



**HAL**  
open science

## Installation de la nouvelle version d'un logiciel bancaire

Vincent Augerd

► **To cite this version:**

Vincent Augerd. Installation de la nouvelle version d'un logiciel bancaire. Génie logiciel [cs.SE]. 2010. dumas-00524184

**HAL Id: dumas-00524184**

**<https://dumas.ccsd.cnrs.fr/dumas-00524184>**

Submitted on 7 Oct 2010

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

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

CONSERVATOIRE NATIONAL DES ARTS ET METIERS  
LYON

---

MEMOIRE  
PRESENTE EN VUE D'OBTENIR  
LE DIPLOME D'INGENIEUR CNAM  
EN  
INFORMATIQUE

PAR

VINCENT AUGERD

---

INSTALLATION DE LA NOUVELLE VERSION D'UN  
LOGICIEL BANCAIRE

SOUTENU LE 25/06/2010

---

JURY

PRESIDENT : Christophe PICOULEAU

MEMBRES : Bertrand DAVID  
Claude GENIER  
Jean-François PRINCE  
Florence JULLIEN



## Remerciements

En tout premier lieu je tiens à souligner l'importance du CNAM et des personnes qui y travaillent dans le système d'apprentissage français. Cette institution rend de précieux services et offre une formation de qualité reconnue au sein des entreprises.

Ensuite, je souhaite adresser mes remerciements les plus sincères aux personnes qui m'ont apporté leur aide tout au long de mes études et qui ont contribué à l'élaboration de ce mémoire et plus particulièrement :

- GCE Technologies, mon employeur, qui a insisté pour que je fasse mon mémoire d'ingénieur sur un projet de l'entreprise et notamment :
  - Stéphanie DUBREUIL qui a suivi mon dossier de près,
  - Jean-François PRINCE qui m'a affecté sur un projet éligible au mémoire,
  - Florence JULLIEN qui m'a fait confiance tout au long du projet
  - et Philippe GABILLON pour tous ses conseils avisés et sa bienveillance.
- Les membres du jury qui consacrent beaucoup de temps aux auditeurs du CNAM notamment lors des épreuves probatoires et de mémoire
- Ma famille qui m'encourage depuis le début, merci à ma fidèle lectrice
- Mon épouse, Coralie, qui aura tout fait pour que j'y arrive



# Table des Matières

<b>Introduction .....</b>	<b>7</b>
<b>PARTIE I CADRE DU PROJET.....</b>	<b>9</b>
1. GCE Technologies .....	9
1.1. GCE Technologies dans le groupe BPCE.....	9
1.2. PIA et financement.....	11
1.3. Son histoire récente : un système d'information unique.....	11
1.4. Son organisation .....	12
1.5. MySys, sa plateforme informatique .....	13
2. L'auditeur et GCE Technologies .....	14
2.1. La situation de l'auditeur dans l'organisation .....	14
2.2. Référent adjoint de la BAFI .....	15
3. Présentation de la BAFI .....	15
3.1. La BAFI .....	15
3.2. Sa mise en œuvre au sein des établissements financiers .....	17
3.3. Ses contraintes .....	17
4. La BAFI chez GCE Technologies.....	17
4.1. Organisation.....	17
4.2. Utilisation d'un progiciel : Evolan Report.....	19
4.3. Alimentation du logiciel Evolan Report.....	21
4.4. D'autres applications autour de Evolan Report .....	21
4.5. Principe de fonctionnement.....	22
4.6. Architecture technique .....	24
4.7. Architecture organisationnelle.....	25
4.8. Langages utilisés .....	26
4.9. Les environnements.....	26
5. Réforme de la BAFI : SURFI.....	27
5.1. Présentation de la réforme.....	27
5.2. Impacts dans les établissements financiers .....	28
5.3. Impacts pour GCE Technologies .....	28
5.4. Contraintes de la réforme.....	28
6. Conclusion .....	29
<b>PARTIE II LE PROJET .....</b>	<b>31</b>
1. Equipe projet et méthodologie .....	31
1.1. Découpage en chantiers .....	31
1.2. Les ressources intervenant sur le projet .....	31
1.3. Suivi de projet .....	33
1.4. Méthodologies et leur application au sein du projet .....	34
1.5. Planning du chantier Unix/Windows.....	38
2. Un environnement technique à construire.....	39
2.1. Mise en place de la configuration logicielle .....	39
2.2. Montage d'environnements de recette.....	42
3. Utilisation d'un nouveau langage : XBRL .....	44
3.1. Introduction au langage XBRL .....	44
3.2. Les enjeux du langage XBRL.....	46
3.3. Les contrôles.....	47
4. Des analyses et des solutions.....	47
4.1. La vision utilisateurs.....	48
4.2. Etude d'impact .....	48
4.3. Réorganisation des traitements.....	50
4.4. Nouveaux transferts de fichiers.....	56
4.5. Mémoire Java et des risques à éviter.....	60
5. La réalisation.....	62
5.1. Nouvelle organisation : identifier les acteurs.....	62
5.2. Transformer l'étude d'impact en plan d'action.....	64
5.3. Coordonner l'intervention des acteurs.....	66
5.4. Développer.....	67
5.5. Documenter .....	68
6. Les tests.....	69
6.1. Une procédure de test à mettre en place.....	70

6.2.	Tests et table de vérité .....	72
6.3.	Recette utilisateur et sa préparation.....	73
7.	<b>La mise en production .....</b>	<b>74</b>
7.1.	Sécurisation de la mise en production .....	74
7.2.	Plan de démarrage.....	76
8.	<b>Conclusion .....</b>	<b>77</b>
<b>PARTIE III SYNTHÈSE ET BILAN .....</b>		<b>79</b>
1.	<b>Synthèse des actions .....</b>	<b>79</b>
1.1.	Le point de départ : l'analyse .....	79
1.2.	Proposer des solutions et choisir .....	79
1.3.	Mobiliser et coordonner.....	80
1.4.	Tester.....	80
1.5.	Mise en production sécurisée.....	80
2.	<b>Difficultés rencontrées.....</b>	<b>81</b>
2.1.	Nouvelle organisation.....	81
2.2.	Mobilisation des acteurs.....	81
2.3.	Mise en référentiel non aboutie .....	81
3.	<b>Les succès .....</b>	<b>82</b>
3.1.	Respect des délais.....	82
3.2.	Méthodologie .....	82
3.3.	Procédure de test.....	82
3.4.	Responsabilité du chantier Unix/Windows .....	82
<b>Conclusion.....</b>		<b>83</b>
<b>Liste des figures.....</b>		<b>85</b>
<b>Glossaire .....</b>		<b>87</b>
<b>Références bibliographiques .....</b>		<b>89</b>
1.	Livres .....	89
2.	Revue et publications .....	89
3.	Sites internet .....	89
<b>Résumé .....</b>		<b>91</b>

## INTRODUCTION

La crise financière du deuxième semestre de l'année 2008 a révélé au grand public le rôle majeur que tiennent les établissements financiers dans la santé économique d'un pays. Au-delà des pays, l'économie mondiale est dépendante de la situation de ces établissements financiers. Mais bien avant cette crise, les autorités de tutelle ont mis en place des mécanismes pour surveiller et protéger la bonne santé du secteur bancaire.

La BAFI (Base des Agents Financiers) fait partie de l'arsenal mis en place pour jauger la situation économique des établissements financiers. Elle permet de présenter officiellement la situation financière et comptable aux autorités de tutelle (Banque de France et Banque Centrale Européenne). Mise en place en 1993, la BAFI fait l'objet d'une réforme sous l'impulsion du Secrétariat Général de la Commission bancaire ainsi que de la Banque de France ; cette réforme permettra le remplacement de la BAFI par SURFI (Système Unifié de Reporting Financier).

GCE Technologies qui développe et exploite le système d'information des Caisses d'Epargne joue un rôle important dans la mise en œuvre de la BAFI au sein du groupe BPCE (Banque Populaire et Caisse d'Epargne). BPCE a choisi le progiciel Evolan Report de la société Sopra pour la production des états réglementaires ; états qui sont ensuite envoyés aux autorités de tutelle. GCE Technologies exploite ce progiciel et l'alimente avec les données bancaires du système d'information mises au format attendu. Cette réforme qui permet le remplacement de la BAFI par SURFI concerne donc directement GCE Technologies.

Ce mémoire rend compte de notre travail dans le projet « SURFI, réforme de la BAFI », projet qui a pour objectif la mise en œuvre de la réforme SURFI au sein de GCE Technologies. Notre travail a été principalement axé sur l'installation de la nouvelle version du progiciel Evolan Report mais aussi sur les évolutions à apporter au socle technique UNIX.

Pour mieux comprendre dans un premier temps le cadre dans lequel s'est déroulé le projet, nous présentons l'entreprise GCE Technologies, puis notre situation et notre rôle dans cette entreprise. Ensuite nous abordons la BAFI de manière générale et sa mise en œuvre au sein de GCE Technologies. Pour terminer de poser le cadre, nous expliquons les enjeux de la réforme SURFI.

Dans un deuxième temps nous abordons le projet lui-même : son équipe et son environnement technique ainsi que l'utilisation d'un nouveau format d'échange d'informations financières : XBRL (eXtensible Business Reporting Language). Nous décrivons les phases d'analyse, de proposition de solution et de réalisation. Et enfin, les tests réalisés et la préparation à la mise en production.

Une dernière partie est consacrée à un bilan du projet et à la description des différents rôles que nous avons tenus. Les difficultés rencontrées y sont décrites ainsi que les succès qui ont jalonné le projet.





# PARTIE I

## CADRE DU PROJET

### 1. GCE Technologies

---

#### 1.1. GCE Technologies dans le groupe BPCE

Le groupe BPCE est issu du rapprochement de 2 réseaux coopératifs bancaires : les Banques Populaires et les Caisses d'Épargne. Le groupe compte 37 millions de clients, 7 millions de sociétaires et comporte 120 000 collaborateurs. De par leur importance, ces chiffres montrent la place prépondérante de la BPCE dans le paysage bancaire français.

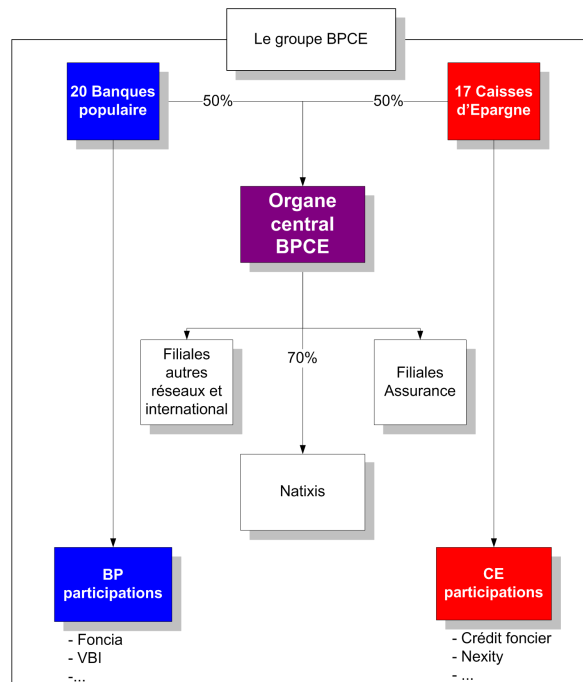


Figure 1 : Organigramme financier du groupe BPCE<sup>1</sup>

---

<sup>1</sup> Source : BPCE

Les Caisses d'Epargne et les Banques Populaires participent financièrement à hauteur de 50% envers l'organe central du groupe BPCE. Ce dernier possède quant à lui plus de 70% de Natixis mais aussi des filiales comme des sociétés d'assurance ou bien d'autres réseaux bancaires nationaux ou internationaux. Enfin les Caisses d'Epargne et les Banques Populaires possèdent des participations dans d'autres sociétés bancaires ou immobilières.

GCE Technologies est un GIE (Groupement d'Intérêt Economique) filiale du groupe BPCE. Elle assure les activités de maîtrise d'œuvre informatique des Caisses d'Epargne en répondant à leurs besoins correspondant à la stratégie informatique définie par le groupe BPCE. La maîtrise d'ouvrage du système d'information est, quant à elle, assurée par un deuxième GIE : GCE Business Services (GCE BS). Notons qu'une part importante des besoins en informatique des Caisses d'Epargne provient de contraintes réglementaires imposées.

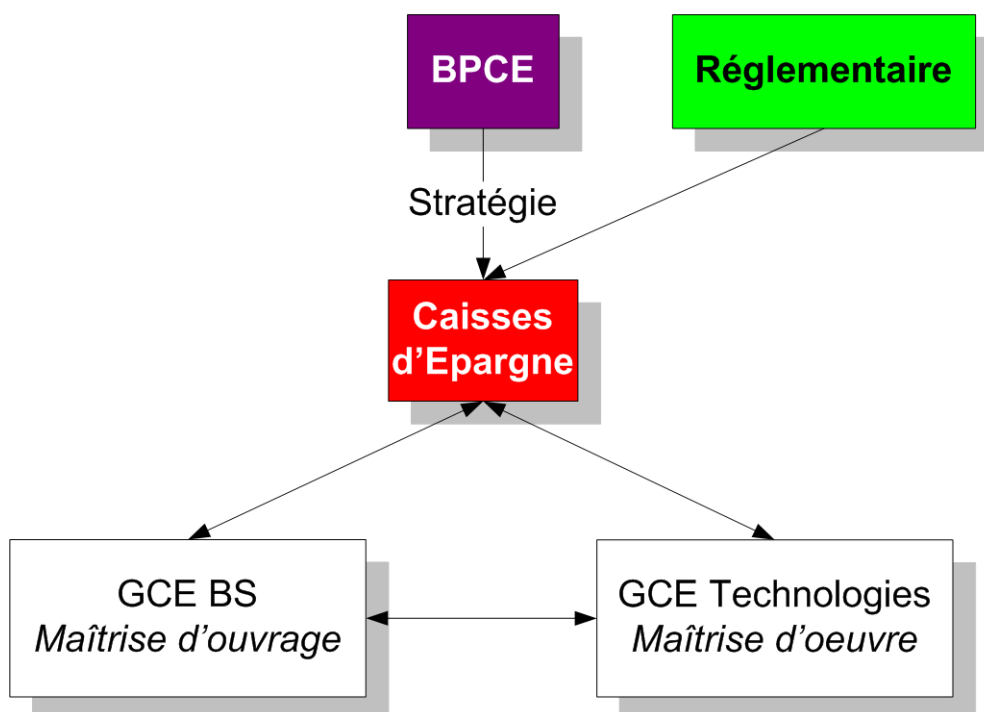


Figure 2 : Gouvernance du système d'information

Les missions de GCE Technologies sont les suivantes :

- Spécification des besoins exprimés par les métiers et proposition de solutions
- Développement et exploitation du système d'information des Caisses d'Epargne et des filiales
- Gestion des interfaces avec les systèmes interbancaires
- Gestion d'activités industrielles bancaires telles que l'édition de documents, le traitement de valeurs ou la surveillance d'automates bancaires.

## **1.2. PIA et financement**

GCE Technologies est financé par l'ensemble des Caisses d'Epargne à travers le PIA (Plan Informatique Annuel). Le PIA constitue pour les Caisses d'Epargne et BPCE la formalisation de l'engagement de GCE BS et GCE Technologies de livrer les nouvelles fonctionnalités du système d'information.

Un PIA est élaboré de la façon suivante : des réunions organisées par métier permettent d'exprimer les besoins d'évolution du système d'information. Les équipes de maîtrise d'ouvrage recensent, hiérarchisent et formalisent les besoins exprimés. Ceux-ci sont chiffrés puis un arbitrage se fait compte tenu de l'enveloppe budgétaire. Les projets retenus constituent le PIA. La réalisation du PIA par GCE BS et GCE Technologies est financée par les Caisses d'Epargne.

## **1.3. Son histoire récente : un système d'information unique**

### ***1.3.1. Vers un système d'information unique***

Auparavant chaque Caisse d'Epargne régionale avait son propre service informatique. Chaque service informatique, même s'il élaborait les mêmes produits, avait un fonctionnement, des socles techniques, une culture d'entreprise propres à chacun.

Afin de rester concurrentielles en France, des Caisses d'Epargne Régionales se sont regroupées ayant pour conséquence le même regroupement au niveau de leurs services informatiques. Ainsi, en 2000 ne restaient plus que huit plateformes applicatives pour gérer l'informatique des Caisses d'Epargne.

Le projet stratégique du Groupe Caisse d'Epargne pour les années 2000-2003 consacrait une large place à la rationalisation de l'informatique. Dès 2000 il avait été décidé de n'avoir plus que 3 plateformes informatiques : Arpège, RSI et SIRIS. Ce sera chose faite en 2003.

Fin 2006, les Caisses d'Epargne décident d'adopter un système d'information unique pour des enjeux aussi bien commerciaux que financiers. Simultanément les Caisses d'Epargne régionales continuent de fusionner entre elles. Toutes les Caisses d'Epargne ont migré sur ce nouveau système en mai 2010. Pour ce faire, le projet PSI (Performance du Système d'Information) est lancé. La plateforme informatique de SIRIS est choisie pour supporter le futur système d'information unique des Caisses d'Epargne, il s'appelle désormais MySys.

### ***1.3.2. Le dernier acte pour construire la plateforme unique***

Comme nous venons de le voir, le regroupement des Caisses d'Epargne sur un seul système d'information se termine en mai 2010. Le projet PSI a donc permis de passer de trois à une plateforme mais également d'effectuer des fusions de Caisses d'Epargne. Ce qui nous intéresse ici, c'est d'exposer les grands principes de ce programme et de montrer les impacts sur le déroulement des projets d'édition.

Dans le cadre du projet PSI une nouvelle organisation au sein de GCE Technologies a été mise en place. Nous souhaitons nous attarder sur ce dernier point pour souligner les changements opérés et leurs impacts sur les projets. Cette nouvelle organisation a entraîné une importante réorganisation des services. Les équipes ont changé, de nouveaux processus ont été mis en place, des cultures d'entreprise différentes ont dû cohabiter. Notre projet a débuté dans cet environnement ce qui a entraîné un travail supplémentaire pour maîtriser les nouveaux processus et identifier les différents services à solliciter pour intervenir au sein de notre projet.

### 1.3.3. La restructuration Caisses d'Épargne et Banques Populaires

Le rapprochement des Caisses d'Épargne avec les Banques Populaires impactent peu MySys pour le moment. Les deux banques ont leur propre système d'information. Des changements apparaissent au niveau des applications de consolidation au niveau groupe. A contrario, notre projet se déroule quasiment à l'identique d'une situation sans les Banques Populaires ; cette restructuration nous impacte très peu.

### 1.4. Son organisation

GCE Technologies est dirigée par un directoire sous lequel se trouvent 4 directions : l'Édition, les Grands Projets, les Relations Adhérents et la Production. L'organisation est résumée dans la figure 3. Nous avons mis en couleur les principaux départements concernés par le projet.

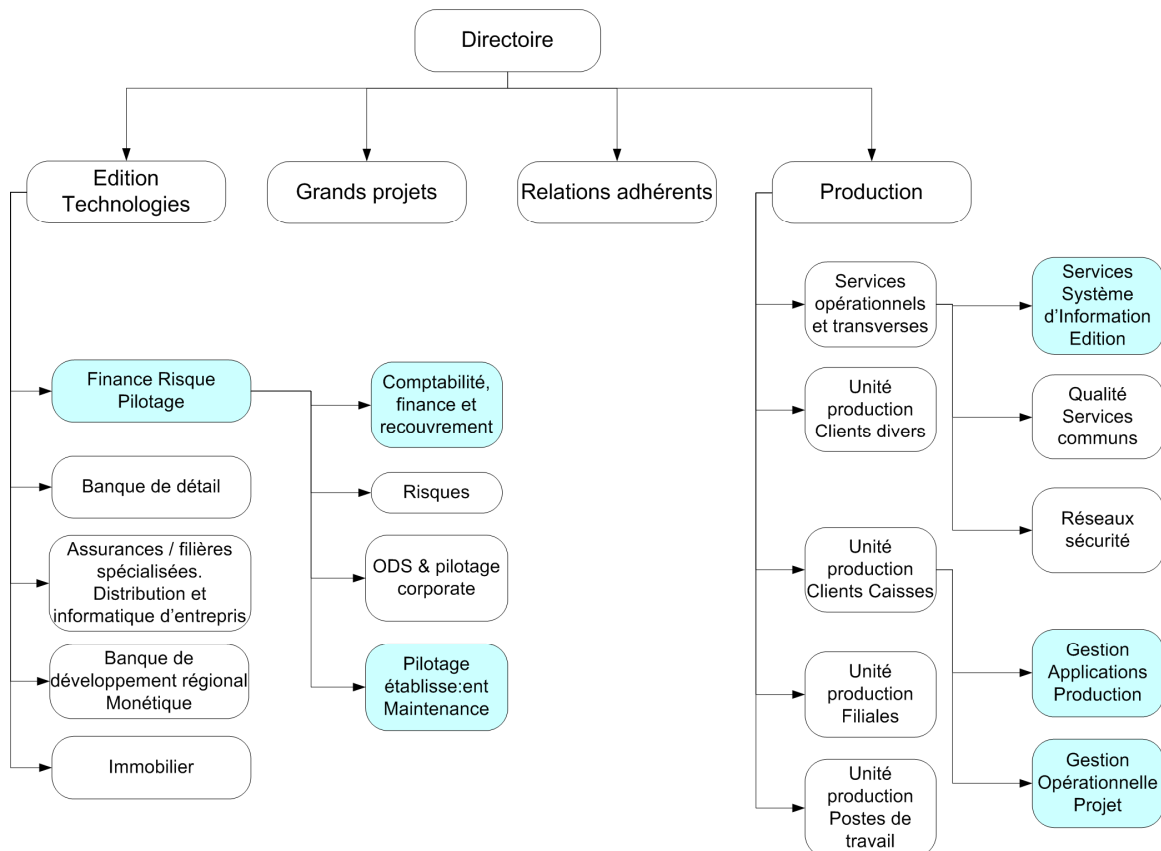


Figure 3 : Organisation de GCE Technologies

## **1.5. MySys, sa plateforme informatique**

### **1.5.1. L'architecture de MySys**

Le projet PSI évoqué précédemment a permis de disposer d'une plateforme informatique unique au service des Caisses d'Epargne. Cette plateforme a été baptisée MySys (contraction de My System : Mon Système en Français).

MySys se veut une plateforme d'accueil multi-bancaire (des banques autres que les Caisses d'Epargne y sont hébergées), multi-canal (les clients peuvent accéder à leur banque via Internet, leur téléphone portable, les automates bancaires, les points de vente, etc...) et multiservices (différents services proposés : banque de détail, d'investissement, de développement régional, etc.). Elle est constituée de divers éléments qui interagissent entre eux :

- Le système central : il est le constituant principal du système ; il fournit les principaux services bancaires tels que la gestion client, l'octroi de crédits, la gestion des produits bancaires (comptes de dépôt, livrets, cartes bancaires, etc.). Ce système central est constitué de machines IBM zSeries gérées par le système d'exploitation z/OS. Les données sont stockées dans des bases de données DB2.
- Les systèmes péricentraux : ils hébergent des applications ne nécessitant pas une implantation sur un système central ou bien ne pouvant être hébergées sur un système central. Ces systèmes péricentraux sont alimentés majoritairement par des données en provenance du système central. Ils supportent des applications de crédit, de banque à distance (Internet), de backoffice (gestion du risque, décisionnel), etc. Les données sont stockées dans diverses bases de données (SQL Server, Oracle, Informix, etc.).
- Les postes de travail : dans les Caisses d'Epargne ils permettent la réalisation des opérations bancaires. Dans les centres informatiques ils permettent les évolutions et le pilotage de MySys ainsi que sa maintenance.

### **1.5.2. Deux instances de production**

La plateforme MySys, nous l'avons vu, est basée sur celle de SIRIS. Cette dernière était multi établissement. Autrement dit chaque fichier de données qui venait alimenter les traitements comportait les informations de tous les établissements. Nous étions en présence d'une seule instance de production. Nous entendons par instance de production l'ensemble des traitements nécessaires à la gestion d'un groupe d'établissement.

La fusion des trois plateformes informatiques en une seule a entraîné des opérations de migration : les Caisses d'Epargne hébergées sur les systèmes RSI et Arpège, systèmes non retenus, migrent petit à petit sur le système MySys. Afin d'absorber la nouvelle charge de traitement des nouvelles Caisses d'Epargne, il a été décidé de mettre en place une deuxième instance de production parallèle à l'instance historique baptisée IP1 (Instance de Production 1). La nouvelle instance se nomme IP2. Tous les traitements de IP1 ont été clonés sur IP2. Ainsi chaque programme est installé deux fois sur MySys.

Cette architecture technique n'a théoriquement pas d'impact pour les projets de développement car, par principe, les deux instances de production doivent être strictement identiques. Les mises en production des évolutions logicielles se font en « Y ».

Sur la figure 4, nous pouvons voir qu'une évolution d'un traitement est mise en production simultanément sur les deux instances de production de manière identique. Cela forme un « Y » à l'envers. Chaque instance traite les données d'un ensemble de Caisses d'Epargne définies.

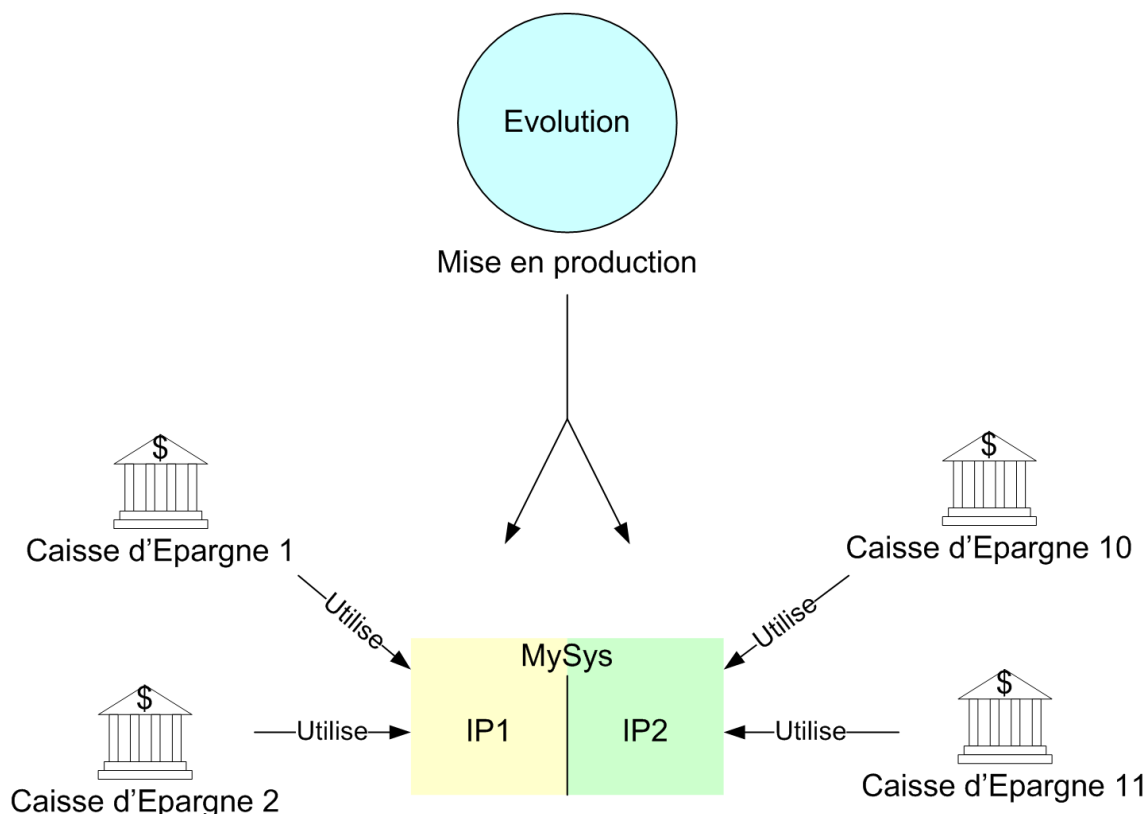


Figure 4 : Deux instances de production

## 2. L'auditeur et GCE Technologies

---

### 2.1. La situation de l'auditeur dans l'organisation

Nous nous trouvons au sein de la direction de l'Edition, dans le département Finance Risque Pilotage. Ce département est découpé en 4 pôles dont le pôle Comptabilité, Finance et Recouvrement au sein duquel nous travaillons. Ce département a la responsabilité du développement et de l'évolution de diverses applications liées à la comptabilité (comme l'intégration de progiciels comptables permettant la comptabilité des établissements), à la finance (gestion de bilan par exemple) mais aussi au recouvrement de créances. Il compte une quarantaine de collaborateurs.

