TABLE DES MATIÈRES

LISTE DES TABLEAUX

Page

LISTE DES FIGURES

LISTE DES ABRÉVIATIONS, SIGLES ET ACRONYMES

BPMN	Business Process Management Notation: un formalisme pour la représentation graphique des processus d'affaires
EDF	Resource Description Framework
ERDF	Embedded Resource Description Framework
FMC	Fundamental Modeling Concept : un formalisme pour modéliser les systèmes d'informations dynamiques d'une manière très structurée
GPA	Gestion de processus d'affaires, ce terme équivaut au terme anglais (Business Process Managment :BPM)
Hive	Un système d'entrepôt de données pour Hadoop. Il permet le regroupement des données, des requêtes ad-hoc et analyse de grands volumes de données
Hive	Un système d'entrepôt de données pour Hadoop. Il permet le regroupement des données, des requêtes ad-hoc et analyse de grands volumes de données
Hibernate	Est une bibliothèque formée de plusieurs composants logiciels pour gérer la persistance des objets des bases de données avec le langage Java
IaaS	Infrastructure as a Service
Informatique: de nuage	Ce terme équivaut au terme anglais «*Cloud computing* », utilisé pour la première fois en 2006 par le PDG de Google, Eric Schmidt. Il est souvent utilisé pour désigner une infrastructure composée de grappes de plusieurs ordinateurs offrant des services et ressources informatiques payants à des utilisateurs distants à travers l'internet.
IPV6	Internet Protocol Version 6
JSON	JavaScript Object Notation
PaaS	Platform as a Service
Plugiciel	Logiciel d'application complémentaire qui associé à un fureteur internet, ajoute de nouvelles fonctionnalitées. Plugiciel le terme recommandé par l'office québécois de la langue française pour le terme anglais : Plugin
Pig	Est une plateforme intégrée avec Hadoop pour l'analyse de grands volumes de données

SGPA	Système de Gestion de processus d'affaires, ce terme équivaut au terme anglais (*Business Process Management System : BPMS*
SaaS	Software as a Service
SGBD	System de Gestion de Base de données
SSH	Secure Shell : est un Protocol d'accès réseau qui permet d'échanger les données entre les différentes unités des réseaux à travers des canaux de transmission sécurisés
SVG	Scalable Vector Graphics
WF-Nets	Workflow Nets : Une variante de réseaux de Pétri

INTRODUCTION

1.1 Introduction au problème

La mondialisation de l'économie a imposé aux organisations un rythme de croissance et d'adaptation aux changements accéléré. La survie des entreprises est liée directement à leur performance et à leur tolérance aux changements, ainsi qu'à leur capacité d'adapter rapidement leurs processus d'affaires aux nouvelles réalités économiques mondiales et aux exigences organisationnelles qui en résultent. De plus, les organisations sont toujours à la recherche d'innovation quant à leur processus d'affaires afin de devancer la concurrence. La capacité d'adaptation nécessite une culture de gestion de processus d'affaires structurée et efficace.

L'efficacité de la gestion de processus d'affaires passe impérativement par l'utilisation d'outils leur permettant de modéliser et de simuler les processus avant leur implantation afin de réduire les coûts et maximiser les chances de succès des nouveaux processus. Ceux-ci, leur permettront de détecter les points faibles des processus d'affaires tels qu'identifier les points d'étranglements et les points qui ralentissent le cours prévu des activités et raffiner leurs processus d'affaires afin de les optimiser et d'augmenter leur efficacité.

Les fournisseurs de logiciels offrent une grande variété de systèmes de gestion de processus d'affaires (SGPA) afin de répondre à la demande sans cesse croissante des organisations. En effet, les analystes prévoient que le marché des systèmes de gestion de processus d'affaires , aux Etats Unis , passera de 500 millions de dollars en 2006 à six milliards de dollars en 2011. (Oracle Corporation 2008).

Malgré leur besoin croissant envers ces systèmes, les organisations sont confrontées aux coûts élevés de leur acquisition, de l'implantation et de support (Celia Wolf, Paul Harmon 2010). De plus, pour plusieurs d'entre elles, il est rare qu'un seul SGPA soit capable de répondre à tous leurs besoins. Ces problématiques peuvent les décourager à acquérir un

SGPA et par conséquent diminuer leur capacité concurrentielle. Il devient impératif que ces organisations aient accès à un SGPA complet à un coût abordable.

Afin de répondre à ce besoin, de nouveaux modèles d'affaires émergent actuellement pour offrir des SGPA's sous forme de service logiciel, ce qui permettrait aux entreprises de ne payer que le temps d'utilisation de ces systèmes, leur permettant ainsi d'y avoir accès plus facilement et de mieux contrôler leurs coûts.

Ces modèles d'affaires sont désormais possibles grâce au concept émergeant de l'informatique de nuage : un concept permettant d'offrir les systèmes informatiques sous forme de service à travers l'internet. Le défi consiste à rendre les SGPA compatibles avec ce nouveau modèle d'affaires.

Afin de faire avancer ce domaine de recherche, le travail consiste à adapter, dans la cadre de la présente recherche un logiciel libre de gestion de processus d'affaires pour le rendre opérationnel sur une plateforme d'informatique de nuage.

1.2 Méthodologie de la recherche

Notre recherche consiste à investiguer comment effectuer la migration d'un logiciel libre de gestion de processus d'affaires vers la technologie de l'informatique de nuage. Ce problème n'est pas complètement résolu.

Pour définir cette recherche, la démarche, basée sur le cadre de Basili adapté pour les études exploratoires (Annexe 1) (Alain Abran, Lucie Laframboise et al. 1999) sera utilisée. La recherche sera structurée en quatre phases (voir tableau 1):

Tableau 1.1 Démarche méthodologique basée sur le cadre de Basili

Phase du cadre de Basili	Démarche méthodologique de notre étude
Définition	1. **Identification du problème** : Comment migrer un logiciel vers la technologie de l'informatique de nuage. 2. **Portée et objectifs de la recherche :** Le but principal de cette étude est d'effectuer l'analyse d'impact pour effectuer la migration. 3. **Utilisateurs des résultats de la recherche :** Le résultat final de cette étude est un prototype fonctionnel qui peut être utilisé par des entreprises afin de modéliser leurs processus d'affaires sur Hadoop
Planification	1. **Étapes de la phase de planification :** Réviser la littérature, faire un inventaire des logiciels libres de gestion de processus d'affaires, identifier un logiciel libre candidat de gestion de processus d'affaires et effectuer l'analyse d'impact de la migration du logiciel existant vers la plateforme de Hadoop **2. Intrant de la phase de planification :** Logiciel libre de gestion de processus d'affaires, logiciel libre d'informatique de nuage : Hadoop, un processus d'affaires à modéliser et critères de succès pour l'évaluation de l'essai. **3. Liste des livrables de la phase de planification :** Chapitre 1 du mémoire décrivant : l'état de l'art du domaine, un sommaire de la technologie Hadoop, la liste des logiciels libres de gestion des processus, l'identification d'un logiciel libre de gestion de processus et les raisons du choix.
Exécution	Description des modifications nécessaires du logiciel libre candidat afin qu'il puisse fonctionner sur le Cloud (avec Hadoop).
Interprétation de résultat	Évaluation des activités de migration nécessaires, contexte d'interprétation, extrapolation des résultats et travaux futurs.

1.3 Problématique

Les systèmes de gestion des processus d'affaires (SGPA) sont des outils suceptibles d'être utilisés par les organisations afin d'améliorer leur performance et leur efficacité. Cependant, plusieurs contraintes ont découragé les gestionnaires à acquérir cette technologie et profiter de tout son potentiel :

- Les SGPA de par leur nature complexe, nécessitent souvent des investissements majeurs en ressources matérielles, logiciel et humaines. Plusieurs organisations n'ont pas les moyens de se doter de cet outil de gestion.
- La globalisation de l'économie a donné naissance à des processus d'affaires inter organisationnels. Les SGPA déployés dans les infrastructures informatiques de la majorité des organisations ne supportent pas ce genre de processus.
- La diversité des activités des organisations rend impossible qu'un seul SGPA réponde à toutes les exigences d'affaires d'une organisation.

Dans le but d'atténuer l'impact de ces contraintes et d'encourager les gestionnaires à adopter cet outil, plusieurs fournisseurs de systèmes de gestion de processus d'affaires ont commencé à explorer les potentiels de la technologie de l'informatique de nuage[1] et à développer des SGPA qui puissent fonctionner sur cette nouvelle technologie. Le développement de ces systèmes dans un contexte industriel protégé, ne permet pas la vulgarisation du savoir et l'accessibilité de la technologie à tous ceux qui en ont besoin.

Dans la perspective de faciliter l'accès des organisations à cette technologie et vulgariser la connaissance dans ce domaine, il a été décidé d'étudier comment adapter un logiciel de gestion de processus d'affaires libres pour qu'il puisse fonctionner sur la technologie de l'informatique de nuage de Hadoop.

---

[1] L'informatique de nuage est un concept technologique émergent et très prometteur ou les services informatiques matériels et logiciels sont fournis sur demande aux organisations à travers l'Internet et par le biais de fournisseurs centralisés

La question à laquelle la recherche tente de répondre est : en quoi consisteraient les modifications nécessaires afin de migrer la technologie d'un logiciel de gestion de processus d'affaires existant vers la technologie de l'informatique de nuage (Hadoop)?

1.4 Portée de la recherche

Dans le but d'améliorer l'accessibilité et la disponibilité de cette technologie pour les organisations, cette recherche vise à intégrer la technologie des systèmes de gestion de processus d'affaires à la technologie d'informatique de nuage en utilisant les logiciels libres.

Les systèmes de gestion de processus d'affaires sont des systèmes complexes et sont composés de plusieurs composants, à savoir, les outils de modélisation de processus, d'exécution des processus modélisés, de simulation, d'analyse, de surveillance, etc.

Cette recherche portera seulement sur l'analyse d'impact (étude des changements nécessaires) afin d'effectuer la migration d'un logiciel libre de modélisation de processus d'affaires vers Hadoop : un logiciel libre d'informatique de nuage. Les composantes d'exécution, d'analyse, de simulation et de surveillance sont hors portées de cette recherche.

1.5 Objectif de la recherche

Afin de réaliser cette recherche, les étapes suivantes ont été planifiées:

1. Identifier un logiciel libre (candidat) de modélisation de processus d'affaires propices à être étudié pour une migration sur Hadoop
2. Effectuer l'analyse de l'impact de la migration de ce logiciel afin qu'il fonctionne sur Hadoop
3. Effectuer une expérimentation de migration sommaire
4. Présenter un résumé des résultats de cette recherche.

1.6 Utilisateurs de la recherche

Le résultat espéré, de cette recherche, est de définir les étapes afin d'effectuer la migration d'un logiciel existant (qui utilise la technologie de base de données relationnelle) vers la technologie de l'informatique de nuage (qui utilise des bases de données NO-SQL).

CHAPITRE 2

REVUE DE LA LITTÉRATURE

Ce chapitre, présente la revue de littérature des cinq thèmes principaux de cette recherche :

1. La notion et systèmes de gestion des processus d'affaires,
2. Le concept d'informatique de nuage ('Cloud Computing'),
3. La gestion des processus d'affaires avec l'informatique de nuage,
4. Le logiciel libre de l'informatique de nuage : Hadoop,
5. Les logiciels libres actuels de gestion de processus d'affaires.

2.1 Gestion des processus d'affaires (GPA)

La notion de gestion de processus d'affaires a plus de trois décennies, malgré cela, il n'y a toujours pas un consensus sur ce qu'est la GPA et ce qu'elle n'est pas. Cette section permet de clarifier ce concept.

2.1.1 La gestion des processus d'affaires

L'évolution de l'économie a fait de la performance des organisations la préoccupation principale des gestionnaires. À l'origine, les entreprises étaient structurées autour de leurs unités fonctionnelles comme les ventes, les finances, les ressources humaines, la production, etc. Chaque unité fonctionnelle était gérée par rapport à ses objectifs internes et la performance de l'organisation était mesurée par rapport aux objectifs individuels de chacune d'elles. Afin de briser ces barrières, la technologie de l'information a mis à la disposition des organisations plusieurs progiciels comme les ERP et les CRM qui permettent la communication inter-unités d'une manière interactive. L'évolution de l'économie, la globalisation et la férocité de la concurrence ont tous créé de nouveaux besoins chez les gestionnaires que la manière classique de gestion en silo ne pouvait plus satisfaire les besoins

de la clientèle à savoir : le besoin d'un transfert rapide des informations d'une unité fonctionnelle à une autre, la prise de décisions rapide, l'adaptation rapide aux changements, etc. Ces exigences ont donné naissance à une nouvelle discipline qui est la gestion électronique des processus d'affaires.

Depuis la naissance de cette discipline, plusieurs définitions ont été proposées dans la littérature. En effet, Elzinga (Elzinga, Horak et al. 1995) propose que la GPA est une approche structurée et systématique d'analyse, d'amélioration, de contrôle et de gestion des processus d'affaires dont le but est d'améliorer la qualité des produits et services. Zairi (Zairi 1997) propose que la GPA est une approche structurée qui consiste à analyser continuellement et améliorer les fonctions fondamentales de l'organisation, i.e. les activités reliées à la production, à la publicité, aux communications et autres. Recker (Recker, Indulska et al. 2006), lui, stipule que la GPA est une façon structurée, cohérente et consistante de comprendre, modéliser, analyser, simuler, exécuter et continuellement changer les processus d'affaires de bout en bout.

Même s'il n'y a pas un consensus autour d'une définition unifiée de la GPA, il y a un consensus décrivant que la GPA doit être une approche structurée et systématique.

2.1.2 Cycle de vie de la GPA

La diversité des définitions de la GPA a conduit à une diversité de points de vue des auteurs sur la définition du cycle de vie de la gestion des processus. Cette recherche, prend en compte le cycle de vie (voir figure 2.1) (van der Aalst 2004) :

- **Modélisation de processus :** représentation graphique des processus actuels ou futurs, elle peut inclure les tâches manuelles ou automatiques
- **Configuration du système :** cette phase inclue la préparation de l'environnement où les processus modélisés vont être exécutés. Cela inclut la définition des rôles des employés, la structure organisationnelle, les critères de performance, etc.

- **Implantation de processus :** Dans cette phase, les processus d'affaires modélisés dans la phase de modélisation sont déployés et exécutés dans le système de gestion des processus d'affaires.
- **Diagnostic de processus :** Les analystes, en utilisant les outils de surveillance et d'analyses, surveillent la performance des processus déployés, analysent les faiblesses et proposent des améliorations.

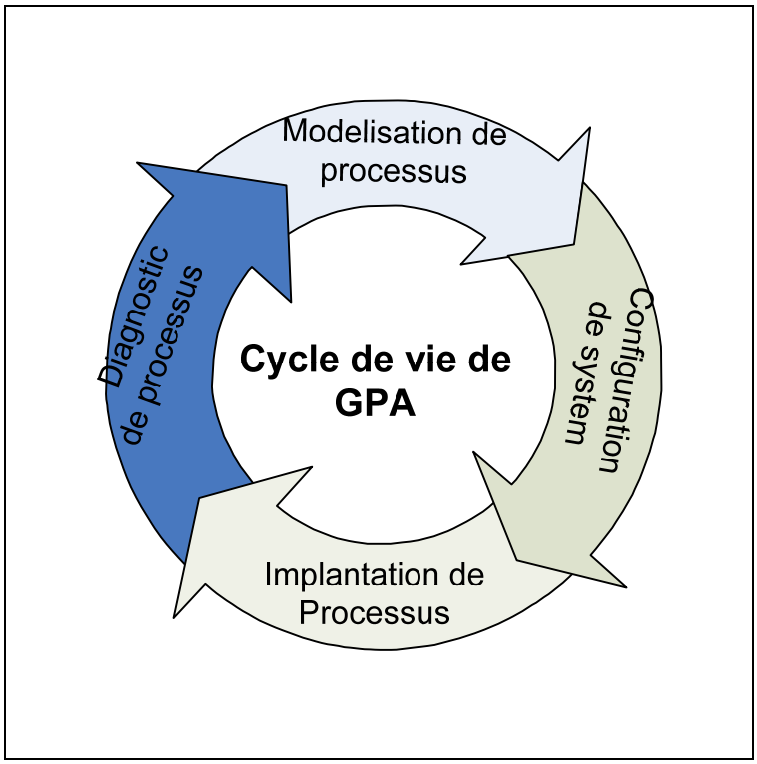

Figure 2.1 Cycle de vie de GPA
Tirée de van der Aalst (2004)

Ce cycle de vie a été implanté par plusieurs fournisseurs de logiciels de gestion de processus d'affaires, par exemple : Appian, IBM, Oracle, Lombardi, Pegasystems, TIBCO et Software AG. Chacun de ces fournisseurs a développé ses propres outils : des outils de conception, de modélisation, de simulation, de gestion des règles de gestion, d'intégration, d'exécution et ceux de surveillance et d'optimisation. Ces logiciels sont coûteux. Il y a actuellement plusieurs propositions de logiciels libres qui surgissent. La prochaine section traite de ce sujet.

2.1.3 Les logiciels libres de gestion de processus d'affaires

Un grand nombre de logiciels de modélisation et de gestion des processus d'affaires sont offerts en logiciels libres, par exemple : Enhydra Shark, jBPM, wFMOpen, Intalio, ProcessMaker, uEngine et OpenWFE (java-source.net 2010; Manageability 2010). Un logiciel libre est un logiciel offert sous les conditions de certains types de licences garantissant une circulation libre du code source. Plusieurs organisations ont adopté cette catégorie de logiciels de GPA pour leurs coûts d'acquisition abordables, la facilité de leur adaptation aux besoins propres des organisations et un service de support gratuit offert par la grande communauté des logiciels libres (Harmon 2007). (Harmon 2007; Wohed, Russell et al. 2009) ont identifié jBPM, OpenWFE et Enhydra Shark comme étant les logiciels libres de gestion de processus d'affaires les plus populaires de cette catégorie de logiciels. Le graphique suivant (figure 2.2) donne un aperçu sur l'historique de téléchargement des logiciels libres de GPA à partir des dépôts de code source : SourceForge, RubyForge, ObjectWebForge, Tigris.org, BountySource, BerliOS, JavaForge, et GNU Savannah.

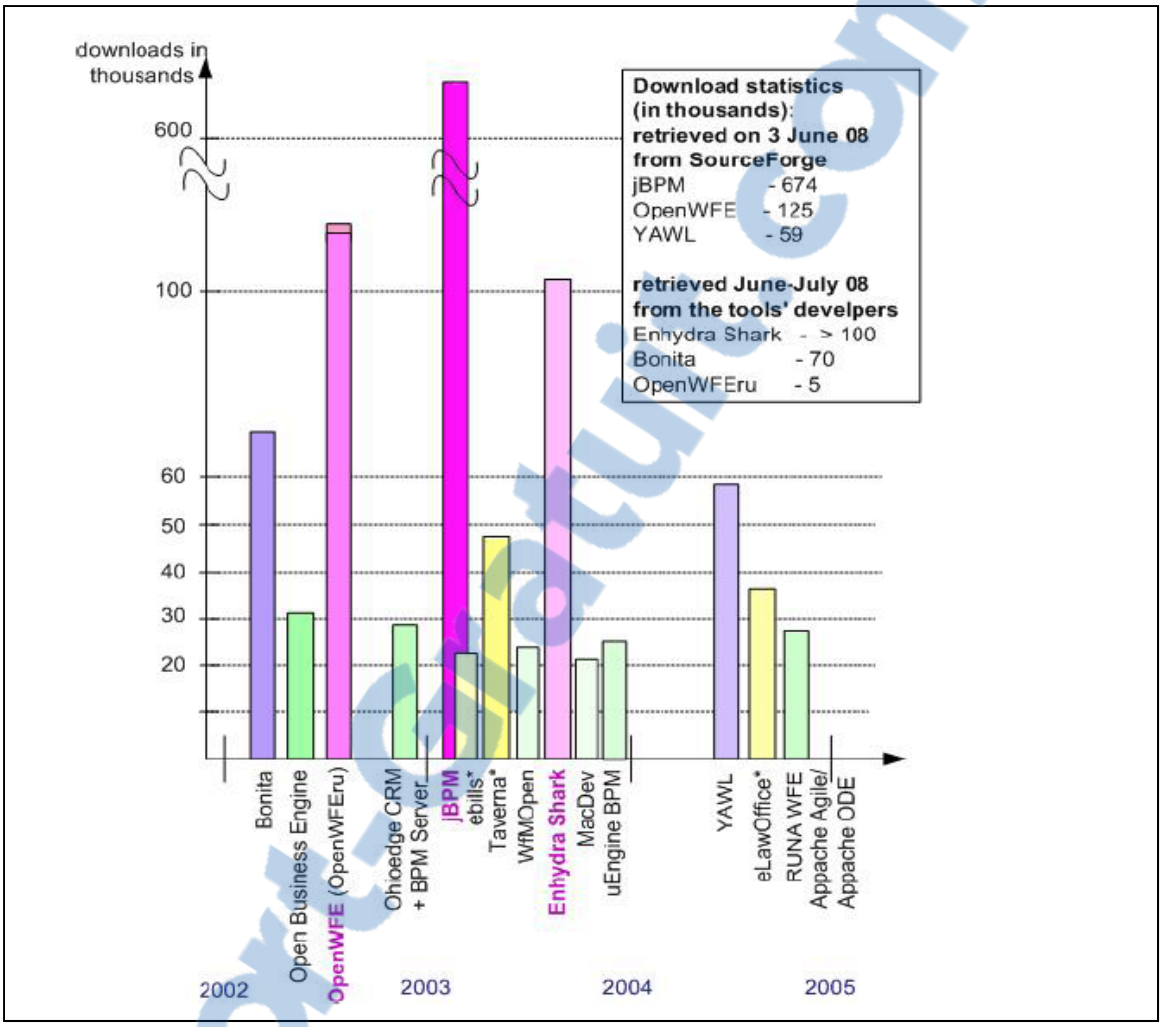

Figure 2.2 Liste de GPA la plus téléchargée
Tirée de Wohed, Russell et al. (2009)

Trois SGPA se distinguent par leur nombre de téléchargements au dessus du seuil de 100,000: JBPM, OpenWFE et Enhydra Shark. JBPM est de loin le plus téléchargé avec 600,000 téléchargements. L'étude comparative effectuée par Wohed et al. (Wohed, Russell et al. 2009) précise que les trois SGPA présentent des fonctionnalités et caractéristiques très similaires, à l'exception de OpenWFE, qui offre plus de fonctionnalité que les deux autres dans le volet du contrôle des flux de processus d'affaires.

