Table des matières

Table des Figures

Liste des Tables

Liste des Algorithmes

Chapitre 1 : Introduction Générale

1.1. Introduction

La demande croissante pour de nouveaux services informatiques plus économiques a permis l'émergence d'une nouvelle architecture qu'est le **Cloud Computing**.
Le Cloud Computing représente la cinquième génération de l'informatique après les mainframes, les ordinateurs personnels, le paradigme client/serveur et le web (World Wide Web). Il désigne un modèle dans lequel les ressources telles que la puissance de calcul, le stockage ou encore la bande passante sont fournies comme des services qui peuvent être loués par des utilisateurs via Internet à la demande.

Cette technologie offre plusieurs avantages comme un déploiement rapide, un paiement à l'usage, une réduction de coûts, une évolutivité facile, une délivrance de service plus rapide, un accès au réseau omniprésent, une plus grande résilience…etc. en raison de ces diverses caractéristiques, il devient une solution intéressante pour les entreprises et les chercheurs.

Cependant, son adoption est confrontée à un certain nombre de défis, tels que les problèmes de sécurité, les défis juridiques et de conformité et les défis organisationnels (Andrei & Jain, 2009) (Buyya, Yeo, & Venugopal, 2008) (Catteddu & Hogben, 2009) (Khajeh-Hosseini, Greenwood, & Sommerville, 2010). Tous ces défis présentent un élément commun qui est la problématique de sûreté entre les consommateurs et les fournisseurs, puisque le Cloud Computing exige de faire confiance aux fournisseurs sur la gestion des ressources informatiques et l'administration des données. En conséquence, la confiance représente l'un des principaux freins pour l'adoption de ce nouveau paradigme.

À travers cette thèse, nous procédons à l'identification des défis liés à l'adoption du Cloud Computing. Nous nous intéressons en particulier aux risques liés à la confiance en proposant un système de gestion de confiance pour la sélection de services fiables et sûrs dans un environnement du Cloud Computing.

1.1.1. Définition du problème

Ces dernières années, l'étude de la confiance a reçu beaucoup d'attention dans les environnements distribués en général et dans le Cloud Computing en particulier. Toutefois, les diverses recherches menées sur l'adoption du Cloud Computing et en particulier les recherches sur la confiance des consommateurs sont minimes et sous-estimées.

Cette thèse propose de fournir un système de gestion de confiance pour la sélection de fournisseurs de services du Cloud Computing, en examinant les considérations de sécurité, les considérations de conformité et les considérations de qualité de service. Pour cela, plusieurs questions sont posées:

Question 1: *Quels sont les principaux obstacles de sécurité à l'adoption du Cloud Computing?*

Question 2 : *Comment concevoir une architecture générique permettant de réduire les risques liés à l'adoption du Cloud Computing?*

Question 3 : *Quels facteurs et qualités de service influent la fiabilité et la sûreté de service?*

Question 4 : *Comment amener les consommateurs à faire confiance aux fournisseurs de services?*

Question 5 : *Est-il possible pour le client et le fournisseur de se faire confiance mutuellement et de collaborer pour la réussite de l'adoption d'un projet en Cloud?*

À partir de ces diverses interrogations notre problématique générale pourrait se résumer avec l'interrogation suivante : « ***Comment faire confiance à un fournisseur de services dans un environnement du Cloud Computing?*** »

1.1.2. Objectifs

Bien que le Cloud Computing ait connu une évolution très rapide, il reste encore plusieurs barrières de sécurité qui découragent les utilisateurs à adopter ce système. Dernièrement, la sécurité a été répertoriée comme la préoccupation la plus importante du Cloud Computing, selon l'enquête menée par International Data Corporation (IDC) [1], [2]. Le National Institute of Standards and Technology (NIST) avait aussi exprimé ses préoccupations sur la sécurité par : «*les défis de sécurité [que] le Cloud Computing offre sont formidables*», [3]. Les chercheurs du « Massachusetts Institute of Technology (MIT) » ont également déclaré que *la sécurisation du Cloud Computing est "un grand défi de technologie de l'information"*

[4]. Ces préoccupations sur les questions de sécurité dans le Cloud Computing servent comme principales motivations derrière cette recherche.

De plus, lorsque l'on considère d'adopter un service en Cloud, la première question que tout consommateur se pose est « j*usqu'à quel point nos données seront sécurisées dans les mains d'un fournisseur de services ?"*. Tout incident relatif à la vie privée ou à la sécurité, même mineur, est capable de détruire la confiance des consommateurs.