La grande majorité des projets de développement au sein de ce pôle sont à destination d'utilisateurs de backoffice des Caisses d'Epargne (agences ou sièges).

## **2.2. Référent adjoint de la BAFI**

Au sein du pôle Comptabilité, Finance et Recouvrement, les collaborateurs organisés en binôme sont responsables d'applications (ou de domaines) privilégiées sur lesquelles ils sont amenés à travailler prioritairement. Ce sont des référents et référents adjoints. Ceci leur permet d'acquérir une expertise poussée de l'application qu'ils ont en responsabilité.

Dans notre cas, notre rôle de référent adjoint du domaine de la BAFI (que nous allons présenter dans le chapitre suivant) sur MySys nous amène à travailler prioritairement sur cette application. Nous tenons ce rôle depuis 2008. Auparavant nous travaillions sur le système d'information Arpège et sur d'autres applications.

Nous avons hérité d'une application gérée par une équipe de collaborateurs qui travaillaient exclusivement sur celle-ci depuis de nombreuses années. Leur connaissance de l'application était grande, ils n'avaient pas éprouvé le besoin de disposer d'une documentation produit complète. Notre premier travail a été de produire cette documentation qui n'existait pas ou que partiellement. Cette documentation s'est révélée par la suite indispensable dans le projet qui nous intéresse. Elle nous a permis d'une part d'acquérir une bonne connaissance de l'application et de ses particularités et d'autre part de disposer d'un support exhaustif pour les futures études d'impact de notre projet.

## **3. Présentation de la BAFI**

---

### **3.1. La BAFI**

#### **3.1.1. Historique**

Dans les années soixante-dix, les systèmes d'information conçus par les banques étaient principalement orientés vers la collecte d'informations comptables. Avec les années quatre-vingt apparaissent les bases de données et les fichiers permettant un regroupement des informations concernant un même client (lieu de résidence, catégorie d'agent économique, etc.). Les années quatre-vingt-dix voient apparaître la gestion globale du risque nécessitant un système d'information complet et performant. C'est ainsi qu'a été mise en place la Base des Agents Financiers, la BAFI en 1993.

#### **3.1.2. Qu'est-ce que la BAFI ?**

La BAFI est un système homogène de collecte d'informations que les établissements financiers doivent transmettre. Cette collecte permet de répondre aux besoins du contrôle prudentiel mais aussi à l'élaboration de statistiques monétaires. La BAFI permet de donner une vision globale de l'activité d'un établissement (et donc de sa santé) puisqu'elle se fait selon deux axes : les données issues de la gestion (situation fin de mois des comptes courants, des



crédits, etc.) et les données issues de la comptabilité (soldes de chaque compte comptable). Un rapprochement est effectué entre les deux sources afin d'en vérifier la cohérence.

La collecte est restituée sous forme d'états informatiques transmis périodiquement. Ces états informatiques, une fois alimentés, sont encapsulés dans des fichiers de format propriétaire suite à une action de « déclaration » de la part des utilisateurs. Effectuer une déclaration signifie créer le fichier qui contient l'ensemble des états à envoyer à la Banque de France. L'alimentation des états se fait en fonction de critères différents pour chacun. Une même opération peut être analysée de manière différente en fonction du critère qu'on lui attribue.

Outre la production d'états réglementaires, la mise en place de la BAFI en 1993 s'est accompagnée de l'alimentation d'une piste d'audit sur les données traitées. C'est-à-dire une nécessité de garder une traçabilité ascendante et descendante sur toute donnée d'alimentation de la BAFI. En effet, une donnée peut avoir été produite à partir de plusieurs données ; il est nécessaire de garder une trace de cela.

La figure 5 nous donne l'exemple d'une opération bancaire en agence et de sa répercussion jusque dans la BAFI. Cet exemple est simplifié car le niveau de détail des états réglementaires ne va pas jusqu'au client. Il y a toute une série d'agrégations qui sont tracées dans la piste d'audit.

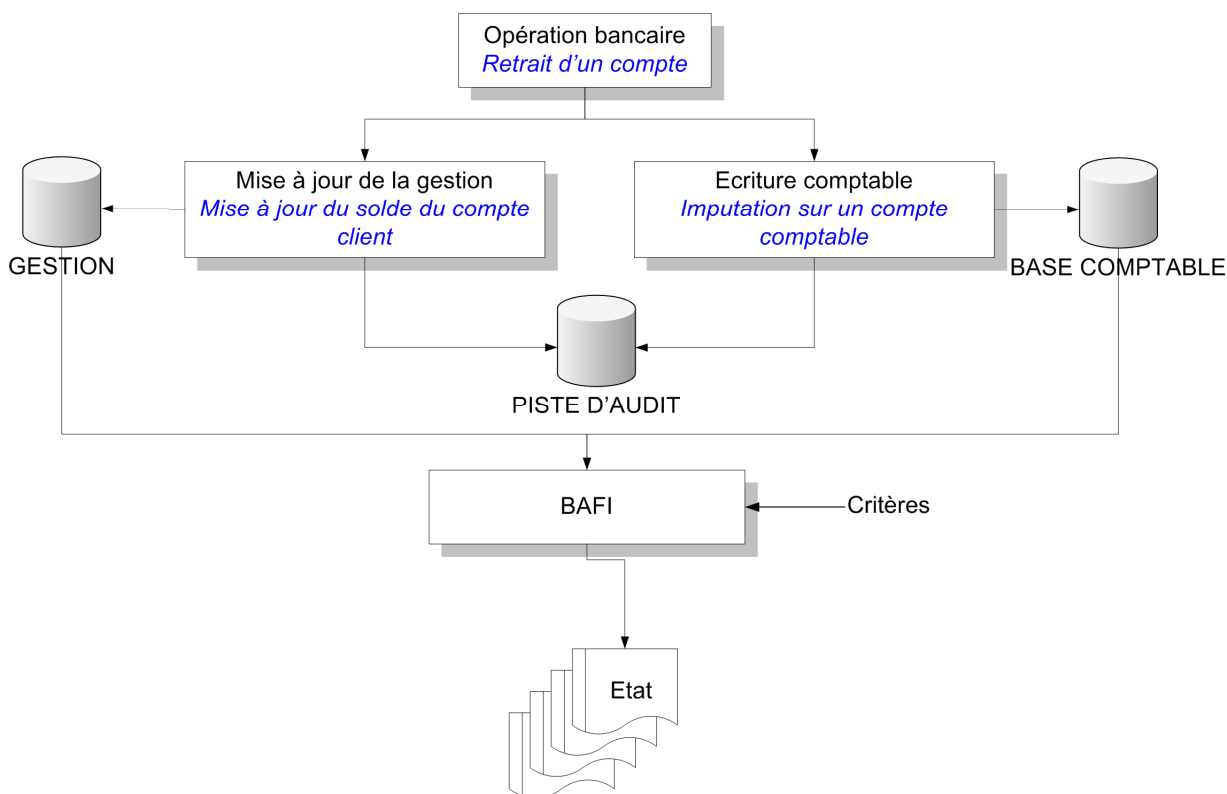


Figure 5 : Principe de la BAFI

### **3.2. Sa mise en œuvre au sein des établissements financiers**

La mise en place de la BAFI au sein d'un établissement financier nécessite un certain degré d'informatisation des données de gestion et a demandé une remise à plat de certains systèmes d'information. En effet, l'information demandée est riche et la mise en place d'une piste d'audit implique l'identification de certaines données en permanence à l'aide d'opérations techniques automatisées. Par contre lors de la mise en place de la BAFI les établissements sont restés libres quant à l'organisation interne de l'information pour autant que la fiabilité des données soit assurée par l'existence d'une piste d'audit.

### **3.3. Ses contraintes**

Comme nous l'avons vu précédemment, la BAFI est réglementaire et doit donc respecter certaines contraintes imposées par les autorités de tutelle.

#### **3.3.1. Fiabilité des données**

La fiabilité des informations fournies est assurée par une piste d'audit qui permet à partir de n'importe quelle donnée de la BAFI d'effectuer une remontée à la source et de consulter les informations qui ont contribué à l'alimentation de celle-ci. Des inspections de la commission bancaire sont réalisées au sein des établissements pour vérifier la conformité de cette piste d'audit.

#### **3.3.2. Délais de remise**

Des contraintes de délai de restitution des états réglementaires sont imposées par les autorités de tutelle (Commission Bancaire et Banque de France). Les états sont restitués chacun à des périodicités différentes : il y a ainsi des états mensuels, trimestriels et annuels. Ensuite pour chaque état, l'autorité impose une date buttoir de remise au-delà de laquelle l'établissement financier subit de lourdes pénalités financières dissuasives. Les établissements financiers font ainsi le maximum pour arriver dans les temps.

#### **3.3.3. Synthèse**

Ces contraintes ne sont pas anodines et remontent de toute évidence jusqu'au système d'information qui doit fournir un service de qualité, rapide et robuste. Nous verrons plus loin que cette contrainte des délais a pris une place importante dans notre phase d'analyse du projet.

## **4. La BAFI chez GCE Technologies**

---

### **4.1. Organisation**

Nous évoquons ici l'organisation de la BAFI de manière générale au quotidien. L'organisation mise en place par le projet qui nous intéresse est détaillée plus loin dans ce mémoire.

Nous avons vu précédemment que nous nous situons dans le département Finance Risque et Pilotage (FRP) au pôle Comptabilité, Finance et Recouvrement. Nous sommes référent BAFI, de ce fait, nous participons prioritairement aux projets liés à la BAFI. Cela concerne ici la partie développement logiciel. La partie maintenance des applications est quant à elle, assurée par des équipes dédiées qui interviennent en cas de soucis lorsque les applications sont en production. L'équipe de maintenance de la BAFI est également dans le département FRP mais dans un pôle différent : « Pilotage Etablissement et Maintenance ». Nous verrons par la suite que nous avons été amenés à établir un dialogue permanent avec cette équipe, source de conseils lors de nos analyses du projet. De même, lorsqu'un projet est en développement, l'équipe de maintenance intervient pour s'informer des évolutions apportées et éventuellement effectuer des réserves quant à la conception.

Enfin nous devons également mentionner l'équipe d'exploitation qui veille au bon déroulement des traitements informatiques et donc ceux de la BAFI. Ces équipes sont plutôt en relation avec les équipes de maintenance, mais nous les avons également sollicitées pour avoir leur avis sur certains points précis du projet.

La figure 6 nous présente les différentes interactions qu'il y a entre les acteurs participant au fonctionnement de la BAFI chez GCE Technologies.

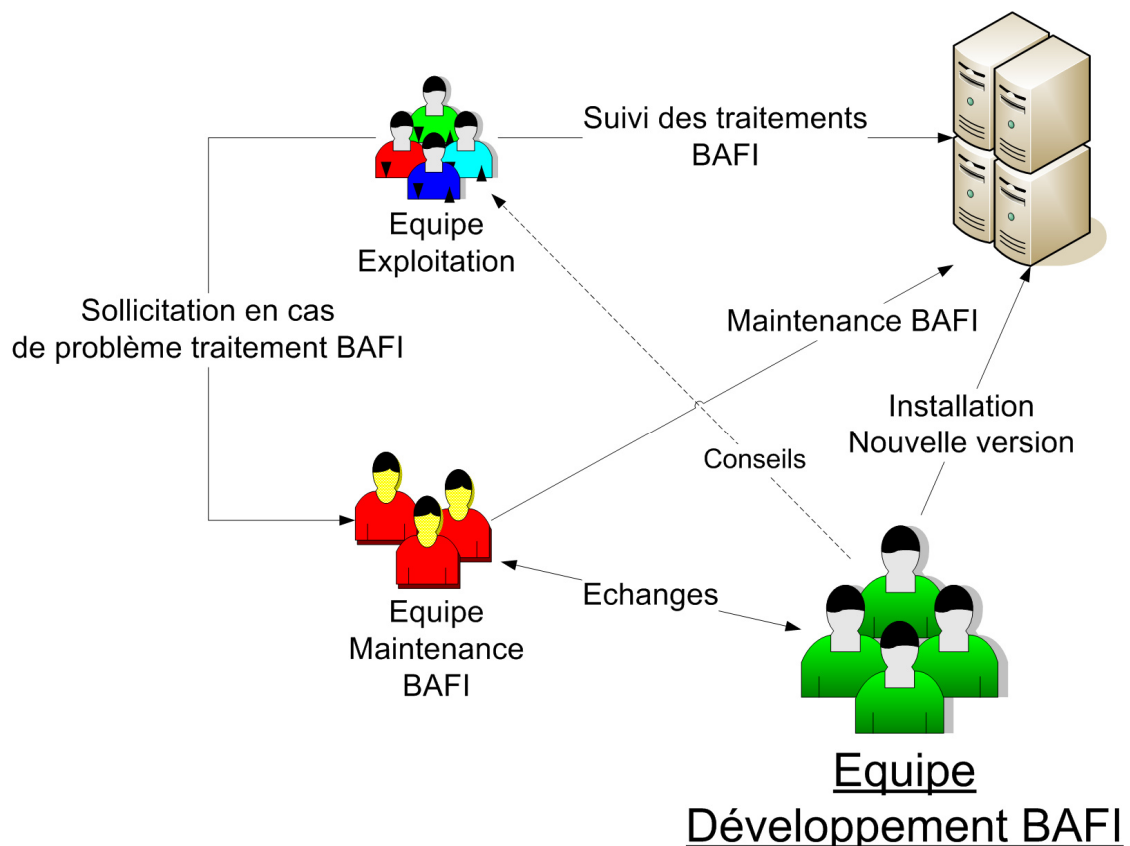


Figure 6 : Organisation de la BAFI chez GCE Technologies

## 4.2. Utilisation d'un progiciel : Evolan Report

Différents éditeurs proposent des solutions logicielles pour la production des états réglementaires BAFI. Ainsi des sociétés comme Sopra, Viveo ou Invoke tiennent une large place dans le domaine du reporting financier et donc de la BAFI.

La solution retenue pour MySys est celle de l'éditeur Sopra : le logiciel « Evolan Report ». A part le module Client, le reste du logiciel peut être indifféremment installé sur des socles UNIX, Windows ou encore z/OS (gros système IBM).

Evolan Report se compose de quatre modules ayant chacun son propre rôle : Client, Server, Engine et Références croisées (que nous appelons XRef pour plus de lisibilité). La figure 7 nous donne les interactions existantes entre chacun des modules ainsi que le point d'entrée des données. Nous voyons que deux bases de données sont présentes, la première alimentée par le module Server et la deuxième par le module Engine.

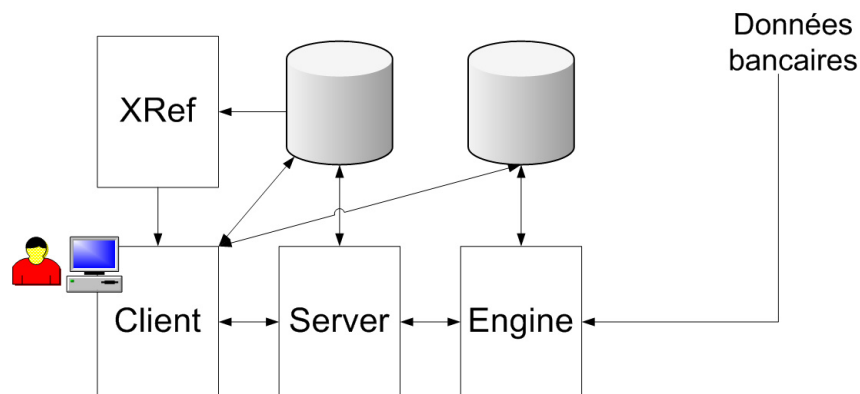


Figure 7 : Architecture du logiciel Evolan Report

### 4.2.1. Principe général de fonctionnement

Le fonctionnement général d'Evolan Report est le suivant : l'utilisateur effectue des demandes de fabrication d'états réglementaires à travers le module client. Ces demandes transformées en fichiers par le module Server sont envoyées au module Engine qui, à l'aide des données bancaires, fabrique les résultats. Ceux-ci sont ensuite renvoyés vers le module Server puis chargés dans la base de données.

### 4.2.2. Le module Client

Ce module permet d'offrir aux utilisateurs une interface graphique à travers laquelle ils effectuent toutes les opérations nécessaires à la production des états réglementaires :

- envoi des demandes de fabrication
- consultation/modification des états
- consultation des données ayant servies à l'alimentation des états

- production des fichiers réglementaires
- paramétrage
- etc.

La figure 8 nous montre un état réglementaire alimenté tel qu'il apparaît à l'utilisateur à travers le module Client.

SITUATION MENSUELLE					M_SITMENS	
Montants en unités d'Euros						
Société : C						
Date d'arrêté : 31/03/2010						
Activité : France						
Monnaie : Euros						
Périmètre : Social						
	ACTIF	Amortissements dépréciations	Résidents	Non-Résidents EMUM	Non-Résidents Non EMUM	Total
		1	2	3	4	5
1	<b>OPERATIONS DE TRESORERIE ET OPERATIONS INTERBANCAIRES</b>					
1.1	Billets et monnaies					1 172 793 011,21
1.2	Autres valeurs et créances interbancaires	1 856 954,30	4 554 669 066,09			4 554 669 066,09
2	<b>OPERATIONS AVEC LA CLIENTELE</b>					
2.1	Concours à la clientèle	15 048 354,46	1 185 850 226,83	799 212,35	852 164,56	1 187 501 603,74
2.2	Prêts à la clientèle financière		361 113,16			361 113,16
2.3	Valeurs non imputées		12 201 971,29			12 201 971,29
3	<b>OPERATIONS SUR TITRES</b>					
3.1	Titres reçus en pension livrée					
3.2	Portefeuille titres		0,01			0,01
3.3	Instruments conditionnels achetés					
4	<b>VALEURS IMMOBILISEES</b>					
4.1	Prêts subordonnés					
4.2	Parts dans les entreprises liées, titres de participation, autres immobilisations financières	19 032 123,02	722 775 821,46			722 775 821,46
4.3	Actifs immobilisés					77 086 392,87
4.4	Crédits-bail et opérations assimilées (encours comptable)					
4.5	Créances douteuses	579 034,23	606 809,21			606 809,21
5	<b>AUTRES POSTES DE L'ACTIF</b>					242 502 587,87
6	<b>TOTAL ACTIF</b>					7 970 518 376,31

Figure 8 : Etat Evolan Report.

#### 4.2.3. Le module Server

A l'écoute permanente du Client, le module Server prend en compte les demandes faites par l'utilisateur pour les transmettre ensuite au module Engine. A l'inverse ; il permet la transmission des résultats de fabrication produits par Engine vers le Client. Ce module possède sa base de données dans laquelle il stocke les demandes faites par l'utilisateur ainsi que les résultats de fabrication.

#### 4.2.4. Le module Engine

Le module Engine est en activité lors de traitement batch. Engine prend en compte les demandes de l'utilisateur transmises par le Server et produit des résultats de fabrication qui permettent d'alimenter les états à partir des données bancaires. Engine possède sa base de données dans laquelle sont stockées les données bancaires.

#### 4.2.5. Le module Références Croisées

Le module Xref, basé sur un serveur Web, propose à l'utilisateur d'effectuer des recherches d'informations croisées selon différents axes ; par exemple, quels attributs alimentent un état ou alors quels états sont concernés par un compte comptable.

Dans la figure 9, nous pouvons visualiser le résultat d'une requête Xref qui montre les attributs concernés par un état.

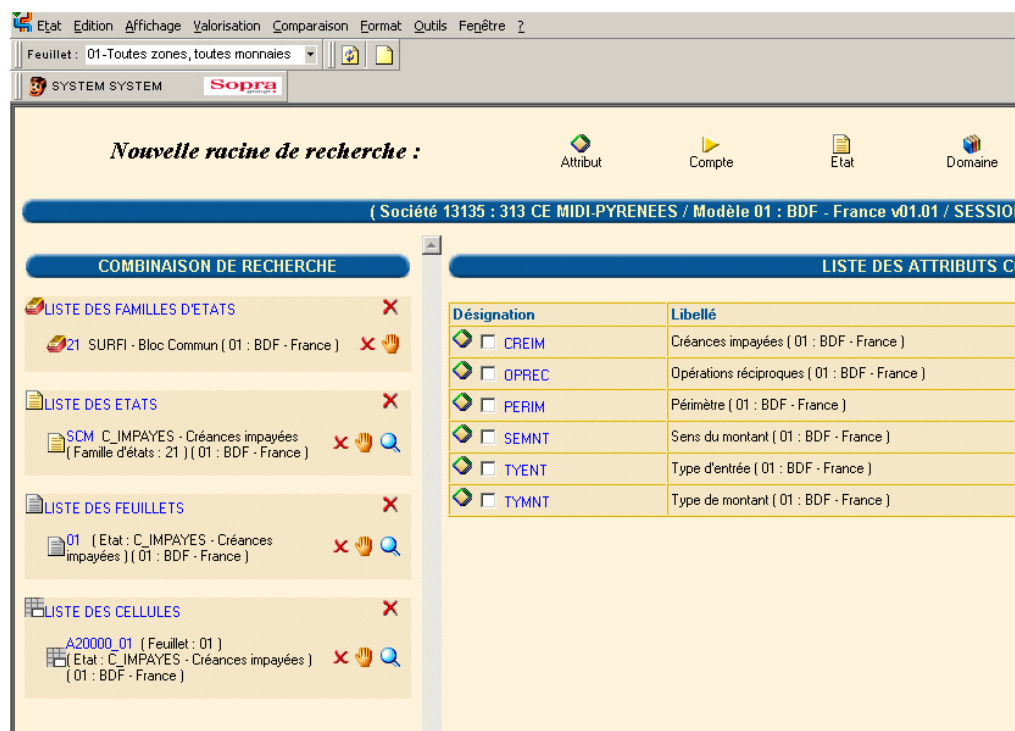


Figure 9 : Résultat d'une requête références croisées

### 4.3. Alimentation du logiciel Evolan Report

Nous avons vu précédemment que la BAFI était alimentée d'une part avec les données des applications de gestion et également avec les données de la comptabilité. Chez GCE Technologies, ces deux catégories de données sont produites sur gros système z/OS sous forme de CRI (Compte Rendu d'Inventaire). Ces CRI ne peuvent être présentés tels quels à l'entrée d'Evolan Report. En effet, Evolan Report attend des ESTD (Entrées standards), c'est-à-dire des enregistrements ayant une structure déterminée : chaque enregistrement possède une partie fixe non modifiable imposée par Sopra et une partie variable comportant les attributs. Dans cette dernière les attributs sont structurés à la discrétion de l'entreprise.

Ce format particulier des données nécessite de transformer les CRI en ESTD. C'est le rôle de l'interfaçage.

### 4.4. D'autres applications autour de Evolan Report

#### 4.4.1. GEIDE

Les traitements du module Engine produisent des états contenant des informations sur les données d'alimentation (ESTD). Certaines données sont notamment rejetées car non-conformes à ce qui est attendu (attribut non renseigné ou montant à zéro par exemple). Ces états sont récupérés en fin de

traitement Engine pour ensuite être envoyés vers le z/OS. A l'arrivée des états sur z/OS, un traitement les transfère sur le serveur GEIDE (Gestion Electronique d'Informations et de Documents de l'Entreprise). Les états sont alors accessibles via une interface web par les utilisateurs.

#### **4.4.2. BO**

Nous avons vu précédemment qu'une piste d'audit est en place concernant la constitution des données d'alimentation de la BAFI. Les traitements du z/OS fabriquent des fichiers de piste d'audit qui sont transférés sur le serveur UNIX pour être enfin chargés dans une base de données.

Les utilisateurs, via le logiciel BO (Business Object), peuvent consulter cette piste d'audit depuis leur poste et donc interroger le détail des données qui sont fournies par le z/OS en entrée du logiciel Evolan Report.

### **4.5. Principe de fonctionnement**

Nous allons expliciter les notions fondamentales du fonctionnement du logiciel Evolan Report : la session, le traitement Engine et l'organisation générale.

#### **4.5.1. La session**

La notion de base du logiciel Evolan Report est la « session ». A une session correspond un environnement de travail. Par exemple, dans la session numéro un, l'utilisateur travaille sur la constitution des états à remonter mensuellement et sur la session numéro deux, il travaille sur les états trimestriels. Ou encore plusieurs utilisateurs d'une même Caisse peuvent travailler simultanément sur des sujets différents et donc sur des sessions différentes. Chaque Caisse d'Epargne dispose de vingt sessions.

Les sessions sont étanches et indépendantes les unes des autres au niveau de leurs données. Ainsi un utilisateur peut effectuer une modification d'une donnée en entrée du module Engine, cette modification ne se fait que pour la session en question. Les autres sessions ne sont pas impactées par cette modification.

Une session peut faire l'objet de différentes actions/demandes qui, pour être complètes, doivent toutes être soumises à un traitement Engine :

- Ouverture : L'utilisateur ouvre une session en effectuant une sélection d'états (parmi une liste prédéfinie) qui participent à la session. D'autres paramètres sont à saisir pour définir la session tels que la périodicité de la session, la date d'arrêté traitée, le nom de la session, etc.
- Fabrication : ne peut se faire que sur une session ouverte, l'utilisateur effectue des modifications de paramétrage ou de données en entrée d'Engine et demande une nouvelle fabrication des états
- Fermeture : L'utilisateur n'a plus besoin de la session, il a produit ses fichiers de déclaration à envoyer à la Banque de France, il peut donc fermer sa session
- Archivage : la session est archivée dans des fichiers physiques qui sont conservés sur la machine UNIX. Cette session peut faire l'objet d'une

restauration. A noter que les données qui ont participé à l'élaboration de la session sont également archivées et peuvent également être restaurées.

- Restauration : une session archivée est mise à disposition de l'utilisateur dans l'interface du logiciel

#### 4.5.2. Le traitement Engine

Le module Engine se compose de programmes qui sont lancés en batch. Le batch prend en compte les demandes faites par le client (ouverture, fabrication ou fermeture de session) et y répond à l'aide des données envoyées par le z/OS. Il produit des résultats de fabrication qui sont ensuite chargés dans une base de données du module Server. Ces résultats sont alors consultables via le client à travers les états remplis.

#### 4.5.3. Fonctionnement général

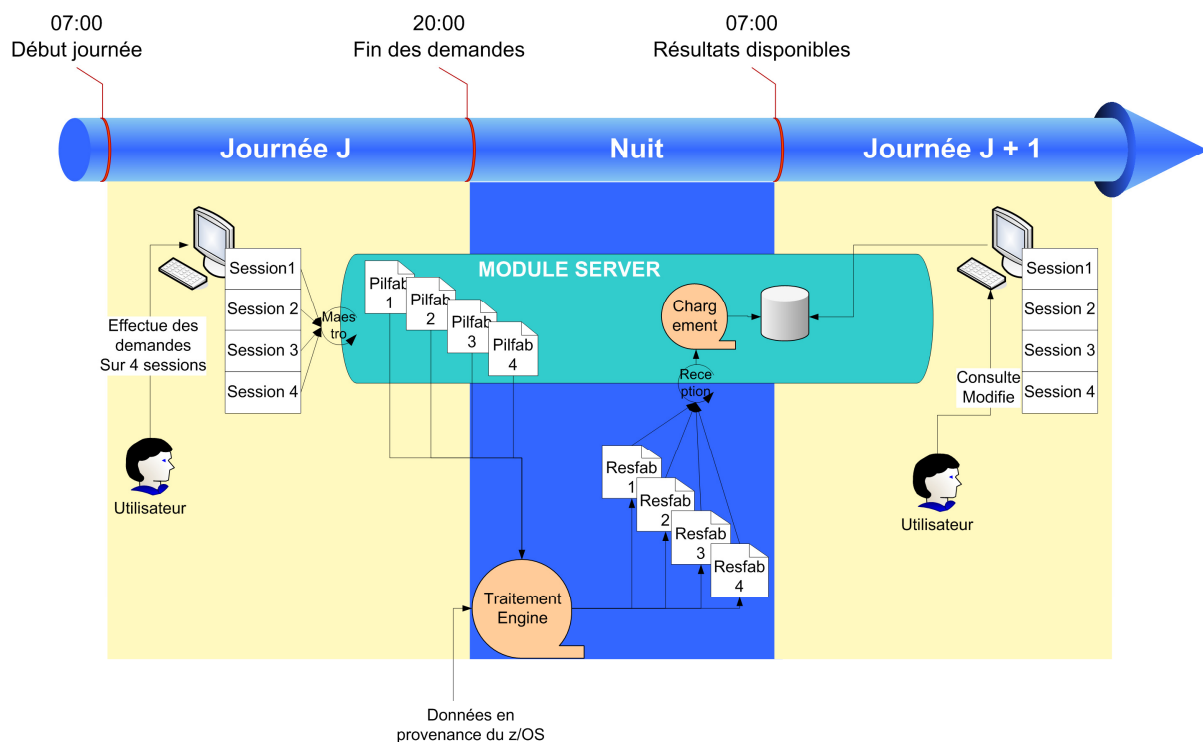


Figure 10 : Les journées de la BAFI

La figure 10 donne un aperçu des traitements dans le cas d'un utilisateur qui travaille sur quatre sessions simultanément.