2.2 Informatique de nuage

2.2.1 Informatique de nuage: qu'est ce que c'est ?

L'informatique de nuage, un nouveau terme populaire dans le domaine du logiciel. Les premières apparitions de ce terme datent de 2006 et avaient un sens restreint de l'informatique sur l'Internet (Aymerich et al., 2008). Depuis, plusieurs études ont été entamées pour définir ce nouveau concept technologique. Actuellement, et à cause de la nouveauté du domaine, les opinions divergent toujours en ce qui concerne la définition de ce concept. Souvent ces définitions sont personnalisées par les objectifs de recherches des chercheurs ou par les intérêts commerciaux des industriels.

Certains auteurs définissent le concept d'informatique de nuage comme un ensemble d'applications informatiques, les systèmes qui supportent ces applications et tout matériel nécessaire pour leur fonctionnement fournis comme un service rendu à travers l'Internet et géré par une entente de niveaux de service entre l'utilisateur et le fournisseur de service (Michael Armbrust, Armando Fox et al. 2009). Alors que Foster (Ian Foster, Yong Zhao et al. 2008) introduisent dans leur définition des notions de 1) paradigme informatique spécialisé et largement distribué, 2) encapsulation des services informatiques dans des entités abstraites fournies sur demande à des clients externes au nuage informatique, 3) dirigeable par l'économie d'échelle, 4) échelonnable et 5) virtuel.

Les publications accessibles permettent de voir que les mêmes concepts reviennent dans la majorité des définitions, mais avec des degrés d'importance différents. Vaquero et al. (Luis M. Vaquero et al, 2009) définissent l'informatique de nuage comme un : service informatique fourni sur demande aux clients via une entente de service. Ces services sont échelonnables, dans un environnement dynamique, permettant la réduction et l'optimisation des coûts d'acquisition et d'opération. Certains auteurs ont ajouté la notion de convivialité des services fournis à travers le nuage informatique. Cette recherche utilise la définition suivante de l'informatique de nuage : l'informatique de nuage est un concept qui permet aux

organisations d'acheter les droits d'utilisation des infrastructures informatiques, matérielles et logiciels offerts par les fournisseurs de services informatiques et gérés par des ententes de service assez flexibles. Elles peuvent ainsi se réapprovisionner en infrastructures informatiques, logiciels et matériels au niveau de performance et d'efficacité comparable à celui des services informatiques gérés par les organisations elles-mêmes, accessibles via n'importe quel périphérique capable d'exécuter un fureteur et ce de manière fiable, rapide et à des coûts très avantageux par rapport à l'acquisition de logiciels ou de matériels.

2.2.2 Les types d'informatique de nuage

Il y a trois types d'informatique de nuage : public où les services sont fournis aux clients via l'Internet par des fournisseurs externes, privé où les services sont fournis et gérés à l'intérieur de l'organisation, et hybride où les services fournis par l'informatique de nuage privé et ceux de l'informatique de nuage public sont combinés (ISO 2010).

Le type de nuage utilisé dépend essentiellement de la nature de services offerts, lesquels, s'inscrivent dans les trois catégories suivantes : 1) Infrastructure comme service, connue par le terme anglais *IaaS (Infrastructure as a Service)* : les fournisseurs de service internet gèrent un grand ensemble de ressources informatiques (le stockage des données, la capacité de traitements, etc). Avec la virtualisation de ces ressources, ils les redimensionnent, les partagent et les offrent aux clients selon le besoin exprimé, 2) Plateforme comme service, connue avec son appellation anglaise *PaaS (Plateform as a Service)* : Au lieu de fournir une portion d'infrastructure avec la virtualisation, les fournisseurs offrent à leurs clients la plateforme où s'exécutent les applications. Le meilleur exemple de ce type de service est : Google Apps Engine. 3) Logiciel comme service, connu sous l'appellation anglaise *SaaS (Software as a Service)* : les applications sont mises à la disposition des clients sur demande, une alternative à l'exécution des applications localement. (Luis M. Vaquero et al, 2009)

2.3 GPA sur l'informatique de nuage

Les concepts de gestion de processus d'affaires et celui de l'informatique de nuage ont tous deux apparu pour répondre à un besoin d'adaptation aux changements dans les organisations. L'objectif de l'étude est d'utiliser ces deux technologies. Cependant, la question de l'utilisation des logiciels de gestion des processus d'affaires avec la technologie de l'informatique de nuage n'a pas été publiée dans les milieux académiques. Par contre, dans les milieux industriels, elle émerge actuellement chez les fournisseurs de logiciels et de services informatiques. Les informations techniques publiées par ces compagnies sont rares et sont teintées de positionnement marketing propres à leurs objectifs commerciaux. Ces fournisseurs de logiciels se sont intéressés au concept de la GPA offerte sur la technologie de l'informatique de nuage à cause de l'importante opportunité d'affaire qu'elle représente. En effet, selon Michele Cantara, analyste chez Gartner, les parts des investissements au Etats Unis qui visent l'intégration des processus d'affaires avec l'informatique de nuage devraient augmenter de 500 % d'ici l'année 2012 (Cantara 2008).

La revue de différentes déclarations des fournisseurs de logiciels de gestion de processus d'affaires utilisant la technologie de l'informatique de nuage a permis de constater qu'il y a un consensus autour de l'utilité et de l'opportunité de ce concept. Toutefois, il faut remarquer qu'il y a des divergences d'opinion concernant le modèle le plus approprié de déploiement de la GPA sur la technologie d'informatique de nuage. Jeff Kaplan, fondateur et directeur de THINKstrategies Inc, soulève la question du défi de la dispersion géographique de la main d'œuvre dans les grandes organisations et propose le modèle *SaaS* (ebizQ ,2009). Ce modèle est aussi proposé par Prakash Kannoth, directeur chez ITQuadrant, pour résoudre la problématique des coûts de ce genre de système (ebizQ ,2009). Alors que John Pyke, stratège chez Cordys, pour sa part propose d'offrir la GPA comme plateforme sur le nuage au lieu de logiciel ou processus comme service sur l'Internet.

La divergence de la vision des fournisseurs de logiciels de GPA sur ce qu'est la GPA, et le modèle d'architecture d'informatique de nuage qui lui est le plus approprié a mené vers un

grand nombre de solutions de GPA sur des architectures d'informatique de nuage différentes: BlueWorks de IBM ,Interstage de Fujistsu, SmartBPM Suite 5.1 de Pegasystms, Oracle Fusion de Oracle, etc.

2.4 Hadoop : un logiciel libre d'informatique de nuage

2.4.1 L'informatique de nuage avec Hadoop

Selon EMC, le volume des données numériques dans le monde crées entre 2006 et 2011 va augmenter de 60%. C'est l'équivalent de 10 fois plus que le volume créé en 2006 (The Diverse and Exploding Digital Universe, 2008). La quasi-majorité de ces données est complexe et non structurée. Afin de répondre à ce besoin, la fondation du logiciel Apache a initié le projet de Hadoop. C'est un projet de logiciel libre fondé sur la plateforme d'Apache et inspiré de l'infrastructure de Google (Fay, Dean et al. 2008). Hadoop est développé en Java et permet sa portabilité entre les différents environnements : Windows, Solaris, Linux, Mac OS/X (Shafer 2010). Actuellement, Hadoop est utilisé par plusieurs fournisseurs de logiciels et de services informatiques, par exemple : Amazon web services, Facebook, Adobe, IBM, AOL, Yahoo, Google, etc. Le projet Hadoop a pour objectif de développer un outil qui permet le traitement de grands volumes de données, de manière économique, en utilisant des ordinateurs de commodité. Hadoop permet la gestion efficace d'un grand nombre d'ordinateurs de commodité (personnels) en permettant le traitement parallèle des données dans plusieurs ordinateurs en même temps. Hadoop assure ainsi la fiabilité grâce à la duplication des données à travers les nombreux ordinateurs qui composent les grappes d'ordinateurs.

Le projet Hadoop est composé de plusieurs sous projets inter reliés (Hadoop 2010) que les utilisateurs utilisent pour interagir avec le noyeau de Hadoop (figure 2.3).

- **Noyau de Hadoop** : Composé d'un système de gestion de fichiers qui permet la gestion de l'emmagasinage des ressources distribuées à travers une grappe d'ordinateurs de commodité et qui utilise le modèle de programmation parallèle *MapReduce*,

- **HBase :** La base de données de Hadoop, modélisée avec le principe de base de données de Google *Bigtable*,
- **Chukwa :** Un système de collection de données pour la gestion et la surveillance de systèmes distribués sur Hadoop.
- **HDFS :** Un système de gestion de fichiers distribués. C'est le système d'emmagasinage de ressources principales utilisées par les applications de Hadoop.
- **Hive :** une infrastructure d'entrepôt de données à utiliser par les applications de Hadoop. Il permet le compactage, l'interrogation et l'analyse des données des systèmes emmagasinés dans le système de gestion de fichiers distribués de Hadoop.
- **MapReduce :** un modèle de programmation parallèle et un environnement de développement de logiciels distribués.
- **Zookeeper** : Un service centralisé pour la maintenance des informations de configuration, des appellations et de synchronisation des applications distribuées.

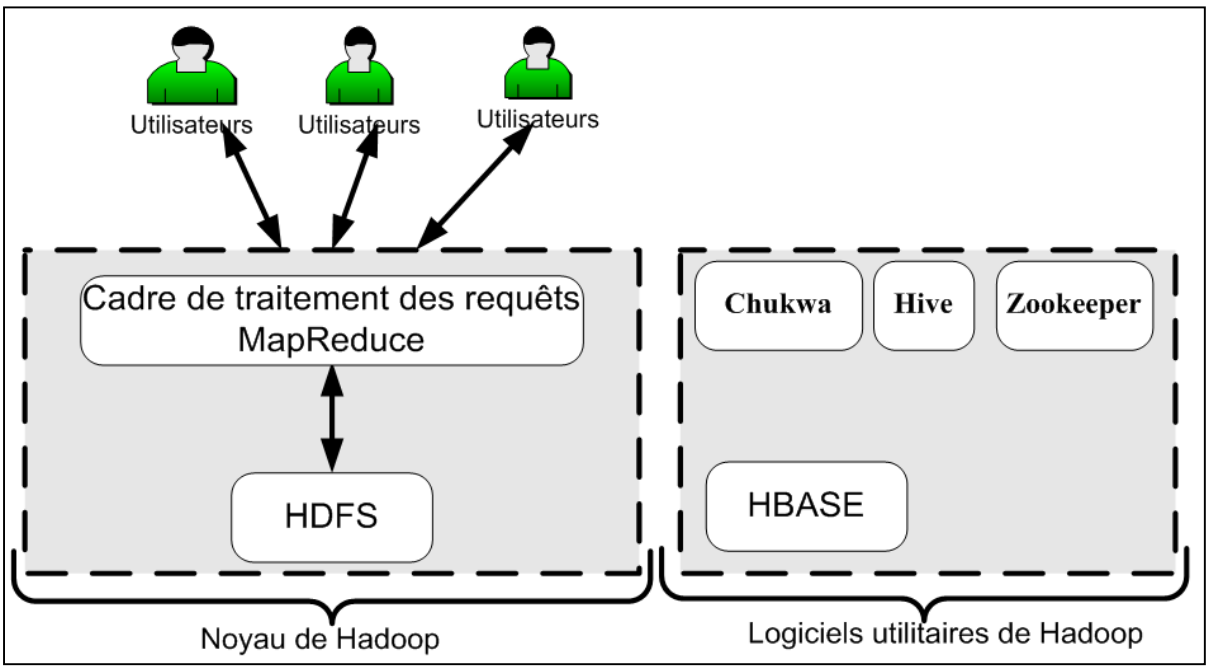

Figure 2.3 Les composants de Hadoop
Tirée de Borthakur (2007)

2.5 Système de stockage de données de Hadoop : HDFS

HDFS (*Hadoop Distributed File System*) est le système de gestion de fichiers distribués utilisé par les applications de Hadoop. Développé en Java pour des considérations de portabilité, il peut-être déployé sur des plateformes hétérogènes. Le concept de HDFS a été inspiré du concept du système GFS de Google (*Google File System*). Sa force réside en sa capacité de traiter de grands volumes de données complexes et non structurées avec une grande tolérance aux fautes. Il est dit « distribué » parce que les données des applications et des utilisateurs sont emmagasinées dans une ou plusieurs grappes d'ordinateurs de commodité inter-reliés.

Le système de fichiers distribués de Hadoop (HDFS) utilise une architecture maître/esclave. Chaque grappe d'ordinateurs HDFS est composée d'un seul nœud de nom (*NameNode) :* un serveur maître qui gère le système de fichier de l'espace de nommage, un nœud secondaire pour l'emmagasinage et la gestion des journaux de transactions effectuées sur le système de fichier et plusieurs nœuds de données (*DataNodes*) qui gèrent les données enregistrées sur ces nœuds. Les données des utilisateurs sont emmagasinées dans des fichiers, chaque fichier est subdivisé en blocs de petites capacités : souvent 64 Mb et parfois 128MB. La gestion de ces blocs de données est assurée par le serveur *NameNode*. Il s'occupe du mappage des blocs avec les nœuds de données, d'ouverture, de fermeture et de reformage des fichiers et des répertoires. Les nœuds de données sont responsables de la gestion des blocs. Ils assurent la création, la suppression, la réplication des blocs et les requêtes de lectures et écritures. Les données dans les nœuds de données sont enregistrées dans un système de gestion de fichiers traditionnel comme ext3.(figure 2.4) (Borthakur 2007).

Le gestionnaire du nœud des noms (NameNode) et celui de données (DataNode) sont des composants logiciels développés en Java qui peuvent être exécutés sur n'importe quelle plateforme qui supporte ce langage.

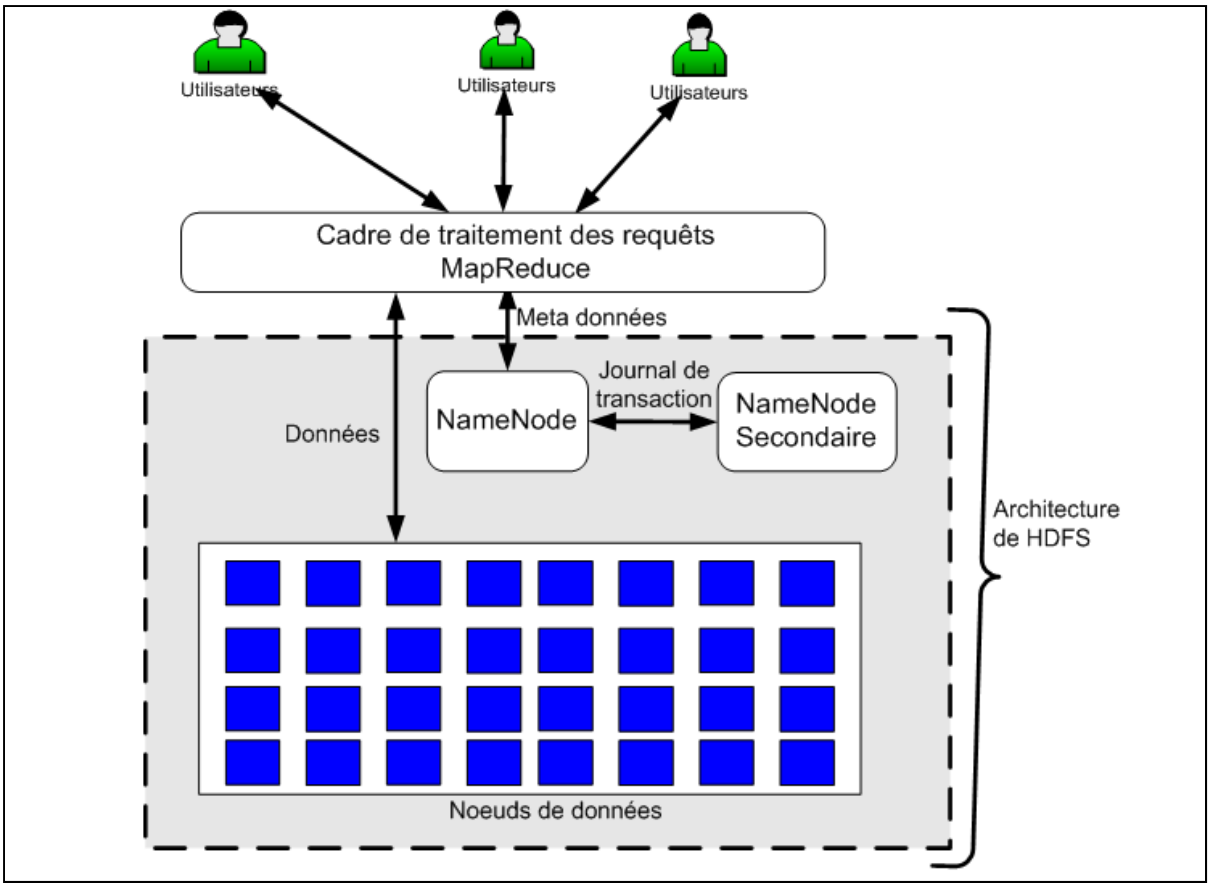

Figure 2.4 L'architecture de HDFS
Inspirée de Borthakur (2007)

2.6 Environnement de traitement de Hadoop : MapReduce

MapReduce est un nouveau modèle de programmation inventé par Google (Cryans, April et Abran, 2008). En utilisant ce modèle de programmation, Hadoop a créé un cadre de développement d'applications capable de gérer de grands volumes de données (Multi terabytes) résidant sur plusieurs grappes d'ordinateurs de commodités, d'une manière parallèle et avec une grande tolérance aux fautes (Map/Reduce Tutorial 2010)

2.7 Le modèle de programmation de MapReduce

Le principe de base de ce modèle de programmation est inspiré des fonctions de mappages et de réductions des langages de programmations fonctionnelles comme LISP : combiner deux fonctions pour effectuer des opérations sur des ensembles de données volumineux de l'ordre du petabyte (1 million de gigabyte):

- Fonction de mappage qui prend en entrée un ensemble de données et les transforme en un ensemble de couples :(clé, valeur), un couple pour chaque donnée reçue par la fonction en entrée qui seront triées par rapport aux valeurs de clés.
- Fonction de réduction qui a pour entrée les couples générés par la fonction de mappage et les transforme en une autre liste de couples (clé, valeur) plus réduite. La réduction de la liste initiale est faite en fusionnant les clés de chaque couple.
- Fonction principale qui combine les deux fonctions précédentes avec les opérations de contrôle de tâches et d'entrées/sorties de fichiers.

