

ÉCOLE DE TECHNOLOGIE SUPÉRIEURE
UNIVERSITÉ DU QUÉBEC

MÉMOIRE PRÉSENTÉ À
L'ÉCOLE DE TECHNOLOGIE SUPÉRIEURE

COMME EXIGENCE PARTIELLE
À L'OBTENTION DE LA
MAÎTRISE EN GENIE
CONCENTRATION: RÉSEAUX DE TÉLÉCOMMUNICATIONS
M. ING.

PAR
Fahmi TOUATI

ENTREPÔT DE DONNÉES ORIENTÉ SERVICES

MONTRÉAL, LE 23 JUIN 2014

©Tous droits réservés, Fahmi TOUATI, 2014

©Tous droits réservés

Cette licence signifie qu'il est interdit de reproduire, d'enregistrer ou de diffuser en tout ou en partie, le présent document. Le lecteur qui désire imprimer ou conserver sur un autre media une partie importante de ce document, doit obligatoirement en demander l'autorisation à l'auteur.

PRÉSENTATION DU JURY

CE MÉMOIRE A ÉTÉ ÉVALUÉ

PAR UN JURY COMPOSÉ DE :

Dr. Alain Abran, directeur de Maîtrise
Département de génie logiciel & technologie de l'information à l'École de technologie supérieure

Dr. Mohamed Taleb, codirecteur de Maîtrise
Département de génie logiciel & technologie de l'information à l'École de technologie supérieure

Dr. Abdelouahed Gherbi, président du jury
Département de génie logiciel & technologie de l'information à l'École de technologie supérieure

Dr. Ghizlane El Boussaidi, membre du jury
Département de génie logiciel & technologie de l'information à l'École de technologie supérieure

IL A FAIT L'OBJET D'UNE SOUTENANCE DEVANT JURY ET PUBLIC

12 JUIN 2014

À L'ÉCOLE DE TECHNOLOGIE SUPÉRIEURE

REMERCIEMENTS

En préambule de ce projet, je souhaite exprimer mes gratitude et mes reconnaissances envers des personnes qui m'ont soutenu en me relevant de toutes mes peines. Pour cela, je tiens à remercier:

Monsieur Alain Abran mon directeur de projet, professeur à l'École de technologie supérieure, pour l'orientation, pour m'avoir confié ce travail de recherche ainsi que pour l'aide et le temps qu'il m'a consacré.

Monsieur Mohamed Taleb mon co-directeur de projet, professeur associé à l'École de technologie supérieure, pour sa disponibilité et l'effort qu'il a consacré pour réussir ce projet, ses conseils méthodologiques, son expertise et ses critiques échangées dans de longues discussions et tout au long de mon projet. Il a tout cru en moi en m'encourageant à surmonter les difficultés et en mettant à ma disposition une salle équipée. Je n'oublie pas ses accueils bienveillants dans son bureau, ses écoutes, sa gentillesse et ses sourires.

Un grand merci à ma très chère famille, Papa, Maman, mes sœurs Asma et Meriem, pour l'aide et la patience, pour l'encouragement et leurs présences à mes côtés.

ENTREPÔT DE DONNÉES ORIENTÉ SERVICES

Fahmi TOUATI

RÉSUMÉ

La diversité des applications, l'hétérogénéité des sources de données et la multiplication des techniques ont rendu les systèmes d'informations de plus en plus complexes et le dialogue entre organisateurs et informaticiens difficile. Une des solutions apportées pour rapprocher les technologies de l'information et les besoins métiers est l'architecture orientée services - *Service-Oriented Architecture* (SOA) pour tenir compte de l'existant, garantir la performance et la simplicité. Malgré cette flexibilité, Inmon trouve que la SOA est basée sur le transactionnel, sur des données en perpétuelles modifications auxquelles s'ajoutent l'hétérogénéité et la diversité des formats. Pour réagir et anticiper aux événements, l'entreprise doit s'appuyer sur un fonds de données historisées orientés métiers, fiables et sans ambiguïté d'interprétations afin de répondre adéquatement aux besoins stratégiques. Une solution qui répond aux exigences de l'entreprise est l'architecture des entrepôts de données mais pose des problèmes tels que le coût par la diversité des produits qui sont propriétaires et non réutilisables.

Pour pallier à ces insuffisances, Inmon considère que la SOA est la solution qui répond idéalement aux exigences de l'entreprise à condition qu'elle soit assise sur une base de connaissances qui est l'entrepôt de données. Notre contribution dans ce projet de recherche consiste à intégrer l'entrepôt de données dans la SOA appelé «*Service Oriented Data Warehouse Architecture* (SODWA)».

Ce projet couvre uniquement la partie conception de la SODWA. La conception intègre les deux technologies standards éprouvées qui sont les services Web et l'«*Enterprise Service Bus* (ESB)» : elle intègre les services métiers de fournisseurs d'applications d'analyses tels que l'«*Online Analytical Processing* (OLAP)», les services métiers de fournisseurs d'intégration des données tels que l'extraction, la transformation et le chargement (ETL pour *Extract Transform and Loading*) et un référentiel de données composé de l'entrepôt de données et des «*Data Marts*».

Ce document est divisé en trois chapitres. Le premier chapitre présente l'état de l'art sur les architectures entrepôt de données et les architectures orientées services. Le deuxième chapitre introduit la problématique de la recherche avec ses buts, ses objectifs, ses limites et la méthodologie de recherche. Le troisième chapitre présente la conception d'un Entrepôt de données orientée services - Service-Oriented Data Warehouse Architecture (SODWA) pour faciliter l'intégration de l'entrepôt de données dans une architecture orientée services en présentant ses concepts clés, ses justifications, sa vue détaillée ainsi qu'un exemple d'utilisation de la SODWA. Une évaluation qualitative préliminaire et partielle de la solution est aussi présentée.

Mots-clés: Entrepôt de données, SOA, ESB, Services Web, SODWA, ETL, OLAP

DATA WAREHOUSE SERVICE-ORIENTED

Fahmi TOUATI

ABSTRACT

The diversity of applications, heterogeneous data sources and propagation techniques has made information more complex and the dialogue between organizers and computer systems more difficult. One of the solutions to bring the information technology and business needs is Service-Oriented Architecture (SOA) to respond to user requests and take into account the existing guarantee the performance and simplicity. Despite this flexibility, Inmon mentions that SOA is based on the transactional data in constant changes together with heterogeneity and diversity of formats. To anticipate and react to events, organization should rely upon a historical data base oriented business, with reliable and unambiguous interpretations to adequately meet the strategic needs. A solution is a data warehouse architecture but it poses problems such as costs coming from the diversity of proprietary and non-reusable products.

To overcome these shortcomings, Inmon believes that SOA is a solution that meets the requirements of the company provided that it is sitting on a knowledge base that is the data warehouse. Our contribution in this research project is to integrate the data warehouse into SOA, referred to as a Service Oriented Data Warehouse Architecture (SODWA).

This project covers only the design of the SODWA. This design integrates two proven technologies: standard Web services and Enterprise Service Bus (ESB). This architecture integrates the services of the providers of analyses applications, such as the Online Analytical Processing (OLAP) applications, business services provider of data integration such as extraction, transformation and loading (ETL) and a data repository which consists of a data warehouse and Data Marts.

This document is divided into three chapters. The first chapter presents the state of the art on data warehouse architectures and Service-oriented architectures. The second chapter introduces the research goals, objectives, limitations, and the research methodology. The third chapter presents the design of the Service-Oriented Data Warehouse Architecture (SODWA) to facilitate the integration of the data warehouse into a Service-oriented architecture with its core concepts, its rationale, and an example of using the SODWA. A preliminary and partial qualitative assessment of the architecture is also presented

Keywords: Data warehouse, SOA, ESB, Web services, SODWA, ETL, OLAP

TABLE DES MATIÈRES

	Page
INTRODUCTION	1
CHAPITRE 1 ÉTAT DE L'ART	5
1.1 Entrepôt de données	5
1.1.1 Définitions	5
1.1.2 Architecture d'un entrepôt de données	8
1.2 Architecture orientée service	12
1.2.1 Définitions	12
1.2.2 Les technologies standards de mise en œuvre de la SOA	21
1.2.2.1 Les services web	21
1.2.2.2 Entreprise Service Bus (ESB)	25
1.3 SOA et Entrepôt de données	27
1.4 Pourquoi combiner les entrepôts de données et la SOA	29
1.5 Sommaire du chapitre	32
CHAPITRE 2 LA PROBLÉMATIQUE DE LA RECHERCHE	33
2.1 But et objectifs de la recherche	33
2.2 Limites de la recherche	34
2.3 Méthodologie de la recherche	34
2.4 Sommaire du chapitre	36
CHAPITRE 3 ENTREPÔT DE DONNÉES ORIENTÉ SERVICES	37
3.1 Les concepts clés de la SODWA	37
3.2 Justifications de la SODWA	39
3.3 Vue détaillée de la SODWA	40
3.4 Exemple d'utilisation de la SODWA	48
3.5 Évaluation qualitative partielle et préliminaire de la SODWA	55
3.6 Sommaire du chapitre	59
CONCLUSION	61
ANNEXE I LES PHASES DE DÉVELOPPEMENT D'UN PROJET D'ENTREPÔT DE DONNÉES	63
ANNEXE II LES LOGICIELS ETL ET LES LOGICIELS D'ANALYSES ET DE RAPPORTS	67
BIBLIOGRAPHIE	71

LISTE DES FIGURES

	Page	
Figure 1.1	Données orientées sujets dans un entrepôt de données	6
Figure 1.2	Données non volatiles dans un entrepôt de données	7
Figure 1.3	Architecture simplifiée d'Inmon: Usine de l'information d'entreprise	8
Figure 1.4	Architecture d'entrepôt de données de Kimball	9
Figure 1.5	Invocation d'un service dans une architecture orientée services	14
Figure 1.6	Les différentes couches de services	16
Figure 1.7	Situation traditionnelle	18
Figure 1.8	Services métiers	18
Figure 1.9	Services techniques	19
Figure 1.10	Services et couches dans l'architecture orientée service	20
Figure 1.11	Principe de fonctionnement des services Web	23
Figure 1.12	Enterprise Service Bus	26
Figure 1.13	Modèle de processus pour un entrepôt de données dans l'architecture SOA	29
Figure 2.1	Méthodologie de recherche	35
Figure 3.1	Vue Globale de la SODWA	38
Figure 3.2	Vue Détailée de la SODWA	40
Figure 3.3	Rapport simple Tirée de IBM (2012, p. 22)	45
Figure 3.4	Formation des employés par niveau d'organisation	46
Figure 3.5	Exposition des services par l'ESB	49
Figure 3.6	Invocation du service métier sélection par l'ESB	50
Figure 3.7	Invocation du service métier extraction par l'ESB	51

Figure 3.8	Invocation du service métier chargement par l'ESB	52
Figure 3.9	Invocation du service métier OLAP par l'ESB.....	53
Figure 3.10	Invocation du service métier tableau de bord par l'ESB	54
Figure 3.11	Utilisation de l'ETL traditionnel.....	57
Figure 3.12	Conception d'intégration des composants de l'ETL dans une SOA.....	58

LISTE DES ABRÉVIATIONS, SIGLES ET ACRONYMES

BPM	Business Processing Management
ESB	Enterprise Service Bus
ETL	Extraction / Transformation / Loading
FTP	File Transfer Protocol
HTTP	Hyper Text Transfer Protocol
JCA	J2EE Connector Architecture
OLAP	Online Analytical Processing
REST	Representational State Transfer
SMTP	Simple Mail Transfer Protocol
SOA	Service-Oriented Architecture
SOAP	Simple Object Access Protocol
SODWA	Service-Oriented Data Warehouse Architecture
UDDI	Universal Description Discovery and Integration
URL	Uniform Resource Locator
W3C	World Wide Web Consortium
WSDL	Web Services Description Language
WS-POLICY	Web Services Policy
WS-SECURITY	Web Services Security
XML	Extensible Markup Language
XML-RPC	Extensible Markup Language-Remote Procedure Call
XSD	XML Schema Definition

INTRODUCTION

0.1 Mise en contexte

Aujourd’hui, l’investissement simultané dans la SOA et l’architecture d’entrepôt de données est considérable.

En effet, la SOA constitue une solution pour l’intégration des applications hétérogènes en des services métiers réutilisables afin d’accroître la souplesse et d’augmenter la rentabilité de l’entreprise. Cependant, cette architecture est dans l’incapacité de répondre aux nouveaux besoins stratégiques de l’entreprise qui sont des besoins décisionnels, évolutifs, dynamiques et qui doivent s’appuyer sur des données intégrées, cohérentes et historisées et sur un référentiel de données (Inmon, 2006).

L’architecture d’entrepôt de données est une solution qui répond aux exigences précitées par la mise en place d’un gisement de données de connaissances orientées sujets, fiables, avec un historique sur la donnée et conçu en étroite collaboration avec l’utilisateur et l’informaticien (Kimball, 1996) et (Inmon, 2005). Pour exploiter cette base de connaissances, divers fournisseurs d’applications d’analyses, d’exploitation et d’intégrations de données ont développé leurs propres solutions pour répondre aux besoins exprimés par l’utilisateur (*Voir Annexe II, p.83*). L’avantage de ces produits est de donner plus d’autonomie à l’utilisateur en formulant ses propres requêtes de façon dynamique via des interfaces conviviales et de déployer ses résultats au niveau de l’entreprise. Cependant, certaines applications ont des fonctionnalités plus développées que d’autres ce qui oblige l’entreprise à acquérir à chaque fois un nouveau produit spécifique pour répondre à ses besoins. Cette diversité d’applications a confronté l’entreprise d’une part à des coûts d’investissements importants (le prix élevé, la portabilité de ces applications des fournisseurs, l’obligation d’acheter des fournisseurs externes tous les produits de leurs applications logiciel, etc.) et d’autre part n’envisage que les utilisateurs internes de l’entreprise ce qui la laisse enfermée et isolée de l’extérieur pour l’échange de données.

Pour pallier à ces insuffisances, la SOA est une solution jugée idéale selon (Inmon, 2006), mais avec une architecture qui s'appuie sur des données intégrées qui est l'entrepôt de données pour donner plus de confiance à l'entreprise à prendre des décisions objectives sur l'existant et le futur projeté.

0.2 Objectif et méthodologie de recherche

Notre objectif de recherche, est formulé comme suit : « Conception d'intégration de l'entrepôt de données dans une architecture SOA en tant que référentiels de données basé sur des services métiers réutilisables et faiblement couplés pour l'intégration des données , l'analyse et l'exploitation des données en utilisant des technologies standards pour faciliter la maintenance et l'évolution».

Pour atteindre l'objectif de la recherche, ce projet présente la conception d'un entrepôt de données orienté services (SODWA). La SODWA est basée sur les concepts clés de l'architecture orientée services et sur les principes fondamentaux de l'architecture des entrepôts de données comme illustré dans la Figure 0.1.

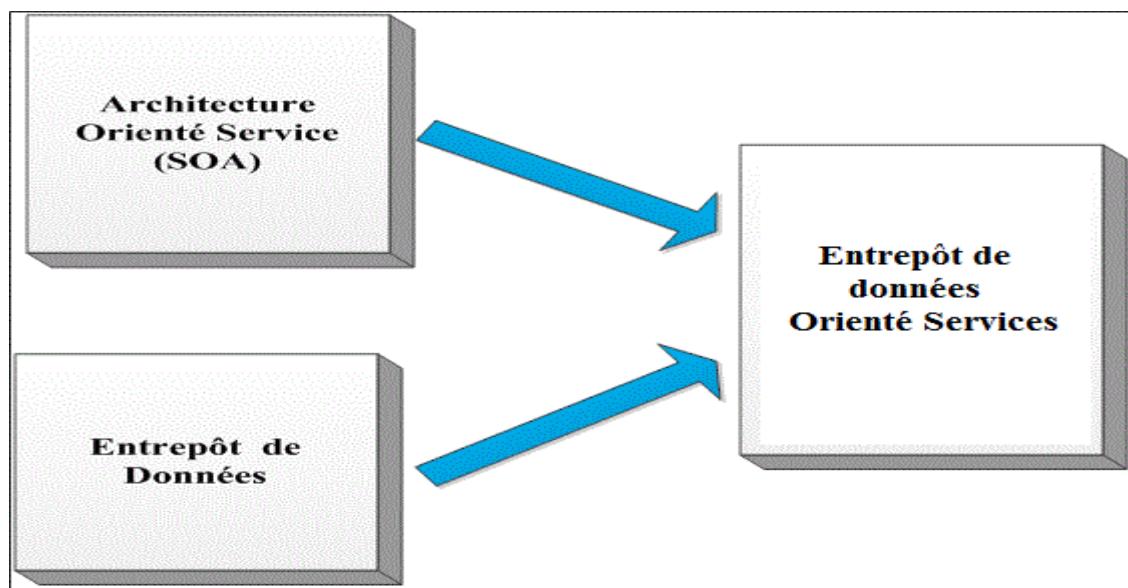


Figure 0.1 Entrepôt de données orienté services

Notre projet de recherche se limitera à la conception pour l'intégration de l'entrepôt de données dans la SOA.

La méthodologie consistera à intégrer dans la nouvelle conception les composants suivants:

- le référentiel métier (l'entrepôt de données, les «Data Marts »),
- l'extraction, la transformation et le chargement des données (ETL pour « *Extract Transform Loading* »),
- le moteur « *Online Analytical Processing (OLAP)* » pour l'analyse et l'exploration rapide des données, et
- les applications d'aide à la décision.

Une description détaillée dans le chapitre 3 montre le fonctionnement des différents composants de la SODWA.

Dans la SODWA, tous les services seront gouvernés par le middleware ESB qui découpe le consommateur du service du fournisseur du service, et centralise la gestion des processus métiers pour assurer l'intégration des données et l'intégration des applications.

0.3 Organisation du document

Ce document est composé de trois chapitres :

Le premier chapitre présente une revue de la littérature concernant les différents aspects de l'architecture d'entrepôts de données et l'architecture orientée services à savoir: leurs principes de fonctionnements, et des points forts et des points faibles de ces architectures.

Le deuxième chapitre définit l'objectif du projet de recherche ainsi que les limites de la recherche et de la méthodologie de la recherche.

Le troisième chapitre présente les résultats de ce projet de recherche, décrivant essentiellement la conception d'une nouvelle architecture appelée SODWA pour faciliter

l'intégration des entrepôts de données dans l'architecture orientée services incluant: ses principes et concepts clés, ses justifications et sa description détaillée.

Enfin, ce travail résume les principales contributions, les forces et les faiblesses, et les perspectives futures de recherche.

CHAPITRE 1

ÉTAT DE L'ART

Ce chapitre présente une revue de la littérature sur l'architecture des entrepôts données et l'architecture SOA.

La première section introduit certaines définitions des entrepôts de données, les architectures des entrepôts de données et quelques approches d'analyses menées sur ces architectures. La seconde section introduit certaines définitions de l'architecture SOA, les concepts clés de l'architecture et les différents styles d'architecture suivis d'une approche d'analyse. La troisième section discute de la combinaison de l'architecture des entrepôts de données et de la SOA. La quatrième section présente une évaluation de ces architectures et identifie les différences dans ces architectures et la problématique de recherche reliée.

1.1 Entrepôt de données

1.1.1 Définitions

La définition informatique d'un entrepôt de données introduite par (Inmon, 2005) dans son ouvrage de référence « *Building the Data Warehouse* » est la suivante :

« *A data warehouse is a subject-oriented, integrated, non-volatile, and time-variant collection of data in support of management's decisions* ».

Données orientées sujet (parfois traduit par: données orientées métier)

Cette approche des données orientées sujet (parfois traduit par: données orientées métier) permet de développer le système décisionnel (entrepôt de données) via une incrémentation sujet par sujet. Cette orientation permet d'éviter la redondance des données concernées par plusieurs sujets. La Figure 1.1 présente un exemple des données orientées sujet dans un entrepôt de données telles que Client, Vendeur, Produit (Vangenot, 2005).

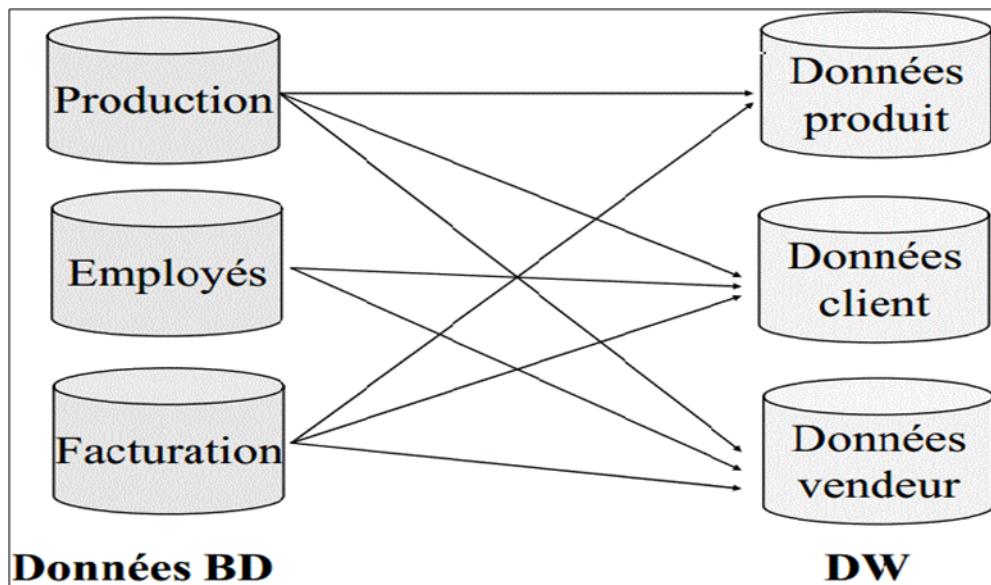


Figure 1.1 Données orientées sujets dans un entrepôt de données
Tirée de Vangenot (2005, p. 14)

Données intégrées

L’entrepôt de données est alimenté par plusieurs sources de données opérationnelles ou transactionnelles déjà existantes dans l’entreprise. Une donnée issue de l’opérationnel peut avoir des formats différents, des attributs différents, des descriptions différentes, etc. Pour avoir une image physique unique des données dans l’entrepôt de données, les données doivent être intégrées c’est-à-dire converties, reformatées, résumées, etc. (Inmon, 2005).