Le cycle de production des états BAFI n'est pas instantané et répond à une organisation des traitements qui est la suivante : en journée, les utilisateurs effectuent autant de demandes qu'ils souhaitent. Ces demandes sont traduites par le module Server en fichiers « Pilfab » (pilotage de fabrication) qui sont stockés en attente d'envoi au module Engine. La nuit le traitement Engine se déroule : prise en compte des Pilfab et production des résultats de fabrication (Resfab) à partir des données envoyées par le z/OS. Les résultats de fabrication sont donc disponibles le lendemain pour les utilisateurs.



## 4.6. Architecture technique

### 4.6.1. Architecture technique générale

La figure 11 nous donne une description de l'architecture technique chez GCE Technologies.

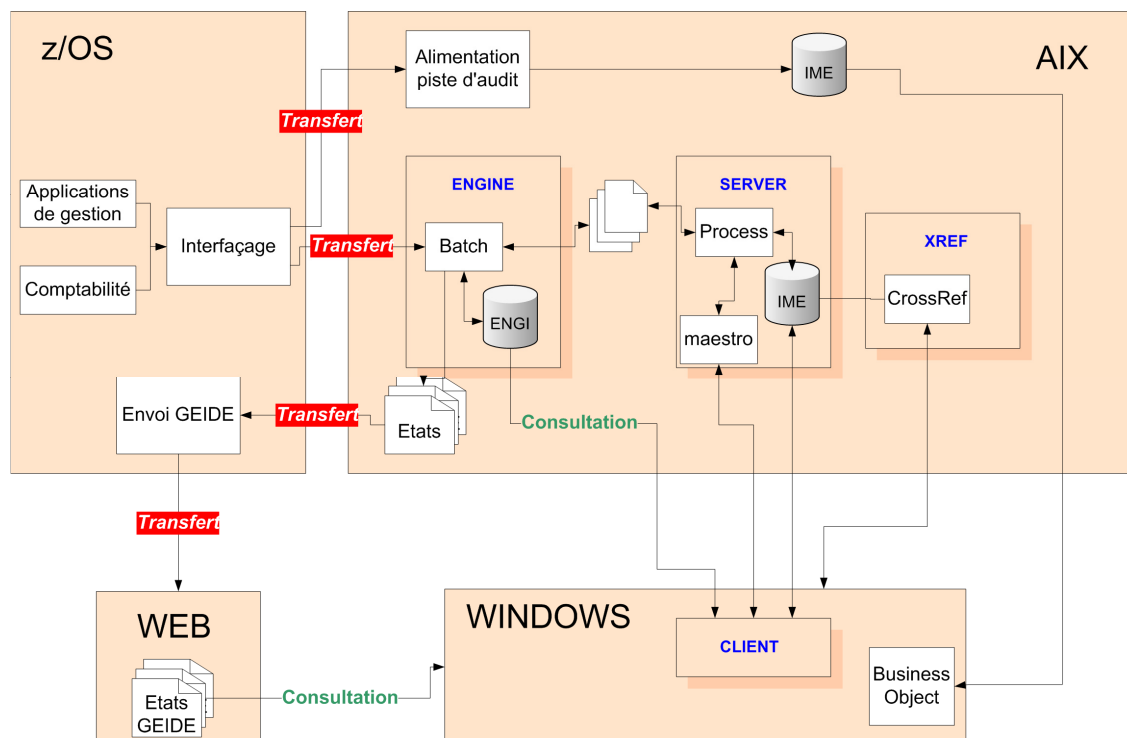


Figure 11 : Architecture technique de la BAFI chez GCE Technologies

Les trois modules Engine, Server et XRef sont hébergés sur un serveur UNIX de type AIX. Le module Client quant à lui se trouve sur une machine Windows.

Nous pouvons également mentionner la partie alimentation des données qui se trouve sur serveur central z/OS et qui vient en entrée du batch du module Engine pour la fabrication des résultats.

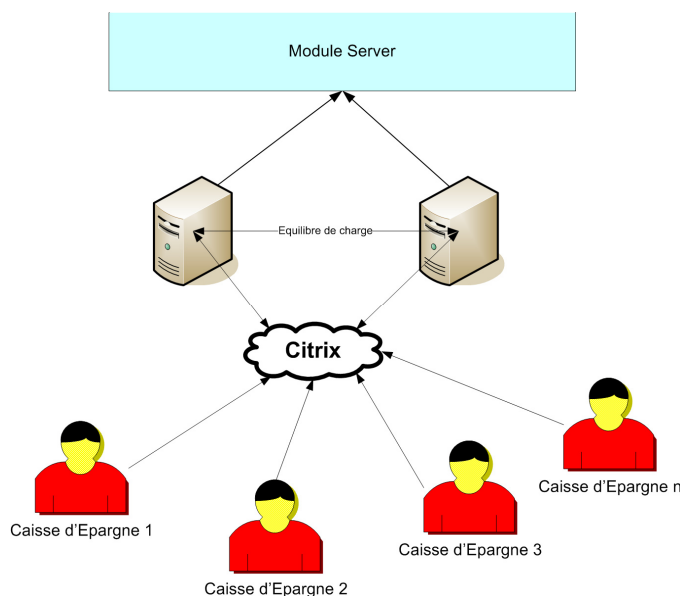
Nous découvrons également sur le module Server un service appelé Maestro. Celui-ci est à l'écoute des demandes du client et les transmet à un deuxième service appelé Process. Ce dernier fabrique des fichiers, traduisant les demandes faites par l'utilisateur, appelés « fichiers de pilotage ». Ils sont envoyés à Engine.

### 4.6.2. Architecture technique du module Client

Afin d'accroître les performances et garantir la disponibilité de l'application Evolan Report, un système de double serveur Windows avec équilibre de charge a été mis en place pour supporter le module Client.

Le module Client, tout comme les autres modules, est installé sur MySys et non dans les Caisses d'Épargne. Chaque utilisateur dans les établissements accède donc à Evolan Report à distance, rien n'est installé sur son poste. Cet accès se fait via une connexion Citrix. Cela veut dire que plusieurs utilisateurs de plusieurs établissements accèdent simultanément au module Client.

Une seule machine ne peut supporter toutes les connexions pendant un long moment et l'apparition d'une panne est synonyme de coupure de service pour l'ensemble des établissements. Deux machines ont été montées sur chaque instance de production. Le module Client d'Evolan Report a donc été installé deux fois par instance. Les deux machines fonctionnent en équilibre de charge. Ainsi, la coupure d'une machine n'entraîne pas une coupure de service puisque la deuxième peut prendre temporairement le relais. Les performances sont accrues car l'ensemble des établissements d'une instance de production se partagent deux machines.



**Figure 12 : Machines Windows**

#### **4.7. Architecture organisationnelle**

Nous décrivons ici le fonctionnement global de la BAFI et les interactions entre les différents acteurs que sont les utilisateurs, MySys et BPCE. Il s'agit de comprendre notamment les différents échanges pratiqués entre chacun.

Voici le détail de chacun des échanges observés sur la figure 13 :

- 1 : L'utilisateur se connecte à l'interface d'Evolan Report et émet une demande de fabrication
- 2 : La demande de fabrication est transmise immédiatement au module Server puis la nuit au module Engine qui traite les demandes et crée des résultats de fabrication qu'il renvoie au module Server
- 4 : Les états sont remplis et à disposition de l'utilisateur
- 5 : L'utilisateur peut consulter les états à partir de l'interface Evolan Report
- 6 : L'utilisateur effectue une déclaration : des fichiers contenant les états déclaratifs sont produits sur son poste et envoyés à BPCE via une interface dédiée.
- 7 : BPCE envoie les fichiers de tous les établissements du groupe à la Banque de France.

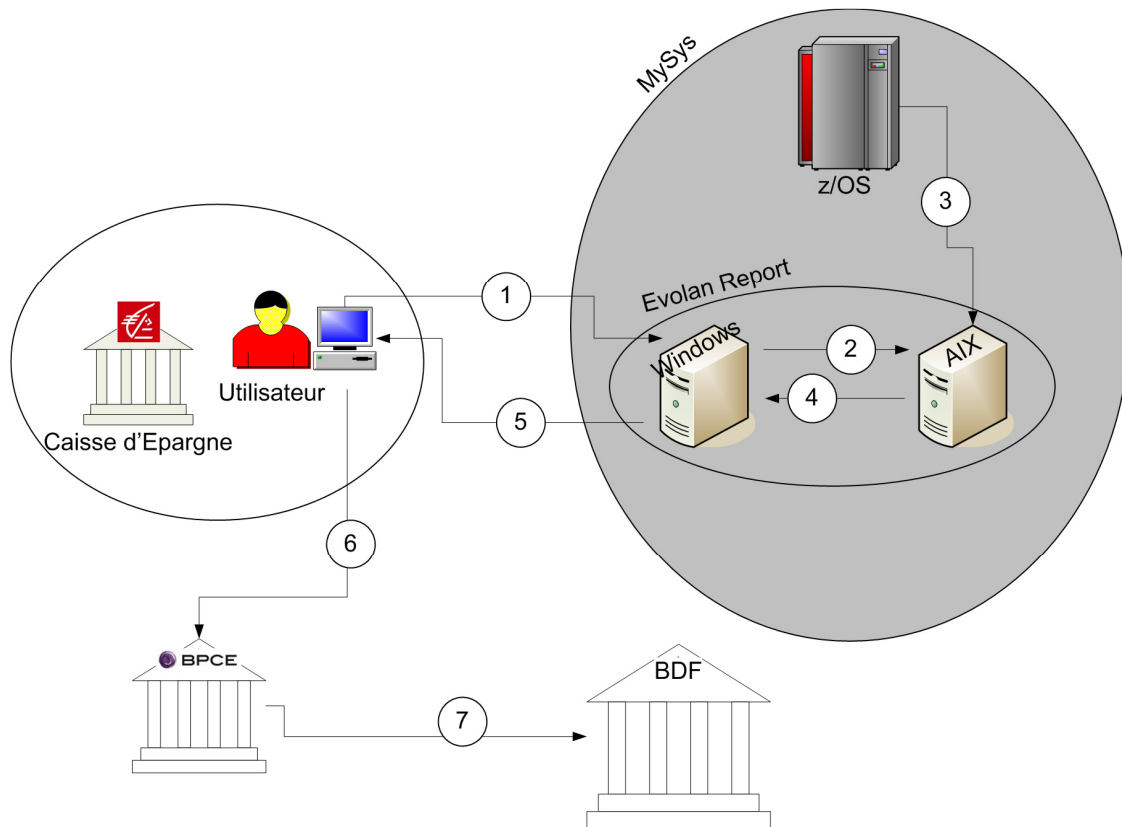


Figure 13 : Architecture des échanges de la BAFI

#### 4.8. Langages utilisés

La BAFI s'appuyant sur une solution à base de progiciel, il faut distinguer deux catégories : d'une part les langages utilisés dans le logiciel Sopra et d'autre part ceux mis en oeuvre pour l'intégration du logiciel Evolan Report dans MySys. Nous pourrions ne pas nous intéresser aux langages du logiciel de Sopra, mais certains livrables Sopra sont adaptés pour fonctionner sur MySys, il est donc important de les mentionner. Ainsi nous rencontrons du Cobol, du SQL et du Korn-Shell dans le logiciel Evolan Report. Des programmes utilisant ces deux derniers langages ont été modifiés pour permettre un fonctionnement optimal sur le système d'information MySys.

Concernant l'intégration du logiciel à MySys, les programmes sont développés en Korn-Shell et SQL pour la partie UNIX et en batch DOS pour la partie Windows.

Enfin, tous les traitements Unix sont lancés de manière automatique par un ordonnanceur. Sur MySys, l'ordonnanceur est Visual TOM (VTOM).

#### 4.9. Les environnements

Au commencement du projet SURFI en début d'année 2009, la BAFI possédait deux environnements : la production sur laquelle tourne le logiciel au profit des Caisses d'Épargne et un environnement de test qui n'était pas à l'image de la

production (en terme notamment d'architecture technique). Cet environnement permettait à l'équipe de maintenance d'effectuer leur travail au quotidien et notamment le test des correctifs à chaud.

Pour des questions de sécurisation, nous n'avons pas voulu que le projet SURFI se déroule dans ces conditions techniques avec un manque important d'environnement de développement et de recette conformes à l'architecture technique de l'environnement de production. Nous avons donc fait la demande de montage d'un environnement de développement qui nous permettrait d'effectuer les tests unitaires et d'un environnement de recette utilisateur, à l'image de la production. Nous avons obtenu l'environnement de recette utilisateur seulement. Le suivi du montage et sa réception font l'objet d'une description plus loin dans ce mémoire.

Parallèlement à ces demandes, au lancement du projet, la maîtrise d'ouvrage a exprimé le besoin de disposer, pour la recette du projet, d'un environnement de volumétrie réelle. Cette demande fait également l'objet d'une description plus tard dans le texte. Il a été monté avec la plus grosse Caisse d'Epargne en termes de volumétrie de données et a nécessité un investissement en temps important pour sa construction et sa mise au point.

## ***5. Réforme de la BAFI : SURFI***

---

### **5.1. Présentation de la réforme**

Nous décrivons maintenant les raisons qui ont conduit les autorités de tutelle à lancer une réforme de la BAFI.

Nous l'avons vu la BAFI permet la collecte d'informations en provenance de la gestion et de la comptabilité. Nous pouvons rajouter que deux familles d'états se côtoient : les états 4000 (états périodiques et comptables) et les états 8000 (états statistiques et monétaires) incluant la surveillance des grands risques.

Des critiques sont apparues en provenance aussi bien des établissements financiers que des autorités de tutelle sur le fond et sur la forme. La BAFI souffre d'un format propriétaire pour la production des états, lourd à faire évoluer et non homogène avec les autres reportings (par exemple le COREP). Enfin certaines informations sont redondantes, une rationalisation de celles-ci doit être faite.

En 2007, la Banque de France et la Commission bancaire décident de lancer une réforme de la BAFI intitulée le SURFI (Système Unifié de Reporting Financier). Cette réforme a les objectifs suivants :

- Réduire de 25% la charge de travail nécessaire à la production des reportings tout en continuant à garantir la continuité de supervision. Ceci en réduisant le nombre d'informations demandées et en supprimant les redondances.
- Fournir des données statistiques demandées par la BCE (Banque Centrale Européenne) améliorant la qualité de la supervision.

- Changer de format d'échange entre les établissements financiers et la Banque de France, à l'instar du COREP, en utilisant le XBRL (eXtensible Business Reporting Language) permettant ainsi une meilleure exploitation des informations.

## **5.2. Impacts dans les établissements financiers**

Les impacts se rencontrent à deux niveaux : du côté du backoffice de chaque établissement financier et au niveau des systèmes d'information.

Au niveau des systèmes d'information :

- Production de nouvelles données dans les systèmes de gestion.
- Evolution majeure du logiciel de fabrication des états réglementaires

Au niveau du backoffice :

- Nouvelles méthodes de remise à la Banque de France. Cette dernière met en place un nouveau portail intitulé « Onegate » à partir duquel les fichiers des états réglementaires seront déposés à partir de juillet 2010.
- Réduction des délais de remise à la Banque de France : J+25 calendaires à J+10 ouvrés pour l'équivalent des états de bilan trimestriels.

## **5.3. Impacts pour GCE Technologies**

Chez GCE technologies, la mise en place de SURFI entraîne :

- La production de nouvelles données en provenance des applications de gestion et à mettre en entrée du batch Engine
- La modification des interfaces de mise au format ESTD pour prendre en compte les nouvelles données à alimenter
- Une montée de version du logiciel Evolan Report
- Une logique de transfert des fichiers de déclaration à revoir

## **5.4. Contraintes de la réforme**

La contrainte majeure de la réforme se trouve au niveau des délais. En effet, tous les établissements financiers doivent effectuer une déclaration SURFI sur les données de fin de mois au 30 juin 2010. Cela donne peu de marge de manœuvre en cas de difficultés et amène, nous le verrons, une sécurisation accrue de la mise en production du projet.

Une autre contrainte est liée à l'utilisation du logiciel Evolan Report provenant d'un éditeur externe. Nous sommes en effet dépendants de la date de livraison des composants. Or GCE Technologies possède des normes de mise en production basées sur un système de version. Une version correspond à l'ensemble des évolutions apportées au système d'information MySys à une date prédéfinie. GCE Technologie, de manière générale, propose deux dates par an à l'occasion desquelles le système d'information peut être modifié par l'installation d'une version. SURFI va être mise en production lors de la version de juin. Or une

version possède des jalons qu'un projet doit respecter. La livraison en plusieurs lots du logiciel Evolan Report a nécessité des aménagements par rapport aux contraintes de la version.

## **6. Conclusion**

---

Cette première partie nous a permis de tracer le cadre du projet SURFI. SURFI, Système Unifié de Reporting Financier est une réforme de la BAFI, mécanisme de reporting financier imposé par les autorités de tutelle. GCE technologies, exploitant le système d'information des Caisses d'Epargne dans le groupe BPCE, met en œuvre ce reporting et doit donc mettre en place la réforme SURFI. Nous avons ainsi décrit le fonctionnement de la BAFI chez GCE Technologies et les contraintes apportées par la réforme.

Nous pouvons maintenant décrire le projet tel qu'il s'est déroulé et plus particulièrement les actions que nous avons effectuées. Ceci est l'objet de la deuxième partie.



# PARTIE II

## LE PROJET

Nous l'avons vu précédemment, la réforme de la BAFI initiée par la Banque de France doit être mise en place dans les établissements financiers à la mi-2010. Cela s'est traduit chez GCE Technologies par la mise sur pied d'un projet qui s'intitule « SURFI, réforme de la BAFI ». Ce mémoire s'appuie sur ce projet et plus particulièrement sur l'un des chantiers du projet. Nous décrivons ici sa mise en place, son déroulement et nos interventions au sein de celui-ci.

### ***1. Equipe projet et méthodologie***

---

#### **1.1. Découpage en chantiers**

Nous l'avons déjà évoqué précédemment, les impacts de la mise en place de SURFI sont d'une part de nouvelles alimentations de données et donc une modification des interfaces et d'autre part une montée de version du logiciel Evolan Report. Ces deux impacts concernent des technologies différentes. Ainsi tout ce qui concerne l'alimentation et l'interfaçage des données est réalisé sur gros système, type z/OS alors que le logiciel Evolan Report est installé sur de l'UNIX et du WINDOWS. Il a été décidé de découper le projet en deux chantiers techniques :

- un chantier z/OS ayant pour responsabilité les évolutions à apporter à l'alimentation et l'interfaçage des données
- un chantier UNIX/Windows ayant en charge la montée de version du logiciel Evolan Report ainsi que toutes les évolutions de technologies Windows et UNIX.

Ce deuxième chantier est le support de notre mémoire.

#### **1.2. Les ressources intervenant sur le projet**

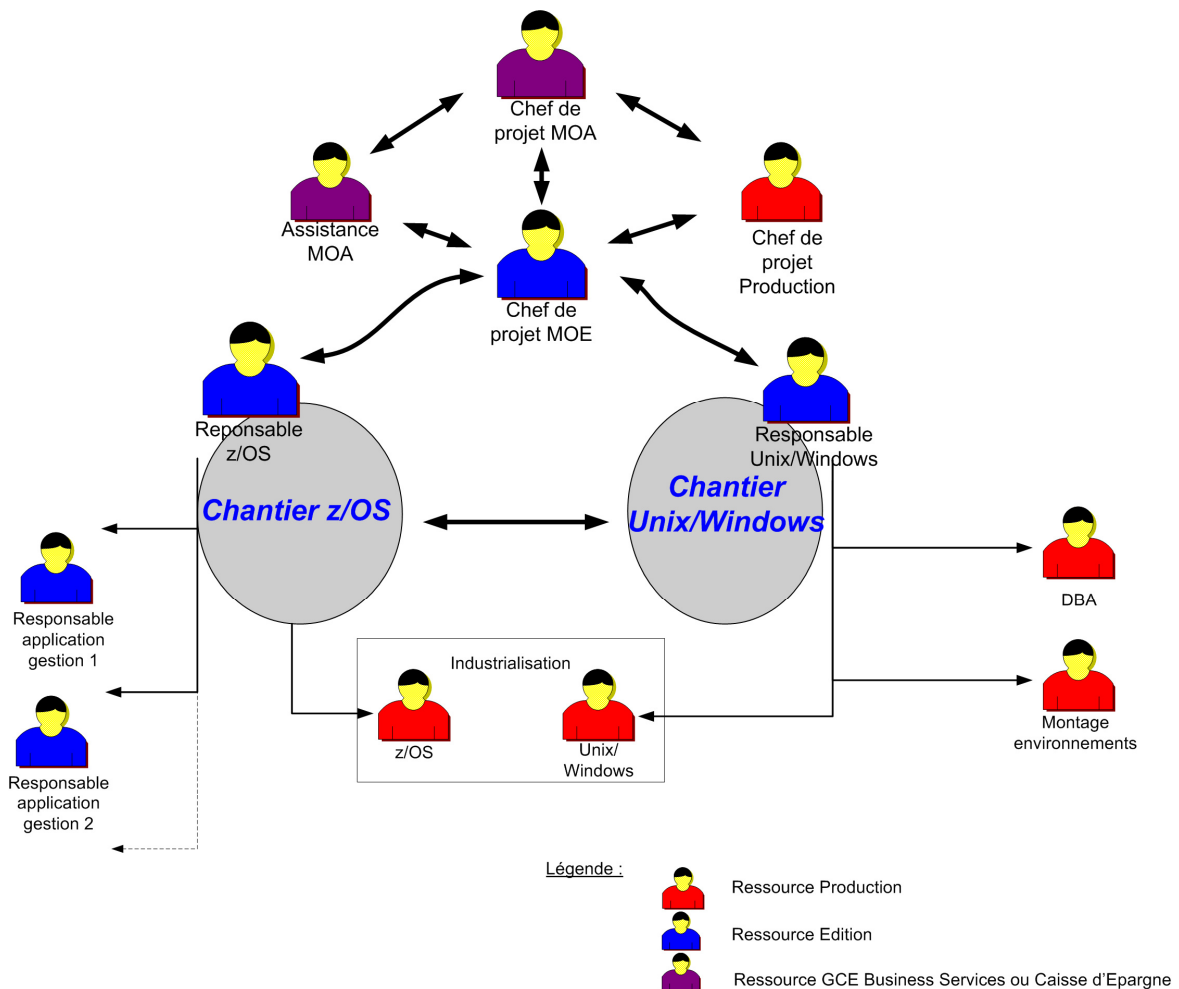
Du côté de la réalisation du projet, celui-ci s'articule autour de trois personnes principales: le chef de projet maîtrise d'œuvre (MOE), le responsable du chantier



z/OS et le responsable du chantier Unix/Windows, responsabilité qui nous a été confiée.

La figure 14 nous montre qu'à côté de ces trois personnes, d'autres intervenants participent au projet, nous avons :

- le chef de projet maîtrise d'ouvrage (MOA) du GIE GCE Business Services qui permet de faire le lien entre le projet et les utilisateurs.
- Le chef de projet « Production » qui appartient au pôle « Gestion Opérationnelle Projet ». Il permet de coordonner et faciliter les relations entre les équipes de la division de l'Édition (dont nous faisons partie) et les équipes de la division de la Production. Nous pouvons qualifier sa fonction de « facilitateur ».
- Une autre personne de GCE Business Services intervient également lors des recettes avec les utilisateurs : l'assistance MOA, elle assure la préparation des recettes, assiste les utilisateurs et effectue le reporting des recettes.
- Des responsables d'application de gestion, ces applications qui livrent les données sous forme de CRI (Compte Rendu d'Inventaire) avant l'interfaçage. Ces responsables sont en liaison avec le chantier z/OS qui coordonne les évolutions à apporter dans les applications de gestion.
- Des personnes de l'industrialisation. Les personnes sont différentes en fonction des technologies : il y a une personne pour le chantier z/OS et une pour le chantier Unix/Windows. L'industrialisation permet l'intégration des composants développés par un projet dans des chaînes de traitement automatisées. Ces personnes appartiennent au pôle « Gestion Applications Projet » (GAP). Ainsi ce sont ces équipes qui programment l'ordonnanceur Visual TOM (dit VTOM).
- Un administrateur de base de données (DBA) Oracle effectuant les modifications dans la base de données accédée par le logiciel Evolan Report.
- Une personne pour le montage des environnements de développement qui se trouve dans le pôle « Services Système d'Information Édition » (SSIE). Elle-même a plusieurs interlocuteurs qu'elle coordonne. Elle effectue également des installations de programme sur les environnements de l'édition.



**Figure 14 : Les ressources intervenant sur le projet**

### 1.3. Suivi de projet

Les suivis du budget et du planning ne sont pas sous notre responsabilité mais incombent au chef de projet. Néanmoins, il est important d'en avoir connaissance, notre travail doit être réalisé en conservant à l'esprit que le budget n'est pas extensible et que le respect du planning est essentiel.

#### 1.3.1. Aspect budgétaire

Les charges budgétaires calculées en jours/homme sont estimées au lancement du projet. Une répartition des charges est effectuée selon les différentes phases du projet (Conception, Réalisation, Recettes, Déploiement, Pilotage). Ces charges sont revues à l'issue de la phase de conception.

Le suivi du budget est réalisé à l'aide de l'outil Clarity, cet outil permet d'avoir une vision synthétique du consommé et du reste à faire.

#### 1.3.2. Planning

Un planning prévisionnel est réalisé au lancement du projet. Certaines dates sont données à titre indicatif et seront adaptées en fonction de la charge nécessaire à la réalisation des actions.

Des points projets hebdomadaires permettant un suivi régulier et précis de l'avancement du projet sont réalisés entre le chef de projet, le responsable du chantier z/OS et le responsable du chantier Unix/Windows. Ceux-ci permettent une réactualisation du planning de manière régulière et éventuellement la création d'alertes en cas de dérive trop importante du planning. Ces alertes sont ensuite suivies d'un plan d'actions destinées à contrer ces dérives.

## 1.4. Méthodologies et leur application au sein du projet

### 1.4.1. Théorie

Chez GCE Technologies, les projets sont gérés selon la méthodologie SDM/S et conçus selon la méthode Merise.

La méthode SDM/S utilisée classiquement pour le développement de grands projets de systèmes d'information, prévoit un découpage du projet en dix phases distinctes, comme le montre la figure 15.

- Demande de Service - Etude d'opportunité DS/EO
- Définition des besoins (DBS)
- Choix d'architecture (CAS)
- Spécifications externes (SES)
- Spécifications internes (SIS)
- Programmation (PROG)
- Tests (TST)
- Conversion (CONV)
- Installation (INST)
- Bilan

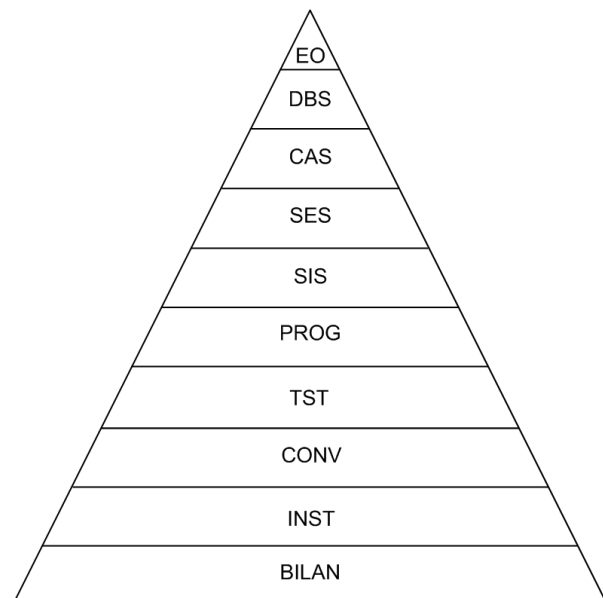
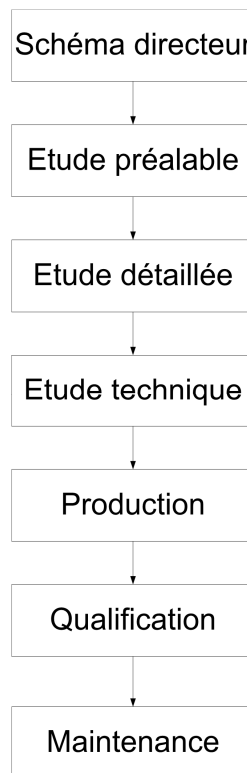


Figure 15 : Phases de SDM/S<sup>2</sup>

---

<sup>2</sup> D'après [LAF03]

Quant à la méthode Merise, la figure 16 nous montre les différentes étapes.