Les fonctions de mappages et de réductions sont écrites par les utilisateurs, les notions de clés et valeurs varient selon le contexte et selon le domaine des données manipulées par les fonctions de mappage et de réduction. Ce modèle peut être appliqué à plusieurs cas dans la vie réelle. Ci-dessous sont présentés quelques exemples inspirés du document de Jeffrey Dean and Sanjay Ghemawat (Jeffrey Dean and Sanjay Ghemawat 2004) du langage MapReduce.

Exemple : Compter le nombre d'occurrences des mots dans une grande collection de documents.

L'utilisateur écrit les fonctions de mappage et de réduction (figure 2.5) :

```
map(String key, String value):
  // key: document name
  // value: document contents
  for each word w in value:
    EmitIntermediate(w, "1");

reduce(String key, Iterator values):
  // key: a word
  // values: a list of counts
  int result = 0;
  for each v in values:
    result += ParseInt(v);
  Emit(AsString(result));
```

Figure 2.5 Exemple des fonctions Mapp et reduce
Tirée de Jeffrey Dean and Ghemawat (2004)

La fonction de mappage retourne une liste de couple : (mots, 1), la fonction de réduction va sommer tous les compteurs retournés pour chaque mot et obtenir ainsi une liste plus réduite.

Fichier_1 : mapreduce.txt = “MapReduce est un composant de Hadoop ”
Fichier_2 : HDFS.txt =”HDFS est un autre composant de Hadoop »

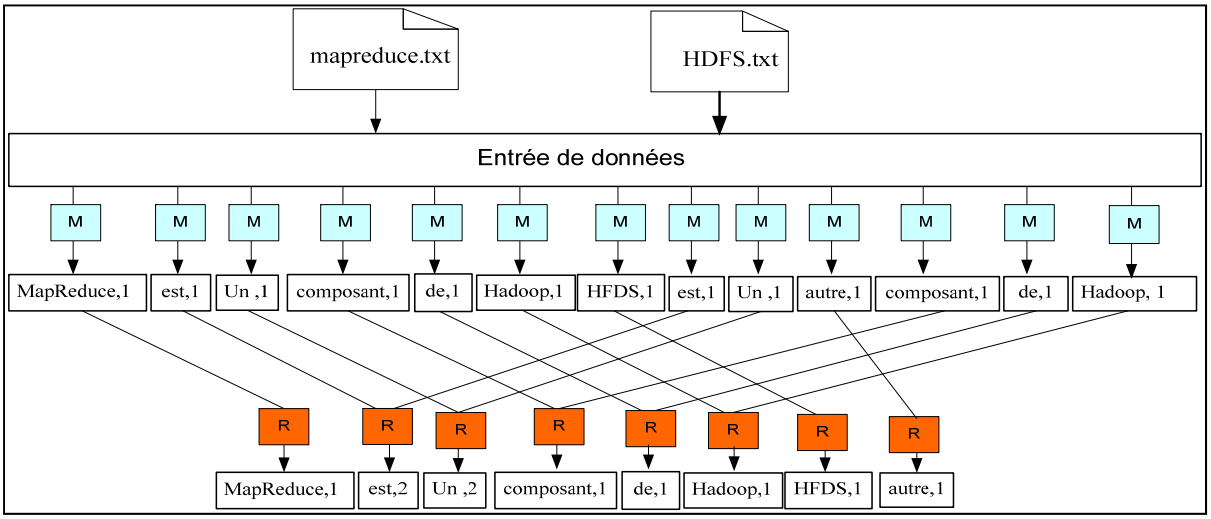

Figure 2.6 Exemples de MapReduce

La figure 2.6 décrit d'une manière très simplifiée le fonctionnement de MapReduce. Plusieurs instances de la fonction de mappage sont créées pour traiter les données en entrée et ainsi produire des couples (clés, valeurs), de même, plusieurs instances de la fonction de réduction sont aussi créées pour réduire la liste des couples en entrées en calculant les occurrences des mots dans les fichiers textes traités en entrées. Les instances des fonctions de mappage et de réduction sont créées et déployées sur plusieurs ordinateurs assurant ainsi un traitement parallèle.

Le principe du mappage et de réduction peut être appliqué à n'importe quel domaine, il suffit seulement d'adapter ces deux fonctions aux types des domaines d'applications et aux résultats que l'on voudrait obtenir.

CHAPITRE 3

CHOIX DE LOGICIEL SGPA LIBRE

3.1 Critères de choix

Tel que précisé dans les sections : *portée de la recherche* et *Objectif de la recherche*, l'objectif de cette étude est de modifier un logiciel libre de modélisation de processus d'affaires afin qu'il puisse fonctionner sur une plateforme d'informatique de nuage : Hadoop. Il existe un grand nombre de logiciels libres qui permettent de modéliser les processus d'affaires. Cependant, chacun a été développé pour répondre à des besoins bien précis.

Afin de faciliter l'intégration du logiciel libre choisi dans la plateforme Hadoop, il est nécessaire d'établir une liste de caractéristiques souhaitées. Le tableau 3.1 présente la liste des critères de sélection que nous allons utiliser pour choisir un logiciel pour notre étude.

Tableau 3.1 Critères de sélection de logiciels libres

Critère	Poids	Description
Un logiciel libre	1	Le logiciel choisi doit être un logiciel libre géré par un type de licence permettant de modifier le code source sans aucune restriction
Code source disponible	1	Il ne doit pas y avoir de restrictions sur l'accès au code source.
Riche en langages supportés de modélisation	2	Le logiciel doit permettre la modélisation de processus d'affaires avec des langages qui sont supportés par les solutions SGPA. Ceci permettra dans le future d'étendre cette étude à l'intégration d'autres modules qui composent les systèmes de gestion de processus d'affaires sur une plateforme d'informatique de nuage
Technologie Internet	3	La technologie utilisée dans le développement du logiciel doit être compatible avec les plateformes Internet pour assurer une meilleure intégration avec le logiciel d'informatique de nuage Hadoop.

3.2 Choix du logiciel

Le processus de sélection considère les logiciels libres de modélisation d'affaires les plus cités.

Tableau 3.2 Évaluation de logiciels libres par rapport aux critères de sélection établis

Critère / Logiciel	Logiciel Libre	Source disponible Disponibilité du code	Technologie Internet	Richesse en langages de modélisation supportés	Commentaire
Dia	+	+	-	-	✓ Dia est un logiciel de modélisation de processus d'affaires inspiré du logiciel de Microsoft Visio. ✓ Il supporte quelques langages de modélisation à savoir : EPS, WMF, XML mais pas les langages les plus utilisés comme BPMN et XPDL ✓ Il a été développé en langage C, un langage pas très flexible pour une infrastructure Internet
OpenOffice Draw	+	+	-	-	✓ OpenOffice Draw est un composant du logiciel libre open office, il permet la modélisation des flux de données ✓ Il a été développé en langage C++ ✓ Il ne supporte pas les langages de modélisation de processus d'affaires ce qui ne permet pas son intégration avec des solutions SGPA

Critère / Logiciel	Logiciel Libre	Disponibilité du code Source disponible	Technologie Internet	Richesse en langages de modélisation supportés	Commentaire
Kivio	+	+	-	-	✓ Kivio est un composant de la solution KOffice. Il permet de représenter sous forme graphique les flux de données, les diagrammes de structures organisationnelles, les réseaux, etc. Cependant, il ne supporte pas les langages de modélisation des processus d'affaires standards ce qui ne facilite pas son intégration avec les SGPA. ✓ Le langage Python est utilisé par KOffice pour représenter les graphiques en script. Même si Python est un langage qui permet le développement Internet, l'architecture de Kivio n'est pas adéquate pour une plateforme Internet
Intalio	+/-	-	+	-	✓ Intalio est un système de gestion de processus d'affaires (SGPA) complet. Il a été développé autour du logiciel de modélisation libre de droit Eclipse. ✓ Il supporte seulement la notation BPMN ✓ Le code source du logiciel est disponible en partie seulement pour les clients de Intalio

Critère / Logiciel	Logiciel Libre	Source disponible	Technologie Internet	Richesse en langages de modélisation supportés	Commentaire
jBPM	+	+	+	-	✓ jBPM est un logiciel libre de droit de gestion de processus d'affaire. Il est écrit en Java et supporte seulement la notation BPMN. Les modèles sont sauvegardés dans des fichiers JPDL
wFMOpen	+	+	+	-	✓ wFMOpen est un logiciel libre de droit de modélisation de processus d'affaire, écrit en Java. Il supporte seulement la notation XPDL.
Process Maker	+	+	+	-	✓ ProcessMaker est un logiciel libre de droit de gestion de processus d'affaire. Il comporte un modélisateur de processus BPMN 2.0, un module de configuration et d'exécution de processus modélisés, un module de gestion d'utilisateur, etc. ✓ Il est développé en PHP et Ajax et supporte l'architecture Orienté service.
uEngine	+	+	+	-	✓ uEngine est un logiciel libre de gestion de processus d'affaire. Il supporte seulement la notation BPMN. Developpé avec la technologie Java, Ajax et JEEE.
Bonita Soft	+	-	+	-	✓ BonitaSoft est un logiciel de gestion de processus d'affaire libre de droit, il supporte uniquement la notation BPMN. ✓ Le code source de la librairie de Bonita n'est pas accessible

Critère / Logiciel	Logiciel Libre	Source disponible — Disponibilité du code	Technologie Internet	Richesse en langages de modélisation supportés	Commentaire
Oryx	+	+	+	+	✓ Oryx est un logiciel libre de modélisation de processus d'affaires avec un accès complet sans restriction au code source ✓ Il supporte plusieurs langages de modélisation : BPMN, EPC, etc. ✓ Oryx possède une architecture Internet. Les modules du serveur sont écrits en java et les modules du client sont en javascript

La liste des logiciels libres cités dans le tableau ci-dessus n'est pas une liste exhaustive, il y a beaucoup d'autres logiciels qui sont disponibles, cependant, ils ne satisfont pas les critères de sélection établis dans le tableau 3.1.

Le tableau 3.2, permet de constater qu'Oryx est un logiciel approprié au contexte de la recherche.

Les sections suivantes décrivent l'architecture actuelle du logiciel Oryx et permettra l'investigation de la possibilité de l'adapter à Hadoop.

3.3 Description du logiciel Oryx

3.3.1 Introduction

Oryx est le résultat du projet académique « B7 - Browser-based Business Process Editor » initié par le département « Business Process Technology » de l'institut Hasso-Plattner à l'université de Postdam en Allemagne et supervisé par le professeur Mathias Weske, responsable du groupe de recherche des processus d'affaires (Martin A. Czuchra. 2007). L'objectif du projet était de développer une application Internet pour la modélisation des processus d'affaires qui permettraient de supporter plusieurs notations graphiques standards : BPMN, EPC, Réseaux Pétri, etc. C'est en 2007 qu'Oryx a été déployé et mis à la disposition des utilisateurs et des développeurs des logiciels libres.

Oryx possède deux points forts, le premier est sa capacité de supporter n'importe qu'elle langage de modélisation de processus d'affaires grâce au module de définition de langage (Stencil Set) et de son infrastructure de plugiciel qui lui permet d'être ouvert facilement à de nouvelles fonctionnalités, le deuxième est que les utilisateurs peuvent créer des modèles de processus d'affaires dans un fureteur et de les partager avec tous les autres utilisateurs.

3.3.2 Architecture d'Oryx

La figure suivante décrit l'architecture actuelle d'Oryx :

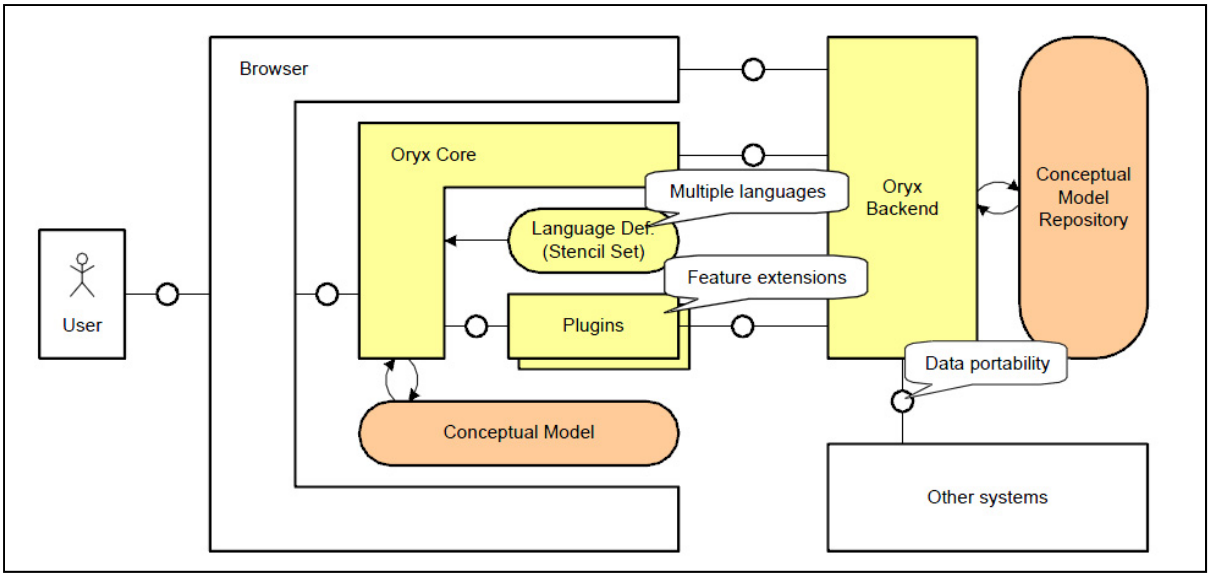

Figure 3.1 Architecture d'Oryx
Tirée de Decker, Overdick et al. (2008)

Oryx est une application Internet développée avec une architecture client-serveur. Il est constitué des composants suivants:

a-Éditeur d'Oryx: Une interface *client* disponible aux utilisateurs à partir de n'importe quel navigateur Internet pour dessiner leurs processus.

b-Module de définition des langages de modélisation de processus (*Stencil Set*):
Le module de définition des langages de modélisation (*Stencil Set*) permet à Oryx d'être capable de supporter tous les langages de modélisation de processus d'affaires. En effet, pour ajouter un nouveau langage de modélisation, il suffit de définir l'ensemble des symboles utilisés par le langage et les règles qui gèrent leur utilisation et qui les lient entre eux.

Oryx utilise trois technologies dans la définition de langages de modélisation de processus d'affaires :

- La notation JavaScript (*JSON*) pour décrire les langages de modélisation, leurs propriétés et les règles de manipulation des symboles qui forment les langages.
- *Scalable Vector Graphic (SVG)* pour décrire les représentations graphiques des symboles.
- *Javascript* pour décrire le comportement des objets qui composent les langages de modélisation. Cela permet de définir la partie dynamique des langages de modélisation.

La version actuelle d'Oryx contient la définition des langages de modélisation les plus utilisés par les SGPA à savoir: BPMN1.2, BPMN2.0, EPC, Réseaux Pétri, WF-Nets, etc.

c-Plugiciel : La majorité des fonctionnalités d'Oryx est dans des plugiciels que l'on peut ajouter séparément au navigateur selon le besoin. Par exemple, les fonctions d'édition, les fonctions de conversion de modèles BPMN en modèles réseau pétri, etc.

d-Système principal d'Oryx: Il forme le noyau d'Oryx et est implanté sur un serveur Internet capable d'exécuter le code java. Il est composé de l'implantation du code de l'éditeur, des composants de déploiement des modèles, de la gestion des accès, etc.

CHAPITRE 4

MIGRATION D'ORYX VERS HADOOP

4.1 Introduction

Oryx est une application Internet qui permet à l'utilisateur de créer et ou de modifier des processus d'affaires en utilisant différents formalismes de modélisation tels que BPMN, EPC, les réseaux de petri, diagramme de class UML, etc.

Dans ce qui suit, nous allons présenter le fonctionnement actuel d'oryx, les objectifs et les difficultés de sa migration vers la plateforme de Hadoop, une vision sur une nouvelle solution ainsi que les étapes suivies pour son développement, sa mise en œuvre et finalement son test.

4.2 Fonctionnement actuel d'Oryx

4.2.1 Identification des utilisateurs

Les utilisateurs accèdent à l'éditeur graphique d'Oryx à partir d'un fureteur Internet. Ils doivent d'abord s'identifier pour pouvoir sauvegarder et retrouver les processus modélisés. Actuellement, l'éditeur graphique d'Oryx requiert l'utilisation du fureteur Firefox pour pouvoir éditer les processus.

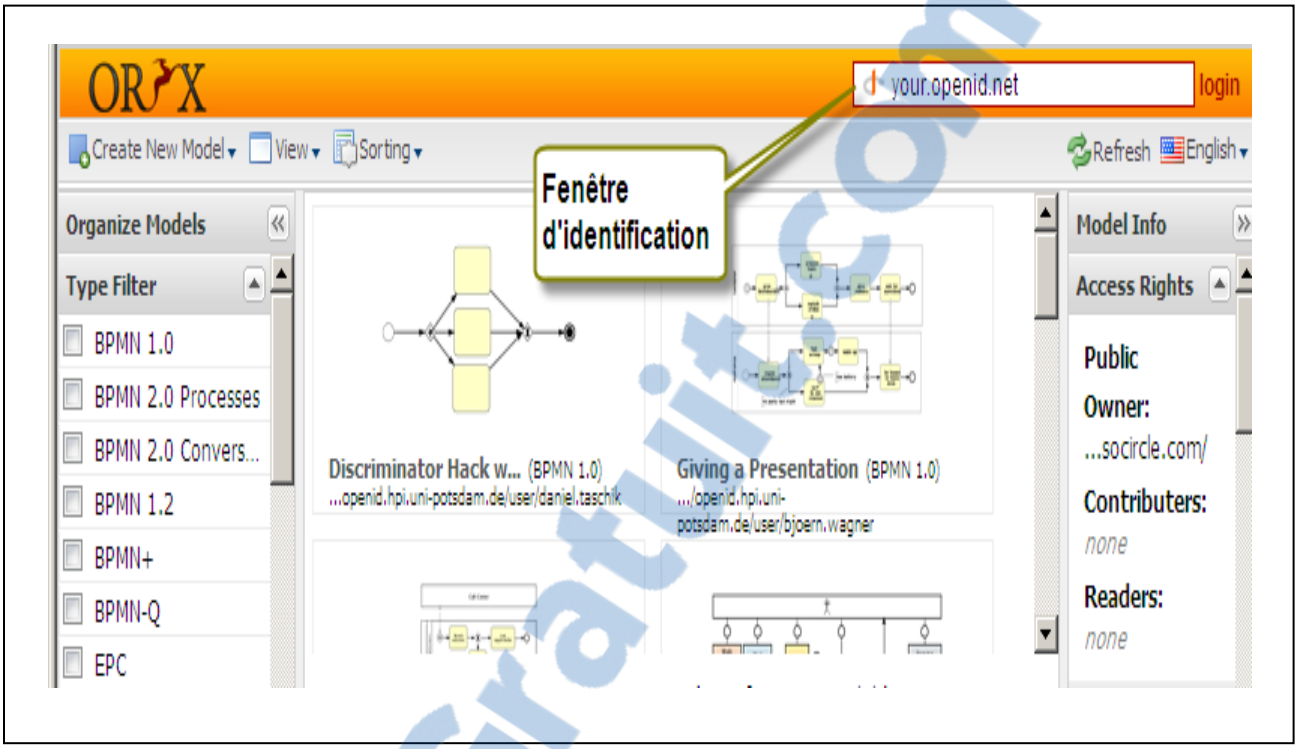

Figure 4.1 Interface utilisateur d'Oryx

Oryx utilise le système d'identification personnel OpenID pour identifier les utilisateurs. Tout nouvel utilisateur doit d'abord s'inscrire à un serveur OpenID (comme MyOpenID), puis utiliser son code d'accès enregistré pour se connecter au serveur d'Oryx.

4.2.2 Créer un nouveau processus d'affaires

Pour créer un nouveau processus, l'utilisateur choisi un formalisme parmi les différents formalismes de modélisation de processus disponibles, tels que, BPMN, EPC, Réseau Petri, XForms, Diagram de class UML, etc. Oryx affiche une palette de symboles graphiques correspondant au formalisme choisi. L'utilisateur modélise et sauvegarde son processus. Oryx sauvegarde les informations qui identifient les processus et le code **SVG** (*Scalable Vector Graphics)* dans une base de données relationnelle (PostgreSQL).