Données historisées

L’analyse de l’évolution des données dans le temps est fondamentale pour un système décisionnel. Elle permet à la fois de prendre des décisions et de mesurer leurs effets (Inmon, 2005).

Données non-volatiles

La non-volatilité dans un système décisionnel consiste à historiser les données tout en gardant les versions de mise à jour durant leurs existences. Il s’agit toujours d’une opération d’insertion dans l’historique et non de mise à jour.

La Figure 1.2 illustre la non-volatilité des données dans un entrepôt de données. Dans une base de données opérationnelle (à gauche de la Figure 1.2), les données sont fréquemment manipulées, consultées et mises à jour à chaque enregistrement. Alors que dans un entrepôt de données (à droite de la Figure 1.2), les données sont chargées généralement en masse et accessibles en lecture. Toutes les modifications ultérieures se traduisent par des insertions d'enregistrements avec conservation de l'historique (Inmon, 2005).

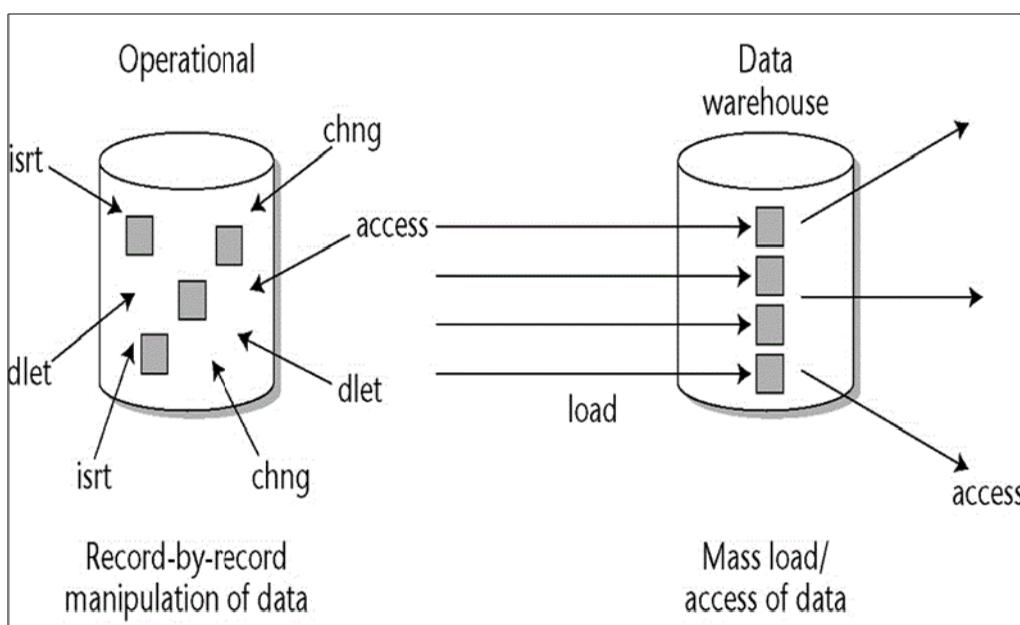


Figure 1.2 Données non volatiles dans un entrepôt de données
Tirée de Inmon (2005)

D'autres définitions ont été suggérées pour l'entrepôt de données:

- (Kimball, 1996), «*A data warehouse is a copy of transaction data specifically structured for query and analysis*»;
- (Rainardi, 2008, p.1),

«*A data warehouse is a system that retrieves and consolidates data periodically from the source systems into a dimensional or normalized data store. It usually keeps years of history and is queried for business intelligence or other analytical activities. It is typically updated in batches, not every time a transaction happens in the source system*».

Pour développer un projet d'entrepôt de données, les différentes phases de développement sont présentées à l'ANNEXE I.

1.1.2 Architecture d'un entrepôt de données

(Adamson, 2010) a donné une représentation simplifiée de l'architecture d'Inmon telle qu'elle apparaît dans la Figure 1.3. Cette architecture est composée de trois parties:

- l'ETL un processus qui collecte les données du système opérationnel sous différents formats pour les homogénéiser;
- les bases de données qui englobent une base de données normalisée (entrepôt de données) et des bases de données multidimensionnelles orientées sujets alimentées par l'entrepôt de données;
- la partie exploitation et visualisation des données pour l'analyse, la fouille et l'exploration des données afin d'aider l'utilisateur final à prendre les bonnes décisions.

Des exemples d'outils ETL et d'outils d'analyses et d'explorations sont présentés et détaillés à l'ANNEXE II.

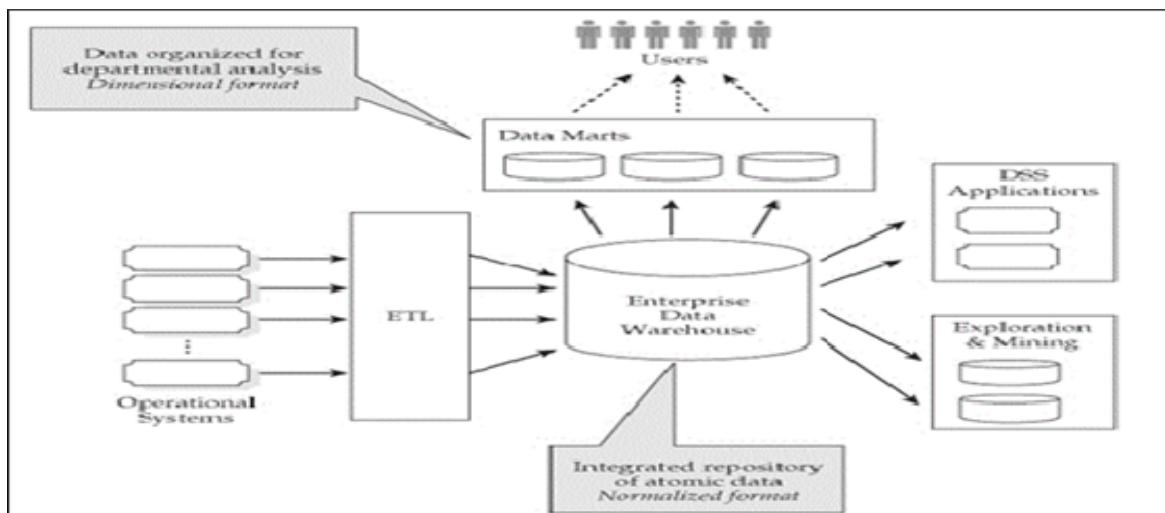


Figure 1.3 Architecture simplifiée d'Inmon: Usine de l'information d'entreprise
Tirée de Adamson (2010)

Sur cette approche d'architecture, Adamson a voulu corriger un mythe: «Le projet de loi Inmon est anti-schéma en étoile». En effet, Inmon préconise la modélisation en étoile au niveau des magasins de données (« Data Marts ») et ceci est justifié par ses propres travaux de recherche.

Par contre (Kimball, 1996) a introduit une autre architecture d'entrepôts de données (Voir Figure 1.4) dont l'approche est similaire à l'architecture d'Inmon pour ce qui est de l'utilisation des données opérationnelles par la couche ETL. Ce qui la différencie de la méthode d'Inmon est la vision sur la construction de l'entrepôt de données : Kimball construit l'entrepôt de données à partir des besoins des utilisateurs avec la modélisation multidimensionnelle. Les données sont organisées dans une série de dimensions en utilisant la modélisation de « schéma en étoile » ou de cubes analysés à partir des besoins identifiés.

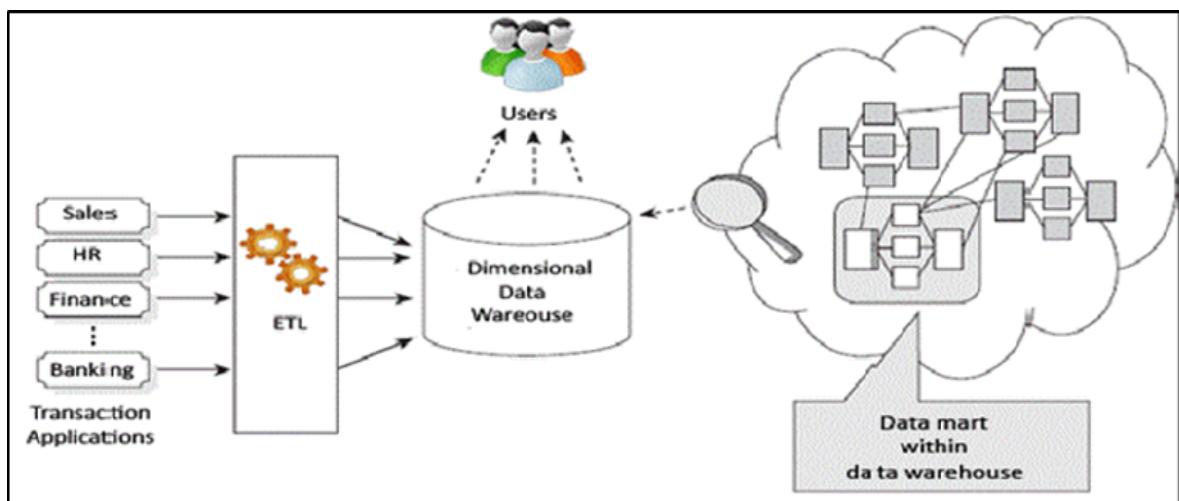


Figure 1.4 Architecture d'entrepôt de données de Kimball
Tirée de Zentut (2013)

Ces deux approches d'architectures d'Inmon et Kimball ont fait l'objet de discussions sur Internet sous le slogan « Inmon versus Kimball »: les discussions sont pratiquement de même type et ne font que répéter plus ou moins les mêmes points.

Une critique qui nous paraît plus complète est celle de (Breslin, 2004) qui a soulevé les similitudes et les différences entre Inmon et Kimball (*Voir le Tableau 1.1*).

Tableau 1.1 Comparaisons des modèles Inmon Vs Kimball
Tiré de Breslin (2004, p.16)

	Inmon	Kimball
Methodology and architecture		
Over all approach	Top down	Bottom up
Architectural structure	Enterprise wide (atomic) data warehouse "feeds" departmental databases	Data marts model a single business process, enterprise consistency achieved through data bus and conformed dimensions
Complexity of the method	Quite complex	Fairly simple
Comparaison with established development methodologies	Derived from the spiral methodology	Four step process ; a departure from RDBMS methods
Discussion of physical design	Fairly thorough	Fairly light
Data modeling		
Data orientation	Subject or data driven	Process oriented
Tools	Traditional (ERDs, DISs)	Dimensional modeling ; a departure from relational modeling
End-users accessibility	Low	High
Philosophy		
Primary audience	IT professionals	End users
Place in the organization	Integral part of the Corporate Information Factory (CIF)	Transformer and retainer of operational data
Objective	Deliver a sound technical solution based on proven database methods and technologies	Deliver a solution that makes it easy for end users to directly query the data and still get reasonable response times

Selon (Breslin, 2004), les principaux points de similitude sont :

- l'attribut temps: dans l'approche Inmon, cet attribut est lié à une base de données qui peut se propager dans différentes tables normalisées. Dans l'approche Kimball, l'attribut temps est représenté dans une seule dimension;
- les deux approches utilisent le processus d'extraction, transformation et chargement pour développer l'entrepôt de données. Les données extraites de différentes sources de données doivent être intégrées, optimisées et transformées avant le chargement dans la base de données.

Pour les différences, (Breslin, 2004) mentionne qu'elles sont à la fois au niveau de la méthodologie de développement, de la modélisation des données et de la philosophie:

Méthodologie et architecture

Inmon propose une approche architecturale descendante pour créer un entrepôt de données centralisé. Inmon utilise les techniques traditionnelles de modélisation des bases de données (Diagramme Entité-Relation (ER)) où les données sont stockées en troisième forme normale (3FN). Pour compléter le modèle de données, tous les «*Data Marts*» obtiennent leurs données à travers cette base. Les efforts de développement de cette approche architecturale descendante a un certain degré de complexité inévitable bien que la présentation d'Inmon soit claire. L'intérêt principal d'Inmon est de s'assurer que la solution technique fonctionne, ce qui a rendu l'architecture et sa méthodologie trop techniques. L'architecture d'Inmon assure la consistance et la cohérence des données car ces dernières proviennent d'un entrepôt de données atomiques.

En revanche, Kimball prévoit quatre étapes pour l'architecture et sa méthodologie de développement : (a) la sélection du processus d'affaire, (b) la déclaration du niveau détail de la granularité des données dans le modèle dimensionnel, (c) le choix des dimensions et (d) l'identification des faits. Cette modélisation est facilement accessible par l'utilisateur final. Il peut comprendre les techniques de concept des dimensions sans étude approfondie contrairement à apprendre et à interpréter les diagrammes entités relations (DER).

Modélisation des données

Pour la modélisation des données, Inmon adopte l'orientation vers les données par les règles et les techniques de modélisation. Par contre, Kimball prend une orientation processus d'affaires qui signifie que la modélisation des données devient une tentative pour définir l'interaction des données à travers le processus d'affaire.

Philosophie

La philosophie d'Inmon consiste à commencer par la construction d'un grand entrepôt centralisé de données basée sur des techniques solides, orienté sujet, intégré, variant dans le temps et non-volatile.

Contrairement à celle d'Inmon, la philosophie de Kimball consiste à commencer tout d'abord par les « *Data Marts* » qui répondent aux besoins analytiques de l'entreprise. Ensuite, il faut intégrer dans ces « *Data Marts* » des données de façon cohérente. Kimball fait usage du modèle multidimensionnel pour répondre aux besoins de l'entreprise dans différents domaines.

1.2 Architecture orientée service

1.2.1 Définitions

Les technologies et innovations informatiques sont en rapide évolution afin de répondre aux besoins grandissants des utilisateurs concernant l'accès à des sources de données multiples, l'interopérabilité et l'adaptation à des environnements hétérogènes. Cette évolution se reflète à travers la migration des applications monolithiques et centralisées vers des applications distribuées et, plus concrètement aujourd'hui, par la mise en place de systèmes d'informations réparties reposant sur l'architecture SOA.

(He, 2003) a introduit la SOA comme suit:

« SOA is an architectural style whose goal is to achieve loose coupling among interacting software agents. A service is a unit of work done by a service provider to achieve desired and results for a service consumer. Both provider and consumer are roles played by software agents on behalf of their owners».

Cette définition de (He, 2003) introduit la SOA comme étant un style architectural qui consiste à organiser des fonctionnalités techniques et métiers en services, fonctions logicielles autonomes, réutilisables et indépendantes qui peuvent être accédées via un réseau. Comme son nom le suggère, le mot clé de la SOA est le service. Cependant, il n'y a pas de définitions communes de ce qu'est un service. La définition de (Arsanjani, 2004) est largement référencée dans la littérature.

D'après (Arsanjani, 2004), le service est défini comme:

«A service is a software resource (discoverable) with an externalized service description. This service description is available for searching, binding, and invocation by a service consumer. The service provider realizes the service description implementation and also delivers the quality of service requirements to the service consumer. Services should ideally be governed by declarative policies and thus support a dynamically re-configurable architectural style».

La Figure 1.5 illustre un exemple d'architecture orientée services.

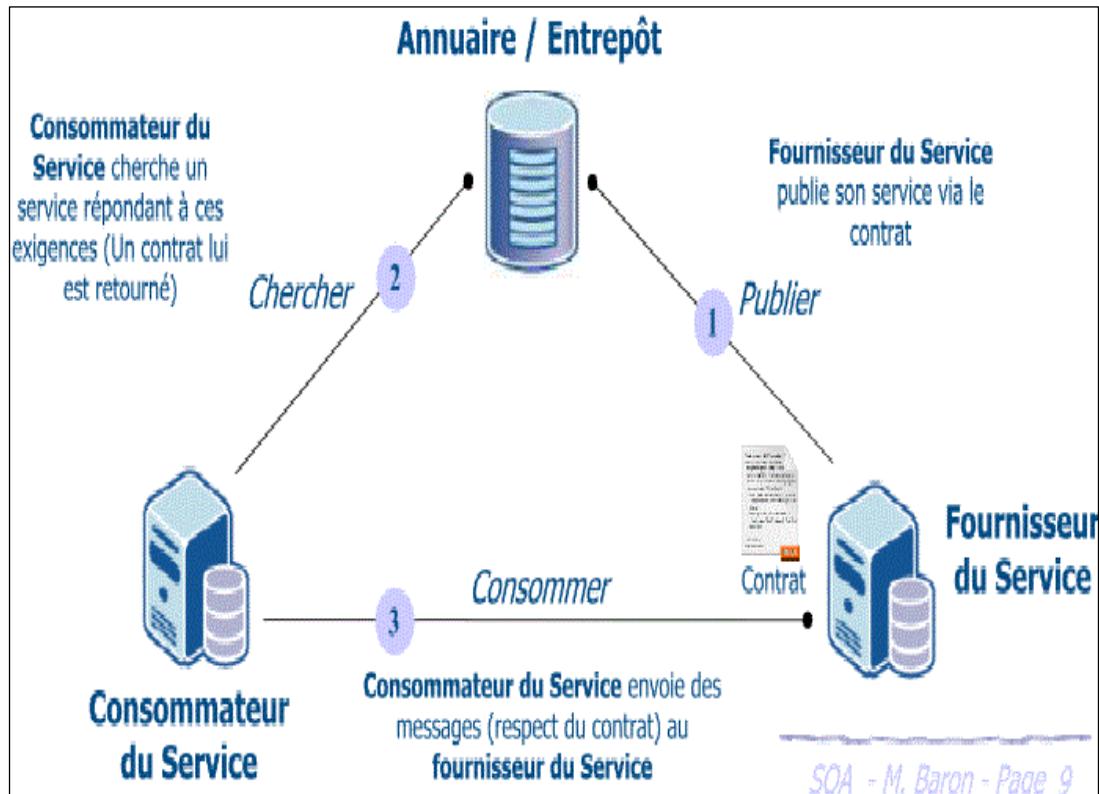


Figure 1.5 Invocation d'un service dans une architecture orientée services
Tirée de Toumi (2012)

Dans la Figure 1.5, le fournisseur du service publie son service dans un annuaire-entrepôt. Le consommateur du service cherche dans l'annuaire le service demandé qui répond à ses exigences. Un contrat est retourné au consommateur du service qui définit l'accord avec le fournisseur du service. En respectant le contrat, le consommateur du service invoque le service par envoi des messages au fournisseur du service.

Dans la recherche bibliographique, il n'y a pas une définition unique du SOA. Certaines définitions sur la SOA, se focalisent sur l'aspect technique, d'autres sur l'aspect métier ou bien sur l'aspect métier et technique.

Une définition par aspect est donnée ci-dessous :

•

«A Service-Oriented Architecture (SOA) is a software architecture that is based on the key concepts of service, service repository, and service bus. A service consists of a contract, one or more interfaces, and an implementation» (Krafzig et al, 2004);

Cette définition introduit la SOA comme étant une architecture applicative basée sur les concepts clés des services, l'annuaire pour la publication des services et le bus pour faire communiquer les services;

•

« SOA is a conceptual business architecture where business functionality, or application logic, is made available to SOA users, or consumers, as shared, reusable services on an IT network. "Services" in an SOA are modules of business or application functionality with exposed interfaces, and are invoked by messages» (Marks, 2006);

Cette définition introduit la SOA comme étant un concept métier, l'aspect technique touche le réseau pour la communication entre les services;

•

« SOA is the architectural style that supports loosely coupled services to enable business flexibility in an interoperable, technology-agnostic manner. SOA consists of a composite set of business-aligned services that support a flexible and dynamically re-configurable end-to-end business processes realization using interface-based service descriptions» (Arsanjani et al, 2008);

Cette définition introduit la SOA comme étant un style architectural qui prend en charge bout à bout les processus métiers d'une manière flexible et dynamique en utilisant les descriptions des services dans l'annuaire.

(Hernández, 2008) illustre dans la Figure 1.6 une vue d'ensemble de l'architecture SOA. La SOA est composée de trois couches de services avec une interface de services (*service repository*) qui référence tous les services recensés dans l'architecture comme suit:

a) Couche Services d'applications

Cette couche contient les fonctionnalités de la couche système. Les applications existantes peuvent être utilisées avec optimisation ou redéveloppées pour les exposer sous forme de services. La plupart du temps, cette couche est implémentée en premier lieu. Chaque service dans cette couche n'appartient qu'à un seul domaine fonctionnel précis qui lui revient. Les technologies standards pour l'invocation des services permettent de cacher les implémentations propriétaires. D'autres services sont d'ordre technique, tels que les services de *logging*, les services de gestion d'exception et les services sur les données sur les opérations de lecture et d'écriture.

b) Couche services métiers

Cette couche contient les services métiers relatifs aux entités (client, fournisseur, etc.) et les traitements métiers (gestion des clients, gestion des commandes, gestion des ventes, etc.).

c) Couche services de processus

Le processus métier est un ensemble de services métiers qui s'exécutent successivement selon un flux de travail et d'une façon automatique. Les processus métiers peuvent être modélisés à l'aide des outils métiers BPM puis développés et testés.

