UNIVERSITÉ DE LORRAINE

ECOLE DOCTORALE IAEM Lorraine

Modélisation de processus métiers sensibilisés aux risques et déploiement en confiance dans le cloud

THÈSE

présentée et soutenue publiquement le 21 Octobre 2015

pour l'obtention du

Doctorat de l'Université de Lorraine

(mention informatique)

par

Elio Goettelmann

Composition du jury

Rapporteurs :	Prof. Salima Benbernou	Université Paris Descartes, France
	Prof. Haralambos Mouratidis	Université de Brighton, Royaume-Uni
Membres :	Prof. Frédérique Biennier	INSA-Lyon, France
	Prof. Eric Dubois	Université de Namur, Belgium
	Dr. Benjamin Gâteau	Luxembourg Institute of Science and Technology
	Prof. Laurent Vigneron	Université de Lorraine, France
Encadrant :	Prof. Claude Godart	Université de Lorraine, France

Fonds National de la Recherche Luxembourg

Loria Laboratoire lorrain de recherche en informatique et ses applications

LUXEMBOURG INSTITUTE OF SCIENCE AND TECHNOLOGY LIST

Layout created with the ’thesul’ class.

Abstract

Nowadays service ecosystems rely on dynamic software service chains that span over multiple organisations and providers. They provide an agile support for business applications, governments of end-users. This trend is reinforced by the Cloud based economy that allows sharing of costs and resources. However, the lack of trust in such cloud environments, that involve higher security requirements, is often seen as a braking force to the development of such services.

The objective of this thesis is to study the concepts of service orchestration and trust in the context of the Cloud. It proposes an approach which supports a trust model in order to allow the orchestration of trusted business process components on the cloud.

The contribution is threefold and consists in a method, a model and a framework. The method categorizes techniques to transform an existing business process into a risk-aware process model that takes into account security risks related to cloud environments. The model formalizes the relations and the responsibilities between the different actors of the cloud. This allows to identify the different information required to assess and quantify security risks in cloud environments. The framework is a comprehensive approach that decomposes a business process into fragments that can automatically be deployed on multiple clouds. The framework also integrates a selection algorithm that combines security information with other quality of service criteria to generate an optimized configuration.

Finally, the work is implemented in order to validate the approach. The framework is implemented in a tool. The security assessment model is also applied over an access control model. The last part presents the results of the implementation of our work on a real world use case.

Keywords: Business Process Management, Cloud Computing, Security Risk Management

Résumé

L'essor du Cloud Computing, permettant de partager les coûts et les ressources au travers de la virtualisation, présage une interconnexion dynamique et flexible entre entreprises et fournisseurs. Cependant, cette mise en commun de ressources, données et savoir-faire implique de nouvelles exigences en termes de sécurité. En effet, le manque de confiance dans les structures du Cloud est souvent vu comme un frein au développement de tels services.

L'objectif de cette thèse est d'étudier les concepts d'orchestration de services, de confiance et de gestion des risques dans le contexte du Cloud. La contribution principale est un framework permettant de déployer des processus métiers dans un environnement Cloud, en limitant les risques de sécurité liés à ce contexte.

La contribution peut être séparée en trois partie distinctes qui prennent la forme d'une méthode, d'un modèle et d'un framework. La méthode catégorise des techniques pour transformer un processus métier existant en un modèle sensibilisé (ou averti) qui prend en compte les risques de sécurité spécifiques aux environnements Cloud. Le modèle formalise les relations et les responsabilités entre les différents acteurs du Cloud. Ce qui permet d'identifier les différentes informations requises pour évaluer et quantifier les risques de sécurité des environnements Cloud. Le framework est une approche complète de décomposition de processus en fragments qui peuvent être automatiquement déployés sur plusieurs Clouds. Ce framework intègre également un algorithme de sélection qui combine les information de sécurité avec d'autres critère de qualité de service pour générer des configuration optimisées.

Finalement, les travaux sont implémentés pour démontrer la validité de l'approche. Le framework est implémenté dans un outil. Le modèle d'évaluation des risques de sécurité Cloud est également appliqué dans un contexte de contrôle d'accès. La dernière partie présente les résultats de l'implémentation de nos travaux sur un cas d'utilisation réel.

Mots-clés: Gestion des Processus Métiers, Cloud Computing, Gestion des Risques de Sécurité

Sommaire

Chapitre 1
Introduction

1.1 Introduction générale 1
1.2 Problématique, questions et méthodologie de recherche 2
1.3 Résumé des contributions 3
1.4 Exemple illustratif 3

Chapitre 2
Etat de l'art

2.1 Cloud Computing: l'informatique en "nuage" 5
2.1.1 Définition 5
2.1.1.1 L'architecture cloud - Les 3 niveaux de service 6
2.1.1.2 Modèles de déploiement 6
2.1.1.3 Les acteurs du cloud 7
2.1.2 Les enjeux du cloud 7
2.2 Gestion des Risques de Sécurité des Systèmes d'Information 7
2.2.1 Définition 7
2.2.1.1 Vocabulaire - Le modèle ISSRM 8
2.2.1.2 Le processus de gestion de risques générique 8
2.2.1.3 Les stratégies de traitement des risques 9
2.2.2 Gestion des risques de sécurité dans le cloud 9
2.3 Gestion des processus métiers 9
2.3.1 Définition 10
2.3.1.1 Le cycle de vie 10
2.3.1.2 Les 3 niveaux BPM 10
2.3.1.3 Architecture de référence 11
2.3.2 Gestion de processus métiers, cloud et risques de sécurité 11
2.3.2.1 Gestion de processus métiers et cloud 11

2.3.2.2 La sécurité dans la gestion des processus métiers 12

Chapitre 3
Alignement des domaines et considérations méthodologiques

3.1 Distribution du cycle de vie et des niveaux de BPM 15
3.1.1 Le fournisseur cloud 16
3.1.2 Le consommateur cloud 16
3.1.3 Le broker 16
3.2 Méthodologie 17
3.2.1 Pré-sélection des services et établissement du contexte 17
3.2.2 Évaluation des risques 18
3.2.3 Traitement des risques 18
3.2.3.1 Transformation sémantique 18
3.2.3.2 Transformation structurelle 18
3.2.3.3 Sélection des offres cloud 18
3.2.3.4 Implémentation de contrôles de sécurité 19
3.2.4 Acceptation des risques 19
3.2.5 Déploiement 19

Chapitre 4
Evaluation des risques dans un contexte cloud

4.1 Aperçu 21
4.2 Modèle formel 22
4.2.1 Le fournisseur cloud: implémentation de contrôles sur les offres 22
4.2.2 Le consommateur cloud: définition des objectifs sur les actifs 23
4.2.3 Le broker: menaces, atténuations et conséquences 24
4.2.4 Couverture: implémentation des contrôles et atténuation des menaces 24
4.2.5 Dommage: exigences de sécurité et conséquences des menaces 25
4.2.6 Risque: probabilité de menace 25

Chapitre 5
Déploiement d'un processus métier sur plusieurs clouds

5.1 Aperçu 27
5.2 Les critères à évaluer 28
5.2.1 Le coût 28
5.2.2 La qualité de service 29
5.2.3 La complexité 29

5.2.4 D'autres exigences fonctionnelles (et non-fonctionnelles) 30
5.3 Approche de décomposition et de déploiement 30
5.3.1 Transformation 30
5.3.2 Pré-partitionnement 30
5.3.3 Sélection optimisée 31
5.3.3.1 Considérer plusieurs critères 31
5.3.3.2 Heuristiques 32
5.3.4 Décentralisation et synchronisation 32
5.3.5 Déploiement 32

Chapitre 6
Implémentation

6.1 Les outils développés 33
6.1.1 Évaluation des risques 33
6.1.1.1 Construction du modèle 33
6.1.1.2 Définition des exigences de sécurité 34
6.1.1.3 Évaluation des risques pour chaque fournisseur 34
6.1.2 Sélection optimisée 34
6.1.3 Déploiement de processus 36
6.1.3.1 Expérience de déploiement 36
6.2 Application dans le domaine du contrôle d'accès 37
6.3 Cas d'étude réel 38

Chapitre 7
Conclusion

7.1 Résumé des contributions 39
7.2 Limites et perspectives 39

Bibliographie **41**

Chapitre 1

Introduction

1.1 Introduction générale

Les entreprises d'aujourd'hui doivent faire face à de nombreux défis pour rester compétitives: "innovation", "diversification", "montée en gamme", "externalisation", "croissance exponentielle", "recentrage de l'activité principale" en sont quelques uns. Tandis que le marché devient de plus en plus incertain et imprévisible, de telles stratégies visent à rendre une entreprise plus efficace dans la distribution des ses produits ou services et plus flexible vis-à-vis des ses dépendances externes (clients, partenaires, fournisseurs, *etc.*). En effet, le consommateur demande des services rapides et fiables, tout en ayant des besoins très changeants. Les entreprises doivent répondre à ces exigences pour réussir à suivre les énormes variations des tendances du marché qui peuvent parfois survenir en quelques jours ou semaines. Dans ces circonstances, des systèmes d'information (SI) efficaces sont essentiels pour supporter de telles transformations stratégiques et structurelles. Ceci est rendu possible par le développement de nouvelles technologies tel que les "systèmes de gestion de processus métier" et le "cloud computing". Les tâches d'une entreprise peuvent être gérées de manière automatique et optimisée en s'appuyant sur des systèmes modulables et qui s'ajustent aux besoins immédiats. Ainsi, les entreprises ne deviennent pas seulement de plus en plus automatisées mais aussi globalisées en dépendant de services mondialement distribués et interconnectés.

En conséquence, la perte de contrôle sur leurs activités, leurs informations ou leurs processus devient une réelle menace pour toute entreprise. Le contexte veut qu'un petit changement à n'importe quel point de la chaîne de services, peut affecter de façon importante de processus global. Surtout dans des environnements cloud, des problèmes tels que *l'interruption de la disponibilité d'une infrastructure*, *le vol ou la perte de données personnelles* ou *des incohérences entre juridictions distinctes* doivent être considérés lors d'une externalisation d'applications métiers. Des situations peuvent apparaître où l'intégralité du savoir-faire et les données d'une entreprise sont gérées par une entité externe. Et vu que la chaîne de service complète n'est pas totalement sous contrôle, la continuité d'un service dépend de tous les participants et ne peut être gérée indépendamment par un seul acteur. Traiter ces risques est crucial: c'est seulement en anticipant de possibles défaillances qu'il est possible de garantir le fonctionnement du service. Mais réaliser des prédictions dans de tels environnements est complexe et dépend souvent de la confiance que l'on a envers les différents participants. La confiance étant un concept difficile à définir au travers de contrats, de standards ou de cadres juridiques, les processus doivent être adaptés pour gérer les situation incertaine, c'est-à-dire devenir "sensibilisés aux risques".

Dans cette thèse nous abordons le problème de l'adaptation de processus métiers aux enjeux de sécurité rencontrés lors de leur déploiement dans un environnement cloud. Nous adoptons une

approche basée sur le risque pour répondre au problème d'incertitude lié à ce contexte. Cela nous permet d'évaluer la sécurité de différents fournisseurs de services cloud et adapter les processus en conséquence.

1.2 Problématique, questions et méthodologie de recherche

Nous formulons notre problématique de recherche ainsi: **Comment prendre en compte les risques de sécurité lors d'une migration d'applications métiers vers un environnement cloud ?**. Cette problématique nous amène à répondre à quatre questions de recherche principales:

RQ.1 Quels sont les enjeux de sécurité liés au cloud ? Cette question nous amène à nous focaliser sur ce qui est *nouveau* lors d'une externalisation vers le cloud en comparaison à un contexte classique. Cela implique de définir clairement en quoi consiste un environnement cloud et ce qui le différencie d'un système d'information traditionnel. Cette question est abordée dans l'état de l'art (Chapitre 2).

RQ.2 Comment évaluer les risques de sécurité cloud ? Cette question soulève deux autres questions: Comment ces risques peuvent-ils impacter les activités d'une entreprise ? Et quelles sont les informations nécessaires de la part des fournisseurs de service cloud ? L'objectif ici est de *mesurer* la sécurité de systèmes basés sur des services cloud. Cette question est abordée dans Chapitre 4.

RQ.3 Comment gérer les risques de sécurité liés au cloud ? Il existe déjà des approches pour gérer les risques de sécurité dans des systèmes d'information classiques. Il est nécessaire d'étudier si elles sont toujours adaptées dans un contexte cloud. Plus précisément nous étudierons comment et par qui ces risques peuvent être gérés sur les processus métier. Cette question est étudiée dans Chapitre 3.

RQ.4 Comment intégrer le risque avec d'autres paramètres ? Le critère de sécurité doit être combiné avec d'autres paramètres qui motivent la décision d'une externalisation vers le cloud. Le coût, la qualité de service et d'autres exigences fonctionnelles sont souvent plus importants que la sécurité. Notre approche basée sur le risque nous aidera à définir une méthode pour essayer de trouver un équilibre entre la sécurité et d'autres critères, et ce de façon automatisée (Chapitre 5).

Pour répondre à ces questions et adresser la problématique de cette thèse, nous appliquons une méthodologie de recherche basée sur la *design science* (science de la conception, [MS95, Wie09]). Cette approche peut-être séparée en trois cycles de recherche:

- **Le cycle de pertinence** qui connecte les activités de recherche à l'*environnement*. Il consiste à fournir les besoins (*exigences métier*), donc les motivations à la recherche et de vérifier que les outils ou les méthodes conçues répondent correctement à ces besoins.
- **Le cycle de rigueur** qui connecte les activités de recherche à la *base de connaissances*. D'abord un état de l'art permet de s'assurer d'être aligné avec l'existant et ensuite la dissémination des résultats complète la base de connaissance.
- **Le cycle de conception** forme la partie centrale de la *design science research*. Elle consiste à construire des modèles, des méthodes ou des outils (en s'appuyant sur les besoins métiers et les fondations scientifiques) qui peuvent être évalués par de la simulation ou des cas d'études avant leur publication (ou leur application).

1.3 Résumé des contributions

Nos contributions principales peuvent être limitées à trois artefacts distincts du **cycle de conception** sous la forme d'une méthode, d'un modèle et d'un framework. La faisabilité de ces trois artefacts a été évaluée à l'aide d'un exemple illustratif et d'une implémentation.

- Une méthode pour sécuriser des processus métier avant de les déployer dans le cloud (Chapitre 3). Cette contribution répond à **RQ.3**.
- Un modèle pour évaluer les risques de sécurité dans des environnements cloud (Chapitre 4). Cette contribution répond à **RQ.2**.
- Un framework pour déployer des processus métier dans un environnement multi-cloud en prenant en compte plusieurs paramètres (Chapitre 5). Cette contribution répond à **RQ.4**.

Pour s'assurer d'une construction rigoureuse de nos artefacts, nos avons effectué un état de l'art dans Chapitre 2. Cette partie répond à **RQ.1**. La seconde partie du **cycle de rigueur** a été faite au travers de différentes publications:

- La méthode pour sécuriser des processus métier avant de les déployer dans le cloud (Chapitre 3) a été publié dans [GMG13] et dans [GMG14].
- Le modèle pour évaluer les risques de sécurité dans des environnements cloud (Chapitre 4) a été publié dans [GDG+14].
- Le framework pour déployer des processus métier dans un environnement multi-cloud en prenant en compte plusieurs paramètres (Chapitre 5) a été publié dans [GFG13] et dans [GDGG14].

Pour identifier les *exigences métier* du **cycle de pertinence**, l'état de l'art (Chapitre 2) prend en compte des références industrielles. Nos artefacts ont été appliqués à l'*environnement* de deux façons:

- Le modèle pour évaluer les risques de sécurité dans des environnements cloud a été appliqué à un cas d'utilisation pour évaluer de vrais fournisseurs cloud (Section 6.3).
- Le modèle a été appliqué dans un domaine différent de celui des processus métier dans [BGPG15] pour améliorer le contrôle d'accès (Section 6.2).

1.4 Exemple illustratif

Pour illustrer les contribution de la thèse nos décrivons un exemple utilisé tout au long de la thèse.

Il s'agit d'une entreprise fictive qui vend des produits à des clients européens. L'entreprise possède un site internet où tous les produits sont exposés, elle possède également une large base de données clients due à son expérience de près de 15 ans. Aujourd'hui l'entreprise fait face à un problème de concurrence, beaucoup d'acteurs ont émergés avec des sites en-ligne de haute disponibilité et des services de livraison très rapides. Pour recentrer leur activité sur leur cœur de métier, l'entreprise décide de considérer les technologies cloud. L'entreprise pense que l'externalisation de leur infrastructure et leurs applications métiers pourrait réduire leur coût d'exploitation. De plus, le cloud permettrait de cibler plus facilement des clients internationaux dû au manque d'élasticité de l'infrastructure actuelle. Cependant, l'actualité concernant les enjeux de sécurité liés au cloud pousse l'entreprise à prendre en compte les risques d'une telle externalisation.