**Figure 16 : Les phases de Merise**

La combinaison de ces deux méthodes donne, chez GCE Technologies, les phases projets suivantes :

- L'expression de besoin : phase qui décrit les besoins de la maîtrise d'ouvrage en termes de nouveau système ou d'évolution du système.
- Le lancement du projet : cette phase permet d'identifier :
  - la portée du projet c'est-à-dire de préciser les enjeux et objectifs, les résultats attendus, le champ de l'étude (étude de l'existant)
  - Les modalités de déroulement du projet : Organisation du projet, découpages en phase, la structure du projet (ressources), l'assurance et contrôle qualité
  - Le planning du projet
- Les spécifications externes qui permettent de décrire les évolutions apportées avec une vision métier. Ces spécifications doivent être validées par la maîtrise d'ouvrage.
- Les spécifications internes qui décrivent techniquement les évolutions apportées.
- Les développements et tests unitaires associés qui permettent la réalisation des programmes nécessaires à l'évolution demandée.
- La recette utilisateurs
- La qualification
- La mise en production

### **1.4.2. Mise en pratique dans notre projet**

Si nous venons de broser les différentes phases d'un projet de développement chez GCE Technologies, nous allons maintenant préciser la démarche que nous avons retenue pour notre projet et plus particulièrement pour le chantier Unix/Windows. Des phases mentionnées précédemment ne seront pas abordées dans la suite car incombant au projet global (sous responsabilité du chef de projet) et non au chantier Unix/Windows.

Le chantier Unix/Windows concerne principalement l'installation d'une nouvelle version du logiciel Evolan Report. D'autres évolutions ont également été menées au niveau de l'intégration du logiciel, évolutions non obligatoires pour le projet SURFI mais qu'il était intéressant de mettre en œuvre simultanément.

Sopra, au regard des délais de réalisation demandés pour le projet SURFI par la Banque de France, ne pouvait livrer une nouvelle version complète rapidement. Par ailleurs les contraintes des établissements financiers au niveau de leur système d'information nécessitent le respect de jalons internes incompatibles avec une livraison tardive du logiciel. Aussi Sopra a décidé de livrer les évolutions par le biais de quatre lots étalés entre décembre 2009 et juin 2010.

Les Banques Populaires ont été volontaires pour tester la bêta version du logiciel Evolan Report. Ceci signifie qu'ils ont eu accès à des livraisons de composants en avance par rapport aux autres clients (les Caisses d'Epargne entre autres). Or les Banques Populaires sont dans le groupe BPCE ; nous avons donc bénéficié de ces livraisons avancées. Notons que les beta versions n'étant pas définitives, le travail réalisé sur celles-ci a été considéré comme prévisionnel et n'a pu être validé que lors de la réception officielle des composants.

Compte-tenu des contraintes liées aux dates de livraison et au lotissement du logiciel par l'éditeur, la démarche méthodologique que nous avons retenue pour le chantier Unix/Windows est la suivante :

- 1<sup>ère</sup> phase : Etude d'impact réalisée à réception de la beta version. Cette phase nous a permis de produire une liste exhaustive des modifications du système à entreprendre et donc des ressources à mobiliser pour les réaliser.
- 2<sup>ème</sup> phase : Plan d'action. Construit à partir de l'étude d'impact, ce plan d'action nous a permis de tracer une ligne directrice dans l'évolution du système.
- 3<sup>ème</sup> phase : réalisation et tests unitaires. Lors de cette phase, nous nous sommes directement appuyés sur le plan d'action de la phase précédente. Cette phase de réalisation nous a permis de découvrir que certaines actions n'avaient pas été prévues. Dans ces cas, nous sommes retournés dans la phase 2 pour mettre à jour le plan d'action.

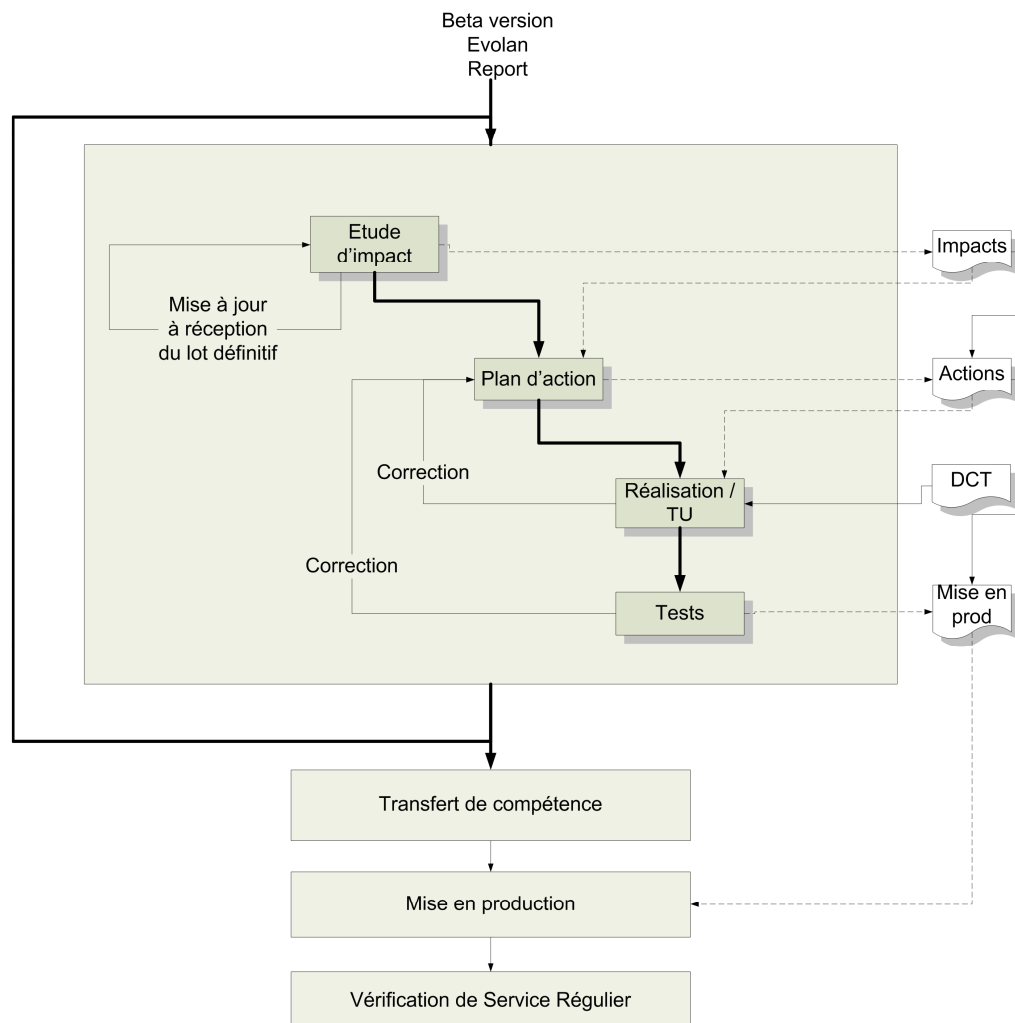
Notons, qu'outre le plan d'action, une autre documentation permet la réalisation de l'ordonnanceur Visual TOM (autrement dit VTOM, il s'agit d'un outil permettant le lancement automatique de traitements selon des conditions prédéfinies). Cette documentation sera décrite plus précisément un peu plus loin.

- 4<sup>ème</sup> phase : Recettes. Phase permettant de tester la non régression et le bon fonctionnement des évolutions apportées. Une fois cette phase réussie, le plan d'action est alors adapté pour devenir une procédure de mise en production. Cette phase permet de valider la fin du chantier Unix/Windows et de lancer les recettes utilisateurs.

Ces quatre phases sont réitérées à chaque nouvelle livraison d'un lot Evolan Report. Les documents créés lors de la première itération (étude d'impact, plan d'action et procédure de mise en production) sont mis à jour à chaque itération.

Une fois sortis de ces itérations nous disposons d'une version logicielle finale qui sera installée en production et que nous pouvons désormais présenter à l'équipe de maintenance qui en aura la charge.

Une dernière phase intervient après la mise en production, il s'agit de la vérification de service régulier (VSR). C'est une période de garantie qui engage l'équipe projet à intervenir en tant qu'expert si des anomalies surviennent sur la nouvelle application en production. La période de VSR est déterminée conjointement entre l'équipe projet et l'équipe de maintenance. Elle est généralement d'un mois.



**Figure 17 : Méthodologie chantier Unix/Windows**

Nous observons que le lotissement du logiciel Evolan Report nous a poussé à adapter la pratique méthodologique de GCE Technologies notamment en y ajoutant un système d'itérations déclenchées par l'arrivée des nouveaux lots du logiciel.

Cette méthodologie nous a permis de gagner du temps en disposant d'un plan d'actions quasiment finalisé avant l'arrivée officielle des composants du logiciel. Cette avance a été mise à profit pour mobiliser rapidement les ressources devant intervenir sur le sujet et ainsi garantir de manière sûre les délais de réalisation.

Nous avons également gagné du temps lors de la transformation des documents : étude d'impact → plan d'actions → mise en production. Le passage naturel d'un document à l'autre a été source de gain de temps.

### 1.5. Planning du chantier Unix/Windows

Le planning décrit dans la figure 18 nous montre que les différentes phases du chantier Unix/Windows ont démarré à des dates différentes mais qu'elles se poursuivent toutes quasiment jusqu'à la fin du projet. Le lotissement de la livraison des composants explique cette caractéristique.

Notons également que la phase de test, une fois démarrée ne s'interrompt pas. En effet cette phase est plus marquée après la réalisation des lots mais reste en tâche de fond tout le long du projet. Cela démontre l'importance de cette tâche.

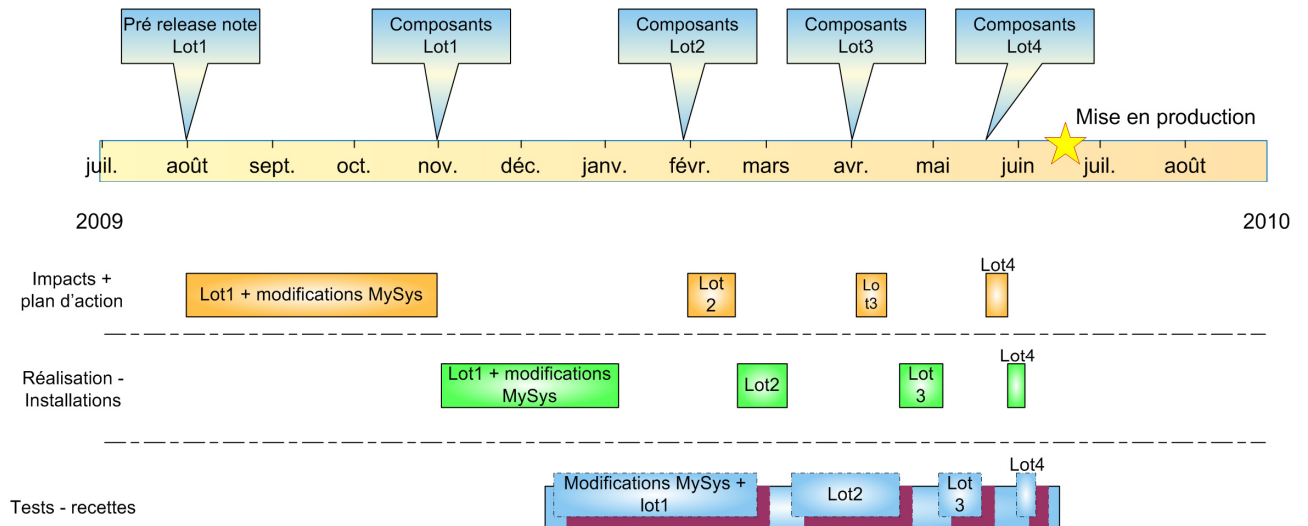


Figure 18 : Planning du chantier Unix/Windows

## ***2. Un environnement technique à construire***

---

L'arrivée du projet SURFI a conduit à mettre en place un environnement technique complet et fiable permettant d'amener sécurité et industrialisation absents jusque là.

### **2.1. Mise en place de la configuration logicielle**

La gestion des composants de l'application BAFI chez GCE Technologies a toujours été faite manuellement, sans processus industrialisé, à la discrétion des responsables d'application. Cet état de fait entraîne des installations en production entièrement manuelles, complexes, fastidieuses et risquées.

De nombreuses applications Unix de GCE Technologies sont dans le même cas. Un projet de mise en GCL (Gestion de Configuration Logicielle), autrement nommé mise en référentiel, de l'ensemble des applications de l'entreprise a été lancé. Nous nous sommes donc insérés dans ce projet afin de mettre en référentiel l'application BAFI et pouvoir bénéficier des outils d'installation automatiques.

#### ***2.1.1. La Gestion de Configuration Logicielle***

La GCL permet de connaître à chaque instant l'état de chaque composant d'une application. Elle permet de valider des états stables pour ces composants (lors de mise en production par exemple), lorsque nous disposons de l'ensemble des composants dont l'état est stable, cela constitue une version de l'application.

Le développement en équipe est facilité par la GCL puisqu'elle détecte les conflits (dans le cas où deux développeurs souhaitent modifier en même temps un même composant). De même il peut coexister plusieurs lignes de versions facilitant la coordination entre les équipes de développement et les équipes de maintenance.

Une GCL permet également de mettre à disposition des outils d'installation automatique facilitant les mises en production. En cas de nécessité, il est possible d'installer une version complète de l'application notamment dans le cas de crash informatique.

Une GCL, de par sa gestion d'états stables référencés et connus par les équipes de développement, permet un accroissement de la sécurité lors des déploiements en production.

Chez GCE Technologies, la GCL des applications Unix et Windows est assurée par le logiciel CM-Synergy de Télélogic.



### 2.1.2. Evolan Report et CM-Synergy, une étude fonctionnelle

Sous l'appellation Evolan Report nous parlons du logiciel livré par l'éditeur mais aussi des composants développés par GCE Technologies autour du logiciel pour l'intégrer à la plateforme MySys ; cela va jusqu'à la documentation.

La mise en GCL d'une application nécessite d'effectuer un découpage fonctionnel. Chaque partie fonctionnelle fait l'objet d'un « Project », notion CM-Synergy qui permet une organisation cohérente des composants. Chaque composant appartient à un domaine fonctionnel et donc à un « project ».

Ainsi nous avons effectué une revue complète de l'application et déterminé les différents découpages fonctionnels. Nous avons identifié cinq domaines fonctionnels. Une fois ce découpage fonctionnel effectué, nous avons, pour chaque « project » pratiqué un autre découpage cohérent à plusieurs niveaux pour organiser au mieux les composants. Ce deuxième découpage peut s'apparenter à un système de répertoires dans lesquels les composants seront rangés. Pour cela, nous avons choisi d'une part de découper par type de composant (KSH, exécutable, paramétrage, cobol, SQL, etc.) et d'autre part en fonction de la provenance du composant (Editeur ou GCE Technologies) ;

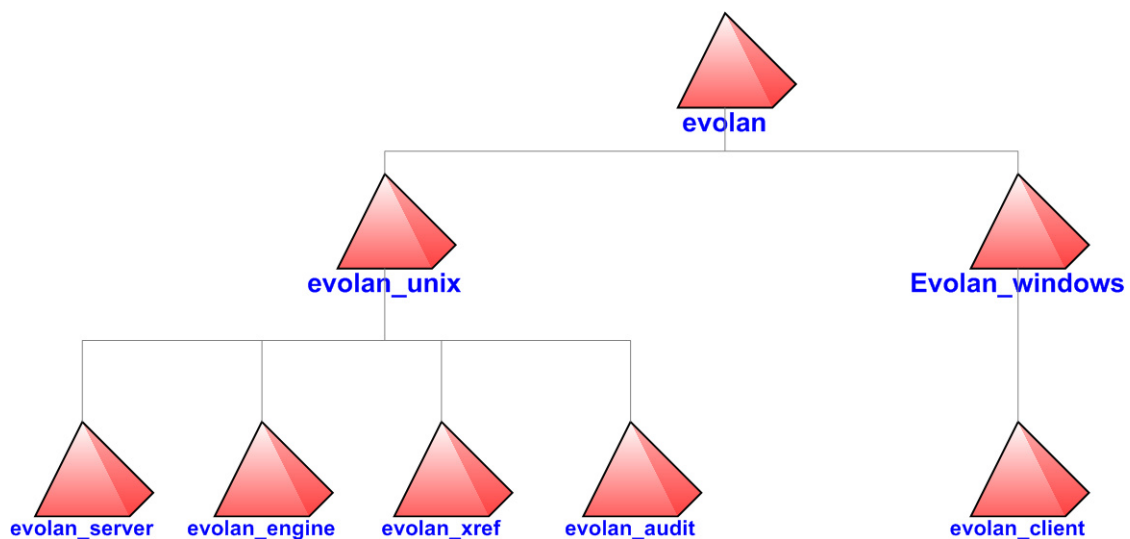


Figure 19 : Découpage fonctionnel Evolan Report

La figure 19 nous montre les cinq parties fonctionnelles correspondant aux quatre modules Evolan Report plus l'audit. Nous pouvons remarquer que ces cinq « project » sont regroupés dans deux projets (evolan\_unix et evolan\_windows) qui sont purement techniques et propres à l'outil de GCL ; ils correspondent aux socles techniques.

L'étude s'est terminée par le recensement de l'ensemble des composants et leur affectation dans chacun des répertoires. A titre d'exemple, nous présentons un extrait du découpage effectué dans le « project » evolan\_engine dans la figure 20.

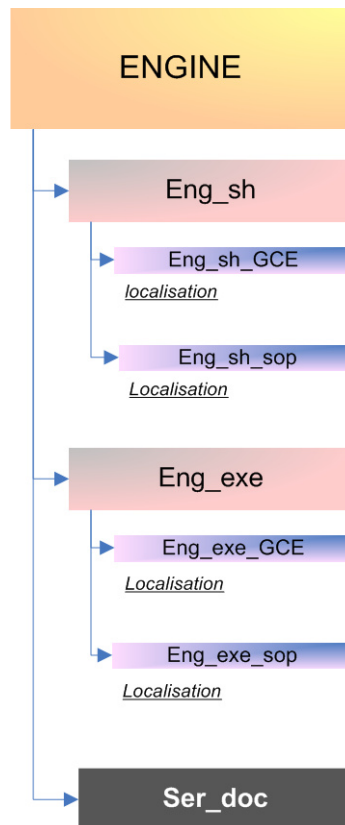


Figure 20 : Extrait du découpage CM-Synergy "Project" Engine

Dans cet extrait, nous présentons deux répertoires : ENG\_sh (qui correspond à des composants en ksh) et Eng\_exe (composants exécutables). Dans chacun des deux répertoires, nous retrouvons des sous-répertoires : un pour les composants en provenance de GCE Technologies et un autre pour ceux en provenance de l'éditeur, Sopra. La mention « Localisation » sous chacun des sous-répertoires remplace le chemin sur la machine de production où sont localisés les composants concernés. Cette information, stockée dans le logiciel de GCL permettra une installation automatisée. A noter la présence d'un répertoire nommé « Ser\_doc » qui contiendra toute la documentation afférente au module Server.

### **2.1.3. Mise en référentiel des composants**

Une fois le découpage fonctionnel effectué, l'équipe d'administration de la GCL ayant créé les « project », nous avons effectué la mise à jour de la GCL avec l'ensemble des composants, sur une version de test. Nous étions alors prêts à basculer les composants sur la version de production.

Le projet de mise en GCL s'est arrêté là pour des problèmes de disponibilité de ressource. La mise en référentiel sur la machine de production est reportée à une date ultérieure.

### **2.1.4. Bilan de la mise en GCL**

Le bilan que nous tirons de ce début de mise en GCL est en demi-teinte. Si l'analyse qui a été faite sur le découpage fonctionnel et le recensement des composants a permis d'acquérir une connaissance de l'application que nous n'avions pas et qui nous sert pour le projet SURFI, l'échec de la mise en

référentiel s'avère dommageable et entraîne des difficultés supplémentaires pour l'installation en production du projet SURFI. Nous y reviendrons dans le bilan.

## **2.2. Montage d'environnements de recette**

Comme nous l'avons déjà précisé, l'application BAFI était dévolue à une autre équipe avant la fusion des trois plateformes informatique en une seule, MySys. L'équipe qui en avait la charge précédemment n'avait pas ressenti la nécessité de disposer d'environnements de développement et de recette, ce qui n'est pas notre cas. Le projet SURFI nous pousse à demander la création de ceux-ci.

Outre ces demandes émanant de l'équipe maîtrise d'œuvre, la maîtrise d'ouvrage du projet SURFI a émis le souhait de disposer d'un environnement de volumétrie réelle du fait de l'ampleur du projet. Celui-ci doit permettre aux utilisateurs d'effectuer une recette exhaustive de l'ensemble des restitutions réglementaires d'un ou deux établissements.

Nous voici donc dans une phase de montage de plusieurs environnements.

### **2.2.1. Demandes d'environnement**

#### **▪ Environnement de recette**

Ayant repris la responsabilité de l'application et le projet SURFI arrivant, il nous fallait disposer d'environnements nous permettant de minimiser les risques lors des mises en production. Nous avons donc effectué une demande d'environnements : un environnement de développement (permettant d'effectuer les tests unitaires), un environnement de recette utilisateurs (environnement conforme à l'architecture de production et permettant les recettes). Cette demande s'est traduite par la présentation d'un dossier technique ainsi qu'une argumentation pour justifier d'un tel besoin.

Après arbitrage, nous avons eu l'accord pour le montage d'un environnement de recette utilisateurs uniquement.

Le fait de ne pouvoir disposer d'un environnement de développement nous oblige à effectuer les tests unitaires sur l'environnement de recette utilisateurs amenant une difficulté supplémentaire dans le projet. En effet, l'environnement de recette utilisateurs n'est pas sous notre responsabilité contrairement à un environnement de développement. Les tests unitaires nécessitent des allers-retours de composants, installation, test, correction puis de nouveau installation. L'installation sur l'environnement de recette se fait par une tierce personne et allonge les délais de réalisation. Enfin, lors des recettes, l'environnement n'est pas disponible pour effectuer des développements.

#### **▪ Environnement de volumétrie réelle**

La démarche pour effectuer la demande est la même que ce que nous venons de décrire à ceci près que la charge nécessaire à la mise en place d'un tel environnement est conséquente, d'autant plus qu'il s'agit d'un environnement temporaire.

Voici les différentes phases nécessaires au montage de cet environnement :

- Pour la partie z/OS :
  - Mise à disposition d'une partition z/OS
  - Installation des chaînes de traitement z/OS nécessaires
  - Extraction des données du ou des établissements sur la machine de production pour réinjection sur la partition z/OS
- Pour la partie Evolan Report :
  - Mise à disposition d'un serveur UNIX
  - Installation de la base de données Oracle
  - Installation de la dernière version du logiciel Evolan Report
  - Installation d'un ordonnanceur de traitement sur UNIX
  - Mise à disposition d'une machine Windows pour supporter le module client Evolan Report
- Une chaîne de liaison complète entre la machine z/OS et Unix, entre le z/OS et le serveur GEIDE (gestion électronique de documents),
- La mise en place de postes Windows où sera installé le client Evolan ainsi que Business Object et à partir desquels les utilisateurs pourront se connecter
- Une campagne de test de recevabilité de l'ensemble de l'installation

Une telle demande est atypique et coûteuse, notamment pour la partie z/OS. Un arbitrage pris lors d'un comité de direction a entériné la décision.

Un élément décisif a également fait pencher la balance : un autre projet avait émis une demande équivalente. Il a donc été décidé de mutualiser l'environnement z/OS en volumétrie réelle entre les deux projets. S'en sont suivis des rendez-vous de coordination, notamment au niveau des dates d'utilisation, entre les deux projets.

### ***2.2.2. Montage des environnements***

Le montage de tout environnement est effectué par des équipes du SSIE (Service Système d'Information Edition) mais nous avons été en support expertise et avons travaillé main dans la main pour le montage. Cette collaboration efficace a permis de monter l'environnement rapidement et sans difficulté majeure.

La conjugaison de notre connaissance de l'application et de la compétence du SSIE a permis de constituer une documentation d'installation à utiliser lors du montage de l'environnement de volumétrie réelle ou de tout autre environnement. Cette capitalisation de l'expérience a amené un gain de temps non négligeable dans le déroulement du projet SURFI.

### ***2.2.3. Recevabilité des environnements, procédure de tests***

Une fois l'environnement de recette utilisateurs monté, nous avons mis sur pied une procédure de tests permettant de prononcer la recevabilité de l'environnement. Cette procédure, décrite dans un document à cette occasion, est réutilisable au cours du projet. La mise en place de cette procédure nous a permis d'acquérir une connaissance approfondie du fonctionnement de l'application. Cette expérience nous a été utile lors des tests et recettes du projet SURFI. Cela

nous a apporté un gain de temps non négligeable sur la résolution des problèmes lors des traitements.

#### **2.2.4. Bilan du montage des nouveaux environnements**

Outre la sécurisation du processus de développement de l'application BAFI, le montage de ces nouveaux environnements a été un vrai bénéfice dans le cadre de notre projet SURFI. En effet, d'une part nous avons mis sur pied une stratégie de test efficace, d'autre part, nous avons renforcé nos compétences sur le fonctionnement de l'application Evolan Report et amené un gain de temps sur le projet SURFI.

Une réserve toutefois concerne l'environnement de volumétrie réelle et la mise en place des transferts entre machines ; plus particulièrement le transfert permettant l'envoi des états vers le serveur de gestion électronique de document. Les équipes responsables ont rencontré des difficultés techniques et ont mis un certain temps à les résoudre. A notre sens, nous avons perdu du temps sur cet aspect car nous avons eu du mal à mobiliser cette équipe située sur un site distant. Un suivi plus précis de leur planning aurait permis de gagner du temps.

### **3. Utilisation d'un nouveau langage : XBRL**

---

Le projet introduit l'utilisation d'un nouveau format de remise des états réglementaires basé sur le langage XBRL. Voici une description de XBRL.

#### **3.1. Introduction au langage XBRL**

##### **3.1.1. Présentation**

XBRL, (eXtensible Business Reporting Language) est un standard de publication électronique des informations financières. XBRL, ouvert et libre car basé sur le XML (eXtensible Markup Language), a été développé par « XBRL International », un consortium de quatre cents entreprises à la fin des années 90.

XBRL remplace les rapports financiers en format texte et propriétaire par des rapports au format électronique lisible par des programmes informatiques. Les destinataires peuvent ainsi, grâce à des logiciels, convertir ces fichiers en rapports financiers lisibles par l'homme et incluant des notions de nommage, hiérarchies et liens des données. Par ce biais, il est aisé d'effectuer des recherches de données et d'effectuer des comparaisons (dans le temps pour une même société ou entre différentes sociétés).

##### **3.1.2. La taxonomie**

La force de XBRL réside dans sa taxonomie. Elle permet de définir la liste des champs, leurs attributs, les liens entre eux, qu'une société utilise pour constituer ses rapports. Une taxonomie robuste, bien définie et stable permet une plus grande précision ainsi qu'une comparaison efficace entre les rapports d'une société. Chose que l'on ne retrouve pas avec les divers formats utilisés actuellement. Une taxonomie permet également la définition des contrôles qui peuvent être appliqués de manière automatique sur les données financières.

Chaque élément décrit dans la taxonomie incorpore des métadonnées, c'est-à-dire des informations qui donnent des détails sur la donnée traitée. Ces métadonnées permettent notamment de décrire la façon dont la donnée a été constituée, son rôle dans le calcul d'autres champs ou encore des références à des textes réglementaires. Ces métadonnées permettent une recherche des données efficace et précise.

Les taxonomies peuvent être fixes ou flexibles. Une taxonomie est fixe notamment lorsqu'elle est fournie par des autorités de tutelle telles que la Banque de France. Dans ce cas, cela permet de récupérer de l'information dans un format standardisé (imposé). Les taxonomies flexibles sont utilisées parce que les régulateurs de marchés financiers s'appuient sur des sociétés hétérogènes. Les données à fournir sont forcément flexibles pour permettre aux sociétés de présenter les chiffres parlant pour elles au regard de leur secteur d'activité.

### 3.1.3. Structure d'un document XBRL

Les données d'un fichier de déclaration XBRL sont enveloppées par les balises <XbrlDelivery> et <XbrlDeclarationReport>.

<XbrlDelivery> est la balise racine du fichier. La balise <Administration> constitue la « tête » d'une instance ; nous entendons par tête le début du reporting d'une instance. Cela est important dans le cas de remontées multi-instances. Sur MySys, les fichiers XBRL sont tous mono-instance, il y a donc une seule tête par fichier XBRL.