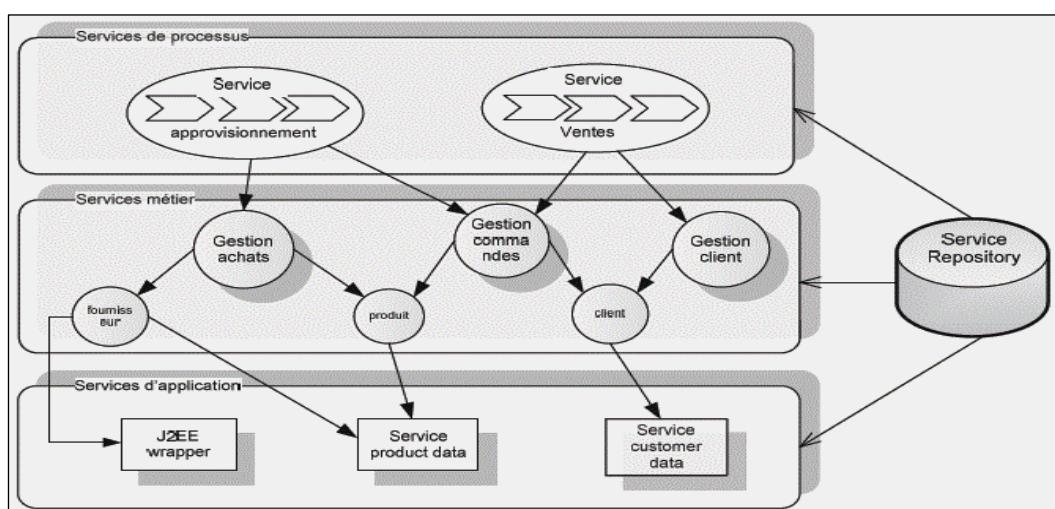


Figure 1.6 Les différentes couches de services
Tirée de Hernández (2008, p. 32)

Par ailleurs, (Van den Berg *et al*, 2007) ont décrit sept concepts de base de l'architecture SOA dans leur ouvrage de référence «*SOA For Profit*» qui sont les suivants:

1) Subdiviser

La SOA assimile les services à des petits blocs de traitement qui peuvent solliciter des tâches pour ajouter une valeur. Si on considère ses services comme des supports métiers, il y a une raison valable pour les subdiviser. Les services doivent être indépendants les uns des autres autant que possible bien définis et ont des activités dans l'entreprise.

2) Convenir de la méthode

L'architecture orientée services utilise un nombre de règles adoptées dans le domaine d'intégration d'applications d'entreprises telles que l'intégration des données et les normes. L'intégration des données repose sur le principe à avoir la même signification d'une donnée utilisée dans des domaines différents de l'entreprise. Les normes consistent à trouver l'accord pour pouvoir communiquer avec les partenaires qui utilisent des langages différents. Au niveau technique par exemple, l'accord est de trouver la manière d'écriture du message sur le réseau. Tandis qu'à un niveau plus élevé, l'accord est la manière de trouver un service disponible au sein du réseau.

3) Utiliser l'existant

L'étude de l'existant est nécessaire avant de se lancer sur la création de nouveaux éléments car certains besoins nouveaux peuvent déjà exister. Avec la SOA, on doit déterminer les éléments informatiques implémentés plusieurs fois pour les mettre en place une seule fois afin de simplifier la maintenance et gagner moins de compétence à entretenir. Ces éléments informatiques sont des services. Un service est réutilisable s'il est indépendant du contexte. Il doit fournir des résultats prévisibles et réagir indifféremment de la situation présentée. La réutilisation se concentre généralement sur la tâche la plus élémentaire réutilisée dans des processus métiers.

4) Passer du sur mesure à l'infrastructure

La Figure 1.7 illustre un exemple de la situation traditionnelle d'un hôpital et montre que le Labo IRM, les soins intensifs et la maternité ont une opération commune qui est l'enregistrement du patient. Cette prise en charge est un service qui peut s'acquérir pour toute l'organisation.

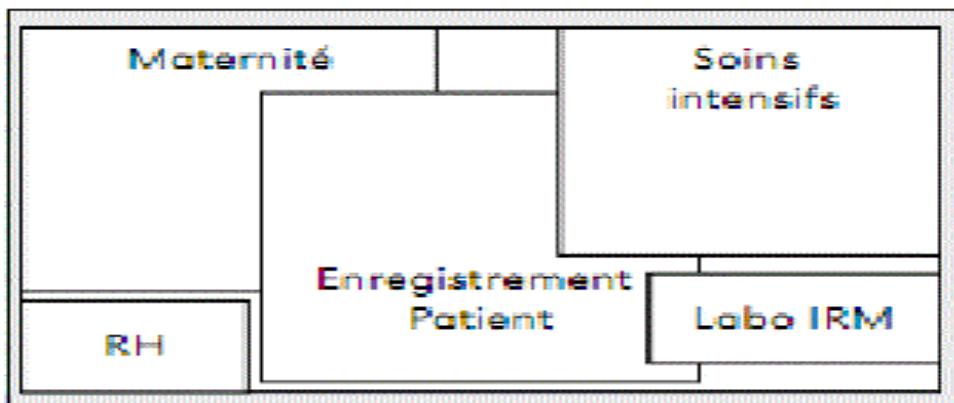


Figure 1.7 Situation traditionnelle
Tirée de Van den Berg (2007, p. 53)

Avec la SOA, les services (Régime d'alimentation, Opération, Médiation, Diagnostic, etc.) sont des services métiers et peuvent être réutilisés au sein de l'organisation, tels qu'ils apparaissent dans la Figure 1.8.

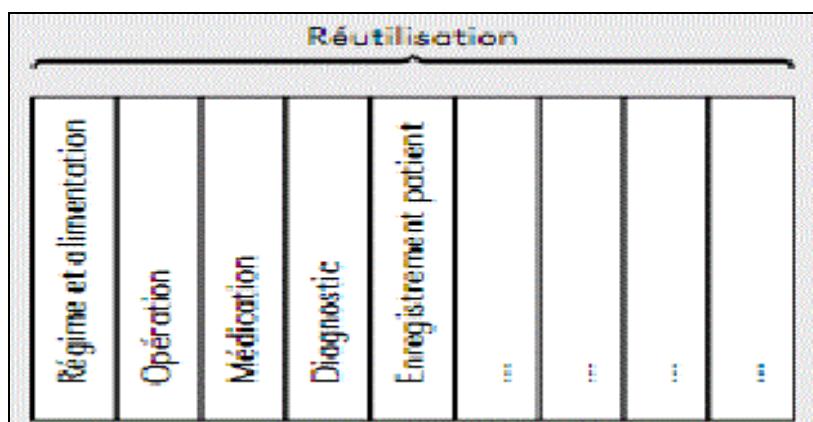


Figure 1.8 Services métiers
Tirée de Van den Berg (2007, p.53)

Avec ces services métiers différents, la SOA permet la réutilisation pour dégager d'autres services communs tels qu'ils apparaissent dans la Figure 1.9. Ces services sont des outils standards ou services techniques (sécurité, autorisations, etc.) qui facilitent la maintenance et la rapidité au niveau développement.

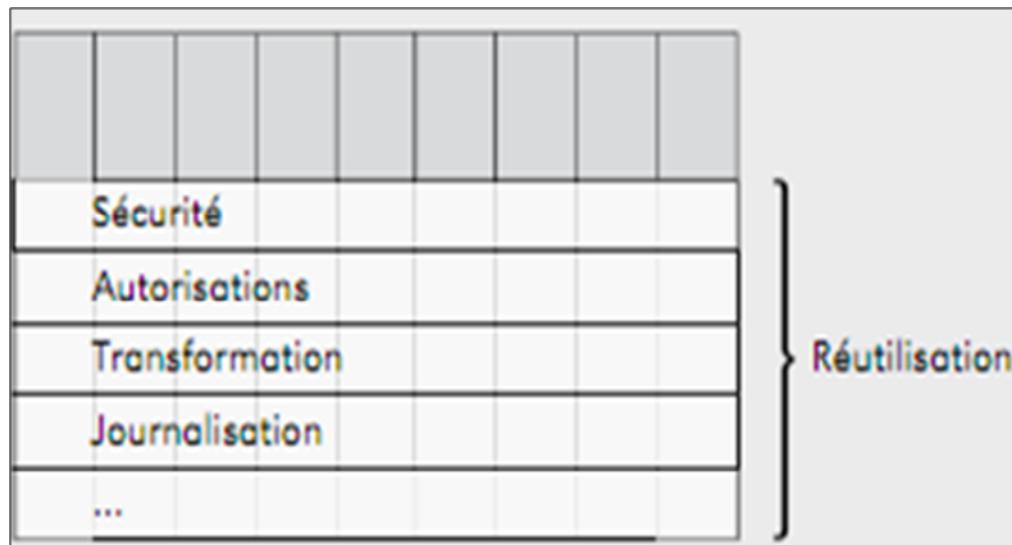


Figure 1.9 Services techniques
Tirée de Van den berg (2007, p.54)

La Figure 1.10 présente ainsi la nouvelle organisation du travail sous une architecture SOA. Dans cette architecture, les applications (maternité, soins intensifs, R.H, Soins intensifs, etc.) sont des processus métiers situés dans la couche supérieure appelée processus métiers. Ces processus font appel à des services métiers réutilisables (régime et alimentations, opération, médication, etc.) et présentés dans la couche sous-jacente appelée services métiers. Ces services métiers font appel à des services techniques réutilisables (sécurité, transformation, journalisation, etc.) dans la couche appelée services techniques. Enfin une quatrième couche appelée couche infrastructure physique ou couche technologique où on trouve les bases de données, les serveurs d'applications, etc. et sont appelés directement par les services techniques.

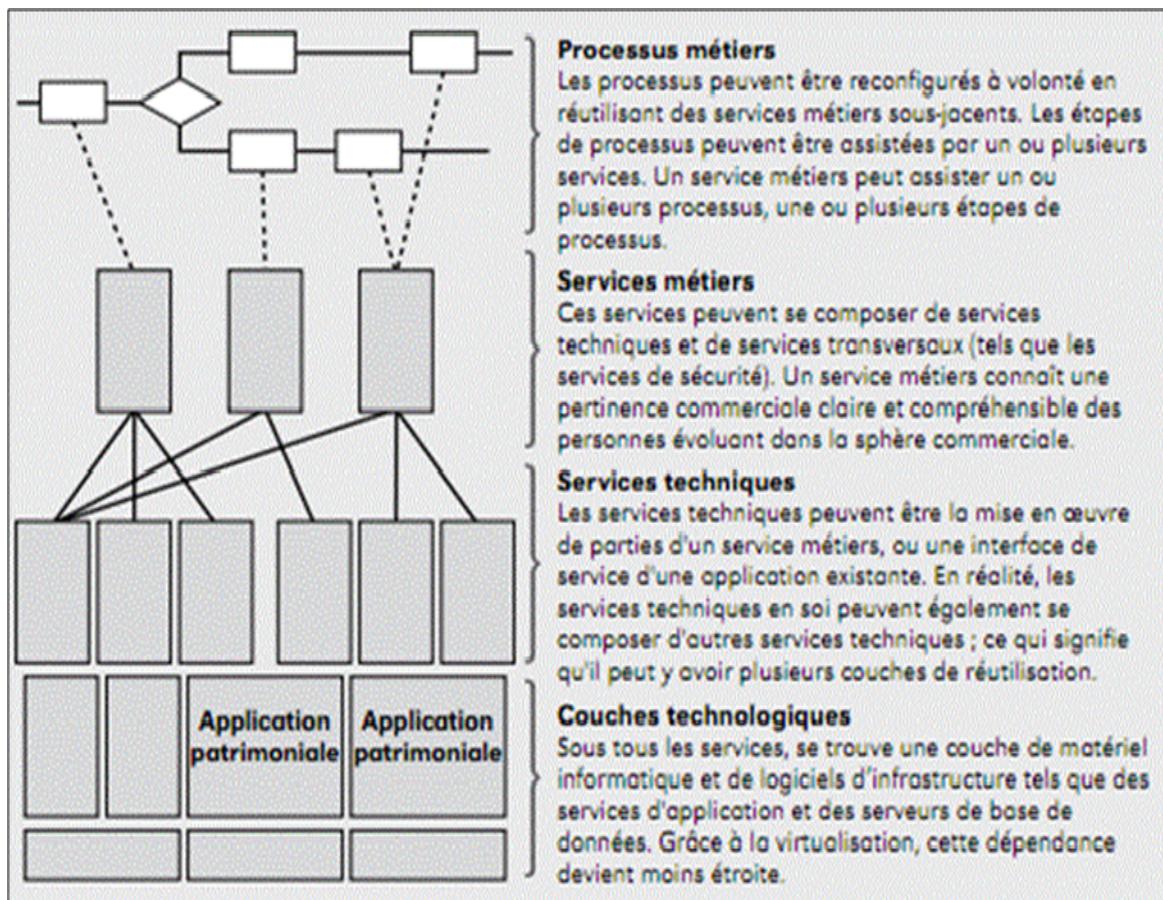


Figure 1.10 Services et couches dans l'architecture orientée service
Tirée de Van den Berg (2007, p.54)

5) Faciliter le changement, s'améliorer en continu

Les concepts de base « 1. Subdiviser » et « 4. Passer du sur mesure à l'infrastructure » permettent d'atteindre l'agilité nécessaire au changement. L'utilisation des méthodes de conception, les méthodes de développement de logiciel et la réutilisation des services contribuent à une amélioration continue, augmentent la productivité et réduisent les risques d'erreurs.

6) Changer pour une raison commerciale

La dépendance des applications avec l'organisation interne de l'entreprise a créé une rigidité entre les services. Pour réduire cette dépendance, la SOA a pu y parvenir en créant un système indépendant de l'organisationnel et répondant aux besoins évolutifs de l'activité

commerciale. La solution portée par la SOA est la création de services réels qui répondent aux activités, stables dans le temps avec une dépendance légère de l'organisation interne de l'entreprise. Les services par la suite peuvent être regroupés pour fournir des services complexes à des consommateurs de services. Cette approche demande un alignement de l'informatique avec l'orientation métier.

7) Réagir à l'environnement

L'informatique doit porter des solutions agiles pour répondre aux changements continus liés aux exigences des activités de l'entreprise. La mise à jour d'une information suite à un événement déclenché est importante pour les prises de décisions et fait raccourcir les délais d'attente. Par conséquent, il faut changer du concept par lot au concept événementiel. Les services peuvent être configurés pour répondre aux événements déclenchés.

1.2.2 Les technologies standards de mise en œuvre de la SOA

1.2.2.1 Les services web

Parmi les technologies standards les plus répandues pour mettre en œuvre la SOA sont les services Web. Ces services sont des tâches automatiques visibles sur internet et qui peuvent s'exécuter de l'intérieur ou de l'extérieur de l'entreprise. Ces services intègrent des applications hétérogènes de l'entreprise et de ses partenaires en utilisant des mécanismes standards orientées web (Boukadi, 2009, p.7).

(Shreeraz, 2007) définit le service Web comme suit:

« Functions or methods that are published on a Web server and which can be invoked remotely from the Internet or Intranet using XML messaging that is based on standards such as WSDL, UDDI and SOAP».

Dans cette définition, le service web est introduit comme une méthode ou fonction publiée à partir d'un serveur Web et invoquée soit de l'intérieur ou de l'extérieur de l'entreprise en utilisant la messagerie XML basée sur les technologies standards « *Web Services Description*

Language (WSDL) », le « *Simple Object Access Protocol* (SOAP) et l' « *Universal Description Discovery and Integration* (UDDI) ».

Ces trois technologies standards constituent les concepts de base des services Web (Mtiraoui, 2013):

- le **SOAP** protocole de communication basé sur XML pour l'échange d'information entre le fournisseur et le consommateur du service;
- le **WSDL** en format XML pour la description de l'interface du service web pour le consommateur du service, les détails techniques, le protocole d'accès et les points d'entrées;
- l'**UDDI** basé sur XML pour enregistrer et publier aux clients les descriptions des services Web sous forme de fichiers WSDL.

Pour communiquer avec les services web, les styles de communication des messages (Mtiraoui, 2013) sont :

- *Representational State Transfer* (REST): le service Web est appelé par une URL en *HyperText Transfer Protocol* (HTTP) puis le serveur envoie la réponse la plupart du temps en XML;
- *Remote procedure call* (XML-RPC) utilise le protocole HTTP pour le transport des données en XML;
- SOAP est un protocole récent éditée par W3C, assure la messagerie dans des environnements distribués;
- *Simple Mail Transfer Protocol* (SMTP) ou *File Transfer Protocol* (FTP) pour le transfert des documents.

La Figure 1.11 schématise le principe de fonctionnement des services web.

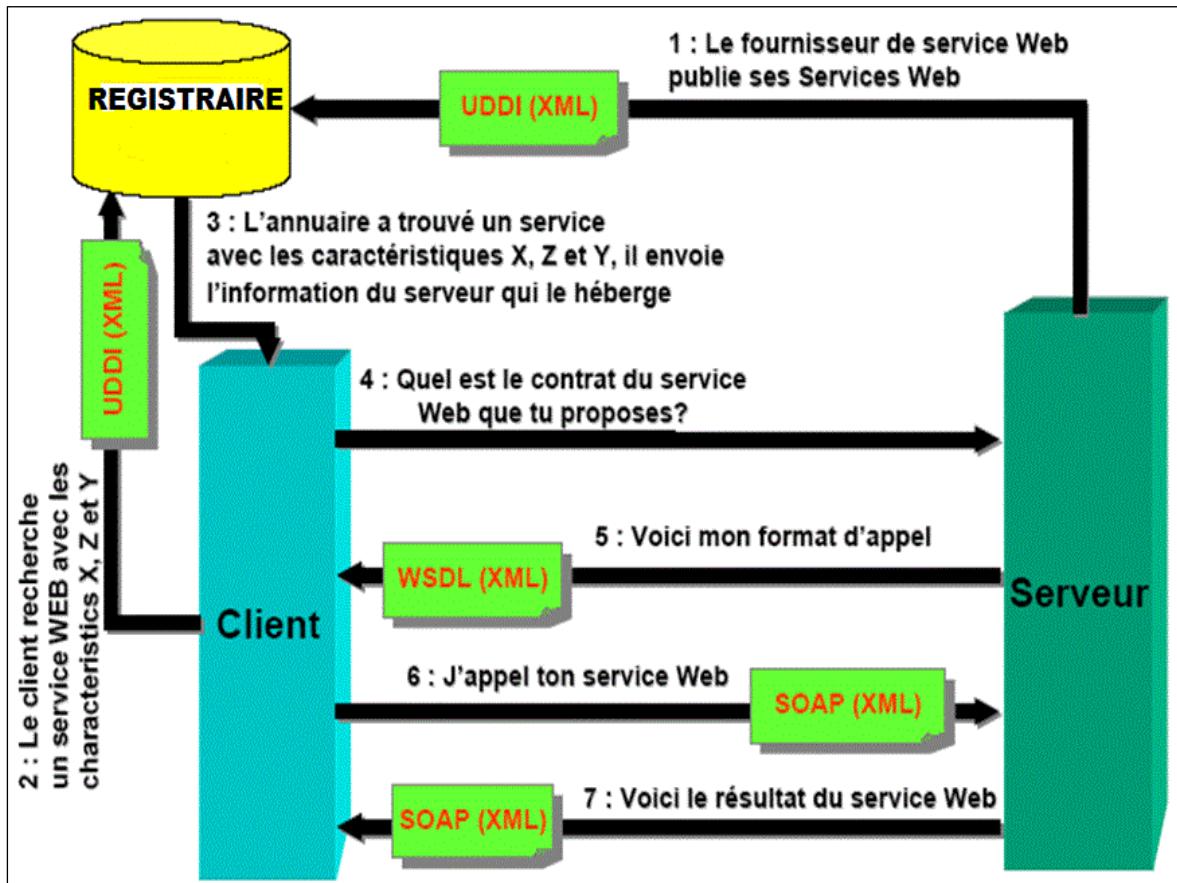


Figure 1.11 Principe de fonctionnement des services Web
Tirée de Cottin (2007, p. 11)

Le principe de fonctionnement (Figure 1.11) est décrit en 7 étapes suivantes :

Dans l'étape 1, le fournisseur de services Web (Serveur) déploie et publie à travers l'Internet son service dans un annuaire de services (UDDI) en décrivant ses caractéristiques dans un fichier WSDL en XML.

Dans l'étape 2, le client (le consommateur de services Web) envoie une requête SOAP à l'annuaire de service en spécifiant ses propres besoins de recherches.

Dans l'étape 3, l'annuaire cherche les services selon les critères spécifiés par le client et lui retourne tous les URL des services disponibles.

Dans l'étape 4, le client demande au fournisseur de services de spécifier le contrat du service du point de vue représentation technique, description informelle du traitement et les engagements du service.

Dans l'étape 5, le fournisseur de services envoie au client le contrat du service.

Dans l'étape 6, le client appelle le service Web avec des requêtes SOAP en XML en spécifiant la méthode d'appel.

Dans l'étape 7, le serveur envoie au client le résultat par un message SOAP en XML.

Selon (Mtiraoui, 2013), les technologies utilisées par les services Web ont pour but de maximiser l'indépendance des services (échange direct de documents structurés dans un environnement hétérogène), de minimiser les contraintes de leurs utilisations (indépendance de la plateforme utilisée et du langage utilisé), de définir rapidement de nouveaux services et de pouvoir les assembler avec les services existants en cas de besoin.

