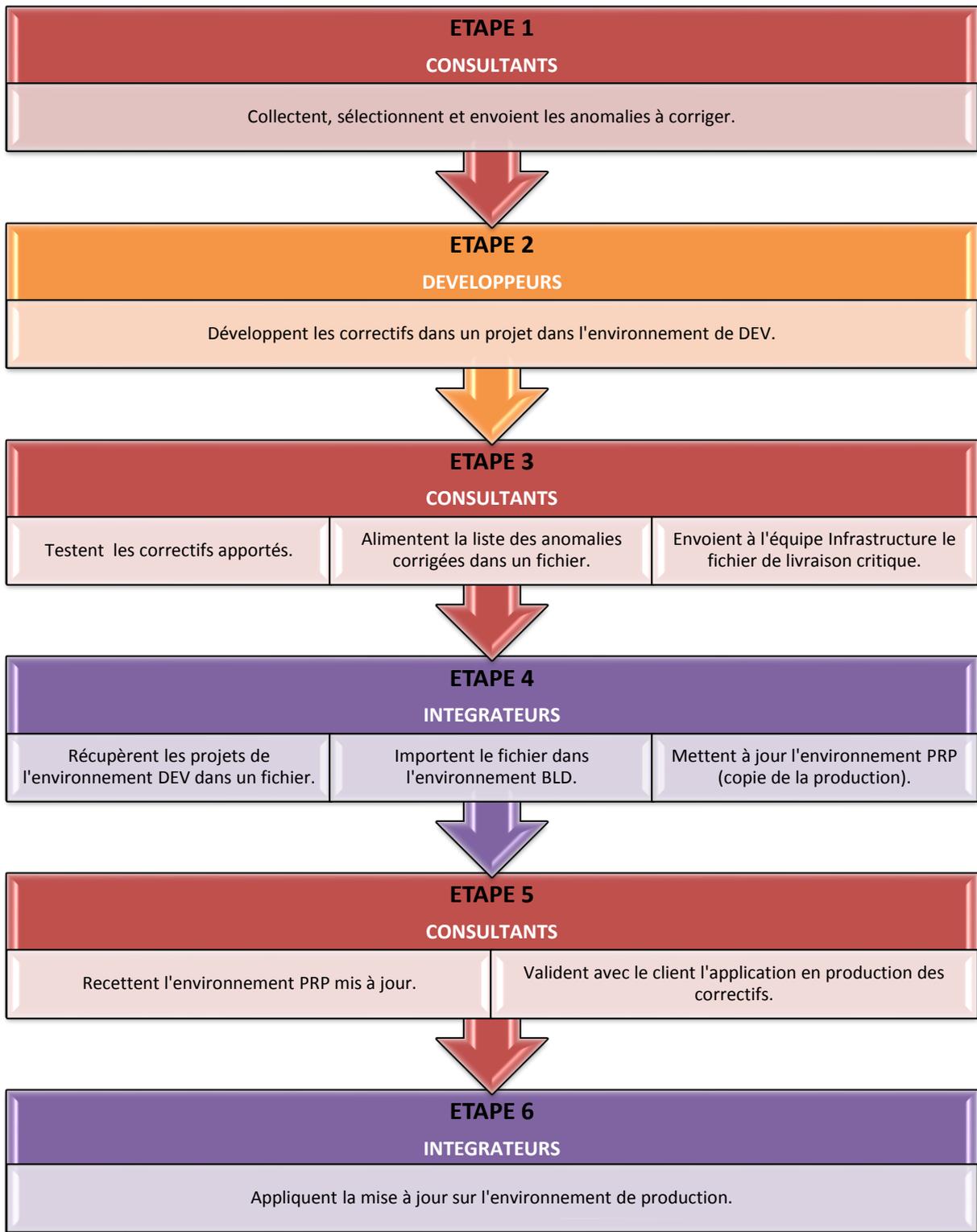


Figure 24 - Organisation des tâches des équipes Gfi-SSP pour les livraisons critiques

Cette répartition des tâches s'est déroulée lors de réunions internes entre les équipes concernées sous la responsabilité du directeur de projet où le travail a été réparti. Concernant, notre équipe, la répartition des tâches qui nous incombent à mon tuteur et moi-même s'effectuent selon les priorités de notre activité. Cette organisation nous a ainsi permis de définir nos périmètres, de pouvoir sécuriser au maximum le processus des livraisons critiques et de devenir surtout très réactif.

En revanche, concernant la disposition d'environnement dupliqué à partir de celui de la production, cette première version des scripts de copie n'a pas pu la réaliser. Effectivement, la copie des environnements se base sur des fichiers de sauvegarde de bases de données. Or, entre les environnements de hors production, il n'existe pas de difficultés particulières car toutes les bases de données SQL Server des environnements installés se basent sur un seul fichier principal (type *.mdf*) et un seul fichier journal (type *.log*). Cependant, pour des questions de performances et d'évolutivité, les bases de données en production sont divisées en plusieurs fichiers principaux (un type *.mdf* et plusieurs fichiers *.ndf*) et plusieurs fichiers journaux (type *.log*). La capacité de stockage mise à notre disposition par le client est différente entre les deux types de plateformes. Il nous est donc impossible de dupliquer automatiquement l'environnement de production vers un environnement hors production. Pour y arriver, il faut que l'on réduise à un moment des opérations, la taille du fichier de sauvegarde de la production de façon manuelle. Dans cette première version, je n'ai donc pas pu encore intégrer cette dimension.

En résumé, pour ce projet, les bénéfices de cette nouvelle structure sont sans aucun doute nombreux :

- Une seule organisation commune des répertoires sur les environnements ;
- Une mise à jour des livraisons Gfi-SSP non bloquante pour la production ;
- La complexité du processus de copie d'environnement mieux maîtrisée ;
- Une mise à niveau facilitée et améliorée de tous les environnements ;
- Une réduction du temps passé à réaliser ces tâches de maintenance ;
- Le travail entre collaborateurs de l'équipe est enfin optimisé.

Evidemment, cette première version (V1.0) est loin d'être parfaite et peut-être encore optimisée. Voici d'ailleurs quelques pistes d'évolution que je propose de résoudre prochainement :

- Mettre en place une meilleure gestion des erreurs dans les scripts ;
- Gérer les erreurs d'importation des fichiers de modèles ;
- Importer un magasin de modèles sur plusieurs environnements de manière simultanée.

Ainsi, ce projet d'industrialisation des mises à jour des environnements a été réalisé dans les délais et les objectifs fixés. Il est cependant loin d'être terminé. En effet, c'est un travail continu. Mais ce premier niveau nous a permis sans aucun doute d'améliorer la qualité de notre processus d'intégration. Je vais passer maintenant au projet suivant qui m'a été confié concernant une autre problématique : la haute disponibilité.

## 9. La gestion de la haute disponibilité dans Dynamics AX

La gestion de la haute disponibilité est une problématique sur laquelle il faut avoir réfléchi avant le passage en production d'un client. En effet, les mécanismes de HA sont déployés uniquement sur la plateforme de production. Ma mission a donc d'abord consisté à analyser quels sont les composants Dynamics concernés par la HA, puis, à m'intéresser à comment la mettre en place, et enfin, dans un contexte client, à analyser les éventuelles difficultés de son déploiement.

### 9.1. Le contexte projet

En vue de son passage prochain en production, un de nos clients a exprimé le besoin de mettre en place une haute disponibilité (High Availability – HA) pour la solution Gfi-SSP. A partir de là, il a été prévu d'organiser des réunions préparatoires afin de lui exposer la haute disponibilité pour les composants Dynamics AX et de lui présenter des propositions d'architectures de HA en fonction de son contexte.

L'objectif que je me suis fixé sur ce projet doit alors permettre de maîtriser la complexité de la HA dans l'ERP Dynamics AX. Pour y répondre, j'ai donc pour cela élaboré un planning d'ordonnancement des tâches à accomplir. Le projet a duré en totalité un mois et demi environ réparti entre la mi-janvier et fin février 2015. Le tableau suivant récapitule toutes les tâches réalisées pour ce projet (Tableau X).

**Tableau X - Planification des tâches pour le projet HA**

TACHE	LIBELLE	TACHES PRECEDENTES	DUREE EN JOURS
<b>ETUDE DU SUJET</b>			
T1	Recherche et analyse sur la HA	--	8
<b>CONCEPTION</b>			
T2	Rédaction d'un livrable	T1	10
<b>DEPLOIEMENT</b>			
T3	Réunions et échange avec le client	T2	2
T4	Mise en place dans un environnement de production	T3	2
T5	Suivi et analyse en production	T4	5
			<b>27 Jours</b>

Ce planning souligne évidemment que la majorité du temps a été consacré à l'analyse et à la rédaction d'un document expliquant les mécanismes de HA ainsi que les propositions d'architecture de HA pour Dynamics AX. Le contexte du projet présenté, je vais maintenant évoquer le premier travail que j'ai effectué : l'étude de la problématique de la HA par composants Dynamics AX.

## **9.2. La haute disponibilité par composants**

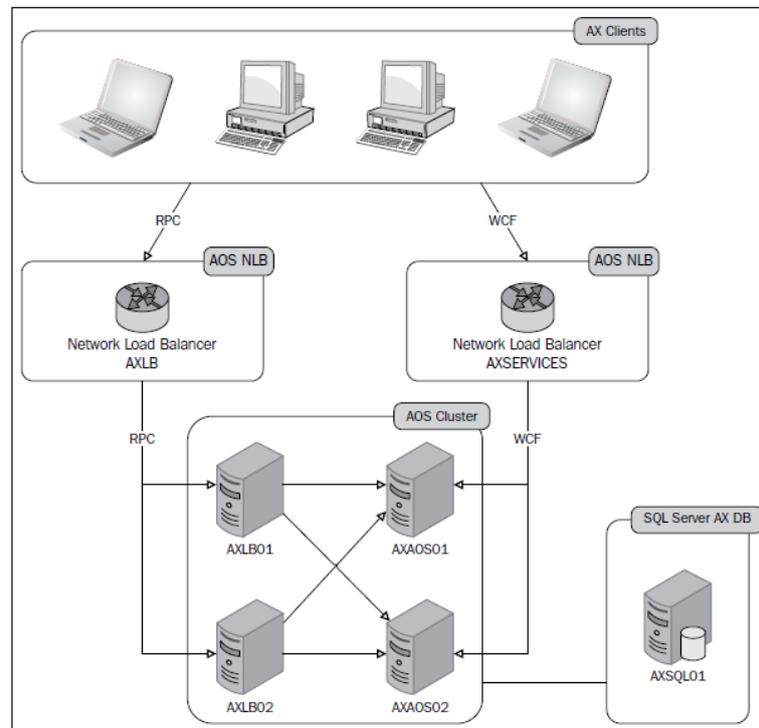
La première étape du projet a été de réaliser une étude sur la haute disponibilité pour chaque composant principal de Dynamics AX. En effet, il est nécessaire de bien comprendre tous les concepts de cette problématique avec les avantages et les limites de chaque mise en œuvre. Par conséquent, j'ai décidé de mettre en valeur les composants suivants : SQL Server, SSRS, SSAS, SharePoint et l'AOS par lequel je vais commencer.

### *9.2.1. Le composant AOS*

Au niveau de l'AOS, il est ainsi possible de mettre en place une répartition de charge (« load balancing »). Celle-ci est en effet un ensemble de techniques permettant de distribuer une charge de travail entre différents ordinateurs d'un groupe. Cela permet donc à la fois de répondre à une charge trop importante d'un service en la répartissant sur plusieurs serveurs, et de réduire l'indisponibilité potentielle de ce service que peut provoquer la panne logicielle ou matérielle d'un unique serveur. Ainsi, dans le cadre de Dynamics AX, cela se traduit par l'ajout de plusieurs instances AOS à un cluster (grappe de serveur). Pour mettre en place cette architecture, il faut alors configurer le cluster et paramétrer dans l'application Dynamics AX les instances ajoutées en tant que « Clustered AOS Instances ».

Dans le schéma suivant (Figure 25), K. Deforche et K. Saelen (2014) proposent une optimisation de la HA pour le composant AOS en utilisant l'équilibrage de charge réseau (Network Load Balancing – NLB). Le NLB est une option dans Windows Server qui équilibre la charge du trafic réseau (à l'aide d'une adresse IP virtuelle de cluster) entre plusieurs serveurs d'un cluster NLB. Ainsi, ils mettent tout d'abord en place un premier NLB (nommé AXLB dans leur schéma) pour toutes les requêtes client RPC vers l'AOS. RPC

(Remote Procedure Call) est un protocole réseau permettant de faire des appels de procédures sur un ordinateur distant à l'aide d'un serveur d'applications. Ce protocole est utilisé dans le modèle client-serveur pour assurer la communication entre les clients et le serveur AOS.



**Figure 25 - Schéma d'architecture de HA pour l'AOS avec deux répartiteurs de charge externes tiré de K. Deforche et K. Saelen (2014)**

Quand un client tente de se connecter à un AOS, le répartiteur de charge réseau choisit alors un hôte approprié (AXLB01 ou AXLB02) et transmet les requêtes client. Ensuite, il revient au répartiteur de charge au niveau application de décider laquelle des instances AOS (AXAOS01 ou AXAOS02) se chargent de la session cliente. En parallèle, ils installent un deuxième NLB (AXSERVICES) qui permet d'acheminer les services clients directement aux instances AOS (AXAOS01 et AXAOS02) au lieu de les acheminer vers les répartiteurs de charge dédiés. Ainsi, leur configuration permet que la communication du service (WCF) soit aussi répartie en charge sur les instances AOS. WCF est une infrastructure permettant de générer des applications orientées services. Ainsi, il est possible d'accéder aux fonctionnalités de Dynamics grâce à des services basés sur Windows Communication Foundation (WCF).

En résumé, la mise en place de la haute disponibilité pour l'instance AOS peut aller du simple au plus complexe. En outre, le coût peut être plus ou moins élevé. Enfin, il est possible d'avoir à tout moment un déploiement de la HA extensible à l'infini.

### 9.2.2. Le composant SQL Server

Un autre exemple d'application de la HA peut s'appliquer sur le deuxième plus important composant de Dynamics AX qui est SQL Server. Il existe effectivement plusieurs options de HA pour l'utilisation de SQL Server avec Dynamics AX que je résume dans le tableau suivant.

**Tableau XI - Options de HA pour SQL Server supportées par Dynamics AX**

CONFIGURATION HA	COMMENTAIRES
<b>Windows Server Failover Clustering (WSFC)</b>	Un cluster WSFC est un groupe de serveurs indépendants qui travaillent ensemble pour accroître la disponibilité des applications et des services.
<b>Log shipping</b>	Cette option permet d'envoyer automatiquement des sauvegardes du journal des transactions à partir d'une base de données primaire sur une instance de serveur principal à une ou plusieurs bases de données secondaires sur des instances de serveurs secondaires distincts.
<b>Transactional replication</b>	La réplication transactionnelle commence généralement par une capture des objets et des données de la base de données de publication. Dès que l'instantané initial est pris, les modifications ultérieures de données et de schémas sont généralement livrées en temps quasi réel. Pour l'utiliser avec Dynamics AX, il faut installer le KB 2765281.
<b>Snapshot replication</b>	La réplication instantanée distribue exactement les données à un moment précis dans le temps. Lorsque la synchronisation se produit, l'instantané est généré et envoyé aux abonnés.
<b>Database mirroring (jusqu'à SQL Server 2008 R2.)</b>	Lors de la synchronisation d'une session de mise en miroir de bases de données, la mise en miroir fait intervenir un serveur de secours qui permet un basculement rapide sans perte de données à partir des transactions validées.
<b>AlwaysOn Availability groups (à partir de SQL Server 2012)</b>	Cette option est une solution qui offre une alternative à la base de données en miroir. Les méthodes synchrone et asynchrone sont supportées par Dynamics AX.