L'objectif principal de cette thèse est d'étudier les diverses questions de sécurité relatives au Cloud Computing et plus particulièrement les défis se rapportant à la confiance. Il s'agit de concevoir et de réaliser un système de gestion de confiance pour la sélection d'un fournisseur de services en Cloud sûr et fiable. Pour atteindre cet objectif, nous contribuons à :

1) étudier les défis et challenges de la sélection de service fondée sur la confiance dans les environnements de Cloud Computing ;
2) examiner et comparer les approches courantes de sélection de service ;
3) construire un modèle de calcul de confiance dans un environnement de Cloud ;
4) concevoir un système de confiance pour la sélection de service afin de surmonter les limites des systèmes actuels ;
5) proposer un modèle d'attaque pour filtrer les avis malhonnêtes et fiabiliser la sélection de service.

1.1.3. Méthodologie

L'approche adoptée lors de nos recherches consistes-en :

- la définition de l'environnement et du domaine ;
- la définition des variables de confiance influant le Cloud Computing ;
- la conception de l'architecture générale du système proposé en se basant sur les résultats des deux premières phases ;
- la proposition d'un modèle de calcul de la confiance pour la sélection de fournisseurs et un modèle d'attaque pour le filtrage des consommateurs malhonnêtes ;
- l'implémentation du système de gestion de confiance proposé.

La figure.1.1 décrit ces différentes étapes :

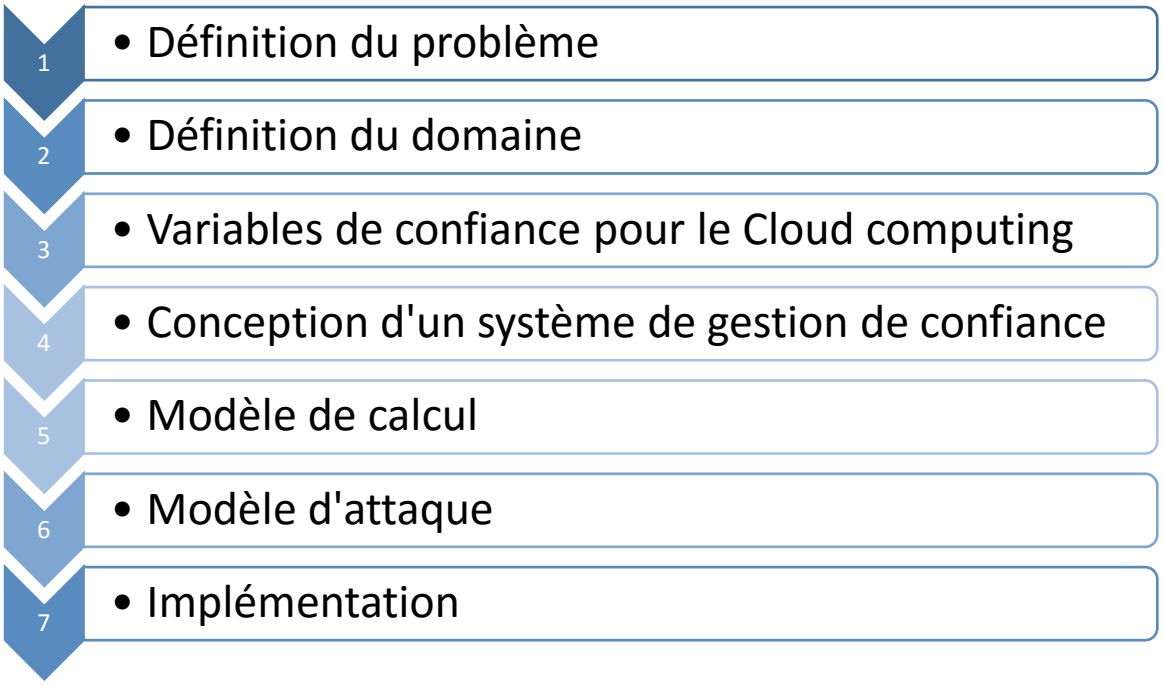

Figure 1.1. Approche de recherche adoptée

1.2. Contributions

Dans la littérature du Cloud Computing actuelle, il n'existe pas de système générique par lequel un consommateur peut prendre une décision fondée sur la confiance par rapport à la sélection d'un fournisseur de services.