Selon (Tari et *al*, 2011), le trafic sur le réseau est l'un des problèmes rencontrés par les services Web. En effet, pour des opérations de transfert répétitives d'un gros document XML sur le réseau, le SOAP peut causer un goulot d'étranglement. Le problème vient des fichiers XML qui génèrent beaucoup de codes supplémentaires par rapport à HTML, SOAP et WSDL basés sur XML. Les messages XML sont compilés au niveau client et au niveau serveur ce qui dégrade la qualité du service web en terme de temps de réponse.

(Tari et *al*, 2011) indiquent que le problème majeur ne vient pas seulement du SOAP, mais aussi du côté serveur. En effet les fichiers XML sont convertis en appel natif (C appels + / Java). Aussi, lorsque le serveur reçoit beaucoup d'appels en XML, il n'est pas en mesure de traiter la file d'attente, et des tâches risquent d'être ignorées. Une solution envisagée, pour améliorer les performances, est la réPLICATION du serveur pour l'équilibrage des charges. Cependant on ne connaît pas exactement le temps CPU alloué pour la requête SOAP du fait qu'elle nécessite de la compilation et de l'exécution au niveau du serveur. Même avec la croissance rapide de la bande passante, elle reste souvent limitée pour traiter des milliers de demandes de services web. Avec ces problèmes posés, (Tari et *al*, 2011) préconisent de bien concevoir les services web pour utiliser efficacement la bande passante disponible.

Cependant, (Tari *et al*, 2011) signalent qu'il y a des solutions proposées pour améliorer les performances au niveau SOAP, soient par l'utilisation de XML binaire au lieu de XML texte, la mise en cache des charges utiles pour SOAP sur le côté client (réutilisation des charges stockées pour créer une autre charge avec changement des paramètres demandées par le client), ou la compression des fichiers XML pour les clients qui sont facturés en volume et non en temps de connexion par leurs fournisseurs.

1.2.2.2 Entreprise Service Bus (ESB)

L'ESB est un type de middleware défini par Gartner Group comme permettant à des applications hétérogènes de fonctionner ensemble en utilisant les services Web, les connecteurs d'applications (JCA pour *J2EE Connector Architecture*) et le langage XML (Hugo, 2005).

Le Gartner Group définit l'ESB ainsi:

« An ESB is an architecture that exploits Web services, messaging middleware, intelligent routing, and transformation. It must support request/response communication between loosely coupled SOA business components and one-way message delivery for sending notifications to event-driven business components. It must also allow more-complex message exchange patterns (MEPs) ».

Selon (Colombe *et al*, 2005), avec l'introduction de la SOA et les services Web, la médiation de l'ESB a pris un nouveau concept et devenu un élément essentiel pour la SOA et les services Web. Les auteurs définissent l'ESB ainsi:

« A middleware providing integration facilities built on top of industrial standards such as XML, SOAP, WSDL, WS-Addressing, WS-Policy, WS-Security and WS-ReliableMessaging, J2EE Connector Architecture[?] ».

La Figure 1.12 illustre la médiation un élément essentiel dans l'ESB.

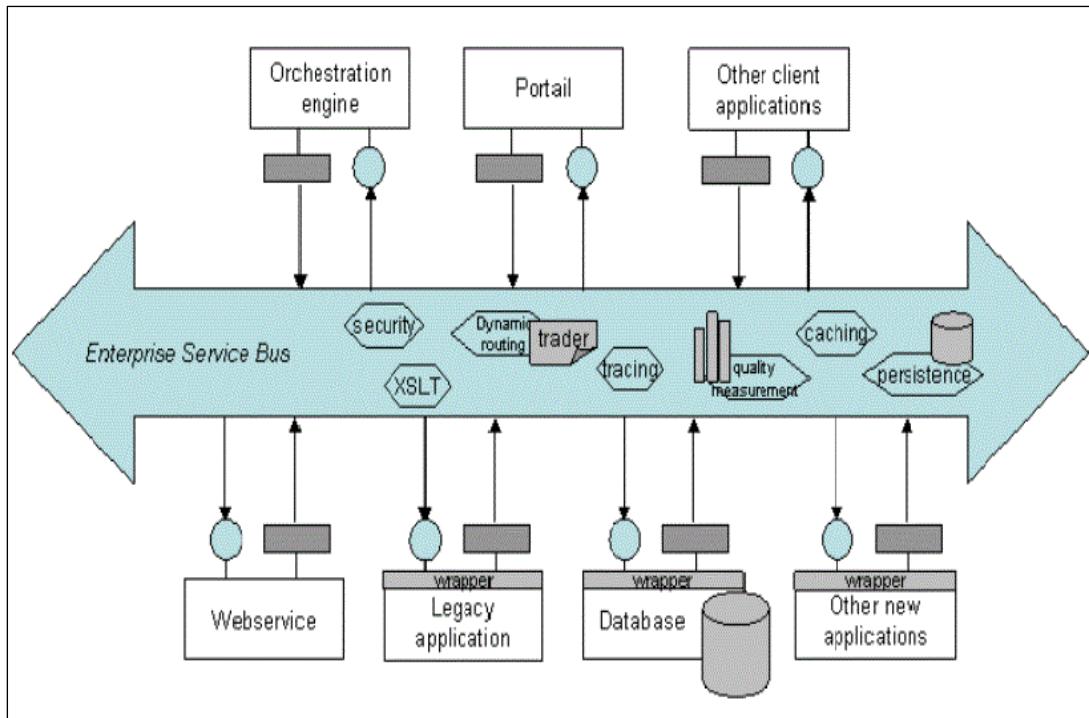


Figure 1.12 Enterprise Service Bus
Tirée de (Colombe *et al*, 2005, p.71)

Selon (Colombe *et al*, 2005) tel que montré dans la Figure 1.12, l'ESB est un middleware qui supporte plusieurs standards tels que XML, SOAP et WSDL, les spécifications associées aux services Web WS-* (*WS-Addressing*, *WS-Policy*, *WS-Security*, ...) compléments de SOAP et WSDL, et le *J2EE Connector Architecture* (JCA) pour la connection à des applications tierces. L'ESB offre aussi le service de communication essentiel pour l'échange de messages en mode asynchrone, le service orchestration pour l'assemblage ou l'agrégation de certains services, etc.

(Colombe *et al*, 2005) montrent qu'en plus des fonctionnalités de transformation, la médiation ESB inclue, comme illustré dans la Figure 1.12 :

La sécurité

Elle est assurée par des systèmes hétérogènes qui utilisent des sécurités différentes pour communiquer, telles que le cryptage des messages pour protéger les données et garantir la conformité, la prise en charge des authentifications personnalisées, etc ;

Le routage dynamique

le médiateur envoie les messages à plusieurs récepteurs soit pour équilibrer les charges ou bien en cas de défaillance d'une source de données ;

La qualité de services

Le médiateur assure d'autres actions non fonctionnelles qui se rapportent à la gestion de la qualité du service (*Quality of Service* -QoS) telles que la mesure de la qualité, la mise en cache, détection de défaillance et récupération, la gestion incomplète des données, etc.

(Colombe *et al*, 2005) distinguent deux approches d'implémentation du médiateur :

- implémentation ad hoc (propriétaire);
- implémentation services Web.

Pour (Bleakley *et al*, 2010), l'ESB pour une solution SOA joue le rôle d'intermédiaire entre les demandeurs et les fournisseurs de services. Toute communication passe par l'ESB qui se charge de la conversion du protocole de transport accepté par le fournisseur du service, du routage des messages entre les services, de la transformation des messages reçus dans un format accepté par la cible. Seul l'ESB connaît l'emplacement de chaque service. Ce découplage a des avantages tels que la réduction des interfaces et leurs complexités, la réduction des modifications apportées sur les applications pour des raisons de changement d'emplacement de services ou de formats, l'intégration entre les ressources disparates et la substitution d'un fournisseur par un autre sans aviser le consommateur du service. L'ESB gère aussi les événements d'affaires lorsqu'ils se produisent.

1.3 SOA et Entrepôt de données

Dans la littérature, certains auteurs ont parlé de l'intégration de l'entrepôt de données dans la SOA.

Selon (El akkaoui, 2009), l'outil ETL est une composante essentielle pour la consolidation des données dans un système décisionnel. La diversité des modèles d'outils ETL a posé un

problème de communication de ces outils avec d'autres modèles d'outils et auquel s'ajoutent la complexité et le coût.

Pour trouver une solution au problème de communication, (El akkoui, 2009) dans sa thèse propose une modélisation des processus ETL dans une architecture SOA. Ce modèle s'appuie sur le standard « *Business Process Model and Notation (BPMN)*» pour la modélisation conceptuelle des processus ETL. Ce modèle utilise aussi le « *Web Services Business Process Execution Language (WS-BPEL)*» un processus qui se base sur les opérations décrites dans le WSDL (les types de messages, les ports utilisés, les opérations, etc.) pour spécifier les interactions entre les différents services Web. Ce processus est décrit dans un document en XML et exécuté par un moteur d'orchestration.

Les avantages de cette modélisation (El akkaoui , 2009) sont de masquer les détails techniques de chaque processus ETL, d'utiliser un langage exécutable standard et de transformer les processus ETL en des processus d'affaires gérés par les utilisateurs finaux.

Sur un autre plan de recherche, (Ronnie, 2008) présente un prototype sur l'intégration de l'entrepôt de données quasi temps réel dans la SOA et ce pour le partage des connaissances stratégiques à travers une grande distribution.

Dans l'architecture proposée (*Voir Figure 1.13*), (Ronnie, 2008) utilise l'entrepôt de données et l'entrepôt de données quasi temps réel comme sources de données pour la SOA. L'entrepôt de données quasi temps réel est alimenté par un processus cycle de chargement des données « *Change data capture- CDC*» qui fait une capture des données modifiées dans les sources de données opérationnelles à partir d'un fichier LOG. L'objectif est de minimiser les ressources requises par l'ETL. L'entrepôt de données est mis à jour par l'entrepôt de données quasi temps réel à une fréquence déterminée. L'architecture SOA et l'architecture entrepôt de données communiquent via le middleware ESB.

Pour le développement du prototype, (Ronnie, 2008) a choisi l'Oracle 10g AS une plateforme qui utilise une architecture informatique orientée services s'appuyant sur des standards.

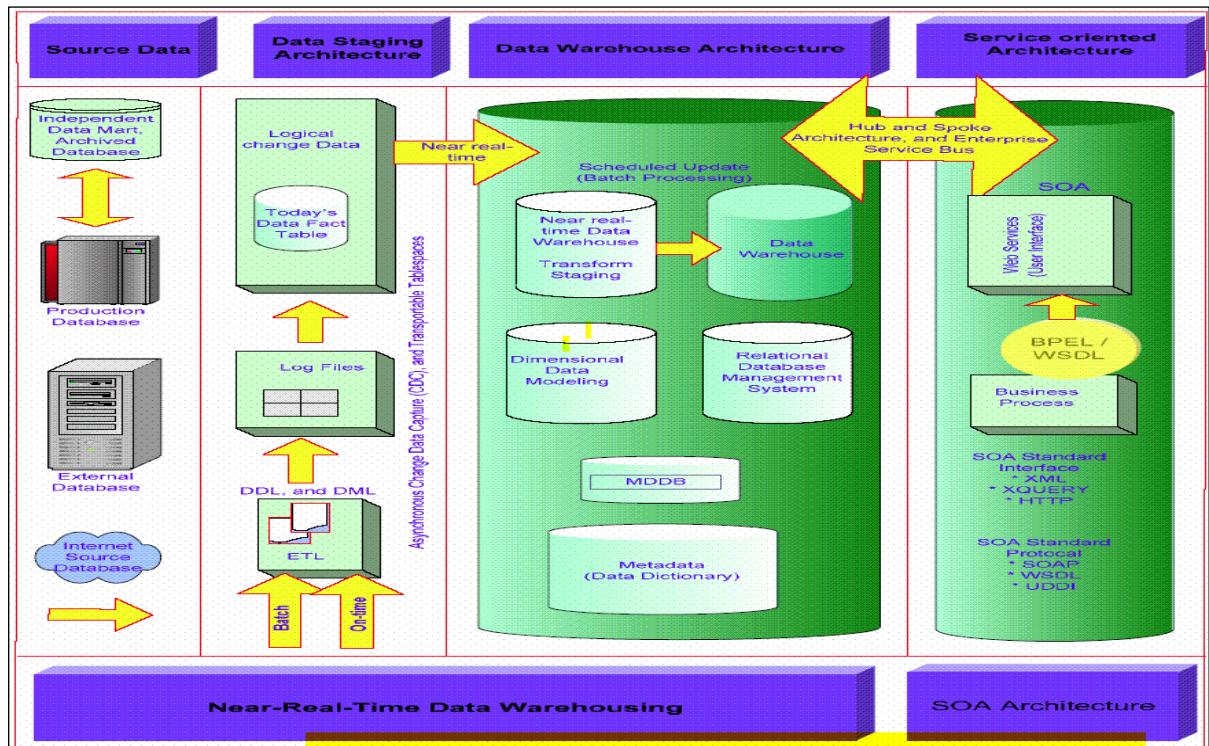


Figure 1.13 Modèle de processus pour un entrepôt de données dans l'architecture SOA
Tiré de (Ronnie, 2008, p.30)

1.4 Pourquoi combiner les entrepôts de données et la SOA

Dans la revue de la littérature, au niveau de l'architecture d'entrepôt de données, les travaux abordés portent essentiellement sur des aspects techniques tels que l'utilisation de l'outil ETL, la modélisation de l'entrepôt de données et les outils d'analyse et d'exploration des données. Certes, cette architecture dotée d'une base de connaissances avec la multiplicité des outils d'exploitations des données est l'une des stratégies de l'entreprise pour lui donner plus d'autonomie pour exprimer ses besoins d'analyses et déployer ses résultats. Cependant, des problèmes fondamentaux surgissent tels que :

- le dialogue de l'entreprise avec ses partenaires sans le WEB ;
- la généralisation de l'utilisation de l'architecture par tous les utilisateurs indépendamment de leurs emplacements géographiques.

Une solution à ces problèmes est l'architecture orientée services (SOA) mais tout en résolvant ses insuffisances soulevées par (Inmon, 2006).

Dans son journal, (Inmon, 2006) détaille les insuffisances des SOA classiques. En effet, la SOA est une interface qui lie les systèmes organisationnels de l'entreprise et le Web, mais elle pose des problèmes qui ne sont pas traités tels que:

La performance

Il s'agit du temps mis pour accéder aux données dans le système organisationnel de l'entreprise. Une requête qui dure plus d'une minute, pour un utilisateur final exigeant une meilleure performance, n'est pas acceptable. Ce problème est dû à diverses sources de données et à la requête qui ne sont pas soigneusement disposées. Une requête lancée dans une transaction pour faire une analyse détaillée sur une période donnée n'aura pas un temps de réponse rapide.

L'intégration des données

Par exemple, pour voir les recettes des magasins, l'interface SOA cherchera l'information dans trois bases de données différentes, puis retourne l'information avec un chiffre de 4,977,330.76 dollars. Mais en pratique ce chiffre peut être composé de dollars canadiens de la première base, de dollars australiens de la deuxième base, puis de dollars américains pour la troisième base et le résultat ramené donne un chiffre non significatif. Ceci exige bien une intégration des données avant l'exploitation.

L'accès aux données historiques

La SOA accède aux applications qui n'ont pas de données historiques auxquelles elles sont attachées. Pour rendre la SOA plus efficace, il faut avoir une base de données historisées qui ne sont pas attachées aux applications.

La cohérence des données

Un processus SOA qui accède par exemple à trois bases de données (A, B, C) a ramené un chiffre de 4.760.669 \$. Pour vérifier ce résultat, un autre processus SOA a accédé à d'autres bases de données (D, E, F) et a ramené un résultat de 5.339.817 \$. Ces deux résultats ramenés laissent un utilisateur final indécis. La SOA met les données à la disposition du consommateur de services, mais ne garantit pas la cohérence des données;

Le référentiel de métadonnées

Pour une bonne interprétation des données existantes, certaines analyses nécessitent des informations sur les données et doivent se référer donc au référentiel de métadonnées. Ce référentiel, par son existence, appuie fortement la SOA.

La réutilisation d'une transaction

Pour un nouveau besoin d'analyse, on ne peut pas réutiliser une partie ou totalement une transaction existante dans la SOA. Généralement, le nouveau besoin est construit à partir de zéro.

Pour répondre à ces insuffisances, (Inmon, 2006) exige que la SOA ait une base de connaissances où on trouve les données intégrées et historisées fondées sur le besoin de performance, les données granulaires et les données agrégées pour une orientation métier, et un référentiel de métadonnées. Cette base de connaissances est la base idéale pour tous les traitements de la SOA.

1.5 Sommaire du chapitre

Ce chapitre a présenté une étude bibliographique sur les architectures entrepôts de données et les architectures orientée services. Des définitions, des modèles d'architectures et des méthodes adoptées pour le développement des dites architectures ont été présentées avec des évaluations critiques de certains auteurs. Cet état de l'art nous a mené par la suite à dégager les insuffisances de ces architectures pour répondre aux besoins posés par les entreprises qui veulent avoir une base de connaissances pour agir efficacement, rentabiliser ses coûts d'investissements et s'ouvrir à l'extérieur.

L'intégration de l'entrepôt de données dans une architecture SOA est une proposition de solution qui permettrait à la SOA de tirer profit de la croissance continue des données dans l'entreprise, en valorisant notamment les gisements de données historiques. Le problème consiste à intégrer l'entrepôt de données dans la SOA.

Le chapitre 2, introduit la problématique de la recherche pour la proposition du projet d'intégration de l'entrepôt de données dans la SOA, son but et ses objectifs en incluant les étapes de la recherche, les limites et la méthodologie.

CHAPITRE 2

LA PROBLÉMATIQUE DE LA RECHERCHE

Ce chapitre décrit les aspects méthodologiques de cette recherche. La première section présente le but et les objectifs de la recherche. La deuxième section traite des limites de la recherche. La troisième section décrit la méthodologie de recherche appliquée pour atteindre les objectifs du projet.

Dans le chapitre 1, on a vu que l'architecture orientée services pose des problèmes d'intégrité, de cohérence des données et de performance. L'architecture d'entrepôt de données pose des problèmes de réutilisation des applications, le manque d'ouverture sur l'extérieur et le coût d'investissement.

2.1 **But et objectifs de la recherche**

Le but est de :

« Concevoir une intégration de l'entrepôt de données dans une architecture SOA en tant que référentiels de données basé sur des services métiers réutilisables et faiblement couplés pour l'intégration des données, l'analyse et l'exploitation des données en utilisant des technologies standards pour faciliter la maintenance et l'évolution».

Pour atteindre ce but, l'objectif de la recherche est d'intégrer l'entrepôt de données dans une architecture SOA pour augmenter le potentiel d'utilisation des informations qui sont disponibles au sein de l'entreprise. Cette intégration s'appuie sur les services Web qui utilisent des technologies standards indépendantes du matériel et du langage de programmation, et sur l'ESB qui constitue un soutien fondamental pour les services Web au niveau routage comme il contrôle les messages et minimise les risques d'indisponibilité des données ou des pannes. Cette conception visera à réduire le coût d'investissement en transformant les applications en des processus métiers agiles et évolutifs qui pointent sur des

services réutilisables tels que les services d'intégration des données (ETL), des services d'analyse et d'exploitation (OLAP), etc.

2.2 Limites de la recherche

Cette recherche ne couvrira pas la réalisation de la solution proposée: seul un exemple de fonctionnement d'un service tableau de bord sera présenté.

2.3 Méthodologie de la recherche

La méthodologie de la recherche (Figure 2.1) est conçue pour atteindre les objectifs de la recherche et inclue les étapes suivantes:

- élaborer l'état de l'art de l'architecture SOA et de l'architecture de l'entrepôt de données (chapitre 1);
- définir les objectifs de la recherche (chapitre 2);
- intégrer l'architecture des entrepôts de données dans l'architecture SOA (chapitre 3) ;
- élaborer un exemple d'utilisation de la SODWA ;
- évaluer de façon préliminaire et partielle l'architecture logicielle en terme de qualité.

La Figure 2.1 résume les principales étapes de la méthodologie de recherche.