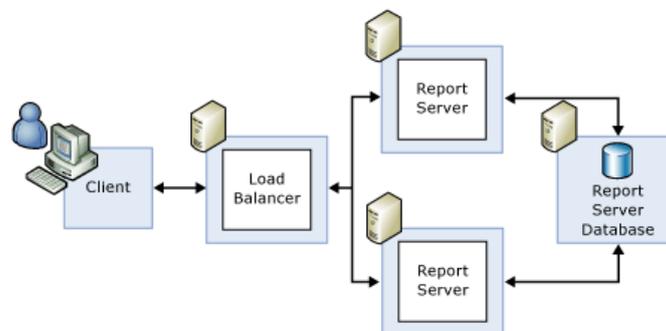
Il faut souligner que parmi toutes ces options, WSFC et AlwaysOn offrent le moins de temps d'arrêt. Mais, la mise en place de la haute disponibilité semble être assez complexe pour le composant SQL Server. De plus, dans le cas des modes actif/passif ou actif/actif du « load balancing », il faut savoir qu'il n'est pas possible d'automatiser le renvoi des instances AOS vers un serveur de basculement SQL Server en cas de panne. En effet, il faut manuellement rediriger les serveurs d'applications vers le serveur de base de données actif. Néanmoins, le serveur de stockage de données est incontestablement un sujet critique et important à considérer dans le déploiement de la haute disponibilité.

### 9.2.3. Le composant SSRS

Le troisième composant important auquel je vais m'intéresser est le serveur de reporting SSRS. Il existe deux principales configurations que j'ai retenues : « failover clustering » et « AlwaysOn Availability Groups ».

Comme on vient de le voir, SQL Server fournit le support « failover clustering » de manière qu'on peut utiliser plusieurs disques pour un ou plusieurs instances SQL Server. Toutefois, ce système est supporté seulement pour la base de données report server. Il n'est pas possible d'exécuter le service Report Server dans le cadre d'un cluster « failover ». Pour héberger une base de données report server sur un cluster « failover » SQL Server, le cluster doit déjà être installé et configuré. Ensuite, il est possible de sélectionner le cluster « failover » comme le nom du serveur quand on crée la base de données du report server dans la page de Démarrage de la base de données de l'outil de configuration Reporting Services. Les déploiements possibles avec cette solution sont :

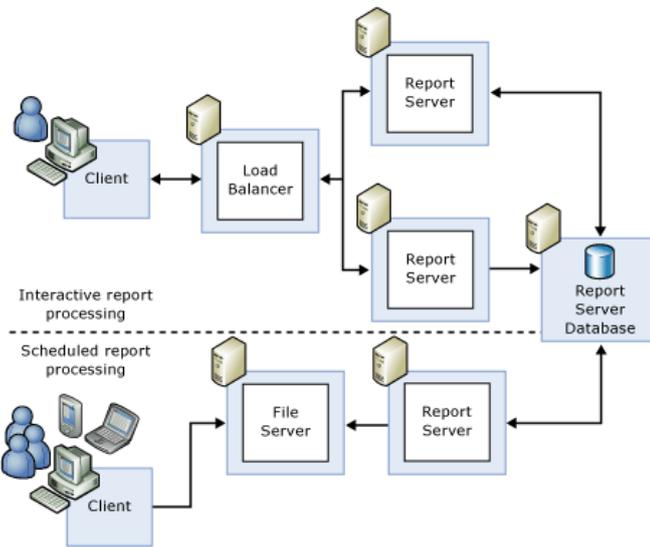
- Déploiement serveur « scale-out » (extensible à l'infini) standard (Figure 26) : dans ce cas de figure, plusieurs report server partagent une seule base de données report server. La base de données report doit être installée sur une instance SQL Server à distance.



**Figure 26 - Exemple de déploiement « scale-out » standard de SSRS tiré du site Technet (<https://msdn.microsoft.com/en-us/library/ms157293%28v=SQL.100%29.aspx>)**

- Déploiement serveur « scale-out » (extensible à l'infini) avancé (Figure 27) : on conserve les mêmes avantages que le déploiement standard, mais, l'environnement est optimisé pour des performances en séparant les serveurs report « load balanced » qui

gèrent le traitement du report interactif et un serveur report qui gère seulement des reports planifiés.



**Figure 27 - Déploiement serveur « scale-out » (extensible à l’infini) avancé standard de SSRS tiré du site Technet (<https://msdn.microsoft.com/en-us/library/ms157293%28v=SQL.100%29.aspx>)**

Ce déploiement extensible à l’infini fournit donc une haute disponibilité et une installation stable du report server. Néanmoins, cette configuration du déploiement server à l’extension infini n’est à recommander que dans des circonstances précises. En effet, celles-ci peuvent être, soit, l’existence d’un haut volume de reporting et une complexité des reports, soit un besoin que l’environnement de reporting ne rencontre pas d’arrêts d’indisponibilité ou soit une nécessité d’améliorer la performance des opérations planifiées et de livraison d’abonnement.

La mise en place de la HA pour SSRS semble donc un peu complexe mais montre une possibilité d’extensibilité. Cependant, le déploiement de la HA pour SSRS doit être réellement motivé. En effet, cela peut en effet s’avérer rapidement coûteuse.

#### 9.2.4. Le composant SSAS

Le quatrième composant important de Dynamics AX est SSAS. Il est en effet possible de concevoir des solutions de haute disponibilité dans SSAS en utilisant les technologies que je résume dans le tableau suivant.

**Tableau XII - Options de HA pour SSAS**

<b>CONFIGURATION HA</b>	<b>COMMENTAIRES</b>
<b>Network Load balancing</b> (équilibre de la charge réseau)	Les bases de données Analysis Services dans cette solution doivent être définies en tant que bases de données en lecture seule afin de garantir la cohérence entre les requêtes.
<b>Failover clustering</b> (clustering avec basculement)	Lorsqu'on doit fournir un accès complet (lecture et écriture différée) à la base de données. Cependant, l'évolutivité sur une solution de cluster de basculement peut être obtenue seulement en augmentant la taille des serveurs de cluster. Toutefois, il est important de noter que lorsqu'une instance échoue sur le nœud disponible suivant, toutes les connexions sont perdues, et tout travail non enregistré est alors également perdu.
<b>Availability Groups</b>	On peut utiliser une base de données dans ce groupe en tant que source de données dans un tableau Analysis Services ou une solution multidimensionnelle. Toutes les opérations suivantes Analysis Services fonctionnent comme prévu : données de traitement ou d'import, interrogation de données relationnelles directement (utilisant le stockage ROLAP ou le mode DirectQuery), et l'écriture différée. Le traitement et l'interrogation sont des charges de travail en lecture seule. On peut améliorer les performances en déchargeant ces charges de travail à une réplique secondaire lisible.

La mise en place de la HA pour SSAS ne paraît pas aussi complexe en comparaison à SSRS et montre une souplesse d'évolution. Cependant, elle doit être quand même justifiée par une réelle nécessité de vouloir améliorer le temps de réponse des requêtes dans Analysis Services ou par un nombre assez important d'utilisateurs finaux. En effet cette mise en place de HA peut en effet s'avérer rapidement coûteuse en machine et en maintenance.

#### *9.2.5. Le composant SharePoint*

Le dernier composant Dynamics AX que je juge intéressant pour la HA est SharePoint. Il y a deux niveaux à prendre en considération : les serveurs SharePoint et les bases de données SharePoint.

Le déploiement d'une batterie de serveurs SharePoint hautement disponible peut s'effectuer avec la redondance qui résout ainsi les domaines d'erreur et ajoute de la robustesse au système. En effet, en installant plusieurs serveurs dédiés à la fourniture de contenu, cela augmente la disponibilité en cas de défaillance dans un ou plusieurs serveurs.

Les bases de données SharePoint constituent aussi un aspect essentiel de la stratégie de haute disponibilité. Il est ainsi possible d'utiliser certaines fonctionnalités de tolérance de panne décrites plus haut pour SQL Server (clustering de basculement, mise en miroir de bases de données, AlwaysOn). Mais, ces solutions ont aussi un coût non négligeable et cela dépend donc de l'importance du composant dans l'environnement du client.

En résumé de cette revue sur la haute disponibilité par composants, il est important donc de souligner, en premier lieu, les nombreux choix possibles de déploiement des solutions. En second lieu, ces mises en place ont rapidement des coûts non négligeables. C'est pourquoi, il est important de bien cibler les besoins du client afin de lui proposer une haute disponibilité en fonction de ses moyens. Enfin, en dernier lieu, dans la majorité des cas, la complexité de mise en place peut augmenter rapidement lorsque l'on va combiner les multiples composants dans une architecture Dynamics AX. C'est d'ailleurs le point suivant que je vais maintenant traiter à travers des exemples de typologie de haute disponibilité.

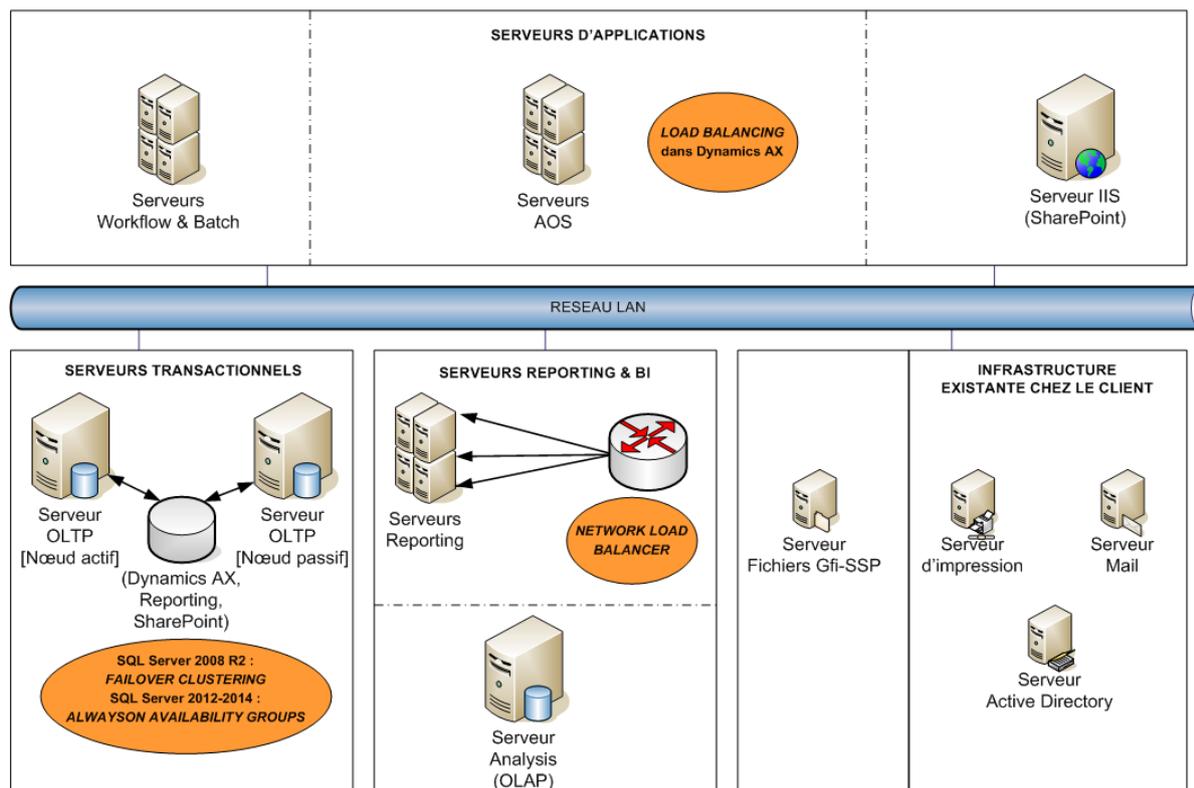
### **9.3. Une typologie de haute disponibilité**

L'étape suivante a été alors de réunir tous les éléments précédents dans le contexte de l'ERP Dynamics AX. Je me suis donc impliqué à synthétiser sous forme de schéma des déploiements de la HA pour Dynamics selon certains critères : un niveau de complexité de mise en place, le coût éventuel et le taux de disponibilité souhaité par les clients. C'est pourquoi, j'ai décidé de créer trois principaux niveaux d'architectures de déploiement de la HA.

#### *9.3.1. Premier niveau*

Dans ce premier niveau, j'ai choisi de mettre en place la HA sur seulement trois composants que je juge les plus critiques (Figure 28) : AOS, SQL Server et SSRS (Reporting).

### ARCHITECTURE POUR LA HAUTE DISPONIBILITE (PREMIER NIVEAU)



**Figure 28 - Proposition d'architecture de HA pour Dynamics AX de premier niveau**

Dans ce schéma, je propose d'abord plusieurs instances AOS installés sur différents serveurs d'applications. La répartition de charge (« load balancing ») est paramétrée directement dans l'application Dynamics AX. Cela permet ainsi de ne pas avoir de serveur AOS surchargé. Un serveur AOS principal joue le rôle de répartir les différentes connexions aux autres serveurs AOS. Concernant le coût de la mise en place de cette solution pour l'AOS, il faut tout de même ajouter des machines virtuelles.

Le deuxième élément que je mets en haute disponibilité est le composant SQL Server. Je propose de mettre en place selon la version de SQL Server soit le « failover clustering » soit le « AlwaysOn Availability Groups ». Le mode choisit par défaut est actif/passif mais il est possible d'activer le mode actif/actif avec l'option « AlwaysOn ». Le coût de ce système est la mise en place d'un deuxième serveur virtuelle.

Enfin, le dernier élément en HA dans ce schéma est le service Reporting. Je propose de mettre en place un serveur virtuel qui joue le rôle de NLB. Ainsi, il se charge automatiquement de répartir entre plusieurs serveurs de Reporting la charge de travail. Dans

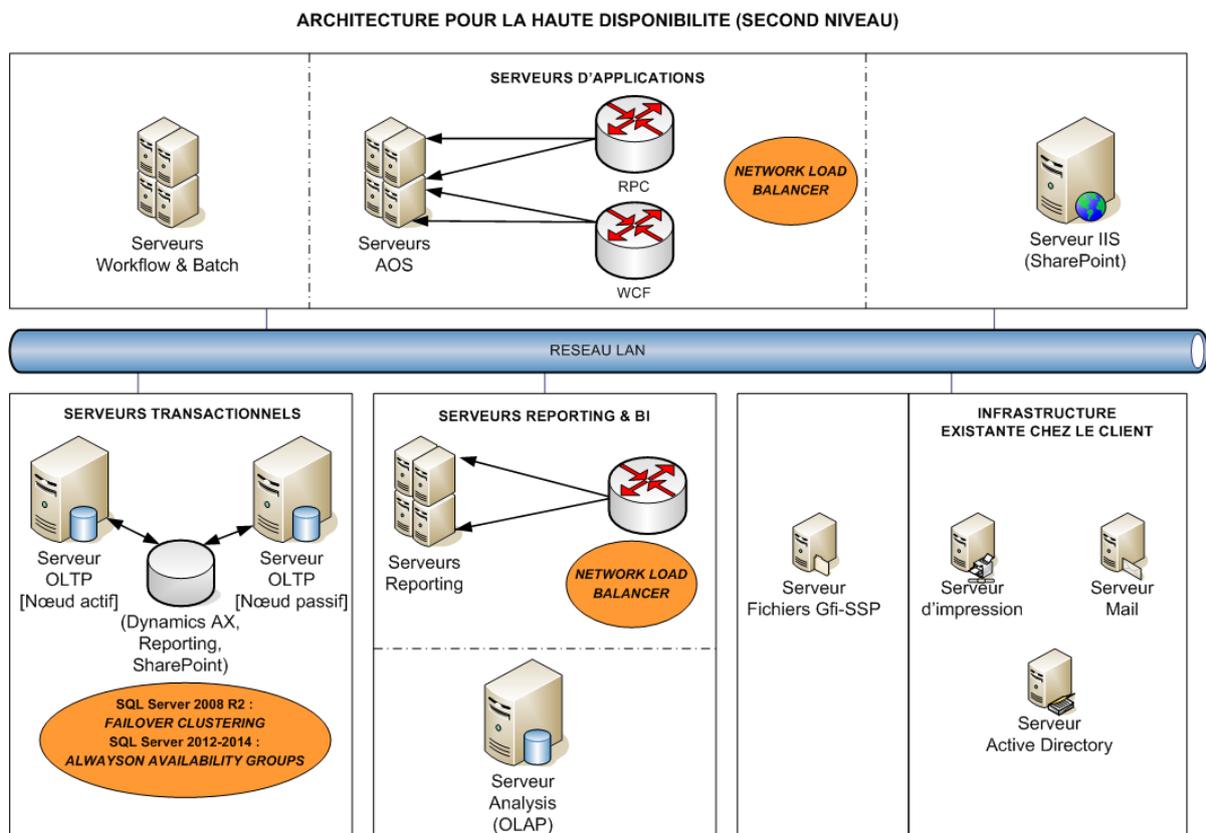
ces conditions, les gains de performance et de disponibilité sont alors évidents. Mais, cela nécessite d'avoir donc au minimum deux serveurs pour le reporting et un autre pour le NLB.