Dans la figure 21, nous trouvons un exemple d'organisation des données pour une instance XBRL :

```
<XbrlDelivery>
  <XbrlDeclarationReport xmlns="http://www.onegate.eu/2010-01-01">
    <Administration creationTime="2009-12-01T14:33:12.546">
      <From declarerType="CIB">22222</From>
      <To>BDF</To>
      <Domain>SUR</Domain>
      <Response feedback="true">
        <Email>bafi@bpce.fr</Email>
        <Language>FR</Language>
      </Response>
      <CustomParameters>
        <Dim prop="og_txn_vrs">Surfip-2010-06-30</Dim>
      </CustomParameters>
    </Administration>
    <Report code="SURFI_P_103">
      Instance1 XBRL
      (Ensemble de cellules valorisées de tableaux SURFI de la
      taxonomie Principale)
    </Report>
  </XbrlDeclarationReport>
</XbrlDelivery>
```

Figure 21 : Document XBRL

Voici une description succincte des différentes balises de la tête : :

- `<creationTime>` : Cette propriété reprend la date de création du fichier, ce champ, généré automatiquement par l'application Evolan Report, va permettre à BPCE de reconstituer la chronologie des déclarations envoyées par les Caisses d'Epargne. Le principe étant : le dernier fichier créé est la version prise en compte pour l'envoi à la Banque de France.
- `<From>` : représente l'acteur économique assujéti à la remise. Dans notre cas il s'agit du CIB (Code Inter Bancaire donnée par la Banque de France à chaque établissement de crédit) de chaque Caisse d'Epargne.
- `<To>` : identifiant qui caractérise le guichet Onegate, la valeur est fixée à BDF.
- `<Domain>` : représente le code de la collecte auquel l'instance fait référence. Ses valeurs sont sur trois caractères. Une remise SURFI a un code égal à « SUR ».
- `<Response>` : contient l'adresse de l'émetteur et le langage de la notification adressée. Une balise à « true » ou « false » indique respectivement si le destinataire souhaite être notifié ou non par e-mail dès réception par le guichet du fichier. Dans notre cas l'émetteur est la BPCE puisque c'est elle qui déposera toutes les instances de toutes les Caisses d'Epargne auprès de la Banque de France.
- `<Report>` : contient les données métier. Pour le cas d'une collecte XBRL, le format contenu dans cette balise est une instance XBRL. Dans les autres cas le contenu est XML.
- `<Custom Parameters>` : contient des informations complémentaires d'administration.

### 3.2. Les enjeux du langage XBRL

Depuis plusieurs années les coûts du reporting financier ont fortement augmenté. XBRL semble être une solution pour les diminuer et accroître la rapidité de constitution des rapports financiers. En effet, XBRL évite la ressaisie d'information et permet de définir précisément la sémantique, le contexte ainsi que d'éventuelles remarques de l'entité effectuant le rapport. XBRL est mis en place pour apporter une diminution des coûts.

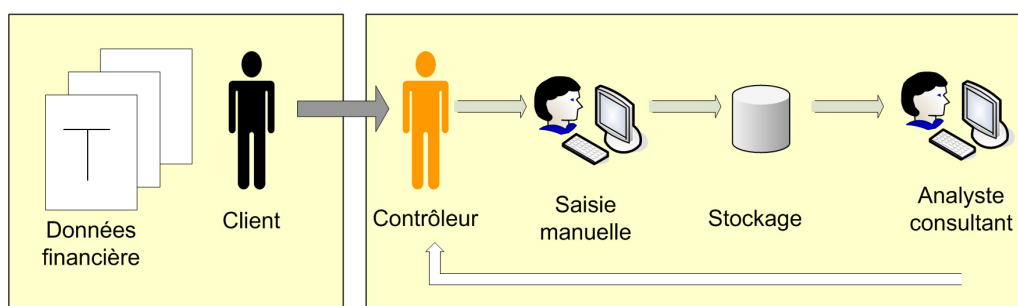


Figure 22 : Avant XBRL, beaucoup de travail manuel<sup>3</sup>

<sup>3</sup> D'après [XBR06]

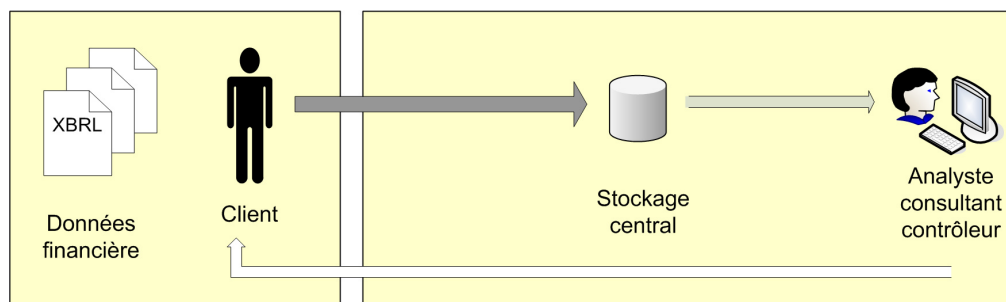


Figure 23 : Avec XBRL, parfaitement défini et prêt pour l'automatisation métier<sup>4</sup>

### 3.3. Les contrôles

#### 3.3.1. A l'origine la BAFI..

La BAFI possède des contrôles qui sont intégrés dans les outils et notamment Evolan Report. Au nombre de neuf mille, ces contrôles sont intégrés via une interface de paramétrage. Depuis les débuts de la BAFI, ces contrôles ont été présentés sous une forme littérale et intégrée dans le CD-ROM BAFI, les mises à jour se faisant via une « lettre d'information BAFI ».

Le déclenchement des contrôles est conditionné afin de tenir compte du système financier de remise, de sa catégorie et de la présence ou non de montants. Nous le voyons, il s'agit d'un système de contrôle complexe, lourd à mettre en place et à mettre à jour.

#### 3.3.2. Taxonomie SURFI

Une taxonomie XBRL permet, nous l'avons vu, d'intégrer des contrôles qui sont exécutés automatiquement. Ces contrôles peuvent être nombreux et complexes. Les mises à jour sont facilitées et rapidement intégrables dans les solutions logicielles des éditeurs grâce à la livraison de nouvelles versions de taxonomie.

La taxonomie SURFI utilise cette fonctionnalité et chaque contrôle est associé à un code et à un libellé. Des messages d'assertion ont été mis en place. Ils permettent de donner des explications en cas de contrôle insatisfaisant.

## 4. Des analyses et des solutions

Nous avons présenté l'équipe projet, la méthodologie appliquée et évoqué la mise en place d'un environnement technique permettant un travail efficace à la fois pour la maîtrise d'ouvrage et pour la maîtrise d'œuvre.

Nous allons maintenant nous intéresser à une phase du projet déterminante pour la suite : l'analyse. L'analyse, une phase qui a été mise en œuvre plusieurs fois sur des sujets différents et qui nous a permis de mettre en place des solutions, de prendre des décisions concernant la suite du projet.

<sup>4</sup> D'après [XBR06]



## **4.1. La vision utilisateurs**

En début de projet nous avons rencontré des utilisateurs de la Caisse d'Epargne Ile de France. Nous pouvons considérer que cette action est importante et représente une forme d'analyse. Les conséquences sont loin d'être négligeables pour le projet, ceci pour plusieurs raisons :

- Une telle prise de contact favorise la communication entre maîtrise d'œuvre et utilisateurs et favorise les échanges.
- La démonstration du travail des utilisateurs peut permettre à la maîtrise d'œuvre de comprendre plus facilement les intérêts des utilisateurs, leurs contraintes, et donc de choisir des solutions adaptées. L'organisation de GCE Technologies ne prévoit pas une telle démarche, la maîtrise d'ouvrage est là pour remplir ce rôle mais l'opportunité qui nous a été offerte a été positive.
- Dans notre cas, de nombreuses ombres demeuraient quant aux méthodes utilisées pour envoyer leurs fichiers de déclaration BAFI vers BPCE. Ces ombres ont été levées et ce déplacement nous a permis de bien comprendre le fonctionnement global de la BAFI dans les Caisses d'Epargne.

## **4.2. Etude d'impact**

Cette étape était déterminante pour la suite de notre projet car elle définissait toutes les actions à entreprendre pour mettre en place les évolutions du système BAFI. L'étude d'impact a été véritablement le fil rouge de notre projet sur le chantier Unix/Windows.

### **4.2.1. Lotissement de l'éditeur**

Nous l'avons déjà évoqué, une des difficultés du projet résidait dans le lotissement du logiciel Evolan Report de la part de Sopra. Ainsi ce n'est pas moins de quatre lots de composants qui ont été livrés.

Outre les livraisons de composants, d'autres livraisons interviennent dans le temps, il s'agit de la documentation. En effet, Sopra, à plusieurs reprises, livre de la documentation avant la phase de livraison des composants. Cela nous a amené à anticiper le travail à réaliser et ainsi gagner du temps.

Nous l'avons déjà mentionné, GCE Technologies profitait des livraisons des versions beta du fait que les Banques Populaires sont pilotes du projet et appartiennent au groupe BPCE. Ceci permettait également une anticipation non négligeable.

Nous le voyons sur la figure 24, les livraisons se sont échelonnées depuis juillet 2009 à juin 2010 pour une mise en production le 21 juin 2010.

Nous pouvons également remarquer que la dernière version officielle intervient à moins d'un mois de la mise en production ce qui nécessite la mise en place d'un plan d'action particulier pour celle-ci. En effet, les jalons imposés par le processus de GCE Technologies pour la mise en production de la version de juin 2010

imposent que le projet ait effectué ses recettes utilisateurs pour la mi-avril. Or des recettes utilisateurs vont intervenir après le premier juin. Le projet SURFI étant réglementaire, il ne peut être refusé. Mais une sécurisation des dernières évolutions doit être entreprise.

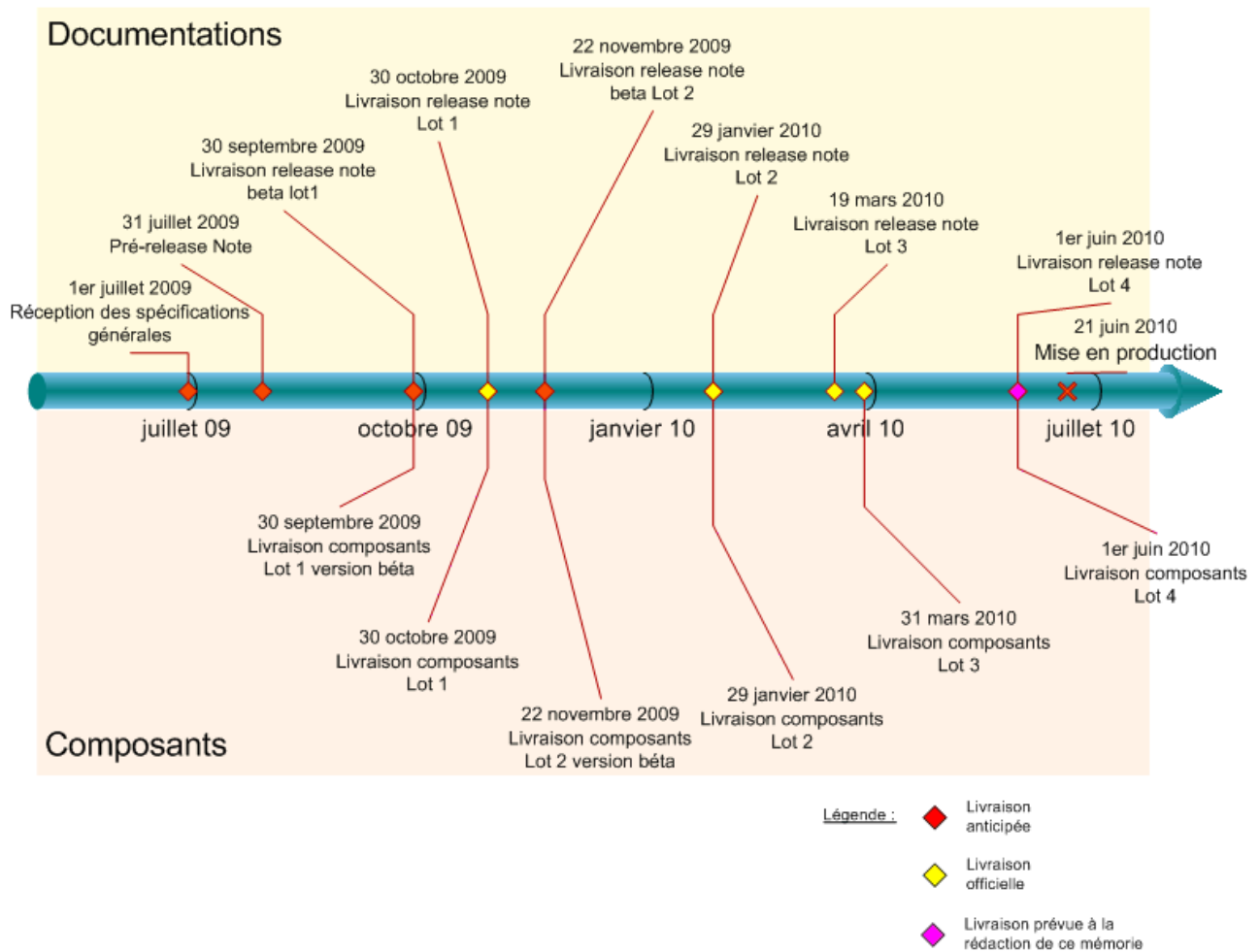


Figure 24 : Planning des livraisons de l'éditeur Sopra

#### 4.2.2. Elaboration de l'étude d'impact

La réception début juillet 2009 des spécifications générales n'apportait que très peu d'éléments sur les évolutions apportées au logiciel Evolan Report et les impacts dans les différents modules. Nous avons donc demandé à Sopra de nous livrer une documentation plus complète qui nous permette de produire un premier jet d'une étude d'impact. La réception de la pré-release note du 31 juillet 2009 nous a satisfait et nous a donc permis de rédiger une première version de l'étude d'impact.

Le but de ce premier jet était essentiellement d'identifier et de mobiliser les ressources qui doivent intervenir sur le projet. En effet, l'équipe projet présentée précédemment n'a pas eu la possibilité d'effectuer toutes les évolutions, certaines opérations étaient sous responsabilité d'autres personnes. Nous pouvons citer pour exemple les modifications de la base de données Oracle. Seules les équipes d'administration des bases de données sont habilitées à intervenir sur les bases de données.

A la réception officielle des composants, le premier jet de l'étude d'impact est revu et corrigé. La réalisation peut commencer car les ressources sont prêtes à intervenir depuis le premier jet.

#### **4.2.3. Itération**

Nous l'avons vu, la multiplication des livraisons de la part de Sopra nécessite de multiples itérations sur cette étude d'impact. Celle-ci est en effet enrichie à chaque nouvelle livraison.

### **4.3. Réorganisation des traitements**

Nous venons de voir que nous avons mis en œuvre deux types d'analyse différents, le rapprochement des utilisateurs et l'étude d'impact. Un autre type d'analyse est mis en œuvre au cours du projet, il s'agit de vérifier une hypothèse de modification de l'architecture des traitements d'une part et d'autre part de valider la possibilité de supprimer une gestion des priorités bridant le travail des utilisateurs.

#### **4.3.1. Expression de besoins**

Le besoin exprimé en premier lieu par l'équipe de maintenance est le suivant : « Est-il possible de reconstituer le traitement Engine en un seul bloc ? ». En effet, le traitement Engine livré par Sopra est constitué d'un programme principal. Celui-ci a été découpé en quatre programmes distincts par l'équipe précédente. Cette possibilité permettrait à l'équipe de maintenance d'être moins confrontée à des dysfonctionnements difficiles à reprendre du fait du découpage des traitements.

Le deuxième besoin, également exprimé par l'équipe de maintenance est : « Suppression de la gestion des mises en attentes des demandes ». Cette fonctionnalité empêche de traiter plusieurs demandes de même nature dans un même traitement Engine (deux fermetures de sessions par exemple).

Nous allons expliquer ces deux besoins et la façon dont nous les avons traités.

### 4.3.2. Restructuration des traitements

#### ▪ Analyse de l'existant

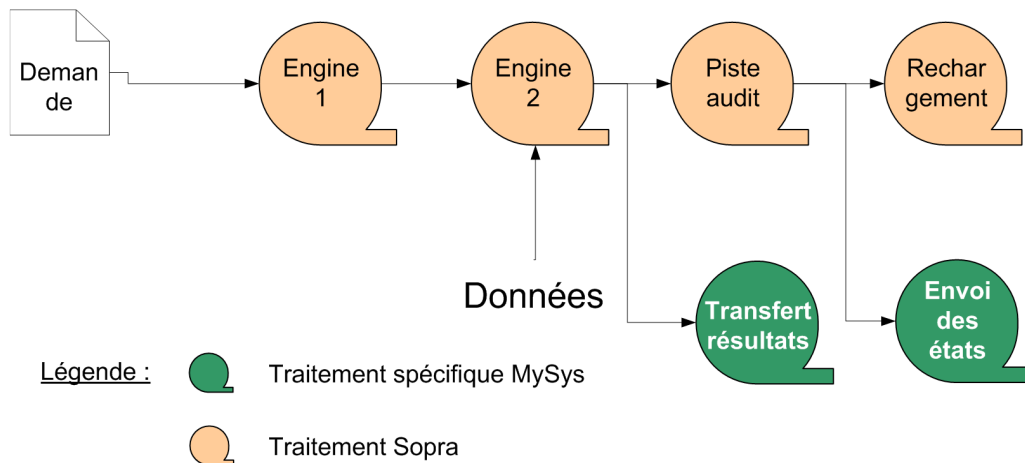


Figure 25 : Traitement Engine existant

La figure 25 nous donne une description simplifiée du traitement Engine tel qu'il était avant le projet SURFI (pour des raisons de lisibilité, seuls les programmes principaux ont été mentionnés). Voici la fonctionnalité de chaque programme découpé :

- Engine 1 : permet l'initialisation technique du traitement Engine
- Engine 2 : permet la production des résultats de fabrication à partir des données en provenance du z/OS et selon les demandes mises en entrée du traitement
- Piste d'audit : permet l'alimentation d'une piste d'audit interne au logiciel Evolan Report
- Rechargement : permet l'alimentation de la base de données Engine avec les données utilisées pour traiter chaque session. Ces données sont ensuite accessibles par l'utilisateur pour consultation et éventuellement correction.

Le découpage du programme principal a été effectué pour plusieurs raisons. Tout d'abord cela devait permettre d'apporter un gain de temps dans les traitements. Ensuite cela devait procurer une certaine souplesse dans le travail des utilisateurs.

Le gain de temps se fait au niveau du programme « Engine1 » qui n'a pas besoin des données du z/OS. Ces données arrivent tard dans la nuit, le démarrage anticipé du traitement (lancement du programme Engine 1 avant l'arrivée des données) n'est plus à faire lorsque les données arrivent, cela représente un gain de temps. Au regard de la puissance actuelle des machines, nous devons analyser si cela est toujours d'actualité et quelles seraient les conséquences si ce montage était supprimé.

La souplesse de travail des utilisateurs est apportée par deux éléments :

- Les résultats de fabrication sont envoyés vers le module Server avant le début des deux derniers programmes (Piste d'audit et Rechargement). En cas de problème dans ceux-ci, l'utilisateur dispose des résultats de fabrication et peut travailler même s'il ne dispose pas des états GEIDE et de la piste d'audit.
- Nous retrouvons le même principe concernant l'envoi des états GEIDE et des données utilisées en consultation.

La suppression du découpage permettrait un travail de la maintenance facilité et des installations de composants plus rapides car conformes à la livraison. En effet, pour ce dernier point, toute nouvelle livraison par Sopra du programme principal d'Engine nécessite une analyse approfondie pour déterminer les impacts dans les différents programmes découpés ce qui est consommateur de temps.

### ▪ Simulation

Pour faciliter notre prise de décision, nous nous sommes intéressés aux temps de traitements de chacun des quatre programmes et avons construit un diagramme de GANTT.

MySys supporte douze Caisses d'Epargne qui ont chacune leur volumétrie et donc des temps de traitement différents. Nous avons décidé de prendre pour chaque programme, les temps de traitement les plus élevés constatés. Notre échantillon de recherche a porté sur quatre mois (de juin 2009 à août 2009).

La figure 26 nous montre le diagramme dans la situation actuelle, avec le découpage.

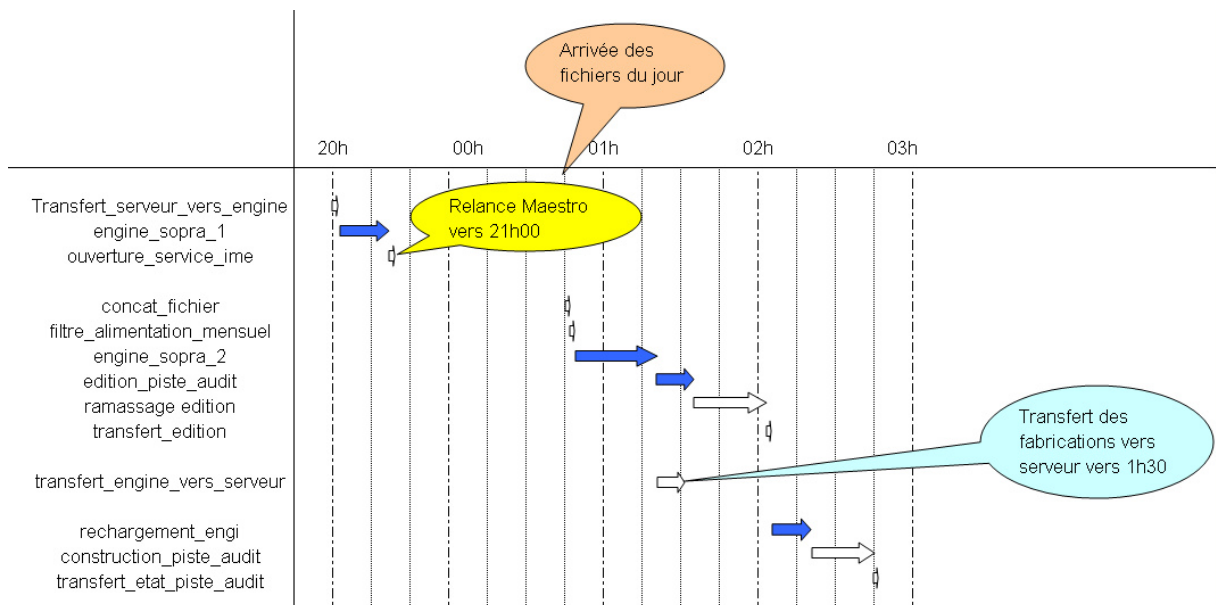


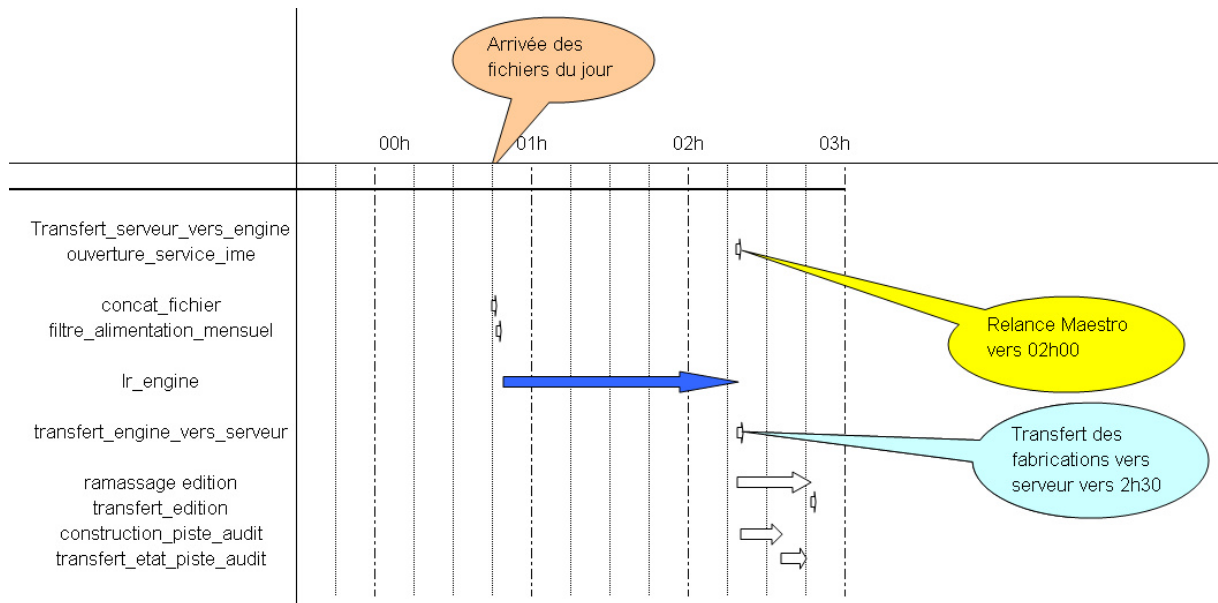
Figure 26 : Diagramme de GANTT traitements Engine situation actuelle

Les flèches en bleu représentent les quatre programmes découpés issus de livraisons de Sopra. Les autres flèches sont des programmes spécifiques MySys.

Nous observons une fin des traitements aux alentours de trois heures du matin.

A partir de là, nous avons confectionné un deuxième diagramme de GANTT dans le cas où programme principal d'Engine est en un seul bloc. Basé sur les temps de traitement actuels, cela nous donne une simulation. Pour ce faire, nous avons additionné le temps des quatre programmes et modifié l'agencement des traitements MySys.

La figure 27 nous en donne le résultat, la flèche bleue représente le temps de traitement du programme principal du module Engine :



**Figure 27 : Diagramme de GANTT traitements Engine situation future**

Nous pouvons observer une fin prévue des traitements aux alentours de trois heures du matin ce qui est semblable à la situation actuelle. Par contre des différences apparaissent notamment au niveau de l'heure prévue de transfert des résultats de fabrication. Ces transferts interviennent une heure plus tard.

Dans un fonctionnement nominal cela n'a aucune conséquence. Par contre des différences apparaissent en cas de plantage du programme avant ce transfert. En effet, les équipes de maintenance n'arrivent que le matin vers huit heures. Le temps qu'elles règlent le problème et relancent les traitements, cela repousse d'autant la mise à disposition des résultats aux utilisateurs. Il en est de même pour les états GEIDE. Si cet état de fait existe dans l'ancienne situation, elle est aggravée avec la nouvelle, la mise à disposition des résultats se ferait encore plus tard.

En synthèse, le regroupement des programmes ne modifie pas l'heure de fin des traitements en cas de fonctionnement normal. Les conséquences apparaissent en cas de plantage durant la nuit, dans ce cas les résultats de fabrications arrivent une heure plus tard. Il en va de même pour les états GEIDE. Une telle modification n'est pas transparente et doit être analysée.

### ▪ **Prise de décision**

Nous ne pouvions prendre la décision nous-mêmes sans consulter les équipes de maintenance. Nous avons donc provoqué une réunion sur le sujet et présenté nos analyses quant aux possibilités de regroupement et aux conséquences. Deux solutions s'offraient à nous : conserver en l'état les traitements ou procéder au regroupement.

La première solution possède les inconvénients suivants :

- gros travail d'analyse d'impact sur les composants livrés par Sopra pour reproduire le découpage ; dans le cadre du projet SURFI mais également pour les projets à venir
- augmentation du temps nécessaire à l'installation des évolutions de Sopra
- risque majeur de dysfonctionnement
- conservation d'une capacité à maintenir difficile

Elle procure un avantage :

- pas de changement de l'heure des mises à disposition des résultats notamment lors des reprises

La deuxième solution quant à elle amène les inconvénients suivants :

- en cas de reprise, la mise à disposition des résultats se fera une heure plus tard, idem pour le redémarrage du maestro (processus à l'écoute des demandes du module Client ; il est coupé pour éviter toute production de nouvelle demande pendant la prise en compte des demandes) et la disponibilité des états GEIDE

Les avantages de la deuxième solution sont :

- un travail de la maintenance facilité car un traitement Engine unique
- une plus grande rapidité à installer les futures évolutions Sopra sur ce composant, notamment dans le cadre du projet SURFI
- la possibilité d'activer le système de reprise des traitements en cas de dysfonctionnements mis en place par l'éditeur (ce système de reprise n'est pas utilisé actuellement)

Après avoir analysé les deux solutions au cours de la réunion la décision a été prise de procéder au regroupement. Les avantages apportés par cette solution sont supérieurs aux inconvénients.