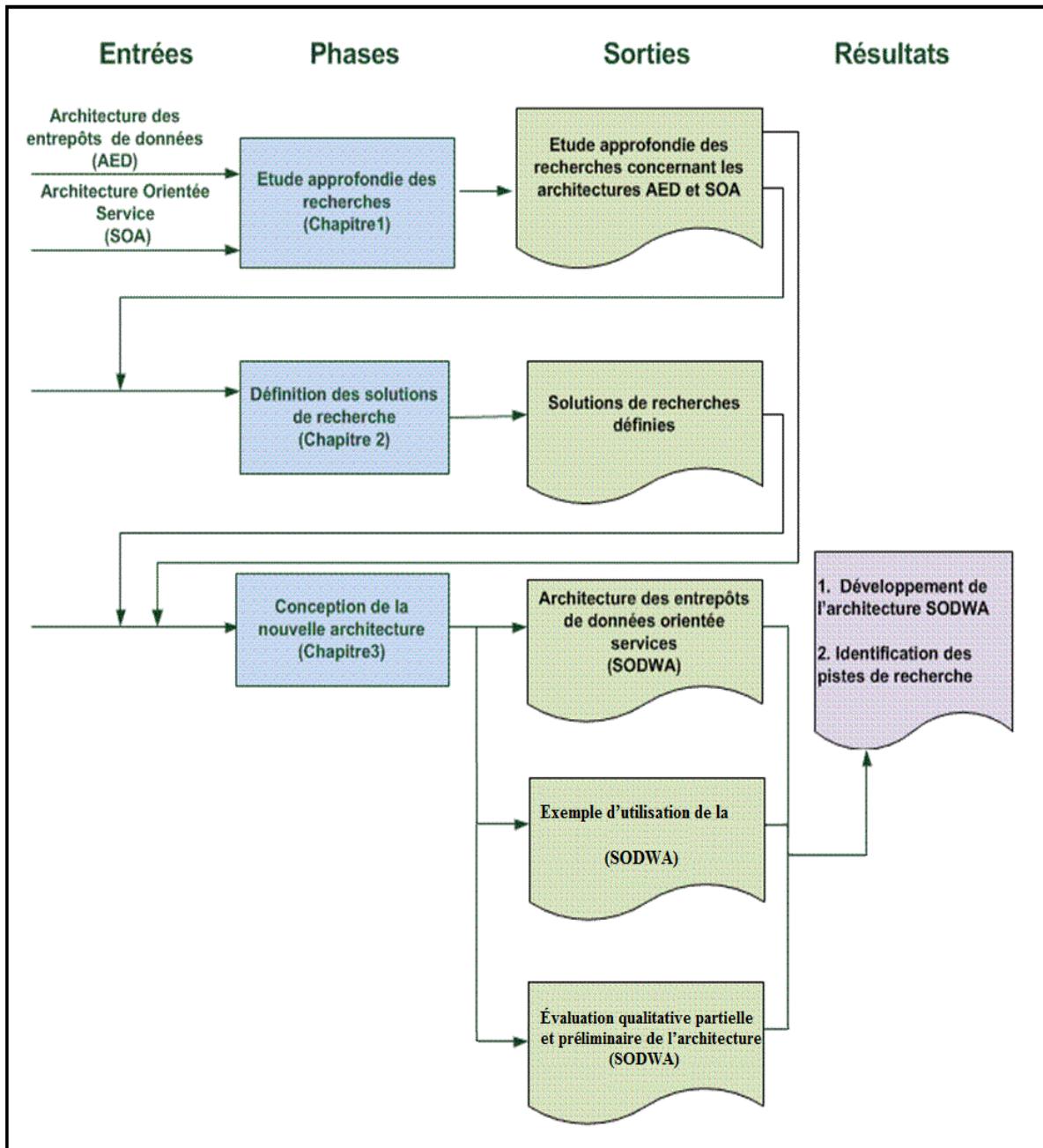


Figure 2.1 Méthodologie de recherche

2.4 Sommaire du chapitre

La solution proposée combinerà la SOA et l'entrepôt de données pour donner naissance à une architecture d'entrepôt de données orientée service. Pour atteindre cet objectif, une méthodologie de recherche a été adoptée pour décrire les principales étapes de ce projet.

Le chapitre 3 présentera la SODWA proposée et décrira les concepts clés, une vue d'ensemble, une vue détaillée avec les justifications, un exemple d'utilisation et une évaluation préliminaire et qualitative de la SODWA

CHAPITRE 3

ENTREPÔT DE DONNÉES ORIENTÉ SERVICES

Ce chapitre présente les concepts clés, une vue détaillée, les justifications et les spécifications de l'entrepôt de données orienté services (SODWA pour *Service-Oriented DataWahouse Architecture*). La première section décrit les concepts clés de la SODWA. La deuxième section introduit les justifications de la SODWA. La troisième section présente une vue détaillée de la SODWA. La quatrième section présente un exemple d'utilisation de la SODWA. La cinquième section présente une évaluation qualitative partielle et préliminaire de la SODWA.

3.1 Les concepts clés de la SODWA

Les concepts clés de la SODWA (*Voir Figure 3.1*) sont:

- les données opérationnelles ;
- l'entrepôt de données ;
- les services ;
- l'analyse et exploitation des données.

La Figure 3.1 montre une vue globale de la SODWA.

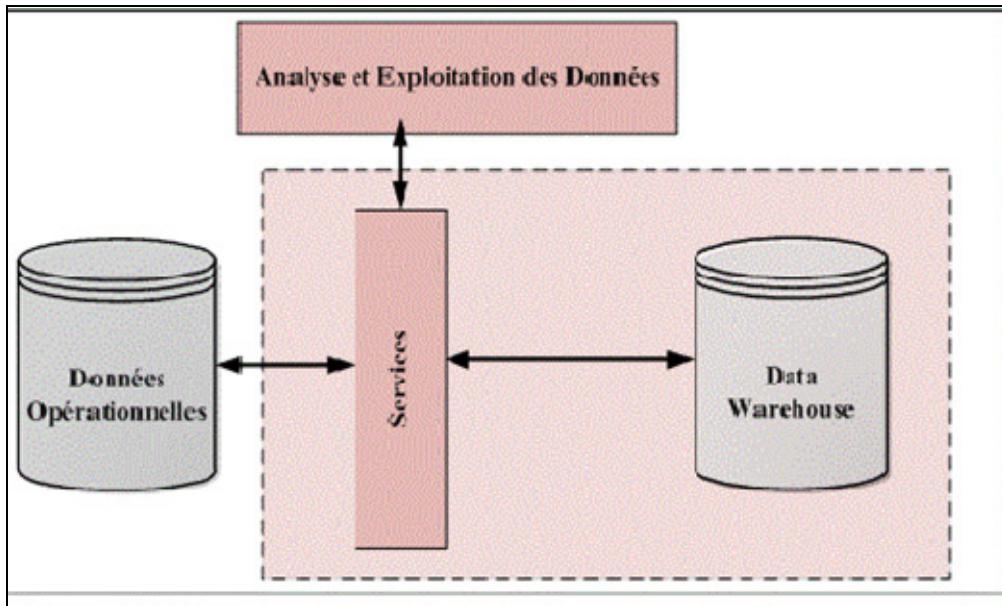


Figure 3.1 Vue Globale de la SODWA

Données Opérationnelles

Ce sont des données orientées applications, détaillées, hétérogènes et dont la mise à jour est interactive. Les données peuvent être sous formats fichiers plats, fichiers Excel, bases de données, services Web, etc.

Entrepôt de données

L'entrepôt de données est une base de données dédiée au stockage de l'ensemble des données collectées de manière homogène à partir de différentes sources de données. Les données de cette base sont organisées et conçues de façon à permettre aux utilisateurs un accès rapide et d'avoir une vision claire pour la prise de décision.

Services

Un service est un processus métier qui peut être utilisé de façon statique ou dynamique pour la réalisation d'une application logicielle. Les fonctionnalités du service sont fournies dans une description pour la recherche et l'invocation du service.

Analyse et exploitation des données

L'analyse des données est la modélisation des données selon plusieurs axes pour apporter des réponses aux questions clés, comprendre les tendances et aider à prendre des décisions.

L'exploitation des données est la modélisation d'une représentation basée sur une requête simple ou complexe pour l'élaboration d'un tableau de bord, d'un document corporatif à des fins statistiques, etc.

3.2 Justifications de la SODWA

Les justifications de la SODWA sont les suivantes:

- la SODWA va hériter des concepts d'ETL tels que l'extraction, la transformation et le chargement des données. Ici, il s'agit de routines qui sont essentiellement les mêmes dans les produits d'ETL ;
- la SODWA va hériter du concept d'intégration de données par lots en utilisant les concepts d'ETL, contrairement à la SOA qui n'intègre que des données transactionnelles ;
- la SODWA va hériter des concepts de l'analyse et l'exploitation des données appelé OLAP qui est essentiellement utilisé dans les produits d'analyses de l'entrepôt de données ;
- la SODWA va hériter des concepts de composition de services, l'abstraction de l'infrastructure, le couplage lâche qui assure l'adaptation aux changements, la sécurité, etc. ;
- tous les concepts hérités seront considérés comme des composants réutilisables dans la SODWA ;
- la SODWA va hériter des concepts de communication tels que les protocoles standards de messagerie basés sur le langage XML, le SOAP (pour Simple Object Access Protocol), etc. Ce principe facilite l'échange d'informations dans la SODWA.

3.3 Vue détaillée de la SODWA

La SODWA proposée (*Voir Figure 3.2*) est l'intégration de l'entrepôts de données dans l'architecture orientée services SOA. La SODWA est composée de quatre couches: la couche « référentiels métiers », la couche « services métiers », la couche « applications et logiciels », et la couche « infrastructure physique ».

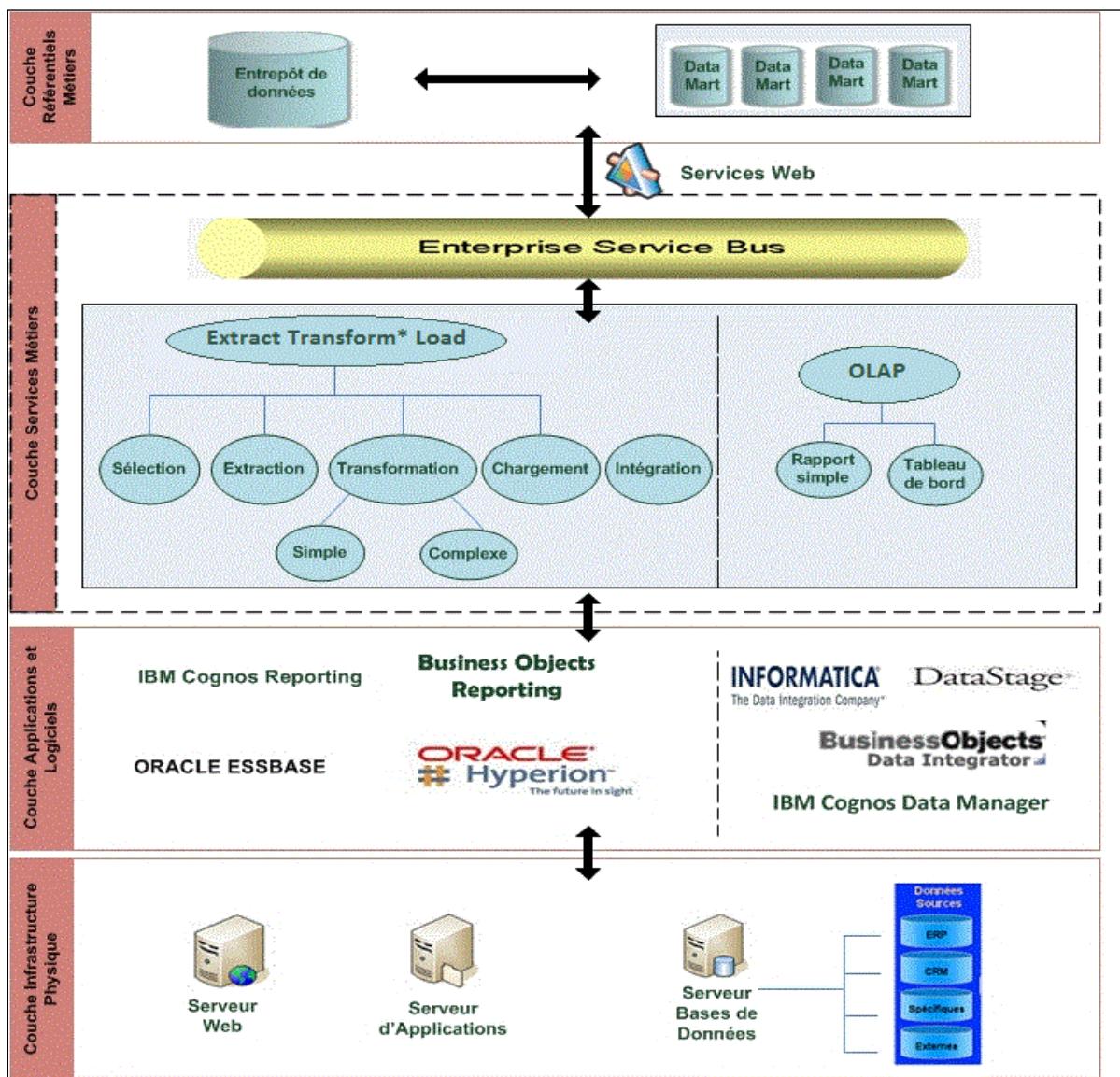


Figure 3.2 Vue Détailée de la SODWA

La SODWA proposée (*Voir Figure 3.2*) inclut ce qui suit:

La couche « référentiels métiers »

Cette couche est formée des deux composants suivants :

- **l'entrepôt de données**

L'entrepôt de données pour la SODWA proposée est un modèle de données normalisées et intégrées dans un souci de rationalisation. Les données sont non volatiles, historisées et datées. La modélisation de ces données est effectuée selon l'approche de (Inmon, 2005). Cette modélisation assure la cohérence des données, donne une solution technique solide et consistante (chapitre1, section 1.1.2);

- **les « Data Marts »**

Dans la SODWA proposée, les «*Data Marts* » sont considérés comme des sous-ensembles de l'entrepôt de données. L'implémentation des «*Data Marts* » est physique. Le chargement des données dans les «*Data Marts* » se fait uniquement à partir de l'entrepôt de données centralisé. Chaque «*Data Mart* » est orienté sujet et optimisé pour des besoins spécifiques à des fins d'analyse statistiques.

Les services Web

Les services Web facilitent l'implémentation et la mise en œuvre de la SODWA proposée. Ces services Web ont pour rôle l'intégration de l'information. Ils constituent une passerelle vers une architecture orientée services pour l'intégration des données de l'entreprise. Avec les services Web, on peut extraire les données et les intégrer, faire communiquer les applications entre elles tout en étant indépendant du langage de la plateforme et du langage de la programmation grâce au langage XML utilisé. En outre, avec les services Web, les applications et les logiciels dans la couche SODWA peuvent communiquer et échanger des données entre eux indépendamment de leurs systèmes d'exploitation ou de leurs évolutions dans d'autres systèmes ce qui assure la réutilisabilité.

La couche services métiers

- *Enterprise Service Bus – ESB*

L'ESB, dans la SODWA, est une plateforme qui inclut les services de plusieurs types. Chaque type de service spécifie l'interface, la sémantique de traitement, l'instance, le protocole, les transformations, le routage. Le type de services est assimilé à une classe de services qui ont un comportement commun. L'exécution d'un service se passe dans un conteneur de services. Le conteneur héberge les services fournies par les applications ce qui garantit que l'application n'est pas enfermée dans un seul et même fournisseur. Chaque conteneur peut exécuter plusieurs services. Pour l'exécution du service, le conteneur a besoin de connaître le type de service. Les services sont répertoriés dans un registre de services sur le bus et accessibles par le consommateur de services.

L'ESB intègre le concept de médiation pour faciliter la conception d'applications basées sur les services Web. La médiation est assurée par la plate-forme ESB, qui est l'intermédiaire entre le fournisseur de services et le consommateur. Dépourvu de cette médiation, le consommateur du service doit connaître l'adresse du service pour y accéder directement. Avec l'ESB, le consommateur délègue l'invocation du service à l'ESB qui se charge de la manipulation, de l'acheminement et de la sécurité des messages. Avec cette médiation, l'entreprise a une vision globale de toutes ses activités et un système d'information facilement utilisable renforcé par l'agilité du système.

Plusieurs ESB propriétaires ont été développés et proposés par les fournisseurs d'architecture orientée services telles que *WebSphere Enterprise Service Bus* d'IBM, *Oracle Service Bus* d'Oracle. Différentes techniques et outils ont été proposés par les fournisseurs pour implémenter et déployer les médiateurs au sein de l'ESB.

- ETL

Dans la SODWA proposée, l'ETL est un service formé d'un ensemble de services. Les services proposés par l'ETL sont les suivants:

- service Sélection

Il filtre les données en lecture afin de n'extraire que l'information pertinente à partir des sources de données ;

- service Extraction

Il permet d'extraire les données pertinentes de différentes sources de données de production de l'entreprise tels que les différentes types bases de données, fichiers textes, fichiers Excel, fichiers XML;

- service Transformation

Il transforme les données de type simple ou complexe :

- ✓ simple

Ce service permet de vérifier les données en termes de formats, types et longueurs pour une représentation unique dans la base de données, de décoder les champs pour les unifier en remplaçant par exemple ‘Féminin’ par ‘F’ et ‘Masculin’ par ‘M’, le format d’une date extraite ‘2013/12/11’ transformé en ’11-12-2013’, et de contrôler qu’une donnée est cohérente par rapport à son historique existant dans la base cible.

- ✓ complexe

Ce service permet de découper les champs complexes à partir de chaînes de caractères telles que la séparation du nom et du prénom dans une chaîne de caractères nom et prénom, la fusion de données par exemple le produit d’une quantité et son prix pour donner le total de vente du produit, la duplication des enregistrements concernant un client par exemple, le pré-calcul des agrégations tels que le calcul des ventes par secteur et par mois. Ce service permet aussi d’éliminer les valeurs de données non conformes au modèle de destination et les doublons.

- service Chargement

Le service chargement stocke les informations de façon correcte dans l'entrepôt de données et les « *Data Marts* ». Ce service insère les données dans les tables cibles (dimensions et faits);

- service OLAP (Online Analytical Processing)

Ce service permet aux utilisateurs un accès rapide et intuitif aux données centralisées simples ou agrégées liées à des fins d'analyse et de documents corporatifs « *Reporting* ». La caractéristique principale du service OLAP est de permettre aux personnes concepteurs de rapports de créer leurs propres rapports simples et / ou les tableaux de bord à partir des données en format XML ;

Ce service est composé de deux services :

- ✓ Service rapport simple

Un rapport simple est un ensemble de résultats d'indicateurs construit généralement de façon périodique (quotidien, mensuel, trimestriel,...). Généralement basé à partir d'une seule source de données présentées dans un format tabulaire (ligne, colonnes).

Au niveau de la SODWA, le service rapport simple génère les résultats de la requête et l'envoie au service ESB qui se charge de l'envoyer au service appelant.

Product line	Quantity
Personal Accessories	389,908
Mountaineering Equipment	301,958
Golf Equipment	99,400
Camping Equipment	866,234
Outdoor Protection	557,854
Summary	2,215,354

Figure 3.3 Rapport simple
Tirée de IBM (2012, p. 22)

La Figure 3.3 présente un rapport simple de type liste généré à partir du produit « *IBM Cognos Query Studio* ». Dans ce rapport on peut appliquer aussi un filtre (parfois connu sous le nom « *Condition* »), un récapitulatif, ou des calculs pour manipuler les données ;

✓ Service tableaux de bord

Pour améliorer la compétitivité et la productivité d'une entreprise, le tableau de bord est un moyen indispensable pour disposer d'une vue rapide, synthétique et pertinente sur la performance d'une activité. Les tableaux de bord permettent aussi de voir si les objectifs stratégiques préalablement définis sont atteints. Donc une véritable réflexion est menée en amont sur des indicateurs clés afin de déterminer les variables d'action. Le soin apporté à son élaboration conditionne la réussite globale du projet décisionnel d'entreprise. Dans la pratique, les tableaux de bord sont généralement un mélange de tableaux de données et graphiques. Ils se tiennent généralement sur une page ou sur un seul écran. Les requêtes sont statiques et prédefinies pour les indicateurs à suivre et générées à partir des « *Data Marts* », ou de l'entrepôt de données.

Au niveau de la SODWA, le service tableaux de bord génère les résultats de la requête et l'envoie au service ESB qui se charge de l'envoyer au service appelant.

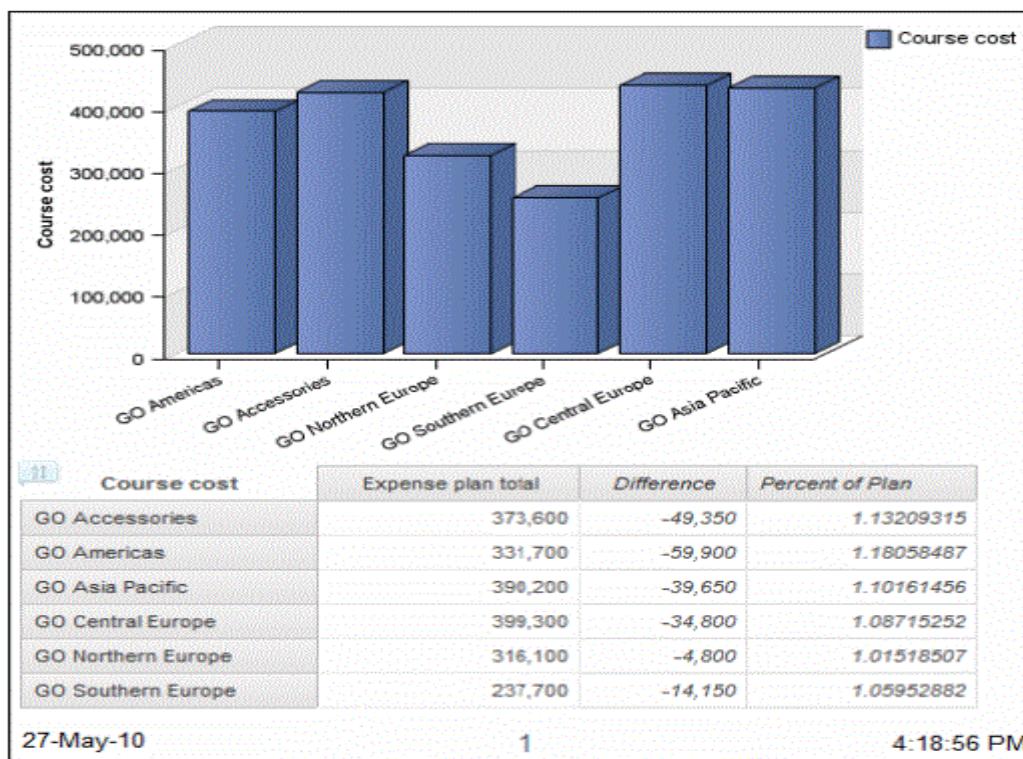


Figure 3.4 Formation des employés par niveau d'organisation
Tirée de IBM (2010, p. 35)

La Figure 3.4 montre un tableau de bord créé avec le produit Web « *IBM Cognos Business Insight* » avec une présentation tabulaire et graphique pour faciliter la prise de décision. Un exemple de service tableau de bord est fourni à la section 3.4.