Ce premier niveau d'architecture de HA peut être par conséquent une bonne base de discussion avec le client car les composants critiques sont bien pris en charge. Le coût global de cette solution reste maîtrisé et l'architecture reste toujours extensible à l'infini. En résumé, voici ce qu'il faut retenir de ce premier niveau :

- Couverture composants : AOS, OLTP et SSRS ;
- Taux de disponibilité : moyennement élevé ;
- Complexité de mise en œuvre : simple ;
- Coût : 4 machines virtuelles supplémentaires au minimum.

### 9.3.2. Second niveau

Dans ce second niveau, je conserve évidemment le déploiement de la HA du premier niveau sur SSRS et SQL Server :



**Figure 29 - Proposition d'architecture de HA pour Dynamics AX de second niveau**

En revanche, j'optimise la HA sur les serveurs d'application AOS. En effet, comme on peut le voir sur le schéma (Figure 29), je propose de suivre les préconisations de K. Deforche et K. Saelen (2014). Ces derniers proposent de mettre en place deux répartiteurs externes qui jouent le rôle de NLB : l'un se charge des communications RPC et l'autre des communications WCF.

Dans ce contexte, la HA est nettement améliorée sur le composant AOS. Néanmoins, le coût de mise en place augmente nettement. De plus, dans cette architecture s'ajoute évidemment une complexité non négligeable dans la mise en place pour le composant AOS. En résumé, voici ce qu'il faut retenir de ce deuxième niveau :

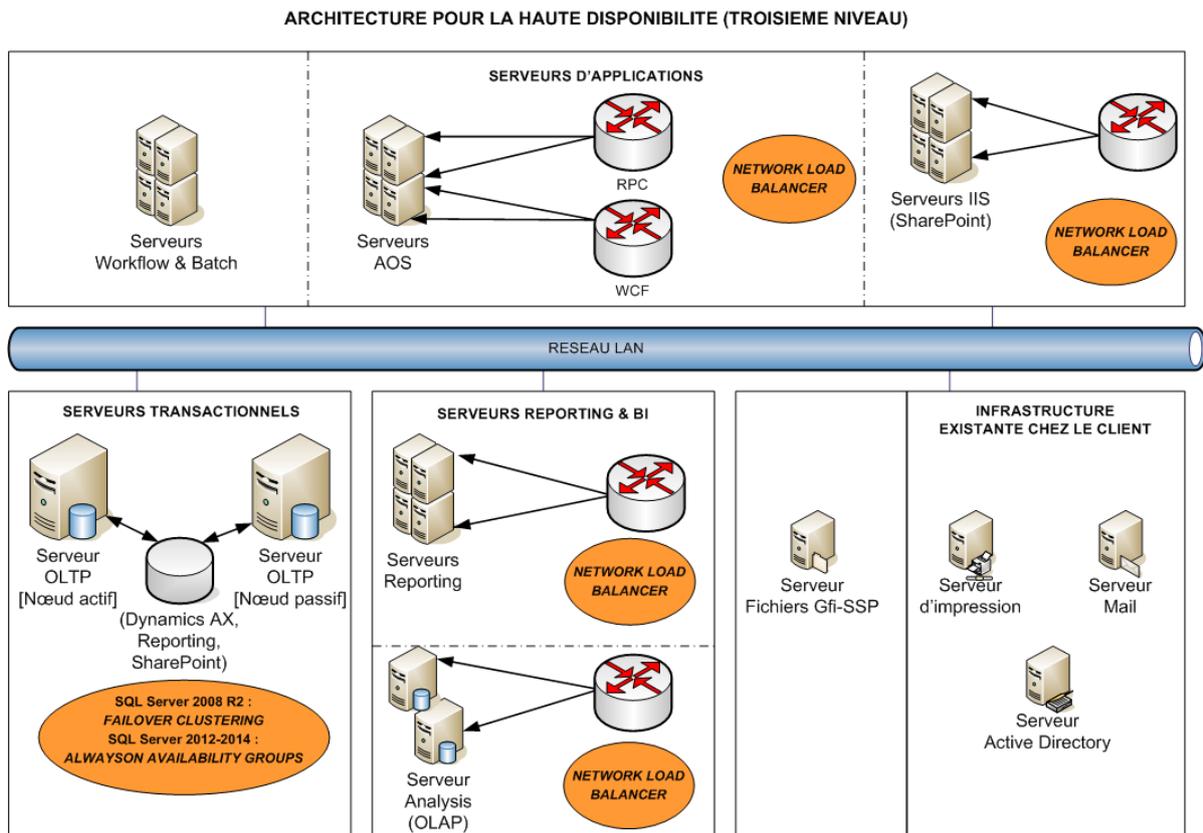
- Couverture composants : AOS, OLTP et SSRS ;
- Taux de disponibilité : élevé ;
- Complexité de mise en œuvre : moyenne ;
- Coût : 6 machines virtuelles supplémentaires au minimum.

### 9.3.3. Troisième niveau

Le dernier niveau de HA que je propose constitue la prise en charge des cinq composants décrits précédemment. Le schéma de départ est évidemment celui du deuxième niveau auquel j'y ajoute des NLB supplémentaires pour la prise en charge du SSAS et de SharePoint (Figure 30).

Une ferme SharePoint est d'abord proposée afin de permettre une redondance et une disponibilité continue. Pour SSAS, une seconde machine virtuelle est ajoutée comme pour les serveurs de Reporting.

Ainsi, tous les principaux composants de Dynamics AX sont pris en charge dans une très haute disponibilité. Néanmoins, le nombre de machines virtuelles explose considérablement ce qui entraîne des coûts matériels et logiciels non négligeables. De plus, il faut ajouter des coûts de maintenance afin de garder cette architecture à ce niveau de HA. C'est pourquoi, cette solution ne peut intéresser que des clients avec un nombre d'utilisateurs très élevé et dont les besoins en Reporting et BI sont capitaux.



**Figure 30 - Proposition d'architecture de HA pour Dynamics AX de troisième niveau**

En résumé, voici ce qu'il faut retenir de ce dernier niveau :

- Couverture composants : AOS, OLTP, SSRS, OLAP et SharePoint ;
- Taux de disponibilité : très élevé ;
- Complexité de mise en œuvre : forte ;
- Coût : 10 machines virtuelles supplémentaires au minimum.

En conclusion, à partir de ces trois exemples d'architectures de HA proposé, il nous est possible à présent d'avoir une vue précise sur ce sujet avec nos clients. On connaît maintenant la manière d'implémenter la HA sur les composants Dynamics AX, les différents coûts impliqués mais aussi les avantages et limites de chaque solution.

#### 9.3.4. Les niveaux d'indisponibilité attendus

Evidemment, selon le mode de HA choisi, on doit s'attendre tout de même à un niveau d'indisponibilité différent. En effet, avec le premier niveau proposé, si l'AOS principal tombe en panne, l'application Dynamics AX n'est plus accessible. Avec le second et le troisième

niveau proposés, si les machines jouant le rôle de NLB tombent en panne, l'application est de nouveau indisponible. En outre, si l'indisponibilité concerne les serveurs de données OLTP, il faut prendre en compte le temps nécessaire pour redémarrer les serveurs AOS afin qu'ils pointent vers le nouveau serveur de données actif. Ces divers problèmes montrent ainsi les limites de la mise en place de la haute disponibilité applicative que je propose. Cependant, en globalité, les temps d'indisponibilité restent raisonnables. Le risque majeur d'avoir des temps d'indisponibilité plus long se situe surtout du fait du choix de la virtualisation et au niveau matériel. C'est alors qu'intervient la notion de continuité d'activité.

Cette dernière a en effet pour objet de remettre en fonctionnement rapidement un site de production lorsqu'un événement majeur survient. Plusieurs notions lui sont alors associées comme le PCA (Plan de Continuité d'Activité), le PRA (Plan de Reprise d'Activité) ou le PSI (Plan de Secours Informatique). Le PCA a pour but de garantir la survie de l'entreprise après un sinistre important touchant le système informatique. Il s'agit de redémarrer l'activité le plus rapidement possible (idéalement sans interruption de service) avec le minimum de perte de données. Le PRA permet d'assurer, en cas de crise majeure ou importante d'un centre informatique, la reconstruction de son infrastructure et la remise en route des applications supportant l'activité d'une organisation. Dans ce cas, aucune infrastructure n'est commune à la plateforme cible préservant ainsi une indépendance totale. Enfin, le PSI est très proche du PRA mais ne permet qu'une reprise locale sur le même LAN.

Ainsi, avec la virtualisation de l'implémentation de Dynamics AX, ces différentes notions prennent une autre dimension. En effet, par exemple, sous VMware un PRA doit faire intervenir des répliquions bas niveau des baies de stockage avec des clusters séparés. Le PCA fait intervenir des notions de haute disponibilité entre sites. Il faut pour cela bâtir une architecture avec un SAN étendu performant avec de faibles latences associée à des techniques de virtualisation de stockage.

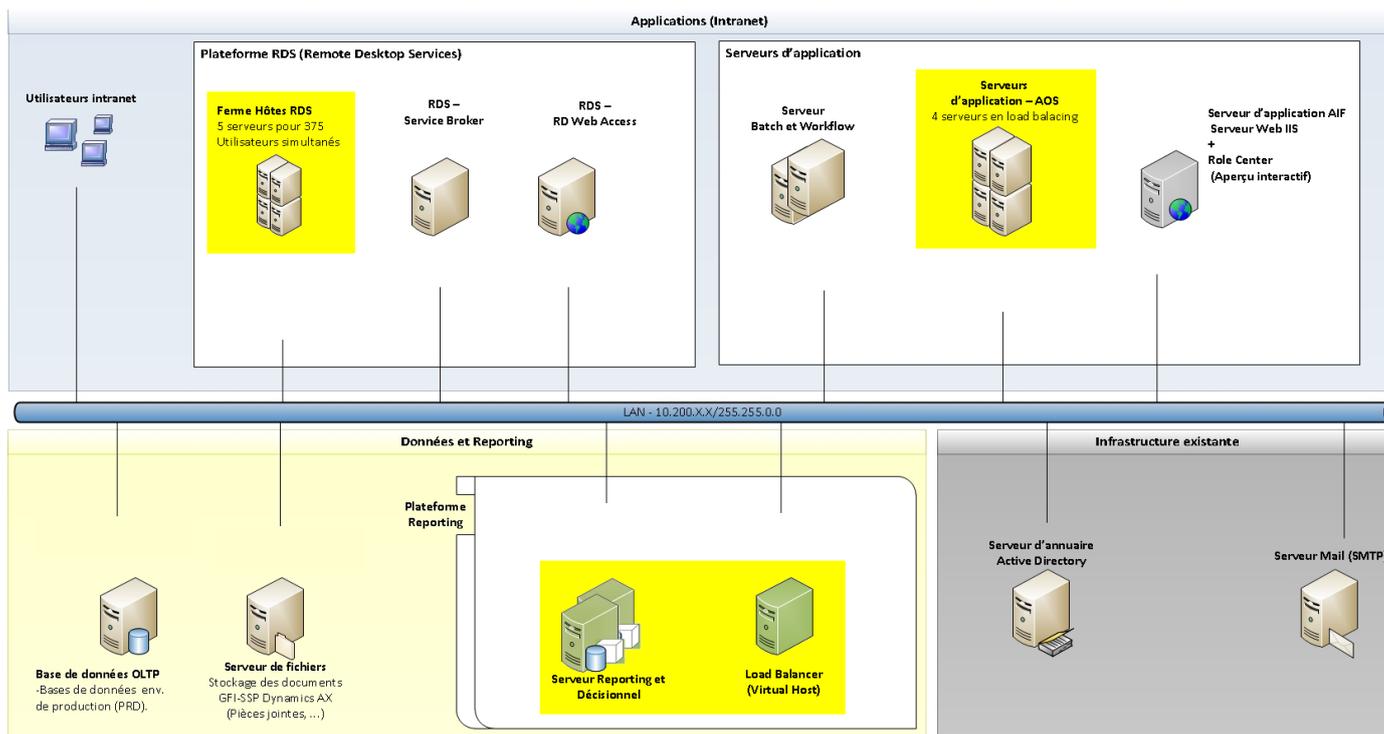
Maintenant, je vais donc traiter dans le point suivant l'expérience de la mise en œuvre de la HA chez un client en production.

#### 9.4. Le déploiement chez un client en production

Lors de réunions préparatoires (par téléphone et sur site) pour le passage en production, le sujet de la haute disponibilité a été évoqué avec le client. A partir du travail que j'ai initié en amont sur cette problématique, j'ai donc exposé lors des entrevues toutes les possibilités offertes pour déployer la HA dans l'environnement Dynamics AX. Ma démarche a été alors la suivante :

- Détailler dans les grandes lignes la HA composant par composant en assistant sur les avantages et les coûts de chaque mise en place.
- Prise en compte du contexte client :
  - ✓ 800 utilisateurs finaux dont 300 utilisateurs simultanés ;
  - ✓ Un déploiement RDS important ;
  - ✓ Une utilisation intense du reporting (SSRS) et de la BI ;
- Proposition d'un schéma de déploiement de la HA basé sur la typologie ;
- Pris en compte des remarques du client et conception d'un nouveau schéma ;
- Validation du choix du schéma par le client ;
- Déploiement de la HA sur l'environnement de production.
- Mise à jour du livrable Document d'Architecture Technique avec le nouveau schéma d'architecture de l'environnement en production.

Ainsi, la HA a été donc d'abord déployé sur les quatre serveurs d'applications AOS (Figure 31). En effet, ces derniers ont été mis en « load balancing » afin de permettre une bonne répartition de la charge. Un serveur virtuel joue le rôle de NLB entre deux serveurs reporting et décisionnel. Enfin, une ferme de cinq serveurs hôtes RDS a été mise en place afin de permettre de toujours pouvoir accéder à l'application Dynamics AX. Il faut noter que le client n'a pas jugé utile de mettre la haute disponibilité sur le serveur base de données transactionnelles (OLTP) essentiellement pour des raisons économiques. De plus, le fait qu'il est nécessaire d'arrêter les AOS afin de pouvoir se reconnecter sur une autre machine OLTP a renforcé leur idée de ne pas en mettre sur ce composant car l'application ne sera pas disponible de toutes les manières durant ce temps.



**Figure 31 - Architecture final de l'environnement en production du client avec la HA**

Ainsi, le niveau de haute disponibilité choisi pour la plateforme de production leur permet de toujours pouvoir :

- Avoir l'application Dynamics AX toujours disponible ;
- Accéder à celle-ci depuis un site distant ;
- Editer des reports.

Ce choix de HA s'explique d'abord par le nombre important de sites du client répartis en France métropolitaine mais aussi en outre-mer. La solution Gfi-SSP doit en effet être toujours disponible. La seconde raison réside dans l'importance d'accéder à l'édition des reports car le client a l'obligation de fournir des états financiers quotidiennement.

Une fois l'architecture validée, le déploiement de la plateforme en production a donc été réalisé en prenant en compte ces recommandations de haute disponibilité. La mise en place du load balancing pour les serveurs AOS a été réalisée en suivant les préconisations Microsoft décrites dans les paragraphes précédents. La ferme des hôtes RDS a été aussi mise en place assez simplement. Néanmoins, la mise en place du NLB pour la plateforme de reporting a été plus difficile.