Compte tenu de l'évolution rapide du Cloud Computing et les implications commerciales, il est très important d'avoir une telle architecture en place. Les principales questions qui ne sont pas étudiées dans la littérature connexe sont :

- l'absence d'un modèle fiable pour la confiance et la réputation spécifique à une architecture en Cloud ;
- l'absence d'un modèle pour calculer et estimer le coût pour chaque niveau de l'architecture en Cloud ;
- Bien que les systèmes de confiance et de réputation aient été largement proposés et mis en œuvre pour différents types de services en ligne, aucun de ces modèles n'a été proposé pour le Cloud Computing ;
- Aucun système n'offre un service de filtrage des avis d'utilisateurs basé sur les attaques relatives à la gestion de confiance.

Cette thèse propose un système de confiance pour la sélection d'un fournisseur de service dans un environnement Cloud. À cet effet, nos contributions se résument par :

- intégration d'un modèle de calcul au système proposé (Filali & Yagoubi, 2015c). Ce modèle est basé sur deux paramètres contrairement aux modèles existants (Shi & Guo, 2010) ; (Mohammed Alhamad, 2011) ; (Ahmad, Ahmad, Saqib, & Khattak, 2012) (S. Habib & Ries, 2013) ; (Pavlidis, Mouratidis, Kalloniatis, Islam, & Gritzalis, 2013) ; (Qu & Buyya, 2014) ; (Macías & Guitart, 2015) : un paramètre pour le calcul du degré de confiance et un paramètre pour définir le niveau de performance du service offert ;
- intégration d'un modèle d'attaque (Filali & Yagoubi, 2015b) pour filtrer les recommandations des consommateurs malhonnêtes afin d'augmenter la crédibilité des notations. Les feedbacks injustes sont automatiquement ignorés sans connaissance a priori des utilisateurs de façon à ne pas ou peu influencer le degré de confiance final ;
- prise en compte de différentes sources pour le calcul de confiance : confiance directe, recommandations des consommateurs, paramètres de qualité de service, préférences du consommateur.
- proposition d'un système générique (Filali & Yagoubi, 2015a) pour la sélection de fournisseurs fiables dans un environnement de Cloud Computing.

Les travaux réalisés au cours de cette thèse ont fait l'objet de communications et de publications suivantes :

1) F. Z. Filali, B. Yagoubi, Classifying and Filtering Users by Similarity Measures for Trust Management in Cloud Environment. Scalable Computing: Practice and Experience, vol. 16, N3, p. 289–301, 2015
2) F. Z. Filali, B. Yagoubi, Global Trust: A Trust Model for Cloud Service Selection, International Journal of Computer Network and Information Security, vol. 7, N5, p. 41-50, 2015.
3) F. Z. Filali, B. Yagoubi, A General Trust Management Framework for Provider Selection in Cloud Environment, 19th East European Conference on Advances in Databases and Information Systems (ADBIS 2015), LNCS 9282, September 8-11, 2015, Poitiers in France, Springer International Publishing Switzerland, 2015.
4) F. Z. Filali, B. Yagoubi, Modeling a trust system for Cloud Computing. 8éme séminaire National en Informatique Biskra. 20-22 Janvier 2015. Biskra, Algérie.

5) F. Z. Filali, B. Yagoubi, A review of trust models in Cloud Computing. 2éme Journées Doctorales en Informatique de Guelma. 4-5 Novembre 2012. Guelma, Algérie.

1.3. Organisation de la thèse

Le reste de ce manuscrit de thèse est organisé comme suit :

Le second chapitre présente l'état de l'art. La première partie porte sur l'étude du Cloud Computing. Ensuite, nous classifions les différents défis et risques pour son adoption, particulièrement les problèmes de sécurité. La deuxième partie du chapitre fournit une présentation de la littérature sur les modèles de confiance existants et plus spécifiquement sur les modèles pour le Cloud Computing.

Le troisième chapitre décrit en détail notre proposition. Nous explicitons l'architecture du système proposé et nous détaillons les modèles de calcul et d'attaque proposés.

Le quatrième chapitre est consacré à l'implémentation du système proposé et les différentes simulations conduites. Ce chapitre porte aussi sur une étude analytique du système et des modèles proposés et une évaluation des performances pour valider la solution apportée.

Le cinquième chapitre conclut notre thèse. Nous soulignons les principales contributions réalisées au cours de nos travaux, ainsi que les limites du système proposé. Enfin, nous énonçons les principales perspectives et futures directions de recherche.

Chapitre 2 : Etat de l'art

2.1. Introduction

Dans ce chapitre, nous présentons d'abord le Cloud Computing, ses caractéristiques, ses modèles de service et de déploiement. Ensuite, nous abordons la sécurité pour le Cloud Computing, les problèmes les plus cités dans ce type d'environnement et enfin les défis et challenges à traiter. Puis, nous définissons la confiance, ses différents aspects, les modèles de confiance en général et les modèles de confiance dans le Cloud Computing.