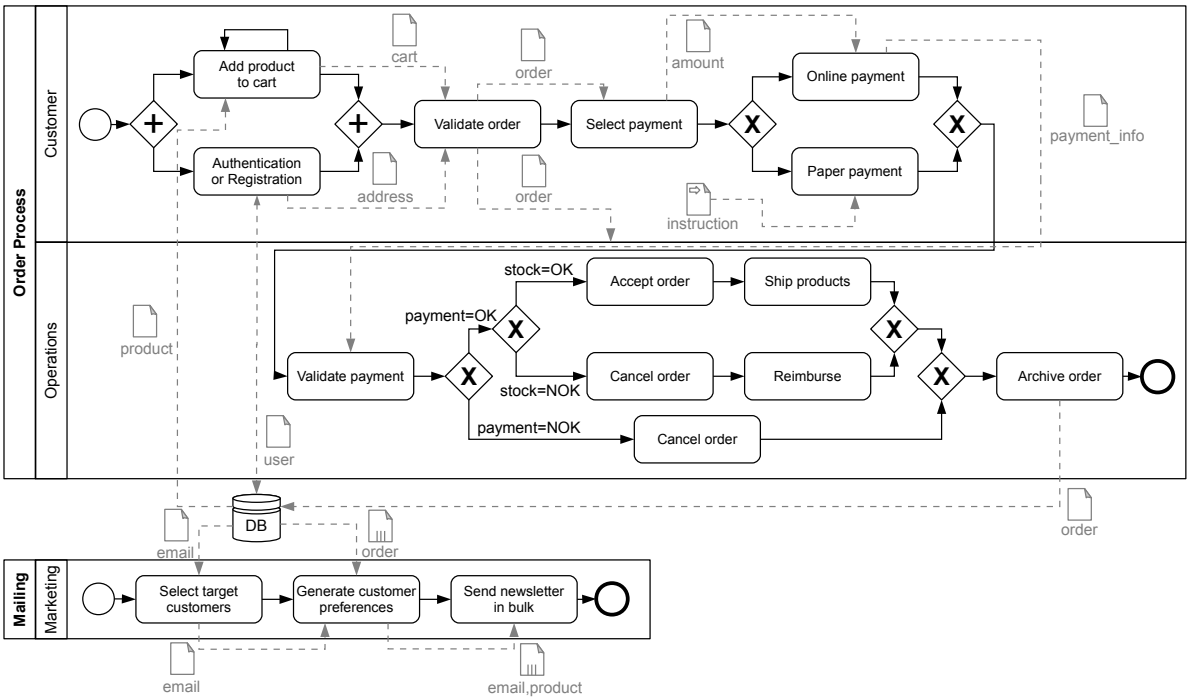

FIGURE 1.1 – Illustrating example - Business processes candidate for being outsourced

Les applications candidats à l'externalisation peuvent être résumés en deux processus métier illustrés dans FIGURE 1.1 en utilisant la notation BPMN 2.0. il s'agit d'un processus de commande intégrant l'interaction du client et un processus de newsletter.

Chapitre 2

Etat de l'art

2.1 Cloud Computing: l'informatique en "nuage"

Historiquement, le cloud a ses origines dans les années 50 quand les premiers ordinateurs accessibles à distance sont apparus (*mainframes*). L'idée était de partager les ressources disponibles et de les utiliser de façon optimisés [Wik13]. Aujourd'hui, le cloud peut être vu comme l'aboutissement des différentes évolutions technologiques de ces dernières années [BKNT11]. Les différentes avancées comme internet, la fibre optique, la virtualisation ainsi que la définition de différents standards poussent à percevoir les technologies de l'information comme une entité unique accessible à la demande, en libre-service, partagé et configurable. D'un point de vue purement hardware, le cloud peut être perçu comme *l'illusion de ressources informatiques infinies* [AFG+09].

2.1.1 Définition

Une définition un peu plus formelle du cloud est donnée dans [MPR+09]:

> *Le cloud est une forme de performances IT dirigées par la demande et flexible. Celles-ci sont disponibles en temps-réel en tant que services sur internet et facturé en fonction de l'utilisation. Ainsi, le cloud permet à ses utilisateurs de transformer des dépenses d'investissements en coûts opérationnels.*

Cette définition met bien en évidence que la "nouveauté" du cloud n'est pas au niveau technologique mais qu'il transforme la façon dont les technologies de l'information sont utilisées. C'est pourquoi on entend souvent le terme d'*énergie numérique* pour faire le parallèle avec l'électricité.

Une liste des caractéristiques essentielles au cloud sont faites dans [MG11] et [AFG+09]:

- **libre-service à la demande** - les ressources sont disponibles à la demande du client, sans interaction humaine ou re-négociation.
- **vaste accès réseau** - une connexion internet est requise (ou au moins un accès réseau, dans le cas d'un cloud local).
- **mise en commun de ressources** - ou virtualisation, les ressources physiques sont partagées avec d'autres clients et allouées dynamiquement.
- **élasticité rapide** - les ressources peuvent être échelonnées rapidement (et automatiquement).
- **services mesurés** - ou "pay-as-you-go", le client paye uniquement ce qu'il utilise.

2.1.1.1 L'architecture cloud - Les 3 niveaux de service

L'architecture cloud peut être séparée en trois niveaux (voir 2.1).

SaaS	messagerie web, applications internet, bureaux virtuels, jeux, etc... Exemples: Amazon EC2, Rackspace Cloud, Joyent
PaaS	environnement d'exécution, serveurs web, bases de données, etc... Exemples: Google App Engine, Cloud Foundry, Force.com
IaaS	machines virtuelles, serveurs, systèmes de stockage, réseaux, etc... Exemples: Google Docs, Microsoft Office 365, Dropbox

FIGURE 2.1 – Les trois niveaux de service cloud

- **Infrastructure as a Service (IaaS)** - le niveau le plus bas, qui vise à fournir les ressources physiques à la demande (donc serveurs, machines virtuelles, réseaux, *etc.*).
- **Platform as a Service (PaaS)** - le second niveau, qui vise à fournir des environnement de développement, d'exécution ou de test. Typiquement, les *moteurs d'exécution de processus métier* se trouvent à ce niveau.
- **Software as a Service (SaaS)** - le niveau le plus haut, qui vise à fournir des services comme des logiciels à un utilisateur final. La plupart du temps au travers d'interfaces web.

2.1.1.2 Modèles de déploiement

En général on considère trois types de modèles de déploiement pour le cloud (voir 2.2).

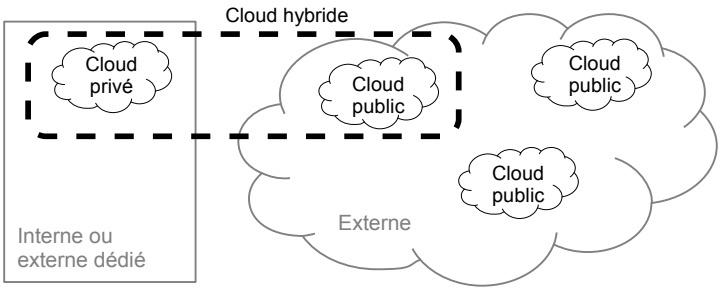

FIGURE 2.2 – Modèles de déploiement cloud

- **Public** - le type de cloud qui est accessible à tous. En général ce sont les offres les plus intéressante financièrement.
- **Privé** - le type de cloud dédié à une personne/entreprise, qui peut être interne ou externe. Ces offres proposent le plus de libertés de configuration.
- **Hybride** - combine les deux modèle précédents pour permettre un déploiement par "fragments".

2.1.1.3 Les acteurs du cloud

Différentes études proposent d'identifier les acteurs du cloud ([LTM+11, MLB+11]). Nous en identifions trois principaux:

- **Le consommateur cloud** - Le NIST [LTM+11] le définit comme "une personne ou une organisation qui maintiens une relation commerciale avec un *fournisseur cloud* en utilisant ses services".
- **Le fournisseur de service** - "Personne, organisation ou entité responsable de rendre les services disponible aux parties intéressées" ([LTM+11]).
- **Le cloud broker** - "Une entité qui gère l'utilisation, la performance et la livraison de services cloud et négocie les relations entre le *consommateur* et le *fournisseur*" ([LTM+11]).

Il est également possible d'identifier d'autres acteurs comme les **auditeurs** ou les **régulateurs**. Cependant ils n'ont pas un grand intérêt dans le cadre de cette thèse.

2.1.2 Les enjeux du cloud

Le cloud computing est encore en pleine émergence, Gartner [Gar13] estime qu'entre 2013 à 2016 près de 677 milliards de dollars seront dépensés au niveau mondial pour des services cloud. Cependant, beaucoup de critiques se font entendre, comme celles avancées par Steve Wozniak [Woz12] ou Richard Stallman [Sta08]. En effet, l'un des problème majeur est que le système d'information n'est plus complètement sous contrôle. Il est donc plus difficile de contraindre un système distant qu'une infrastructure locale. Différentes études ont été faites pour recenser les défis liés au cloud ([GMR+12, AFG+09, CBT11, MLB+11]). Dans ce sens, la Cloud Security Alliance (CSA) a réalisé une liste des menaces liées au cloud les plus importantes [CSA13]. On peut notamment y trouver des menaces comme **le vol ou la perte de données**, **le déni de service**, **l'employé malicieux** ou **les interfaces non sécurisées**.

2.2 Gestion des Risques de Sécurité des Systèmes d'Information

Gérer les risques sur un système d'information est essentiel pour garantir sa sécurité tout en contrôlant les coûts. Différents standards ou méthodologies existent pour aider les entreprises dans cette tâche. L'idée initiale derrière la gestion des risques et de partir du principe qu'un système totalement sécurisé est soit impossible soit beaucoup trop onéreux à réaliser. C'est pourquoi un processus de gestion des risques aide à classer les risques par leur importance et définit lesquels doivent être traités pour qu'ils deviennent *acceptable*. Ainsi, ces approches rendent possible de *mesurer* la sécurité de façon qualitative ais aussi quantitative au travers des coûts.

2.2.1 Définition

Grosso modo, un risque est défini comme la combinaison de la probabilité qu'un événement se produise et ses conséquences [NIS02]. Dans le contexte de sécurité IT, où des composants (*e.g.*, hardware, réseau, *etc.*) supportent des actifs métier (*e.g.*, informations, processus, *etc.*), le risque de sécurité est défini de façon plus précise. L'événement est généralement vu comme une menace qui exploite une ou plusieurs vulnérabilités des composants IT pour créer un impact négatif (*e.g.*, destruction, altération, vol, *etc.*) sur les actifs métiers [May09]. Par exemple, un attaquant vole des données clients (*i.e.* **menace**) au travers une interface compromise (*i.e.* **vulnérabilité**) ce qui entraîne une perte de réputation pour l'entreprise (*i.e.* **impact**).

2.2.1.1 Vocabulaire - Le modèle ISSRM

Ainsi, évaluer un risque de sécurité consiste usuellement à évaluer la formule suivante ([AZ/04, NIS02]):

$$Risque = Vulnérabilité \times Menace \times Impact \tag{2.1}$$

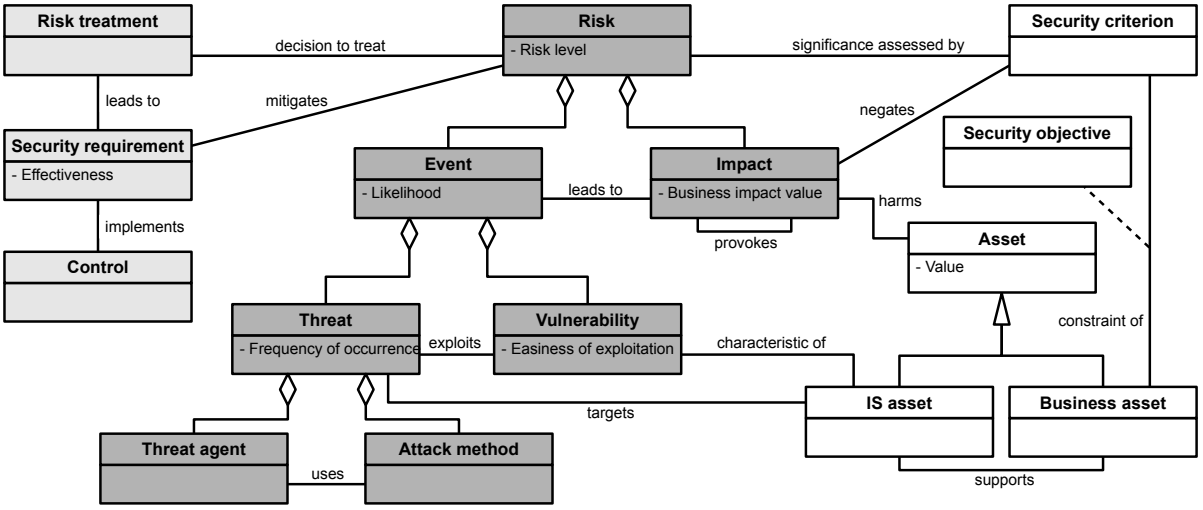

FIGURE 2.3 – ISSRM Domain Model

FIGURE 2.3 définit de façon plus précise un risque de sécurité et relie aux autre composants/éléments qu'il affecte ou qui peuvent l'influencer. Ce modèle a été présenté par Mayer *et al.* dans [MMM+08].

2.2.1.2 Le processus de gestion de risques générique

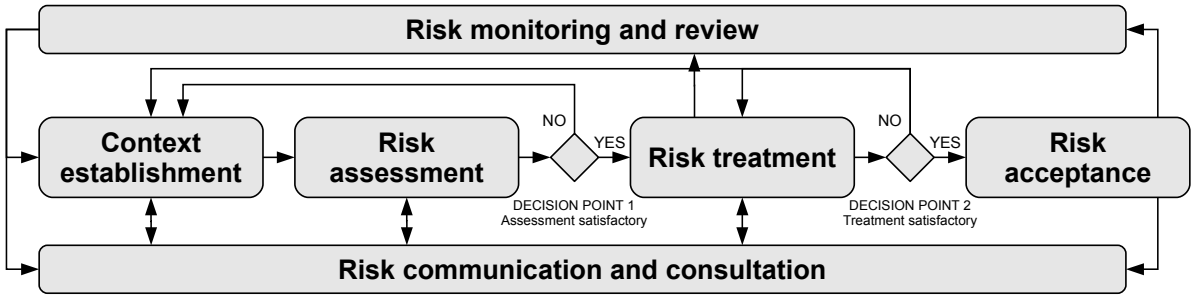

FIGURE 2.4 – Processus de gestion de risques générique

Le processus de gestion des risques peut être généralisé à celui présenté dans FIGURE 2.4. Il est composé des activités suivantes:

- **Établissement du contexte** de l'entreprise, incluant le cadre, les objectifs et définissant clairement quels critères seront utilisés.
- **Évaluation des risques**, qui consiste donc à identifier les sources de risque, leur probabilité et à estimer leur conséquence.
- **Traitement des risques** au travers de la sélection de la stratégie la plus adaptée.

- **Acceptation des risques** (dits “résiduels”) par les responsables de l’entreprise.

2.2.1.3 Les stratégies de traitement des risques

Il existe quatre grandes catégories de stratégies pour traiter les risques [ISO11]:

- **Modification**, le système est modifié en ajoutant ou modifiant certains éléments (des *contrôles de sécurité*).
- **Rétention**, le risque est accepté tel quel sans aucune action supplémentaire.
- **Évitement**, le système (ou même l’activité de l’entreprise) est modifié de telle façon à ce que le risque n’existe plus.
- **Partage**, le risque est partagé avec une tierce partie (comme par exemple une assurance).

2.2.2 Gestion des risques de sécurité dans le cloud

L’étude la plus complète concernant les risques de sécurité liés au cloud a été faite par l’Agence Européenne de la Sécurité Réseau et des Informations (ENISA) [ENI09a]. Ce rapport donne une liste de 35 risques lié à l’utilisation de services cloud et les classe dans quatre catégories principales: **les risques stratégiques et organisationnel**, **les risques techniques**, **les risques juridiques** et **les risques non spécifiques au cloud**. Cette liste est relativement complète et est régulièrement utilisée comme référence même dans le domaine académique.

Le problème majeur lors de la réalisation d’un processus de gestion des risques dans un environnement cloud est que le système n’est jamais entièrement sous contrôle. Il est donc difficile d’accomplir un processus de gestion classique pour deux raisons principales.

- **La perte de contrôle** - L’architecture est définie et gérée par le fournisseur cloud. Le consommateur cloud ne peut pas implémenter les contrôles de sécurité qu’il considère comme étant le plus efficace. Il doit se contenter de ce que le fournisseur lui propose et il se peut qu’il doit accepter certains risques tel quel. Ainsi, l’une des première étape à faire avant de migrer vers le cloud et d’étudier si cette migration doit être faite ou non. Il existe des outils à ces fins tel que le Cloud Securty Readiness Tool [1] de Microsoft. Une autre solution et de jouer sur l’**impact** d’un incident de sécurité. Jensen *et al.* [JSB+11] par exemple propose de fragmenter les applications sur de multiples clouds.
- **Le manque d’information** - Identifier les vulnérabilité d’un système peut devenir très complexe vu que les solutions techniques choisies par le fournisseur ne sont pas accessibles pour le consommateur et sont parfois même volontairement dissimulées par le fournisseur (et ce pour des raisons de sécurité). Différents standards proposent de palier à ce problème en proposant des métriques: la Cloud Security Alliance propose la **Cloud Control Matrix** [CSA14], l’ENISA propose l’**Information Assurance Framework** [ENI09b], Eurocloud propose le **Star Audit** [Eur12]. Et il existe aussi le **Common assurance Maturity Model** [CAM10] et l’**ISO 27017** [ISO15].

2.3 Gestion des processus métiers

La gestion des processus métier est le domaine qui vise à définir, gérer et améliorer les processus d’une entreprise pour livrer des produits adaptés aux besoins de ses clients. Les processus métier

1. `http://www.microsoft.com/trustedcloud`

peuvent être décrits de façon formelle au travers de modèles de processus. Ces modèles ont souvent une représentation graphique (comme pour le BPMN [OMG11]). Lorsque ces processus sont limités à des applications logicielles ils peuvent être exécutés automatiquement, avec ou sans interaction humaine et ce en utilisant des services IT. Ces services peuvent être accessibles de façon locale ou à distance (au travers de web-services).