En effet, un message d'erreur de connexion (« The underlying connection was closed : An unexpeted error occured on a receive ») apparaissait de façon aléatoire dans l'application AX lorsque des utilisateurs essayaient d'exécuter des reports. Les reports SSRS semblaient fonctionner depuis un seul nœud NLB mais pas sur l'autre. Quand on lançait un report depuis l'adresse url pointant sur le serveur 1, cela fonctionnait. Cependant, quand on lançait le même report depuis l'adresse url pointant sur le serveur 2, on avait donc une erreur. La raison est que Microsoft ne supporte pas actuellement l'exécution de SSRS dans un environnement NLB pour Dynamics AX. Effectivement, c'est parce que l'outil de déploiement report déploie seulement les fichiers d'extension dll sur un seul nœud et non pas sur tous les nœuds dans un environnement. D'après Microsoft toujours, la solution consistait à copier tous les fichiers dll du répertoire *\Program Files\Microsoft SQL Server\MSRS10.MSSQLSERVER\Reporting Services\ReportServer\bin* dans le même répertoire sur le deuxième serveur nœud. Il était aussi nécessaire de copier les fichiers de configuration *rssrvpolicy* et *rsreportserver* du premier nœud vers le deuxième nœud pour s'assurer qu'ils soient en synchronisation. Ils doivent être exactement identiques. En exécutant cette solution, l'erreur a donc été résolue.

Ainsi, la gestion de la haute disponibilité dans Dynamics AX est un projet qui a été mené dans un besoin pressant car un client passait en production. Néanmoins, c'est un travail qui nous permet de pouvoir discuter du sujet de la haute disponibilité avec les clients, de leur proposer des architectures de HA et d'améliorer au passage la qualité de notre processus d'intégration technique. Le résultat de ce travail d'ingénierie est d'ailleurs désormais inclus dans notre livrable : le DAT (Document d'Architecture Technique). C'est aussi dans ce même objectif que l'on m'a confié la mise en place d'un dernier projet : la supervision de Dynamics AX.

## 10. La mise en place de la supervision de Dynamics AX

L'intérêt de la supervision de Dynamics AX pour les clients est d'avoir des métriques de performance acceptable et de détecter des défauts de lenteurs dans les environnements pour les résoudre. Après avoir sélectionné les indicateurs de performances pour surveiller un environnement Dynamics AX, il a fallu tester leur mise en place dans une solution de supervision. Enfin, j'ai été confronté à la surveillance d'une plateforme d'un client en production.

### 10.1. Le contexte projet de la supervision

Le projet a été initié par le client lors d'une réunion de comité de pilotage. Son besoin était de pouvoir superviser ses différents environnements Dynamics AX avec une solution de supervision déjà en sa possession : Centreon. L'objectif de cette mission a donc consisté à fournir au client les clés afin de pouvoir superviser ses environnements Dynamics AX avec l'outil Centreon.

Le projet a duré en totalité quarante-un jours entre début septembre 2014 et mi-novembre 2014. Les tâches du projet ont été regroupées en trois grandes macro-tâches : l'étude et l'analyse, la conception et enfin, le déploiement et les tests (Tableau XIII).

**Tableau XIII - Planification des tâches pour le projet de supervision de Dynamics AX**

TACHE	LIBELLE	TACHES PRECEDENTES	DUREE EN JOURS
<b>ETUDE ET ANALYSE</b>			
<b>T1</b>	Préparation et réunion avec le client	--	2
<b>T2</b>	Recherche sélection des moyens et outils de supervision	T1	15
<b>T3</b>	Recherche des seuils d'alerte	T2	8
<b>CONCEPTION</b>			
<b>T4</b>	Rédaction d'un livrable V1.0	T3	5
<b>DEPLOIEMENT ET TESTS</b>			
<b>T5</b>	Installation d'un serveur dédié à la supervision	T1	1
<b>T6</b>	Mise en place et maîtrise de Nagios	T4	2
<b>T7</b>	Mise en place et maîtrise de Centreon	T4	2
<b>T8</b>	Implémentation des compteurs dans Nagios et Centreon	T5, T6	4
<b>T9</b>	Rédaction d'un livrable V2.0	T8	2
			<b>41 jours</b>

Comme le montre ce tableau, j'ai donc commencé par réfléchir sur la manière dont j'allais aborder cette problématique de supervision.

## **10.2. L'étude de la supervision de Dynamics AX**

### *10.2.1. La méthode*

Afin de cerner exhaustivement ce sujet, j'ai élaboré tout d'abord la stratégie d'approche suivante :

- Lister tous les composants candidats intéressants à la supervision :
  - ✓ Au niveau du tiers DATA :
    - SQL Server
    - SSRS
    - SSAS
  - ✓ Au niveau du tiers APPLICATION :
    - AOS
    - EP
- Sélectionner et décrire tous les compteurs pertinents pour une supervision ;
- Se documenter sur la supervision de Dynamics AX ;
- Se documenter sur les outils Nagios et Centreon ;
- Mettre en place un environnement de supervision en interne ;
- Tester l'implémentation des compteurs dans les solutions Nagios ;
- Documenter les résultats dans un livrable.

En mettant en place cette stratégie, j'ai pu ainsi monter progressivement en compétence sur le sujet en essayant de m'appliquer à bien couvrir chaque niveau étudié. Ainsi, j'ai pu commencer à analyser les moyens existants pour pouvoir superviser l'ERP Dynamics AX.

### *10.2.2. Les moyens de supervision*

La solution Microsoft Dynamics AX fournit plusieurs informations concernant l'état de son système à travers trois principaux types de moyens :

- Les compteurs de performance : mesurent l'état ou l'activité du système. Ils peuvent être intégrés dans le système d'exploitation ou appartenir à des applications individuelles.
- Les journaux des événements : enregistrent les événements significatifs. Il existe quatre principaux types d'événements : Application, Sécurité, Installation, Système. Les événements sont classés comme erreur, avertissement ou informations selon la gravité de l'événement. Une erreur indique un problème important, comme la perte de données. Un avertissement est un événement qui n'est pas nécessairement significatif mais qui peut annoncer des problèmes ultérieurs. Une information décrit le bon fonctionnement d'un programme, d'un pilote ou d'un service.
- Les services Windows : peuvent être automatiquement démarrés en même temps que l'ordinateur, suspendus puis redémarrés. Il est ainsi possible de vérifier si les services sont bien en exécution.

Ces différents moyens de supervision peuvent être implémentés dans des solutions de supervision du marché que je vais maintenant présenter.

### 10.2.3. Les solutions de supervision

J'ai entrepris de classer dans un tableau toutes les solutions possibles de supervision pour Dynamics AX (Tableau XIV).

**Tableau XIV - Liste des solutions de supervision pour Dynamics AX**

OUTIL	DESCRIPTION	COMPLEXITE	COUVERTURE DYNAMICS AX
Windows Performance Monitor (PerfMon)	Cet outil permet de collecter des informations sur la manière dont les ressources systèmes telles que le CPU sont utilisées sur une longue période. <b>GRATUIT</b>	Simple	Minimale
Resource Monitor	Il fournit une vue rapide sous forme de tableau de bord d'un état actuel des performances du système.	Simple	Minimale
Dynamics AX Trace Parser	Cet outil est utilisé pour trouver la cause d'un processus lent.	Moyen	Minimale
SQL Server Profiler	C'est l'outil de base de SQL Server pour capturer les requêtes de longues exécutions.	Simple	Minimale
Performance Analyser pour Dynamics AX	Il permet d'aider à traquer la plupart des processus lents dans un système. Cet outil est utilisé pour commencer une évaluation de performance.	Moyen	Intermédiaire
SCOM	Cette solution peut superviser tous les serveurs d'un environnement Dynamics AX et peut être utilisé pour réaliser des opérations de maintenance. <b>PAYANT</b>	Simple	Complète
Nagios / Centreon	Ces solutions libres peuvent aussi également superviser tous les serveurs d'un environnement Dynamics AX et peut être utilisé pour réaliser des opérations de maintenance. <b>GRATUIT</b>	Difficile	Complète

J'ai mis en avant deux critères qui me sont apparus importants : la complexité de déploiement et la couverture de l'ERP Dynamics AX par l'outil. Par conséquent, il y a deux solutions qui couvrent la majorité des besoins en supervision de l'ERP Dynamics AX : SCOM de Microsoft et Nagios / Centreon. Il est évidemment possible d'utiliser conjointement les outils sur les plateformes Dynamics AX. Mais, comme le client possède déjà une plateforme de supervision sous Nagios-Centreon, je me suis donc limité à tester la surveillance de Dynamics AX avec cette solution.

### 10.3. Superviser Dynamics AX dans Centreon

#### 10.3.1. Le choix des compteurs de performances

Les compteurs de performances pour Dynamics ont été sélectionnés par mes soins pour leur pertinence. Ensuite, je les ai classés dans des tableaux par type de composant en indiquant dans la colonne « Utilisation » un seuil permettant de fixer les niveaux d'alertes.

➤ Exemples de compteurs communs à tous les serveurs des environnements :

Le tableau suivant décrit les compteurs qui sont typiquement utilisés dans la supervision du processus AOS, du serveur de base de données et du serveur IIS :

**Tableau XV - Compteurs communs aux serveurs Windows**

OBJET	COMPTEUR	BASE DE DONNEES	AOS	IIS	UTILISATION
Mémoire	Mégaoctets disponibles	X	X	X	<p>Il permet de connaître la quantité de mémoire physique disponible pour les processus en cours d'exécution. Il affiche seulement la dernière valeur observée et il ne constitue pas une moyenne.</p> <p><b>Seuil :</b> Il est préférable d'avoir une valeur supérieure à 100 MB. Une valeur constante de moins de 20% à 25% de la RAM est une indication de mémoire insuffisante.</p> <p><b>Exemples :</b>  <b>Warn = 100</b>  <b>Crit = 90</b></p>

OBJET	COMPTEUR	BASE DE DONNEES	AOS	IIS	UTILISATION
Processus (sqlserv, Ax32Serv, w3wp)	% temps processeur	X	X	X	<p>Ce pourcentage indique le temps écoulé par les processus threads à utiliser le processeur pour exécuter des instructions.</p> <p><b>Seuil</b> : la valeur est calculée sur la base suivante : Nombre de CPUs logiques * 100. Un chiffre donc qui dépasse cette valeur peut donc souligner que le processus utilise largement le « multithreading » et le processeur bien au-dessus de sa capacité. Pour sqlservr, une valeur inférieure à 80% est souhaitable.</p> <p><b>Exemples</b> :</p> <p><b>Warn = 80</b></p> <p><b>Crit = 90</b></p>
	%Privileged Time (% temps privilégié)	X	X	X	<p>Ce compteur indique le pourcentage de temps que les processus threads sont exécutés en mode privilégié. Lorsqu'une application appelle l'utilisation des fonctions du système (par exemple pour allouer de la mémoire ou exécuter un fichier), ces fonctions du système d'exploitation sont exécutées en mode privilégié.</p> <p><b>Seuil</b> : un chiffre qui est constamment au-dessus de 75% peut indiquer un goulot d'étranglement.</p> <p><b>Exemples</b> :</p> <p><b>Warn = 70</b></p> <p><b>Crit = 80</b></p>
Processor (Processeur)	%Privileged Time (% temps privilégié)	X	X	X	<p>Ce compteur indique le pourcentage de temps qu'un « thread » (processus léger) est exécuté en mode privilégié.</p> <p><b>Seuil</b> : un chiffre qui est constamment au-dessus de 75% peut indiquer un goulot d'étranglement.</p> <p><b>Exemples</b> :</p> <p><b>Warn = 70</b></p> <p><b>Crit = 80</b></p>
	%Processor Time (% temps processeur)	X	X	X	<p>Ce compteur est l'indicateur principal de l'activité du processeur. De nombreuses valeurs élevées ne sont pas forcément un mauvais signe. Cependant, si les autres compteurs associés au processeur augmentent linéairement (comme %Privileged Time), il est peut-être utile d'examiner l'utilisation élevée du CPU.</p> <p><b>Seuil</b> : le chiffre général comme seuil limite pour les processeurs est de 65%.</p> <p><b>Exemples</b> :</p> <p><b>Warn = 60</b></p> <p><b>Crit = 70</b></p>

OBJET	COMPTEUR	BASE DE DONNEES	AOS	IIS	UTILISATION
Disque physique	Moyenne disques/lecture	X	X	X	<p>Ce compteur indique le temps moyen, en secondes, d'une lecture de données sur le disque.  <b>Seuil</b> : aucune valeur spécifique.            Toutefois pour exemple :</p> <ul style="list-style-type: none"> <li>▪ Moins de 10 ms : très bon ;</li> <li>▪ Entre 10-20 ms : bon ;</li> <li>▪ Entre 20-50 ms : attention (lent) ;</li> <li>▪ Plus de 50 ms : critique (goulot d'étranglement)</li> </ul> <p><b>Exemples :</b>  <b>Warn = 30</b>  <b>Crit = 50</b></p>
	Moyenne disques/écriture	X	X	X	<p>Ce compteur indique le taux d'opérations d'écriture sur le disque.  <b>Seuil</b> : aucune valeur spécifique.            Toutefois pour exemple sans mis en cache :</p> <ul style="list-style-type: none"> <li>▪ Plus de 20 ms : mauvais ;</li> <li>▪ Moins de 20 ms : moyen ;</li> <li>▪ Moins de 12 ms : bon ;</li> <li>▪ Moins de 8 ms : très bon ;</li> </ul> <p><b>Exemples :</b>  <b>Warn = 15</b>  <b>Crit = 20</b></p> <p>Exemples avec mis en cache :</p> <ul style="list-style-type: none"> <li>▪ Plus de 4 ms : mauvais ;</li> <li>▪ Moins de 4 ms : moyen ;</li> <li>▪ Moins de 2 ms : bon ;</li> </ul> <p>Moins de 1 ms : très bon.  <b>Exemples :</b>  <b>Warn = 3</b>  <b>Crit = 5</b></p>

➤ Exemples de compteurs pour le composant AOS :

Le tableau suivant décrit les compteurs de l'objet *Microsoft Dynamics AX Object Server* pour mesurer le serveur d'objets d'application (AOS).

**Tableau XVI - Compteurs de performance pour le composant AOS**

COMPTEUR	DESCRIPTION	UTILISATION
ACTIVE SESSIONS	Sessions actives	<p>Le nombre actuel de sessions actives.</p> <p>Ce compteur permet ainsi de superviser les utilisateurs connectés sur une instance AOS.  <b>Seuil</b> : cela peut servir à fixer une limite sur les sessions actives et être alerté dans le cas où cette limite est dépassée.</p>
NUMBER OF BYTES RECEIVED BY SERVER	Nombre d'octets reçus par serveur	<p>Le nombre d'octets reçus par l'instance AOS depuis qu'elle a démarré.</p> <p>Ce compteur permet d'obtenir une valeur informative.  <b>Seuil</b> : aucune valeur spécifique.</p>

COMPTEUR		DESCRIPTION	UTILISATION
<b>NUMBER OF BYTES SENT BY SERVER</b>	Nombre d'octets envoyés par serveur	Le nombre d'octets que l'instance AOS a envoyé depuis qu'elle a démarré.	Ce compteur permet d'obtenir une valeur informative. <b>Seuil</b> : aucune valeur spécifique.
<b>NUMBER OF CLIENT REQUESTS</b>	Nombre de requêtes clients	Le nombre de requêtes client-serveur depuis que l'instance AOS a démarré.	Ce compteur permet d'obtenir une valeur informative. <b>Seuil</b> : aucune valeur spécifique.
<b>NUMBER OF CLIENT REQUESTS PER SECOND</b>	Nombre de requêtes clients par secondes	Le nombre de requêtes client-serveur que traite l'instance AOS pas secondes.	Ce compteur permet d'obtenir une valeur informative. <b>Seuil</b> : aucune valeur spécifique.
<b>NUMBER OF SERVER REQUESTS</b>	Nombre de requêtes serveurs	Le nombre de requêtes serveur-client qui ont été traité depuis que l'instance AOS a démarré.	Ce compteur permet d'obtenir une valeur informative. <b>Seuil</b> : aucune valeur spécifique.
<b>TOTAL SESSIONS</b>	Total de sessions	Le nombre total de sessions actives depuis que l'instance AOS a démarré.	Ce compteur permet ainsi de superviser le nombre total de connexion sur une instance AOS. <b>Seuil</b> : cela peut donc servir à fixer une limite sur le total de sessions et être alerté dans le cas où cette limite est dépassée.
<b>TOTAL NUMBER OF HITS</b>	Nombre total de succès	Le nombre total de fois qu'un enregistrement a été trouvé avec succès dans le cache.	Ce compteur peut être mis en relation avec <b>Total Number of Selects on Cached Tables</b> afin de déterminer un pourcentage de réussite. Une valeur élevée peut permettre de conclure que le traitement des enregistrements est assez rapide car ils sont réalisés par le serveur AOS. <b>Seuil</b> : aucune valeur spécifique.
<b>TOTAL NUMBER OF MISSES</b>	Nombre total d'échec	Le nombre total de fois qu'un enregistrement n'a pas été trouvé dans le cache.	Ce compteur peut être mis en relation avec <b>Total Number of Selects on Cached Tables</b> afin de déterminer un pourcentage d'échec. Une valeur élevée peut permettre de conclure que le traitement des enregistrements est assez lent car ils sont réalisés par la base de données. <b>Seuil</b> : aucune valeur spécifique.
<b>TOTAL NUMBER OF SELECTS ON CACHED TABLES</b>	Nombre total de selects sur les tables en cache	Le nombre total de selects sur les tables mises en cache.	Le cache côté serveur AOS est utilisé quand un select est exécuté. Si aucun enregistrement ne se trouve dans le cache, alors l'enregistrement est traité par la base de données. Le nombre maximal d'enregistrements à conserver dans la mémoire cache du serveur AOS est configurable dans la partie Cache Limits de l'application Dynamics AX. La valeur par défaut est à 2000 enregistrements pour une entreprise donnée et par groupe de tables ou par AOS. <b>Seuil</b> : ne pas dépasser la limite fixée dans les paramétrages de l'application car une fois atteint la capacité de la mémoire cache (128 Ko), c'est la recherche disque qui prend le relais. Cependant, elle s'avère beaucoup plus lente que la recherche mémoire.