2.2. Cloud Computing

2.2.1. Infrastructures informatiques distribuées

Les progrès technologiques en matière d'infrastructure matérielle (puissance de calcul, de stockage, etc…) et de connexions réseau (technologie de haut débit) ont permis l'émergence d'une nouvelle branche appelée l'*informatique distribuée*. L'informatique distribuée permet à des applications sur des machines distinctes et distantes de coopérer pour effectuer des tâches coordonnées (Coulouris, Dollimore, & Kindberg, 2012).

Cette section présente l'évolution de l'informatique distribuée au Cloud Computing.

2.2.1.1.Evolution vers le Cloud Computing

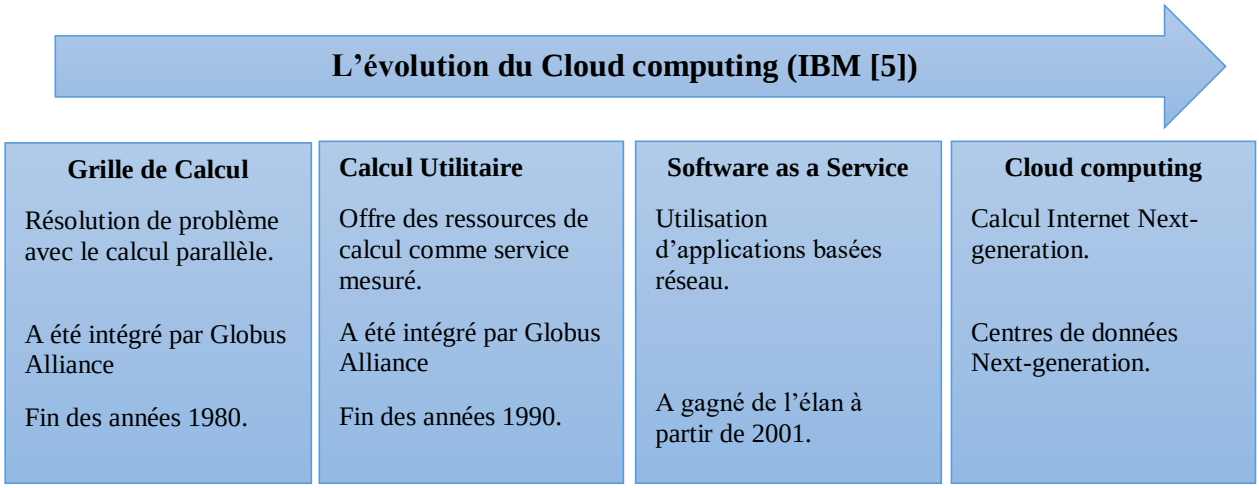

Figure 2.1. Evolution du Cloud Computing (IBM [5])

L'émergence du Cloud Computing trouve son origine dans l'informatique distribuée. Avec l'apparition des grilles de calcul, ce paradigme a connu plusieurs évolutions de

l'informatique utilitaire en passant par les applications web en tant que services pour arriver à la dernière génération qu'est le Cloud Computing, comme l'explicite la figure.2.1.

2.2.1.2.Systèmes distribués

Un système informatique distribué est une collection de machines ou de calculateurs autonomes qui sont connectés à l'aide d'un réseau de communication. Chaque machine exécute des fonctions (séquences de calculs) en utilisant un intergiciel (middleware[1]), qui s'occupe d'activer des fonctions et de coordonner leurs activités de telle sorte qu'un utilisateur perçoive le système comme un système intégré unique.

Il n'y a pas de définition universellement acceptée du terme système distribué, au contraire cela décrit des systèmes d'information avec des architectures et caractéristiques très différentes. Dans la littérature, on peut trouver diverses définitions.

Il a été défini par Tanenbaum (Tanenbaum & Van Steen, 2007) par : « *un ensemble d'ordinateurs indépendants qui apparaît à un utilisateur comme un système unique et cohérent* ».

Et par Lamport (Lamport, 1978) : « *un système réparti est un système qui vous empêche de travailler quand une machine dont vous n'avez jamais entendu parler tombe en panne* ».

Ainsi, un système distribué est un système constitué d'un ensemble d'entités en interaction à travers un canal de communication.