4.2.3 Modifier un modèle de processus

Le moteur d'Oryx utilise le code d'accès de l'utilisateur pour retrouver tous les modèles qu'il a enregistrés et afficher des liens URL pour chacun sur la page d'accueil de l'utilisateur. Il n'a qu'à cliquer sur le lien du modèle sur lequel il veut travailler pour le modifier dans l'éditeur graphique d'Oryx.

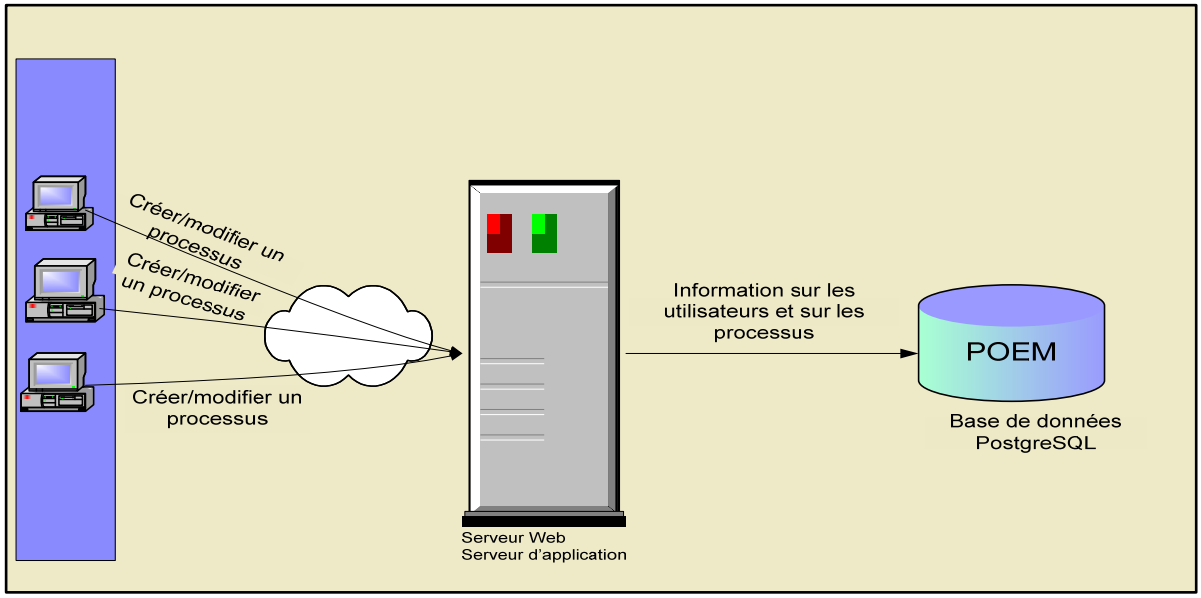

Figure 4.2 Diagramme d'architecture web actuel d'Oryx

4.3 Architecture actuelle d'Oryx

4.3.1 Structure du code d'Oryx

Oryx est une application client serveur. La majorité des fonctionnalités sont développées en langage Java Script. Seulement quelques fonctions du serveur sont développées en Java. Oryx utilise la base de données PostgreSQL pour enregistrer les données et la librairie « Hibernate » pour gérer les accès à la base de données.

4.3.2 Base de données POEM d'Oryx

Oryx utilise la base de données POEM, (une base de données relationnelle) pour sauvegarder puis retrouver les modèles de processus produits par les différents utilisateurs. Elle est gérée par le serveur de base de données PostgreSQL.

4.3.2.1 Vue statique du modèle de données de la base de données POEM

La rétro ingénieurie du modèle physique de la base de données « POEM » et du code d'Oryx nous a permis d'extraire le modèle logique suivant :

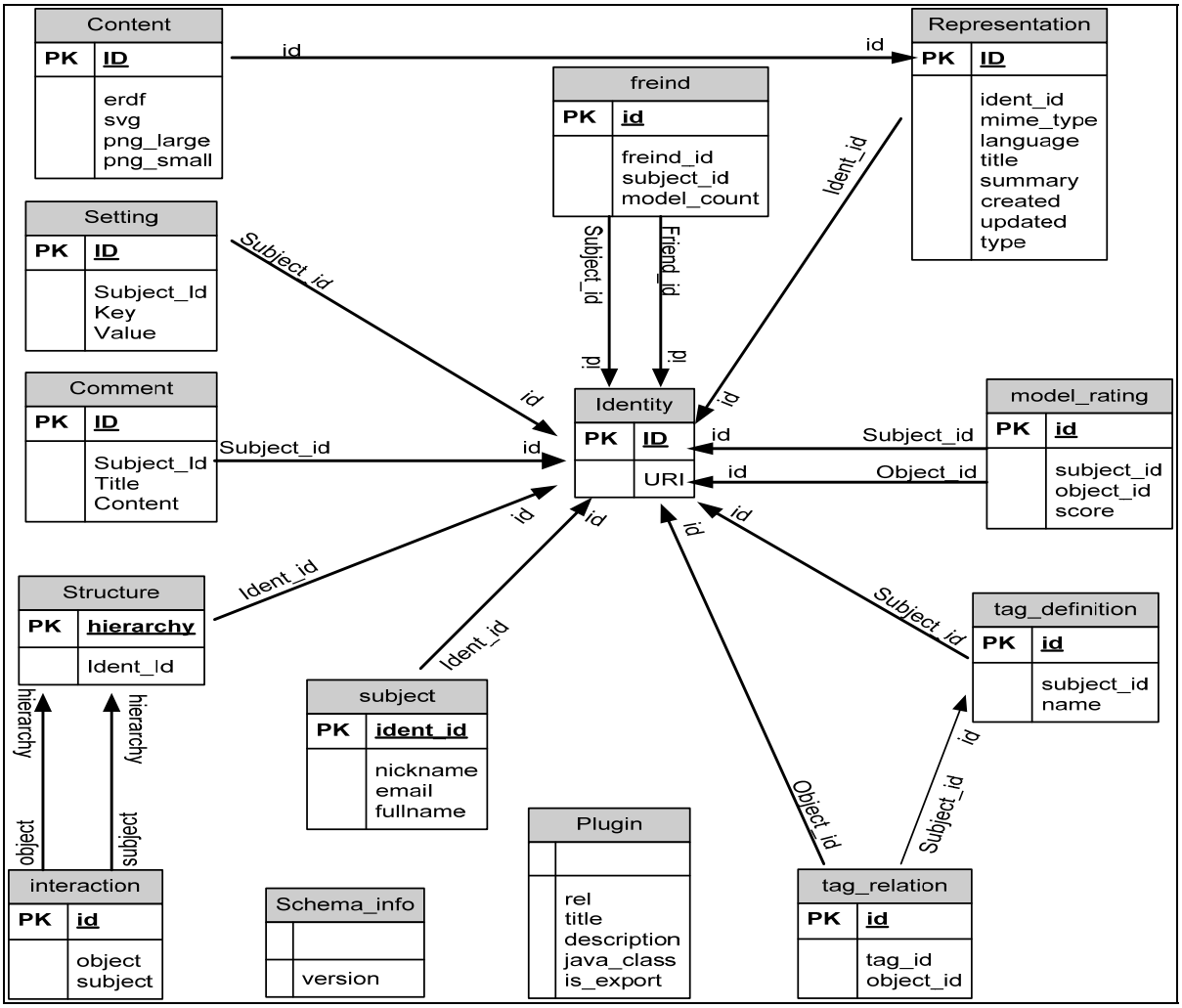

Figure 4.3 Modèle relationnel de la base de données POEM

4.3.2.2 Vue dynamique de la base de données POEM

Nous allons présenter dans le tableau suivant les fonctionnalités principales d'Oryx et la réaction des tables de la base de données lors de l'éxécution de chaque fonction :

Tableau 4.1 Description dynamique des tables de la base de données POEM

Fonction	**S'authentifier**	
Table	**Identity**	Enregistrer le lien complet de l'identificateur personnel OpenID de l'utilisateur. Cette entrée va permettre de retrouver tous les modèles de processus créés par ce dernier **Exemple:** ID= 20 URI: lien vers le « OPenID » de l'utilisateur
	Structure	Créer une entrée dans la table structure avec comme clé un code (hierarchy) qui va servir comme lien entre l'utilisateur et tous les processus qu'il va créer avec Oryx. Cette clé est jointe à celle avec laquelle l'utilisateur a été créé dans la table « identity » **Exemple:** ident_id=20 hierarchie: U21
	Subject	Créer une entrée avec toutes les informations de chaque nouvel utilisateur. Dans cette table, Oryx tient un compteur pour tracer le nombre de fois que chaque utilisateur se connecte à l'application **Exemple :** o ident_id=20 plusieurs champs contenant des infos sur l'usager (nom,email,code postal,..)
	Identity	créer une entrée pour identifier le nouveau modèle **Exemple :** o id= 21 URI= /model/20
	Structure	Créer une entrée pour le nouveau modèle créé **Exemple :**

		o ident_id=21 o hierarchy=U210

Fonction	**S'authentifier**	
Fonction	**Créer un nouveau modèle**	
	Représentation	Créer une entrée contenant des informations sur le nouveau modèle créé Exemple : - ident_id=21 - id=9 - titre du modèle - description du modèle - ….
	Content	créer une entrée dans cette table pour enregistrer le contenu du modèle créé. Ce contenu va permettre de l'afficher et de l'éditer dans le futur **Exemple** : id=9 erdf : propriété du modèle créé svg : code svg pour dessiner le modèle

4.4 Objectif de la nouvelle solution

Le but principal de cette recherche est de mettre à la disposition des organisations un outil de gestion de processus d'affaires complet et performant. Dans un premier temps, nous allons travailler sur un outil de modélisation de processus d'affaires, objectif de cette recherche (*section 1.5 Objectif de la recherche*), avec la perspective de le compléter dans le futur avec d'autres outils pour offrir un SGPA complet aux organisations.

Le logiciel que nous avons choisi, Oryx, est reconnu pour les différents types de formalismes qu'il est capable de supporter comme BPMN, EPC, réseau pétri, etc. et pour son extensibilité

à d'autres formalismes de modélisation. De plus, c'est une application Internet qui offre une flexibilité aux utilisateurs car elle leur permet d'accéder à partir de n'importe quel fureteur Internet. Cependant, il présente les faiblesses suivantes :

Oryx utilise la notation SVG (Scalable Vector Graphics) pour sauvegarder et afficher les modèles de processus d'affaires. Ces fichiers sont sauvegardés dans la base de données POEM dans des champs de type Texte. La performance d'Oryx dépend du volume de données sauvegardées dans la base de données. Plus la taille des tables est grande, plus la nécessité d'investir dans des ressources machines additionnelles pour améliorer la performance est indispensable. Étant donné qu'Oryx est une application Internet pour la modélisation de processus d'affaires, son objectif est d'être ouvert à toute la communauté Internet intéressée par la modélisation de processus d'affaires. De ce fait, il sera rapidement confronté à la problématique de la taille grandissante de la base de données POEM et par conséquent de la performance de serveurs de base de données et d'applications.

Un système de gestion de processus d'affaires est généralement composé de plusieurs modules, dont l'éditeur graphique, un module de simulation, de configuration et d'exécution des processus. Dans une perspective de rendre Oryx une solution SGPA complète, l'utilisation de la base de données POEM sur PostgreSQL risque de poser des problèmes de performance et de disponibilité du système pour de grands volumes de données.

Dans l'objectif de remédier aux faiblesses d'Oryx citées ci-dessus, nous allons tenter de l'intégrer à une plateforme d'informatique de nuage et de le faire profiter de ses avantages. En effet, Hadoop est un logiciel libre d'informatique de nuage développé par la fondation Apache dans le but de gérer et d'analyser de grands volumes de données.

4.5 Architecture de la nouvelle solution

La nouvelle solution doit permettre à Oryx de traiter de grands volumes de données sans avoir de contraintes sur leur nature ni sur leur structure. La façon qu'Oryx enregistre actuellement les données, dans la base de données relationnelle POEM est, à notre opinion, le point principal à améliorer.

Nous avons deux pistes de solutions à explorer :

- Utiliser la base de données HBase de Hadoop : une base de données « open source » adaptée aux plateformes d'informatique de nuage inspirées de la base de données BigTable (Fay, Dean et al. 2008) de Google ayant la capacité de gérer de grands volumes de données non structurées .

- HadoopDB : Une technologie hybride de MapReduce et les SGBD(Abouzied, Bajda-Pawlikowski et al. 2010) développés essentiellement pour les applications de gestions analytiques des données. Il lie plusieurs bases de données relationnelles indépendantes installées sur une grappe d'ordinateurs qui utilise les fonctionnalités de communication et de gestion de tâches de Hadoop.

- HadoopDB ne supporte pas les opérations de mise à jour unitaire, il utilise un module de chargement de donnés en lots. Les utilisateurs d'Oryx ont besoins de créer, modifier, et charger leur modèle de processus en temps réel. Nous avons donc abandonné la piste de HadoopDB dans notre Project.

Dans la suite de cette recherche nous allons explorer comment HBase pourrait contribuer à renforcer l'architecture actuelle d'ORyx pour faire face à de grands volumes de données.

4.5.1 Architecture de HBase

Hbase est une base de données distribuée non relationnelle développée par l'équipe projet de développement de Hadoop. S'appuyant sur le système de fichier distribué de Hadoop, elle permet de structurer la sauvegarde des données sous forme de tables à plusieurs lignes, chacune d'elles est identifiée par un libellé et peut avoir un nombre illimité de colonnes.

4.5.2 Modèle de données de HBase

Hbase utilise un modèle de données similaire à celui de « BigTable» de Google. (Fay, Dean et al. 2008) définissant « BigTable » comme étant «une carte clairsemée, distribuée, persistante, multidimensionnelle, triée et orientée colonne ». Chaque entrée dans la carte est un tableau complexe ayant la structure suivante : (Lignes, colonnes, date et temps).

Le modèle est dit :

- Clairsemé car les lignes peuvent avoir un nombre illimité de colonnes dans chaque famille de colonnes. Si pour une colonne il n'y a pas de données, Hbase / BigTable laissent des vides dans la carte.

- Distribué car il s'appuie sur le système de fichier distribué de Hadoop, ce qui fait que les données peuvent être physiquement enregistrées sur plusieurs machines.

- Persistant car les données restent dans la carte même après la fin des programmes qui les ont créée ou accédée, il a les mêmes propriétés que celles des bases de données relationnelles.

- Multidimensionnelle car chaque ligne dans la carte peut avoir un nombre illimité de familles de colonnes, et chaque famille de colonne peut avoir un nombre illimité de colonnes identifiées par un identifiant, la date et le temps de sa création.

- Trié car la carte est triée en premier lieu par les identifiants des lignes et puis par ceux des familles de colonnes et des colonnes.

- Orienté colonne car le modèle de données est essentiellement basé sur les familles de colonnes et les colonnes, à l'opposé des bases de données relationnelles qui utilisent les entités/relation comme base pour structurer les données.

La structure générale d'une table Hbase peut être représentée comme suit :

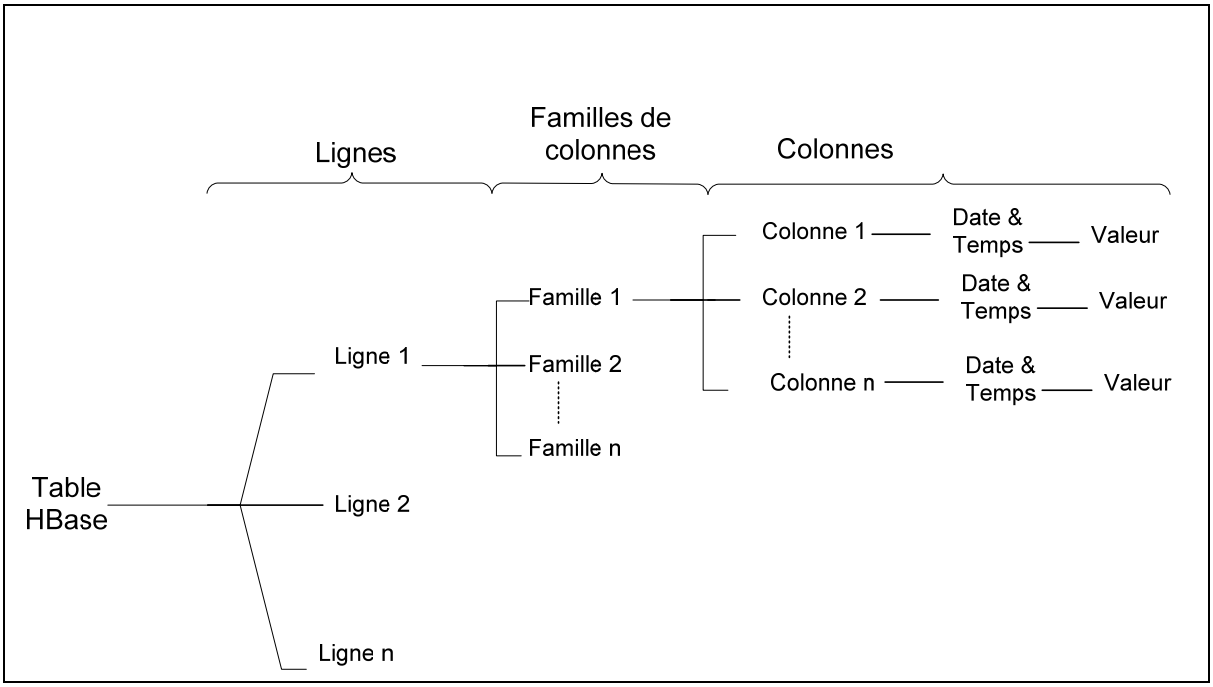

Figure 4.4 Modèle conceptuel de données de Hbase

La notion de familles de colonnes a été introduite par BigTable. Les données sauvegardées dans la même famille de colonnes sont enregistrées ensemble dans le système de fichier, alors que les autres familles de colonnes peuvent être distribuées entre plusieurs machines.

4.6 Vue globale sur la nouvelle architecture d'Oryx avec HBase

L'objectif principal de la transformation d'Oryx vers Hbase est de résoudre le problème du rapport entre la croissance du volume de données sauvegardées dans la base de données PostgreSQL avec la performance du serveur de base de données.

4.6.1 Solutions possibles

À notre avis nous avons deux options :

1. **Transformation globale :** Elle consiste à transformer la base de données PostgreSQL au complet vers la base de données HBase, migrer toutes les données vers la nouvelle version puis modifier le code d'Oryx pour éliminer tous les appels vers la base de données POEM sur Oryx et les remplacer par des appels à la version non-relationnelle de la base de données POEM sur Hbase.

2. **Transformation hybride :** Elle consiste à analyser la base de données relationnelle POEM afin de déterminer les sous schémas susceptibles de recevoir le poids de la croissance de la base de données et risque d'avoir un impact négatif sur la performance du serveur de la base de données et sur celui de l'application. Elle extrait ces sous schémas puis les remplace par d'autres sur la base de données non-relationnelles Hbase. Les données enregistrées sont migrées dans ces sous schémas puis le code d'Oryx est modifié afin de remplacer les appels au sous schéma transformé.

À notre avis, il est plus judicieux d'aller avec la transformation hybride pour les raisons suivantes :

1- La base de données POEM utilise beaucoup de fonctionnalités de la base de données relationnelles tel que les index, les séquences, les fonctions, les déclencheurs, etc.

Transformer toute la base de données vers Hbase engendrerait des modifications majeures dans le code d'Oryx pour remplacer les fonctionnalités non supportées par HBase de PostgreSQL, ce qui risque d'avoir un impact négatif sur la stabilité et la robustesse de l'application.

2- Le modèle de données POEM est un modèle normalisé. Transformer toute la base de données vers HBase nécessiterait la dé-normalisation du modèle : Ceci impliquera la gestion de la duplication superflue des données qui risquent d'en résulter, la perte de l'avantage des chemins de recherche rapides dans le SGBD et la gestion de l'intégrité des données assurée par le SGBD.

3- Le modèle de données d'Oryx sur HBase est composé de vingt-huit (28) tables. Certaines d'entre elles seulement auront à recevoir un grand volume de données et risquent de créer des problèmes de performance aux serveurs de base de données et par conséquent affecter la performance du système Oryx. Il est plus judicieux donc de migrer seulement ce groupe de table.

4- La solution hybride nous permettrait de profiter des avantages des deux systèmes de base de données d'un côté, et de réduire les changements dans le code d'Oryx d'un autre côté ce qui assurera une meilleure stabilité du code.