### ▪ **Une stratégie payante**

Nous voulons mettre l'accent sur les effets bénéfiques de la collaboration entre l'équipe projet et celle de la maintenance. Un travail coopératif sur le sujet a permis une prise de décision commune qui ne sera pas remise en cause lors de la mise en production. L'implication de l'équipe de maintenance très en amont dans le projet a été volontaire de notre part, ceci afin de faciliter le transfert de compétence de la nouvelle version de l'application de l'équipe projet vers l'équipe de maintenance; transfert qui intervient peu de temps avant le déploiement.

### 4.3.3. Suppression de la gestion de priorité

#### ▪ Analyse de l'existant

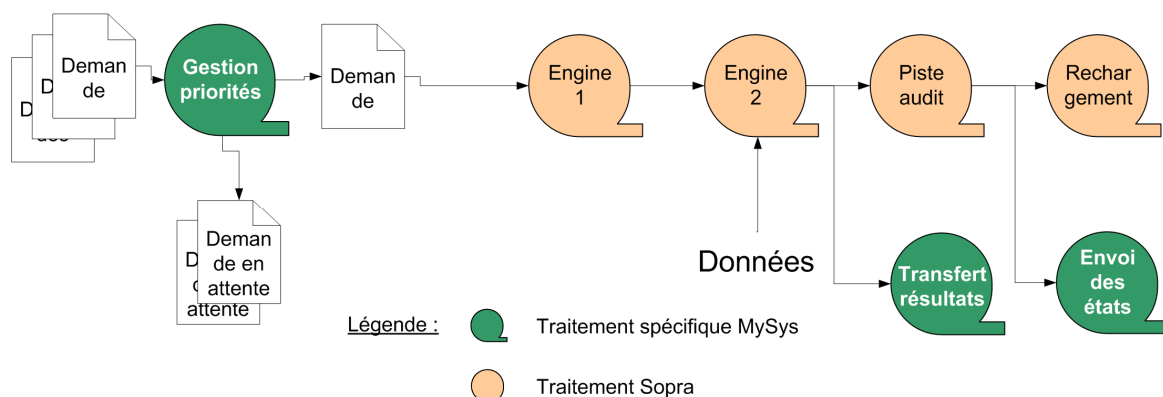


Figure 28 : Gestion des priorités en entrée d'Engine

La problématique est la suivante : les demandes (Pilfab) envoyées par le module client au module Engine via le module Server subissent un traitement de gestion des priorités. Il n'est par exemple pas possible, pour un même traitement Engine, d'effectuer deux fermetures de sessions différentes. Dans ce cas, une demande de fermeture de session est mise en attente pour un prochain traitement. Cela veut dire que si un utilisateur demande la fermeture de deux sessions, il faut qu'il attende deux jours pour que les deux demandes soient traitées par le module Engine.

Cette gestion des priorités a été mise en place par l'équipe précédente pour des dysfonctionnements lors du traitement du module Engine. Nous pensons que ces anomalies proviennent du découpage effectué sur le programme principal du module Engine. Cela semble être confirmé par la lecture approfondie de la documentation de l'éditeur qui mentionne à plusieurs reprises la possibilité de mettre en entrée du traitement Engine plusieurs demandes. A aucun moment il n'est mentionné la nécessité de mettre en œuvre un traitement de filtrage et de priorisation des demandes de fabrication en entrée du module Engine.

#### ▪ Prise de décision

Afin d'être tout à fait certains que la suppression de la gestion de priorité des demandes soit inoffensive, nous décidons d'appliquer la stratégie suivante :

- Une fois reçus les composants de l'éditeur, nous modifions l'enchaînement de nos traitements Engine pour ne plus le découper en quatre programmes.
- Nous supprimons la gestion des priorités
- Nous appliquons une batterie de tests pour éprouver la solution
- En cas de dysfonctionnement, nous remettons en place la gestion des priorités.



Cette procédure empirique permet de valider ou non notre analyse théorique basée sur notre réflexion et sur la documentation. En cas d'invalidation, nous gardons la possibilité de conserver la situation actuelle.

Le risque encouru est d'effectuer des tests incomplets ne révélant pas certains dysfonctionnements. Ceux-ci apparaîtront alors plus tard en production, bien trop tard pour faire un retour en arrière. Il est donc indispensable d'investir du temps dans la préparation des tests afin que ceux-ci soient le plus complets possibles. Nous aborderons plus tard dans ce mémoire leur mise au point et la stratégie retenue afin de minimiser les risques.

#### **4.4. Nouveaux transferts de fichiers**

Attardons-nous maintenant sur une autre analyse qui concerne la mise en place du circuit de transfert pour acheminer les fichiers de déclaration XBRL depuis MySys vers un destinataire.

Nous avons vu qu'actuellement les fichiers de déclaration sont produits sur les postes des utilisateurs puis envoyés par les Caisses d'Epargne via une interface dédiée. Ceci est valable pour tous les états sauf pour deux qualifiés de volumineux et qui peuvent, au choix, être produits sur le poste de l'utilisateur ou sur la machine hébergeant le module Server (Unix).

La nouvelle version d'Evolan Report modifie ce fonctionnement : désormais, les fichiers de déclaration, en format XBRL, seront exclusivement produits sur la machine hébergeant le module Server, en l'occurrence une machine Unix. Ces fichiers de déclaration XBRL doivent donc être acheminés vers la BPCE, l'inconnue demeure sur le transit ou non par les Caisses d'Epargne.

##### **4.4.1. Existant**

Nous venons de le voir, certains états réglementaires qualifiés de volumineux, peuvent être produits sur la machine hébergeant le module Server. Un circuit de transfert de ces fichiers en direction des caisses est déjà présent. Il débute sur Unix, se poursuit sur z/OS et est ensuite envoyé vers les Caisses d'Epargne.

Le protocole utilisé dans ces transferts est CFT (Cross File Transfer). Il s'agit d'un protocole développé par la société Axway. Il est souvent utilisé dans les systèmes informatiques complexes et multiplateformes tels que les systèmes bancaires car il fournit de multiples contrôles (sécurité, reprise après erreur, accusé réception, trace). Il peut s'appliquer sur différents protocoles de communication tels que X25 ou TCP/IP.

La figure 29 nous donne une illustration de l'architecture des transferts telle qu'elle est actuellement.

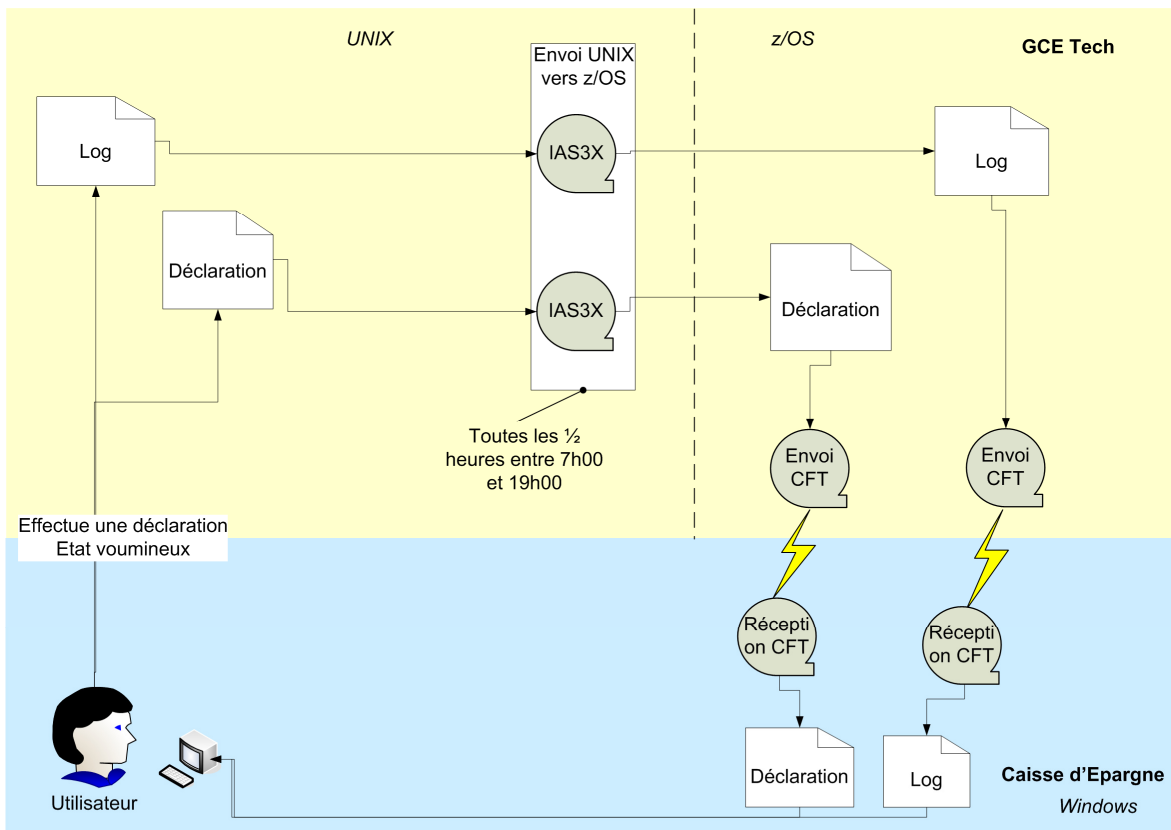


Figure 29 : Circuit de transfert des déclarations actuel

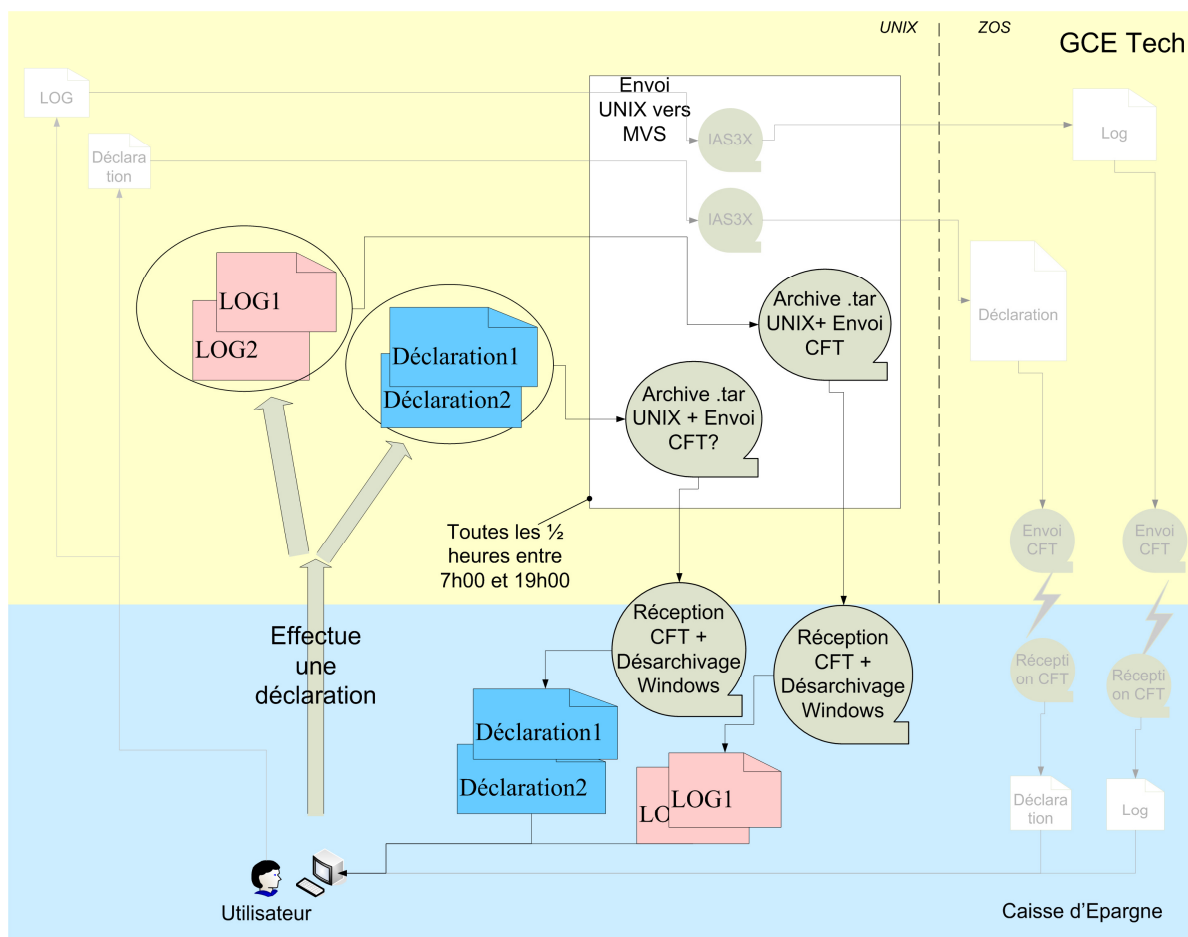
#### 4.4.2. Une première piste

Parallèlement à notre projet au sein de GCE Technologie, un autre projet a débuté au sein de la BPCE. Il consiste à construire une solution logicielle permettant de récupérer tous les fichiers de déclaration issus des Caisses d'Épargne pour en constituer un seul à envoyer à la Banque de France.

Nos premiers échanges avec le projet côté BPCE nous ont dans un premier temps conduit à adopter l'hypothèse suivante : l'envoi des fichiers de déclarations à BPCE est du ressort des établissements. Cette hypothèse amène une certaine complexité dans sa mise en œuvre. En effet autant de circuits de communication doivent être ouverts au départ de MySys qu'il y a de Caisses d'Épargne. De même, le nombre d'interlocuteurs (au moins un par Caisse d'Épargne) est élevé et rend plus difficile la coordination.

L'idée retenue est de se servir des éléments en place et qui permettent l'envoi des états volumineux mais en ne passant plus par le z/OS. Nous devons nous servir uniquement du traitement d'envoi auquel nous rajoutons des fonctionnalités permettant le transfert des nouveaux fichiers XBRL directement vers les Caisses. La solution d'envoi des états volumineux via z/OS avait été mise en place à une époque où il n'y avait pas de moyen de transfert à partir des machines Unix. Nous devons noter que le circuit existant doit être maintenu en place pendant un moment puisqu'un des tableaux volumineux sera toujours produit au début de SURFI dans son ancien format. A partir d'octobre 2010, il sera produit sous format XML et non XBRL pour des questions de volumétrie.

Afin de réaliser la mise en place des nouveaux circuits, nous souhaitons réutiliser le traitement intitulé « Envoi UNIX vers z/OS » mentionné sur la figure 30. Il tourne toutes les demi-heures. Nous y rajoutons une deuxième fonctionnalité d'envoi. Cette solution a l'avantage de n'avoir aucun impact sur l'ordonnanceur puisque le traitement existant déjà, cela permet un gain de temps. Ce gain de temps est important pour nous car, nous l'avons déjà évoqué, le grand nombre d'interlocuteurs et de circuits à ouvrir devient chronophage.



**Figure 30 : Circuit de transfert des déclarations futur**

Nous voyons sur la figure 30 l'ancien circuit conservé (grisé) ainsi que le nouveau proposé. Notons la réutilisation du traitement déjà en place actuellement.

La solution choisie effectue une mise en conteneur (.TAR) de plusieurs fichiers avant envoi. Ce système a l'avantage de permettre une conservation du nom initial du fichier ce qui n'est pas le cas autrement mais aussi d'envoyer plusieurs fichiers en un même envoi. Nous pensons que cette conservation du nom est importante pour l'utilisateur. En effet, lors d'une déclaration effectuée à partir du module Client, le logiciel donne le nom du fichier de déclaration produit sur le module Server. Il devient plus aisé pour l'utilisateur de recevoir un fichier avec le même nom que celui mentionné dans le logiciel.

Une fois cette analyse technique effectuée, nous avons tenté de rentrer en contact avec les interlocuteurs dans les Caisses d'Epargne et leur avons proposé la solution élaborée. Nous avons besoin de leur accord concernant la mise en conteneur des fichiers de déclaration et de log. En effet, selon l'architecture informatique des Caisses d'Epargne, il n'était pas certain que les utilisateurs disposent d'un outil pour ouvrir le conteneur ou bien qu'une procédure automatique le fasse pour eux.

La phase de coordination entre notre chantier et l'ensemble des établissements a donc débuté avec son lot de difficultés : identification, mobilisation et disponibilité des interlocuteurs, hétérogénéité des architectures techniques.

Mais cette piste s'est avérée erronée. La BPCE a décidé entre temps que les fichiers XBRL produits sur MySys devaient leur être directement envoyés sans passer par l'intermédiaire des Caisses d'Epargne.

#### ***4.4.3. Une nouvelle hypothèse***

La décision de BPCE nous facilite la tâche : tous les fichiers de déclaration XBRL, à l'exception d'un établissement, doivent être envoyés vers un seul destinataire. L'exception mentionnée ci-dessus correspond à un établissement bancaire du groupe BPCE qui n'est pas une Caisse d'Epargne mais que MySys héberge pour une partie de son activité. Cet établissement possède son propre logiciel Evolan Report mais une partie de ses données sont produites par MySys. Ainsi nous devons envoyer leurs déclarations pour l'intégration dans leur logiciel. Ils produisent alors d'autres déclarations XBRL qu'ils envoient eux-mêmes à BPCE.

Nous avons eu plusieurs contacts avec BPCE pour la mise au point de ce transfert. En effet, transférer des fichiers entre deux entités nécessite que les paramètres soient partagés entre ces deux entités. Ces discussions ont permis à BPCE d'exprimer un besoin fort : « l'envoi d'un fichier de déclaration doit être le plus rapide possible après sa création ».

Le besoin de BPCE nous empêche d'utiliser le programme actuel d'envoi de Unix vers z/OS car ne tournant que toutes les demi-heures d'une part et étant un traitement commun à l'ensemble de toutes les Caisses d'Epargne d'autre part. Il est préférable que chaque Caisse d'Epargne possède ses propres traitements, notamment concernant des transferts, car les fichiers de déclaration à envoyer ne sont pas tous présents en même temps pour l'ensemble des Caisses d'Epargne. La solution à mettre en place est donc un nouveau traitement tournant toutes les cinq minutes.

La figure 31 nous montre la solution retenue concernant le transfert des déclarations XBRL.

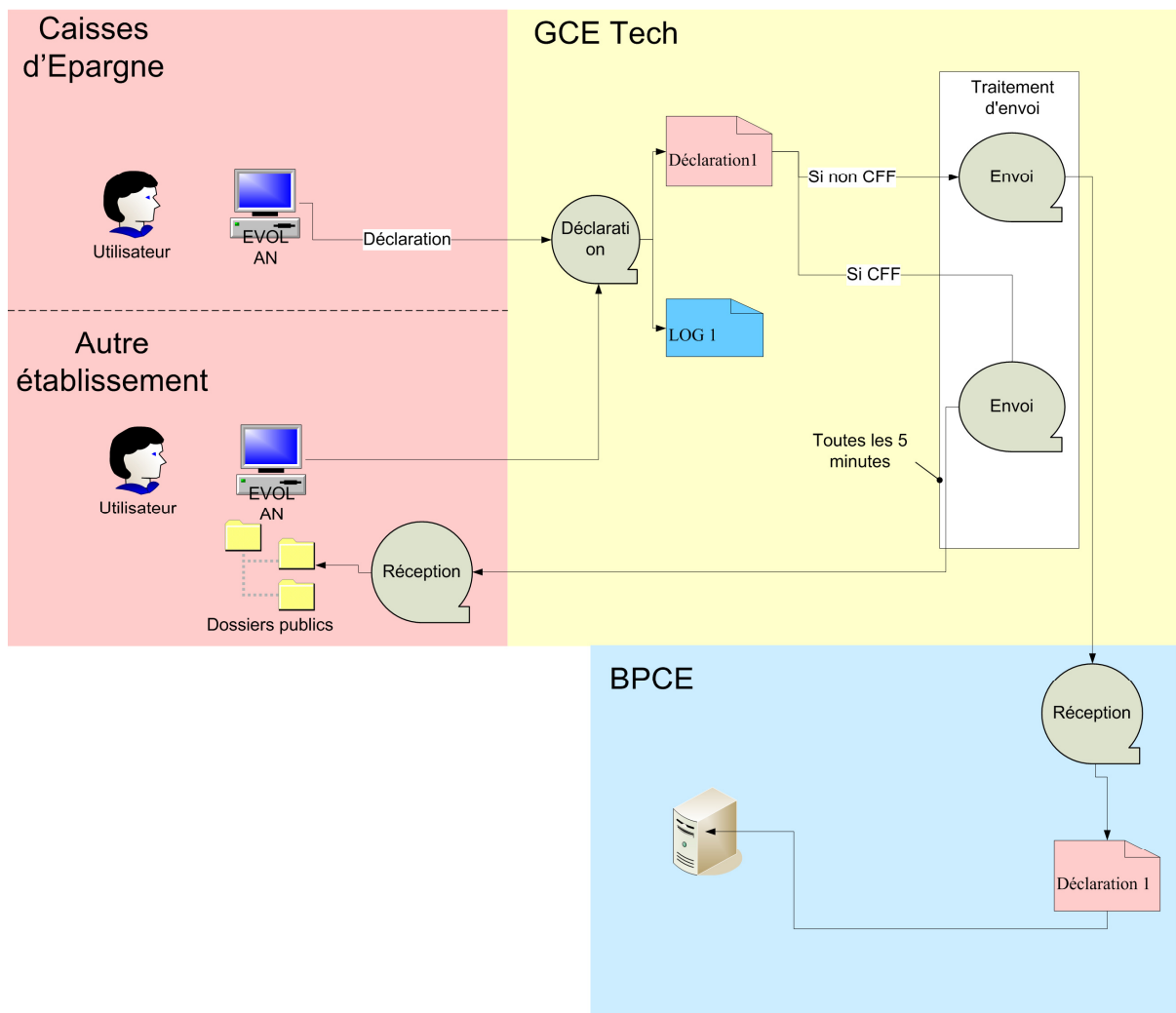


Figure 31 : Nouveau traitement de transfert des déclarations XBRL

#### 4.5. Mémoire Java et des risques à éviter

A l'heure où ces problèmes ont été évoqués dans le projet, la mise en production va intervenir dans moins de deux mois, ce problème est une alerte majeure pour le succès du projet et nécessite de notre part un fort investissement.

##### 4.5.1. Problématique

Une évolution apportée par Sopra pour permettre les déclarations XBRL ainsi que le déroulement de contrôles XBRL provoque des dysfonctionnements sur nos environnements d'édition au niveau de la mise à disposition de mémoire. Ces dysfonctionnements se traduisent par des déconnexions de la machine UNIX voire des plantages et le blocage des utilisateurs sur le module Client sans possibilité d'intervenir. Seule une intervention de l'équipe système permet de débloquent les utilisateurs.

Les déclarations XBRL ou le déroulement de contrôles XBRL s'appuient sur des processus Java très gourmands en mémoire. Le lancement de plusieurs de ces processus en même temps demande une disponibilité de mémoire importante. Ce qui nous inquiète est notre architecture technique de production. Nous avons deux instances de production et donc deux machines Unix pour l'ensemble des

établissements. Chaque machine est supposée supporter une douzaine d'établissements. Chaque établissement étant susceptible d'effectuer des traitements XBRL en même temps, les besoins en mémoires, tels qu'ils ont été dimensionnés dans les environnements de test, sont colossaux.

Nous sommes étonnés des performances de ces fameux traitements XBRL qui semblent très longs, augmentant ainsi les risques de collision avec d'autres traitements XBRL. La première question à résoudre est : est-ce normal que les traitements soient aussi longs ? Ensuite, comment garantir qu'aucun dysfonctionnement n'apparaîtra en production après la mise en place des évolutions SURFI ?

#### **4.5.2. Protocole d'analyse**

Le protocole que nous avons mis en place est constitué de trois axes de recherches en parallèle. Le premier est la piste Sopra : nous sollicitons l'assistance de l'éditeur notamment sur les raisons de telles performances dégradées. Le deuxième, interne à GCE Technologies fait intervenir des ingénieurs systèmes et administrateurs base de données afin d'une part d'analyser les traitements incriminés et d'autre part de préparer l'environnement de production pour se prémunir de toute anomalie liée à ces traitements Java. Enfin le troisième axe se trouve chez les Banques Populaires qui, rappelons-le, font partie du groupe BPCE. Eux-mêmes intègrent également le logiciel Evolan Report pour SURFI et peuvent nous apporter des informations sur leur propre expérience.

- Le premier axe : Sopra en tant qu'éditeur de la solution nous fournit de l'aide dans nos travaux d'intégration du logiciel à notre plateforme. Sur ce sujet, les premiers échanges montrent que Sopra ne s'attendait pas à de telles performances et ne propose pour le moment aucune solution. Selon eux il est normal que les traitements Java consomment autant de mémoire, une étude est en cours de leur côté pour apporter des éléments de réponse quant aux temps de traitement.
- Le deuxième axe, interne à l'entreprise, va nous permettre de mobiliser rapidement les personnes spécialistes systèmes et bases de données. Les risques encourus sur un dysfonctionnement de la production amènent un degré d'urgence important à ce problème. Des réunions sont mises sur pied. Il est décidé de pratiquer des tests techniques : les traitements incriminés sont lancés alors qu'une batterie de contrôles est mise en place afin de révéler le ou les processus responsables de tels temps de traitement. Ces tests ont montré une utilisation excessive de mémoire swap. Or Java gère très mal l'utilisation du swap. Un deuxième test effectué en rajoutant de la mémoire sur notre environnement de test montre que les temps de traitement sont divisés par douze.
- Le troisième axe, le rapprochement avec les Banques Populaires, n'a donné que peu d'éléments car le socle technique sur lequel s'exécutent les traitements Java sont différents entre les deux plateformes (Unix pour MySys, Windows pour les Banque Populaires). L'information que nous retenons est que chez eux, les traitements sont extrêmement rapides par rapport aux nôtres.

### **4.5.3. Moyens engagés**

Les tests effectués sur l'environnement de développement ont montré un besoin en mémoire important. Il faut ainsi prévoir 2Go de mémoire centrale par établissement. Les machines de production ne disposant pas de mémoire suffisante, il est nécessaire de rajouter 44 Go de mémoire centrale.

Ce besoin est important financièrement et à l'heure où nous écrivons ces lignes, les responsables de l'architecture technique n'ont pas validé la mise à disposition de ces moyens pour la mise en production.

En synthèse sur ce problème non terminé, nous remarquons qu'il est indispensable de trouver des axes d'analyses différents pour augmenter les chances de trouver une solution. Mobilisation, coordination et rapidité sont également nécessaires. Il nous faut rassembler le maximum de forces techniques.

## **5. La réalisation**

---

Nous venons de présenter des analyses effectuées à différents niveaux. Celles-ci nous ont permis de proposer des solutions adaptées au travail des utilisateurs mais aussi au travail des équipes de maintenance ainsi qu'aux dysfonctionnements rencontrés lors des tests de la solution.

Nous allons maintenant aborder la réalisation et la mise en place de ces solutions. Nous allons voir que notre projet a débuté dans un contexte particulier de nouvelle organisation qui a nécessité une phase de recherche de processus et d'interlocuteurs associés. Ensuite, nous décrivons la construction d'un plan d'action, référentiel des réalisations et des différentes phases de développement et de documentation.

### **5.1. Nouvelle organisation : identifier les acteurs**

La fusion des trois plateformes informatiques des Caisses d'Épargne fin 2008 a entraîné un bouleversement à deux niveaux : de nouvelles procédures et de nouvelles équipes. Notre problématique était la suivante : devoir mettre en production des évolutions sur l'application BAFI sans connaître les services devant intervenir sur le projet notamment concernant les installations sur les différents environnements. Rappelons à ce stade que nous disposons de deux environnements d'édition, dits environnements de recette. Il s'agit de l'environnement de recette utilisateur et d'un environnement volumétrie réelle. Enfin la production est également un environnement sur lequel les interventions sont réalisées par une équipe dédiée.

Il a donc fallu dans un premier temps effectuer un travail de recherche d'informations pour d'une part connaître les procédures à appliquer et d'autre part trouver les bonnes personnes à mobiliser.

Acteurs	Environnements de recette		Environnement qualification		Environnement production	
	Installation composants	Installation VTOM	Installation composants	Installation VTOM	Installation composants	Installation VTOM
 GAP		✓	✓	✓	✓	✓
 Montage environnements	✓					
 DBA Edition	✓					
 DBA Production			✓		✓	

**Figure 32 : Interventions des ressources**

La figure 32 représente les interventions de chacun des acteurs en ce qui concerne les installations de composants mais aussi de l'ordonnanceur des traitements VTOM (Visual TOM). Le principe général est le suivant : les équipes de montage d'environnement et les administrateurs de bases de données (DBA) Edition interviennent sur les environnements jusqu'à l'environnement de qualification. Ensuite, ce sont les DBA de production et les équipe de Gestion Application Projet (GAP) qui interviennent.