2.3.1 Définition

La gestion des processus métiers fait généralement référence au cycle de vie présenté dans FIGURE 2.5. Les modèles représentant les processus métier évoluent au cours de ce cycle de vie, il y a donc différents *niveaux d'abstraction* pour les représenter (aussi appelés *perspectives*)

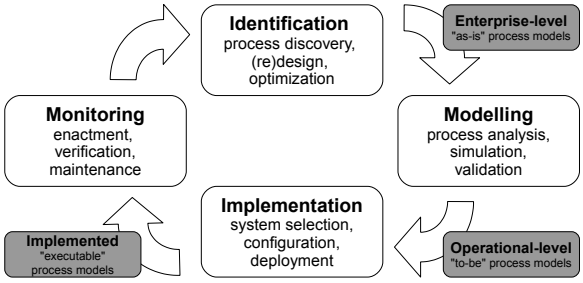

FIGURE 2.5 – Cycle de vie des processus métiers inspiré de [DRMR13], [Wes12]

2.3.1.1 Le cycle de vie

Le cycle de vie se compose en quatre phases:

- **Identification**, dans laquelle l'environnement technique et organisationnel est analysé pour découvrir les processus de l'entreprise.
- **Modélisation**, dans laquelle les processus sont représentés graphiquement pour détailler les informations identifiées à la phase précédente.
- **Implémentation**, dans laquelle les modèles abstraits sont transformés en processus exécutables (donc développés dans le cas de processus IT).
- **Monitoring**, dans laquelle les processus sont exécutés et contrôlés pour vérifier qu'ils sont bien conformes aux exigences initiales.

2.3.1.2 Les 3 niveaux BPM

Dans la littérature on identifie souvent trois niveaux de processus métiers, même si le contenu de ces niveaux n'est pas toujours identiques [MAA11]. Dans cette thèse nous utilisons ceux de Ahmed *et. al* [AM13] qui sont proches de ceux donnés par Dreiling *et al.* [DRA05].

- **Niveau entreprise** - Des processus avec un haut niveau d'abstraction qui relient une entreprise à son environnement (partenaires, *etc.*). Ils servent à décrire les entrées et les sorties des processus et relient les processus entre eux.
- **Niveau opérationnel** - Des processus qui décrivent les activités et leurs relations nécessaires à réaliser les fonctions du métier. Ils sont modélisés de façon plus détaillés mais ne tiennent pas compte de leur implémentation.

- **Niveau implémentation** - Ces processus sont les spécifications techniques nécessaires à la réalisation des activités des processus. Dans un environnement IT il s'agit des composant techniques supportant l'exécution du processus.

Il existe différents langages de modélisation de processus métier permettant la description (graphique ou non) de tels processus. Quelques exemples: **UML** [OMG13], **BPMN** [OMG11], **BPEL** [OAS07] ou encore **YAWL** [AAHR10].

2.3.1.3 Architecture de référence

L'architecture de référence d'un système d'information supportant l'exécution de processus métiers est présentée dans FIGURE 2.6.

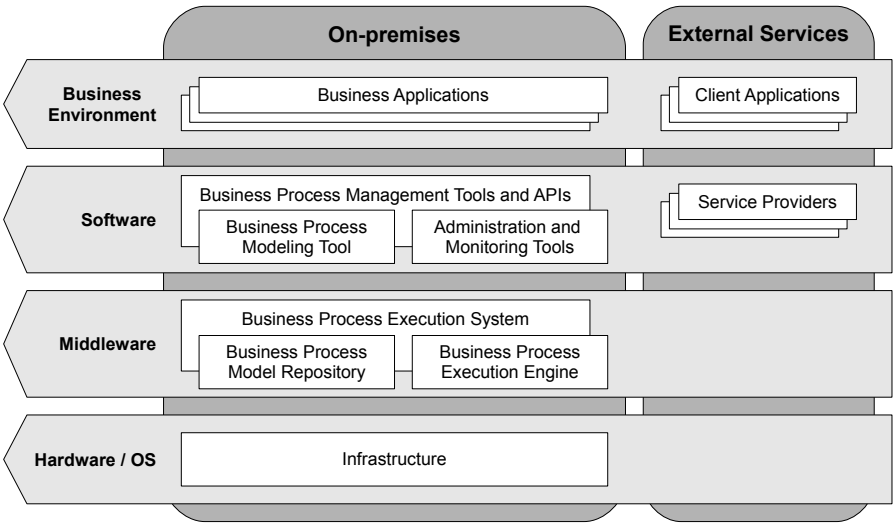

FIGURE 2.6 – Architecture de référence d'un système de gestion de processus métiers (de [DRMR13], [Wes12])

2.3.2 Gestion de processus métiers, cloud et risques de sécurité

Dans cette partie nous étudions les intersections respectives entre chacun de ces trois domaines de recherche.

2.3.2.1 Gestion de processus métiers et cloud

Il existe différentes propositions d'architecture pour le cloud [JLW+11, FY10]. Cependant la plus complète est présenté dans [ALMS09]. Typiquement il existe trois cas possibles:

- **IaaS** - Le service cloud ne fournit que l'infrastructure. Le moteur d'exécution et le système de gestion sont contrôlés par le consommateur. Un exemple d'un tel cas est donné dans [MJ10].
- **PaaS** - Le service cloud fournit l'infrastructure et le moteur d'exécution. Le client contrôle le système de gestion de processus. Un exemple d'une telle implémentation est donné dans [PPKW11]. D'autres sont données dans [MR14, EJF+14] ou encore dans [MF12].

- **SaaS** - Le service cloud fournit directement les processus (on parle aussi de BPaaS). Les problématiques d'une telle approche est décrite dans [Aal11]. [BBDA12] présente les avantages en terme de ré-utilisabilité de ce type d'architecture.

2.3.2.2 La sécurité dans la gestion des processus métiers

Beaucoup de travaux existent pour prendre en compte la sécurité lors de la définition de processus métiers. Ils peuvent être regroupés dans quatre grandes catégories.

La modélisation d'exigences de sécurité dans les processus métiers Les auteurs de [KR01] proposent dès 2001 de considérer des niveaux pour les *objectifs de sécurité* lors de la définition de processus métiers. Une extension à UML, UMLsec [Jür02] propose également d'inclure des propriétés de sécurité dans les modèles de processus. Les auteurs de [TBCB12] proposent une extension à BPMN pour modéliser des contraintes de sécurité. Des travaux similaires sont présentés dans [RFMP07]. Une autre approche utilisant des perspectives pour définir des exigences de sécurité est donnée dans [PGPM12].

Des exigences de sécurité vers leur implémentation Les auteurs de [WMM08] ajoutent des objectifs de sécurité sous forme d'annotations sur des processus métier. Ils sont ensuite capables de générer des politiques de sécurité dans différents langages [WMS+09]. Les auteurs de [TJG+11] définissent des processus sensibilisés aux risques et sont capables de simuler leur conformité. Un autre type d'approche est décrit dans [MSSN04] qui propose d'extraire d'un processus existant une politique de contrôle d'accès.

Adapter les processus métiers aux enjeux de sécurité Une approche consiste à adapter directement les processus en fonction des problèmes de sécurité qui peuvent exister. Les auteurs de [AM14] proposent des *patrons* de sécurité qui peuvent être intégrés dans un processus. Dans [Fil12] une idée intéressante est exploré qui consiste à offusquer les processus en cachant les informations sensibles qui peuvent être extraits directement du modèle.

Exécution sécurisé de processus dans un environnement cloud Les auteurs de [CFBH07] définissent une approche pour sélectionner le web-service le plus adapté en terme de sécurité. Watson [Wat12] propose différents niveau de sécurité pour caractériser les offres cloud et sélectionner la plus appropriée. Une autre approche présenté dans [SLK09] se base sur les vulnérabilités publiées dans le CVSS [2]. Les auteurs de [OBG13] proposent une approche dirigée par les modèles pour garantir une exécution sécurisée de processus dans un environnement cloud.

Autres travaux reliés D'autres travaux considère d'autres types de risques, comme dans [CLRA13] qui considère les erreurs durant l'exécution d'un processus. Les travaux présentés dans [MH06] considèrent les risques pouvant entraîner une mauvaise définition d'un processus. Les auteurs de [WWHJ12] analysent la descriptions de tâches dans un processus pour déterminer lesquelles peuvent être externalisés.

A première vue, la combinaison de ces trois domaines de recherche, gestion de processus métiers, cloud computing et gestion des risques de sécurité devrait aider à construire des systèmes d'information en automatisant la considération de la sécurité. Cependant, aucune approche existante intègre

2. `http://nvd.nist.gov/cvss.cfm`

tous les aspects de ces trois domaines. Les travaux existant couvrent un ou deux aspects de notre problématique de recherche, mais jamais la problématique complète.

Chapitre 3

Alignement des domaines et considérations méthodologiques

Ce chapitre est principalement basé sur les contributions publiées dans [GMG13] et dans [GMG14]. Il consiste à définir un vocabulaire commun aux trois domaines étudiés et définir une approche généralisée pour gérer les risques de sécurité liés au cloud sur des processus métiers.

3.1 Distribution du cycle de vie et des niveaux de BPM

Dans FIGURE 3.1 nous proposons un alignement des trois domaines étudiés en répartissant le cycle de vie des processus métiers aux différents acteurs du cloud. Ainsi nous pouvons identifier les type d'activités de traitement de risque que chacun des acteur peut effectuer.

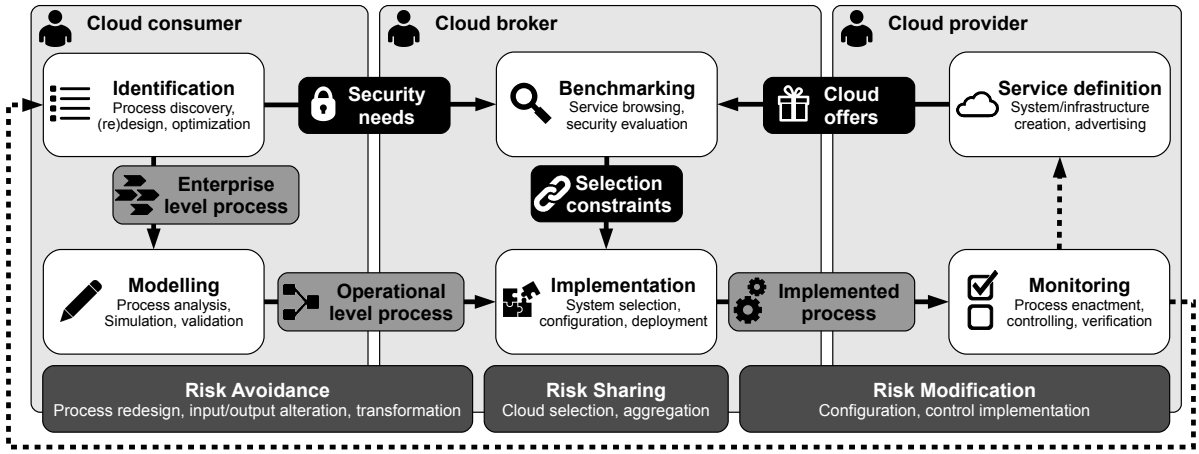

FIGURE 3.1 – Cycle de vie des processus métiers réparti aux acteurs du cloud et les activités de traitement de risque associées

Typiquement, le **consommateur cloud** est responsable d'**identifier** et concevoir les **processus niveau entreprise**. Ceux-ci sont ensuite transformés en **processus opérationnels** au travers d'une **modélisation**. Ces modèles sont transmis, avec les **exigences de sécurité**, au **cloud broker**. Ces exigences lui servent à réaliser une **analyse comparative** des **offres cloud** existantes de laquelle il extrait des **contraintes de sélection**. Le broker utilise ces contraintes avec le modèle de processus

opérationnel le déployer dans le cloud (ce qui correspond à une **implémentation**). Les **processus implémentés** sont **vérifiés** par le **fournisseur cloud** pour s'assurer d'une exécution conforme.

Cet alignement permet d'identifier la répartition des activités de traitement de risques.

3.1.1 Le fournisseur cloud

Il peut **modifier/réduire** les risques en implémentant des contrôles de sécurité pour ses différentes offres. Typiquement, il va analyser son infrastructure et ses services pour identifier les vulnérabilités qui peuvent menacer ses clients. Cependant il ne va pas le faire pour chacun de ses client, mais de façon générique. Ces mesures de sécurité deviennent donc plutôt des caractéristiques d'une offre de service plutôt que des contrôles spécifiques à un processus. D'un point de vue du consommateur, il n'est donc plus possible comme dans une gestion de risque classique de contraindre les actifs de support par des mesures techniques. Le consommateur doit se satisfaire de ce que le fournisseur cloud lui propose, ou alors il devra choisir une offre alternative qui lui conviendra mieux.

3.1.2 Le consommateur cloud

Il peut **éviter** certains risques en modifiant son processus. L'une des façons les plus évidentes est de prendre la décision de ne pas migrer ses applications vers des offres cloud. Aussi, sélectionner un fournisseur plutôt qu'un autre peut également être perçu comme de l'évitement: les risques de l'un sont évités au profit de l'autre. De plus, le consommateur cloud peut définir un certain nombre de contraintes qui limiteront la sélection des offres de services cloud. Nous proposons la catégorisation suivante:

- les contraintes **logiques** - ou des contraintes fonctionnelles qui impactent le *flux de contrôle* du processus. C'est la logique du processus qui est changé par rapport à son fonctionnement initial: on ajoute (ou on retire) certaines tâches. Des exemples sont: **séparer les connaissances en plusieurs fragments**, **séparer la logique des données**, **grouper les connaissances dans un même fragment**, **répliquer des tâches** ou encore **ajouter des tâches de gestion de sécurité**.
- les contraintes **organisationnelles** - relatif à l'allocation des tâches d'un processus (qui fait quoi). Ces contraintes permettent de définir des exigences affectant la manière dont le processus sera déployé. Des exemples sont: **séparation de fragments**, **co-localisation de fragments** ou encore **imposer la rétention d'un fragment en local**.
- les contraintes **informationnelles** - dépendent directement des informations émanant du contexte cloud (et donc des offres de services cloud). Ces contraintes vont directement affectées la sélection des offres cloud. Des exemples sont: **exclure une offre**, **imposer l'utilisation d'une offre** ou encore **imposer un niveau de sécurité minimal**.

3.1.3 Le broker

Il peut **partager** les risques sur de multiples offres cloud. Il s'agit là de la réelle valeur ajoutée du broker, il peut aider le consommateur cloud à sélectionner la ou les offres de services les plus appropriés à ses besoins. À l'inverse, il peut aussi conseiller les fournisseurs cloud sur les contrôles de sécurité qu'ils devraient implémenter pour satisfaire un maximum de clients. En principe, le broker ne peut ni modifier le processus, ni implémenter des contrôles de sécurité, vu qu'il ne contrôle pas les offres directement. Cependant, dans certaines configuration de déploiement, il peut modifier certains risques:

- dans le cas de l'utilisation d'une offre **IaaS** il peut modifier les vulnérabilités émanant du *moteur d'exécution de processus*. Par contre, modifier les risques de la couche *infrastructure* est du ressort du fournisseur.
- dans le cas de l'utilisation d'une offre **PaaS** il peut contrer les vulnérabilités émanant de la couche *application*. Par contre, celles venant des couches *infrastructure* et du *moteur d'exécution* doivent être gérées par le fournisseur.
- dans le cas de l'utilisation d'une offre **SaaS**, le broker ne peut pas contrer les vulnérabilités, vu qu'il n'a accès à rien.

3.2 Méthodologie

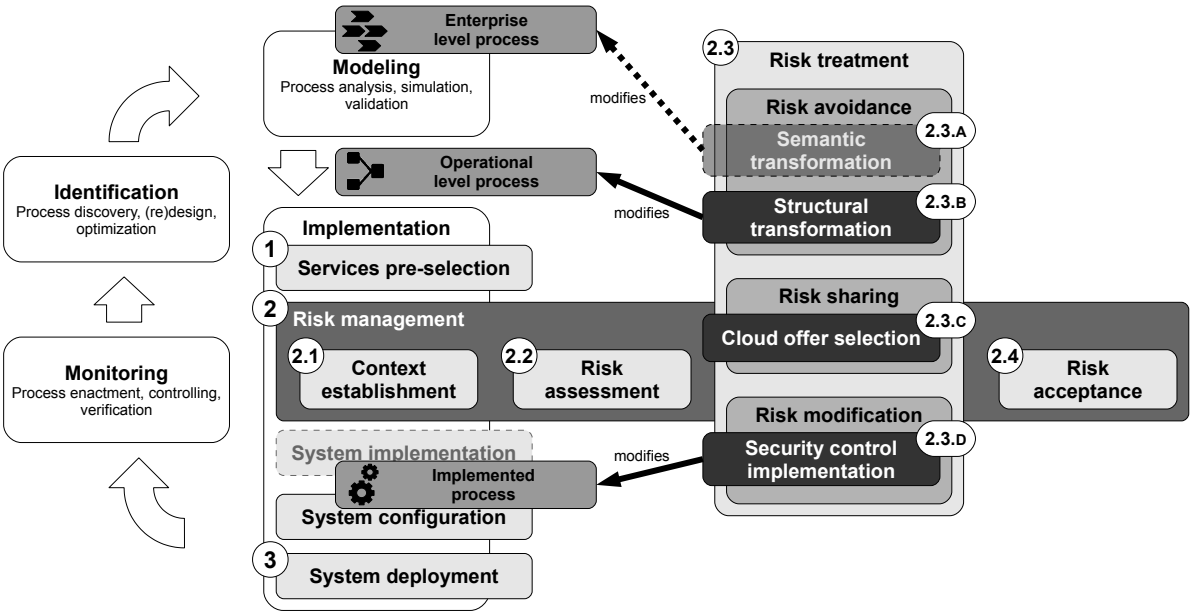

FIGURE 3.2 – Aperçu de la contribution globale

Comme le présente FIGURE 3.2, nous proposons d'aligner le cycle de vie avec les trois stratégies de traitement de risque et les différents acteurs du cloud. Ces acteurs interviennent de façon cyclique en suivant les différentes phases du cycle de vie.