4.6.2 Cadre de travail de la nouvelle solution

Le cadre de travail de la nouvelle solution est composé de quatre composants: L'application Oryx, La base de données relationnelle POEM sur PostgreSQL, La base de données HBase et la plateforme de Hadoop.

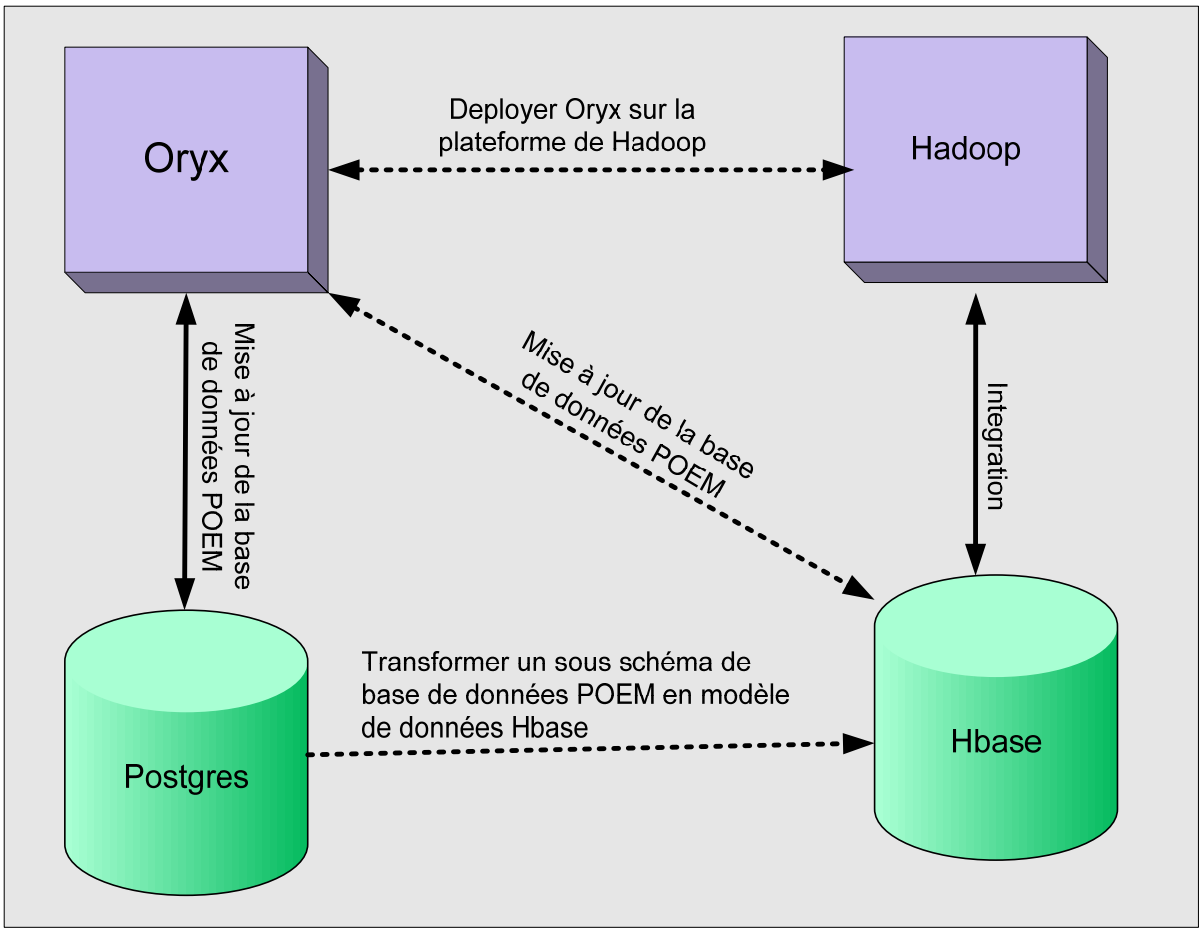

Figure 4.5 Cadre de travail de la nouvelle solution

Dans le schéma ci-dessus, les lignes en pointillés entre les composants du cadre de travail représentent les nouvelles fonctionnalités à développer, alors que les liens représentés par des lignes continues sont des modules existants mais qui doivent subir des changements.

4.6.3 Plan de migration vers de la nouvelle version d'Oryx 'Cloud'

Les 5 étapes à suivre pour la migration sont les suivantes:

1- Analyser la base de données POEM pour déterminer quel sous-schéma devrais migrer vers HBase et expliquer ces décisions;

2- Implanter le modèle de données du nouveau sous schéma sur HBase.

3- Installer et configurer l'environnement de migration.
4- Analyser le code source d'Oryx pour déterminer les modifications nécessaires afin de remplacer les appels à la base de données PostgreSQL par des appels à la base de données Hbase et faire les conversions de types de données nécessaires.
5- Déployer et tester la nouvelle solution.

La prochaine section décrit les travaux effectués pour ces 5 étapes.

4.7 Migration vers la nouvelle version d'Oryx 'Cloud'

4.7.1 Analyse de la base de données relationnelle POEM

Oryx est un éditeur graphique utilisé pour créer des modèles de processus d'affaires. Les données stockées dans la base de données sont les identificateurs personnels des utilisateurs qui servent à associer à chacun les modèles qu'il a créé, les formalismes de modélisation de processus tels que BPMN, EPC, réseau Petri, UML, etc, et les modèles de processus d'affaires créés par les utilisateurs.

Dans le cas d'utilisation à grande échelle d'Oryx par les utilisateurs, les données qui représentent les modèles de processus d'affaires modélisés composeront la majeure partie des données de la base de données.

En effet, pour chaque modèle créé, Oryx enregistre les informations qui permettent de l'identifier telles que la date de création, le titre, la description, etc. Le modèle de processus créé est converti en mode SVG (Scalable Vector Graphics) puis enregistré dans la base de données. La page HTML est enregistrée dans la base de données en format eRDF.

Nous avons construit un modèle de processus très simple avec Oryx et nous avons vérifié la taille du fichier SVG sauvegardé dans la base de données. Nous avons trouvé la taille égale à 150KB. Pour un million d'utilisateurs, avec une moyenne de dix processus par personne, la

base de données aura à emmagasiner plus de 1431 GB (1.39 TB) de données pour seulement la description de modèle dans la base de données. Ce volume de données n'inclus pas les autres informations emmagasinées pour chaque modèle créé.

Tableau 4.2 Prévision de la taille du fichier SVG de processus modélisé

Nombre d'utilisateurs	moyenne Processus/usager	Taille moyenne d'un fichier SVG (KB)	MB	GB	TB
1000	10	150	1470	1.431	0.001397
1000000	10	150	1470000	1431	1.39746
10000000	10	150	14700000	14310	13.9746

Le modèle de données suivant montre le sous schéma utilisé par Oryx pour emmagasiner les informations des modèles de processus.

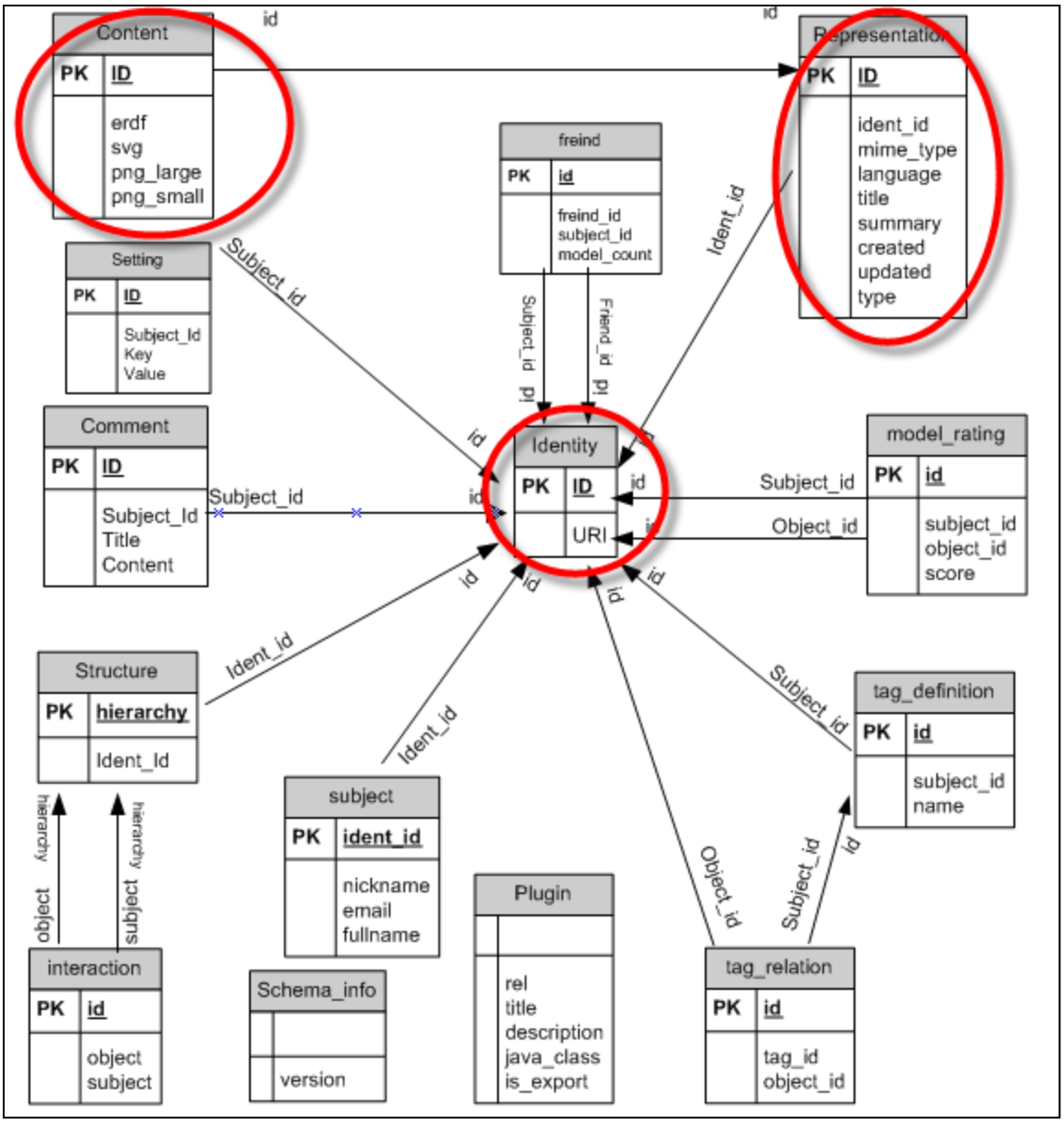

Figure 4.6 Sous-schéma de base de données qui sert pour emmagasiner les données des modèles

ORyx utilise la table « Identity » pour enregistrer les identificateurs des utilisateurs pour pouvoir retrouver tous les processus modélisés par chaque utilisateur. Il utilise la table « Representation » pour emmagasiner les informations nécessaires pour identifier les modèles de processus créés. Ces données ne sont pas de grandes tailles alors que la table

« Content » sert à emmagasiner les informations nécessaires pour recréer le modèle : la description SVG du modèle et la description eRDF de la page Internet.

Suite à ces observations, nous recommandons de migrer le sous-schéma composé des deux tables « Content » et « Representation » vers la base de données HBase qui est plus adaptée aux grands volumes de données.

4.8 Modèle de données HBase de POEM

4.8.1 Sous-schéma de base de données à migrer

Le diagramme ci-dessous décrit le sous schéma de base de données utilisé par ORyx pour enregistrer les données qui permettent de retrouver et de recréer les modèles de processus d'affaires modélisés par les utilisateurs. Il est composé des tables suivantes : « Identity », « Content » et « Representation ».

Chaque utilisateur peut créer une multitude de modèles de processus. La relation entre la table « Identity » et la table « Representation » est de type un à plusieurs, alors que celle de « Representation » et « Content » est de type « un à un » parce que chaque modèle créé possède un et un seul identifiant.

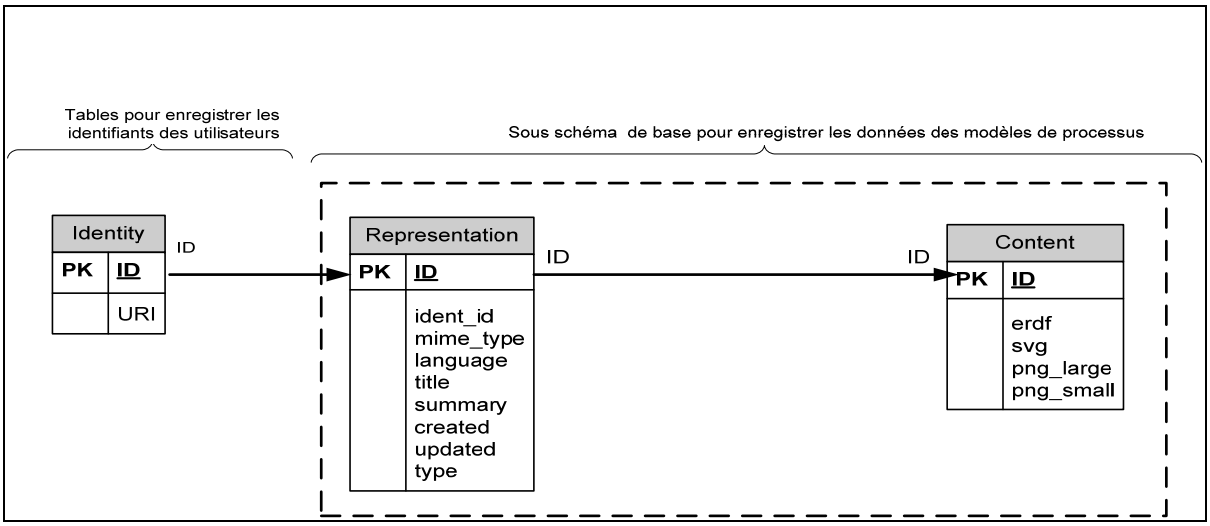

Figure 4.7 Sous-schéma de base à migrer de PostgreSQL à HBase

4.8.2 Transformation du sous-schéma de base de données POEM à Hbase

Afin d'assurer une meilleure performance de la nouvelle solution, nous allons viser les objectifs suivants pour concevoir la version HBase de sous schéma relationnel de POEM :

- Regrouper les données inter reliées ensemble
- Pouvoir retrouver les données facilement et en tout temps
- Minimiser les accès à la base de données pour retrouver les informations recherchées

La base de données POEM à été conçue de manière à ce que toutes les données inter reliées soient regroupées ensemble dans la même table, et que l'accessibilité aux données soit assurée par l'utilisation des clés étrangères. De ce fait, les tables liées dans le modèle relationnel avec la relation « un à un » seront transformées en familles de colonnes, et dans chacune on ajoutera la clé étrangère des autres familles que l'on voudrait référencer. Les tables liées avec des relations « un à plusieurs » seront transformées en tables dans HBase. Chaque clé primaire sera transformée en une ligne.

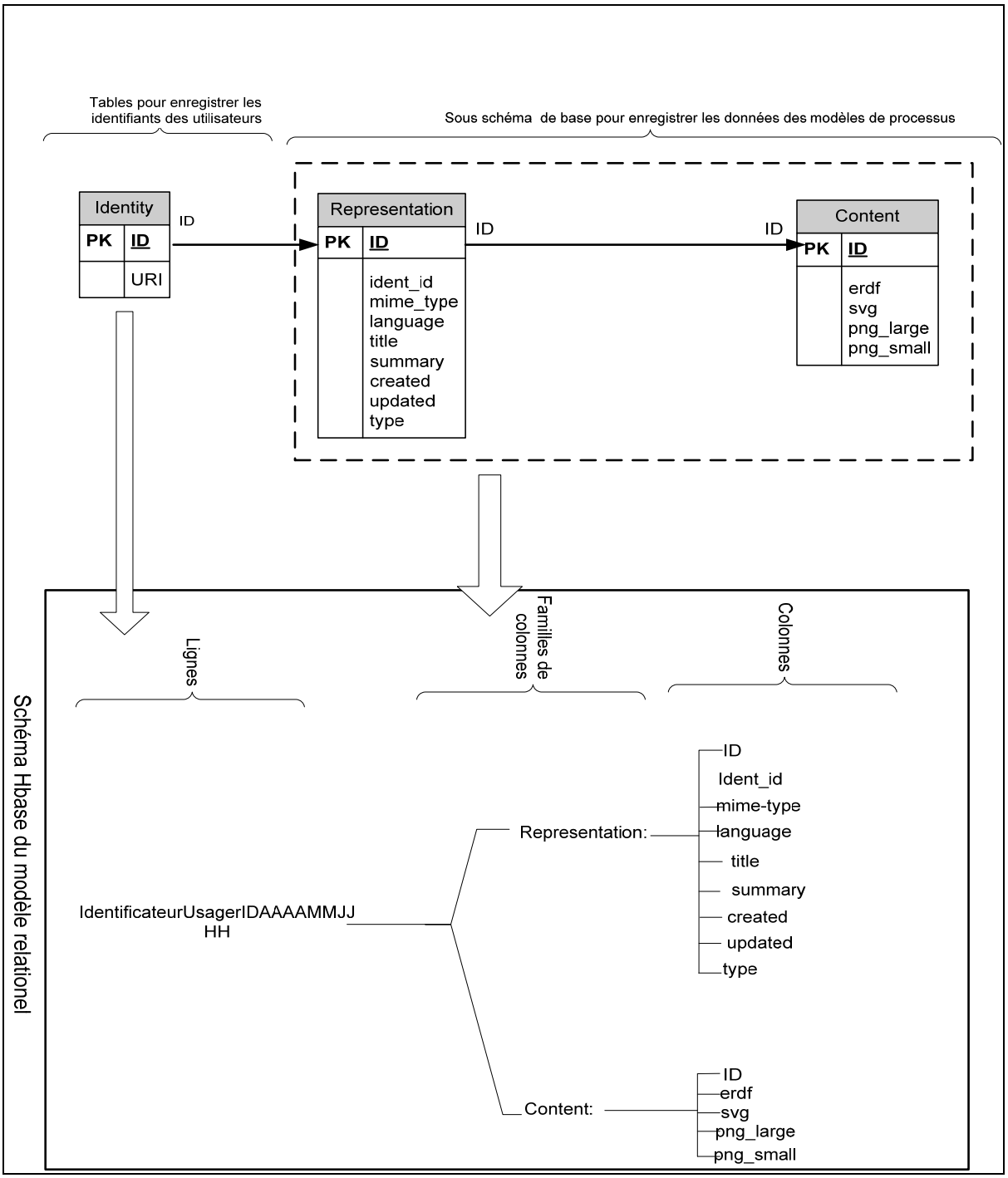

Figure 4.8 Modèle HBase équivalent du sous schéma relationnel POEM

Les deux tables « Representation » et « Content » sont liées par une relation « un à un ». En fait, elles identifient la même entité relationnelle : le modèle de processus d'affaires ainsi créé par l'utilisateur.