➤ Compteurs pour SQL Server :

Le tableau suivant décrit les compteurs qui sont typiquement utilisés dans la supervision de SQL Server :

**Tableau XVII - Compteurs de performance pour SQL Server**

OBJET	COMPTEUR		UTILISATION
SQLServer:Access Methods	Forwarded Records/sec	Enregistrements transmis/sec	<b>Seuil</b> : ce compteur doit être en-dessous de 10% du compteur <b>Batch Requests/sec</b> . Sinon, cela signifie que les opérations insert s'effectuent sur les tables HEAP (tables sans index cluster). Il est possible de résoudre le problème en créant un index cluster sur ces tables HEAP. <b>Exemples</b> : <b>Warn = 10</b> <b>Crit = 20</b>
	FreeSpace Scans/sec	Balayages de l'espace libre/sec	Nombre de scans par secondes qui ont été lancés à la recherche d'un espace libre à l'intérieur de pages déjà allouées pour insérer ou modifier une partie d'enregistrement. Chaque scan peut trouver plusieurs pages. <b>Seuil</b> : inférieur à 10 par 100 <b>Batch Requests/sec</b> .
	Full Scans/sec	Balayages complets/sec	<b>Seuil</b> : il faut avoir 1 <b>Full Scans</b> pour chaque 1000 <b>Index Searches</b> . Sinon, on a plusieurs scans table/index qui sont généralement gourmand en ressources. Ce point est renforcé si on a en même temps une utilisation élevée du CPU. Pour remédier à ce problème, il faut vérifier les index existants et ajouter d'éventuels index manquants.
	Workfiles Created/sec	Fichiers de travail créés/sec	Nombre de fichiers de travail créés par seconde. Par exemple, les fichiers de travail peuvent être utilisés pour stocker temporairement des résultats pour des jointures ou des agrégats de hachage. <b>Seuil</b> : inférieur à 20 par 100 <b>Batch Requests/sec</b> .
	Worktables Created/sec	Tables de travail créées/sec	Nombre de tables de travail créées par seconde. Par exemple, les tables de travail peuvent être utilisées pour stocker les résultats temporaires de requête spool, des variables XML, et des curseurs. <b>Seuil</b> : inférieur à 20 par 100 <b>Batch Requests/sec</b> .
SQLServer:Buffer Manager	Buffer Cache hit ratio	Taux de réussite de la mémoire cache.	Ce compteur indique la fréquence de SQL Server à retrouver les pages de données dans son cache buffer lorsqu'une requête a besoin d'une page de données. <b>Seuil</b> : si la valeur est haute, cela signifie que SQL Server a été en mesure d'obtenir des données pour les requêtes à partir de la mémoire et non du disque. Sinon, un taux faible peut indiquer un problème de mémoire. <b>Exemples</b> : <b>Warn = 30</b> <b>Crit = 20</b>
	Free list stalls/sec	« Free list stalls » /sec	Ce compteur mesure le nombre de requêtes par seconde qui ont dû attendre pour une page libre. S'il n'y a pas de pages libres dans le cache buffer, une requête doit attendre qu'une d'entre elle se libère. <b>Seuil</b> : la valeur recommandée doit être inférieure à 2. Si la valeur est supérieure, vérifier les compteurs <b>Page Life Expectancy/sec</b> et <b>Lazy Writes/sec</b> . Dans le cas où <b>Page Life Expectancy</b> est inférieur à 300/sec et <b>Lazy Writes</b> au-dessus de 2/sec, alors il existe un signe clair de pression de mémoire. <b>Exemples</b> : <b>Warn = 2</b> <b>Crit = 4</b>

OBJET	COMPTEUR		UTILISATION
	Lazy Writes/sec	Ecrits qui prennent du temps.	<b>Seuil</b> : ce compteur doit être inférieur à 20. Sinon, cela indique qu'il faut allouer plus de mémoire au processus SQL Server. <b>Exemples</b> : <b>Warn = 20</b> <b>Crit = 30</b>
	Page Life Expectancy	Espérance de vie de page.	Ce compteur mesure combien de temps les pages restent dans le cache buffer en secondes. Plus une page reste en mémoire et plus SQL Server n'a pas besoin de la lire sur le disque pour résoudre une requête. <b>Seuil</b> : il faut surveiller ce compteur au fil du temps pour déterminer une base de référence afin de déterminer une normalité dans l'environnement supervisé. Rien en-dessous de 300 secondes (5 minutes) peut signifier qu'il est nécessaire d'ajouter de la mémoire.
	Page Lookups/sec	Recherches de page/sec.	<b>Seuil</b> : ce compteur doit être inférieur à celui de ( <b>Batch Requests/sec</b> * 100). Sinon, des plans inefficaces sont générés et trop de pages sont consultées. La cause peut-être parfois dû aux indexes fragmentés et des statistiques obsolètes. Dans d'autres cas, ce sont les requêtes qui ont besoin d'un réglage.
<b>SQLServer:Locks</b>	Lock Requests/sec	Demandes de verrouillage/sec	Ce compteur mesure le nombre de nouveaux verrous et de traitements bloqués par seconde demandés par le gestionnaire de verrouillage. <b>Seuil</b> : <b>Lock Request/sec / Batch Requests/sec</b> < 1000
<b>SQLServer:SQL Statistics</b>	SQL Compilations/sec	Compilations SQL/sec	Ce compteur indique le nombre de fois que SQL Server compile un plan d'exécution par seconde. C'est une opération gourmande en ressources. <b>Seuil</b> : ce compteur doit être comparé à celui de <b>Batch Requests/sec</b> pour obtenir une indication de l'existence ou non de problèmes pouvant nuire à la performance. Pour cela, il faut diviser le nombre de <b>Batch Requests/sec</b> par le nombre de <b>SQL Compilations/sec</b> pour obtenir un rapport. Idéalement, il faut avoir 1 compilation pour 10 requêtes batch.
	SQL Re-Compilations/sec	Recompilations SQL/sec	Lorsque le plan d'exécution est annulé en raison d'un événement important, SQL Server le recompilera. Ce compteur mesure donc le nombre de fois qu'un événement recompilé a été déclenché par seconde. <b>Seuil</b> : les recompilations comme les compilations sont des opérations coûteuses. L'idéal est de conserver ce compteur à moins de 10% du nombre de <b>SQL Compilations/sec</b> .
	Batch Requests/sec	Requêtes Batch/sec	Ce compteur indique le nombre de batch reçus par SQL Server par secondes. C'est un bon indicateur de la quantité de travail en cours de traitement par la boîte SQL Server. <b>Seuil</b> : plus la valeur est haute et plus les requêtes ont été exécutées par la boîte SQL Server. Il faut surveiller ce compteur au fil du temps pour déterminer une base de référence afin de déterminer une normalité dans l'environnement supervisé.

### 10.3.2. Des exemples d'implémentation dans Nagios-Centreon

La supervision sous Nagios des environnements Dynamics AX peut se faire à travers l'installation d'agents Nagios (NSClient++). Parmi les commandes disponibles dans Nagios, il est possible d'utiliser le « check\_nt » pour les services Windows, le « checkEventLog » pour les journaux d'événements Windows et le « checkCounter » pour implémenter les compteurs de performance. Afin d'illustrer cet implémentation, je vais donc présenter des

exemples que j'ai moi-même réalisés dans le cadre de tests. Pour cela, j'ai procédé de la façon suivante :

- Déclarer les machines concernées par la supervision dans le fichier *hosts.cfg* ;
- Déclarer une commande dans le fichier *checkcommands.cfg* ;
- Déclarer ensuite un service dans le fichier *services.cfg* (c'est dans la déclaration de ce service, qu'on lie la commande aux noms des ordinateurs à superviser).

Pour connaître avec exactitude le nom de l'objet et de l'instance à superviser, il existe deux principales méthodes permettant la récupération des noms des compteurs de performances pour un environnement Microsoft Dynamics AX installé. La première consiste à accéder à la liste depuis la base de registre Windows, la deuxième directement à travers l'analyseur de performance perfmon.exe intégré nativement dans Windows Server. L'objet est séparé du compteur par un slash.

#### 1) Exemple de commande pour superviser les compteurs dans Centreon

La syntaxe générale du compteur de performance se compose de la manière suivante : *NOM\_OBJET(NUMERO\_INSTANCE)\NOM\_COMPTEUR*. Voici donc un exemple de commande dans Centreon basé sur le « checkCounter » qui peut ainsi superviser les compteurs de performance (Figure 32) :

```
define command{
    command_name      check_counter
    command_line      $USER1$/check_nt $HOSTADDRESS$ -p $ARG1$ -s $ARG2$ -v COUNTER -l '\\
$ARG3$\\$ARG4$' -d SHOWALL -w $ARG5$ -c $ARG6$
    ;$ARG1$           Port
    ;$ARG2$           Mot de passe
    ;$ARG3$           Nom de l'objet à superviser
    ;$ARG4$           Nom du compteur de performance à superviser
    ;$ARG5$           Niveau d'alerte
    ;$ARG6$           Niveau critique
}
```

**Figure 32 - Commande check\_counter décrite dans le fichier checkcommands.cfg**

Dans cette commande, afin de bien prendre en compte le slash qui sépare le nom de l'objet du compteur de performance, j'ai utilisé la syntaxe suivante entre simple quote : `'\\$ARGn$\\$ARGn$'`. J'ai donc scindé en deux arguments distincts et doublé les slashes pour que les slashes soient pris en compte.

## 2) Exemple de service pour superviser les compteurs dans Centreon

Voici trois autres exemples de définitions de service basé sur la même commande précédente « checkCounter » :

- Définition de service pour le compteur *Microsoft Dynamics AX Object Server(01)\ACTIVE SESSIONS* :

**Tableau XVIII - Extrait du tableau Compteurs pour le composant AOS**

COMPTEUR		DESCRIPTION	UTILISATION
<b>ACTIVE SESSIONS</b>	Sessions actives	Le nombre actuel de sessions actives.	Ce compteur permet ainsi de superviser les utilisateurs connectés sur une instance AOS. <b>Seuil</b> : cela peut servir à fixer une limite sur les sessions actives et être alerté dans le cas où cette limite est dépassée.

Le service suivant supervise l'objet *Microsoft Dynamics AX Object Server(01)* instance *01* et son compteur de performance *ACTIVE SESSIONS* (Figure 33):

```
define service{
    host_name                WIN-03-AX
    service_description      AOS_ACTIVE SESSIONS
    _SERVICE_ID            44
    use                      generic-service
    check_command            check_counter!12489!centreon!Microsoft Dynamics AX Object Server(01)!ACTIVE SESSIONS!5!10
    max_check_attempts      5
    normal_check_interval    10
    retry_check_interval    10
    check_period            24x7
    event_handler_enabled    1
    notification_interval   10
    notification_period     24x7
    notification_options    w,u,c,r,f,s
    contacts                Software_Software
}
```

**Figure 33 - Définition du service AOS\_ACTIVE\_SESSIONS dans le fichier services.cfg**

Voici le résultat dans l'interface Centreon (Figure 34) :

<input type="checkbox"/>	AOS_ACTIVE SESSIONS		OK	2h 26m 39s	17/11/2014 12:31:54	1/5 (H)	2
<input type="checkbox"/>	AOS_TOTAL_NUMBER_CLEARS_TRIGGERED		OK	2h 26m 1s	17/11/2014 12:32:32	1/5 (H)	0
<input type="checkbox"/>	AOS_TOTAL_NUMBER_CLEARS		WARNING	2h 25m 24s	17/11/2014 12:33:09	5/5 (H)	2395
<input type="checkbox"/>	AOS_TOTAL_NUMBER_DELETES		WARNING	2h 24m 46s	17/11/2014 12:33:47	5/5 (H)	1523
<input type="checkbox"/>	AOS_TOTAL_NUMBER_HITS		WARNING	2h 26m 31s	17/11/2014 12:32:02	5/5 (H)	1552

**Figure 34 - Visualisation du service AOS\_ACTIVE\_SESSIONS dans Centreon**

➤ Définition de service pour le compteur *Processus\% temps processeur* :

**Tableau XIX - Extrait du tableau Compteurs communs pour les serveurs**

OBJET	COMPTEUR	BASE DE DONNEES	AOS	IIS	UTILISATION
Processus (sqlserv, Ax32Serv, w3wp)	% temps processeur	X	X	X	<p>Ce pourcentage indique le temps écoulé par les processus threads à utiliser le processeur pour exécuter des instructions.</p> <p><b>Seuil</b> : la valeur est calculée sur la base suivante : Nombre de CPUs logiques * 100. Un chiffre donc qui dépasse cette valeur peut donc souligner que le processus utilise largement le « multithreading » et le processeur bien au-dessus de sa capacité. Pour sqlservr, une valeur inférieure à 80% est souhaitable.</p> <p><b>Exemples</b> :</p> <p><b>Warn = 80</b></p> <p><b>Crit = 90</b></p>

Le service suivant supervise l'objet *Processus (Ax32Serv)* et son compteur de performance *% temps processeur* (Figure 35):

```

define service{
    host_name                WIN-03-AX
    service_description      PROCESS_AX32SERV_PROCESSOR_TIME
    _SERVICE_ID             64
    use                      generic-service
    check_command            check_counter!12489!centreon!Processus(Ax32Serv)!% temps processeur!70!90
    max_check_attempts      5
    normal_check_interval    10
    retry_check_interval     10
    check_period             24x7
    notification_interval   10
    notification_period      24x7
    notification_options     w,u,c,r,f,s
    contacts                 Software_Software
}

```

**Figure 35 - Définition du service PROCESS\_AX32SERV\_PROCESSOR\_TIME dans le fichier services.cfg**

Voici le résultat dans l'interface Centreon (Figure 36) :

<input type="checkbox"/>	PHYSICALDISK_READ		OK	17h 2m 13s	17/11/2014 13:39:46	1/5 (H)	0
<input type="checkbox"/>	PHYSICALDISK_WRITE		OK	17h 6m 28s	17/11/2014 13:35:32	1/5 (H)	0
<input type="checkbox"/>	PROCESS_AX32SERV_PRIVILEGED_TIME		OK	17h 5m 50s	17/11/2014 13:36:08	1/5 (H)	0
<input type="checkbox"/>	PROCESS_AX32SERV_PROCESSOR_TIME		OK	17h 5m 13s	17/11/2014 13:36:45	1/5 (H)	0