Dans la suite nous adoptons le point de vue du broker pour identifier les différentes actions qu'il peut réaliser pour gérer les risques de sécurité émanant du cloud sur les processus métiers. Nous identifions cinq étapes majeures: *sélection des services*, *évaluation des risques*, *traitement des risques*, *acceptation des risques* et *déploiement*.

3.2.1 Pré-sélection des services et établissement du contexte

Le rôle principal du broker est de satisfaire les exigences fonctionnelles du consommateur cloud. C'est pourquoi il pré-sélectionne les offres de services adaptées à ses besoins. Certaines contraintes (comme les contraintes **informationnelles** peuvent déjà amener à exclure certaines offres). Cet ensemble d'offres forme le *contexte* de l'analyse des risques. Cela veut dire que toutes les offres pré-sélectionnées à cette étape devront être analysés quant à leur sécurité. Il faut bien noter que toutes

les offres ne seront pas forcément utilisées pour exécuter le processus, cependant elle doivent être considérées pour l'évaluation des risques. D'où l'intérêt d'une automatisation de l'évaluation des risques: le plus d'offres pré-sélectionnées, le plus il devient probable de trouver une configuration de déploiement suffisamment sécurisé.

3.2.2 Évaluation des risques

En prenant en compte les **exigences de sécurité** du consommateur cloud, le broker est capable de réaliser une évaluation des risques. Chaque couple (actif, service) est associé à une valeur de risque qui représente le risque de déployer l'actif sur le service donné. Ce qui veut dire que tous les actifs ne sont pas nécessairement déployés sur le même service. Cela va dépendre du niveau de détail des exigences: **global**, **par processus**, **par fragment** ou **par tâche/donnée**. Le plus détaillé ces informations sont données, le plus le broker va pouvoir fragmenter les processus pour les distribuer de façon optimisé.

3.2.3 Traitement des risques

Pour les stratégies de traitement des risques nous identifions quatre transformations différentes que le broker peut réaliser.

3.2.3.1 Transformation sémantique

Une façon d'**éviter** certains risques est de changer la sémantique du processus (la logique, ce que le processus fait réellement). Ce type de transformation modifie le **niveau entreprise** du processus. Ce qui veut dire que le fonction globale du processus est modifiée de façon à éviter certains risques émanant du cloud. Bien entendu, il ne s'agit pas ici de l'activité principale du broker, il ne peut pas modifier à lui seul la stratégie de l'entreprise. Cependant, il peut conseiller le consommateur cloud sur certains points, comme par exemple de limiter l'externalisation à certaines parties du système. Cette décision de transformation sémantique doit être prise conjointement entre le broker et le consommateur.

3.2.3.2 Transformation structurelle

Une autre façon d'influencer les risques est de changer la structure du processus (sans changer sa logique). Ce type de transformation change le **niveau opérationnel** du processus. Le fonction globale du processus ne change pas, mais la façon dont elle est réalisée l'est. Un exemple peut être la séparation d'une tâche en plusieurs activités et l'ajout d'une contrainte de séparation. Ou encore la réplication d'une certaine tâche pour augmenter la disponibilité du processus.

Une perspective intéressante donnée par la fragmentation est l'*offuscation* de processus. L'idée est de décomposer ou de déployer le processus d'une telle façon qu'il devient difficile de découvrir ce que le processus réalise.

3.2.3.3 Sélection des offres cloud

Il s'agit du réel cœur de métier du broker: comparer les offres cloud selon différents paramètres et de trouver la configuration de déploiement la plus adaptée. Vu que certaines offres peuvent avoir de meilleurs niveaux de sécurité que d'autres, il peut être intéressant de déployer le processus plutôt sur un endroit plutôt que sur un autre. Bien entendu, il peut être intéressant d'analyser les besoins plus en détail pour pouvoir fragmenter les processus et les déployer sur des services séparés. En

effet, toutes les tâches d'un même processus n'ont pas forcément les mêmes exigences en terme de sécurité.

3.2.3.4 Implémentation de contrôles de sécurité

Comme décrit précédemment, certaines configuration de déploiement permettent l'implémentation de contrôles de sécurité de la part du broker. Un exemple est la combinaison de plusieurs services (comme l'identification ou le chiffrement) pour ajouter des couches de sécurité. D'autre part, les options de configuration de chaque offre peuvent également correspondre à certaines possibilités d'amélioration de la sécurité (comme par exemple sécuriser toutes les communication par SSL).

Une perspective intéressant qui va dans ce sens est la stratégie de *chiffrement homomorphique*. Publié dans [Gen09] par Craig Gentry, ce type d'approche permet d'effectuer certains types de traitement directement sur des informations chiffrer (sans les déchiffrer). Les opérations peuvent donc être réalisées sans jamais dévoiler le contenu des informations.

3.2.4 Acceptation des risques

La dernière étape de la gestion des risques consiste à accepter les risques tel quel. Cette état peut être atteint après plusieurs itérations. En effet, à chaque transformation, le risque doit être ré-évalué. De façon habituelle on fixe un seuil d'acceptation sous lequel toutes les valeurs de risques doivent se trouver pour pouvoir procéder au déploiement.

Il est également possible que cet état ne soit pas atteint (le risque ne pouvant passer sous le seuil). Dans ce cas, soit le consommateur doit revoir à la baisse ces **exigences de sécurité**, soit l'externalisation vers des services cloud est complètement remise en question.

3.2.5 Déploiement

Une fois la solution de déploiement déterminée, les processus peuvent être déployées. Dans le cas d'un déploiement en fragment, les processus doivent être adaptés pour correspondre à la configuration sélectionnée. Ce type de décomposition automatique est présenté dans [FYG09].

Une fois déployés, les processus doivent être contrôlés (en adéquation avec la quatrième phase du cycle de vie des processus métiers), et ce pour trois raisons principales: améliorer les processus par la suite, vérifier la conformité de l'exécution et détecter les changements qui pourrait affecter le processus ou sa sécurité.

Chapitre 4

Evaluation des risques dans un contexte cloud

Ce chapitre est basé sur les contributions publiées dans [GDG+14]. Ce chapitre présente un modèle conceptuel ainsi que sa formalisation pour évaluer les risques de sécurité lié au cloud dans le cadre du déploiement d'un processus métier.

4.1 Aperçu

Notre approche se base sur les concepts présenté dans l'état de l'art: **impact**, **vulnérabilité** et **menace**. Nous partons du principe que dans le contexte du cloud il n'est pas possible pour l'un des acteurs de faire une analyse complète des risques. C'est pourquoi cette analyse doit être distribuée auprès de chacun des acteurs.

Ainsi, nous définissons que l'**impact** peut uniquement être déterminé par le **consommateur cloud**. Il est le seul à pouvoir estimer la *gravité* qu'un incident de sécurité aura sur ses actifs. La réalisation d'un événement affectera directement le processus, et la conséquence que cela peut avoir ne peut pas être estimée par le fournisseur cloud: il ne sait pas forcément dans quel objectif et pourquoi ses services sont utilisés.

De façon similaire nous définissons que la **vulnérabilité** est déterminé par le **fournisseur cloud**. En général, les services cloud correspondent à une boîte noire pour le consommateur, il n'a aucune idée de la façon dont laquelle le service est réalisé. Il est donc difficile pour lui d'analyser les vulnérabilités du système, comme il pourrait le faire s'il contrôlait la totalité de l'infrastructure.

Le **cloud broker** peut aider le fournisseur et le consommateur à déterminer les vulnérabilités et les impacts en définissant les **menaces** qui doivent être considérées dans le cadre de l'utilisation de services cloud. D'après les différentes études qui ont été menées sur les risques liés au cloud ([ENI09a, CSA13]) nous pouvons considérer que ce type d'information est indépendant des deux autres valeurs. Ces listes peuvent même être considérés comme étant statiques: elles s'appliquent systématiquement à toutes les offres dans le cas d'une externalisation.

Le modèle est décrit de façon plus précise dans FIGURE 4.1.

Le consommateur cloud définit des **objectifs** de sécurité à l'aide de **critères** de sécurité sur lesquels ils indique des valeurs (les **exigences**). Ces valeurs sont associées à tous les actifs considérés pour une externalisation.

Le fournisseur cloud définit des **contrôles** de sécurité qu'il peut **implémenter** sur ses **offres** de services. Ces contrôles sont des mesures de sécurité pour empêcher la réalisation de certains incidents de sécurité. Cette valeur peut être quantifiée par le **stage** d'implémentation.

Le broker cloud définit une liste de **menaces** qui doivent être considérées dans le cadre d'une externalisation cloud. Ces menaces ont toutes des **conséquences** sur les critères de sécurité. Cette relation peut être quantifiée par leur **sévérité** (certaines conséquences sont plus importantes que d'autres). D'autre part, ces menaces peuvent être **atténuée** par des contrôles de sécurité. Cette relation peut être quantifiée par le **degré** d'atténuation (certains contrôles sont plus efficaces que d'autres).

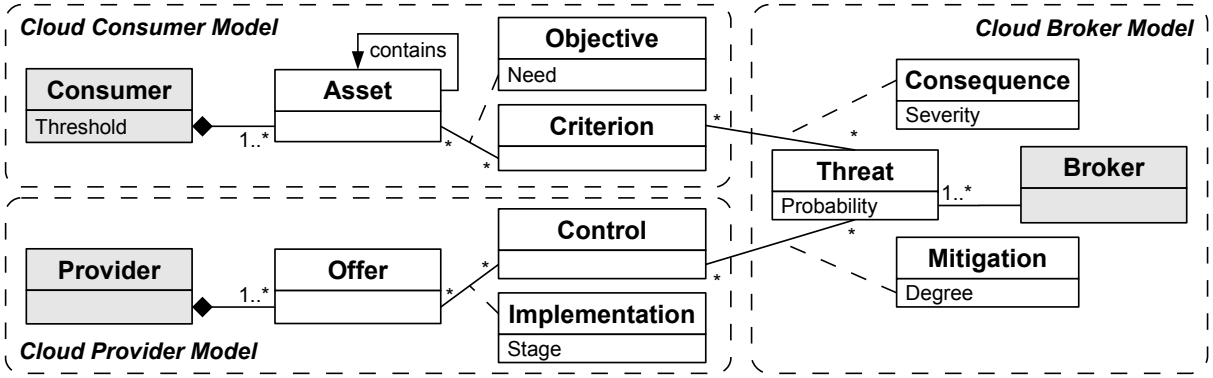

FIGURE 4.1 – Modèle d'évaluation des risques dans un contexte multi-cloud

Une évaluation des risques cloud résulte donc à une liste de triplet (actif, offre, menace) auxquels on associe une valeur de risque. Cette valeur représente le risque que la menace donnée se produise en déployant l'actif sur l'offre en question.

4.2 Modèle formel

Dans cette partie nous présentons une formalisation du modèle conceptuel précédent. Il introduit également deux valeurs supplémentaires: la **couverture** et le **dommage**.

4.2.1 Le fournisseur cloud: implémentation de contrôles sur les offres

Définition 1 (Fournisseur Cloud) *Une entité qui peut proposer plusieurs offres de services cloud. Elle est responsable d'implémenter des contrôles de sécurité pour protéger ces offres d'attaques, d'éviter des incidents et de se conforme à des régulations (des exemples sont les contrôles donnés par le CSA [CSA14] ou dans l'ISO 27017 [ISO15]). Formellement:*

Offre, *un ensemble d'offres disponibles chez le fournisseur.*

Contrôle, *un ensemble de contrôles de sécurité donné dans des standards pour éviter des incidents de sécurité.*

Implémentation *: Offre × Contrôle → Stage, définit de quelle façon un contrôle est implémenté sur une offre. Le Stage peut donner une information plus précise qu'un simple oui/non.*

Au lieu de se focaliser sur les vulnérabilités comme le font Sackmann *et al.* dans [SLK09], nous adoptons la perspective des **contrôles** de sécurité. En général, un fournisseur cloud est plus intéressé à dissimuler les vulnérabilités de ses offres que de les publier. Effectivement, publier des failles de sécurité exposerait le fournisseur de façon superflue à de possibles attaques. Or, les contrôles de

sécurité sont plus facilement publiable, un exemple est le Security, Trust and Assurance Registry [Clo14] où les fournisseurs cloud peuvent rendre public les contrôles qu'ils implémentent.

Dans notre approche nous considérons que cette information peut être plus précise qu'une simple valeur binaire. Un contrôle peut par exemple être implémenté de façon "partielle" on encore "prévu" d'être implémenté.

4.2.2 Le consommateur cloud: définition des objectifs sur les actifs

Définition 2 (Consommateur Cloud) *Une entité qui utilise un ou plusieurs services cloud adaptés à ses besoins (fontionnels ou non). Formellement:*

Actif, *l'ensemble des actifs (biens essentiels) du consommateur.*

Critère, *un ensemble de critères de sécurité, typiquement {Confidentialité, Intégrité, Disponibilité} (parfois Authenticité et Non-répudiation peuvent y être ajoutés).*

Objectif *: Actif × Critère → Exigence, définit une exigence de sécurité. Donne une description de la "quantité" de sécurité dont l'actif à besoin pour le critère en question.*

Seuil *: Ω → Niveau, défini un niveau global de risque acceptable. Ω étant le système complet, signifiant que ce seuil est défini pour le système entier.*

Un **objectif** de sécurité aide à déterminer l'impact d'un risque de sécurité. Il est généralement défini avec les critères de sécurité CIANA [NIS02]. Cependant, notre modèle n'est pas limité à ces critères, d'autres références peuvent être utilisées. Pour chaque actif on définit une **Exigence**, qui équivaut à un niveau pour le critère en question. Généralement on utilise des termes comme *Faible*, *Moyen* et *Élevé* définit sur une échelle graduée. Pour notre approche nous proposons un intervalle de [0,1], 1 étant la valeur maximale et 0 signifiant que l'actif n'a aucune exigence sur ce critère.

Le **seuil** est utilisé pour la sélection finale. Il indique le niveau de risque acceptable est sert à exclure les services cloud où le niveau est trop élevé.

Dans le cadre de déploiement de processus métiers, nous annotons les processus avec de tels objectifs pour ensuite décider quels fragments seront déployés sur quels offres. Un élément essentiel de la gestion de processus métier est que les modèles de processus sont généralement orienté **tâches**. Donc les activités du processus et leur ordre d'exécution plutôt que ce qu'ils produisent (les données). Il existe d'autres représentations ([Hul08, KLW08]), mais dans le contexte du cloud où les offres sont décrites sous forme de services, cette notation est bien plus adaptée. Pourtant, les exigences en terme de sécurité sont souvent décrites sur les données ([GBO+07, ZSM+10]). Effectivement, on parle de sécurité de l'information qui est donc orienté **donnée**. Pour palier à ce problème nous proposons une approche similaire à celle présenté par Watson [Wat12] qui est inspirée d'un modèle de contrôle d'accès type Bell-LaPadula. Nous annotons les données d'un processus avec leurs exigences en terme de sécurité et les traduisons en exigences sur les tâches avec la formule suivante:

Définition 3 (Exigences de sécurité orienté tâches) *Pour une tâche t_i, les données entrantes et sortantes d de t_i and le critère de sécurité o_i, l'exigence de sécurité orienté tâche est donnée par:*

$$Objectif(t_i, o_i) = \max_{d_j \in données(t_i)} \Big(Objectif(d_j, o_i) \Big) \tag{4.1}$$

4.2.3 Le broker: menaces, atténuations et conséquences

Définition 4 (Broker Cloud) *Une entité qui peut fournir trois types de services [LTM+11]: améliorer un service existant, combiner plusieurs services ou comparer différents services entre eux. Formellement:*

Menace, *l'ensemble des menaces de sécurité cloud. Cet ensemble devrait être générique pour tous les cas d'utilisation, étant donné que tout service cloud y est exposé.*

Contrôle, *le même ensemble de contrôle que celui considéré par le fournisseur cloud.*

Critère, *le même ensemble de contrôle que celui considéré par le consommateur cloud.*

Conséquence: *Menace* × *Critère* → *Sévérité, indique comment une menace affecte un critère.*

Atténuation : *Menace* × *Contrôle* → *Degré, indique comment un contrôle atténue une menace.*

Il existe différentes menaces dans le contexte cloud qui peuvent à tout moment produire un incident de sécurité [CSA13]. Pour évaluer le possible importance (*i.e.* **impact**) que leur réalisation aurait, nous relions ces menaces aux **critères** de sécurité par les **conséquences**. Chaque menace peut avoir une conséquences plus ou moins importante sur ces critères. Par exemple un *Déni de service* affecte surtout la *disponibilité* et non la *confidentialité*. Dans notre approche, cette information peut être quantifiée par un intervalle de [0,1], la **Sévérité**.

Pour évaluer les vulnérabilités d'une offre cloud à une menace donnée, nous utilisons les **atténuations**, qui relient les contrôles aux menaces. Chaque contrôle de sécurité à comme objectif de réduire la probabilité qu'un incident de sécurité se produise. Dans le cadre des contrôles donnés par le CSA, ces contrôles peuvent directement être reliés aux menaces (voir [CSA13]). De plus, certains contrôles peuvent être plus efficaces que d'autres, c'est pourquoi ici aussi nous proposons un intervalle de [0,1] pour indiquer le **Degré** d'atténuation.