2.2.1.3.Grille

Les origines des grilles sont assez floues, aux alentours des années 70. Certains disent [10] que le précurseur des grilles est la société Apple, plus précisément l'entreprise NeXT avant qu'elle ne se fasse racheter par Apple, s'appuyant sur une idée de Xerox. Le docteur Richard Crandall (NeXT) serait le premier à avoir expérimenté les grilles, grâce à son programme de calcul parallèle distribué baptisé Zilla, le programme utilisait des machines chaînées entre elles pour des traitements mathématiques complexes. D'autres (Abbas, 2004) disent que l'idée serait venue de trois personnes, du docteur en Mathématiques et en Informatique Ian Foster (Directeur du laboratoire National Argonne à Chicago aux États-Unis), de Monsieur Carl Kesselman chercheur en informatique à « University of Southern California » et du Steve Tuecke ingénieur en informatique au Laboratoire National Argonne. Ces trois chercheurs sont surnommés «

[1] Middleware est le terme anglophone d'intergiciel

fathers of the Grid » (les fondateurs de grille) et sont à l'origine de l'une des plus importantes organisations de grilles « *The Globus Alliance* » (Foster & Kesselman, 1997).

Une grille fonctionne comme un réseau de distribution électrique ; celui-ci fournit à chaque utilisateur toutes les ressources dont il a besoin au moyen d'une interface simplifiée, une prise de courant. Toute la complexité du réseau sous-jacent (de la centrale électrique en particulier) est complètement cachée. De plus, l'utilisateur peut faire varier brutalement sa consommation sans démarche préalable. Dans une grille, puissance de calcul et capacités de stockage sont pratiquement illimitées puisque toutes les ressources de la grille peuvent être mobilisées en cas de besoin. Elle permet de mettre sans effort en production intensive une application développée localement et de mieux partager les ressources disponibles (dans les centres de calcul et dans les laboratoires ou bien dans les différents sites d'une entreprise).

La première définition la plus citée d'une grille reflétant ses origines et a été suggérée par Foster et Kesselman (Czajkowski et al., 1998) :

«*Une grille de calcul est une infrastructure matérielle et logicielle qui fournit un accès fiable, constant, omniprésent et peu coûteux pour des capacités de calcul haute performance.* » Le partage de ressources a été étendu par la suite à d'autres domaines.

Selon Foster (Foster, Kesselman, Nick, & Tuecke, 2002), un système de grille est donc un système qui :

- coordonne les ressources qui ne sont pas soumises à un contrôle centralisé ;
- utilise des protocoles et interfaces standards, ouverts, à usage général ;
- délivre des qualités de service non triviales.

Les principales ressources qui peuvent être partagées dans une grille sont :

- calcul / puissance de traitement ;
- stockage de données / systèmes de fichiers en réseau ;
- communications et bande passante ;
- logiciels d'application ;
- outils scientifiques.

Une grille est maintenant considérée par la communauté des chercheurs comme une couche intergicielle permettant le partage fiable, sûr et efficace de ressources et de données entre des entités organisationnelles indépendantes (Schikuta et al., 2009). Les grilles furent aussi adoptées par l'industrie avec différentes interprétations. IBM par exemple décrit les grilles

indirectement en se référant à ses caractéristiques: « *Une grille informatique permet de rassembler un ensemble de serveurs, systèmes de stockage et réseaux dans un seul grand système fournissant ainsi la puissance de multiples systèmes de ressources à un seul utilisateur dans un but précis. Pour un utilisateur, un fichier de données ou une application, le système semble être un seul énorme système informatique virtuel* ». (Kourpas, 2006)

Inspiré par l'omniprésence, la facilité d'utilisation et la fiabilité du réseau électrique, les informaticiens dans les milieux des années 1990 ont commencé à explorer la conception et le développement d'une infrastructure analogue appelée Grille de puissance de calcul. Une grille de calcul permet d'accéder à la puissance de calcul et l'application à tout moment ou lieu, comme nécessaire sans la nécessité de posséder l'infrastructure nécessaire pour produire le service (Buyya, Abramson, & Venugopal, 2005).

2.2.1.4.Informatique Utilitaire

Avec la popularité et l'utilisation croissante des grilles de calcul, les grandes installations de grilles ont rencontré de nouveaux problèmes, telles que les demandes excessives de ressources. Initialement, la gestion des ressources n'assure pas un accès équitable aux ressources dans de nombreux systèmes. Les paramètres traditionnels (débit, temps d'attente, latence) n'ont pas permis de garantir les exigences les plus subtiles des utilisateurs. Il n'y avait pas de réelle flexibilité et souplesse pour les exigences des utilisateurs en matière de ressources, ni de dispositions pour accueillir les utilisateurs avec des travaux urgents.