Le schéma de base Hbase de notre sous schéma de base relationnelle à convertir sera donc le suivant :

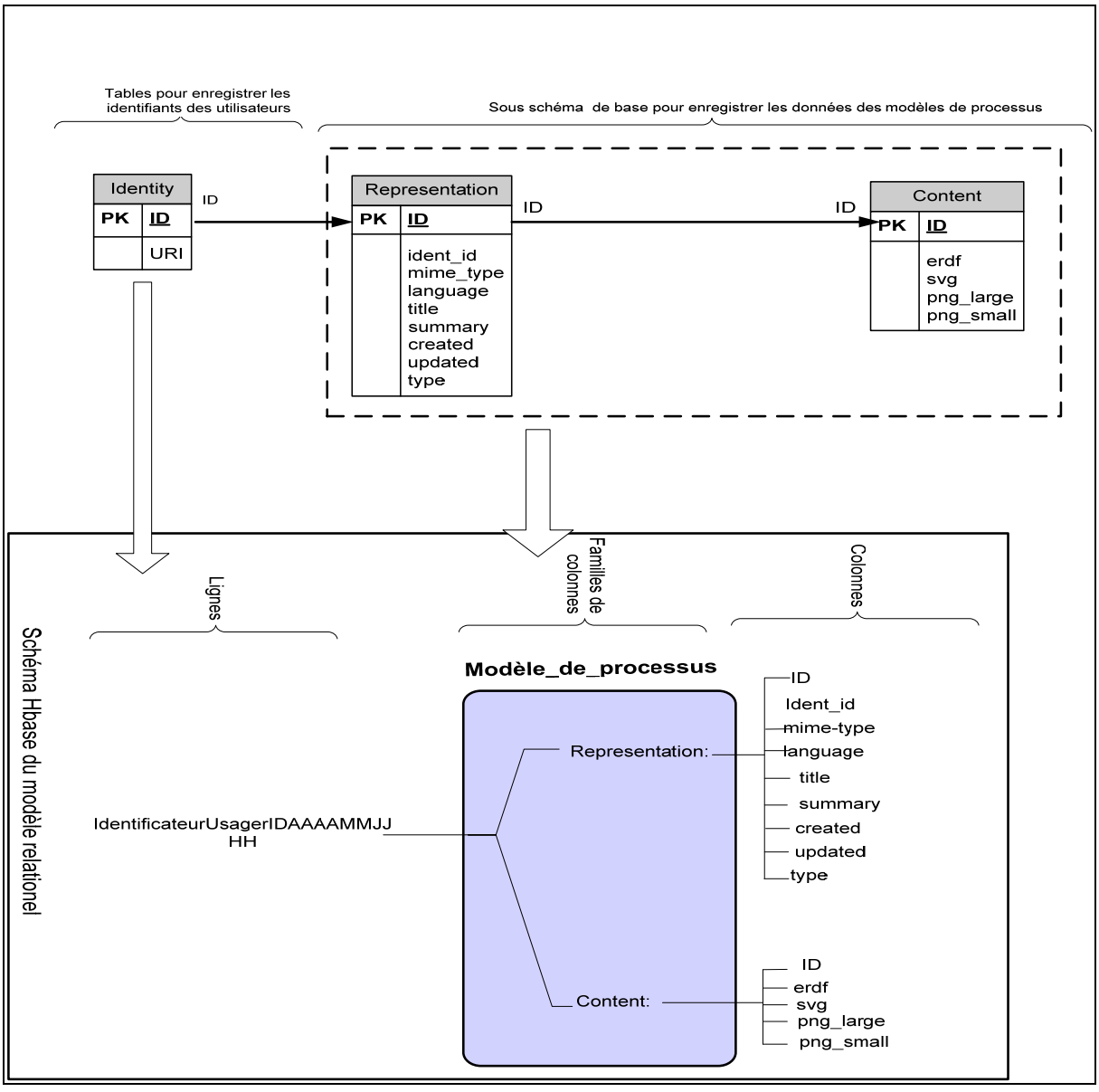

Figure 4.9 Modèle conceptuel Hbase de sous schéma relationnel POEM

Tableau 4.3 Modèle conceptuel Hbase de sous schéma relationnel POEM

Lignes	Famille de colonne	Colonnes
IdentificateurUsagerID	Modèle_de_processus 1	Ident_id mime_type language title summary created updated type erdf svg png_large png_small
	…….	
	Modèle_de_processus n	Ident_id mime_type language title summary created updated type erdf svg png_large png_small

4.9 Intégration de la base de données Hybride dans le code d'Oryx

Oryx est une application client serveur, la partie « client » est développée en JavaScript alors que le serveur est écrit en Java. Le serveur utilise la librairie « Hibernate » pour la gestion des accès à la base de données relationnelle PostgreSQL.

Hadoop met à la disposition des utilisateurs trois outils pour la gestion des données: Hive, Pig, et une bibliothèque de fonctions Java. Les deux premiers sont plus adaptés au mode de traitement de données par lots, c'est pourquoi nous allons utiliser les fonctions de la bibliothèque Java de HBase pour la gestion des accès des données entre Oryx et HBase.

La gestion des données dans Oryx est faite dans la partie serveur. Notre intervention serait seulement au niveau du serveur d'Oryx.

Dans ce qui suit nous allons présenter le mécanisme d'intégration de HBase dans l'application Oryx actuelle :

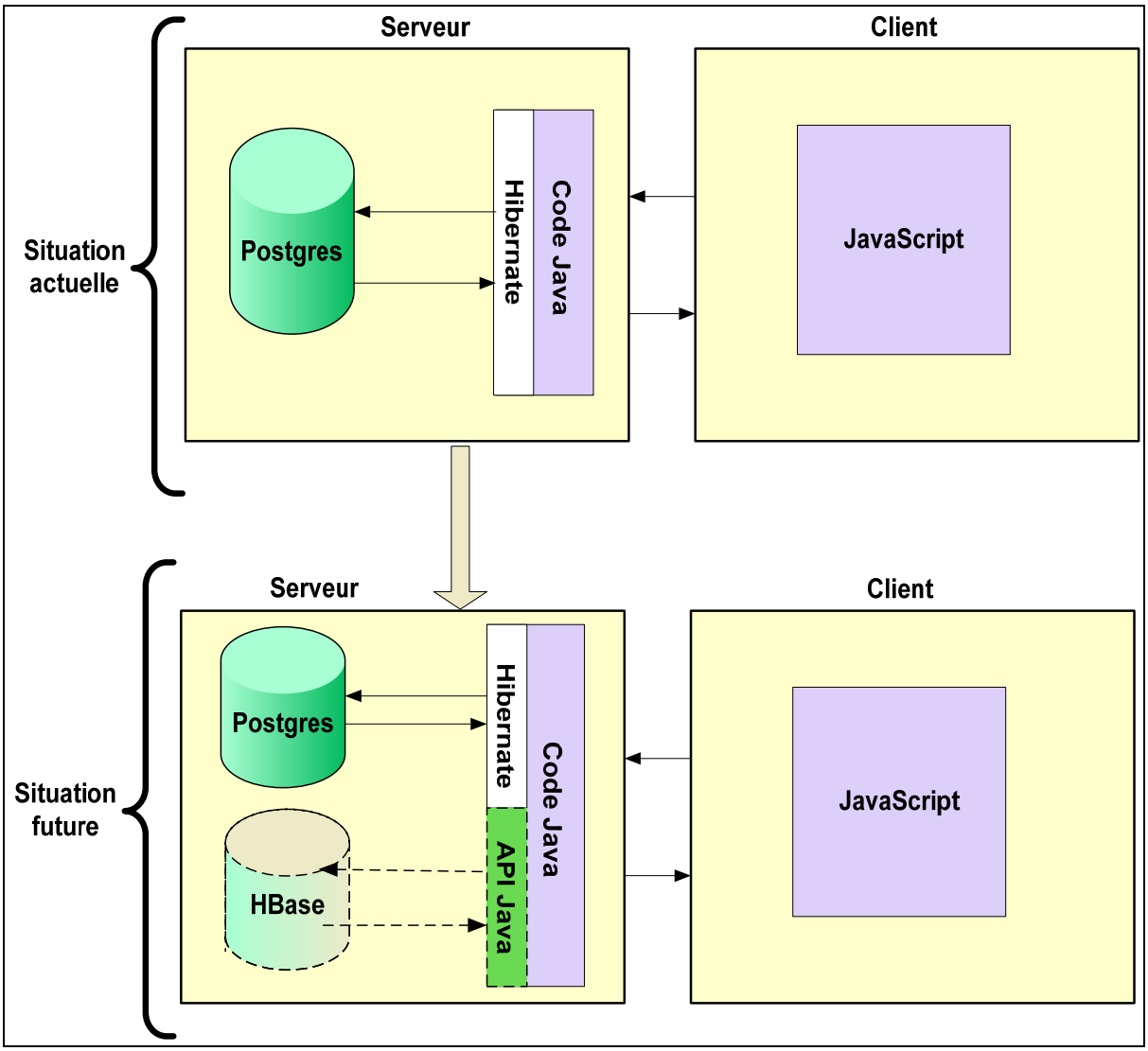

Figure 4.10 Schéma d'intégration de HBase dans l'architecture d'Oryx

Dans la solution que nous proposons, une couche de fonction de gestion de données sera ajoutée à celle de Hibernate au niveau du code d'Oryx afin de gérer les données des modèles de processus dans le sous schéma de base crée dans la base de données Hbase.

Le tableau suivant liste la logique que nous comptons implanter dans Oryx :

Tableau 4.4 Liste des changements à apporter sur Oryx

#	Action	Logique	Nouvelle fonction
1	**Créer un nouveau processus**	a) Un identificateur pour le nouveau processus est créé dans la table "Identity" associé à l'URL de l'utilisateur	
		b) S'il s'agit d'un nouvel utilisateur, une table est créée dans HBase pour l'utilisateur avec son URL comme identificateur	☑
		c) Créer une entrée dans la famille de colonne "Processus". L'entrée sera le nom du processus concaténé avec l'identifiant du processus créé dans la table "Identity"	☑
		d) Saisir les informations associées au processus à savoir le titre, la description, la date de création, le type du langage utilisé pour la modélisation, la représentation du modèle de processus en métalangage eRDF et SVG, etc.	☑
2	**Retrouver un processus**	a) Utiliser l'URL de l'utilisateur pour localiser tous les processus d'affaire créé par lui dans la table "Identity"	
		b) Charger tous les URL des processus créés par l'utilisateur sur la page internet de l'utilisateur	
		c) L'utilisateur sélectionne le processus dont il veut charger. Oryx extrait l'identifiant du processus sélectionné de la table "Identity"	

#	Action	Logique	Nouvelle fonction
		d) Utiliser l'URL de l'utilisateur pour localiser sa table HBase	☑
		e) Utiliser l'identifiant du processus sélectionné pour localiser la famille de colonne du processus en question et charger les informations stockées dans les colonnes	☑
		f) Utiliser les informations chargées du processus pour afficher le modèle sur la page internet	☑
3	**Modifier un processus**	a) Retrouver le Processus	
		b) Une fois les modifications faites sur le modèle, l'utilisateur appuie sur le bouton "Sauvegarder"	
		c) Localiser la famille de colonne de processus dans la table HBase de l'utilisateur	☑
		d) Remplacer les informations des colonnes (eRDF, SVG, date de modification, etc) avec les nouvelles informations	☑
		e) Remplacer les informations des colonnes (eRDF, SVG, date de modification, etc.) avec les nouvelles informations	☑
4	**Supprimer un processus**	a) Trouver un processus	
		b) supprimer la famille de colonne du processus sélectionné	☑

4.10 Architecture du code d'Oryx

Dans le but d'identifier les endroits qui peuvent abriter les changements nécessaires pour intégrer la base de données Hbase avec celle de PostgresSQL, nous allons documenter la structure du code source d'Oryx.

Dans un premier temps, Nous avons communiqué avec l'équipe qui travaille sur le projet d'Oryx à l'institut Hasso-Plattner à l'université de Postdam en Allemagne pour essayer d'obtenir la documentation du code source d'Oryx. Malheureusement, nous avons été informés qu'aucune documentation n'avait été produite en ce sens. Dans cette perspective, nous avons concentré nos efforts dans l'identification des points les plus appropriés dans le code pour contenir les changements nécessaires à l'intégration de Hbase dans le système.

La structure du code source d'Oryx consiste en trois parties principales :

- **Gestion de la configuration :** Cette partie contient des fichiers qui permettent la compilation des différents modules du système, des fichiers pour la gestion de l'environnement de développement et des fichiers XML utilisés par la bibliothèque « Hibernate » pour mapper le modèle relationnel de la base de données POEM avec le modèle orienté objet créé pour la base de données dans le code source d'Oryx.

- **Gestion de l'éditeur graphique :** Les programmes nécessaires à la spécification des modules de définition de langage (Stencil Set) et à la représentation des processus d'affaires dans l'interface graphique d'Oryx. Le code de cette partie est essentiellement développé en Javascript.

- **Gestion des données:** C'est la partie qui concerne le plus notre étude. Elle contient les programmes nécessaires à la gestion des données dans la base de données POEM d'Oryx.

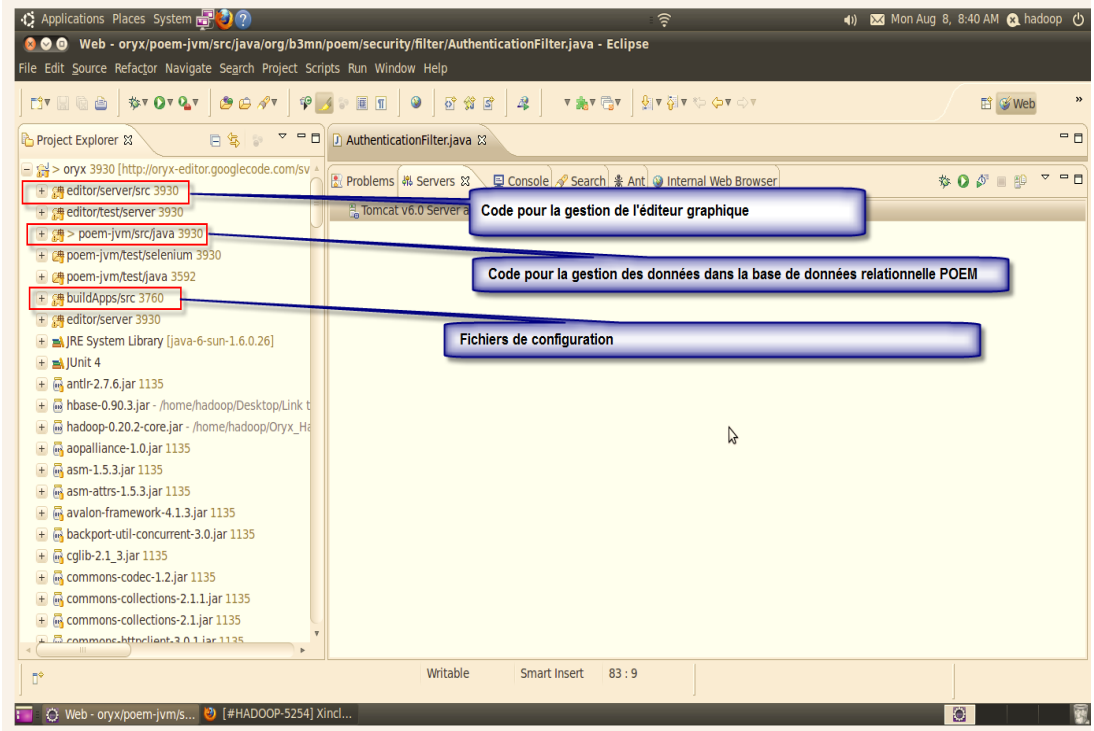

Figure 4.11 Structure du code source d'Oryx

Après analyse du code source d'Oryx, nous avons identifié quelques modules qui pourraient contenir le nouveau code qui va servir à intégrer la base de données Hbase avec celle de PostgresSQL à savoir : ConfigurationManager.Java, Identity.Java, Subject.Java, Model.Java, User.Java, Representation.Java.

Cette liste n'est citée qu'à titre d'exemple. Une étude plus approfondie du code source pourrait conduire à d'autres emplacements pour le nouveau code d'intégration de Hbase.

4.11 Réalisation du prototype

4.11.1 Configuration de l'environnement de développement

Pour la réalisation d'un prototype nous avons choisi de travailler avec la version 10.04.3 de Ubunto : un système d'exploitation Linux, pour la stabilité de Hadoop sur cette plateforme.

L'environnement de développement d'Oryx nécessite l'installation et la configuration d'une version récente de Java (au moins la version 6) et d'un serveur Internet capable d'exécuter des pages JSP. Pour la réalisation du prototype nous avons choisi le serveur Internet Tomcat-6, la version 2.5.2 de Python, une version 8.3 ou une version plus récente du serveur de base de données PostgreSQL qui va héberger la base de donnée POEM de Oryx et un environnement de développement intégré pour Java. Nous avons opté pour Eclipse pour le développement de notre prototype. Une fois l'environnement de développement d'Oryx installé et configuré, nous avons chargé la dernière version du code source d'Oryx.

Les annexes III et IV contiennent en détail les étapes de configuration de l'environnement de développement d'Oryx.

Pour la réalisation du prototype nous avons opté pour la dernière version stable de Hadoop et de HBase, soit la version 0.20.2 de Hadoop et la version 0.90.3 de HBase. Les annexes III et V détaillent tous les prés requis et les étapes à suivre pour l'installation et la configuration de Hadoop et de HBase sur Ubunto.

4.11.2 Module de maintenance du sous-schéma POEM sur HBase

Dans le tableau 4.4 nous avons énuméré les changements nécessaires à apporter sur Oryx pour déployer la nouvelle architecture, soit le partage du storage des données entre la base de données relationnelle POEM sur PostgreSQL et le sous-schéma de POEM sur Hbase.

Pour la réalisation du prototype nous nous sommes limités à la fonction de création d'un nouveau processus pour un nouvel utilisateur. Pour cela nous avons besoin des fonctionnalités suivantes :

- Création d'une nouvelle table dans HBase pour chaque nouvel utilisateur
- Création d'une entrée dans la famille de colonnes pour chaque nouveau processus

- Retrouver les informations d'un processus nécessaires pour l'affichage du modèle sur l'interface graphique de Oryx

Nous avons regroupé ces fonctions dans un seul package que nous allons intégrer dans le code d'Oryx :

```
hbase_hadoop {
  Hbase_hadoop_methods{
   creerUtilisateur { Une méthode pour créer une nouvelle entrée dans
      Hbase pour chaque nouvel utilisateur
    }
   creerProcessus{Une méthode pour créer une entrée dans Hbase pour
chaque
     nouveau processus créé par un utilisateur donné
     }
   trouverProcessus{ Une méthode pour retrouver et retourner les
données
      d'un processus dans la base Hbase
  }
}
```

Dans la section 4.8.2 nous avons le sous-schéma de la base de données POEM à déployer sur Hbase, soit :

```
Utilisateur : Identificateur de lignes (IdentificateurUsagerID) {
        Famille_de_colonne(Nom_modèle_de_processus){
            Colonne(Ident_id): Valeur
            Colonne(mime_type) : Valeur
            Colonne(language) : Valeur
            Colonne(title) : Valeur
            Colonne(summary) : Valeur
            Colonne(created) : Valeur
            Colonne(updated) : Valeur
            Colonne(type) : Valeur
            Colonne(erdf) : Valeur
            Colonne(svg) : Valeur
            Colonne(png_large) : Valeur
            Colonne(png_small) : Valeur
        }
}
```

L'annexe VI contient le code source Java du package hbase_hadoop et un exemple pour tester les fonctions de gestion des données dans Hbase.

4.11.3 Intégration de HBase dans Oryx

Nous avons intégré la bibliothèque des fonctions de gestion des données du sous-schéma de la base de données POEM sur Hbase décrite dans la section précédente avec Oryx. Le test a échoué à cause de l'incompatibilité de l'analyseur syntaxique des fichiers XML utilisé par Oryx avec celui utilisé par Hadoop.0.20.2. En effet, l'analyseur utilisé par Oryx contenu dans la librairie *xerces* version 2.9.0 utilise le mécanisme « xinclude » pour combiner plusieurs fichiers XML, alors que la bibliothèque Java utilisée par Hadoop.0.20.2 pour gérer la base de données Hbase utilise la version 2.9.1, cette dernière ne reconnaît plus ce mécanisme.

Lors de nos tentatives de résoudre le problème nous avons essayé les options suivantes et aucune d'elle n'est avéré concluante:

1. Dans le fichier de configuration de Hadoop : /conf/hadoop-env.sh , nous avons ajouté la ligne suivante pour forcer la machine virtuelle de Java à utilser l'analyseur syntaxique de Java. Ce dernier est supposé avoir la fonctionnalité de traitement des fichiers XML activée. Nous contournons ainsi, la problématique des deux versions de l'analyseur syntaxique dans le fichier de configuration de Java (classpath) :

   HADOOP_OPTS= "-Djavax.xml.parsers.DocumentBuilderFactory
   =com.sun.apache.xerces.internal.javaxp.DocumentBuilderFactoryImpl"

2. Dans le fichier de configuration de Hadoop /conf/core-site.xml , nous avons forcé l'activation de l'utilisation de l'analyseur syntaxique:

   <configuration xmlns:xi=http://www.w3.org/2001/XInclude>
   ...
   </configuration>

3. Nous avons fait le test avec la version 0.20.0 de hadoop, le module de gestion des données dans Hbase fonctionne correctement alors que la version integrée a échoué avec la même erreur : *javax.xml.parsers.ParserConfigurationException: Feature 'http://apache.org/xml/features/xinclude' is not recognized.*

Nous avons fait la recherche dans les forums des utilisateurs de Hadoop et nous avons constaté que ce problème était déjà connu par la communauté des utilisateurs de Hadoop : l'utilisation de l'analyseur syntaxique des fichiers XML dans hadoop cause beaucoup de problèmes. Plusieurs pistes de solution ont été suggérées et nous les avons essayé mais sans obtenir de résultats concluants.