4.2.4 Couverture: implémentation des contrôles et atténuation des menaces

Définition 5 (Couverture) *Un score calculé pour une offre cloud et une menace donnés. Il est calculé à l'aide des contrôles de sécurité que le fournisseur implémente et leur atténuation des menaces. Formellement:*

$$\begin{aligned} \textbf{Couverture} : & Offre \times Menace \rightarrow Score \\ & o,t \mapsto Couverture(o,t) \\ & = \min\left(1, \sum_{c \in Contrôle} \Big(Implémentation(o,c) \times Atténuation(t,c)\Big)\right) \end{aligned} \tag{4.2}$$

Généralement, une offre qui implémente beaucoup de contrôles de sécurité est plus sécurisée qu'une autre offre qui en implémente moins. Cependant cela peut être influencé par l'efficacité des contrôles implémentés et par la façon dont ils sont implémenté. Ainsi on peut obtenir un **Score** qui indique comment une offre cloud répond à une menace donnée. Tel que défini, notre score est une valeur sur l'intervalle [0,1], 1 signifiant que l'offre cloud n'est pas du tout exposé à la menace en question (elle est totalement couverte).

4.2.5 Dommage: exigences de sécurité et conséquences des menaces

Définition 6 (Dommage) *Un taux calculé pour un actif et une menace donnés. il est obtenu en combinant les exigences de sécurité de l'actif et les conséquences de la menace concernée. Formellement:*

$$\begin{aligned} \textbf{Dommage} : & Actif \times Menace \rightarrow Taux \\ & a, t \mapsto Dommage(t,a) \\ & = 1 - \prod_{c \in Critère} \Big(1 - \big(Conséquence(t,c) \times Objectif(a,c)\big)\Big) \end{aligned} \tag{4.3}$$

Le **Taux** de **Dommage** représente l'impact que peut avoir une menace sur un actif donné. Il permet de différencier les actifs du consommateur cloud, vu que tous n'ont pas les mêmes exigences en terme de sécurité. Tel que défini, ce taux est une valeur sur l'intervalle [0,1], 1 signifiant que le dommage serait maximal. En principe, certaines menaces seront beaucoup plus importantes pour certains actifs que d'autres. Un exemple, certains actifs n'ont pas besoin d'un bon niveau de *disponibilité*, ils ne seront donc pas grandement affecté par un *déni de service*.

4.2.6 Risque: probabilité de menace

Définition 7 (Niveau de risque) *Un niveau calculé pour une menace, un actif et une offre donnés. C'est le produit du dommage de la menace sur l'actif et de la vulnérabilité de l'offre à cette même menace. Cette valeur peut être pondérée par la probabilité de la menace. La vulnérabilité est obtenu en prenant le complémentaire de la couverture. Formellement:*

$$\begin{aligned} \textbf{Risque} : & Menace \times Actif \times Offre \rightarrow Niveau \\ & t, a, o \mapsto Risque(t,a,o) \\ & = \frac{k_h \times Dommage(t,a) + k_c \times \Big(1 - Couverture(o,t)\Big)}{k_h + k_c} \times Probabilité(t) \\ & \text{avec } k_h, k_c \in \mathbb{N} \end{aligned} \tag{4.4}$$

La probabilité de menace est un pourcentage (intervalle [0,1]). Elle permet de se concentrer sur certaines menaces plutôt que d'autres. En effet, il est généralement accepté que certaines menaces sont plus probables que d'autres. Le CSA donne ce type de pourcentage dans son rapport [CSA13].

Notre calcul du niveau de risque est similaire à celui donnée dans l'état de l'art. Par contre nous utilisons une somme pour mieux montrer l'indépendance entre les vulnérabilités et l'impact: le fournisseur ne peut pas influencer la valeur d'impact, le consommateur ne peut pas influencer la valeur de la vulnérabilité. La pondération peut être intéressante dans certains cas pour mettre plus en évidence l'un des deux aspects.

Chapitre 5

Déploiement d'un processus métier sur plusieurs clouds

Ce chapitre est principalement basé sur les contributions publiées dans [GFG13]. Il détaille comment un processus métier peut être découpé en plusieurs fragments et comment ces fragments peuvent être déployés dans un environnement multi-cloud. Cette partie s'intègre dans des travaux plus génériques publiés dans [FDGG14]. De plus, nous présentons un algorithme d'optimisation multi-critères pour prendre en compte d'autres paramètres que la sécurité.

5.1 Aperçu

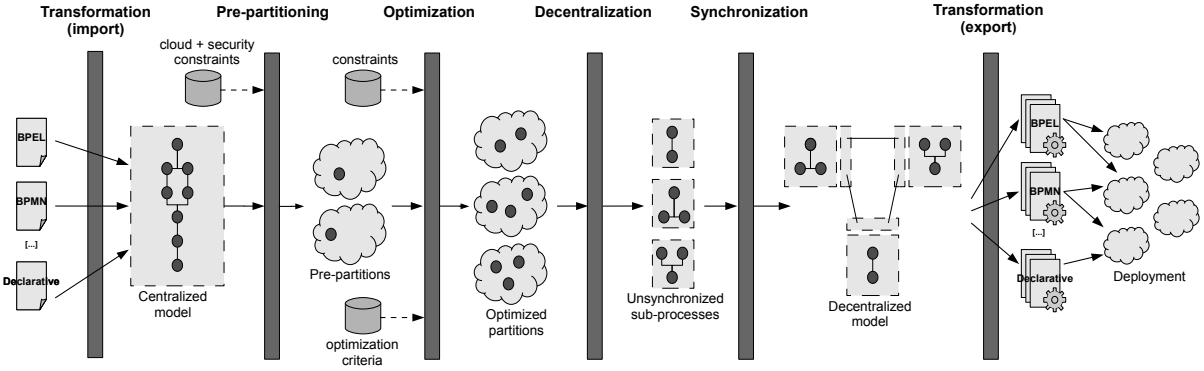

FIGURE 5.1 – Aperçu de l'approche de décomposition

La base de notre approche est un algorithme pour découper un processus métier en partitions (voir FIGURE 5.1). Cette algorithme pour effectuer la décomposition est expliqué en détail dans [FYG09]. Ce partitionnement transforme un processus centralisé en sous-processus qui peuvent être distribués sur des services distants tout en préservant le comportement initial du processus centralisé. Ces partitions interagissent entre elles à travers un mécanisme de messages asynchrones.

L'approche est divisée en cinq parties: la **transformation** (importation et exportation des modèles), le **pré-partitionnement**, l'**optimisation**, la **décentralisation** et enfin la **synchronisation**.

5.2 Les critères à évaluer

Avant d'entrer dans les détails de l'approche de décomposition et de déploiement, nous présentons d'autres critères à prendre en compte lors d'une externalisation vers des services cloud. En effet, il nous semble indispensable d'intégrer d'autres paramètres que la sécurité pour sélectionner la configuration de déploiement la plus appropriée.

5.2.1 Le coût

Tout d'abord le coût, qui est en général le facteur le plus important pour motiver et analyser l'intérêt d'une externalisation vers des services cloud. Nous proposons un modèle de coût qui intègre trois informations distinctes et qui reflète de façon assez fidèle les modèles de coût utilisés par les offres cloud existantes.

- **Coût d'utilisation (C_{usg})** qui correspond au prix de la performance offerte par le service cloud (généralement en Dollars par GigaHertz par heure). Cette information doit être mise en relation avec le *coût d'exécution (C_{exc})* qui correspond aux besoins en termes de performance qui peut être annoté sur chacune des tâches du processus.
- **Coût de stockage (C_{str})** qui correspond au prix de stockage offert par le service cloud (généralement en Dollars par GygaBytes par heure). Cette information doit être mise en relation avec le *coût d'espace (C_{sze})* qui correspond aux besoins en termes d'espace de stockage qui peut être annoté sur les données du processus. Additionnellement on indique une *période de rétention (RP)* pour indiquer combien de temps cette donnée doit être stockée.
- **Coût de transfert (C_{trs})** qui correspond au prix pour transmettre des messages à des services externes (généralement en Dollars par Gygabytes). Généralement, les fournisseurs facturent les données qui doivent transiter sur le réseau. Ici les sous-processus distants doivent communiquer entre eux, donc cette information est importante et dépend du *coût d'espace (C_{sze})* des données échangées.

De plus, certaines tâches ne sont pas exécutées tout le temps (à cause des branches *OR*). C'est pourquoi nous annotons les tâches du processus avec une probabilité d'exécution (P_{exc}). Cette probabilité peut-être supérieure à 1 dans le cas de boucles.

Définition 8 (Coûts) *Pour une configuration de déploiement donnée (i.e. l'assignation de tâches, leurs données et les messages à différentes offres cloud), le coût pour exécuter une instance de ce processus est défini de la manière suivante:*

$$\begin{aligned} Coûts(instance) = & \sum_{t \in tâches} P_{exc}(t) \times C_{exc}(t) \times C_{usg}(o_t) \\ & + \sum_{d \in données} P_{exc}(t_d) \times RP(d) \times C_{sze}(d) \times C_{str}(o_d) \\ & + \sum_{m \in msgs} P_{exc}(t_m) \times C_{sze}(m) \times \left(C_{trs}(o_{m/in}) + C_{trs}(o_{m/out})\right) \end{aligned} \tag{5.1}$$

avec o_t l'offre où t est déployé, o_d l'offre où d est stocké
$o_{m/in}$ l'offre où m sort, $o_{m/out}$ l'offre dont m sort
t_d la tâche qui créée d, t_m la tâche qui envoie m

5.2.2 La qualité de service

La qualité de service peut inclure beaucoup de paramètres différents. Il existe différentes offres commerciales qui proposent de comparer les offres cloud quant à leur qualité de service. Quelques exemples sont: CloudScreener [3], SoftwareInsider [4], HostAdvice [5] ou encore TopTenReviews [6].

En ce qui concerne la communauté scientifique, il existe différentes contributions qui proposent des solutions pour évaluer la qualité de services cloud. Les auteurs de [AJG+15] ont fait une étude systématique pour identifier les différents types d'approches existantes. Les auteurs de [BH14] donnent une liste de métriques pour mesurer différentes caractéristiques qui peuvent être considéré comme des attributs de qualité de service. Baliyan *et. al* [BK13] proposent une approche basée sur la logique floue pour donner des niveaux similaire à notre approche risque. Becker *et. al* [BLB15] définissent six unités de mesures différentes pour évaluer les critères comme l'élasticité ou la mise à l'échelle. D'autres approches comme celle présentée dans [BYOG13] proposent d'évaluer la qualité de service selon deux critères: le coût et le temps d'exécution.

Même s'il n'existe pas de compromis commun pour une définition de la qualité de service, nous pouvons tout de même conclure qu'en général il s'agit d'une valeur numérique associée à une échelle (donc uns sorte de score ou une note).

5.2.3 La complexité

Comme expliqué précédemment, notre approche repose sur le déploiement de processus sur plusieurs services cloud. Cependant, une telle configuration peut créer de gros problème d'interopérabilité, puisque tous les services n'utilisent pas forcément les mêmes technologies, standards, langages, protocoles, *etc.*. Des adaptations ou des configurations spécifiques peuvent être nécessaires dans certaines situations, et donc entraîner des coûts supplémentaires. Aussi, certaines autres tâches, comme la *négociation de contrat* ou la *souscription au service* doivent du coup être réalisés plusieurs fois. Cela rend l'externalisation plus complexe qu'en utilisant qu'un seul service.

Cas différents points nous mènent à dire que certaines configurations sont plus "complexes" que d'autres, ce qui doit être pris en compte lors de la sélection de la configuration finale. Vu que ces paramètres peuvent être très compliqué à mesurer ou quantifier, nous proposons une métrique très simple basée sur le nombre de services impliqués dans la configuration (Nb_{srv}) et le nombre de tâches du processus à déployer (Nb_{tsk}).

Définition 9 (Complexité) *La complexité d'une configuration de déploiement d'un processus métier est donnée par la formule suivante:*

$$Complexité(conf) = \frac{Nb_{srv}(conf)}{Nb_{tsk}(conf)} \tag{5.2}$$

Le dénominateur rend la valeur plus générique: il est plu facile d'accepter beaucoup de services impliqués pour des processus avec beaucoup de tâches.

Un point intéressant est que cette valeur peut également être perçue différemment. En effet, dans certains cas une complexité "élevé" peut être synonyme d'une bonne protection du savoir-faire: il devient plus difficile pour un attaquant de recomposer le processus global que s'il était déployé sur un seul service.

3. `http://www.cloudscreener.com/en/`
4. `http://cloud-computing.softwareinsider.com/`
5. `http://hostadvice.com/`
6. `http://cloud-services-review.toptenreviews.com/`

5.2.4 D'autres exigences fonctionnelles (et non-fonctionnelles)

En général il est assez complexe de décrire des spécifications fonctionnelles de façon formelle. Il existe certaines approches comme par exemple le eSourcing Capability Model for Client Organization (eSCM-CL, [KH07]) pour aligner les exigences d'un client avec les capacités d'un fournisseur. Spécifiquement aux processus métiers il existe le méta-modèle présenté dans [FJ12] implémenté sous forme d'extension BPMN et conçu pour une sélection dynamique de services.

Vu qu'il ne s'agit pas du coœur de la problématique de notre thèse, nous limitons notre proposition à la prise en compte de ce type d'exigences avec une simple fonction booléenne:

Définition 10 (Exigences fonctionnelles)

$$\textbf{Deployable} : Tâche \times Offre \rightarrow Booléen$$

$$t, o \mapsto Déployable(t,o) = \begin{cases} Vrai, \text{ si } o \text{ peut exécuter } t \\ Faux, \text{ sinon} \end{cases} \tag{5.3}$$

De la même façon, il est possible de prendre en compte d'autres exigences non-fonctionnelles. Des exemple seraient: la géolocalisation du service (pour des raisons légales), le temps de réponse garanti, la documentation, la protection environnementale, *etc.*.

De façon plus générale, notre approche distingue deux grandes classes de critères: ceux qui sont équivalent à des contraintes et qui doivent être respectées et ceux qui son considérés comme des critères à optimiser. Le premier type amène à exclure certaines configurations, puisque une configuration ne pas respectant ces contraintes ne pourra pas être sélectionnée. Le second type aide à définir quand une configuration est meilleure qu'une autre.

5.3 Approche de décomposition et de déploiement

Notre approche de décomposition et de déploiement ce fait en cinq étapes

5.3.1 Transformation

La transformation est découpé en deux parties distinctes. La première est responsable d'importer les modèles de processus (BPMN et JSON) dans un format interne (une structure équivalente à un graphe orienté et typé). La seconde est responsable d'exporter cette structure, une fois décomposée, dans les formats de sortie (BPMN, BPEL ou dotGraph). Certaines notations nécessitent des adaptations structurelles puisque certains patrons ne sont pas supportés (comme le *multi-send* et le *multi-receive* en BPEL).

Ce type d'architecture permet assez facilement d'ajouter le support pour d'autres notations (comme par exemple YAWL ou EPC).

5.3.2 Pré-partitionnement

La phase de pré-partitionnement distribue les tâches dans des *partitions* pour garantir le respect des contraintes qui ont été imposé au préalable (typiquement les contraintes de co-localisation ou de séparation). L'objectif est de générer un premier ensemble de partitions respectant toutes ces contraintes, sachant que ces contraintes ne sont pas *orthogonale* (*i.e.* certaines peuvent être contradictoire). Il est donc nécessaire dès cette étape de vérifier qu'il est possible de trouver au moins une solution de déploiement.

5.3.3 Sélection optimisée

Cette phase d'optimisation a comme objectif de maximiser ou minimiser les critères présentés précédemment. L'intérêt et de trouver la solution optimale.

Cependant, ce problème est un problème d'optimisation multi-critères (ou multi-objectifs), qui n'est pas simple à résoudre. Effectivement, il n'existe pas de méthode universelle, notamment dû au fait qu'il est difficile de définir *la solution optimale* dans ce cas de figure. Dans le cas de plusieurs critères, on utilise souvent les solutions de *Pareto*, mais celles-ci sont rarement limitées à une solution unique.

De plus, ce problème est un problème *NP-complet*, où N tâches peuvent être assignées à P services cloud. Plus précisément il s'agit d'un Problème d'Assignement Quadratique (QAP, [BePP98]).

5.3.3.1 Considérer plusieurs critères

Selon [HM79] il existe quatre catégories principales de méthodes d'optimisation multi-critères:

- les stratégies *sans-préférence*, où un compromis neutre est déterminé sans aucune interaction d'un *décideur*.
- les stratégies *a priori*, où un *décideur* définit ses préférences avant la recherche de la solution.
- les stratégies *a posteriori*, où un *décideur* sélectionne sa solution préférée parmi un ensemble de solutions de Pareto.
- les stratégies *interactives*, où un *décideur* recherche la solution préférée de façon interactive.

Les méthodes les plus communes implémentant ces stratégies sont les suivantes: **l'agrégation** (on agrège les critères en un seul critère à optimiser), **ϵ-contrainte** (on choisit un seul critère à optimiser, on transforme les autres en contraintes) ou **l'ensemble de Pareto** (on élimine toutes les solutions qui sont dominées par une autre). Pour notre approche nous définissons la méthode **hybride** suivante pour combiner les avantages de chacune des méthodes précédentes:

Définition 11 (Approche hybride)

$$\forall x \in S, \nexists y \in S \text{ tel que } y \succ x, \text{ avec}$$
$$x \succ y \Leftrightarrow \begin{cases} \sum_{i=0}^{k} w_i[c_i(x) >_i c_i(y)] > T, & \text{avec } T \text{ le seuil global} \\ \forall i \in \{1,...,k\}, c_i(x) >_i c_i(y) +_i t_i, & \text{avec } t_i \text{ le seuil de } c_i \end{cases} \quad (5.4)$$

Nota Bene: le signe $>_i$ ne signifie pas "supérieur à" mais "meilleur que". En effet, certaines valeurs de critères (comme le coût ou le risque) sont "meilleurs" lorsqu'il sont plus faibles alors que pour d'autres (comme la qualité de service) c'est l'inverse.