**Figure 36 - Visualisation du service PROCESS\_AX32SERV\_PROCESSOR\_TIME**

- Définition de service pour le compteur *SQLServer:Buffer Manager\Free list stalls/sec*

**Tableau XX - Extrait du tableau Compteurs communs pour SQL Server**

OBJET	COMPTEUR	UTILISATION
SQLServer: Buffer Manager	Free list stalls/sec	<p>Ce compteur mesure le nombre de requêtes par seconde qui ont dû attendre pour une page libre. S'il n'y a pas de pages libres dans le cache buffer, une requête doit attendre qu'une d'entre elle se libère.</p> <p><b>Seuil</b> : la valeur recommandée doit être inférieure à 2. Si la valeur est supérieure, vérifier les compteurs <b>Page Life Expectancy/sec</b> et <b>Lazy Writes/sec</b>. Dans le cas où <b>Page Life Expectancy</b> est inférieur à 300/sec et <b>Lazy Writes</b> au-dessus de 2/sec, alors il existe un signe clair de pression de mémoire.</p> <p><b>Exemples</b> :</p> <p><b>Warn = 2</b></p> <p><b>Crit = 4</b></p>

Le service suivant supervise l'objet *MSSQL\$SIGF\_AX\_HPR:Buffer Manager* instance *SIGF\_AX\_HPR* et son compteur de performance *Free list stalls/sec* (Figure 37):

```

define service{
    host_name                WIN-01-AX
    service_description      MSSQL_FREE_LIST_STALLS
    _SERVICE_ID             71
    use                      generic-service
    check_command            check_counter!12489!centreon!MSSQL$SIGF_AX_HPR:Buffer Manager!Free
list stalls/sec!2!4
    max_check_attempts      5
    normal_check_interval   10
    retry_check_interval    10
    check_period            24x7
    notification_interval   10
    notification_period     24x7
    notification_options    w,u,c,r,f,s
    contacts                Software_Software
}

```

**Figure 37 - Définition du service MSSQL\_FREE\_LIST\_STALLS dans le fichier services.cfg**

Voici le résultat dans l'interface Centreon (Figure 38) :

<input type="checkbox"/>	MSSQL_FREE_LIST_STALLS		OK	2m 53s	17/11/2014 13:39:50	1/5 (H)	0
<input type="checkbox"/>	PHYSICALDISK_READ		OK	17h 7m 28s	17/11/2014 13:35:15	1/5 (H)	0
<input type="checkbox"/>	PHYSICALDISK_WRITE		OK	17h 6m 50s	17/11/2014 13:35:54	1/5 (H)	0
<input type="checkbox"/>	PROCESS_SQLSERV_PRIVILEGED_TIME		OK	16h 46m 13s	17/11/2014 13:36:30	1/5 (H)	0
<input type="checkbox"/>	PROCESS_SQLSERV_PROCESSOR_TIME		OK	17h 5m 35s	17/11/2014 13:37:08	1/5 (H)	0

**Figure 38 - Visualisation du service MSSQL\_FREE\_LIST\_STALLS dans Centreon**

### 3) Distinction entre plusieurs instances dans le cas d'un cluster de serveurs à superviser

Pour choisir une instance en particulier, il faut juste indiquer à la suite de l'objet et en parenthèse le numéro de l'instance comme par exemple :

<i>Microsoft Dynamics AX Object Server(01) ou Microsoft Dynamics AX Object Server(02)</i> <i>Processeur(0) ou Processeur(3)</i> <i>Disque Physique(0 C :) ou Disque Physique(1 D :)</i>
---

Parfois, pour certains objets, on peut vouloir superviser la totalité des instances en même temps. Il faut alors ajouter entre parenthèse la syntaxe suivante (*\_TOTAL*) comme par exemples :

<i>Processeur(_TOTAL)</i> <i>Disque Physique(_TOTAL)</i>
---

Après avoir exposé ces différents exemples d'implémentation dans Centreon, je vais maintenant livrer les résultats de ce projet de supervision de Dynamics AX.

#### *10.3.3. Les résultats de ce projet*

J'ai rencontré évidemment certaines difficultés lors de ce projet. Tout d'abord, il a fallu ne pas se perdre dans la jungle des nombreux compteurs de performance qui existent. De plus, après leur sélection, il a été nécessaire de chercher et analyser des seuils pour utiliser ces compteurs quand c'était possible. Enfin, j'ai créé des scripts et tester leur implémentation dans la solution Nagios-Centreon. La prise en main de ce produit de supervision ne s'est en outre pas faite si facilement.

Néanmoins, le livrable du projet a été présenté sous forme de documentation pour le client. Il est désormais incorporer dans le DET (Document d'Exploitation) qu'on livre après chaque démarrage en production des clients Gfi-SSP. Mais il est évidemment encore possible d'améliorer cette mise en œuvre de la supervision dans Dynamics AX.

Effectivement, il faut penser à incorporer les indicateurs de performances d'autres composants Dynamics AX comme SSRS, SSAS ou encore SharePoint. Deuxièmement, il faut continuer à approfondir la question de la supervision de Dynamics par les journaux Windows et notamment la supervision du Workflow. Le manque de temps et la demande cadrée du

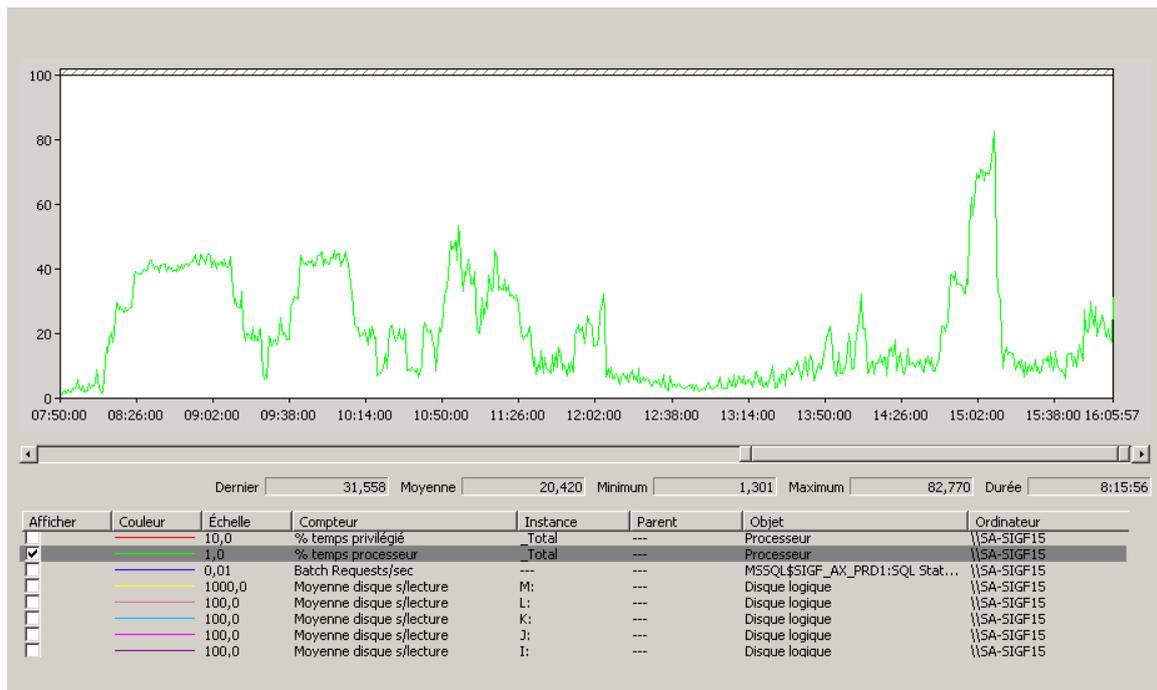
client ne m'ont pas permis d'explorer ces pistes dans cette première version. Cependant, ce premier travail fournit une base qui sera alimentée au fur et à mesure des demandes.

En complément à cette mission, j'ai été aussi confronté chez un autre client à une autre dimension de la problématique de supervision de Dynamics AX. En effet, juste après le passage en production de ce client, il a été question de mener la supervision de leur plateforme. C'est donc l'objet du point suivant.

#### **10.4. La supervision de Dynamics AX en production**

Lors du passage en production d'un de nos clients, au sein d'une équipe projet Gfi-SSP, j'ai été impliqué dans le suivi quotidien des performances de leur plateforme Dynamics AX. La mission a duré environ trois semaines sur site entre janvier et février 2015. Le besoin du client était d'obtenir un rapport quotidien détaillé afin de connaître l'évolution de son nouveau système. En partant de mes travaux réalisés sur le sujet, j'ai donc décidé d'insérer les principaux compteurs de performance dans l'outil Windows PerfMon. Le choix de cette solution standard de Microsoft s'explique par le fait que le client ne souhaitait pas investir dans un produit de supervision et n'avait pas encore les ressources nécessaires pour exploiter sa nouvelle plateforme.

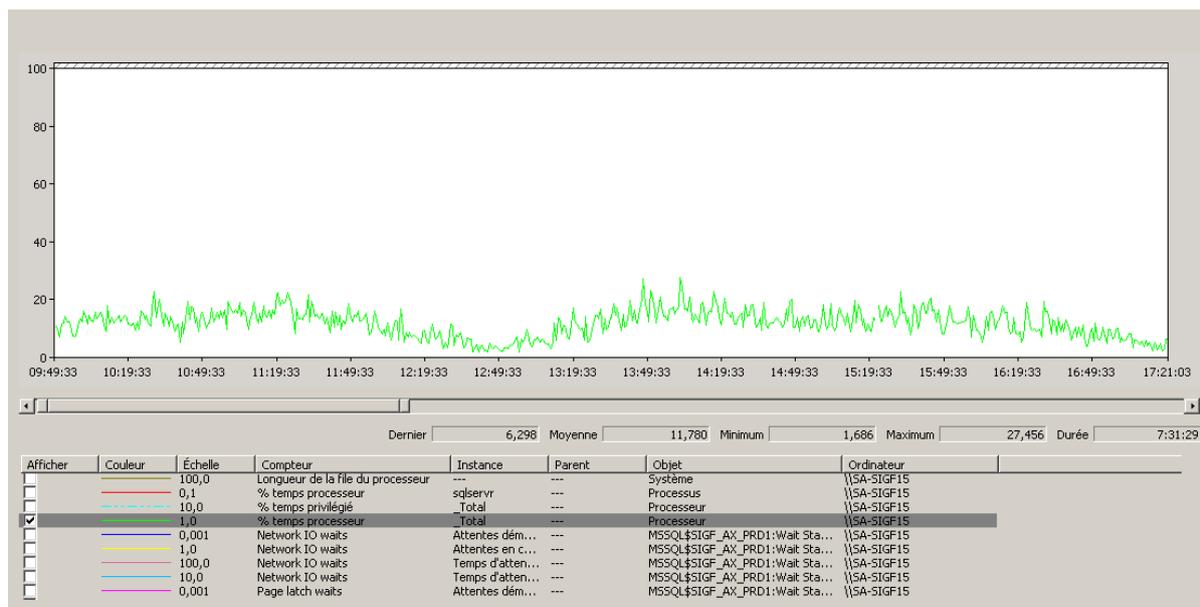
Sur leur serveur de base de données, la mise en place de la supervision m'a ainsi permis de découvrir une suractivité avec des pics anormaux (à plus de 80%) concernant la charge CPU (Figure 39). En parallèle, des lenteurs sur certains formulaires de l'application Dynamics AX ont été remontées par les utilisateurs et par les consultants fonctionnels dans des fiches incidents.



**Figure 39 - %temps processeur sur la machine de base de données le 28/01/2015**

Suite à une réunion avec le client, j'ai donc proposé au client deux actions urgentes afin de déterminer les raisons de ce problème. Premièrement, renforcer la supervision en ajoutant une vingtaine de compteurs de performances sur le serveur SQL Server (écriture et lecture disque, statistiques d'attente, verrous, transactions sur les bases de données et la gestion tampon). Deuxièmement, vérifier les requêtes exécutées par Dynamics AX dans SQL Server lors de la montée en pic de la charge de CPU avec un deuxième outil de supervision : SQL Server Profiler. Ce dernier est une interface puissante où les événements sont enregistrés dans un fichier de trace qui peut être analysé ou utilisé ultérieurement pour relire une série d'étapes spécifique lors d'une tentative de diagnostic d'un problème.

Ce travail a eu comme résultat de montrer que certaines requêtes particulières généraient des temps d'attente CPU anormalement long (de l'ordre parfois de la demi-heure). En parallèle, dans l'application, l'utilisateur subissait des lenteurs sur son poste. J'ai alors transmis les requêtes consommatrices à l'équipe de développement pour analyse et optimisation de celles-ci. Une fois le correctif livré (ajout d'index supplémentaires), les pics de montée en charge du CPU ont alors ensuite totalement disparu comme l'atteste la figure suivante (Figure 40).



**Figure 40 - %temps processeur sur la machine de base de données le 10/02/2015**

La mise en place de la supervision chez ce client a démontré en contexte réel l'utilité du travail que j'avais donc effectué en amont dans la documentation. Effectivement, j'ai pu tout de suite proposer des actions pour résoudre un problème de performance en situation de production. Partant de ce fait, il est possible de souligner que ce projet de supervision de Dynamics AX est une contribution importante à l'amélioration de la qualité de notre processus d'intégration. Il nous sert maintenant de base pour tous les projets avenir et permet de donner aux clients une visibilité sur cette problématique. La prochaine étape peut-être de construire une base de connaissances sur les différentes supervisions réalisées sur nos clients en production afin de pouvoir affiner les seuils d'alertes proposés. Enfin, du point de vue collaboration, cela fut très riche car j'ai pu ainsi aider les équipes des consultants et des développeurs dans la résolution en urgence d'un problème critique.

Dans cette troisième partie, j'ai donc présenté les trois grands projets qui m'ont été confiés. A chaque fois, j'ai procédé avec la même méthode. En effet, premièrement, commencer à comprendre la situation initiale et les objectifs attendus. Ensuite, identifier le périmètre des projets pour me focaliser sur les problèmes posés par la mission. Puis, établir un choix de solutions pour résoudre les difficultés. Enfin, mettre en place une solution et observer les résultats obtenus. C'est cette démarche qui m'a ainsi permis d'atteindre les objectifs de ces projets avec succès et dans les délais fixés.

## Conclusion

Accomplie pendant six mois au sein de la division SSP, cette mission d'ingénierie est donc couronnée de succès. En effet, la qualité du processus d'intégration technique de la solution Gfi-SSP a subi des améliorations notables dans les différents projets que j'ai dirigé et cela pour plusieurs raisons.

Tout d'abord, grâce à cette étude menée sur les problématiques posées par l'intégration, nous sommes maintenant capables de concevoir une architecture technique plus fiable chez nos clients. Effectivement, nous avons une meilleure compréhension des différents enjeux posés par l'implémentation de l'ERP Dynamics AX. Par conséquent, cela nous permet d'avoir une plus large visibilité du processus et d'améliorer par la même occasion nos compétences liées à d'autres besoins clients.

Avec un nombre de clients qui croît et par conséquent un nombre de plateformes techniques en augmentation, l'apport de mon travail est aussi visible dans le perfectionnement de l'industrialisation des mises à jour des environnements. La nouvelle organisation des répertoires et le développement de nouveaux scripts en langage objet permettent assurément de travailler dans de meilleures conditions. En effet, cela se traduit pour notre équipe par une maintenance allégée sur les scripts, une automatisation des tâches redondantes ou encore par une optimisation du délai passé aux opérations de livraison. Pour les clients, l'exploitation de leurs plateformes est désormais simplifiée.

La qualité a aussi sensiblement progressée avec la possibilité maintenant de proposer à nos clients de véritables mécanismes de haute disponibilité avec la solution Gfi-SSP. C'est un sujet où mon apport a été encore une fois déterminant. A présent, nous disposons effectivement d'une meilleure connaissance sur ce thème par type de composants Dynamics AX. De plus, la création d'une typologie nous donne un cadre de base pour définir au plus juste la HA avec nos clients.