4.12 Résultat de la recherche

L'originalité de cette recherche porte sur l'exploration de la migration d'une base de données relationnelle vers une base de données non relationnelle via une étude de cas de migration d'un logiciel de gestion de processus d'affaires vers une plateforme d'informatique de nuage.

Un objectif a été fixé pour cette recherche : faire une étude d'impact de migration d'un logiciel, libre de droit, de gestion de processus d'affaires vers Hadoop.

Pour atteindre cet objectif, nous avons fait une revue de la littérature dans les domaines de la gestion des processus d'affaires, des systèmes SGPA, de l'informatique de nuage, des modèles de données relationnels et non relationnels. Nous avons constaté que plusieurs études académiques ont été effectuées sur la définition et les domaines d'application du concept de GPA ainsi que sur les outils de modélisation utilisés, par contre, très peu de chercheurs se sont penchés sur les systèmes des gestion de processus d'affaire, l'informatique de nuage, les modèles des données relationnelles dans le contexte d'informatique de nuage et les modèles de données non relationnelles. D'un autre côté, nous trouvons que les nombreuses études publiées par les compagnies de l'industrie des technologies de l'informatique manquent beaucoup d'objectivité compte tenu de leur caractère publicitaire. Finalement, nous n'avons trouvé aucune étude sur la conversion des systèmes de gestion de processus d'affaires vers une plateforme d'informatique de nuage.

Lors de notre projet de recherche, nous avons constaté et expérimenté les problèmes liés à la conversion d'applications existantes (utilsant les bases de données relationnelles) vers la technologie de l'informatique de nuage. Nous avons développé une liste de critères pour évaluer et choisir un logiciel de gestion de processus d'affaires (SGPA) libre de droit pour notre étude de cas. Après l'analyse et l'expérimentation de plusieurs logiciels SGPA comme Intalio, BonitaSoft, ProcessMaker, JBPM, uEngine, wFMOpen, nous avons choisi Oryx pour notre étude de cas à cause de sa richesse en formalismes de modélisation de processus et de sa flexibilité pour supporter toute nouvelle notation.

Suite à l'analyse du fonctionnement et la rétro ingénierie de la base de données et du code du logiciel de modélisation de processus Oryx, nous avons proposé une stratégie

d'intégration basée sur la création d'un modèle de données hybride : relationnel et non relationnel. Une partie de la base de données relationnelle Postgres a été convertie en modèle non relationnel Hbase et une librairie de fonctions pour gérer ce sous schéma à été créée et testée. Nous avons ensuite mis en place un environnement de développement et des composants fonctionnels pour la partie Hbase de la base de données qui démontrent le potentiel du concept des modèles hybrides. Puis, nous avons développé des guides d'installation des environnements de développement sur la plateforme de Hadoop et Hbase. Ceci nous a permis d'identifier un problème (bogue) de traitement des fichiers XML dans les versions 0.20.0 et 0.20.2 de Hadoop.

En réponse à la question que nous nous sommes posée dans notre recherche, la liste suivante énumère les tâches nécessaires à la migration d'une application existante qui utilise une base de données relationnelle vers une plateforme d'informatique de nuage (Hadoop, Hbase):

1. Analyser la base de données de l'application à convertir et identifier les sous-schémas qui supportent la charge du volume de données et qui seraient plus avantageux de les migrer vers un modèle de données non relationnel
2. Concevoir le modèle de données non relationnel des sous-schémas identifiés
3. Modifier la conception de la base de données relationnelle après l'élimination du sous-schéma migré : revoir les liens logiques entre les tables, les déclencheurs, les vues, les procédures stockées, les clés étrangères, etc.
4. Identifier les modules du code qui gèrent le sous-schéma migré, ainsi que ceux qui seront affectés par la modification de ces modules
5. Développer un code pour gérer le sous-schéma migré vers la nouvelle plateforme on utilisant le API de la base de données de destination (Hbase)
6. Intégrer le nouveau code dans l'application principale et faire les changements nécessaires sur les modules affectés par l'intégration
7. Migrer les données du sous-schéma convertit de la base de données source vers la base de données de destination

CONCLUSION

La complexité des opérations au niveau des organisations, la mondialisation de l'économie, la vitesse et la fréquence des changements imposés aux organisations ont fait de la gestion efficace et rapide des processus d'affaires un élément incontournable pour la survie de ces organisations. D'où l'importance de l'utilisation des systèmes de gestion automatisée des processus d'affaires.

Le marché des logiciels offre un très grand choix de SGPA. Toutefois, les organisations n'ont souvent ni les moyens financiers pour se procurer ces systèmes, ni les moyens organisationnels pour supporter l'infrastructure informatique requise, ni les ressources humaines nécessaires pour supporter les opérations de ces systèmes.

La technologie de l'informatique de nuage qui permet d'offrir les logiciels sous forme de service (SAAS) devient une solution à considérer sérieusement par les organisations qui n'ont pas les moyens de se procurer des SGPA ou qui ne veulent tout simplement pas encombrer leurs infrastructures informatiques actuelles avec de nouveaux systèmes complexes.

Considérant le grand potentiel dans ce marché, les grandes compagnies informatiques comme IBM, Microsoft, HP, Pegasystems, etc. ont commencé une course dans le développement de systèmes SGPA, chacun adapté à sa propre plateforme d'informatique de nuage. Étant donné les grands enjeux financiers de ces opportunités d'affaires, toutes les informations sur les avancées du processus de développement sont gardées secrètes et donc non accessibles pour la communauté scientifique. À notre avis cela représente un obstacle majeur pour une intégration homogène de ces deux technologies sur des bases scientifiques.

Dans le but d'apporter une contribution à ce sujet, nous avons mené l'expérience d'intégration d'un SGPA avec une technologie d'informatique de nuage, tous deux choisis parmi les logiciels libres.

Compte tenu de sa popularité, nous avons choisi Hadoop comme technologie d'informatique de nuage pour notre expérience. En effet, nos recherches nous ont indiqué que plusieurs grandes compagnies utilisent Hadoop, parmi lesquelles : Ebay avec une grappe d'ordinateurs de 532 nœuds, Yahoo avec un réseau de plus de 40 000 ordinateurs, New York times, Linkedln, Blue cloud d'IBM, Facebook, etc.

Vu le grand nombre de formalismes de modélisation de processus d'affaires, nous avons cherché un logiciel libre capable de supporter le maximum de formalismes de modélisation. Dans notre processus de sélection, nous avons fait l'évaluation des logiciels par rapport à une liste de critères que nous avons établis : La disponibilité de tout le code source du logiciel sans aucune restriction, l'adaptabilité du logiciel à la technologie de l'Internet et la richesse en langages de modélisation supportés. Notre processus de sélection nous a permis de choisir Oryx : un logiciel libre de modélisation de processus d'affaires développé par l'institut Hasso-Plattner à l'université de Postdam en Allemagne, supervisé par le professeur Mathias Weske, directeur du groupe de recherche des processus d'affaires.

Nous avons abordé l'intégration d'Oryx dans Hadoop avec la perspective d'améliorer sa capacité de traitement pour un volume important de données.

Oryx utilise une base de données relationnelle PostgreSQL pour la gestion de ses données alors que Hadoop utilise une base de données non relationnelle Hbase inspirée du modèle de base de données BigTable utilisée par Google.

Nous avons envisagé deux scénarios d'intégration : Une migration complète de la base de données relationnelle PostgreSQL vers Hbase et une migration partielle ou hybride, c'est-à-dire faire cohabiter les deux modèles de bases de données dans le même système. Dans notre étude, nous avons privilégié le deuxième scénario, celui d'une migration hybride pour qu'Oryx puisse bénéficier des avantages des deux modèles de bases de données : relationnelles et non relationnelles qui cohabitent dans le même système. Nous avons identifié les sous schémas de la base de données relationnelle susceptibles de recevoir la

charge d'un grand volume de données et nous avons créé leur équivalent dans la base de données non relationnelle Hbase.

À notre avis, les bases de données relationnelles et non relationnelles sont deux modèles complémentaires et non des adversaires comme certains les présentent. Ils peuvent apporter beaucoup plus aux systèmes informatiques s'ils sont utilisés ensemble que séparément.

RECOMMANDATIONS

Nous avons tenté la réalisation d'un prototype (d'Oryx utilisé avec une base de données relationnelle et une base de données No-SQL), Toutefois, nous avons constaté qu'il y avait un bogue au niveau de la bibliothèque de Hadoop 0.20.2 et 0.20.0. Cette dernière ne reconnaissait pas la dernière version de l'analyseur syntaxique « XInclude » utilisé par Oryx.

La question de l'incompatibilité de la version de l'analyseur syntaxique « XInclude » n'est qu'une des pistes à investiguer étant donné que les causes de l'erreur n'ont été clairement identifiées ni lors de nos tentatives de résoudre le problème ni par la communauté des utilisateurs de Hadoop. Toutefois, nous avons identifié trois paramètres pouvant faire partie des causes du problème : La version de Hadoop, la version de la librairie *Xerces* et le système d'exploitation utilisé. Dans notre expérience nous avons essayé les versions de 0.20.0 et 0.20.2 , sur la version 10.2 de Ubunto avec la version 2.9.1 de la librairie *Xerces* qui est supposée avoir la fonctionnalité de traitement des fichiers XML activée.

Pour la suite de ce travail, nous suggérons de résoudre la problématique de traitement des fichiers XML, puis de compléter la réalisation du prototype selon la logique décrite dans le Tableau 4-5 : Liste des changements à apporter sur Oryx.

Dans la perspective d'obtenir un système de gestion des processus d'affaires puissant sur une plateforme hybride de gestionnaire de données qui combine les forces des modèles relationnelles avec celles des modèles non relationnelles, nous suggérons de développer un simulateur de processus d'affaires qui utilise HBase et Hadoop comme système de gestion de données et de l'intégrer avec le prototype d'Oryx qui utilise les systèmes de HBase, Hadoop et PostgresSQL pour la gestion de ses données.

ANNEXE I

LE CADRE DE BASILI

Tableau 4.5 Cadre de BASILI : Sommaire de définition de projet
(Alain Abran, Lucie Laframboise et al. 1999**)**

Définition			
Motivation	**Objet**	**Objectif**	**Utilisateur de la recherche**
• Améliorer l'accessibilité aux systèmes de gestion de processus d'affaires pour les entreprises en proposant des solutions plus abordables et nécessitant moins de ressources humaines et organisationnelles • Étudier de nouvelles technologies • Explorer les possibilités d'intégration de nouvelles technologies pour proposer des solutions à des problèmes d'affaires. • Travailler sur des technologies libres de droits afin d'améliorer l'accéssibilité aux technologies de l'informatique de nuage et à celles des systèmes de gestion de processus d'affaires	Le but principal de cette étude est d'effectuer l'analyse d'impact pour effectuer la migration logicielle d'une technologie à une autre	Explorer ce qui doit être fait pour convertir un logiciel existant vers la technologie de l'informatique en nuage (similaire à BIGTABLE utilisée par Google)	Chercheurs dans le domaine du génie logiciel

Tableau 4.6 Cadre de BASILI : Sommaire de planification de projet
(Alain Abran, Lucie Laframboise et al. 1999**)**

Planification		
Étapes du projet	**Entrées du projet**	**Livrables du projet**
Étudier l'état de l'art concernant les systèmes de gestion de processus et celle de l'informatique de nuage	• L'état de l'art	• Synthèse de l'étude de l'état de l'art de l'informatique de nuage et des systèmes de gestion de processus d'affaire
Identifier des critères de sélection du logiciel libre de GPA.	NA	• Liste des critrères de selection expliqués et priorisés
Identifier un logiciel libre, parmi plusieurs candidats à être convertis vers l'informatique en nuage (avec la technologie du logiciel libre Hadoop).	• Liste des critères de sélection • Liste préliminaires de logiciels libres de gestion de processus d'affaire	• Logiciel de gestion de processus d'affaire libre de droits qui corréspond le plus aux critères de selection
Étudier la conception du logiciel choisi afin d'identifier une approche de migration possible	• Logiciel de gestion de processus d'affaire selectionné • Littérature traitant de la migration des modèles de données relationnelels vers les modèles NO-SQL	• Approche d'intégration du logiciel de gestion de processus • Schéma de base de données Hbase équivalent
Installer et configurer des environnements de développements complexes et effectuer des essais	• Plateforme d'informatique de nuage Hadoop • Logiciel de gestion de donnéés NO-SQL (Hbase) • Composants technologiques du logiciel GPA libre	• Environnement de développement pour la nouvelle solution
Analyser le code du logiciel GPA pour identifier les sections du code à modifier pour intégrer le modèle NO-SQL, modifier le code du logiciel et faire des éssais de la nouvelle solution	• Environnement de développement de la nouvelle solution • Schéma de base de données NO-SQL	• Nouveau code source du logiciel • Implémentation du sous-schéma NO-SQL sur Hbase • Résultat des éssais

Tableau 4.7 Cadre de BASILI : Sommaire de l'execution de projet
(Alain Abran, Lucie Laframboise et al. 1999**)**

Exécution	
• Identifier des critères de sélection du logiciel libre de GPA.	Une liste de critères de sélection de logiciel GPA libre de droit pour être utilsée dans l'étude de cas a été établie.
• Identifier un logiciel libre parmi plusieurs candidats à être convertis vers l'informatique en nuage (avec la technologie du logiciel libre Hadoop). • Faire l'étude et la sélection d'un logiciel candidat pour l'étude de cas.	Nous avons retenu une liste de logiciels GPA libre de droit les plus cités et les plus utilisés, puis nous avons parcouru les fonctionnalités de chacun pour en retenir un groupe de quatre. La liste des critères de sélection a été appliquée sur ce groupe de logiciels et nous avons retenu Oryx pour l'étude de cas.
• Étudier la conception du logiciel afin d'identifier une approche de migration possible ✓ Analyse des scénarios de migration ✓ Étude des schémas de bases de données pour identifier les entités candidates. ✓ Concevoir des schémas équivalents en Hbase (No-SQL) pour la migration	Nous avons éffectué une analyse des technologies et des scénarios de migration des bases de données relationnelles vers les modéles No-SQL, nous avons opté pour le scénario d'une migration partielle, ce qui permet d'expérimenter la cohabitation des modèles relationnels et non relationnels pour obtenir un modèle hybride. Nous avons procédé à une étude détaillée du fonctionnement et de l'architecture d'Oryx afin d'identifier les scénarios de fonctionnement, les sous-schéma de base à migrer et les endroits dans le code source qui doivent être changés pour supporter le nouveau modèle de données. Une fois le sous-schéma de base de données relationnelle à migrer a été identifié, nous avons conçu le schéma équivalent sur Hbase.
• Installer et configurer des environnements de développement complexes puis éffectuer des essais des technologies	Nous avons : - Installé et configuré l'environnement de dévelopement d'Oryx sur le système d'éxploitation Ubunto, - Implanté le sous-schéma de base à migrer sur Hbase, - Créé une librairie de fonction pour manipuler les données dans la base Hbase et faire des essais. - Integré la nouvelle librairie dans le code d'Oryx et utilisé ses méthodes pour gérer les modèles de processus d'affaire dans Hbase - Faire des essais avec deux versions de Hadoop

Tableau 4.8 Cadre de BASILI : Sommaire de l'interpretation des résultats du projet (Alain Abran, Lucie Laframboise et al. 1999

Interprétation		
Contexte d'interprétation	**Extrapolation des résultats**	**Études futures**
une étude de cas avec un logiciel réel	l'étude démontre un potentiel de solutions pour résoudre ce problème et ouvre une voie vers une troisième alternative de modélisation des données : modèles hybrides	• Résoudre la problématique de l'analyseur syntaxique des fichiers XML dans Hadoop et finir l'expérience • Développer un module de simulation de processus d'affaire dans Oryx en utilisant le modèle de données hybride • Considérer la possibilté de développer un processus de conversion automatique des modèles de base de données et ceux de migration des données à partir des bases de données relationnelles vers les bases de données NO-SQL.

ANNEXE II

ÉTAPES DE CONFIGURATION DE L'ENVIRONNEMENT DE DÉVELOPPEMENT

Tableau 4.9 Étapes de configuration de l'environnement de développement

Ref	Item	Sous item
1	**Installer Ubunto**	Installer Ubunto version 10.4 LTS
2	**Installation de Hadoop**	2.1 Installer Java 6 2.2 Créer un utilisateur « hadoop » avec suffisamment de privilèges système. Tout l'environnement de développement sera installés avec le nom d'utilisateur « hadoop » 2.3 Configurer SSH pour permettre les accès sécurisés 2.5 Télécharger et installer Hadoop à partir du site Clouera 2.6 Configurer Hadoop
3	**Installation d'Oryx**	3.1 Installer le serveur Internat Tomcat 3.2 Installer l'environnement de développement de Java « Eclipse » 3.3 Installer le Plugin Aptana nécessaire pour le développement d'Oryx 3.4 Installer le Plugin « subversion » 3.5 Créer un projet Oryx dans Eclipse et charger le code source d'Oryx 3.6 Installer le langage de programmation « Python » 3.7 Installer le serveur de base de données Postgresql version 8 3.8 Créer la base de données POEM dans le serveur de base de données nécessaire pour Oryx
4	**Installation de HBase**	4.1 - Télécharger et installer Hbase
5	**Configurer Eclipse**	5.1 Ajouter les librairies d'Hbase et de hadoop pour pouvoir utiliser les méthodes de la bibliothèque de Hbase

ANNEXE III

INSTALLATION DE HADOOP

1- Pré-requis

Apache Hadoop nécessite l'installation des pré-requis suivants :

- Une version récente de java (au moins la version 6),
- La configuration du protocole de sécurité SSH pour pouvoir échanger les données dans le réseau formé par la grappe d'ordinateurs,
- La désactivation du protocole d'Internet IPV6.

1.1 Installation de Java 6

Voici les commandes pour installer Java sur Ubunto

- $sudo add-apt-repository "deb http://archive.canonical.com/lucid partner"
- $sudo apt-get update
- $sudo apt-get install sun-java6-jdk
- $update-java-alternatives -s java-6-sun

1.2 Configuration de SSH

Voici les étapes à suivre pour installer et configurer le « Shell » sécurisé sur Ubunto :

- Installer le client et le serveur ssh

  $sudo apt-get install openssh-server openssh-client

- Configurer ssh :
  - ✓ Générer une clé ssh pour l'utilisateur « hadoop »

$su – hadoop

$ssh-keygen -t rsa -P "mot_de_passe"

  - ✓ Autoriser l'accès à cette machine avec la clé SSH récemment créée

~$ cat $HOME/.ssh/id_rsa.pub >> $HOME/.ssh/authorized_keys

$ssh localhost

2- Installation de Hadoop

- Télécharger une version stable de Hadoop à partir de site d'Apache suivant : http://hadoop.apache.org/common/releases.html
- Créer un répertoire « Hadoop » et sauvegarder le fichier d'archives téléchargé dans l'étape précédente :

  $mkdir hadoop

- Extraire le fichier d'archives de Hadoop dans le répertoire

  $cd hadoop

  $ sudo tar xzf hadoop-0.20.2.tar.gz

  $ sudo mv hadoop-0.20.2 hadoop

- Changer le propriétaire de tous les fichiers pour l'utilisateur « hadoop »

  $ sudo chown -R hadoop:hadoop hadoop

3- Configuration de Hadoop

Pour notre expérience, nous avons opté pour une installation de Hadoop sur un seul nœud. Le même ordinateur va servir à la fois pour le nœud serveur (name node) et le nœud des données (data node).