De manière moins formelle, notre approche consiste à garder uniquement un ensemble de solutions qui ne se domine pas entre elles. La condition de la domination d'une configuration x sur une configuration y est définie ainsi:

- la somme des poids des critères où x est meilleur que y doit être supérieur à un seuil global (généralement fixé à 50% du total des valeur de pondération)
- pour tous les critères, la valeur de y ne peut pas être meilleure que celle de x de plus de la valeur du seuil de ce critère (ce seuil représente la différence pour laquelle la première condition est reconsidérée)

5.3.3.2 Heuristiques

Le problème de *NP-complétude* avait déjà été étudié dans de précédents travaux [FDGG14], cependant avec une approche d'agrégation. La proposition consistait en une combinaison d'un algorithme construisant une solution initiale dite de "*Greedy*" [MF02] qui était par la suite améliorée avec une recherche "*Tabu*" [GL97]. Notre approche est une adaptation de cette méthode, cependant elle génère un ensemble de solutions qui sont considérées comme étant de "bons" candidats.

A cette étape, les tâches du processus sont assignées à des partitions, qui sont elles-mêmes assignées à des offres de services cloud. Ni les tâches, ni les partitions ne sont reliées entre elles. Il est à noter que des partitions différentes peuvent être assignées à un même service cloud.

5.3.4 Décentralisation et synchronisation

Une fois les partitions générées et assignées, il faut générer les sous-processus correspondants. Pour cela l'approche utilise des *Table de Dépendances de Contrôles Transitives* (TDCT). La construction des sous-processus est expliqué en détail dans [FYG09]. Par la suite il faut synchroniser les sous-processus entre eux pour garantir le comportement initial du processus centralisé. Cela s'effectue en deux temps:

- **Synchronisation du flot de contrôle** - Pour garantir l'équivalence entre le processus initial et la décomposition générée, il faut s'assurer que les tâches s'exécutent dans le même ordre. On insère donc dans les sous-processus des tâches d'*envoi* et de *réception* de messages. Les tâches de *réception* bloquent l'exécution du sous-processus qui doivent attendre que le sous-processus distant correspondant le notifie au travers l'*envoi* d'un message qu'il peut continuer son exécution. Encore une fois, voir [FYG09] pour plus de détails.
- **Synchronisation du flot de données** - Pour s'assurer que les données requises par chacun des sous-processus soit disponibles lorsqu'elles sont demandées, il faut envoyer les données une fois qu'elle ont été produites aux sous-processus qui en ont besoin. Des détails sur ce type de synchronisation sont donnés dans [GFG13].

5.3.5 Déploiement

Une fois les sous-processus générés et synchronisés, ceux-ci peuvent être transformé en processus exécutables (par exemple au format BPEL) pour qu'ils puissent être déployés sur des services cloud. Les points d'interaction (pour les messages de synchronisation) sont décrit dans des fichiers WSDL pour qu'ils deviennent accessibles sous forme de web-services.

Dans le cas de plate-formes ayant des APIs (comme pour le moteur d'exécution Apache ODE [7]), les processus peuvent même être déployés à distance et de façon automatique.

Un point intéressant avec ces processus décomposés est que chaque cloud ne "voit" que les interfaces des sous-processus distants, et n'a donc pas de vue complète du processus global. Un gain majeur en terme de *préservation de savoir-faire*.

7. Orchestration Director Engine, `http://ode.apache.org`

Chapitre 6

Implémentation

Ce chapitre a comme objectif de démontrer la faisabilité et l'utilité de notre approche globale. Elle est divisée en trois partie: les outils développés, application dans le domaine du contrôle d'accès et cas d'étude réel.

6.1 Les outils développés

L'approche présenté dans cette thèse est implémenté dans 3 outils différents.

6.1.1 Évaluation des risques

Notre outil d'évaluation de risques de sécurité cloud est accessible sous la forme d'une interface web en tant que prototype [8]. L'outil est basé sur trois librairies principale qui sont Blockly [9], KnockoutJS [10] et JQuery [11]. L'évaluation des risques se fait en trois étapes.

6.1.1.1 Construction du modèle

L'objectif de l'outil est d'offrir à l'utilisateur une flexibilité maximale en instanciant chacun des concept de notre modèle de la façon qui lui convient le mieux. Il est donc possible de travailler avec n'importe quel référentiel de **critères** de sécurité (CIA, CIANA, STRIDE, *etc.*) et il est même possible de définir ses propres critères.

Des niveaux de sécurité peuvent être associés à chacun de ces critères. Chaque critère peut avoir un nombre de niveaux différents (on peut très bien évaluer la confidentialité sur 3 niveaux, et la disponibilité sur 10). Comme expliqué pour le modèle d'évaluation des risques, nous attribuons des valeurs à ces niveaux sur un intervalle de [0,1]. Au travers de fonctions d'ajustement (*normalisation*, *minimisation*, *maximisation*, *centrage*, *etc.*), nous pouvons aligner ces niveaux entre eux.

De la même façon il est possible de sélectionner le type de **menaces** sur lesquelles l'analyse va être faite ou encore les **contrôles** de sécurité sur lesquels les fournisseurs seront évalués. Par cette construction "par blocs" du modèle, la méthode d'évaluation devient très flexibles par rapport aux besoins spécifiques du cas d'utilisation.

Les fonctions d'agrégation pour calculer le **dommage** et la **couverture** (comme celle présentée dans CHAPITRE 4), peuvent être facilement changé. Dans certains cas il peut être plus utile de passer

8. L'outil est disponible en ligne à: `http://elio.goettelmann.fr/projects/cloudra_v2/`
9. `https://developers.google.com/blockly/`
10. `http://knockoutjs.com/`
11. `https://jquery.com/`

par les fonctions *union*, *intersection*, *maximum*, *minimum* ou *moyenne* pour combiner les **conséquences** avec les **objectifs** et les **implémentations** avec les **mitigations**.

6.1.1.2 Définition des exigences de sécurité

Une fois le modèle construit et donc que la façon dont le risque est calculé a été déterminé, le processus d'évaluation peut commencer. Une interface de saisie est générée automatiquement lors de la construction du modèle. Ainsi, pour chaque **actif** renseigné, il devient possible de sélectionner l'**exigence** de sécurité (le niveau) correspondant à chacun des **critères**.

De la même manière, l'utilisateur de l'outil peut configurer les relations (et les quantifier) entre les **menaces** et les **critères** de sécurité, les **contrôles** qui sont **implémentés** sur les différentes **offres**, les **menaces** qui sont **atténuées** par ces **contrôles**, *etc.*. L'interface est toujours générée à-la-volée, en fonction des blocs sélectionnés et des informations entrées.

6.1.1.3 Évaluation des risques pour chaque fournisseur

Dans notre outil, les valeurs finales de risque de sécurité sont affichées dans une vue séparée. Un exemple est montré dans FIGURE 6.1. Pour chaque triplet *{menace, actif, offre}*, une valeur de risque est indiquée. Cette valeur est également colorée en fonction de son intensité pour identifier directement les risques les plus importants.

Risks

Asset0

	Offer0	Offer1	Offer2
Data breaches	0.25	0.25	0.75
Data loss	0.50	0.00	0.50
Account or Service traffic Hijacking	0.50	0.00	0.00
Insecure Interfaces and APIs	1.00	0.50	0.50
Denial of Service	0.75	0.75	0.25
Malicious Insiders	0.50	0.50	0.00
Abuse of Cloud Services	0.50	0.00	0.00
Insufficient Due Diligence	0.00	0.00	0.00
Shared Technology Vulnerabilities	1.00	1.00	1.00

FIGURE 6.1 – Les valeurs de risque affichées dans l'outil web

Il est également possible d'ajouter des filtres pour extraire uniquement les offres acceptables, les menaces les plus importantes ou les actifs critiques par exemple. Une vue complète de notre outil est présenté dans FIGURE 6.2. Les blocs sont sélectionnés depuis le menu sur la gauche et peuvent être ajouté à l'espace de travail. Les onglets supérieurs permettent de naviguer au travers de chacune des vues pour indiquer les différentes informations nécessaires à l'analyse.

6.1.2 Sélection optimisée

La seconde partie de notre implémentation est l'algorithme d'optimisation qui effectue une sélection multi-critères parmi un ensemble d'offres de services cloud. En rappel, voici les points clés de

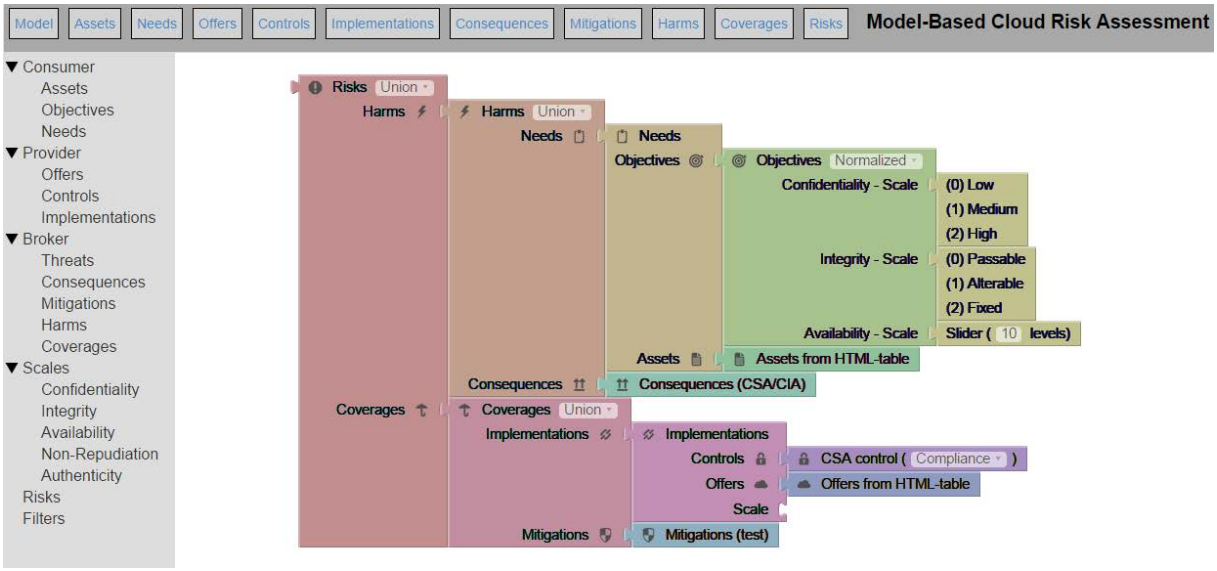

FIGURE 6.2 – Outil d'évaluation des risques de sécurité cloud basé sur les modèles

notre approche multi-critères:

- pour chaque critère nous définissons une pondération qui représente l'importance de ce critère.
- un seuil global définit une valeur au delà de laquelle la "somme des pondérations" détermine la domination d'une des solutions.
- un seuil est défini pour chaque critère, qui correspond à une différence pour laquelle la première assertion est reconsidérée.
- le résultat est un ensemble de solutions qui ne se dominent pas entre elles.

Algorithme 1: Comparaison multi-critères

Input : S; // Configurations possibles
Output : Res; // Bonnes configurations
1 $Res_i \leftarrow \{\}$; // Ensemble intermédiaire
2 **foreach** $s_i \in S$ **do**
3 **foreach** $s_j \in S \setminus \{s_i\}$ **do**
4 **if** $s_j >_P s_i$ **then** // s_j domine au sens de Pareto s_i
5 $S \leftarrow S \setminus \{s_i\}$
6 **else if** $s_i \in Res$ **and** $s_i >_H s_j$ **then** // s_i domine au sens hybride s_j
7 $S \leftarrow S \setminus \{s_j\}$;
8 $Res_i \leftarrow Res_i \cup \{s_j\}$;
9 **else if** $s_j >_H s_i$ **and** $s_j \in (Res \cup Res_i)$ **then** // s_j domine au sens hybride s_i
10 $Res_i \leftarrow Res_i \cup \{s_i\}$;
11 **else**
12 $Res \leftarrow Res \cup \{s_i\}$;
13 **end**
14 **end**
15 **end**

Nous avons effectué différentes expériences avec l'algorithme qui ne seront cependant pas présentés ici. Les conclusions que nous tirons de ces tests sont les suivantes:

- l'algorithme permet d'intégrer efficacement la valeur de risque avec les autres critères, en se passant de formules d'agrégation complexes et en effectuant une optimisation réelle sur chacun des critères.
- l'algorithme permet de réduire le nombre de configurations retenues de façon relativement raisonnable (autours de 50%), ce qui est plutôt correct pour des tests effectués sur un ensemble généré aléatoirement.
- des expériences supplémentaires sont nécessaires pour vérifier le comportement de l'algorithme avec plus de quatre critère et surtout sur un ensemble réel de possibles configurations.

6.1.3 Déploiement de processus

Comme déjà expliqué auparavant, notre approche s'intègre dans une approche plus globale pour décomposer des processus en fragments. Nous ne présentons pas l'outil global puisqu'il n'est pas une contributions directe de cette thèse. Cependant, un aspect spécifique à notre contexte cloud est développé dans cette section.

Lors d'une montée en échelle, avec beaucoup de fragments assignés à beaucoup de clouds, il peut devenir rapidement complexe de déployer et gérer ces fragments distribués. C'est pourquoi nous avons développé dans le cadre de nos travaux un outil de déploiement à distance de ce types de fragments. L'outil a été développé en JAVA.

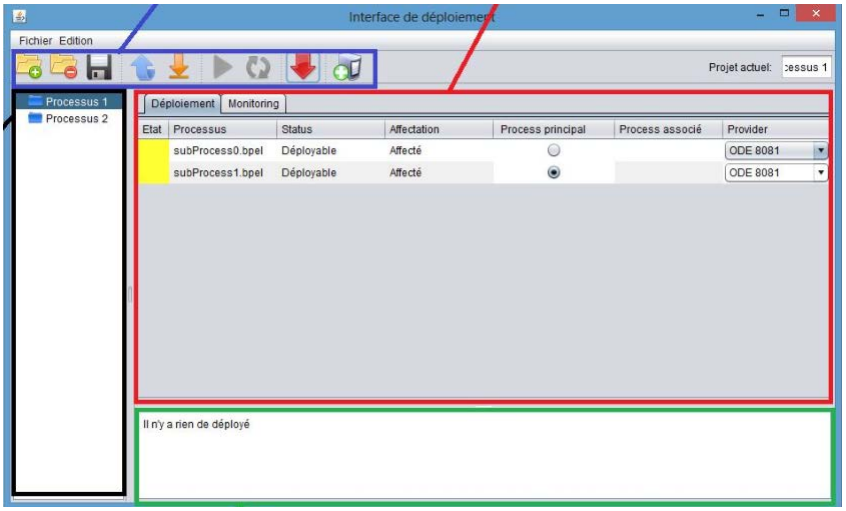

FIGURE 6.3 – Interface de l'outil de déploiement et de monitoring

L'interface est présentée dans FIGURE 6.3, elle permet d'assigner manuellement chaque fragment à une offre cloud disponible, puis de déployer l'ensemble. Un onglet de monitoring permet également de récupérer l'état des processus distants, les instances en cours exécution et d'autres informations intéressantes. Cet outil fonctionne avec des plate-formes d'exécution de processus accessibles en services web.

6.1.3.1 Expérience de déploiement

De plus, nous avons effectué une expérience pour démontrer le fonctionnement de l'outil et une possible exécution de processus distribués sur de multiples environnements. Nos expérimentations se

sont déroulées avec deux offres cloud distantes et un cloud local ayant son propre moteur d'exécution de processus.

La conclusion des expériences était un bon déroulement de toute la partie déploiement à distance. Cependant, quelques problèmes ont été rencontré lors de l'exécution d'instances de ces processus. Effectivement, les plate-formes distantes ne garantissaient pas la distribution des messages de synchronisation. Certains messages étaient supprimés et l'ordre d'envoi/réception n'était pas systématiquement respecté. Certaines situations bloquantes apparaissaient où des tâches attendaient des messages qui avait déjà été supprimés.

Ce type de problème est généralement réglé à l'aide de la mise en place d'une pile de messages associée à chaque processus (ce qui n'était pas le cas sur ces offres distantes). Cependant, nous estimons qu'il s'agit ici d'une limitation technique des offres disponibles et ne remet en aucun cas en question notre approche.

6.2 Application dans le domaine du contrôle d'accès

Pour explorer les limites de notre approche et faire de première avancées dans la considération automatique des risques de sécurité, nous avons implémenté notre modèle d'évaluation des risques dans le domaine du contrôle d'accès [BGPG15].

En parallèle de nos travaux, le LORIA [12] réalisait un projet industriel avec une entreprise sur les Réseaux Sociaux Professionnels dans le cadre du cloud. L'un des objectif de ce projet était d'étudier les mécanismes de contrôle d'accès avec des politiques de sécurité définies de façon distribuées. Nous avons saisi cette opportunité pour proposer des amélioration avec la considération de risques de sécurité dans ce types de contextes.

Nous avons dons formalisé nos différents concepts pour le contexte du contrôle d'accès. L'objectif était évaluer la possibilité que malgré l'existence d'un mécanisme d'autorisation, *un utilisateur arrive à accéder à une ressource alors qu'il ne devrait pas*. Cela pouvant se produire pour diverses raisons, par exemple: une politique défaillante, une usurpation d'identité, des failles dans le mécanisme d'authentification, *etc.*.