Enfin, nous sommes dorénavant dans la capacité de pouvoir superviser l'ensemble des éléments des environnements Dynamics AX. La maîtrise de ce sujet permet ainsi d'établir des systèmes de supervision proactifs chez nos clients. En effet, il est possible d'assurer la

surveillance du système de gestion financière en émettant des alertes avant même qu'un problème grave ne survienne. Notre processus d'intégration ainsi que nos livrables ont donc incontestablement augmenté en valeur ajoutée.

Parallèlement, en ce qui me concerne, c'est également une mission riche de nouvelles expériences. J'ai d'abord pris conscience du travail et de l'activité d'un ingénieur. J'ai pu jouer un rôle moteur dans les projets menés, assurer le respect de leurs délais et la qualité des différents livrables. Je me suis aussi familiarisé avec un nouveau domaine : l'ERP Microsoft Dynamics AX. J'ai ainsi enrichi mes connaissances avec les différents composants Microsoft nécessaires au déploiement. J'ai compris également l'importance de l'analyse, de l'organisation et de la rigueur nécessaires dans le bon accomplissement des projets.

En outre, il est indéniable que mon intégration dans l'équipe PIT s'est déroulée de la manière la plus réussie. Mon tuteur m'a effectivement donné la confiance nécessaire pour gérer tout seul les projets initiés. J'ai pu de cette manière prouver mes capacités à raisonner sur les différentes problématiques rencontrées en déroulant une méthodologie claire et préétablie. J'ai su être également force de proposition et de décision à certains moments importants des projets. J'ai pu aussi convaincre notre équipe des solutions proposées toujours en motivant, expliquant et justifiant mes choix. Je me suis aussi bien intégré dans les équipes projets Gfi-SSP sur site client qui a été évidemment bénéfique sur tous les points. J'ai pu ainsi apporter mon aide, ma disponibilité et mes conseils. Mais, j'ai su aussi écouter les consultants et les développeurs pour bien réussir les diagnostics des problèmes soulevés. Enfin, j'ai pu démontrer une certaine réactivité chez les clients dans la résolution de leurs difficultés.

Certes, gérer une équipe ne faisait pas partie des attributions de la mission car la structure même de cette dernière était composée de deux personnes uniquement. Cependant, je pense que cette expérience m'a donnée d'abord une importante visibilité sur l'allocation des ressources et la gestion de la charge de travail liée aux projets mais aussi une sérieuse préparation pour le futur rôle de chef d'équipe. Il est en tout cas clair, qu'après avoir réalisé cette mission d'ingénieur, je me sens conforté dans le choix des métiers de la branche Software ERP dans l'entreprise Gfi Informatique. L'activité connaît une belle croissance comme le montre les nouveaux projets gagnés récemment. Je souhaite donc poursuivre dans cette voie en étant un élément innovateur et décisif.

## Annexes

**Tableau XXI - Le rôle de l'Architecte Dynamics d'après Microsoft Sure Step**

<b>FICHE ARCHITECTE SOLUTION DYNAMICS</b>	
<b>DEFINITION</b>	
Fournir à la fois des connaissances techniques et fonctionnelles sur la manière d'implémenter Dynamics et sur toutes les technologies connexes Microsoft nécessaires à l'implémentation.	
<b>DESCRIPTION</b>	
<p>Etablir une vue d'ensemble des exigences définies par le client ; mener la définition des besoins techniques et d'architecture du business et identifier toutes les incohérences, hypothèses et éléments essentiels qui doivent être considérés dans une solution d'entreprise.</p> <p>Définir le plan de la solution technique Dynamics inclus dans la proposition cliente ; participer avec les spécialistes techniques et fonctionnels dans l'analyse des écarts.</p> <p>Démontrer de l'expérience dans l'utilisation de méthodes communes d'implémentation de logiciels ; comprendre et appliquer les frameworks et les techniques de résolution de problème relatif à l'engagement client et des projets internes.</p> <p>Contribuer au « scope management », management du risque, ce qui signifie identification, évaluation et plan de réduction.</p> <p>Aider à conduire à l'estimation de l'effort pour réaliser l'implémentation de la solution proposée.</p> <p>Interagir avec les parties prenantes pour fournir le point de vue de l'ingénierie sur la vision utilisateur des structures et fonctions du système ainsi que dans les domaines du système capables d'évoluer.</p> <p>Développer la stratégie d'intégration, suivant les concepts fondamentaux de l'architecture et en conformité avec les exigences et les contraintes du client ; veiller que les solutions Microsoft et les sources de données client sont correctement intégrés.</p> <p>Planifier et superviser l'exécution de test en collaboration avec le chef de projet et recueillir les commentaires des premiers utilisateurs de la solution.</p> <p>Démontrer une importante base de connaissances sur les questions actuelles et tendances, les facteurs clés, et leur impacts.</p> <p>Démontrer une aptitude à planifier et gérer les engagements pris et d'allouer les ressources de manière appropriée dans les contraintes de délais, budget et qualité.</p> <p>Travailler en collaboration avec l'équipe de direction afin de développer et d'améliorer les solutions « best practices », outils et méthodes.</p>	
<b>COMPETENCES</b>	
<b>Capacités requises</b>	Appliquer l'architecture solution de haut niveau et la conception.
	Analyser, comprendre et disséquer les processus business.
	Analyser l'infrastructure informatique d'une organisation.
	Appliquer efficacement les techniques d'intégration Microsoft.
	Utilisez les techniques de modélisation de processus et de données.
<b>Expériences confirmées</b>	Effectuer le calibrage du matériel basé sur l'architecture.
	Mise en œuvre méthodologie.
	.NET, Visual Studio, environnements de développement spécifiques Microsoft Dynamics.
	Mise en place de réseau local et étendu.
	Systèmes d'exploitation serveur et bureau.
<b>ACTIVITES SELON LES PHASES</b>	
<b>Diagnostic</b>	Analyse support et planification du système business et solutions proposées.
	Examen et présentation des résultats de la phase.
	Définir la licence support et les estimations des services pour les tâches techniques.
	Estimation du service support pour les tâches de développement
<b>Analyse</b>	Analyse support des interfaces entre les systèmes existants et les solutions tierces.
	Validation de l'infrastructure et mise à jour de la conception.
	Examen et présentation des résultats de la phase.
<b>Conception</b>	Validation de la sécurité de la conception
	Soutien de la création de la documentation de la conception pour la compréhension du client
	Examen et présentation de la documentation et la conception de l'application aux clients.
	Examen des configurations systèmes.
<b>Développement</b>	Examen des résultats du développement.
	Validation de la sécurité.
<b>Déploiement</b>	Examen de la préparation du déploiement.
<b>Optimisation</b>	Examen des technologies existantes afin de déterminer de possibles améliorations.
<b>Mise à jour</b>	Validation que la structure actuelle du système prend en charge la nouvelle version de Dynamics.

**Tableau XXII - Le rôle du consultant technique Dynamics d'après Microsoft Sure Step**

<b>FICHE CONSULTANT TECHNIQUE DYNAMICS</b>	
<b>DEFINITION</b>	
Fournir des connaissances techniques de l'infrastructure informatique et de SGBD nécessaire au bon fonctionnement de Dynamics. Fournir une expertise sur les outils Dynamics utilisés pour la migration des données et l'intégration ainsi que la connaissance des technologies Microsoft connexes.	
<b>DESCRIPTION</b>	
Fournir des conseils dans les décisions commerciales d'un point de vue technique (performance, évolutivité, fiabilité et sécurité). Définir le plan de la solution technique Dynamics inclus dans la proposition cliente ; participer avec les spécialistes techniques et fonctionnels dans l'analyse des écarts. Aider le client dans la mise à jour matériel et logiciel afin de répondre aux exigences d'installation de Dynamics. Créer les rapports de test de validation des résultats, précision, fiabilité et conformité aux normes. Exécuter la planification et la mise en œuvre des tests de système pour assurer la fonctionnalité, la configuration et la conformité des données aux spécifications et aux normes. Effectuer ou soutenir la formation du personnel technique. Installer et configurer les environnements Dynamics.	
<b>COMPETENCES</b>	
<b>Expériences confirmées</b>	Architecture et dimensionnement du matériel Dynamics.
	Intégration de la plateforme Microsoft et de Dynamics.
	Mise en œuvre méthodologie.
	.NET, Visual Studio, environnements de développement spécifiques Microsoft Dynamics.
	Mise en place de réseau local et étendu.
	Systèmes d'exploitation serveur et bureau.
	Technologies serveurs : SQL Server database, Reporting Services et Analysis Services, Serveurs Web,... Technologies Internet : Internet Information Server (IIS), Microsoft BizTalk Server, firewalls,...
<b>ACTIVITES SELON LES PHASES</b>	
<b>Diagnostic</b>	Préparation et analyse de l'infrastructure et de sa conception.
	Analyse des interfaces entre les systèmes existants et les autres solutions tierces.
	Documentation et présentation des résultats de la phase Diagnostic.
	Définition de la licence support et estimation des services pour les tâches techniques.
<b>Analyse</b>	Formation des utilisateurs techniques clés.
	Analyse des interfaces entre les systèmes existants et les solutions tierces.
	Analyse et cartographie de la migration des données prévues.
	Validation de l'infrastructure et mise à jour de la conception.
<b>Conception</b>	Documentation et présentation des résultats de la phase Analyse.
	Création des spécifications de conception de migration des données.
	Validation de la sécurité de la conception.
	Création des documents de conception pour la compréhension du client et son acceptation.
<b>Développement</b>	Présentation de la documentation et de la conception de l'application au client.
	Développement de la migration des données.
	Test de la migration des données.
<b>Déploiement</b>	Validation de la sécurité.
	Organisation de la formation aux utilisateurs techniques clés (dans le cas d'une implémentation rapide)
	Planification de la migration des données (dans le cas d'une implémentation rapide)
	Configuration de l'environnement production et de l'environnement test pour le test final du système.
	Test final de la migration des données.
<b>Exploitation</b>	Préparation et réalisation des tests de charge.
	Support post-démarrage.
	Résoudre les questions en suspens.
<b>Optimisation</b>	Finalisation de la documentation.
	Optimisation de l'infrastructure.
<b>Mise à jour</b>	Amélioration de la performance.
	Examen de l'infrastructure et des logiciels.
	Evaluation des personnalisations de mise à jour nécessaires pour les nouvelles versions de Dynamics.

Figure 41 - Ancien script utilisé pour la copie entre environnement

```
#####
# Script permettant de faire un backup/restore d'une base de données d'un serveur SQL source vers
# un serveur SQL destination.
# NB: Utiliser le script d'export des paramètres pour remplir les informations.
# Auteur : ZAa
# Créé le: 05/02/2014 par ZAa :
# Modifié le : XX/XX/XXXX par XXX: Commentaire
# GFI Progiiciels
#####

param(
[string] $Logfolder = "F:\copyenv\PRP-PRD\logs",
[string] $SQLInstanceSourceModel = "SA-SIGF15\SIGF_AX_PRD1",
[string] $DataBaseSourceModel = "SIGF_AXDB_PROD_model",
[string] $BackupPathModel = "\\10.200.6.45\BackupTemp\SIGF_AXDB_PROD_model_TEMP.bak",
[string] $BackupDescriptionModel = "SIGF_AXDB_PROD_model-Complète Base de données Sauvegarde",
[string] $SQLInstanceDestModel = "RW200105\SIGF_AX6R2_INT",
[string] $DataBaseDestModel = "SIGF_AXR2_PRP_model",
[string] $DataBaseMdfPathModel = "F:\SIGFINST_AX6R2_INT\MSSQL10_50.SIGFINST_AX6R2_INT\MSSQL\DATA\SIGF_AXR2_PRP_model.mdf",
[string] $DataBaseLdfPathModel = "E:\SIGFINST_AX6R2_INT\MSSQL10_50.SIGFINST_AX6R2_INT\MSSQL\Data\SIGF_AXR2_PRP_model_log.LDF",
[string] $DataBaseMdfNameModel = "SIGF_AXDB_PROD_model",
[string] $DataBaseLdfNameModel = "SIGF_AXDB_PROD_model_log",

[string] $SQLInstanceSourceTransac = "SA-SIGF15\SIGF_AX_PRD1",
[string] $DataBaseSourceTransac = "SIGF_AXDB_PROD",
[string] $BackupPathTransac = "\\10.200.6.45\BackupTemp\SIGF_AXDB_PROD_TEMP.bak",
[string] $BackupDescriptionTransac = "SIGF_AXDB_PROD-Complète Base de données Sauvegarde",
[string] $SQLInstanceDestTransac = "RW200105\SIGF_AX6R2_INT",
[string] $DataBaseDestTransac = "SIGF_AXR2_PRP",
[string] $DataBaseMdfPathTransac = "F:\SIGFINST_AX6R2_INT\MSSQL10_50.SIGFINST_AX6R2_INT\MSSQL\DATA\SIGF_AXR2_PRP.mdf",
[string] $DataBaseLdfPathTransac = "E:\SIGFINST_AX6R2_INT\MSSQL10_50.SIGFINST_AX6R2_INT\MSSQL\Data\SIGF_AXR2_PRP_log.LDF",
[string] $DataBaseMdfNameTransac = "SIGF_AXDB_PROD",
[string] $DataBaseLdfNameTransac = "SIGF_AXDB_PROD_log",
[string] $T_SSAS_DATASOURCE_NAME = "SIGF_AXR2_PRP",
[string] $T_SSAS_CONNECTIONSTRING = "Provider=SQLNCL10.1;Data Source=RW200105;Integrated Security=SSPI;Initial Catalog=SIGF_AXR2_PRP",
[string] $T_SSAS_LOGFILEPATH = "",
[string] $T_SSAS_PROJECTFILENAME = "",
[string] $T_SSAS_ANALYSISSEVICEDATABASENAME = "DynAX_PRP",
[string] $T_HELP_URLHelp = "http://rw200106:8081/DynamicsAX6HelpServer/HelpService.svc",
[string] $T_AIF_VirtualDirectoryShare = "{Destination AIF VirtualDirectoryShare}",
[string] $T_AIF_URL = "{Destination AIF URL}",
[string] $T_WF_Account_SID = "S-1-5-21-1708537768-1659004503-839522115-72319",
[string] $T_WF_Account_NETWORKALIAS = "ADMSIGF_WF_INT",
[string] $T_SMTMP_ATTACHMENTPATH = "",
[string] $T_SMTMP_SMTMPRelayServerName = "",
[string] $T_SMTMP_PortNumber = "",
[string] $T_SMTMP_ServerIPAddress = "",
[string] $T_SMTMP_UserName = "",
[string] $T_SMTMP_DNSServerName = "",
[string] $T_DOC_SysFileStoreParameters = "",
[string] $T_SharePoint_HOMEPAGESITEID = "b69aa5eb-49a3-43b7-ac33-ba89b128b890",
[string] $T_SharePoint_DEVELOPMENTSITEID = "b69aa5eb-49a3-43b7-ac33-ba89b128b890",
[string] $T_SharePoint_EXTERNALURL = "http://rw200106:8189/sites/DynamicsAxPrp",
[string] $T_SharePoint_INTERNALURL = "http://rw200106:8189/sites/DynamicsAxPrp",
[string] $T_SharePoint_SITEID = "b69aa5eb-49a3-43b7-ac33-ba89b128b890",
[string] $T_SharePoint_COMPANYID = "M831",
[string] $T_DMP_SHARED_FOLDERPATH = "",
[string] $T_Security_AosAccountDest = "DMNCNFPT\ADMSIGF_AOS_INT",
[string] $T_Security_BcpAccountDest = "DMNCNFPT\ADMSIGF_BC_INT"
)

."F:\copyenv\PRD-PRP\COMMON\BackupRestore.ps1"
."F:\copyenv\PRD-PRP\COMMON\ModifyDatabaseSecurity.ps1"
."F:\copyenv\PRD-PRP\COMMON\COPYSourceDest.ps1"
```

```

#####
# Traitement principal - DEBUT
#####
# get-service winrm
# Enable-PSRemoting -force
# winrm s winrm/config/client '@(TrustedHosts="GFITESTAOS")'

# winrm quickconfig

# Write-Host "Arrêt AOS - DEBUT"
# Stop-Service 'AOS60402'
# Write-Host "Arrêt AOS - FIN"

# Copie de la base des modèles
CopyDatabase -Logfolder "1" -SQLInstanceSource $SQLInstanceSourceModel -DataBaseSource $DataBaseSourceModel -BackupPath $BackupPathModel `
-BackupDescription $BackupDescriptionModel -SQLInstanceDest $SQLInstanceDestModel -DataBaseDest $DataBaseDestModel `
-DataBaseMdfName $DataBaseMdfNameModel -DataBaseLdfName $DataBaseLdfNameModel -DataBaseMdfPath $DataBaseMdfPathModel -DataBaseLdfPath $DataBaseLdfPathModel

# Copie de la base transactionnelle
CopyDatabase -Logfolder "2" -SQLInstanceSource $SQLInstanceSourceTransac -DataBaseSource $DataBaseSourceTransac -BackupPath $BackupPathTransac `
-BackupDescription $BackupDescriptionTransac -SQLInstanceDest $SQLInstanceDestTransac -DataBaseDest $DataBaseDestTransac `
-DataBaseMdfName $DataBaseMdfNameTransac -DataBaseLdfName $DataBaseLdfNameTransac -DataBaseMdfPath $DataBaseMdfPathTransac -DataBaseLdfPath $DataBaseLdfPathTransac

# Modification de la sécurité après la restauration
ModifyDatabaseSecurityModelStore -Logfolder $Logfolder -SQLInstanceDest $SQLInstanceDestModel `
-AOSName $T_AOSName -AosAccountDest $T_Security_AosAccountDest `
-BcpAccountDest $T_Security_BcpAccountDest
ModifyDatabaseSecurityOLTP -SQLInstanceDest $SQLInstanceDestTransac -DataBaseDest $DataBaseDestTransac `
-aosServerName $T_aosServerName -aosInstance $T_aosInstance `
-AX_Database $DataBaseDestTransac `
-AX_BaseLineDatabase $DataBaseDestBaseline `
-tempDBName $T_tempDBName -AosAccountDest $T_Security_AosAccountDest `
-createStoredProcedures $T_createStoredProcedures `
-BcpAccountDest $T_Security_BcpAccountDest -Logfolder $Logfolder

# Traitement Post Backup/restore
CopySourceDestination -LogFolder $LogFolder -SQLInstance $SQLInstance `
-P_AOS $P_AOS `
-P_ClusterNameAOS $P_ClusterNameAOS `
-P_BatchAOS $P_BatchAOS `
-P_SQL $P_SQL `
-P_SSRS $P_SSRS `
-P_BC_Account $P_BC_Account `
-T_DBName $T_DBName `
-T_SchemaDB $T_SchemaDB `
-T_AOS $T_AOS `
-T_ClusterNameAOS $T_ClusterNameAOS `
-T_BatchAOS $T_BatchAOS `
-T_SQL $T_SQL `
-T_SSRS_SERVERID $T_SSRS_SERVERID `
-T_SSRS_SERVERURL $T_SSRS_SERVERURL `
-T_SSRS_REPORTMANAGERURL $T_SSRS_REPORTMANAGERURL `
-T_SSRS_SERVERINSTANCE $T_SSRS_SERVERINSTANCE `
-T_SSRS_AOSID $T_SSRS_AOSID `
-T_SSRS_CONFIGURATIONID $T_SSRS_CONFIGURATIONID `
-T_BC_SID $T_BC_SID `
-T_BC_Account $T_BC_Account `
-T_USER_List $T_USER_List `
-T_SID_Admin $T_SID_Admin `
-T_NETWORKALIAS_Admin $T_NETWORKALIAS_Admin `
-T_SSAS_SRVNAME $T_SSAS_SRVNAME `
-T_SSAS_ANALYSISSEVICESDATABASENAME $T_SSAS_ANALYSISSEVICESDATABASENAME `
-T_HELP_URLHelp $T_HELP_URLHelp `
-T_AIF_VirtualDirectoryShare $T_AIF_VirtualDirectoryShare `
-T_AIF_URL $T_AIF_URL `
-T_WF_Account_SID $T_WF_Account_SID `
-T_WF_Account_NETWORKALIAS $T_WF_Account_NETWORKALIAS `
-T_SMTP_ATTACHMENTPATH $T_SMTP_ATTACHMENTPATH `
-T_SMTP_SMTPRelayServerName $T_SMTP_SMTPRelayServerName `
-T_SMTP_PortNumber $T_SMTP_PortNumber `
-T_SMTP_ServerIPAddress $T_SMTP_ServerIPAddress `
-T_SMTP_UserName $T_SMTP_UserName `
-T_SMTP_DNSServerName $T_SMTP_DNSServerName `
-T_DOC_SysFileStoreParameters $T_DOC_SysFileStoreParameters `
-T_SharePoint_HOMEPAGESITEID $T_SharePoint_HOMEPAGESITEID `
-T_SharePoint_DEVELOPMENTSITEID $T_SharePoint_DEVELOPMENTSITEID `
-T_SharePoint_EXTERNALURL $T_SharePoint_EXTERNALURL `
-T_SharePoint_INTERNALURL $T_SharePoint_INTERNALURL `
-T_SharePoint_SITEID $T_SharePoint_SITEID `
-T_SharePoint_COMPANYID $T_SharePoint_COMPANYID

Write-Host "Démarrage AOS - DEBUT"
# Start-Service 'AOS60402'
Write-Host "Démarrage AOS - FIN"

#iisreset /noforce

#AxUpdatePortal.exe -updatealll -websiteurl $T_SharePoint_EXTERNALURL

#####
# Traitement principal - FIN
#####