Dans les environnements informatiques utilitaires, les utilisateurs attribuent une valeur "d'utilité" à leurs jobs, où l'utilité est une évaluation fixe, variable dans le temps définissant différentes contraintes de qualité de service-QoS[2] (délai, priorité, satisfaction). L'évaluation est le montant qu'ils sont prêts à payer à un fournisseur de services pour satisfaire leurs demandes.

Les fournisseurs de services tentent de maximiser leur utilité qui est alors en corrélation avec leur profit. Ils peuvent choisir de donner une plus grande priorité (le bénéfice se fait par unité de ressources) aux jobs des utilisateurs, conduisant à un scénario où les systèmes partagés sont considérés comme un marché, où les utilisateurs se disputent des ressources basées sur l'utilité ou la valeur de leurs jobs. (Bhatti, 2005)

[2] QoS : Quality of Service (Qualité de Service)

2.2.2. Cloud Computing

Le Cloud est une métaphore de l'Internet, provenant de sa représentation commune dans les diagrammes réseau (ou plus généralement des composants qui sont gérés par d'autres) comme nuages.

Le concept sous-jacent remonte à 1960 lorsque John McCarthy (McCarthy, 1960) a estimé que « *le calcul pourrait un jour être organisé comme une entreprise de service public* » (en fait, il partage des caractéristiques avec les bureaux de service qui remontent aux années 1960) et la durée du mandat du Cloud était déjà en usage commercial autour du tournant du 21e siècle. Des solutions de Cloud Computing avaient commencé à apparaître sur le marché, même si la plupart de l'attention à cette époque se portait sur le logiciel comme service. L'année 2007 a vu une activité accrue, comprenant Google, IBM et un certain nombre d'universités se lancèrent dans un projet de recherche à grande échelle de Cloud Computing, à l'époque le terme a commencé à gagner en popularité dans la presse grand public.

2.2.2.1.Définition

Beaucoup de chercheurs dans les domaines industriels et universitaires ont tenté de définir exactement ce qu'est le "***Cloud Computing***" et quelles sont les caractéristiques uniques qu'il présente.

Buyya (Buyya et al., 2009) l'a défini comme suit : « *Le Cloud Computing est un système de calcul parallèle et distribué constitué d'un ensemble d'ordinateurs interconnectés et virtualisés. Ces ordinateurs sont dynamiquement provisionnés et présentés comme des ressources informatiques unifiées, basées sur des accords SLA[3] établis par voie de négociations entre les fournisseurs de services et les consommateurs.* »

D'autres définitions ont été proposées pour définir le Cloud Computing. On retrouve par exemple la définition des analystes de Gartner : « *Un style de calcul pour lequel des moyens informatiques évolutifs et massifs sont fournis en tant que service à plusieurs utilisateurs externes, en utilisant les technologies Internet.* »[6] ou encore la définition exprimée par les analystes d'IDC : « *un développement informatique émergent et un modèle de déploiement, permettant la distribution de produits, services et solutions en temps réel à travers l'Internet.*»[7]

[3] SLA : (Service Level Agreement), est un document qui définit la qualité de service requise entre un fournisseur et un client.

Le National Institute of Standards and Technology (NIST) a donné une définition succincte qui reprend ses principes de base : « *Le Cloud Computing est un modèle pratique, à la demande, pour établir un accès par le réseau à un ensemble partagé de ressources informatiques configurables (réseaux, serveurs, stockages, applications et services) qui peuvent être rapidement mobilisées et mises à disposition en minimisant les efforts de gestion ou les contacts avec le fournisseur de services.* » (Mell & Grance, 2009)

2.2.2.2.Caractéristiques

Le Cloud Computing est un modèle permettant l'accès au réseau à la demande. Les ressources sont partagées et la puissance de calcul est configurable en fonction des besoins. Le client peut bénéficier d'une flexibilité importante avec un effort minimal de gestion.

Il se caractérise par :

1. **Accès en libre-service à la demande aux capacités de calcul**. Ce service sera le plus souvent effectué par le fournisseur de services de façon automatique sans nécessiter d'interaction humaine.
2. **Accès ubiquitaire au réseau.** Les capacités sont disponibles sur le réseau et accessibles par des mécanismes standards, qui favorisent l'accès au service par des clients lourds ou légers via des plates-formes hétérogènes.
3. **Mise en commun des ressources.** Les ressources de calcul sont mises à disposition des clients sur un modèle multi locataires, avec une attribution dynamique des ressources physiques et virtuelles en fonction de la demande. Le client n'a généralement aucun contrôle ou connaissance sur l'emplacement exact des ressources fournies. Toutefois, le client peut imposer de spécifier l'emplacement à un niveau plus haut d'abstraction (par exemple le pays, l'état ou le Data Center).
4. **Evolutivité rapide.** Les capacités proposées peuvent augmenter ou diminuer en fonction des besoins.
5. **Service mesuré en permanence.** Les systèmes contrôlent et optimisent automatiquement l'utilisation des ressources par rapport à une moyenne estimée de consommation du service. L'utilisation des ressources peut être gérée, contrôlée et communiquée, fournissant ainsi une transparence au client et au fournisseur.