Nous avons donc défini le modèle de la façon suivante:

- **Impact** - Nous proposons de définir l'impact par la gravité qu'aurait l'action demandée par l'utilisateur sur la ressource si elle n'était non-autorisée. Nous proposons d'évaluer cela par le nombre d'utilisateurs ayant ce droit, divisé par le nombre d'utilisateurs total du système. De façon plus simple, cela correspond à dire que si beaucoup d'utilisateurs peuvent effectuer une certaine action, ce n'est pas bien grave si une personne non autorisé peut le faire également. Au contraire, une action autorisée pour très peu de personnes est un bon indicateur de gravité d'un tel événement.
- **Vulnérabilité** - Les vulnérabilités menant à un tel événement est fortement lié au mécanisme d'identification de l'utilisateur. Un mécanisme qui peut difficilement être déjoué (comme la reconnaissance par biométrie) présente un niveau de vulnérabilité plus faible qu'un simple mot de passe. Bien entendu, l'approche n'est pas limité à ce type de considération, il existe d'autre méthodes qui permettent de quantifier les vulnérabilités d'un système et qui peuvent être utilisé à ce niveau.
- **Menace** - En principe, la menace correspond à la probabilité que l'événement se produise. Ici, nous considérons que cette information est directement relié à l'utilisateur. C'est pour-

12. Laboratoire lorrain de recherche en informatique et ses applications: `http://www.loria.fr`

quoi nous proposons un concept qui est largement utilisé dans le domaine du contrôle d'accès: la confiance. Nous définissons donc la valeur de la menace comme l'inverse du niveau de confiance de l'utilisateur. De façon simple, un utilisateur qui possède un bon niveau de confiance, sera moins disposé à faire une action malveillante qu'un autre.

Nous avons effectué différentes expériences avec notre modèle qui montre des résultats intéressants. Premièrement, la valeur de risque ajoute une information utile pour le contrôle d'accès, d'autant plus qu'il s'adapte correctement au comportement du système: un système avec beaucoup de rejets par défaut aura un taux de risque moyen plus élevé. Ensuite, il n'y a pas d'augmentation drastique en terme de rejet de requêtes. Donc l'introduction de cette métrique de risque a une influence raisonnable. Enfin, les rejets basés sur le risque sont globalement cohérents avec les rejets par la politique normale.

6.3 Cas d'étude réel

Durant ce projet de thèse nous avons collaboré avec une entreprise confronté au problème de migration de leur infrastructure IT vers des services cloud. Rapidement, il fallait faire face à la problématique de comparer différentes offres cloud entre elles. Cela nous a donc permis d'appliquer notre modèle dans un cas concret et de le valider partiellement.

Dans le cadre de cette étude, deux fournisseurs de services cloud ont été contacté pour qu'ils répondent au questionnaire CAIQ publié par le CSA. En analysant les réponses à ce questionnaire nous étions capable de comparer les deux offres basé sur les contrôles de sécurité qu'ils implémentent ou non. En appliquant notre modèle d'évaluation, nous avons pu calculer le score de **couverture** de chacun de ces fournisseurs par rapport aux 9 menaces données par le CSA.

En y ajoutant les la pondération due au **dommage** éventuel que la réalisation d'une telle menace pourrait avoir, nous avons pu établir les valeurs de risques présentées dans FIGURE 6.4.

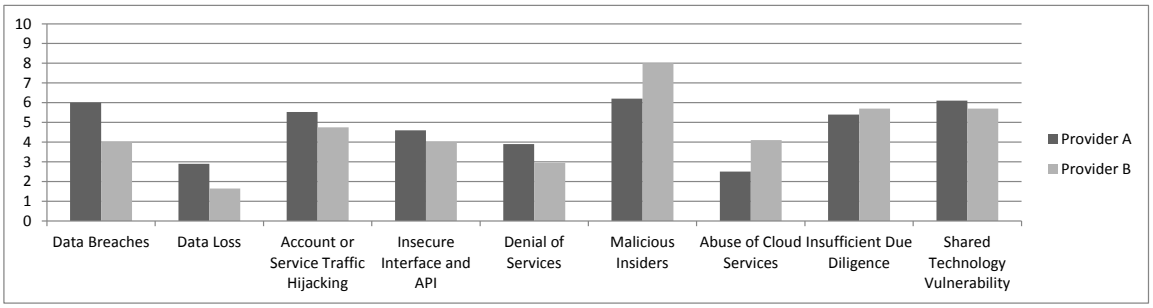

FIGURE 6.4 – Niveaux de risque des deux fournisseurs cloud

Finalement, c'est *Provider A* qui a été retenu comme offre finale pour la migration. Globalement, le retour sur cette étude a été positive et que le modèle était bien adapté à ce type de comparaison. Bien entendu d'autres critères sont à prendre en compte (qui ne sont pas forcément quantifiables), mais le modèle permet de prendre en compte les aspects de sécurité de façon relativement efficace. L'avantage principal de ce modèle est de fournir des métriques pour la sécurité.

Chapitre 7

Conclusion

7.1 Résumé des contributions

Dans cette thèse nous avons fait différentes contributions concernant le déploiement en confiance de processus métiers dans un environnement cloud. Tout d'abord, une méthode supporte l'adaptation semi-automatique de modèles de processus en se basant sur des objectifs de sécurité et des contraintes organisationnelles. Un processus existant peut ainsi être transformé en processus sensibilisé aux risques de sécurité liés au cloud. Ensuite, un modèle d'évaluation des risques cloud supporte l'estimation du risque de déployer des application dans des environnements cloud. Ce modèle prend en compte des annotations sur des modèles de processus pour évaluer leurs exigences de sécurité et analyse les informations données par les fournisseurs cloud pour estimer leurs réponses aux risques. Finalement, nous présentons un framework supportant le déploiement automatique de processus métiers dans un environnement multi-cloud. Nous avons inclus dans ces travaux existants un algorithme d'optimisation multi-critères pour sélectionner la configuration la plus adaptée selon différents critères de qualité de service.

Nous prétendons que notre contribution aide à augmenter la confiance que les entreprises peuvent avoir en déployant leurs processus dans un environnement cloud. Notre stratégie globale se divise en deux facettes. D'une part nous adaptons les processus pour mieux gérer les menaces liées au contexte du cloud. Et d'autre part nous évaluer le niveau de sécurité d'offres cloud pour quantifier le risque et sélectionner la meilleure configuration possible. Cette approche est supporté par un outil pour automatiser la transformation, l'évaluation et le déploiement en prenant en compte d'autres critères important en plus du risque.

La validation de nos travaux a été réalisée de différentes façons. Tout d'abord, pour démontrer la faisabilité de l'approche, nous l'avons implémenté sur différents prototypes: une interface pour évaluer la sécurité d'offres cloud, un algorithme d'optimisation multi-critère intégré à un framework de décomposition de processus et enfin un outil pour déployer automatiquement et contrôler des processus dans des environnements cloud. Ensuite, nous avons adapté notre modèle d'évaluation des risques cloud à un système de contrôle d'accès pour l'améliorer avec des métriques de risque. Finalement, nous avons réalisé une étude comparative sur des offres clouds réelles avec notre modèle d'évaluation des risques pour conseiller une entreprise sur une possible externalisation.

7.2 Limites et perspectives

Dans l'objectif d'adresser certaines limites de nos travaux, nous proposons différentes perspectives qui pourraient améliorer ou approfondir les contributions de cette thèse.

Modifications durant l'exécution L'un des premiers aspects qui n'a pas été adressé dans nos travaux est la nature dynamique de l'environnement cloud. Notre approche se place lors de la *conception* et n'intègre pas les changements qui peuvent survenir lors de l'exécution du processus. Ces changements peuvent survenir de l'environnement ou du processus en lui-même. En effet, les offres cloud disponibles peuvent changer (en terme de coût, de qualité et de sécurité) mais même les risques eux-mêmes peuvent évoluer. Ainsi, la configuration choisie lors du déploiement peut ne plus l'être par la suite. Il serait intéressant de prendre en compte ce type de changements, comme par exemple au travers un re-déploiement dynamique, voir un déploiement à-la-volée. La Cloud Security Alliance travaille dans cette direction en proposant différent niveaux de certification, dont l'un intègre le monitoring continu du niveau de sécurité des fournisseurs cloud. Les travaux présentés dans [BGPG15] montrent que notre modèle reste compatible dans un contexte dynamique. De plus, notre décision d'utiliser des heuristiques pour l'algorithme de sélection optimisé est déjà motivé par cette perspective: le temps nécessaire pour trouver une bonne solution de déploiement est un facteur très important lorsque l'on travaille dans un contexte "à l'exécution".

Automatisation de la définition d'exigences de sécurité Un autre aspect qui n'a pas été exploré dans cette thèse est la définition précise des exigences en terme de sécurité. Nous nous sommes conformé aux directives des méthode d'évaluation de risques les plus courantes qui supposent que ces informations sont donnée de façon manuelle par des experts en sécurité. Cependant, ces informations existent peut-être déjà autre part, et il serait intéressant de les générer automatiquement. Il existe différents modèles formels pour définir des exigences de sécurité (comme Secure Tropos [MMGG03]) qui pourraient être utilisés pour explorer ces perspectives. En effet, un inconvénient majeur de l'approche actuelle est que les exigences de sécurité sont subjectifs: deux personnes différentes peuvent définir des exigences différentes. En les générant automatiquement, les niveaux de sécurité ne serait plus sujet à interprétations. Ou du moins, empêcher de réaliser la même tâche plusieurs fois (puisque ces informations sont probablement déjà renseigné à un autre moment lors de la définition des processus).

Offuscation La dernière, et probablement la plus intéressante des perspective est celle d'offusquer des modèles de processus avant de les déployer dans un environnement cloud. Une technique déjà utilisé en programmation, et qui consiste à cacher le code source des applications, pourrait être source d'inspiration pour ce type de travaux. Pour ce faire, les processus métier doivent être évalués automatiquement quant à leur complexité pour identifier les zones critiques qui nécessitent le plus de protection. De plus, des patrons de sécurité, comme la redondance, de faux messages/tâches ou des contraintes de séparation pourraient être développés. De tels patrons pourraient être automatiquement intégré dans un processus existant pour augmenter sa complexité et réduire la probabilité qu'un attaquant comprenne le processus. De première avancées dans cette direction ont été publiées dans [GANYG15]. L'objectif principal de cet contribution est de cacher le savoir-faire contenu dans un processus avant de le déployer sur des services cloud.

Bibliographie

[AAHR10] Wil M.P. Van Der Aalst, Michael Adams, Arthur Ter Hofstede, and Nick Russell. *Modern Business Process Automation - YAWL and its Support Environment*. Springer Berlin Heidelberg, 2010.

[Aal11] Wil M.P. Van Der Aalst. Business process configuration in the cloud: How to support and analyze multi-tenant processes ? In *Web Services (ECOWS), 2011 Ninth IEEE European Conference on*, pages 3–10, Sept 2011.

[AFG+09] Michael Armbrust, Armando Fox, Rean Griffith, Anthony D. Joseph, Randy H. Katz, Andrew Konwinski, Gunho Lee, David A. Patterson, Ariel Rabkin, and Matei Zaharia. Above the clouds: A berkeley view of cloud computing. Technical report, 2009.

[AJG+15] Abdelzahir Abdelmaboud, Dayang N.A. Jawawi, Imran Ghani, Abubakar Elsafi, and Barbara Kitchenham. Quality of service approaches in cloud computing: A systematic mapping study. *Journal of Systems and Software*, 101:159–179, mar 2015.

[ALMS09] T. Anstett, F. Leymann, R. Mietzner, and S. Strauch. Towards bpel in the cloud: Exploiting different delivery models for the execution of business processes. In *Services - I, 2009 World Conference on*, pages 670–677, July 2009.

[AM13] Naved Ahmed and Raimundas Matulevicius. A taxonomy for assessing security in business process modelling. In *RCIS*, pages 1–10, 2013.

[AM14] Naved Ahmed and Raimundas Matulevičius. Securing business processes using security risk-oriented patterns. *Comput. Stand. Interfaces*, 36(4):723–733, Jun 2014.

[AZ/04] AS/NZS 4360 SET Risk Management, Australian/New Zealand Standards, 2004.

[BBDA12] Mehdi Bentounsi, Salima Benbernou, Cheikh S. Deme, and Mikhail J. Atallah. Anonyfrag: an anonymization-based approach for privacy-preserving bpaas. In *1st International Workshop on Cloud Intelligence (colocated with VLDB 2012), Cloud-I '12, Istanbul, Turkey, August 31, 2012*, page 9, 2012.

[BePP98] Rainer E. Burkard, Eranda Çela, Panos M. Pardalos, and Leonidas S. Pitsoulis. The quadratic assignment problem. In *Handbook of Combinatorial Optimization*, pages 241–238. Kluwer Academic Publishers, Dordrecht, 1998.

[BGPG15] Ahmed Bouchami, Elio Goettelmann, Olivier Perrin, and Claude Godart. Enhancing access control with risk metrics in collaborative federated cloud environments. In *14th IEEE International Conference on Trust, Security and Privacy in Computing and Communications, TrustCom 2015, Helsinki, Finland, August 20-22, 2015*, 2015.

[BH14] Amid Khatibi Bardsiri and Seyyed Mohsen Hashemi. Qos metrics for cloud computing services evaluation. *International Journal of Intelligent Systems and Applications (IJISA)*, 2014.

[BK13] Niyati Baliyan and Sandeep Kumar. Quality assessment of software as a service on cloud using fuzzy logic. In *2013 IEEE International Conference on Cloud Computing in Emerging Markets (CCEM)*. IEEE, oct 2013.

[BKNT11] Christian Baun, Marcel Kunze, Jens Nimis, and Stefan Tai. *Cloud Computing: Web-Based Dynamic IT Services*. Springer Publishing Company, Incorporated, 1st edition, 2011.

[BLB15] Matthias Becker, Sebastian Lehrig, and Steffen Becker. Systematically deriving quality metrics for cloud computing systems. In *Proceedings of the 6th ACM/SPEC International Conference on Performance Engineering - ICPE'15*. ACM Press, 2015.

[BYOG13] Kahina Bessai, Samir Youcef, Ammar Oulamara, and Claude Godart. Bi-criteria strategies for business processes scheduling in cloud environments with fairness metrics. In *IEEE 7th International Conference on Research Challenges in Information Science, RCIS 2013, Paris, France, May 29-31, 2013*, pages 1–10, 2013.

[CAM10] Common Assurance Maturity Model Guiding Principles. `http://www.common-assurance.com/resources/Common-Assurance-Maturity-Model-vision.pdf`, 2010.

[CBT11] Shankar Babu Chebrolu, Vinay Bansal, and Pankaj Telang. Top 10 cloud risks that will keep you awake at night. Technical report, 2011. `https://www.owasp.org/images/4/47/Cloud-Top10-Security-Risks.pdf`.

[CFBH07] B. Carminati, E. Ferrari, R. Bishop, and P.C.K. Hung. Security conscious web service composition with semantic web support. In *Data Engineering Workshop, 2007 IEEE 23rd International Conference on*, pages 695–704, April 2007.

[Clo14] Cloud Security Alliance. Security, Trust and Assurance Registry. `https://cloudsecurityalliance.org/star/`, 2014.

[CLRA13] Raffaele Conforti, Massimiliano De Leoni, Marcello La Rosa, and Wil M.P. Van Der Aalst. Supporting risk-informed decisions during business process execution. In Camille Salinesi, MoiraC. Norrie, and Óscar Pastor, editors, *Advanced Information Systems Engineering*, volume 7908 of *Lecture Notes in Computer Science*, pages 116–132. Springer Berlin Heidelberg, 2013.

[CSA13] The Notorious Nine - Cloud Computing Top Threats in 2013. Technical report, Cloud Security Alliance, 2013. `https://downloads.cloudsecurityalliance.org/initiatives/top_threats/The_Notorious_Nine_Cloud_Computing_Top_Threats_in_2013.pdf`.

[CSA14] Cloud Control Matrix. Technical report, Cloud Security Alliance, 2014. `https://cloudsecurityalliance.org/download/cloud-controls-matrix-v3/`.

[DRA05] Alexander Dreiling, Michael Rosemann, and Wil M.P. Van Der Aalst. From conceptual process models to running workflows : a holistic approach for the configuration of enterprise systems. In *PACIS'05*, pages 363–376, 2005.

[DRMR13] Marlon Dumas, Marcello La Rosa, Jan Mendling, and Hajo A. Reijers. *Fundamentals of Business Process Management*. Springer, 2013.

[EJF+14] Seven Euting, Christian Janiesch, Robin Fischer, Stefan Tai, and Ingo Weber. Scalable business process execution in the cloud. In *2014 IEEE International Conference on Cloud Engineering, Boston, MA, USA, March 11-14, 2014*, pages 175–184, 2014.

[ENI09a] Benefits, risks and recommendations for information security. Technical report, European Network and Information Security Agency, 2009. `http://www.enisa.europa.eu/activities/risk-management/files/deliverables/cloud-computing-risk-assessment/at_download/fullReport`.

[ENI09b] Information Assurance Framework. Technical report, European Network and Information Security Agency, 2009. `https://www.enisa.europa.eu/activities/risk-management/files/deliverables/cloud-computing-information-assurance-framework/`.

[Eur12] EuroCloud Deutschland eco e.V. Eurocloud Star Audit. `https://eurocloud-staraudit.eu/`, 2012.

[FDGG14] Walid Fdhila, Marlon Dumas, Claude Godart, and Luciano García-Bañuelos. Heuristics for composite web service decentralization. *Software and System Modeling*, 13(2):599–619, 2014.

[Fil12] H.-G. Fill. Using obfuscating transformations for supporting the sharing and analysis of conceptual models. In Susanne Robra-Bissantz and Dirk Mattfeld, editors, *Multikonferenz Wirtschaftsinformatik 2012 - Teilkonferenz Modellierung betrieblicher Informationssysteme*, Braunschweig, 2012. GITO Verlag.

[FJ12] Ales Frece and Matjaz B. Juric. Modeling functional requirements for configurable content- and context-aware dynamic service selection in business process models. *Journal of Visual Languages & Computing*, 23(4):223 – 247, 2012.