```

## Bibliographie

### ERP

J.-L. DEIXONNE. *Piloter un projet ERP*. Dunod, 286 p., 2011.

P. JOUFFROY. *ERP méthode pratique de mise en œuvre pour PME et PMI*. Eyrolles, 317 p., 2010.

J.-L. LEQUEUX. *Manager avec les ERP*. Eyrolles, 380 p., 2008.

J.-L. TOMAS. *ERP et PGI sélection, méthodologie de déploiement et gestion du changement : les clés du succès, les facteurs de risques*. Dunod, 312 p., 2007.

J.-L. TOMAS, Y. GAL. *ERP et conduite des changements*. Dunod, 304 p., 2011.

### Microsoft Dynamics AX

*Microsoft Dynamics AX 2012 Installation Guide*. Microsoft Corporation, 330 p., 2014.

R. BADAUWY. Deploying customizations across Microsoft Dynamics environments. *White Paper*. Microsoft Press, 36 p., 2013.

K. DEFORCHE, K. SAELEN. *Microsoft Dynamics AX 2012 R2 Services*. Packt Publishing, 264 p., 2014.

### Méthodologie Sure Step

C. SHANKAR, V. BELLEFROID. *Microsoft Dynamics Sure Step 2010*. Packt Publishing, 360 p., 2011.

### Ressources sur Internet (consultées depuis le 09/09/2014)

Forums, blogs, articles et vidéos sur Microsoft Dynamics AX :

<https://community.dynamics.com/ax/default.aspx>

Bibliothèque technique sur Microsoft Dynamics AX 2012 :

<http://technet.microsoft.com/en-us/library/gg852966.aspx>

## Table des figures

Figure 1 - Offres proposées par la solution Gfi-SSP .....	16
Figure 2 - Organigramme du management de l'équipe Software SSP Dynamics AX .....	18
Figure 3 - Schéma des phases du projet Dynamics définies selon Sure Step adaptée de Shankar et Bellefroid (2011) .....	29
Figure 4 - Modèle 3-tiers de l'architecture Microsoft Dynamics AX.....	32
Figure 5 - Schéma des composants applicatifs de la solution Microsoft Dynamics AX.....	33
Figure 6 - Composants de la couche client.....	33
Figure 7 – Couche base de données .....	35
Figure 8 - Architecture de la sécurité dans l'ERP Dynamics AX .....	42
Figure 9 - Exemples de menu contextuel proposé par TortoiseSVN .....	53
Figure 10 - Site comparatif d'hébergement « Subversion (SVN) Hosting Comparison » .....	54
Figure 11 - Interface de Windows PowerShell ISE.....	55
Figure 12 - Schéma de l'ancien cycle de mise à jour des environnements AX.....	56
Figure 13 - Schéma du nouveau cycle de mise à jour des environnements AX.....	57
Figure 14 - Contenu du sous-dossier <i>COMMON</i> .....	58
Figure 15 - Contenu du dossier FRM-RCT .....	59
Figure 16 - Contenu du dossier FRM.....	59
Figure 17 - Contenu du dossier <i>2-Exploitation</i> .....	60
Figure 18 - Contenu du dossier <i>Livraison</i> .....	60
Figure 19 - Extrait du config.xml standard .....	62
Figure 20 - Extrait du diagramme de classe modélisant l'objet environnement .....	63
Figure 21 - Détails du répertoire <i>Framework</i> .....	64
Figure 22 - Détails du dossier <i>Main</i> .....	65
Figure 23 - Extrait de l'enchaînement actuel des opérations de copie d'environnement.....	67
Figure 24 - Organisation des tâches des équipes Gfi-SSP pour les livraisons critiques.....	72
Figure 25 - Schéma d'architecture de HA pour l'AOS avec deux répartiteurs de charge externes tiré de K. Deforche et K. Saelen (2014) .....	77
Figure 26 - Exemple de déploiement « scale-out » standard de SSRS tiré du site Technet ( <a href="https://msdn.microsoft.com/en-us/library/ms157293%28v=SQL.100%29.aspx">https://msdn.microsoft.com/en-us/library/ms157293%28v=SQL.100%29.aspx</a> ) .....	79
Figure 27 - Déploiement serveur « scale-out » (extensible à l'infini) avancé standard de SSRS tiré du site Technet ( <a href="https://msdn.microsoft.com/en-us/library/ms157293%28v=SQL.100%29.aspx">https://msdn.microsoft.com/en-us/library/ms157293%28v=SQL.100%29.aspx</a> ) .....	80
Figure 28 - Proposition d'architecture de HA pour Dynamics AX de premier niveau .....	83
Figure 29 - Proposition d'architecture de HA pour Dynamics AX de second niveau .....	84
Figure 30 - Proposition d'architecture de HA pour Dynamics AX de troisième niveau .....	86
Figure 31 - Architecture final de l'environnement en production du client avec la HA.....	89
Figure 32 - Commande <code>check_counter</code> décrite dans le fichier <code>checkcommands.cfg</code> .....	100
Figure 33 - Définition du service <code>AOS_ACTIVE_SESSIONS</code> dans le fichier <code>services.cfg</code> .....	101
Figure 34 - Visualisation du service <code>AOS_ACTIVE_SESSIONS</code> dans Centreon .....	101
Figure 35 - Définition du service <code>PROCESS_AX32SERV_PROCESSOR_TIME</code> dans le fichier <code>services.cfg</code> .....	102
Figure 36 - Visualisation du service <code>PROCESS_AX32SERV_PROCESSOR_TIME</code> .....	102
Figure 37 - Définition du service <code>MSSQL_FREE_LIST_STALLS</code> dans le fichier <code>services.cfg</code> .....	103
Figure 38 - Visualisation du service <code>MSSQL_FREE_LIST_STALLS</code> dans Centreon.....	103
Figure 39 - %temps processeur sur la machine de base de données le 28/01/2015 .....	106
Figure 40 - %temps processeur sur la machine de base de données le 10/02/2015 .....	107
Figure 41 - Ancien script utilisé pour la copie entre environnement .....	112

## Liste des tableaux

Tableau I - Composition et missions des équipes en charge des projets vertical Gfi-SSP.....	17
Tableau II - Projets Gfi-SSP par clients classés date de démarrage prévue en production .....	20
Tableau III - Enchaînement des tâches de notre équipe infrastructure PIT Dynamics AX.....	28
Tableau IV - Tableau des prérequis réseau par Microsoft pour Dynamics AX .....	37
Tableau V - Avantages et inconvénients des systèmes RAID retenus pour Dynamics AX.....	39
Tableau VI - IOPS nécessaire pour le stockage de Dynamics AX.....	40
Tableau VII - Exemple de planning des sauvegardes des bases chez un client .....	43
Tableau VIII - Taux de disponibilité théorique selon le type adapté du site <a href="http://technet.microsoft.com/en-us/library/cc748824%28v=office.15%29.aspx">http://technet.microsoft.com/en-us/library/cc748824%28v=office.15%29.aspx</a> .....	46
Tableau IX - Planification des tâches pour le projet d'industrialisation .....	52
Tableau X - Planification des tâches pour le projet HA .....	75
Tableau XI - Options de HA pour SQL Server supportées par Dynamics AX .....	78
Tableau XII - Options de HA pour SSAS .....	81
Tableau XIII - Planification des tâches pour le projet de supervision de Dynamics AX .....	91
Tableau XIV - Liste des solutions de supervision pour Dynamics AX.....	93
Tableau XV - Compteurs communs aux serveurs Windows.....	94
Tableau XVI - Compteurs de performance pour le composant AOS.....	96
Tableau XVII - Compteurs de performance pour SQL Server.....	98
Tableau XVIII - Extrait du tableau Compteurs pour le composant AOS.....	101
Tableau XIX - Extrait du tableau Compteurs communs pour les serveurs .....	102
Tableau XX - Extrait du tableau Compteurs communs pour SQL Server .....	103
Tableau XXI - Le rôle de l'Architecte Dynamics d'après Microsoft Sure Step.....	110
Tableau XXII - Le rôle du consultant technique Dynamics d'après Microsoft Sure Step.....	111

## Résumé

Au sein de la branche SSP (Solutions Secteur Public) de Gfi Informatique, j'ai effectué ma mission d'ingénieur en tant qu'architecte sur le produit Microsoft Dynamics AX. C'est dans ce contexte que j'ai alors été confronté aux problématiques d'intégration de la solution financière Gfi-SSP basée sur cet ERP. Mais, ce travail d'ingénierie a surtout consisté à rechercher comment augmenter la qualité dans notre intégration. En effet, avant mon arrivée, certaines activités importantes de ce processus manquaient de maîtrise et certaines opérations n'étaient pas optimisées. Je me suis donc impliqué à apporter premièrement une amélioration dans la connaissance des problématiques d'intégration. En second lieu, mes travaux ont porté sur la nécessité de perfectionner l'industrialisation des mises à jour des environnements clients. Ensuite, j'ai apporté une connaissance approfondie sur la question du déploiement de la haute disponibilité (HA) dans Dynamics AX qui garantit maintenant la qualité de notre offre technique. Enfin, suite à mon implication dans la supervision de Gfi-SSP, un pas majeur a été franchi permettant de contrôler les principaux composants de l'ERP. Par conséquent, l'ensemble de ces projets réalisés avec succès ont abouti à augmenter la valeur ajoutée de l'activité PIT (Pôle Intégration Technique) Dynamics AX.

Mots-clés : ERP, Microsoft Dynamics AX, Gfi-SSP, intégration, industrialisation, supervision, haute disponibilité.

---

## Summary

This document describes the work I performed in the SSP (Public Sector Solutions) branch of Gfi Informatique company. My job there as a software architect was to assist and manage the deployment of the financial solution proposed by Gfi-SSP based on Microsoft Dynamics AX, period during which I had to solve several problems concerning the integration process. My contribution was fourfold. First, I had to collect and analyze data, with a look on how to improve the quality of the integration. The main goal at this stage was to detect which activities lacked control and which were not optimized. Second, I participated in perfecting the industrialization of updates on customer environments based on aforementioned analysis. Third, I made a thorough study on the issue of deploying high availability (HA) in Dynamics AX which now guarantees the quality of our technical offer. Finally, following my involvement in overseeing Gfi-SSP, a major step has been taken to control the main components of the ERP. As a consequence of my work, our technical offer can now be mastered and deployed more easily. Also, my work increased the overall added value of the PIT Dynamics AX business in my enterprise.

Keywords : ERP, Microsoft Dynamics AX, Gfi-SSP, implementation, industrialization, monitoring, high availability.