La couche applications et logicielles

Dans cette couche on trouve les fournisseurs d'application logiciels de l'ETL et les fournisseurs d'applications d'analyses et de rapports. Les composants de cette couche sont gérés par l'ESB pour les faire communiquer ou simplement appeler un composant

La couche infrastructure physique

- **serveur Web**

Le serveur Web est un logiciel qui implémente le protocole HTTP et à l'écoute aux connections provenant de l'extérieur sur un port donné. L'internaute envoie sa requête au serveur qui à son tour lui transmet sa réponse soit dans une page HTML, soit dans un fichier selon le type du service rendu. Comme serveur Web, on peut citer *Apache HTTP Server*, *Apache Software*, *Internet Information Services* (IIS) de Microsoft. ;

- **serveur d'applications**

Le serveur d'applications est une machine de haute performance qui offre le déploiement et l'exécution des applications dans un environnement intégré sous sa plateforme (linux, Windows,...) tout en étant indépendante de la configuration du poste client. Ces applications répondent aux demandes des utilisateurs transmises par réseau depuis les postes clients ou d'autres applications.

Le serveur d'applications assure la répartition des charges, la reprise en cas d'incidents par la réplication de l'application sur plusieurs serveurs physiques, le *pooling* de connexions pour éviter le goulet d'étranglement sur les bases de données et ce par la répartition des différentes requêtes sur les connexions disponibles, la sécurité (cryptage, *Lightweight Directory Access Protocol* (LDAP), etc.), l'administration telles que la configuration de répartition des charges, la configuration des fichiers logs, etc.

Comme serveurs d'applications les plus connus, on peut citer *WebLogic Server* d'Oracle, *ASP.Net* de Microsoft, *WebSphere* d'IBM, etc. ;

- **serveur bases de données**

Le serveur de bases de données est un serveur qui héberge les systèmes de gestion de bases de données et les données et accessibles par le réseau pour l'ensemble des

utilisateurs. La connexion à la base de données identifie l'utilisateur. L'accès aux données est contrôlé en attribuant des rôles prédéfinis aux utilisateurs ou au groupe d'utilisateurs tels que lecture, lecture et écriture, administrateur, administrateur du serveur.

Les serveurs d'applications ou les serveurs Web s'interfacent avec le serveur de bases de données pour l'envoi et la réception de la requête du client. Comme serveurs de bases de données les plus connues, on peut citer Database 11g d'Oracle, *SQL Server 2010* de Microsoft, *DB2 Express-C* d'IBM, etc.

3.4 Exemple d'utilisation de la SODWA

Pour illustrer le fonctionnement général de la SODWA, voici un exemple de service métier lancé par un utilisateur (consommateur de services) pour la création d'un tableau de bord:

- l'utilisateur, à travers son portail, lance un service métier pour la création d'un tableau de bord par exemple : les ventes d'un produit;
- le service ESB, le médiateur entre le consommateur de services et le fournisseur de services, prend en charge ce service métier pour vérifier la sémantique de traitement, l'instance, le protocole, les transformations et le routage.

Disposant de ses informations, l'ESB va s'occuper de l'acheminement de chaque service appelé et les interactions entre les services en jeu. La communication des services se fait à travers des paramètres d'entrées et de sorties.

Voici les étapes de fonctionnement de la SODWA pour l'élaboration d'un tableau de bord sollicité par un consommateur de service. L'outil utilisé pour générer le tableau est *IBM Cognos Reporting*.

L'ESB expose les services dans un annuaire ou registre de services. Le consommateur de services recherche le service fournisseur « Tableau de bord » sur les ventes d'un produit dans l'annuaire, puis appelle le service (Voir Figure 3.5). Les paramètres d'appel tels que le protocole ou le format de données par le service consommateur sont indépendants de ceux du service fournisseur

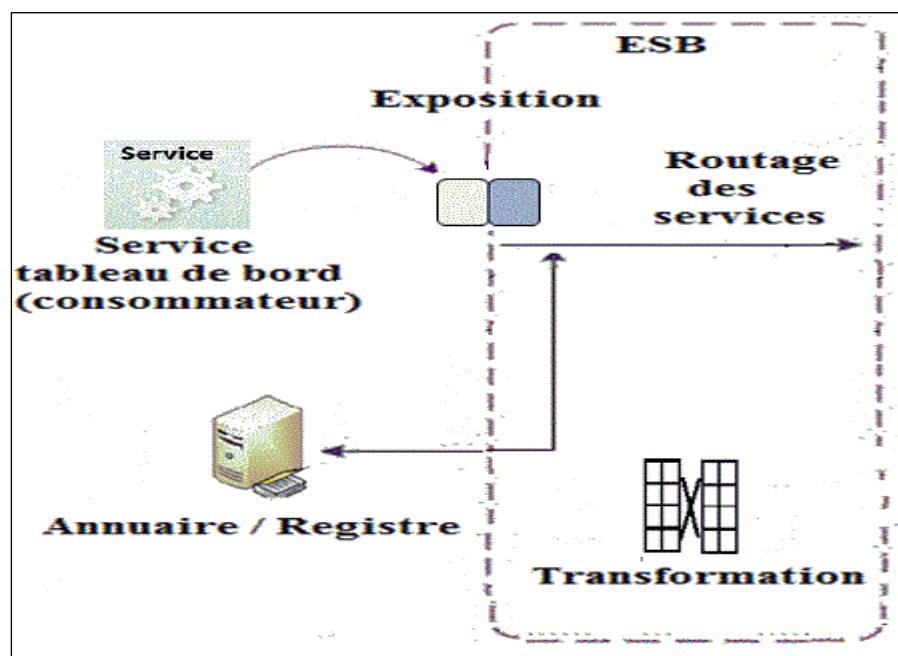


Figure 3.5 Exposition des services par l'ESB

L'ESB, moyennant le routage des services, va invoquer dans l'ordre, le service sélection, le service extraction, le service chargement, le service OLAP, puis le service tableau de bord. A chaque invocation du service, l'ESB consulte l'annuaire de services pour vérifier son existence.

- 1) Pour l'invocation du service fournisseur « Sélection » (Voir Figure 3.6), l'ESB réalise une transformation de format de données reçues par le service consommateur selon le format de données pris en charge par le service fournisseur. Ce service exécute une requête de sélection des données dans le « Data Mart » « Ventes produits », puis retourne à l'ESB les données sélectionnées.

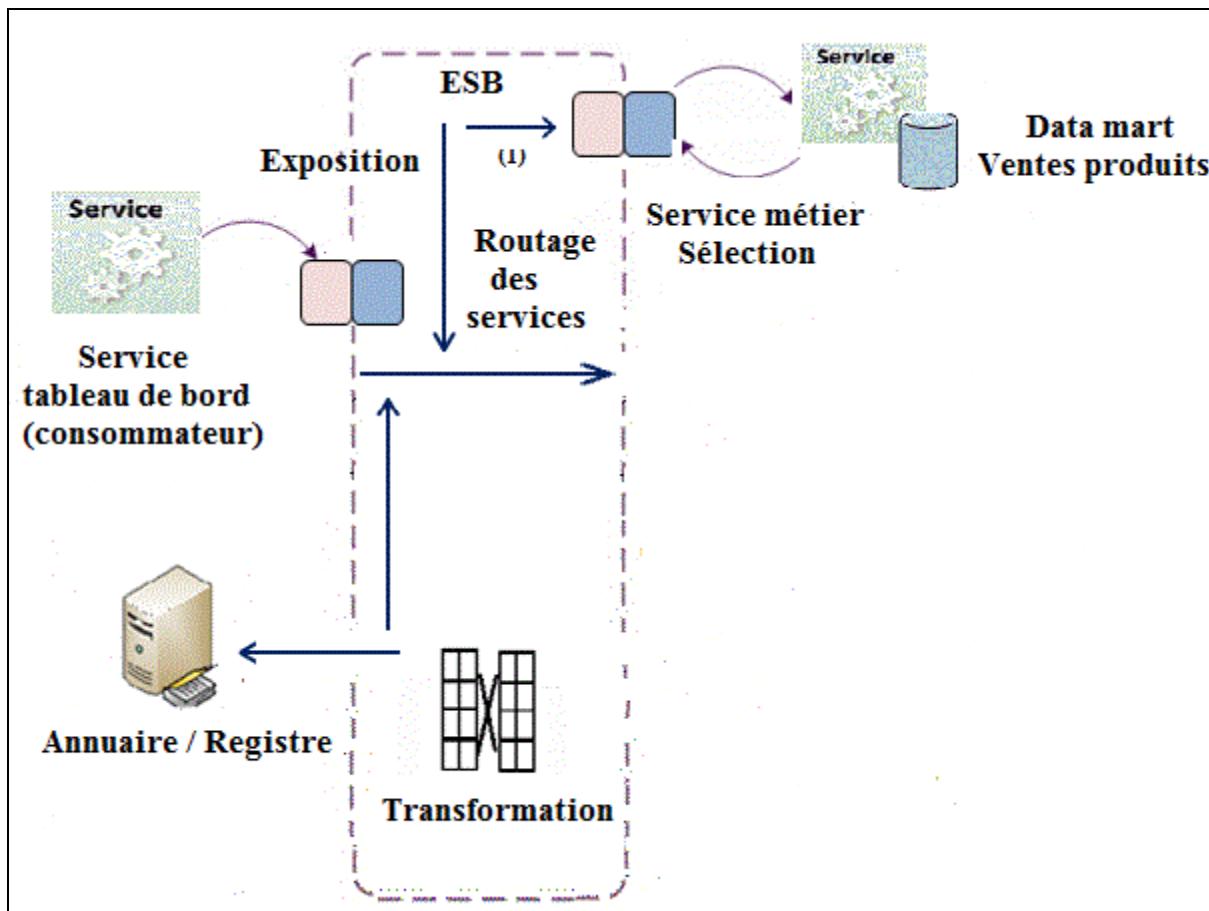


Figure 3.6 Invocation du service métier sélection par l'ESB

- 2) Pour invoquer le service fournisseur « Extraction » (Voir Figure 3.7), l'ESB réalise une transformation de format de données reçues par le service « Selection » pour les communiquer au service fournisseur selon le format des paramètres d'entrées pris en charge. Ce service extrait les données à partir du « Data Mart », puis retourne à l'ESB les données extraites.

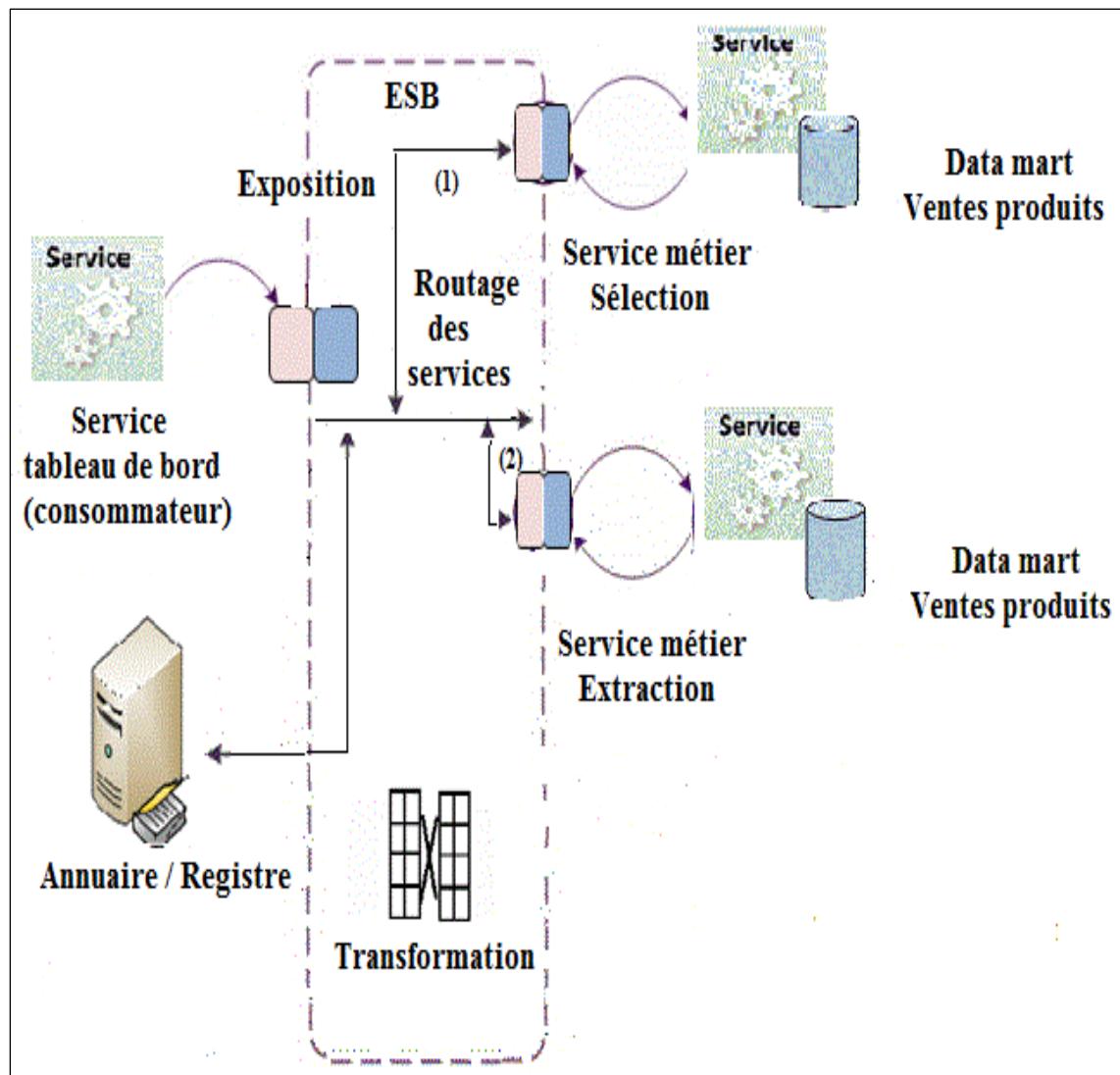


Figure 3.7 Invocation du service métier extraction par l'ESB

- 3) Pour invoquer le service fournisseur «Chargement» (Voir Figure 3.8), l'ESB réalise une transformation de format de données sur les données reçues par le service « Extraction », puis les communiquent au service fournisseur. À partir de ces données, le service génère une vue multidimensionnelle qu'il fait retourner par la suite à l'ESB.

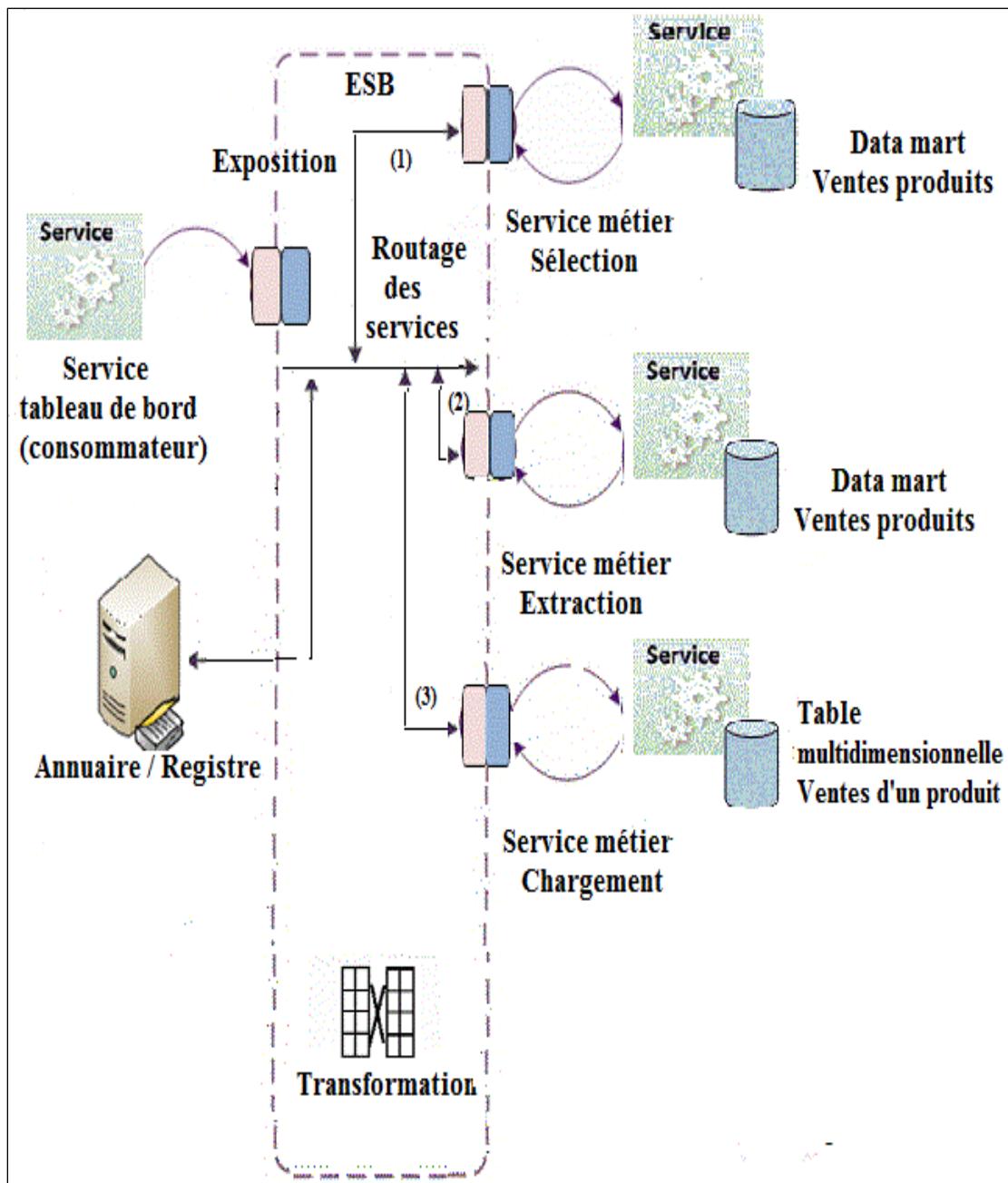


Figure 3.8 Invocation du service métier chargement par l'ESB

- 4) Pour invoquer le service fournisseur «OLAP», l'ESB réalise une transformation de format de données de la vue multidimensionnelle reçues par le service « chargement » puis les communiquent au service fournisseur. Ce service génère le cube OLAP et le fait retourner à l'ESB.

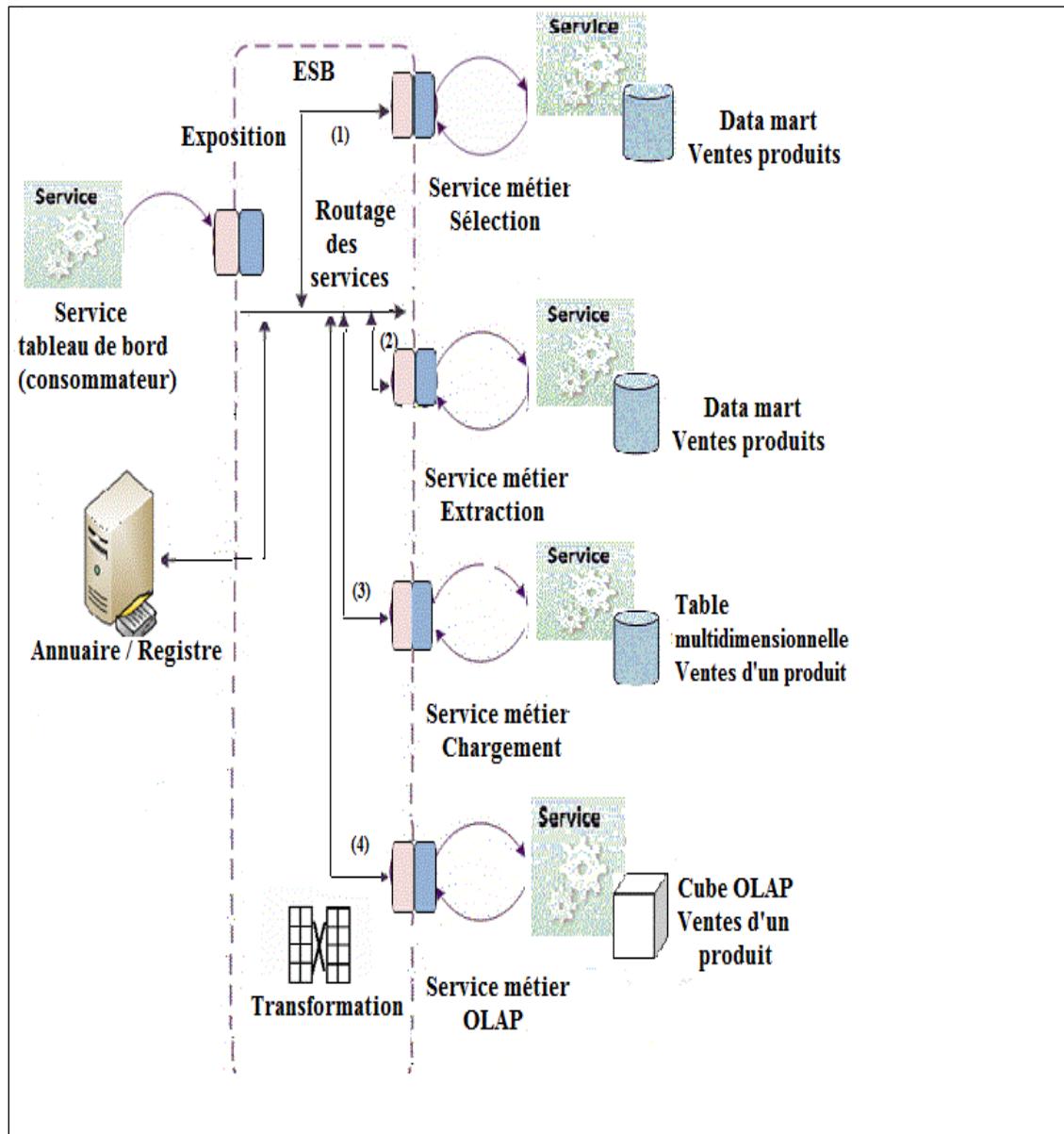


Figure 3.9 Invocation du service métier OLAP par l'ESB

- 5) Pour invoquer le service fournisseur «tableau de bord», l'ESB réalise une transformation de format de données sur le cube OLAP reçu par le service « OLAP », puis les communiquent au service fournisseur. Ce service fournisseur génère le tableau de bord et le fait retourner à l'ESB.