Nous remarquons qu'un environnement apparaît alors que nous ne l'avons pas mentionné auparavant ; il s'agit de l'environnement de qualification. Nous ne disposons pas de cet environnement. Un environnement de qualification permet de valider l'ensemble d'une version. Toutes les évolutions d'une même version doivent y être installées et les traitements être déroulés. Cela permet de qualifier une version et de s'assurer que toutes les évolutions apportées fonctionnent entre elles.

Cette absence d'environnement de qualification représente une difficulté car en temps normal il permet à GAP d'effectuer une répétition de l'installation avant celle effectuée sur l'environnement de production. Il nous fallait donc trouver un moyen pour permettre à GAP d'effectuer une répétition correcte de l'installation. Tout naturellement, l'environnement de volumétrie réelle s'est présenté comme idéal pour remplir cette fonction. L'équipe « Montage Environnement » installe sur l'environnement de recette utilisateur, GAP installe sur l'environnement de volumétrie réelle, façonne une documentation d'installation et met en production.



En ce qui concerne les DBA, à l'instar de l'équipe « Montage Environnement » et GAP, une équipe a la responsabilité d'installer dans les environnements de recette. Les DBA de production interviennent à partir de la qualification. Dans notre projet, contrairement à GAP, les DBA de production n'ont pas souhaité effectuer de répétition d'installation sur un environnement, ils installeront en production directement.

## **5.2. Transformer l'étude d'impact en plan d'action**

### **5.2.1. Le plan d'action**

Une fois les acteurs identifiés, il nous fallait trouver un moyen efficace pour communiquer et transmettre les consignes d'installation aux différents intervenants. C'est le rôle du plan d'action.

Le plan d'action, construit à partir de notre étude d'impact comporte l'ensemble des phases nécessaires à l'installation complète de notre projet. Il est découpé sur plusieurs niveaux :

- le premier niveau de découpage est fait en fonction des acteurs devant intervenir : les DBA pour les modifications Oracle, SRV pour les installations sur l'environnement de recette, GAP pour ce qui concerne l'ordonnanceur et les installations de composants sur l'environnement de volumétrie réelle et enfin nous pour le développement de programmes.
- Le deuxième niveau est d'ordre chronologique. Il donne une progression dans l'installation des différents composants. En effet, certaines phases de l'installation ne doivent pas être réalisées avant d'autres.

### **5.2.2. La livraison des composants**

L'éditeur Sopra livre ses composants sur une plateforme de téléchargement dans des archives ZIP. Une fois ceux-ci récupérés, il est nécessaire de faire un premier tri : en effet les composants que nous avons reçus ne peuvent pas tous être installés tels quels, certains doivent être adaptés pour fonctionner sur MySys. D'autres qui sont contenus dans la même archive ZIP ne seront pas installés par les mêmes intervenants.

Afin de faciliter le travail des installateurs, nous avons choisi de construire des packages d'installation (des archives ZIP) :

- Un package par répertoire cible d'installation. Par exemple, le module Engine possède des composants dont le répertoire d'installation est COM\_d ; tous les composants à installer dans ce répertoire seront rassemblés dans une archive ZIP nommée COM.zip
- Un package pour les composants Oracle
- Un package pour les programmes développés par MySys et qui permettent l'intégration du logiciel Evolan Report sur MySys

Nous avons vu précédemment, que l'éditeur avait loti ses livraisons. Chaque package porte cette notion de lotissement. Ainsi la livraison de composant dans le répertoire COM\_d d'Engine sont rassemblés dans un package COM\_Lot2.zip.

Apporter un niveau de détail aussi poussé dans la constitution des packages de livraison permet de rendre l'installation facile et détaillée limitant ainsi les risques d'erreur.

La figure 33 présente un extrait de ce plan d'action.

N° Chrono	Socle	Livrable	Composant(s)	Action	Type de transfert	Vérifications à effectuer
3	Oracle	Oracle.zip	update385instance.sql	Script à adapter pour le passer sur chaque instance IP1 et IP2 des bases IMEXX (où XX correspond au n° de couloir)	-	- Vue VD_CEL_TAX_IDN créée - Vue VP_CEL_TAX_IDN créée - La colonne TAX_IDN de la table TAX_IDN est un VARCHAR2(1025)
4			Géré par SRV	Ajout de 12 colonnes à la table PISTE_AUDIT_ESTD_CUMUL sur chaque instance de IP1 et IP2 des bases IMEXX (où XX correspond au n° de couloir)	-	- Les 12 colonnes sont ajoutées à la table PISTE_AUDIT_ESTD_CUMUL - Les données sont conservées
5		-	conditionnement_traitement.sp	Désinstallation du répertoire /pgcl/LRP/ime/lib/sh	-	Absence du composant dans le répertoire /pgcl/LRP/ime/lib/sh
6		-	edition_piste_audit.shplan	Désinstallation du répertoire /pgcl/LRP/ime/lib/sh	-	Absence du composant dans le répertoire /pgcl/LRP/ime/lib/sh
7		COM.zip	57 fichiers	Installation dans le répertoire /tech/lreport/Engine/com_d	ASCII	La date des 57 composants correspond à la date d'installation
8		-	-	Effectuer un CHMOD 775 sur l'ensemble des 57 fichiers	-	Les 57 composants sont en 775
9		MySys.zip	9 fichiers	Installation dans le répertoire /pgcl/LRP/ime/lib/sh	ASCII	La date des 9 composants correspond à la date d'installation

**Figure 33 : Extrait du plan d'action**

Nous pouvons voir dans la figure 33 le premier niveau représenté par des couleurs différentes : l'orange pour les DBA le bleu pour SRV. Ensuite, le deuxième niveau apparaît avec les numéros chronologiques dans la première colonne. Nous voyons également apparaître les packages de livraison dans la colonne « livrable ».

### 5.3. Coordonner l'intervention des acteurs

Nous l'avons vu, différents acteurs doivent intervenir pour effectuer les modifications nécessaires aux projets dans les environnements de l'édition. Une phase importante pour réussir le projet était donc d'établir une coordination efficace de tous ces acteurs pour éviter la perte de temps et minimiser les risques.

Ainsi, les notions importantes dans la coordination des acteurs sur notre projet sont les suivantes :

- **Disponibilité** : cette notion est primordiale car un problème de disponibilité entraîne une dérive du projet en termes de délais. Or nous avons vu que ce projet réglementaire ne peut souffrir d'une dérive quelconque. Si cet aspect est plutôt le rôle du chef de projet, nous avons participé à cette phase notamment par le fait que nous étions en rapport direct avec les ressources et avons pu nous assurer de leur disponibilité aux moments où nous l'avions planifié.
- **Capacité** : ce n'est pas parce qu'une ressource nous est affectée que celle-ci possède les compétences techniques pour réaliser ce que nous lui demandons. Cet aspect est délicat car ne devrait pas se rencontrer. Mais nous devons rappeler que nous sommes en période de changement, une réduction du nombre de personnes dans les services amène certaines équipes à être en surcharge de travail temporaire et à proposer les personnes disponibles à ce moment là et non les personnes capables de répondre à nos besoins. Nous avons rencontré cette difficulté et avons dû nous adapter en offrant un accompagnement rapproché de la ressource tout au long du projet jusqu'à la mise en production. Nous reparlons de cette difficulté un peu plus loin.
- **Réalisation** : après s'être assuré de la disponibilité des ressources ainsi que de leur capacité à effectuer le travail que nous allons leur demander, vient le temps de la réalisation. Notre rôle est de coordonner l'intervention de chacun sur les différents environnements.
- **Vérification** : une intervention d'un acteur externe doit s'accompagner d'une phase de vérification de son travail. Cette phase est essentielle car elle permet de vérifier la bonne compréhension de notre demande de la part de la ressource. Si des erreurs sont détectées, les procédures sont modifiées jusqu'à ce que le travail désiré soit effectué.

Le plan d'action dont nous venons de parler précédemment a été le moyen de communiquer nos besoins aux différents acteurs. Chaque demande de travail auprès d'un acteur s'accompagnait d'un extrait du plan d'action lui correspondant. Ainsi, il avait à sa disposition toutes les informations nécessaires pour réaliser son travail mais aussi pour le vérifier. Ces informations de vérification permettent aux acteurs d'effectuer eux-mêmes un premier niveau de vérification.

## 5.4. Développer

Cette phase a été principalement effectuée par nos soins. Deux types de développements ont été réalisés : les modifications des programmes spécifiques MySys et les adaptations des programmes d'Evolan Report pour fonctionner sur MySys.

### 5.4.1. Développements spécifiques MySys

Les développements effectués sur des programmes MySys sont essentiellement dus à la décision de modifier l'architecture des traitements, c'est-à-dire la suppression du découpage en quatre du programme principal d'Engine.

### 5.4.2. Modification des programmes du logiciel Evolan Report

Le logiciel Evolan Report tel que livré par l'éditeur Sopra ne peut fonctionner en l'état sur MySys, notamment en ce qui concerne le module Engine. En effet, le logiciel s'installe à partir d'un répertoire choisi par le client appelé répertoire d'installation. Dans ce répertoire d'installation vient une structure de répertoires imposée par l'éditeur.

Or la plateforme MySys impose des normes en matière de répertoire qu'il est nécessaire de respecter. Cette norme est discutable dans le cas de logiciel en provenance d'un éditeur car dans ce cas des composants doivent être modifiés. Il nous semble préférable de conserver l'architecture des répertoires imposée par un éditeur car modifier un logiciel externe est d'une part une source de risques importants et d'autre part nécessite plus de travail lors de montées de version. L'équipe nous ayant précédé a fait le choix de modifier les composants, revenir en arrière est trop coûteux, nous continuons donc dans cette voie.

Extrait de ces normes :

- Toute application doit être installée dans un répertoire « /tech/nom\_application/ » où « nom\_application » dans notre cas correspond à « lreport ». Les programmes livrés par Sopra s'installent donc tous dans celui-ci.
- tout fichier de données doit atterrir dans un répertoire nommé « /pfic/nom\_application ».

Le fonctionnement du logiciel Evolan Report prévoit que tout soit contenu dans le répertoire d'installation (programmes et données), en l'occurrence « /tech/lreport/ ». Ceci est paramétré dans les programmes écrits en KSH. Les programmes livrés par Sopra sont donc modifiés pour permettre d'aller lire le répertoire « /pfic/nom\_application ».

La figure 34 nous donne un aperçu de l'architecture des répertoire de la BAFI sur MySys.

Dans notre projet, l'impact d'une telle modification est important car aucun document ne renseigne de manière exhaustive les programmes à modifier. Il a donc fallu passer en revue l'ensemble des composants KSH et détecter les

modifications à entreprendre. Ce travail fastidieux est source d'erreur et nécessite la réalisation de tests complets et efficaces.

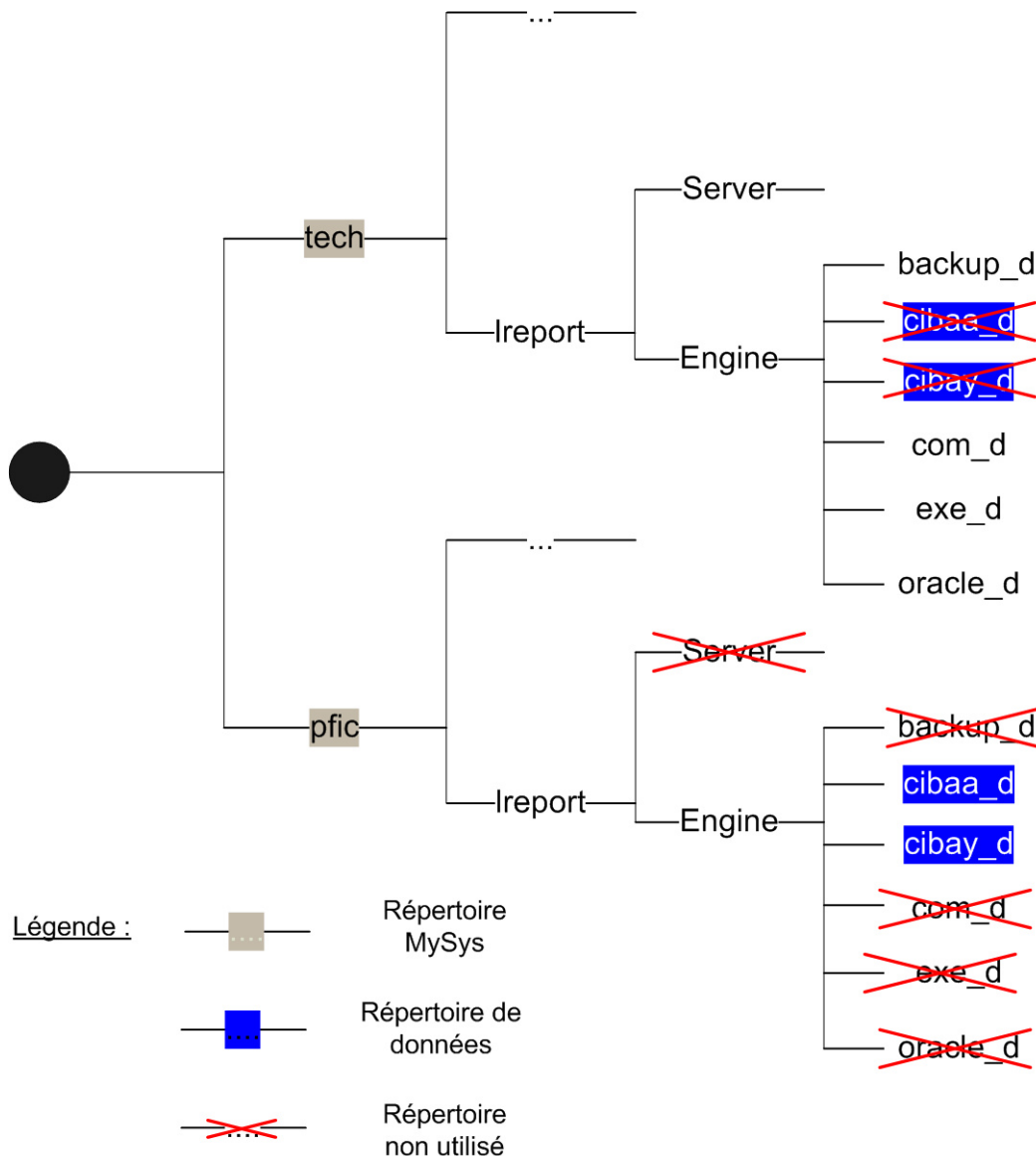


Figure 34 : Arborescence BAFI

## 5.5. Documenter

### 5.5.1. Documentation applicative

A part la documentation fournie par l'éditeur, peu de documentation était à notre disposition lorsque nous avons repris la responsabilité de l'application. Seule une documentation succincte sur les différents traitements enchaînés et leur fonctionnalité avait été rédigée. Elle contenait peu de précisions quant aux traitements spécifiques MySys et une certaine approximation dans les règles de gestion écrites.

Notre prise de responsabilité de l'application a été suivie d'une période de rétro documentation. Cette phase a permis d'une part la mise à jour complète des

règles de gestion appliquées dans les chaînes de traitement et d'autre part l'acquisition d'une connaissance approfondie de l'application BAFI sur MySys.

Au début du projet SURFI, nous avons donc à notre disposition une documentation complète sur les règles de gestion. Celle-ci a été mise à jour au fil des modifications opérées sur les traitements. Elle est transmise à l'équipe de maintenance lors de la présentation que le projet leur fait avant la mise en production dans le cadre du transfert de compétences. En effet, la documentation fait partie des critères pour déterminer si une application est en état d'être reprise ou non par l'équipe de maintenance ; ils ont un avis à donner sur le sujet.

### **5.5.2. Documentation technique**

Concernant la documentation nous devons en mentionner une, essentielle, qui permet à l'équipe projet de formaliser les évolutions à effectuer sur les chaînes de traitements. Ces évolutions, dans notre cadre du chantier Unix/Windows, concerne essentiellement l'ordonnanceur des traitements, VTOM. Cette documentation est appelée DCT (Document de Conception Technique). Elle est le trait d'union entre notre chantier et l'équipe d'industrialisation ; elle permet de lui donner précisément la manière dont les traitements doivent se comporter :

- Conditions de déclenchement des traitements (arrivée de fichiers, horaire, traitement précédent terminé)
- Périodicité des traitements (journalier, mensuel, cyclique)
- Dépendance des traitements les uns par rapport aux autres
- Que faire en fonction des codes retour des traitements (poursuivre, stopper, reprendre, etc.)
- Paramètres à passer aux programmes lancés
- Avec quel « user » doit être lancé tel ou tel traitement

Le DCT doit être rempli avec précision car c'est le comportement de l'application dans son ensemble qui en dépend.

## **6. Les tests**

---

La phase de test est cruciale car elle doit permettre de révéler ce qui ne fonctionne pas ou de manière anormale avant la mise en production.

Dans le cadre de notre projet, les tests ont été mis en œuvre à différentes reprises :

- Lors du montage des environnements techniques afin de prononcer leur recevabilité
- Pour vérifier la non régression des traitements suite à leur réorganisation (suppression du découpage du programme principal d'Engine) ou suite à la suppression du système de gestion des priorités des demandes
- Au cours de la recette utilisateur

## 6.1. Une procédure de test à mettre en place

Le montage des environnements ainsi que l'installation du logiciel Evolan Report ont été faits entièrement manuellement. Aucune procédure automatique pouvant réduire les risques d'erreur humaine n'a été appliquée. Or, l'installation du logiciel est hétérogène et fastidieuse (composants ASCII et binaires, machine UNIX et Windows, lancement d'outil dédié, etc.).

Dans le passé, des dysfonctionnements lors d'une mise en production ont été rencontrés du fait d'erreurs humaines. Nous ne pouvons réitérer une telle situation. C'est pourquoi le montage des environnements techniques ne pouvait s'accompagner d'une phase de test poussée permettant de prononcer la recevabilité sans équivoque. En effet, en cas de problème à l'installation de la nouvelle version du logiciel Evolan Report, il fallait être certain que cela provenait de l'évolution et non d'un dysfonctionnement de l'environnement.

Il fallait également que cette procédure puisse être de nouveau utilisée, notamment lors des montées de version du logiciel Evolan Report ou encore lors de l'application de patches.

Cette procédure permet de solliciter la majeure partie des composants du logiciel Evolan. Elle est le fruit de notre expérience sur les différentes anomalies qui peuvent intervenir. Elle devra donc évoluer avec l'expérience future et les problèmes rencontrés.

Il faut noter que Evolan Report étant un logiciel d'un éditeur externe à notre entreprise GCE Technologies, la majeure partie du fonctionnement de ce logiciel n'est pas apportée par l'éditeur aux clients. Or, nous en avons fait l'expérience, connaître le fonctionnement permet de gagner un temps précieux lorsque des anomalies sont rencontrées : cela évite de contacter le support de Sopra et de subir les délais de réponse et d'aller-retour sur le sujet. C'est pourquoi nous nous sommes astreints à comprendre le plus possible les mécanismes cachés derrière ce logiciel et donc trouver rapidement les causes des dysfonctionnements.

La figure 35 nous permet de visualiser une partie de la procédure de test. Le but est de disposer d'une réponse (polygone vert sur le schéma) pour toutes les anomalies qui peuvent être rencontrées.

Certaines branches sont vides, notamment celles concernant les anomalies. En effet nous n'avons pas rencontré tous les dysfonctionnements possibles et n'avons donc pas toutes les réponses. C'est pourquoi cette procédure est à affiner avec l'expérience et les différentes anomalies rencontrées.

Un bilan de la mise en œuvre de cette procédure nous permet de dire qu'elle révèle des anomalies lorsqu'il y en a et que lorsqu'elle est appliquée avec succès, la bonne installation du logiciel est garantie.



Figure 35 : Extrait de la procédure de test



## 6.2. Tests et table de vérité

Nous l'avons vu, une réorganisation des traitements a été opérée au cours de ce projet SURFI ainsi que la suppression de la gestion des priorités. Une étude d'impact a été faite auparavant afin de valider la possibilité de faire ces évolutions et de révéler les modifications de comportements de l'application. Notre connaissance du produit Evolan Report n'étant pas complète, il demeurait une part d'incertitude quant au bon fonctionnement de l'application. Il nous fallait donc mettre sur pied une phase de test spécifique à ces évolutions.

La procédure de test mentionnée dans le paragraphe précédent, si elle permet un premier niveau de test ne suffisait pas, il fallait effectuer des tests plus poussés, tenter d'aller plus loin et vérifier la non régression des évolutions.

La gestion des priorités avait été mise en place afin de ne pas présenter au traitement Engine plusieurs demandes de fabrication incompatibles entre elles (une fermeture et une ouverture de session, deux fermetures en même temps, une restauration avec toute autre demande, etc.). Il nous fallait donc tester ces différentes combinaisons de manière exhaustive pour tester cette non régression.

Pour cela, nous nous sommes appuyés sur une table de vérité (cf. figure 36) qui permet de tester l'ensemble des combinaisons. Quatre commandes différentes peuvent être mises en entrée (Ouverture, Fermeture, Archivage et Restauration). Cela nous donne  $2^4=16$  variantes de test. Mais il nous fallait également tester si deux commandes de même nature (deux fermetures par exemple) fonctionnaient en même temps. Nous avons donc rajouté 4 cas supplémentaires.

Cas de test	Ouverture 1	Ouverture 2	Fermeture 1	Fermeture 2	Archivage 1	Archivage 2	Restauration 1	Restauration 2
1								
2							✓	
3					✓			
4					✓		✓	
5			✓					
6			✓				✓	
7			✓		✓			
8			✓		✓		✓	
9	✓							
10	✓						✓	
11	✓				✓			
12	✓				✓		✓	
13	✓		✓					
14	✓		✓				✓	
15	✓		✓		✓			
16	✓		✓		✓		✓	
17	✓	✓						
18			✓	✓				
19					✓	✓		
20							✓	✓

Figure 36 : Table de vérité pour les tests

La table de vérité représentée dans la figure 36, montre au travers des quatre lignes colorisées les quatre cas de présentation de double commande en entrée du module Engine.

La mise en œuvre des tests selon la table de vérité a permis de valider le bon fonctionnement de l'application après la mise en place des évolutions. Un cas n'a pas réussi, le cas numéro 20, deux restaurations mises en entrée du traitement Engine. Après contact auprès de Sopra, l'éditeur nous a indiqué qu'il s'agissait d'un dysfonctionnement du logiciel déjà répertorié chez eux.

### **6.3. Recette utilisateur et sa préparation**

Tous les tests dont nous venons de parler précédemment ont été exécutés par la maîtrise d'œuvre. Ces tests, entièrement techniques, ont permis de valider la bonne installation de la nouvelle version du logiciel Evolan Report mais aussi la non régression des traitements suite à la modification de leur architecture.

Une autre phase de test appelée « recette utilisateur » est mise en place. Elle fait intervenir les utilisateurs et est fortement orientée sur le fonctionnel. C'est au cours de ces tests que les modifications qui ont été faites par le chantier z/OS sur les données sont validées.

Ce type de recettes se déroule traditionnellement sur le site de la maîtrise d'œuvre. Une préparation est nécessaire notamment en termes d'environnement : mise à disposition de postes, configuration de ceux-ci pour accéder au logiciel Evolan Report, accès à Business Object, etc. Le temps consacré à la mise en place n'est pas négligeable mais nécessaire : un environnement qui fonctionne sans problèmes est un atout pour établir des rapports efficaces et faciles entre maîtrise d'œuvre et utilisateurs.

Ces recettes se sont déroulées en plusieurs phases. Une première recette exécutée sur l'environnement de recette utilisateur, une deuxième sur l'environnement de volumétrie réelle. Si le but était de valider l'aspect fonctionnel, ces recettes ont aussi permis aux utilisateurs de tester le logiciel, poser des questions quant à son fonctionnement et vérifier que leurs actions habituelles sur le logiciel fonctionnaient toujours. C'est ainsi que ces recettes ont soulevé un problème dû à la modification de l'architecture des traitements Engine; problème résolu par la suite.

En synthèse, ces recettes ont été importantes car faisant intervenir les utilisateurs. Outre la validation des évolutions fonctionnelles des données, elles ont révélé un dysfonctionnement du traitement Engine.

## **7. La mise en production**

---

A l'heure où nous écrivons ce mémoire, la mise en production du projet n'est pas encore effectuée puisque prévue pour le 21 juin 2010. Néanmoins, cette phase nécessite une préparation minutieuse, plusieurs semaines auparavant afin de la sécuriser un maximum et garantir la bonne installation des évolutions. Nous allons donc décrire l'avancée de cette préparation, telle qu'elle est, à la rédaction de ces lignes.

### **7.1. Sécurisation de la mise en production**

#### **7.1.1. Des risques**

Nous avons vu que le plan d'installation découlait du plan d'actions. Chaque installation de lot dans les environnements de l'édition a fait l'objet d'une procédure d'installation mise en œuvre et donc testée – rappelons que l'éditeur nous a fourni les composants du logiciel Evolan Report sous forme de quatre lots à des dates différentes.

L'ensemble de ces procédures assemblées constitue un squelette de plan d'installation. Nous parlons de squelette car le déploiement en production diffère des installations dans les environnements de recette. En effet, d'une part les environnements de recettes de la BAFI ne possèdent qu'un voire deux établissements alors que la production supporte l'ensemble des établissements sur deux instances de production. D'autre part le socle Windows est doublé en production (deux machines Windows en équilibre de charge sur chaque instance de production) et non dans les environnements de l'édition.

Les différences nécessitent un aménagement des procédures d'installation réalisées dans les environnements d'édition. Une fois ces aménagements effectués, le plan d'installation est conforme pour être appliqué à la production et doit normalement être répété sur un environnement dit de « qualification », environnement identique à la production du point de vue architecture technique. Concernant la BAFI l'environnement devrait donc avoir une machine Unix pour deux machines Windows. Nous l'avons vu, nous ne disposons pas d'un tel environnement, la procédure d'installation complète n'a donc pas été répétée dans sa version finale. Cela constitue un premier risque.

Un deuxième risque se présente du côté des compétences techniques de l'installateur. Le plan d'installation, conformément aux normes GCE Technologies, doit être élaboré par les équipes qui effectuent le déploiement et non par les équipes de l'édition. Dans notre projet et plus particulièrement pour le chantier Unix/Windows, nous avons élaboré la majeure partie de ce document, le responsable du déploiement ayant effectué quelques modifications. Nous expliquons cela par le fait que notre connaissance de l'application, de l'architecture technique, des techniques d'installation mises en œuvre pour le projet était supérieure à celle du responsable du déploiement. La rédaction du plan d'installation par une équipe différente de celle qui va installer représente un risque.

### **7.1.2. Minimiser les risques**

Les deux risques à sécuriser sont :

- l'absence de répétition en condition réelle du déploiement
- un plan d'installation rédigé par une équipe différente de celle d'installation.

La réponse unique à ces deux risques est d'appliquer un plan de sécurisation basé sur des points de validation à mettre en place au cours de l'installation.

Le déploiement en production prévoit que les tâches du plan d'installation seront exécutées à plusieurs reprises d'une part du fait qu'il existe deux instances de production et d'autre part parce que sur chaque instance de production, deux machines Windows sont installées.

En cas d'anomalie d'installation, afin d'empêcher que celle-ci soit propagée à l'ensemble de la production, nous préférons intervenir dès la première exécution de la tâche et vérifier sa bonne exécution. Ainsi des points d'arrêts dans la procédure d'installation sont prévus pour permettre notre intervention. Au cours de ces points d'arrêt, nous exécuterons deux types de contrôle : des contrôles statiques et des contrôles dynamiques.

- Contrôles statiques : effectués sur le socle Unix, ils permettront de vérifier que les composants ont été correctement installés, qu'il s'agit de la bonne version, que les répertoires ont bien été créés, etc. Afin que ce contrôle ne soit pas fastidieux et long, nous mettons sur pied un outillage spécifique et le plus automatique possible qui a pour but d'effectuer les vérifications rapidement, sur l'ensemble de l'installation. Cet outillage effectue des tests de différentes natures : vérification de la date d'installation, vérification de la taille, vérification de la présence de composants mais également vérification de la non altération de composants due à une erreur de transfert FTP (File Transfert Protocol) ; par exemple utilisation du mode binaire au lieu d'ASCII.
- Contrôles dynamiques : ils se traduisent par la mise en œuvre de l'application et donc vérifient qu'elle fonctionne. Ces tests sont tout de même rapides et rudimentaires. Nous n'appliquons pas la procédure de test évoquée précédemment. Ces contrôles dynamiques consistent à lancer l'application Evolan Report pour un établissement et vérifier que la chaîne de liaison fonctionne (visualisation des sessions ouvertes de l'établissement, connexion à une session, ouverture de quelques états dans la session).