2.2.2.3.Modèle de service

Les modèles de services du Cloud Computing sont divisés en trois classes, à savoir: infrastructure en tant que service (IaaS), plate-forme en tant que service (PaaS) et application en tant que service (SaaS). La figure 2.2 représente l'organisation en couches de la pile de Cloud de l'infrastructure physique aux applications. Ces niveaux d'abstraction peuvent également être considérés comme une architecture en couches où les services d'une couche supérieure peuvent être composés des services de la couche sous-jacente.

Modèle de service	Outils d'accès et de gestion	Services offerts
SaaS 	Navigateur Web	Réseaux sociaux, suites bureautiques, CRM, traitement vidéo.
PaaS 	Environnement de développement	Langages de programmation, systèmes, gestionnaire de données structurés.
IaaS 	Gestionnaire d'infrastructure virtuelle	Serveurs de calcul, stockage de données, Pare-feu.

Figure 2.2. Pile de services Cloud

1) Application en tant que Service - Software as a Service (SaaS) :

L'utilisateur a la possibilité d'utiliser les applications du fournisseur de services via le réseau. Ces applications sont accessibles à travers différentes interfaces, clients légers, navigateur Web, etc… Le client ne gère pas et ne contrôle pas l'infrastructure Cloud sous-jacente incluant le réseau, les serveurs, les systèmes d'exploitation, le stockage, mais peut éventuellement bénéficier d'accès à des configurations restreintes, spécifiques à des catégories d'utilisateurs. Ce modèle allège la maintenance des logiciels pour les clients et simplifie le développement et les tests pour les fournisseurs.

Salesforce[4], qui s'appuie sur le modèle SaaS, offre des applications de productivité d'entreprise (CRM[5]) qui résident entièrement sur leurs serveurs, ce qui permet aux clients d'accéder et de personnaliser les applications à la demande.

2) Plateforme en tant que Service - Platform as a Service (PaaS) :

Le consommateur peut déployer sur l'infrastructure Cloud ses propres applications, dans la mesure où le fournisseur supporte le langage de programmation. L'utilisateur gère et ne contrôle pas l'infrastructure Cloud sous-jacente (réseau, serveurs, systèmes d'exploitation, stockage) mais il a le contrôle sur les applications déployées et la possibilité de configurer l'environnement d'hébergement applicatif.

Google App Engine[6] est un exemple de PaaS qui offre un environnement évolutif pour le développement et l'hébergement d'applications Web, écrites dans des langages de programmation spécifiques tels que Python ou Java.

3) Infrastructure en tant que Service - Infrastructure as a Service (IaaS) :

Pour ce modèle, l'utilisateur peut louer des capacités de traitement, de stockage, de réseau et autres ressources de calcul. L'utilisateur ne gère pas et ne contrôle pas l'infrastructure physique Cloud sous-jacente mais il a le contrôle sur les systèmes d'exploitation, le stockage, les applications déployées et la possibilité de sélectionner des composants réseau.

Amazon Web Services[7] offre principalement des services IaaS, qui dans le cas de son service EC2 propose des machines virtuelles avec une pile logicielle personnalisable de façon similaire à un serveur physique ordinaire. Les utilisateurs ont des privilèges pour effectuer de nombreuses activités pour le serveur, telles que : le démarrage et l'arrêt, la personnalisation en installant des logiciels, l'ajout de disques virtuels et la configuration des autorisations d'accès et des règles de pare-feu.

[4] http://www.salesforce.com/
[5] CRM : (Customer Relationship Management - Gestion de la relation client), ensemble des outils et techniques destinés à capter, traiter, analyser les informations relatives aux clients, , dans le but de les fidéliser en leur offrant le meilleur service
[6] https://cloud.google.com/appengine/
[7] https://aws.amazon.com

2.2.2.4.Modèle de déploiement

Bien que le Cloud Computing ait émergé principalement des besoins de services publics, d'autres modèles de déploiement ont été utilisés. Ainsi, on distingue divers modèles de déploiement de services selon les approches des entreprises :

1) Cloud Public

Ce type d'infrastructure est accessible à un large public et appartient à un fournisseur de « services Cloud ».