- 6) l'ESB retourne les données générées par le service « tableau de bord » vente d'un produit selon le format de données et le protocole du consommateur du service initial (utilisateur).

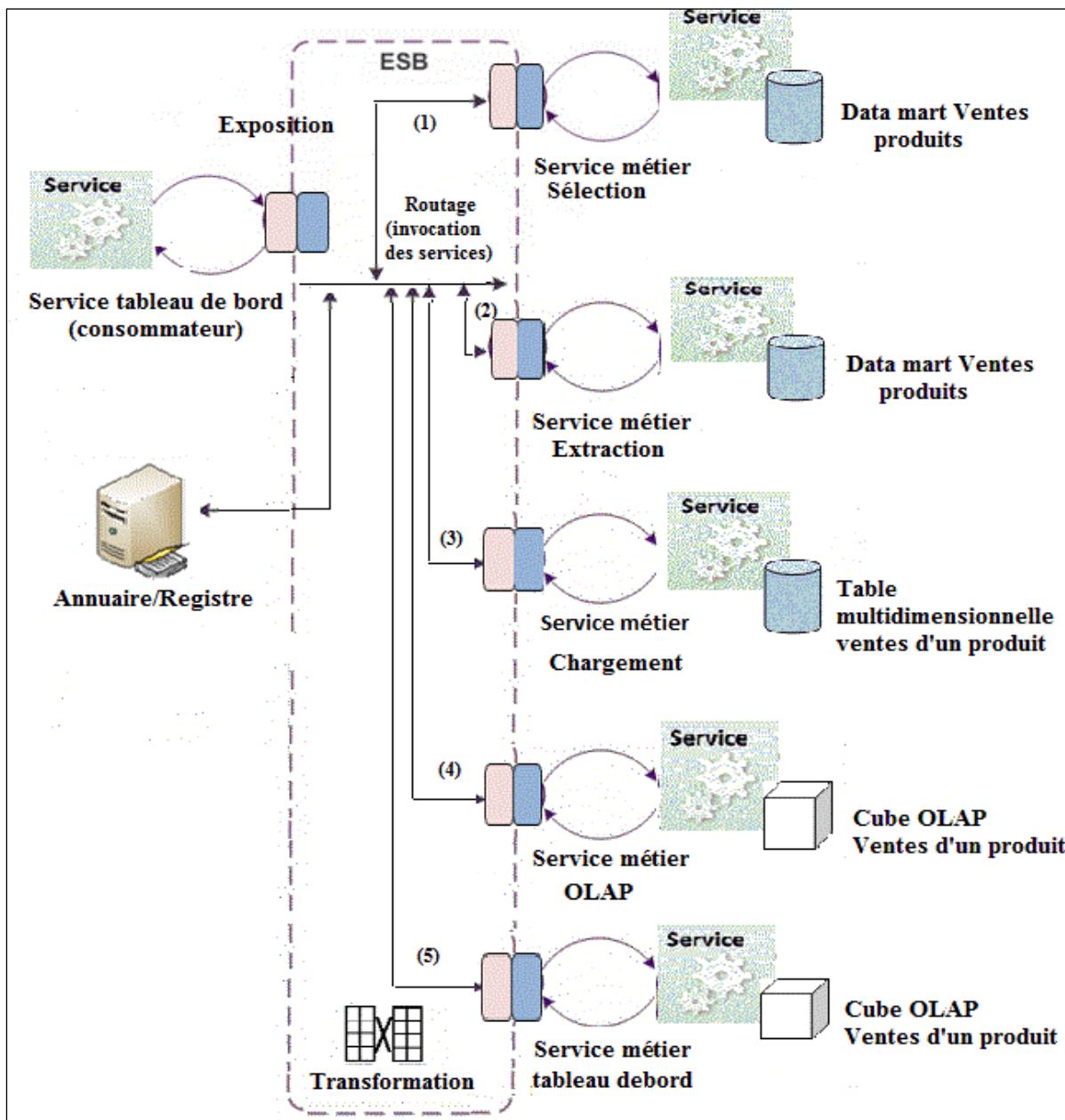


Figure 3.10 Invocation du service métier tableau de bord par l'ESB

3.5 Évaluation qualitative partielle et préliminaire de la SODWA

L'évaluation de la qualité du logiciel est une composante importante et critique qui permet d'évaluer les propriétés auxquelles le logiciel doit répondre.

Selon (Corrales, 2007), plusieurs méthodes ont été développées telles que « Architecture Tradeoff Analysis Method (ATAM) » et « Attribute-Based Architecture (ABAS) » pour évaluer une architecture logicielle selon les attributs de qualité tels que la performance, la fiabilité, la réutilisabilité, la disponibilité, la sécurité et autres.

Pour évaluer la SODWA proposée, les principaux attributs de qualités considérés pour illustrer une évaluation partielle et préliminaire de l'architecture sont:

- la disponibilité

Selon (Corrales, 2007), cet attribut doit assurer en tout temps pour l'ensemble des clients la disponibilité du système à chaque appel et que les interventions en cas d'anomalies observées sur les services soient minimes.

Pour cet attribut de qualité, la SODWA utilise l'ESB qui permet d'intégrer des solutions de sécurité efficaces selon (Bou Nassar, 2012, p.38) telles que:

- 1) la mise en place des services de sécurité connectés à l'ESB pour l'authentification sur les services;
- 2) le contrôle les messages en fonction des politiques de sécurité préétablies;
- 3) le contrôle pour vérifier l'acheminement des messages de sécurité (jetons d'authentification, autorisation d'accès, etc.) dans les services composés interconnectés à travers l'ESB pour transmettre les messages aux services concernés;
- 4) etc.

D'après la revue de la littérature, l'ESB assure aussi:

- le routage dynamique

Le médiateur envoie les messages à plusieurs récepteurs soit pour équilibrer les charges ou bien en cas de défaillance d'une source de données;

- la qualité de services

Le médiateur assure d'autres actions non fonctionnelles qui se rapportent à la gestion de la qualité du service (Quality of Service -QoS) tels que la mesure de la qualité, la mise en cache, la détection de défaillance et la récupération, la gestion incomplète des données, etc.

- la maintenabilité

Selon (Corrales, 2007), la maintenabilité est la charge allouée pour cerner les erreurs, les corriger et exécuter les modifications dans le système.

Au niveau de la SODWA, par exemple l'outil ETL est décomposé en services tels que le service « Extraction », le service « Transformation » et le « service Chargement » pour faire face à des demandes de mise à jour, de localiser directement l'erreur. En effet les composants de l'ETL qui sont sur le marché sont fortement couplés et s'exécutent avec une dépendance les uns des autres comme le montre la Figure 3.11.

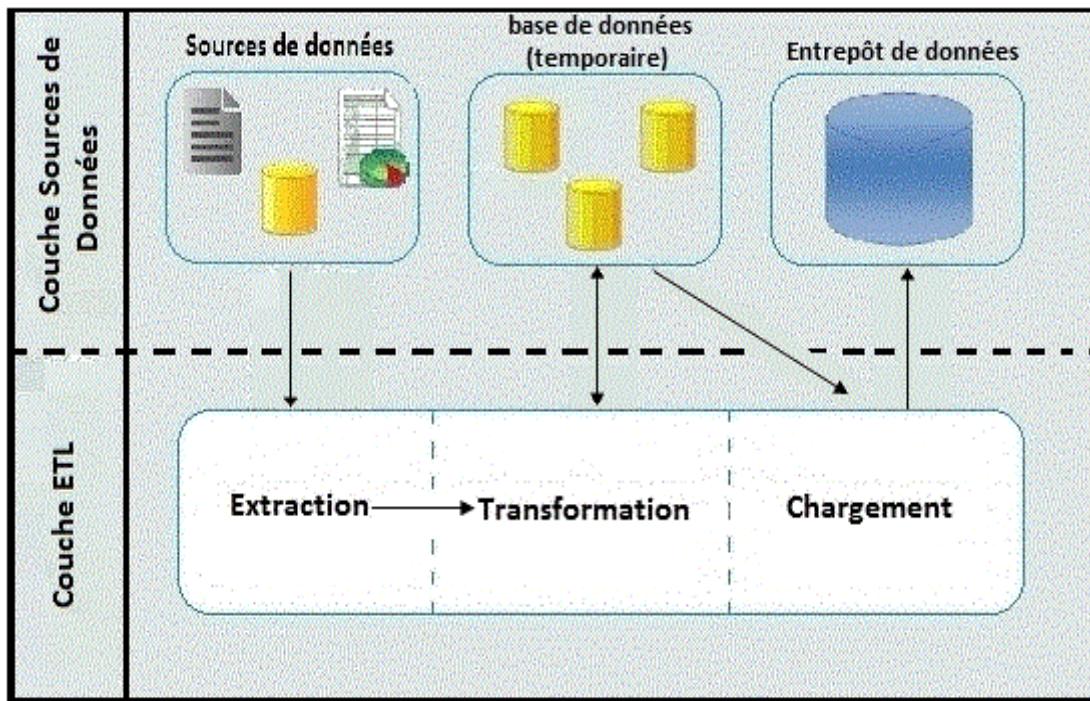


Figure 3.11 Utilisation de l'ETL traditionnel

Cette dépendance des composants de l'ETL a créé un obstacle pour toute opération d'ajout de nouvelles fonctionnalités, de modification dans le produit ETL ou de localisation de l'erreur ce qui a mené l'entreprise parfois à revenir pour développer en interne les tâches de l'ETL tout en assumant les coûts de maintenance à chaque besoin.

Pour faire face à ces contraintes, dans la SODWA, le produit ETL est fragmenté en composants, faiblement couplés avec l'autonomie de leurs redistributions par l'orchestration de l'ESB comme le montre la Figure 3.12.

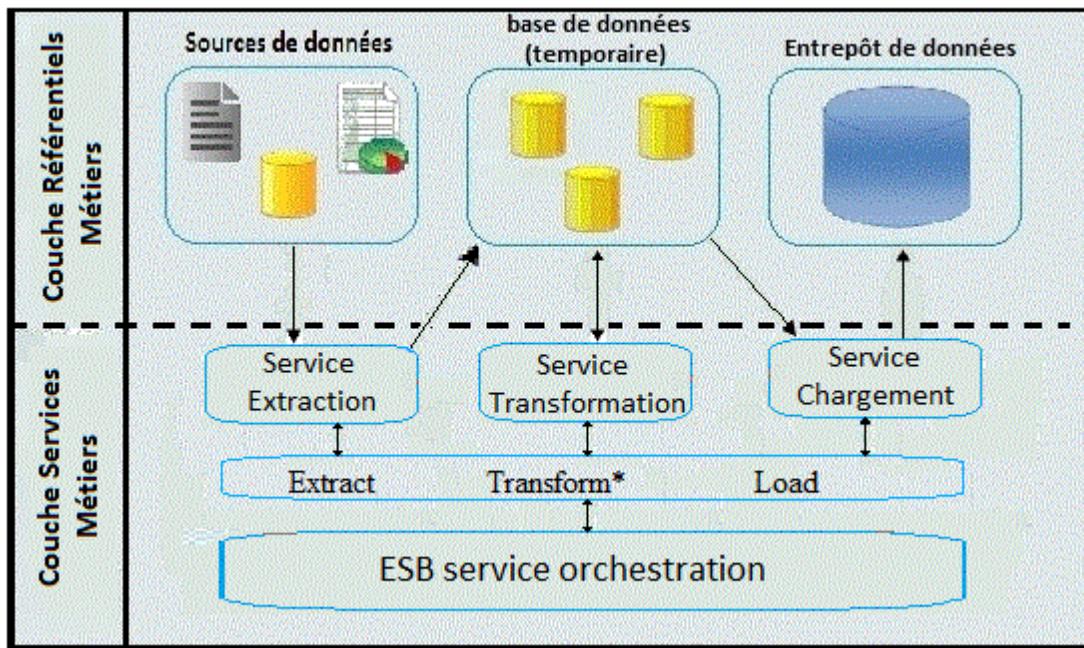


Figure 3.12 Conception d'intégration des composants de l'ETL dans une SOA

Dans la Figure 3.12, l'outil ETL est composé d'un ensemble de trois services (respectivement l'Extraction, la Transformation et le Chargement) dont chacun est considéré comme un fournisseur de services et en même temps consommateur de services. L'orchestration de ces services est gérée par l'ESB qui se base sur un annuaire de services pour la description. Cette décomposition en services permet d'intégrer de nouvelles demandes et exigences de l'entreprise, d'inclure d'autres composants par exemple l'appel d'un service rapport simple après l'appel du service extraction ou transformation ou chargement, et de localiser l'erreur puisque chaque service est autonome et tous les services sont faiblement couplés.

- la réutilisabilité

Selon (Corrales, 2007), la réutilisabilité est la souplesse de réutiliser les composants du logiciel dans un autre système en tant que tels ou avec une légère modification apportée.

Si on revient à la Figure 3.12, tous les services de l'ETL sont des services réutilisables par des applications de la couche application et logiciels de la SODWA. Pour éditer par exemple un rapport simple, une application COGNOS peut utiliser le service Extraction un composant réutilisable de l'ETL pour extraire les données.

3.6 Sommaire du chapitre

La SODWA proposée englobe les caractéristiques de l'entrepôt de données et de l'architecture orientée services (SOA).

En effet, dans la SODWA, les services sont pleinement intégrés. Le service ETL est intégré comme un ensemble de services d'extraction, de transformation et de chargement des données. Le moteur OLAP est aussi intégré en tant que service et fait appel à des services de rapports simples ou des tableaux de bord.

Les données sont pleinement intégrées. L'entrepôt de données et les « Data Marts » constituent les seules sources pour l'exploitation et l'analyse des données par les services métiers. Il s'agit de référentiels de données cohérentes et précises, lesquels fournissent une grande capacité d'accès rapide et direct par leurs structures. Ces référentiels sont alimentés par des sources de données diversifiées par le biais des services d'extraction, de transformation et de chargement.

Les applications existantes sont intégrables, et ce, par l'intégration des services Web pour réutiliser les logiciels d'applications existantes couvrant l'ensemble de l'entreprise. Les services Web sont appuyés par l'ESB pour le routage intelligent.

Les services peuvent accéder à des volumes importants de données historisées en utilisant des requêtes simples ou complexes. Les requêtes ad-hoc ou intuitives sont prises en compte pour faire par exemple du « *Reporting Self-Service* », accéder à certaines sources de données, faire le tri et le filtrage, descendre ou remonter dans la hiérarchie des données.

Un exemple est présenté pour illustrer le fonctionnement de la SODWA ainsi qu'une évaluation qualitative partielle et préliminaire de la SODWA.

CONCLUSION

Les travaux effectués dans ce projet de recherche visent à répondre à la problématique d'intégration de l'entrepôt de données dans une architecture SOA, et ce, dans le but de concevoir une intégration de l'entrepôt de données dans une architecture SOA en s'appuyant sur les services Web qui utilisent des technologies standards indépendantes du matériel et du langage de programmation, et sur l'ESB qui constitue un soutien fondamental pour les services Web au niveau routage et contrôle des messages.

L'étude et l'examen actuel des architectures orientées services et des architectures entrepôts de données sont le point de départ pour cerner la problématique de la recherche et définir l'objectif ciblé.

Une démarche méthodologique nous a guidé à identifier le système attendu pour proposer la conception de l'intégration de l'entrepôt de données dans la SOA.

La conception est basée sur quatre couches :

- la couche « Référentiels métiers » pour intégrer l'entrepôt de données et les magasins de données;
- la couche « Services métiers » pour le service ETL (service extraction, service transformation, service chargement), et le service OLAP (service rapport simple, service tableau de bord);
- la couche « Applications et logiciels »;
- la couche « Infrastructure physique » pour les serveurs Web, les serveurs bases de données et les serveurs d'applications.

Les limites pour l'évaluation de la SODWA sont :

- il est nécessaire de définir des mesures qualitatives rigoureuses pour évaluer cette nouvelle conception ;

- l'utilisation limitée de la norme ISO 25000 et d'autres *attributs* de qualité, tels que la communicabilité, la facilité d'apprentissage (« *Learn ability* »), la fiabilité et l'utilisabilité.

Les futures étapes requises pour améliorer la SODWA sont :

- intégrer les bonnes pratiques collaboratives pour la conception, le développement et la réalisation afin d'assurer le bon fonctionnement de la SODWA;
- faciliter l'utilisation des services par les consommateurs de services;
- évaluer l'efficacité et le temps d'apprentissage de la SODWA pour les utilisateurs novices et experts;
- faciliter l'utilisation de services par les débutants ainsi que des experts;
- soutenir l'approche orientée services pour la conception;
- favoriser la communication et la réutilisation des compétences individuelles concernant les bonnes pratiques de conception;
- intégrer toutes les différentes nouvelles technologies, y compris, mais sans s'y limiter, les ordinateurs de bureau traditionnels de bureau, ordinateurs portables, ordinateurs de poche (Palmtop), PDA, avec ou sans claviers, téléphones mobiles et téléviseurs interactifs.

Soucieux de la poursuite de ce projet de recherche avec d'autres travaux de recherche pour le développement et la réalisation de la SODWA, nous avons identifié des perspectives additionnelles qui consisteraient à intégrer les particularités de la SODWA proposée pour:

- quantifier cette architecture de logiciels pour en analyser les relations de leur taille fonctionnelle avec leurs efforts de développement;
- appliquer des techniques d'estimation de l'effort de développement de logiciel SODWA ;
- prévoir et quantifier les éventuelles évolutions du logiciel ;
- évaluer la charge nécessaire à la réalisation de futures exigences des utilisateurs ;
- etc.

ANNEXE I

LES PHASES DE DÉVELOPPEMENT D'UN PROJET D'ENTREPÔT DE DONNÉES

Le développement d'un projet d'entrepôt de données est composé de 16 étapes qui sont regroupées dans 6 phases (Moss, 2003). Seules les phases pertinentes pour notre projet sont décrites ci-dessous.

Phase 1 : Analyse d'affaire

Étape 1: Définition des besoins du projet

Dans cette phase, on définit les besoins du projet décisionnel. Les besoins doivent contenir le problème d'affaires existantes, les justifications de la solution et la résolution du problème. Tous les domaines d'affaires qui se rapportent au projet décisionnel doivent être détaillés. Toutes les sources de données doivent être définies ainsi que leurs nettoyages, l'historisation des données, la qualité des données, etc. Comme les utilisateurs veulent tout à court terme, le chef de projet doit procéder par le prototypage qui a pour rôle de cerner les besoins réels et de familiariser l'utilisateur avec les solutions d'affaires.

Étape 2 : Analyse des données

Cette étape est la plus rigoureuse et la plus critique à la réussite du projet. Pour mener un projet décisionnel, les données doivent être de qualité. Le consommateur de données d'analyses doit spécifier les sources de données de l'opérationnel pour la sélection. Il identifie aussi le processus de nettoyage et de transformation des données. Toute donnée dans le système décisionnel doit avoir une vue unique.

Étape 3 : Prototypage de l'application

Cette étape est facultative. Cependant, elle consiste à estimer le coût et le temps nécessaire pour la réalisation définitive du projet. La validation d'un prototype permet à l'utilisateur de revérifier ses besoins et les réajuster sans attendre la livraison définitive de la solution.

Étape 4 : Analyse du référentiel de métadonnées

Les métadonnées sont des données de connaissances sur les données. Elles sont centralisées dans un référentiel qui décrit dans le temps les règles de transformation, les règles de gestion, les règles techniques et non techniques sur la donnée, etc. Ce référentiel est nécessaire pour l'utilisateur qui veut justifier une donnée analysée pour la prise de décision. Ce référentiel doit être documenté dans un modèle logique de métadonnées.

Phase 2 : Conception

Étape 5 : Conception des bases de données

La conception des bases de données d'un système décisionnel est différente de celles des bases de données de l'opérationnel. Il faut maîtriser les nouvelles technologies pour la conception d'un système décisionnel. Selon les besoins de l'utilisateur, les bases de données cibles peuvent être détaillées ou agrégées. Cependant, toute opération d'accès à ces bases doit répondre aux besoins formulés par l'utilisateur. Dans cette étape, il faut faire la conception logique et physique des bases de données ciblées.

Étape 6 : Conception du processus ETL

Dans un projet décisionnel, il est important d'avoir un processus ETL qui centralise les données provenant de l'opérationnel. L'objectif principal de la conception de ce processus est l'intégration des données dans un format standard dans les bases de données cibles. Cette conception doit définir le diagramme de flux de données, le nettoyage et la transformation des données, la charge incrémentale des données, etc.