Si les contrôles sont satisfaisants, nous donnons le feu vert à l'installateur pour qu'il poursuive l'installation sur le restant de la production. En cas d'anomalie ou de dysfonctionnement, nous effectuons un diagnostic et mettons en place sur le champ une procédure de reprise de l'installation défectueuse.

Outre la sécurisation et le suivi au plus près de l'installation, un tel dispositif doit également nous permettre un gain de temps en cas de problème à l'installation. Il est en effet préférable de s'apercevoir tôt d'un problème qu'une fois l'installation complète réalisée ; la reprise n'en est que plus longue.

La figure 37 nous donne le macro-plan de l'installation en production.

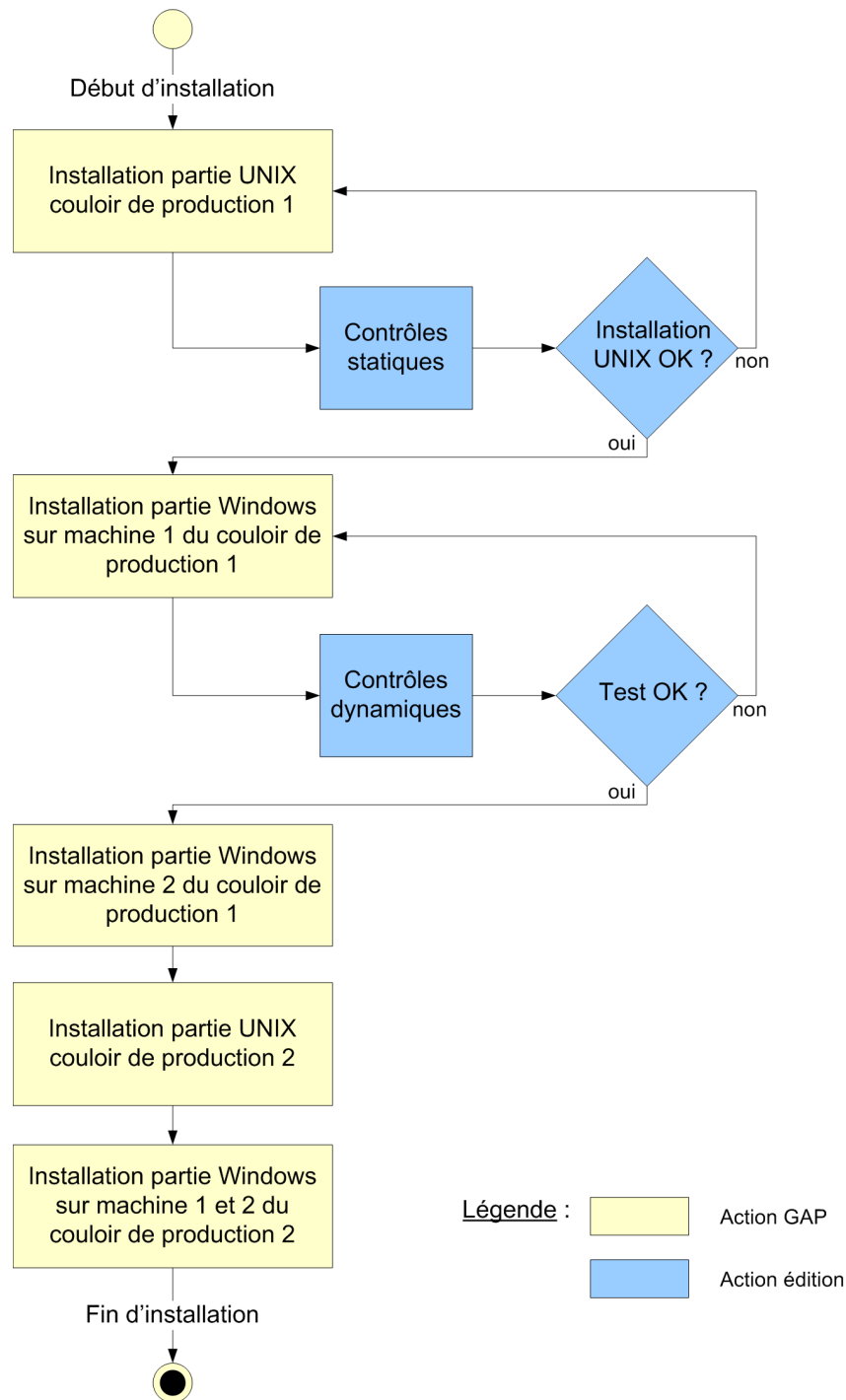


Figure 37 : Sécurisation de l'installation

## 7.2. Plan de démarrage

Une fois toutes les évolutions du chantier Unix/Windows installées sur les deux environnements de production de manière sécurisée, au regard de l'importance du projet et des délais imposés par les autorités de tutelle, un plan de démarrage doit être mis sur pied. Ce plan est une garantie de bon fonctionnement lors de l'utilisation réelle qui sera faite début juillet 2010.

Les délais sont courts car l'installation du projet est prévue le 21 juin 2010. Les établissements vont commencer de travailler sur les données de fin juin 2010 à partir du premier juillet. D'ici là, il faut nous assurer que tout fonctionne.

Le plan de démarrage va impliquer l'ensemble des établissements. L'équipe projet sera en relation avec chacun des utilisateurs afin que toutes les actions de vérification soient réalisées entre le 21 et le 30 juin. Ce délai nous laisse peu de marge en cas de dysfonctionnement.

Voici les actions qui seront réalisées soit par l'équipe projet soit par les utilisateurs :

- Connexion à l'application Evolan Report
- Ouverture d'une session
- Passage du batch Engine dans la nuit
- Vérification de la bonne alimentation des tableaux SURFI
- Vérification de la présence des références croisées
- Interrogation de la piste d'audit des ESTD (entrées standards)
- Production des contrôles XBRL
- Déclaration XBRL

Une telle procédure doit nous garantir qu'aucun dysfonctionnement n'interviendra à partir du 1<sup>er</sup> juillet 2010. Nous devons rappeler que les pénalités financières imposées par la Banque de France en cas de retard dans la remise des états financiers réglementaires sont très importantes.

## **8. Conclusion**

---

Nous venons de décrire le projet SURFI tel qu'il s'est déroulé et plus particulièrement les actions que nous avons entreprises pour le mener à bien.

Nous avons vu la méthodologie appliquée spécifiquement pour ce projet, le montage d'un environnement technique, l'utilisation d'un nouveau langage (XBRL). La phase d'analyse a été déterminante pour la réalisation qui s'en est suivie et les tests nous ont permis de mettre en œuvre des outils performants et complets. Enfin, un travail a été mené sur la préparation à la mise en production afin de sécuriser celle-ci.

Nous pouvons désormais effectuer une synthèse de ce projet ainsi qu'un bilan qui décrit les difficultés et les succès rencontrés.



# PARTIE III

## SYNTHESE ET BILAN

Ce projet nous a confronté à de multiples situations, nous a permis de mener différentes actions, d'effectuer des analyses débouchant sur des solutions. Nous avons été confrontés à des difficultés mais aussi des succès qu'il convient d'analyser afin de capitaliser notre expérience pour le futur.

Nous allons effectuer une synthèse des différentes actions que nous avons effectuées mais aussi tenter de faire un bilan du projet en expliquant les difficultés et les succès rencontrés.

### ***1. Synthèse des actions***

---

#### **1.1. Le point de départ : l'analyse**

Une partie importante et décisive du projet a été l'analyse :

- Importante notamment en terme de temps passé; ces analyses ont porté aussi bien sur les composants reçus de la part de Sopra et leur impact sur la plateforme MySys. que sur des sujets d'architecture des traitements MySys.
- Décisive car ces analyses ont conditionné la suite et permis la prise de décision

Cette phase d'analyse nous a permis de disposer de toutes les cartes pour ensuite proposer des solutions et choisir. Des analyses moins poussées n'auraient pas permis de tels gains que ce soit pour l'exploitabilité de l'application (nouvelle architecture des traitements) ou pour la mise en place des évolutions liées à SURFI (éviter la multiplication des anomalies).

#### **1.2. Proposer des solutions et choisir**

Un des objectifs d'une analyse est de dégager plusieurs solutions pour répondre à une problématique. La possibilité de regrouper des traitements en un seul fut



typiquement dans ce cas là. L'analyse de ce cas précis nous a permis d'exposer deux solutions à un groupe d'experts de l'application. Ces solutions ont été argumentées techniquement afin que les personnes puissent prendre une décision rapidement et en toute connaissance de cause. Une analyse moins poussée aurait très certainement fait planer un doute sur les conséquences de telle ou telle décision.

### **1.3. Mobiliser et coordonner**

La coordination a été une activité majeure dans notre chantier Unix/Windows. Différentes ressources sont intervenues à plusieurs reprises sur des environnements différents. Ainsi des administrateurs base de données, des installateurs de composants, des spécialistes du montage d'environnements ont participé à différentes phases du projet. Par ailleurs l'éditeur imposant ses contraintes, notamment le lotissement des livraisons du logiciel, cela a intensifié les besoins de mobilisation et de coordination des ressources. Constamment nous avons dû nous assurer de la disponibilité des ressources et ensuite coordonner leurs interventions selon un ordre donné. Un oubli d'une personne et c'est un retard important qui est donné au chantier. Cette tâche de mobilisation/coordination a donc été essentielle dans notre chantier et également très consommatrice de temps et d'énergie.

### **1.4. Tester**

Une tâche plus technique mais tout aussi importante est celle concernant les tests. Nous y avons également consacré un temps important pour que ceux-ci soient exhaustifs et représentatifs. Les résultats sont satisfaisants au regard du faible nombre d'anomalies détectées dans le cadre des recettes.

Un point également très positif est l'élaboration d'une procédure de tests qui sera réutilisée lors des prochaines montées de version ou autres projets sur SURFI. Il est important pour un projet de capitaliser l'expérience acquise afin de l'utiliser dans d'autres projets. Cet aspect n'est pas forcément évident au sein d'une entreprise comme la notre où les changements d'organisation relativement fréquents font bouger les équipes. Ceci entraîne d'une part de la perte d'informations et d'expérience mais aussi certainement n'encourage pas à capitaliser pour les futurs projets. Or c'est dans ces contextes qu'il est justement nécessaire d'emmagasiner un maximum d'expérience pour mieux la transmettre ensuite. Cette procédure de test devra perdurer aussi longtemps que l'application Evolan Report tourne sur la plateforme MySys.

### **1.5. Mise en production sécurisée**

Il peut paraître surprenant que ce soit l'équipe projet qui s'occupe de sécuriser une mise en production alors même qu'un service est dédié aux installations et donc plus à même d'offrir une sécurisation de cette phase. Nous avons fait le choix de nous engager dans cette sécurisation car peu de marge nous était laissée quant à la réussite de l'installation et les difficultés rencontrées par l'équipe d'installation au niveau des ressources. D'un commun accord avec la hiérarchie, nous avons accompagné l'installateur dans ses démarches.

## **2. Difficultés rencontrées**

---

Au cours du projet SURFI et plus particulièrement au sein du chantier Unix/Windows nous avons rencontré un certain nombre de difficultés liées soit au contexte de l'entreprise soit à des impondérables. Aucune de ces difficultés n'a mis en péril le chantier mais elles ont fait perdre du temps et amoindri la sécurité notamment lors de la mise en production.

### **2.1. Nouvelle organisation**

Indiscutablement la fusion des trois plateformes informatiques en une seule a amené beaucoup de changements et un manque de visibilité de l'organisation. Le « qui fait quoi » dans l'entreprise étant peu connu, des difficultés sont apparues dès le début du projet et ont perduré tout au long de celui-ci, chaque fois qu'il était nécessaire de solliciter une ressource pour la première fois. Ainsi, les problèmes rencontrés sur les performances Java ont nécessité de faire intervenir à moins de deux mois du déploiement des personnes non identifiées. Cette difficulté a eu pour conséquence principale d'augmenter les délais de réalisation.

### **2.2. Mobilisation des acteurs**

Nous l'avons vu, plusieurs ressources sont intervenues au sein du chantier pour notre compte et pour des tâches diverses. Ces interlocuteurs interviennent sur plusieurs projets en même temps et notre principale difficulté a été de les mobiliser pour qu'ils interviennent rapidement et complètement.

A titre d'exemple nous pouvons citer le cas de la mise à disposition de l'environnement de recette à volumétrie réelle où des dysfonctionnements sont apparus sur les transferts des états GEIDE entre la machine UNIX et le z/OS. Cette anomalie devait faire l'objet d'une analyse technique afin de détecter ce qui ne fonctionnait pas. A plusieurs reprises nous avons contacté la personne responsable sur ce sujet. A chaque fois elle débutait l'analyse mais ne la terminait jamais, appelée par un autre projet. Si ce dysfonctionnement ne revêtait pas un caractère d'urgence lorsqu'il a été décelé, le temps passant, il l'est devenu et nous avons émis une alerte. Ceci a eu pour conséquence l'intervention de la hiérarchie afin que l'intervention soit complète et que le dysfonctionnement disparaisse. Cet exemple montre les difficultés que nous avons eu à mobiliser et convaincre, un aspect aggravé par le fait que l'ensemble des ressources hors projet étaient basées sur des sites distants.

A l'avenir, nous garderons de ces difficultés la nécessité d'être plus ferme dès le départ et d'amener une argumentation plus convaincante ; encore une fois cela permettant un gain de temps et d'énergie.

### **2.3. Mise en référentiel non aboutie**

La mise en référentiel de l'application n'a pas abouti et n'a pu être utilisée par notre chantier. Rappelons que cette mise en référentiel consistait à intégrer les composants de l'application dans un logiciel de gestion de configuration logicielle.

Cela devait permettre le référencement de chaque version de composants et faciliter leur installation par le biais d'un outil d'installation automatique.

Les conséquences ont été importantes : nombreuses erreurs humaines lors de l'installation dans les environnements de recette et risques importants sur le déploiement d'où la mise en place d'un plan de sécurisation de l'installation ; en synthèse beaucoup de perte de temps.

Nous avons pris toute la mesure de la nécessité de disposer d'un outil de gestion de configuration logicielle au service de l'application. Nous allons donc tenter d'engager une nouvelle mise en référentiel afin que les projets futurs puissent être facilités et sécurisés au niveau de la gestion des composants.

### **3. Les succès**

---

#### **3.1. Respect des délais**

Malgré les difficultés le projet SURFI est toujours dans les délais. Au regard des conséquences financières d'une dérive des jalons du projet, le succès majeur est la tenue des délais. Le projet va être déployé dans les temps, nous pouvons dire que c'était l'objectif numéro un. Malgré les difficultés rencontrées, nous arrivons à temps pour permettre le déploiement dans de bonnes conditions. Toutes nos décisions ont été prises en fonction de cet impératif.

#### **3.2. Méthodologie**

Un deuxième succès concerne la qualité globale des évolutions apportées. Très peu de dysfonctionnements ont été révélés lors des recettes et ce succès est en partie dû à la méthodologie mise en place sur ce projet. Elle a permis une analyse rapide des évolutions à mettre en place et un suivi facilité des actions à entreprendre.

#### **3.3. Procédure de test**

La mise en place de la procédure de test est également un succès car elle a permis tout d'abord de sécuriser la mise en place des évolutions au sein de notre chantier, ensuite elle sera réutilisée dans les projets futurs sur cette application. Nous avons déjà souligné l'importance de la capitalisation de l'expérience de chaque projet. La mise en place d'une telle procédure en est un bon exemple.

#### **3.4. Responsabilité du chantier Unix/Windows**

Nous estimons que le chantier Unix/Windows est une réussite à la lumière des résultats. En effet, peu d'anomalies ont été constatées du fait d'une bonne maîtrise de l'application. Nos fonctions de coordinations des différents intervenants, malgré les difficultés de mobilisation, ont été remplies correctement.

## CONCLUSION

Cela fait plusieurs années que les organismes de tutelle des établissements financiers (Banque de France et Banque Centrale Européenne) ont mis en place un arsenal permettant de connaître la santé financière des banques. Ce système est basé sur la remontée périodique de rapports financiers obligatoires. La BAFI (Base de Agents Financiers) en fait partie depuis 1993, date de sa mise en place. Elle permet d'alimenter des bases statistiques monétaires et d'effectuer du contrôle prudentiel. La remontée des rapports financiers de la BAFI se fait dans un format propriétaire, lourd à maintenir et à faire évoluer.

SURFI (Système Unifié de Reporting Financier), projet réglementaire français débuté en 2007, impose aux établissements une réforme de la BAFI. Elle consiste à diminuer le nombre d'informations en supprimant les redondances et également à réduire les délais de remise des rapports financiers.

GCE Technologies, en tant que maîtrise d'œuvre du système d'information des Caisses d'Epargne devait faire évoluer la BAFI conformément au projet SURFI. Un nouveau projet a ainsi été lancé début 2009 au sein de notre entreprise. Ce projet a revêtu deux aspects : l'installation d'une nouvelle version du logiciel utilisé pour fabriquer les états réglementaires de la BAFI, Evolan Report, et des évolutions de données mises en entrée du logiciel. Ces deux aspects ont été transcrits en deux chantiers distincts ; nous avons pris part au chantier d'installation de la nouvelle version du logiciel, le chantier baptisé Unix/Windows.

A l'heure où nous écrivons ces lignes le projet n'est pas encore déployé, cette action est prévue pour la mi juin 2010. Mais nous pouvons d'ores et déjà effectuer un bilan positif. Nous avons mis en œuvre des analyses de natures différentes qui ont permis d'effectuer des choix entre différentes solutions ou encore qui ont validé des hypothèses. Cette phase a été enrichissante pour nous car nous avons été confrontés à différentes difficultés que nous avons réussi à contourner. Notons également notre succès dans la mise en œuvre d'une procédure de test robuste, fiable et réutilisable. Cette capitalisation de l'expérience nous semble indispensable dans un contexte organisationnel difficile où il est important de garder à l'esprit ce besoin de transmission du savoir.

Outre le déploiement des évolutions du projet SURFI, un projet SURFI lot 2 démarre actuellement pour se terminer en octobre 2010. Plus petit que le précédent, notre place y est tout naturellement et nous pourrons mettre en œuvre notre expérience acquise au sein du premier.

Ce projet nous a permis d'acquérir de fortes compétences sur l'application au niveau du socle technique Unix et Windows de l'application. Il nous a également apporté des clés de réussite pour les projets futurs que ce soit en termes d'organisation du travail, de méthodologie ou encore de coordination des ressources. La réussite de ce projet, support de notre mémoire d'ingénieur, nous engage à continuer dans cette voie et à poursuivre notre lancée vers une activité d'ingénieur au sein de GCE Technologies.



## LISTE DES FIGURES

Figure 1 : Organigramme financier du groupe BPCE.....	9
Figure 2 : Gouvernance du système d'information.....	10
Figure 3 : Organisation de GCE Technologies .....	12
Figure 4 : Deux instances de production.....	14
Figure 5 : Principe de la BAFI .....	16
Figure 6 : Organisation de la BAFI chez GCE Technologies .....	18
Figure 7 : Architecture du logiciel Evolan Report .....	19
Figure 8 : Etat Evolan Report.....	20
Figure 9 : Résultat d'une requête références croisées.....	21
Figure 10 : Les journées de la BAFI.....	23
Figure 11 : Architecture technique de la BAFI chez GCE Technologies .....	24
Figure 12 : Machines Windows.....	25
Figure 13 : Architecture des échanges de la BAFI.....	26
Figure 14 : Les ressources intervenant sur le projet .....	33
Figure 15 : Phases de SDM/S.....	34
Figure 16 : Les phases de Merise.....	35
Figure 17 : Méthodologie chantier Unix/Windows.....	37
Figure 18 : Planning du chantier Unix/Windows.....	38
Figure 19 : Découpage fonctionnel Evolan Report.....	40
Figure 20 : Extrait du découpage CM-Synergy "Project" Engine .....	41
Figure 21 : Document XBRL .....	45
Figure 22 : Avant XBRL, beaucoup de travail manuel .....	46
Figure 23 : Avec XBRL, parfaitement défini et prêt pour l'automatisation métier .....	47
Figure 24 : Planning des livraisons de l'éditeur Sopra .....	49
Figure 25 : Traitement Engine existant .....	51
Figure 26 : Diagramme de GANTT traitements Engine situation actuelle.....	52
Figure 27 : Diagramme de GANTT traitements Engine situation future .....	53
Figure 28 : Gestion des priorités en entrée d'Engine .....	55
Figure 29 : Circuit de transfert des déclarations actuel .....	57
Figure 30 : Circuit de transfert des déclarations futur .....	58
Figure 31 : Nouveau traitement de transfert des déclarations XBRL .....	60
Figure 32 : Interventions des ressources .....	63
Figure 33 : Extrait du plan d'action.....	65
Figure 34 : Arborescence BAFI.....	68
Figure 35 : Extrait de la procédure de test .....	71
Figure 36 : Table de vérité pour les tests.....	72
Figure 37 : Sécurisation de l'installation.....	76



## GLOSSAIRE

ARPEGE	Ancienne plateforme informatique qui avait en charge une partie des Caisses d'Épargne
Attribut	Un attribut est un critère de décomposition du solde d'un compte de comptabilité générale
BAFI	Base des Agents Financiers : Système français de remontée d'informations financières des établissements bancaires à destination des organismes de tutelle, Banque de France et Banque Centrale Européenne
BCE	Banque Centrale Européenne. Organe régulateur chargé de définir la politique monétaire de la zone euro
BO	Business Object. Logiciel permettant d'interroger des bases de données. Il offre une présentation métier des données
BPCE	Banques Populaires Caisse d'Épargne. Organe central à la tête de deux réseaux commerciaux : les Banques Populaires et les Caisses d'Épargne
COREP	COmmon solvency ratio REPorting. Projet de reporting prudentiel européen
CRI	Compte-Rendu d'Inventaire. Enregistrement informatique contenant des situations fin de mois différents du CRE, Compte-Rendu d'Événement qui est l'enregistrement informatique d'une opération.
DBA	Administrateur de bases de données
DCT	Document de Conception Technique. Documentation interne à GCE Technologies destiné à recenser les évolutions à opérer au niveau de l'industrialisation d'une application
ESTD	Entrée Standard. Format des données mises en entrée du progiciel Evolan Report
FRP	Finance Risque Pilotage. Département de GCE Technologies ayant en charge l'édition et la maintenance des applications de comptabilité, finance, risque, recouvrement
GAP	Gestion Application Projet. Service de GCE Technologies ayant la charge de l'industrialisation et de la mise en production des composants applicatifs



GCL	Gestion de Configuration Logicielle. Outil permettant la gestion des composants d'un système informatique ainsi que les modifications opérées sur le système.
GEIDE	Gestion Electronique d'Informations et de Documents de l'Entreprise. Outil permettant le stockage et la consultation de données de l'entreprise
GOP	Gestion Opérationnelle Projet. Service de GCE Technologies permettant la coordination des différents intervenants de la production lors d'un projet.
GCE BS	GCE Business Service. Entreprise du groupe BPCE assurant la maîtrise d'ouvrage du système d'information des Caisses d'Epargne
GIE	Groupement d'Intérêt Economique. Structure intermédiaire entre société et association. Elle permet de mettre en commun entre plusieurs membres leurs activités.
MOA	Maîtrise d'Ouvrage. Donneur d'ordre. Au sein du groupe BPCE, la maîtrise d'ouvrage est assurée par GCE Business Service
MOE	Maîtrise d'œuvre. Réalise les demandes de la maîtrise d'ouvrage. Au sein du groupe BPCE, cette fonction est assurée par GCE Technologies.
MySys	My System. Nom donné à la plateforme informatique des Caisses d'Epargne.
PIA	Plan Informatique Annuel. Ensemble des évolutions qui sont entreprises sur le système d'information dans une année. Ce plan est validé par BPCE.
RSI	Ancienne plateforme informatique qui avait en charge une partie des Caisses d'Epargne
SIRIS	Ancienne plateforme informatique qui avait en charge une partie des Caisses d'Epargne
TAR	Tape Archiver. Programme permettant l'archivage des fichiers Unix. Les fichiers archivés ont une extension .tar
VTOM	Visual TOM. Ordonnanceur permettant le lancement et l'enchaînement automatiques de traitements informatiques
z/OS	Système d'exploitation développé pour les gros ordinateurs de type « mainframe » par IBM. Ce système d'exploitation est couramment utilisé au sein des banques et des assureurs.

# REFERENCES BIBLIOGRAPHIQUES

## 1. Livres

---

- [DUE91] DUET Daniel, *Les Caisses d'Epargne*, PUF, 1991
- [LAF03] LAFITTE Michel, *Les grands projets de système d'information dans les établissements bancaire*, Revue Banque édition, 2003
- [HAM09] HAMON Pierre, *Comprendre XBRL et la taxonomie Comptes annuels*, BOD, 2009
- [HOF06] HOFFMAN Charles, *Financial Reporting Using XBRL : IFRS and US GAAP Edition*, UBMatrix, 2006
- [PUJ02] PUJOLLE Guy, *Les réseaux*, Eyrolles, 2002
- [XBR06] XBRL France, *XBRL : Introduction à la technologie XML des rapports financiers, livre blanc*, XBRL France  
<http://bnb.be/DOC/BA/XBRL/Pub/Livre%20blanc%20XBRL%20France.pdf> consulté le 18/04/10

## 2. Revues et publications

---

- [AGE09] AGEFI Hebdo, *La remise des 'reporting' à la commission bancaire fonctionne*, 2009

## 3. Sites internet

---

Banque de France, consulté le 18/04/10  
<http://www.banque-france.fr/>

BPCE, consulté le 18/04/10  
<http://www.bpce.fr/>

XBRL Business Information Exchange, consulté le 18/04/10  
<http://xbrl.squarespace.com/>

XBRL France, consulté le 18/04/10  
<http://xbrl.org/fr/>

Conférence sur la taxonomie SURFI et les formules XBRL, consulté le 18/04/10  
<http://www.thewebconsulting.com/media/index.php?2010/03/30/12027-conference-sur-la-taxonomie-surfi-et-les-formules-xbrl>

Réforme SURFI dans les établissements bancaires à échéance 2010 : la réponse concrète et globale de Viveo, consulté le 18/04/10

[http://www.cfo-news.com/Reforme-SURFI-dans-les-etablissements-bancaires-a-echeance-2010-la-reponse-concrete-et-globale-de-Viveo\\_a8936.html](http://www.cfo-news.com/Reforme-SURFI-dans-les-etablissements-bancaires-a-echeance-2010-la-reponse-concrete-et-globale-de-Viveo_a8936.html)

SURFI - Un pas de plus vers le "Fast and better close", consulté le 18/04/10

<http://www.otc-conseil.fr/fre/newsletters/lettre-n-39/surfi.html>

# RESUME

Installation de la nouvelle version d'un logiciel bancaire

Mémoire d'ingénieur CNAM, Lyon 2010

---

La BAFI (Base des Agents FINANcier), système permettant la remontée de données financières des établissements bancaires auprès des organismes de tutelle (Banque de France et Banque Centrale Européenne), fait l'objet d'une réforme intitulée SURFI (Système Unifié de Reporting Financier). GCE Technologie, maîtrise d'œuvre du système d'information des Caisses d'Epargne, met en œuvre un logiciel du commerce, intitulé Evolan Report, pour effectuer la production des rapports financiers BAFI. La réforme SURFI nécessite une évolution de ce logiciel et donc une nouvelle installation sur le système d'information des Caisses d'Epargne, appelé MySys.

Ce mémoire décrit le déroulement des évolutions apportées à la BAFI au sein de GCE Technologies et plus particulièrement sur l'installation de la nouvelle version du logiciel Evolan Report. Il permet également de souligner les différentes tâches que nous avons exécutées pour mener à bien la réforme notamment notre responsabilité du chantier Unix/Windows.

## Mots clefs

BAFI SURFI REPORTING BANQUE REGLEMENTAIRE XBRL

## *Summary*

The BAFI (Base des Agent Financiers), system allowing the reporting of financial data of banking institutions to supervisory bodies (Banque de France and European Central Bank), is touched by a reform, SURFI (Système Unifié de Reporting Financier). GCE technologies, in charge of the information system of the French Caisses d'Epargne, implements a software, entitled Evolan Report, to make the production of the BAFI financial reports. The reform SURFI requires an evolution of this software and thus a new installation on the information system of the french Caisses d'Epargne, entitled MySys.

This report describes the progress of the evolutions brought to the BAFI within GCE Technologies and more particularly the installation of the new version of the software Evolan Report. It also allows to underline the various tasks which we executed to go to a success conclusion of the reform in particular our responsibility of the Unix/Windows platform.

## *Keywords*

BAFI SURFI REPORTING BANK REGULATION XBRL