- Spécifier le chemin de Java pour Hadoop : Ouvrir le fichier de configuration hadoop-env.sh de hadoop situé dans le répertoire « conf » et changer la ligne qui spécifie le chemin d'installation de java par celui de notre ordinateur :

  $cd <chemin d'installation de hadoop>/conf

  $sudo gedit hadoop-env.sh

- Changer le contenu de la variable globale JAVA_HOME

  export JAVA_HOME=/usr/lib/jvm/java-6-sun

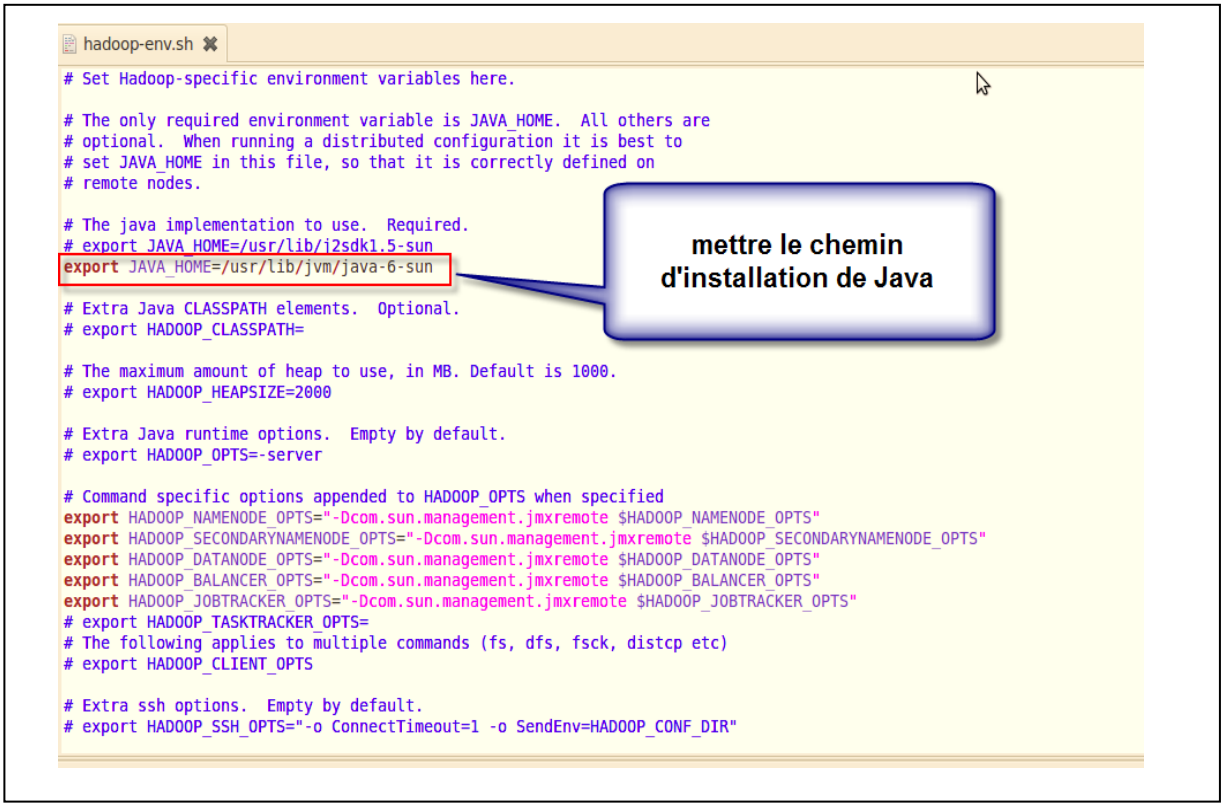

Figure III.12 Configuration de la variable global de Java pour hadoop

- Spécifier le répertoire que Hadoop va utiliser pour sauvegarder les fichiers de données et les numéros de ports qu'ils utilisent.

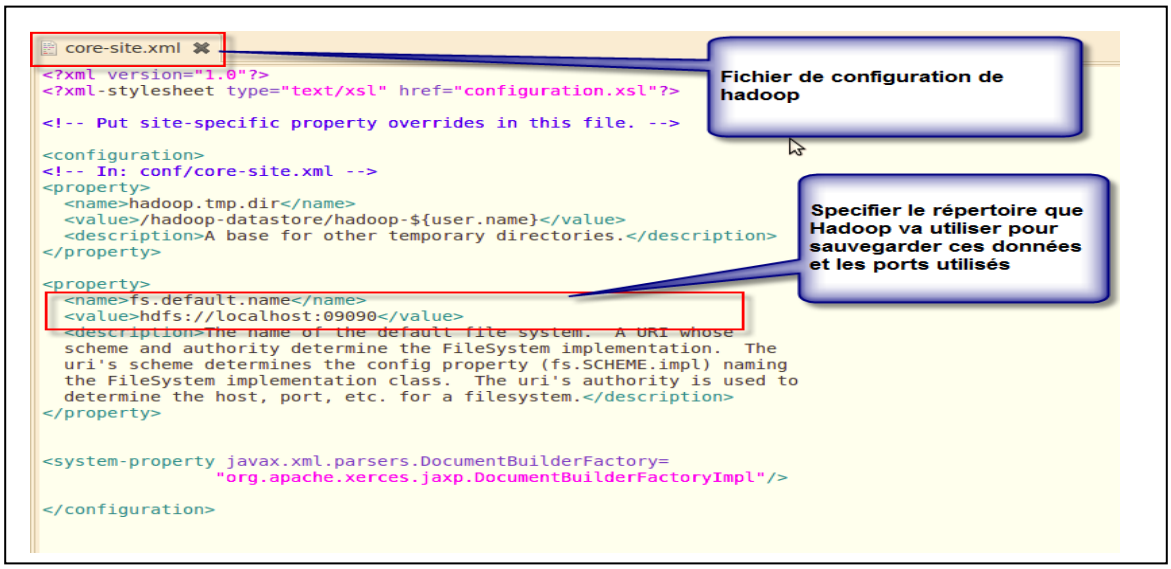

Figure III.2 Configuration du système du fichier de Hadoop

ANNEXE IV

CONFIGURATION DE L'ENVIRONEMENT DE DÉVELOPPEMENT D'Oryx

1- **Installer une version récente de java**

Voir Annexe III

2- Installer une version récente du serveur Tomcat

- Télécharger une version stable de Tomcat à partir de : http://tomcat.apache.org/download-60.cgi

Dans notre expérience nous avons utilisé la version 6 du serveur Tomcat.

3- Installation et configuration d'Eclipse

- Télécharger "Eclipse IDE for Java EE Developers" de http://www.eclipse.org/downloads/. Aucune installation n'est requise, ouvrir simplement le fichier d'archive téléchargé.

- Oryx nécessite le plugiciel « Aptana » pour dessiner les styles de modèles de processus d'affaires.
  Dans le logiciel Eclipse, allés dans « Menu\help\Install new software\add » puis ajouter le lien suivant pour installer le plugiciel "Aptana".
  http://d3lq98emif3szr.cloudfront.net/tools/studio/plugin/install/studio/2.0.5.1278522500/

- Installer le plugiciel « subversion » pour pouvoir charger le code source d'Oryx.
  Dans le logiciel Eclipse, allés dans « Menu\help\Install new software\add » puis ajouter le lien suivant pour installer le plugiciel "Subversion" :
  http://download.eclipse.org/releases/helios

4- Charger le code source d'Oryx

Dans Eclipse, utiliser le plugiciel Subversion pour télécharger le code source d'Oryx. Le code est disponible à l'adresse suivante : http://oryx-editor.googlecode.com/svn/trunk/"

- Dans Eclipse, Sélectionner "File -> Import..." puis choisir "SVN -> Project from SVN"

5- Installer Python version 2.5.2

La version 2.5.2 utilisée par Oryx peut être téléchargée à partir de cette adresse : http://www.python.org/download/releases/2.5.2 . Aucune configuration n'est nécessaire, il suffit simplement d'extraire les fichiers à partir du fichier d'archive ainsi téléchargé.

6- Installation et configuration du serveur de base de données

- Pour installer postgresSql, Il suffit d'exécuter la commande suivante:

  ```
  $sudo apt-get install postgresql-8.4 postgresql-plpython-8.4
  ```

- Une fois installé, nous avons besoin de configurer le serveur de base de données Postgres afin qu'il n'utilise pas de mots de passe pour les connections TCP locales.

  Nous avons besoin d'éditer le fichier pg_hba.conf et de modifier le contenu comme suit :

```
# Database administrative login by UNIX sockets
local    all        postgres          ident    sameuser

# TYPE DATABASE USER CIDR-ADDRESS METHOD

# "local" is for Unix domain socket connections only
local    all        all       trust
# IPv4 local connections:
host     all        all       127.0.0.1/32    trust
# IPv6 local connections:
host      all       all ::1/128       md5
```

- Ajouter le chemin d'installation de PostgresSql aux variables globales du système d'exploitation Ubunto. Nous avons besoin d'éditer le fichier de configuration d'ubunto ~/.bashrc et d'y ajouter le chemin du répertoire « bin » de Postgres

- Créer un utilisateur nommé « POEM » qui peu accéder au serveur de base de données sans mot de passe :

```
$su postgres
$createuser --superuser --echo poem
```

- Créer le schéma de la base de données POEM d'Oryx :

```
$cd /Oryx/poem-jvm/data/database/db_schema.sql
$createuser –U postgres –echo –pwprompt –encrypted poem
$createdb –U postgres –echo –encoding utf8 –owner poem poem
$psql –U postgres --dbname poem --file db_schema.sql
```

ANNEXE V

INSTALLATION D'HBASE SUR UBUNTO

1- **Télécharger Hbase à partir du lien suivant:**
http://www.apache.org/dyn/closer.cgi/hbase/

2- **Ouvrir le fichier d'archive que nous venons de télécharger pour installer Hbase.**

```
$ tar xfz hbase-0.20.6.tar.gz
```

3- **Configurer Hbase pour fonctionner en mode autonome**

Éditer le fichier \hbase\conf\hbase-env.sh et modifier le contenu comme suit :

```
<property>
 <name>hbase.rootdir</name>
 <value>file:///tmp/hbase-${user.name}/hbase</value>
</property>
```

ANNEXE VI

CODE SOURCE POUR LA GESTION DES DONNÉES DANS HBASE

```
package hbase_hadoop;

import org.apache.hadoop.hbase.HBaseConfiguration;
import org.apache.hadoop.hbase.client.Get;
import org.apache.hadoop.hbase.client.HTable;
import org.apache.hadoop.hbase.client.Put;
import org.apache.hadoop.hbase.client.Result;
import org.apache.hadoop.hbase.client.ResultScanner;
import org.apache.hadoop.hbase.client.Scan;
import org.apache.hadoop.hbase.util.Bytes;
import org.apache.hadoop.hbase.client.*;
import org.apache.hadoop.conf.*;
import org.apache.hadoop.hbase.*;
import java.io.*;

public class Hbase_hadoop_methods {

       public static int creerProcessus(String tblename,
              String cfName, String cRow,
              String sIdent, String Smimetype,String sLanguage,
              String sTitle, String sSummary, String sCreated,
              String sUpdated,String sType,String sErdf,
              String sSvg, String spnglarge,String sPngsmall)
     throws IOException {

//Objet de configuration pour informer le client ou se connecter
//L'objet HBaseConfiguration charge les parametres de connection
// dans le fichier hbase-site.xml et hbase-default.xml

          Configuration config = HBaseConfiguration.create();

// creer une nouvelle table pour le nouveau utilisateur
          HTable table = new HTable(config, tblename);
          byte[] row1 = Bytes.toBytes(cRow);
          Put p1 = new Put(row1);
          byte[] databytes = Bytes.toBytes(cfName);
          p1.add(databytes, Bytes.toBytes("Ident_id"),
                  Bytes.toBytes(sIdent));
          p1.add(databytes, Bytes.toBytes("mime_type"),
                  Bytes.toBytes(Smimetype));
          p1.add(databytes, Bytes.toBytes("language"),
                  Bytes.toBytes(sLanguage));
          p1.add(databytes, Bytes.toBytes("title"),
                  Bytes.toBytes(sTitle));
          p1.add(databytes, Bytes.toBytes("summarty"),
                  Bytes.toBytes(sSummary));
```

```
            p1.add(databytes, Bytes.toBytes("created"),
                    Bytes.toBytes(sCreated));
            p1.add(databytes, Bytes.toBytes("updated"),
                Bytes.toBytes(sUpdated));
            p1.add(databytes, Bytes.toBytes("type"), Bytes.toBytes(sType));
            p1.add(databytes, Bytes.toBytes("erdf"), Bytes.toBytes(sErdf));
            p1.add(databytes, Bytes.toBytes("svg"), Bytes.toBytes(sSvg));
            p1.add(databytes, Bytes.toBytes("png_large"),
                Bytes.toBytes(spnglarge));
            p1.add(databytes, Bytes.toBytes("png_small"),
                Bytes.toBytes(sPngsmall));
            table.put(p1);

return 0;
}
      /**************************************************
       * Methode pour retrouver les informations d'un utilisateur
       * @param tblename : Nom de la table, soit l'addresse courriel
       * @param cfName: Nom de las colonne de famille,
       * @param cRow: le nom du processus a charger
       * @param sRec: le nom de l'information recherche
       * @return
       * @throws IOException
       */
       public static String trouverProcessus(String tblename,
            String cfName, String cRow, String sRec) throws IOException {
                Configuration config = HBaseConfiguration.create();
                HTable table = new HTable(config, tblename);
            // Retrieve my values
            // Now, to retrieve the data we just wrote. The values that
            //   come back are
            // Result instances. Generally, a Result is an object that
            //    will package up
            // the hbase return into the form you find most palatable.
            Get g1 = new Get(Bytes.toBytes(cRow));
            Result r = table.get(g1);
            byte [] value = r.getValue(Bytes.toBytes(cfName),
              Bytes.toBytes(sRec));
            // If we convert the value bytes, we should get back 'Some
            // Value', the value we inserted at this location.
            String valueStr = Bytes.toString(value);


            return valueStr;
}
```

```
        /*************************************************
        *  Methode pour creer une table hbase pour un nouvelle utilisateur
        *
        * @param tblename: nom de la table, soit, l'addresse courriel de
        * l'utilisateur
        * @param cfName: nom des colonne de famille qui composent la table4
        * hbase
        * @return
        * @throws IOException
        ************************************************* /

        public static int creerUtilisateur(String tblename, String cfName)
            throws IOException {
              Configuration config = HBaseConfiguration.create();

              // Creer la table hbase
              HBaseAdmin admin = new HBaseAdmin(config);
              HTableDescriptor htd = new HTableDescriptor(tblename);

              HColumnDescriptor hcd = new HColumnDescriptor(cfName);
              htd.addFamily(hcd);

              admin.creerUtilisateur(htd);
              byte[] tablename = htd.getName();

              HTableDescriptor[] tables = admin.listTables();
              if (tables.length == 0 && Bytes.equals(tablename,
                 tables[0].getName())) {
                    throw new IOException("Creation de la table: Echec");
           }

       return 0;
}
        /************************
         * Fonction main pour tester les methode de gestion de la base de
         * donne Hbase @param args
         ****************************************/
        public  static void main(String[] args)  {
              int nouvel_utilisateur,nouvelle_Ligne;

              Configuration config = null;

               try {
              nouvel_utilisateur=creerUtilisateur("newuser_address_email",
                  "Nom_processus");
              } catch (IOException e1) {
                    // TODO Auto-generated catch block
                    e1.printStackTrace();
              }
           try {
              nouvelle_Ligne = creerProcessus("newuser_address_email",
               "Nom_processus", "process_1", "1", "mimetype", "UTF-8",
               "Titre processus 1", "sommaire","date de creation",
               "date de modification","Type","ERDF","SVG", "png large",
               "png small");
```

```
                } catch (IOException e) {
                      // TODO Auto-generated catch block
                      e.printStackTrace();
                }
                  try {
                System.out.println("GET:**** " +
                   trouverProcessus("newuser_address_email","Nom_processus",
                                    "process_1","title"));
                System.out.println("GET:**** " +
                   trouverProcessus("newuser_address_email","Nom_processus",
                                    "process_1","mime_type"));
                System.out.println("GET:**** " +
                   trouverProcessus("newuser_address_email","Nom_processus",
                                   "process_1","language"));
                System.out.println("GET:**** " +
                   trouverProcessus("newuser_address_email","Nom_processus",
                                   "process_1","summarty"));
                System.out.println("GET:**** " +
                   trouverProcessus("newuser_address_email","Nom_processus",
                                   "process_1","created"));
                System.out.println("GET:**** " +
                   trouverProcessus("newuser_address_email","Nom_processus",
                                   "process_1","created"));
                System.out.println("GET:**** " +
                   trouverProcessus("newuser_address_email","Nom_processus",
                                   "process_1","updated"));
                System.out.println("GET:**** " +
                   trouverProcessus("newuser_address_email","Nom_processus",
                                   "process_1","erdf"));

                } catch (IOException e) {
                            // TODO Auto-generated catch block
                            e.printStackTrace();
        }

}


}
```

LISTE DE RÉFÉRENCES BIBLIOGRAPHIQUES

Abouzied, A., K. Bajda-Pawlikowski, et al. (2010). *HadoopDB in action: Building real world applications*. 2010 International Conference on Management of Data, SIGMOD '10, June 6, 2010 - June 11, 2010, Indianapolis, IN, United states, Association for Computing Machinery.

Alain Abran, Lucie Laframboise, et al. (1999). *A Risk Assessment Method and Grid for Software Measurement Programs*. Submitted to Communications of the ACM, Janvier 1999.

Borthakur, D. (2007). "*HDFS Architecture*." The Apache Software Foundation 1 - 14.

Cantara, M. (2008) "*Business Processes and Cloud Computing*." Gartner Summit Events.

Decker, G., H. Overdick, et al. (2008). *Oryx - Sharing conceptual models on the web 27th International Conference on Conceptual Modeling*, ER 2008. T. Springer Verlag, Heidelberg, D-69121, Germany. Barcelona, Spain, Springer Verlag, Tiergartenstrasse 17, Heidelberg, D-69121, Germany. 5231 LNCS: 536-537.

Elzinga, D. J., T. Horak, et al. (1995). "*Business process management: survey and methodology*." Engineering Management, IEEE Transactions on 42(2): 119-128.

Fay, C., J. Dean, et al. (2008). "*Bigtable: a distributed storage system for structured data*." ACM Transactions on Computer Systems **26**(Copyright 2009, The Institution of Engineering and Technology): 4 (26 pp.).

Hadoop. (2010). "*Welcome to Apache Hadoop!*" Accédé le June-09-2010 2010, de http://hadoop.apache.org.

Harmon, P. (2007). "*Exploring BPMS with Free or Open Source Products*." Accédé le 23 Juin 2010, 2010, de http://www.bptrends.com/publicationfiles/advisor200707312.pdf .

Ian Foster, Yong Zhao, et al. (2008). "*Cloud Computing and Grid Computing 360-Degree Compared*." accédé le Accédé le 23 Juin 2010, 2010, de http://arxiv.org/ftp/arxiv/papers/0901/0901.0131.pdf.

ISO/IEC JTC 1/SC 38 (2010). *Report of JTC 1/SWG-P on possible future work on Cloud Computing in JTC 1, JTC 1 SWG on Planning***:** 23.

Java-source.net. (2010). "*Open Source Workflow Engines in Java.*" Accédé le 23 Juin 2010, de http://java-source.net/open-source/workflow-engines.

Jeffrey Dean and S. Ghemawat (2004). *MapReduce: Simplified Data Processing on Large Clusters.* I. Google. OSDI**.**

Manageability. (2010). "*Open Source Workflow Engines Written in Java.*" Accédé le 23 Juin 2010, de http://www.manageability.org/blog/stuff/workflow_in_java.

Martin A. Czuchra. (2007). Oryx: Embedding Business Process Data into the Web. Hasso Plattner institut.

Martin A. Czuchra. (2007). Oryx: Embedding Business Process Data into the Web. Hasso Plattner institut.

Maatuk, Abdelsalam, Akhtar Ali et al (2008). *Relational database migration: a perspective*. 19th International Conference, DEXA 2008, Berlin, Germany

Michael Armbrust, Armando Fox, et al. (2009). *Above the Clouds: A Berkeley View of Cloud Computing.* EECS Department University of California - Berkeley.

Oracle Corporation (2008). *State of the Business Process Management Market*, Oracle. Accédé le 23 Juin 2010 de http://hosteddocs.ittoolbox.com/soa_us_en_wp_state.pdf.

Recker, J. C., M. Indulska, et al. (2006). *How good is BPMN really? Insights from theory and practice*. 14th European Conference on Information Systems. Goeteborg, Sweden.

Shafer, J. R., S ; Cox, A.L (2010). *The Hadoop Distributed Filesystem: Balancing Portability and Performance*. 2010 IEEE International Symposium on Performance Analysis of Systems & Software (ISPASS 2010). White Plains, NY, USA, IEEE**:** 122-133.

Van der Aalst, W. M. P. (2004). *Business process management demystified: a tutorial on models, systems and standards for workflow management.* Lectures on concurrency and Petri nets. Advances in Petri nets (Lecture Notes in Comput. Sci. Vol.3098). Berlin, Germany, Springer-Verlag**:** 1-65.

Wohed, P., N. Russell, et al. (2009). "*Patterns-based evaluation of open source BPM systems: The cases of jBPM, OpenWFE, and Enhydra Shark.*" Information and Software Technology 51(Compendex): 1187-1216.

Zairi, M. (1997) "*Business process management: a boundaryless approach to modern competitiveness.*" Business Process Management Journal 3, 64-80 DOI: 10.1108/14637159710161585.