Étape 7 : Conception du référentiel de métadonnées

Phase 3 : Construction

Étape 8 : Développement du processus ETL

La sélection, le nettoyage et la transformation des données sont des activités importantes dans un projet décisionnel. Il s'agit d'éliminer les données non utiles et les transformer pour obtenir des données standardisées. Ce processus ETL est un outil existant dans le marché qui répond généralement aux besoins, sinon développer une extension pour le compléter. Il ne faut pas négliger la phase de test au complet de ce processus pour voir les résultats attendus par l'utilisateur.

Étape 9 : Développement de l'application

Il s'agit de développer les différents rapports qui sont définis par l'utilisateur dans la phase de conception. Le prototype étant validé et les besoins fonctionnels sont claires et arrêtés, on peut commencer le développement des applications. Sur le marché, il existe des outils pour la création de rapports et d'analyse des données. Un plan d'action de formation est nécessaire aux utilisateurs pour les initier et les former afin de maîtriser les outils et voir les possibilités offertes.

Étape 10 : Exploration des données (*Data Mining*)

Sur le marché, il existe des outils dédiés à l'exploration de données. Ces outils utilisent des algorithmes qui explorent des informations cachées dans le système. Par exemple voir la solvabilité d'un client et prédire qu'il est un bon client ou pas. Le *data mining* est une composante essentielle dans un système décisionnel pour découvrir des connaissances cachées dans le système.

Étape 11 : Développement du référentiel de métadonnées

Le référentiel de métadonnées dans un système décisionnel permet à l'utilisateur de prendre des décisions de manière plus efficace. On peut acheter la licence pour disposer d'un référentiel ou bien le développer. Ce référentiel est enrichi au fur et à mesure de l'utilisation du processus ETL. Ce référentiel nous donne les statistiques sur le chargement des données, les rejets sur les données chargées et les raisons de rejets. Pour avoir un référentiel sain et utile, l'administrateur de ce référentiel doit collaborer pleinement avec tous les intervenants dans le projet décisionnel tels que l'équipe ETL, l'équipe d'analyse des données, l'équipe de *Data mining*, l'administrateur des bases de données cibles et les personnes qui maintiennent les données sources de l'opérationnel. Dans le plan du projet, il faut prévoir suffisamment de temps pour tester ce référentiel.

Phase 4: Déploiement

Étape 12 : Implantation

Dans cette étape, il faut suivre une approche itérative comme celle utilisée dans la phase de développement de l'application. Le but de cette approche est de faire associer à l'équipe de départ les personnes qui n'ont pas suivi au départ le projet et en même temps réduire les problèmes rencontrés dans l'application avec toute l'organisation. Comme travaux à effectuer dans cette étape sont tels que la fixation de la date d'implantation, la mise en place de l'application en production et l'alimentation des bases de données de production. Il faut mettre aussi l'accent sur les tâches relatives à l'aspect gestion de la sécurité, gestion de sauvegarde des données, l'utilisation des ressources pour suivre les performances, etc.

Étape 13 : Évaluation de la version

Il est recommandé de faire la revue post-implantation pour tenir compte lors de la prochaine version du prototype des améliorations demandées et les suggestions formulées par l'organisation. Cette évaluation permet aussi de voir les processus développés et non utilisés et d'estimer le coût du projet.

ANNEXE II

LES LOGICIELS ETL ET LES LOGICIELS D'ANALYSES ET DE RAPPORTS

Cet annexe donne une vue d'ensemble sur les principaux fournisseurs d'outils ETL et d'applications d'analyses et d'explorations des données de l'entrepôt de données.

Les logiciels ETL

- **Informatica PowerCenter** (Dargues, 2009)

Informatica Power Center est un outil ETL qui permet de centraliser des données provenant de sources variées en y appliquant plus ou moins de transformations. Il est effectivement possible d'extraire des données de fichiers plats (délimités, XML, positionals), ou de bases de données comme Oracle, IBM DB2, Informix, MSSQL, d'y appliquer des transformations, suivant des règles métiers, et de charger le tout dans un entrepôt de données (Data Warehouse) ou même dans des fichiers plats. A l'installation de Power Center, les applications clientes disponibles sont :

- le Repository Manager pour la gestion des référentiels, des utilisateurs, les droits d'accès, etc ;
- le Repository Server pour la gestion des connexions, des logs, etc ;
- le Designer pour définir les sources de données, les cibles et développer des mappings (extraction, transformation, et chargement des données dans les bases cibles) ;
- le Workflow Manager pour l'ordonnancement des mappings suivant un ordre précis ou en parallèle, et pour la configuration des mappings ;
- le Workflow Monitor pour le suivi des « *workflows* » tels que le lancement du traitement, son arrêt, l'accès à ses propriétés ainsi qu'aux logs de traitement.

- **DataStage** (Alur et al, 2008)

DataStage est un outil ETL d'IBM sous le nom « *IBM InfoSphere DataStage* ». Il permet d'intégrer les données dans les bases cibles à partir de données sources simple ou complexes en temps réel ou différé. Les données sources sont telles que les fichiers plats, les bases de données relationnelles, les messages des files d'attente. La haute performance de l'outil permet de traiter des données volumineuses. Le traitement des données peut se faire en temps réel ou en batch. Les bases de données qui sont compatibles avec DataStage sont Oracle, IBM, Teradata, ODBC, Microsoft SQL Server et Informix.

DataStage est un ensemble de composants ou modules (*DataStage Designer*, *DataStage Director* et *DataStage Administrator*). Le *DataStage Designer* est module pour la conception des travaux relatifs à l'extraction des données à partir des sources

de données, la transformation et le chargement des données. Le *DataStage Director* est un module de suivi des activités lancées par l'utilisateur pour visualiser la qualité des résultats obtenus et donner les détails de l'activité en cas de problème. Le *DataStage Administrator* est un module qui gère tous les projets et l'utilisateur administrateur peut créer et supprimer un projet.

DataStage offre des fonctions principales telles que :

- la connexion aux applications et aux bases de données;
- le déploiement et la haute performance;
- la conception, la création et l'exécution des tâches via une interface;
- la gestion des droits d'accès.

- **Business Object Data Integrator** (SAP, 2012)

Le logiciel SAP *Data Integrator* est un outil ETL graphique qui permet d'intégrer des données structurées (les fichiers plats, les fichiers XML, les services web, les bases de données, etc.) et non structurées (les e-mail, les journaux web, les blogs, etc.) à n'importe quelle fréquence de rafraîchissement dans les systèmes cibles pour donner une vue complète d'informations fiables et de qualité à l'entreprise.

Le SAP *Data Integrator* a des fonctionnalités de débogage avancées pour analyser les données à problèmes durant tout le processus d'extraction, de transformation et de chargement.

- **IBM Cognos Data Manager** (COGNOS, 2006)

IBM Cognos Data Manager ou (*Decision Stream*) est une interface graphique qui simplifie les procédures d'extraction, de transformation et de chargement des données pour l'informatique. Cet outil ETL supporte plusieurs sources de données pour charger l'entrepôt de données et les data marts tels que Oracle SQL*Net, SQL SERVER (OLE-DB), Teradata, Sybase, ODBC, IBM DB2, cubes OLAP, données d'ERP, XML et les données des gros systèmes. Il est doté aussi d'un moteur puissant d'agrégation et de fusion de données.

Ses principales fonctions d'intégration sont :

- la simplification des procédures d'intégration des données. Cette fonction automatise la plus part des procédures complexes tels que la création des tables et leurs gérances;
 - le chargement de l'historique des dimensions ;
 - la validation des tables hiérarchiques selon le niveau d'agrégation des données;
 - la création de prototype avec un déploiement rapide ;
 - la prise en charge de différentes langues pour étendre l'intégration des données.
- Cette fonction reconnaît les données et les métadonnées qui intègrent des caractères spécifiques de langues étrangères en s'appuyant sur des standards Unicode;

Pour les processus automatiques qui se lancent en temps différé, le processus est le *Job stream*.

Les logiciels d'analyses et de rapports

- **SAP Business Objects XI** (Decivision, 2010)
 SAP Business Objects XI offre des outils d'analyse et de rapports aux utilisateurs décisionnels tels que :
 - **Crystal report**
 Est un outil pour la création de rapports. Comme fonctions, il permet la publication des rapports sur le web par l'utilisation d'un assistant, le lancement de rapports à une fréquence donnée en format PDF, RDF et Excel, la publication d'un rapport à un large public qui a la possibilité d'afficher le tout ou un sous ensemble de données du rapport publié, l'insertion d'objets (texte, image, formules, etc.) dans le rapport, l'insertion des objets par des liens qui se trouvent dans la base de données, la personnalisation du rapport, offre des connectivités à plusieurs sources de données (plus de 35 pilotes), la prise en charge des fichiers en XML, la création de tableaux croisés, la diversité des langues par l'utilisation du format Unicode, la création de rapports par l'exploration des données relationnelles et OLAP, etc.
 - **Web intelligence**
 Est une application qui permet de faire des requêtes ad hoc et des rapports sur le serveur à partir d'un url sur le poste client. L'utilisateur se dispose d'une interface graphique riche en conception de modèles préétablis de rapports et avec des fonctionnalités (boutons, palette d'outils) pour créer facilement son rapport. Dans le rapport, l'utilisateur peut intégrer des liens hypertextes, explorer les données par drill Down, faire des filtres sur des données et des ruptures. Dans le rapport, l'utilisateur peut intégrer des données externes et les comparer aux données du système. Le rapport peut être sauvegardé en format CSV.
 - **Desktop Intelligence**
 Est une application client-serveur qui permet à l'utilisateur, dans son poste de travail, la création de rapports et de requêtes dans un document *Desktop intelligence*. L'utilisateur peut utiliser des fichiers de données personnels, insérer des calculs et des formules, regrouper les données, utiliser les fonctionnalités d'OLAP (analyse descendante et ascendante), rafraîchir les données par la connexion au serveur et déployer son rapport sur le serveur en mode public.
- **ORACLE Hyperion** (Edward, 2010)
Oracle Hyperion Reporting Interactive est un produit personnalisable et puissant pour la conception et le déploiement des documents (rapport simple, tableau de bord) basés sur des requêtes d'analyses. Ce produit s'intègre dans un espace de travail (*Workspace*) semblable à un portail où tous les rapports sont sauvegardés, partagés et

publiés selon le privilège d'accès pour la création de modèles, la modification de requêtes, l'enregistrement et l'importation de documents.

L'outil « *Hyperion Reporting* » offre un service flexible et facile d'utilisation pour la navigation Web appelé client Web « *Plug-In* ». Il existe des fonctionnalités robustes telles que l'accès aux bases de données relationnelles, multidimensionnelles (incluant cubes OLAP pour les requêtes, la manipulation des résultats des requêtes, l'affichage graphique du rapport au format similaire *Microsoft Excel Pivot* (plug-in de Microsoft), le filtrage des objets affichés sur le rapport, etc.

BIBLIOGRAPHIE

- Adamson, Christopher. 2010. «*Star Schema: The Complete Reference*». «Mc Graw Hill». New York, 512 p. En ligne. <<http://library.books24x7.com/toc.aspx?bookid=35985>>. Consulté le 20 février 2012.
- Alur, Nagraj, Celso Takahashi, Sachiko Toratani and Denis Vasconcelos. 2008. «*IBM InfoSphere DataStage Data Flow and Job Design*». IBM Redbooks. 658 p. ISBN:9780738431116. En ligne. <<http://library.books24x7.com/toc.aspx?bookid=30615>>. Consulté le 8 Juin 2013.
- Arsanjani, Ali, Ghosh, S., Allam. A., Abdollah.T., Ganapathy. S., and Holley.K. 2008. «*SOMA: A method for developing service-oriented solutions* ». IBM Systems Journal, 47 (3), p.377-396. En ligne <<http://www.iieom.org/ieom2011/pdfs/IEOM071.pdf>>. Consulté le 29 octobre 2013.
- Arsanjani, Ali. 2004. «*Service-oriented modeling and architecture: An architectural template for a SOA*». In Le site d'IBM : developper Works. En ligne. <<http://www-adele.imag.fr/Les.Publications/reports/PHD2008Mar.pdf>>. Consulté le 3 mai 2012.
- Bleakley, Darrell, Jenny Chow, Kirstine Clapperton, Sowmya Hebbur Dayananda Vivek. Grover et Sunho Sung. 2010. «*Connecting Your Business Using IBM WebSphere Message Broker V7 as an ESB*». IBM reedbooks. 558p. En ligne. <<http://library.books24x7.com/toc.aspx?bookid=37045>>. Consulté le 1 novembre 2013.
- Breslin, Mary. 2004. «*Data Warehousing Battle of the Giants: Comparing the Basics of the Kimball and Inmon Models*». *Business intelligence journal* winter .2004. En ligne. <https://cours.etsmtl.ca/mti820/public_docs/lectures/DWBattleOfTheGiants.pdf>. Consulté le 2 avril 2012.
- Boukadi, Khouloud. 2009. «*Coopération interentreprises à la demande : Une approche flexible à base de services adaptables* ». Thèse de doctorat, Saint-Étienne, 302 p. En ligne. <www.emse.fr/spip/IMG/pdf/Khouloud-boukadi-diff_v2.pdf>. Consulté le 2 avril 2012.
- Bou Nassar, Pascal, 2012. « Gestion de la sécurité dans une infrastructure de services dynamique : Une approche par gestion des risques ». Thèse de doctorat, Lyon, 235p. En ligne. <<http://tel.archives-ouvertes.fr/docs/00/82/85/98/PDF/these.pdf>>. Consulté le 6 avril 2014.
- COGNOS. 2006. « Integration des Données avec Cognos 8 Business Intelligence ». En ligne. <<http://www.gec-software.fr/vdoc/easysite/action/WebdriveActionEvent/oid/02m-00000j-02r>>. Consulté le 30 Juin 2013.

Colombe Herault, Thomas.GaÄel et and Lalande.Philippe. 2005. «*Mediation and Enterprise Service Bus: A position paper*». En ligne. <<http://ceur-ws.org/Vol-168/MEDIATE2005-paper5.pdf>>. Consulté le 22 mars 2014.

Corrales, Yaqueline, et Claude Y.Laporte 2007. «Étude de cas: Évaluation de la migration d'une architecture logicielle d'une société de commerce électronique». pp. 40-55.

Cottin, Nicolas, Agour Frendy. 2007. « *XML-Family :Web Services Description W.S.D.L*». En ligne. 29 p. <http://frendag.free.fr/AWSIA/Presentation_WSDL/WSDL.pdf>. Consulté le 11 mars 2013.

Dargues, Jean Sébastien. 2009. « Présentation de la suite Informatica Power Center ». En ligne.<<http://jsdarges.developpez.com/tutoriels/presentation-suite-informatica-power-center/>>. Consulté le 20 juin 2013.

Decivision. 2010. « Livre Blanc Business Objects ». En ligne. <http://dosi.univ-avignon.fr/uploads/res_adm_gestion/DeciVision-Architecture_et_detai_BO.pdf>. Consulté le 21 décembre 2013.

Edward, J. Cody. 2010. «*The Business Analyst's Guide to Oracle Hyperion Interactive Reporting 11: Quickly Master this Powerful Business Intelligence Product* ». ISBN:9781849680363. En ligne. <<http://library.books24x7.com/toc.aspx?bookid=37977>>. Consulté le 22 décembre 2013.

El Akkoui, 2009. «Incorporating the Service-Oriented Architecture into Data Warehouses». Thèse de doctorat, Wallonie- Bruxelles, 57 p. En ligne. <<http://code.ulb.ac.be/dbfiles/Ela2010masterthesis.pdf>>. consulté le 17 juin 2014.

He, Hao, 2003. «*What is Service-Oriented Architecture* ». In. O'Reilly Xml.com. En ligne. <<http://webservices.xml.com/pub/a/ws/2003/09/30/soa.html>>. Consulté le 10 mai 2012.

Hernández, Otto Poveda. 2008. «*La gouvernance SOA Ses aspects théoriques et pratiques*» Travail de master en Informatique de Gestion, Fribourg, 128 p. En ligne <http://diuf.unifr.ch/main/is/sites/diuf.unifr.ch.main.is/files/file/studentprojects/M-2008_Otto_Poveda.pdf>. Consulté le 10 mai 2012.

Hugo, Etiévant. 2005. « Le dictionnaire des développeurs : Définition de ESB ». En ligne. <<http://dico.developpez.com/html/303-Generalites-ESB-Enterprise-Service-Bus.php>>. Consulté le 19 Janvier 2014.

IBM. 2012. «*IBM Cognos Query Studio Version 10.2.0: Guide d'utilisation*». 135 p. En Ligne.<http://public.dhe.ibm.com/software/data/cognos/documentation/docs/fr/10.2.0/ug_cr_qstd.pdf>. Consulté le 9 juillet 2013.

- IBM. 2010. «*IBM Cognos Business Intelligence Version 10.1.0*». 72 p. En ligne. <http://public.dhe.ibm.com/software/data/cognos/documentation/docs/fr/10.1.0/wig_cr.pdf>. Consulté le 9 juillet 2013.
- Inmon, W.H. 2006. «*Making the SOA Implementation Successful*». *The Data Administration Newsletter (TDAN)*. Publié le 1 octobre 2006. En ligne. <<http://www.tdan.com/view-articles/4593>>. Consulté le 13 novembre 2013.
- Inmon, W. H. 2005. «*Building the Data Warehouse*», 4ème édition, John Wiley & Sons, 574 p. En ligne. <<http://library.books24x7.com/assetviewer.aspx?bookid=12458&chunkid=978465169&rowid=95>>. Consulté le 10 février 2012.
- Kimball, Ralph, et Ross Margy. 1996. «*The Data Warehouse Toolkit*». En ligne. <<http://www.cerkinfo.be/media/memoires/2006/verhaegen.pdf>>. Consulté le 10 février 2012.
- Krafcig, D, Banke.K, et Slama.D. 2004. «*Enterprise SOA: Service-Oriented Architecture Best Practices*». Prentice Hall PTR, 408 p. ISBN: 978-0131465756 En ligne <<http://hal.archives-ouvertes.fr/docs/00/74/35/53/PDF/these.pdf>>. Consulté le 2 avril 2012.
- Marks, E.A. et Bell. M. 2006. "Service Oriented Architecture (SOA): A Planning and Implementation Guide for Business and Technology": John Wiley & Sons, 384 p. ISBN: 9780471768944. En ligne. <<http://www.etsmtl.ca/Bibliotheque/>>. Consulté le 29 octobre 2013.
- Moss, Larissa T et Atre Shaku. 2003. «*Business Intelligence Roadmap: The Complete Project Lifecycle for Decision-Support Applications*», Addison-Wesley. En ligne. 576 p. <<ftp://193.33.64.118/pub/Storage00/books/Project Management Collection 2/Larissa T. Moss, Shaku Atre - Business Intelligence Roadmap - The Complete Project Lifecycle for Decision-Support Applications - L T Moss & S.pdf>>. Consulté le 23 mars 2012.
- Mtiraoui, Abderraouf. 2013. «Web services : Etude et conception d'une plateforme de services pour un environnement numérique de travail d'Université», Institut numérique. En ligne. <<http://www.institut-numerique.org/12-les-services-web-51f7a7f9c2daa>>. Consulté le 4 octobre 2013.
- Rainardi, Vincent. 2008. «*Building a Data Warehouse: With Examples in SQL Server*». Apress, New York: Jeffrey Pepper, 521 p. En ligne. <<http://www.springerlink.com/content/g280260183654505/fulltext.pdf>>. Consulté le 20 février 2012.

Ronnie, H. Abrahiem. 2008. «*Data Warehousing with Service-oriented Architecture: Designing and Implementing Prototype Models For an Integration of Near-Real-Time*». 232 p. Lambert Academic Publishing, ISBN 978-3838303673.

SAP. 2012. «*SAP® Data Integrator: Delivering Trusted Data for Your Enterprise*». En ligne. <<http://drh.img.digitalriver.com/DRHM/Storefront/Site/bobjamer/cm/multimedia/PDF/SAP-Data-Integrator.pdf>>. Consulté le 18 Décembre 2013.

Shreeraz, Shah. 2007. «*Hacking Web Services*». 352 p. ISBN:9781584504801. <<http://library.books24x7.com/toc.aspx?bookid=26193>>. Consulté le 31 octobre 2013.

Tari, Zahir, Ann. Khoi ,Anh. Phan, Malith. Jayasinghe and Vidura. Gamini Abhaya. 2011. «*On the Performance of Web Services*» 250 p. ISBN:9781461419297. En ligne. <<http://library.books24x7.com/toc.aspx?bookid=45852>>. Consulté le 31 octobre 2013.

Toumi, Abdennour. 2012. «*Introduction to SOA*». En ligne. 4 p. <<http://dl.dropbox.com/u/55765722/delivred/abdenour/tutoPDF/soa/1.introductionSOA.pdf>>. Consulté le 11 mars 2013.

Vangenot, C. 2005. «*Datawarehouse*». Ecole polytechnique fédérale de Lausanne, 77 p. En ligne. <<http://lbd.epfl.ch/files/courses/ibd-2005/DW.pdf>>. Consulté le 10 février 2012.

Van den Berg, Martin, Norbert Bieberstein , et Erik Van. Ommeren. 2007. «*SOA for Profit : Guide du manager pour une SOA réussie*» Pays-Bas : IBM et Sogeti, 280 p. En ligne. <http://www.sogeti.com/upload/COM/Curious_about_us/Documents/SOA_for_Profit_French_version.pdf>. Consulté le 30 septembre 2011.

Zentut, 2013. In «*Ralph Kimball Data Warehouse Architecture: Introduction to Ralph Kimball data warehouse architecture*». En ligne. <<http://www.zentut.com/data-warehouse/ralph-kimball-data-warehouse-architecture/>>. Consulté le 20 janvier 2013.