[FY10] Z. Fang and C. Yin. Bpm architecture design based on cloud computing. In *Intelligent Information Management, Vol. 2 No. 5*, pages 329–333, 2010.

[FYG09] Walid Fdhila, Ustun Yildiz, and Claude Godart. A flexible approach for automatic process decentralization using dependency tables. In *IEEE International Conference on Web Services, ICWS 2009, Los Angeles, CA, USA, 6-10 July 2009*, pages 847–855, 2009.

[GANYG15] Elio Goettelmann, Amina Ahmed-Nacer, Samir Youcef, and Claude Godart. Paving the way towards semi-automatic design-time business process model obfuscation. In *Web Services (ICWS), 2015 IEEE International Conference on*, June 2015.

[Gar13] Gartner. Gartner says worldwide public cloud services market to total $131 billion. `http://www.gartner.com/newsroom/id/2352816`, 2013. [Online ; accessed 21-May-2013].

[GBO+07] Tyrone Grandison, Marcel Bilger, L. O'Connor, Marcel Graf, Morton Swimmer, Matthias Schunter, Andreas Wespi, and Nev Zunic. Elevating the discussion on security management: The data centric paradigm. In *Proceedings of BDIM 2007, 2nd IEEE/IFIP International Workshop on Business-Driven IT Management, May 21, 2007, Munich, Germany*, pages 84–93, 2007.

[GDG+14] Elio Goettelmann, Karim Dahman, Benjamin Gâteau, Eric Dubois, and Claude Godart. A security risk assessment model for business process deployment in the cloud. In *IEEE International Conference on Services Computing, SCC 2014, Anchorage, AK, USA, June 27 - July 2, 2014*, pages 307–314, 2014.

[GDGG14] Elio Goettelmann, Karim Dahman, Benjamin Gâteau, and Claude Godart. A formal broker framework for secure and cost-effective business process deployment on multiple clouds. In *Information Systems Engineering in Complex Environments - CAiSE Forum 2014, Thessaloniki, Greece, June 16-20, 2014, Selected Extended Papers*, pages 3–19, 2014.

[Gen09] Craig Gentry. Fully homomorphic encryption using ideal lattices. In *Proceedings of the 41st Annual ACM Symposium on Theory of Computing, STOC 2009, Bethesda, MD, USA, May 31 - June 2, 2009*, pages 169–178, 2009.

[GFG13] Elio Goettelmann, Walid Fdhila, and Claude Godart. Partitioning and cloud deployment of composite web services under security constraints. In *2013 IEEE International Conference on Cloud Engineering, IC2E 2013, San Francisco, CA, USA, March 25-27, 2013*, pages 193–200, 2013.

[GL97] Fred Glover and Manuel Laguna. *Tabu Search*. Kluwer Academic Publishers, Norwell, MA, USA, 1997.

[GMG13] Elio Goettelmann, Nicolas Mayer, and Claude Godart. A general approach for a trusted deployment of a business process in clouds. In *Fifth International Conference on Management of Emergent Digital EcoSystems, MEDES '13, Luxembourg, Luxembourg, October 29-31, 2013*, pages 92–99, 2013.

[GMG14] Elio Goettelmann, Nicolas Mayer, and Claude Godart. Integrating security risk management into business process management for the cloud. In *IEEE 16th Conference on Business Informatics, CBI 2014, Geneva, Switzerland, July 14-17, 2014 - Volume 1*, pages 86–93, 2014.

[GMR+12] Nelson Gonzalez, Charles Miers, Fernando Redigolo, Marcos Simplicio, Tereza Carvalho, Mats Näslund, and Makan Pourzandi. A quantitative analysis of current security concerns and solutions for cloud computing. *Journal of Cloud Computing*, 2012.

[HM79] Ching-Lai Hwang and Abu Syed Md. Masud. Methods for multiple objective decision making. In *Multiple Objective Decision Making — Methods and Applications*, volume 164 of *Lecture Notes in Economics and Mathematical Systems*, pages 21–283. Springer Berlin Heidelberg, 1979.

[Hul08] Richard Hull. Artifact-centric business process models: Brief survey of research results and challenges. In Robert Meersman and Zahir Tari, editors, *On the Move to Meaningful Internet Systems: OTM 2008*, volume 5332 of *Lecture Notes in Computer Science*, pages 1152–1163. Springer Berlin Heidelberg, 2008.

[ISO11] ISO/IEC 27005, Information tech., Security techniques, Information security risk management, 2011.

[ISO15] ISO/IEC 27017, Information tech., Security techniques, Code of practice for information security controls for cloud computing services based on ISO/IEC 27002, 2015. Status: under development.

[JLW+11] Jiulei Jiang, Jiajin Le, Yan Wang, Jie Sun, and Feng He. The bpm architecture based on cloud computing. In *Knowledge Acquisition and Modeling (KAM), 2011 Fourth International Symposium on*, pages 196–198, Oct 2011.

[JSB+11] M. Jensen, J. Schwenk, J. Bohli, N. Gruschka, and L.L. Iacono. Security prospects through cloud computing by adopting multiple clouds. In *CLOUD'11*, pages 565 –572, 2011.

[Jür02] Jan Jürjens. Umlsec: Extending uml for secure systems development. In Jean-Marc Jézéquel, Heinrich Hussmann, and Stephen Cook, editors, *UML 2002 – The Unified Modeling Language*, volume 2460 of *Lecture Notes in Computer Science*, pages 412–425. Springer Berlin Heidelberg, 2002.

[KH07] Pawan Khera and Bill Hefley. eSourcing capability model for client organizations (eSCM-CL) annotated bibliography. *SSRN Journal*, 2007.

[KLW08] Santhosh Kumaran, Rong Liu, and Frederick Wu. On the duality of information-centric and activity-centric models of business processes. In *Advanced Information Systems Engineering*, volume 5074 of *Lecture Notes in Computer Science*, pages 32–47. Springer Berlin Heidelberg, 2008.

[KR01] Konstantin Knorr and Susanne Röhrig. Security requirements of e-business processes. In Beat Schmid, Katarina Stanoevska-Slabeva, and Volker Tschammer, editors, *Towards the E-Society*, volume 74 of *IFIP International Federation for Information Processing*, pages 72–86. Springer US, 2001.

[LTM+11] Fang Liu, Jin Tong, Jian Mao, Rober Bohn, John Messina, Lee Badger, and Dawn Leaf. NIST cloud computing reference architecture. Technical report, National Institute of Standards and Technology (NIST), 2011.

[MAA11] Carlos Monsalve, Alain April, and Alain Abran. Requirements elicitation using bpm notations: Focusing on the strategic level representation. ACACOS'11, pages 235–241, 2011.

[May09] Nicolas Mayer. *Model-based Management of Information System Security Risk*. PhD thesis, University of Namur, Apr 2009.

[MF02] Peter Merz and Bernd Freisleben. Greedy and local search heuristics for unconstrained binary quadratic programming. *Journal of Heuristics*, 8(2):197–213, 2002.

[MF12] David Martinho and Diogo R. Ferreira. Securely storing and executing business processes in the cloud. In *Business Process Management Workshops - BPM 2012 International Workshops, Tallinn, Estonia, September 3, 2012. Revised Papers*, pages 707–712, 2012.

[MG11] Peter Mell and Timothy Grance. The NIST definition of cloud computing. Technical report, National Institute of Standards and Technology (NIST), 2011.

[MH06] Michael Zur Muehlen and Danny Ting-Yi Ho. Risk management in the bpm lifecycle. In Christoph J. Bussler and Armin Haller, editors, *Business Process Management Workshops*, volume 3812 of *Lecture Notes in Computer Science*, pages 454–466. Springer Berlin Heidelberg, 2006.

[MJ10] Vinod Muthusamy and Hans-Arno Jacobsen. Bpm in cloud architectures: Business process management with slas and events. In Richard Hull, Jan Mendling, and Stefan Tai, editors, *Business Process Management*, volume 6336 of *Lecture Notes in Computer Science*, pages 5–10. Springer Berlin Heidelberg, 2010.

[MLB+11] Sean Marston, Zhi Li, Subhajyoti Bandyopadhyay, Juheng Zhang, and Anand Ghalsasi. Cloud computing - the business perspective. *Decision Support System*, Apr 2011.

[MMGG03] Haralambos Mouratidis, Gordon A. Manson, Abdullah Gani, and Paolo Giorgini. Analysing security requirements of information systems using tropos. Angers, France, 2003.

[MMM+08] Raimundas Matulevicius, Nicolas Mayer, Haralambos Mouratidis, Eric Dubois, Patrick Heymans, and Nicolas Genon. Adapting secure tropos for security risk management in the early phases of information systems development. In *Advanced Information Systems Engineering, 20th International Conference, CAISE 2008, Montpellier, France, June 16-20, 2008, Proceedings*, pages 541–555, 2008.

[MPR+09] Gerald Münzl, Bernhard Przywra, Martin Reti, Jörg Schäfer, Karin Sondermann, Mathias Weber, and Andreas Wilker. Cloud computing - evolution in der technik, revolution im business. Technical report, 2009.

[MR14] Juergen Mangler and Stefanie Rinderle-Ma. CPEE - cloud process execution engine. In *Proceedings of the BPM Demo Sessions 2014 Co-located with the 12th International Conference on Business Process Management (BPM 2014), Eindhoven, The Netherlands, September 10, 2014.*, page 51, 2014.

[MS95] Salvatore T. March and Gerald F. Smith. Design and natural science research on information technology. *Decis. Support Syst.*, 15(4):251–266, dec 1995.

[MSSN04] J. Mendling, M. Strembeck, G. Stermsek, and G. Neumann. An approach to extract rbac models from bpel4ws processes. In *Enabling Technologies: Infrastructure for Collaborative Enterprises, 2004. WET ICE 2004. 13th IEEE International Workshops on*, pages 81–86, June 2004.

[NIS02] Information security - guide for conducting risk assessments. Technical report, National Institute of Standards and Technology, 2002.

[OAS07] Web Services Business Process Execution Language (WSBPEL) 2.0. Technical report, Organization for the Advancement of Structured Information Standards (OASIS), 2007.

[OBG13] Wendpanga Francis Ouedraogo, Frédérique Biennier, and Parisa Ghodous. Model driven security in a multi-cloud context. *IJEBM*, 11(3), 2013.

[OMG11] Business Process Model and Notation (BPMN) 2.0. Technical report, Object Management Group (OMG), 2011.

[OMG13] Unified Modelling Language (UML) 2.5. Technical report, Object Management Group (OMG), 2013.

[PGPM12] Elda Paja, Paolo Giorgini, Stéphane Paul, and PerHåkon Meland. Security requirements engineering for secure business processes. In Laila Niedrite, Renate Strazdina, and

Benkt Wangler, editors, *Workshops on Business Informatics Research*, volume 106 of *Lecture Notes in Business Information Processing*, pages 77–89. Springer Berlin Heidelberg, 2012.

[PPKW11] M. Pathirage, S. Perera, I. Kumara, and S. Weerawarana. A multi-tenant architecture for business process executions. In *Web Services (ICWS), 2011 IEEE International Conference on*, pages 121–128, July 2011.

[RFMP07] Alfonso Rodríguez, Eduardo Fernández-Medina, and Mario Piattini. A bpmn extension for the modeling of security requirements in business processes. *IEICE - Trans. Inf. Syst.*, E90-D(4):745–752, Mar 2007.

[SLK09] Stefan Sackmann, Lutz Lowis, and Kai Kittel. A risk based approach for selecting services in business process execution. In *Wirtschaftsinformatik (1)*, pages 357–366, 2009.

[Sta08] Richard Stallman. Cloud computing is a trap, warns GNU founder Richard Stallman. `http://www.guardian.co.uk/technology/2008/sep/29/cloud.computing.richard.stallman`, 2008. [Online ; accessed 21-May-2013].

[TBCB12] Sameh Hbaieb Turki, Farah Bellaaj, Anis Charfi, and Rafik Bouaziz. Modeling security requirements in service based business processes. In *Enterprise, Business-Process and Information Systems Modeling - 13th International Conference, BPMDS 2012, 17th International Conference, EMMSAD 2012, and 5th EuroSymposium, held at CAiSE 2012, Gdańsk,Poland, June 25-26, 2012. Proceedings*, pages 76–90, 2012.

[TJG+11] S. Tjoa, S. Jakoubi, G. Goluch, G. Kitzler, S. Goluch, and G. Quirchmayr. A formal approach enabling risk-aware business process modeling and simulation. *Services Computing, IEEE Transactions on*, 4(2):153–166, Apr 2011.

[Wat12] Paul Watson. A multi-level security model for partitioning workflows over federated clouds. *Journal of Cloud Computing*, 1(1), 2012.

[Wes12] Mathias Weske. *Business Process Management - Concepts, Languages, Architectures, 2nd Edition*. Springer, 2012.

[Wie09] Roel Wieringa. Design science as nested problem solving. In *Proceedings of the 4th International Conference on Design Science Research in Information Systems and Technology, DESRIST 2009, Philadelphia, Pennsylvania, USA, May 7-8, 2009*, 2009.

[Wik13] Wikipedia. Cloud computing (en). `http://en.wikipedia.org/wiki/Cloud_computing`, 2013. [Online ; accessed 05-April-2013].

[WMM08] Christian Wolter, Michael Menzel, and Christoph Meinel. Modelling security goals in business processes. In *In Modellierung 2008, volume P-127 of LNI*, pages 201–216. Köllen, 2008.

[WMS+09] Christian Wolter, Michael Menzel, Andreas Schaad, Philip Miseldine, and Christoph Meinel. Model-driven business process security requirement specification. *Journal of Systems Architecture*, 55(4):211–223, 2009. Secure Service-Oriented Architectures (Special Issue on Secure SOA).

[Woz12] Steve Wozniak. Apple co-founder wozniak sees trouble in the cloud. `http://www.google.com/hostednews/afp/article/ALeqM5h1p0LVc4iFZxbWlflFGgcHhbRNCQ`, 2012. [Online ; accessed 21-May-2013].

[WWHJ12] Sven Wenzel, Christian Wessel, Thorsten Humberg, and Jan Jürjens. Securing processes for outsourcing into the cloud. In *CLOSER 2012 - Proceedings of the 2nd International Conference on Cloud Computing and Services Science, Porto, Portugal, 18 - 21 April, 2012*, pages 675–680, 2012.

[ZSM+10] Wenchao Zhou, Micah Sherr, William R. Marczak, Zhuoyao Zhang, Tao Tao, Boon Thau Loo, and Insup Lee. Towards a data-centric view of cloud security. In *Proceedings of the Second International CIKM Workshop on Cloud Data Management, CloudDb 2010, Toronto, Ontario, Canada, October 30, 2010*, pages 25–32, 2010.

Abstract

Nowadays service ecosystems rely on dynamic software service chains that span over multiple organisations and providers. They provide an agile support for business applications, governments of end-users. This trend is reinforced by the Cloud based economy that allows sharing of costs and resources. However, the lack of trust in such cloud environments, that involve higher security requirements, is often seen as a braking force to the development of such services.

The objective of this thesis is to study the concepts of service orchestration and trust in the context of the Cloud. It proposes an approach which supports a trust model in order to allow the orchestration of trusted business process components on the cloud.

The contribution is threefold and consists in a method, a model and a framework. The method categorizes techniques to transform an existing business process into a risk-aware process model that takes into account security risks related to cloud environments. The model formalizes the relations and the responsibilities between the different actors of the cloud. This allows to identify the different information required to assess and quantify security risks in cloud environments. The framework is a comprehensive approach that decomposes a business process into fragments that can automatically be deployed on multiple clouds. The framework also integrates a selection algorithm that combines security information with other quality of service criteria to generate an optimized configuration.

Finally, the work is implemented in order to validate the approach. The framework is implemented in a tool. The security assessment model is also applied over an access control model. The last part presents the results of the implementation of our work on a real world use case.

Keywords: Business Process Management, Cloud Computing, Security Risk Management

Résumé

L'essor du Cloud Computing, permettant de partager les coûts et les ressources au travers de la virtualisation, présage une interconnexion dynamique et flexible entre entreprises et fournisseurs. Cependant, cette mise en commun de ressources, données et savoir-faire implique de nouvelles exigences en termes de sécurité. En effet, le manque de confiance dans les structures du Cloud est souvent vu comme un frein au développement de tels services.

L'objectif de cette thèse est d'étudier les concepts d'orchestration de services, de confiance et de gestion des risques dans le contexte du Cloud. La contribution principale est un framework permettant de déployer des processus métiers dans un environnement Cloud, en limitant les risques de sécurité liés à ce contexte.

La contribution peut être séparée en trois partie distinctes qui prennent la forme d'une méthode, d'un modèle et d'un framework. La méthode catégorise des techniques pour transformer un processus métier existant en un modèle sensibilisé (ou averti) qui prend en compte les risques de sécurité spécifiques aux environnements Cloud. Le modèle formalise les relations et les responsabilités entre les différents acteurs du Cloud. Ce qui permet d'identifier les différentes informations requises pour évaluer et quantifier les risques de sécurité des environnements Cloud. Le framework est une approche complète de décomposition de processus en fragments qui peuvent être automatiquement déployés sur plusieurs Clouds. Ce framework intègre également un algorithme de sélection qui combine les information de sécurité avec d'autres critère de qualité de service pour générer des configuration optimisées.

Finalement, les travaux sont implémentés pour démontrer la validité de l'approche. Le framework est implémenté dans un outil. Le modèle d'évaluation des risques de sécurité Cloud est également appliqué dans un contexte de contrôle d'accès. La dernière partie présente les résultats de l'implémentation de nos travaux sur un cas d'utilisation réel.

Mots-clés: Gestion des Processus Métiers, Cloud Computing, Gestion des Risques de Sécurité