2) Cloud Privé

L'infrastructure Cloud fonctionne pour une organisation unique. Elle peut être gérée par l'organisation elle-même (Cloud Privé interne) ou par un tiers (Cloud Privé externe). Dans ce dernier cas, l'infrastructure est entièrement dédiée à l'entreprise et accessible via des réseaux sécurisés de type VPN.

3) Cloud Communautaire

L'infrastructure est partagée par plusieurs organisations qui ont des intérêts communs (par exemple, les exigences de sécurité, de conformité, etc…). Tout comme le Cloud Privé, il peut être géré par les organisations elles-mêmes ou par un tiers.

4) Le Cloud Hybride :

L'infrastructure se compose de deux Cloud ou plus (Privé, Communautaire ou Public), qui restent des entités uniques, mais qui sont liées par une technologie normalisée ou propriétaire, permettant la portabilité des données ou des applications.

2.2.3. Concept de base

2.2.3.1.Virtualisation

La virtualisation est une technique pour l'exécution de plusieurs systèmes d'exploitation indépendants de façon virtuelle sur une seule machine physique. Ce terme fut utilisé pour la première fois vers les années 1960 en référence à une machine virtuelle (parfois appelée pseudo machine). (Rittinghouse & Ransome, 2009)

La virtualisation des ressources est au cœur de la plupart des architectures Cloud. Le concept de la virtualisation permet une vue logique abstraite sur les ressources physiques et serveurs, les magasins de données, les réseaux et les logiciels. L'idée de base est de mettre en

commun les ressources physiques et les gérer comme un tout. Les demandes individuelles peuvent ensuite être servies selon les besoins de ces pools de ressources.

2.2.3.2.Architectures orientées services (SOA[8])

Les architectures orientées services et les services Web sont à considérer comme les conditions préalables fondamentales pour le Cloud Computing. Les architectures orientées services (SOA) sont des architectures dont les composants sont mis en œuvre en tant que services indépendants. Elles peuvent être flexibles, liées ensemble et orchestrées. Elles peuvent communiquer via des messages dans une configuration à couplage faible. Avec le Cloud Computing, des infrastructures informatiques virtualisées, des plates-formes et des applications entières sont mises en œuvre en tant que services et mises à disposition des utilisateurs dans des architectures orientées services. (Baun, 2011)

2.2.2.3.Modèle économique

Pour évaluer les services du Cloud à partir d'un point de vue économique, des modèles de coûts sont utilisés. Dans les calculs de coûts, les coûts d'utilisation des services Cloud (par exemple en heures et en unités de serveur) sont évalués par rapport aux coûts des centres de données (ou les coûts des infrastructures informatiques achetées et entretenues par le fournisseur). (Baun, 2011)

2.2.4. Avantages et défis

Le Cloud offre plusieurs avantages comme un déploiement rapide, paiement à l'utilisation, prix faibles, l'évolutivité, mise en place rapide, élasticité rapide, accès au réseau ubiquitaire, protection contre les attaques réseau, reprise après panne à faible coût, solutions de stockage de données, contrôles de sécurité à la demande, la détection en temps réel de système de falsification et rapide reconstitution de services.

2.2.5. Obstacles et possibilités pour le Cloud Computing

Dans une récente recherche, Armbrust (Armbrust et al., 2010) définit une liste de classement d'obstacles critiques à la croissance du Cloud Computing. Les trois premiers affectent l'adoption du Cloud, les cinq suivants affectent la croissance du Cloud et les deux

[8] SOA : Service Oriented Architecture

derniers sont les obstacles politiques et organisationnels. Chaque obstacle est associé à une solution, allant du développement du produit à des projets de recherche :

1. disponibilité et continuité de service ;
2. données propriétaires (Lock-in) ;
3. confidentialité des données et audit ;
4. transfert des données et congestion du réseau ;
5. performance imprévisible ;
6. stockage évolutif ;
7. bogues dans les systèmes distribués à large échelle ;
8. mise à échelle rapide ;
9. licences des logiciels.

2.3. Sécurité pour le Cloud Computing

L'International Data Corporation (IDC) [2] a mené une enquête auprès de 244 responsables informatiques et hommes d'affaires pour évaluer leurs opinions et comprendre l'utilisation des sociétés de services informatiques dans le Cloud. La sécurité a été classée en premier rang comme plus grand défi ou problème du Cloud Computing.

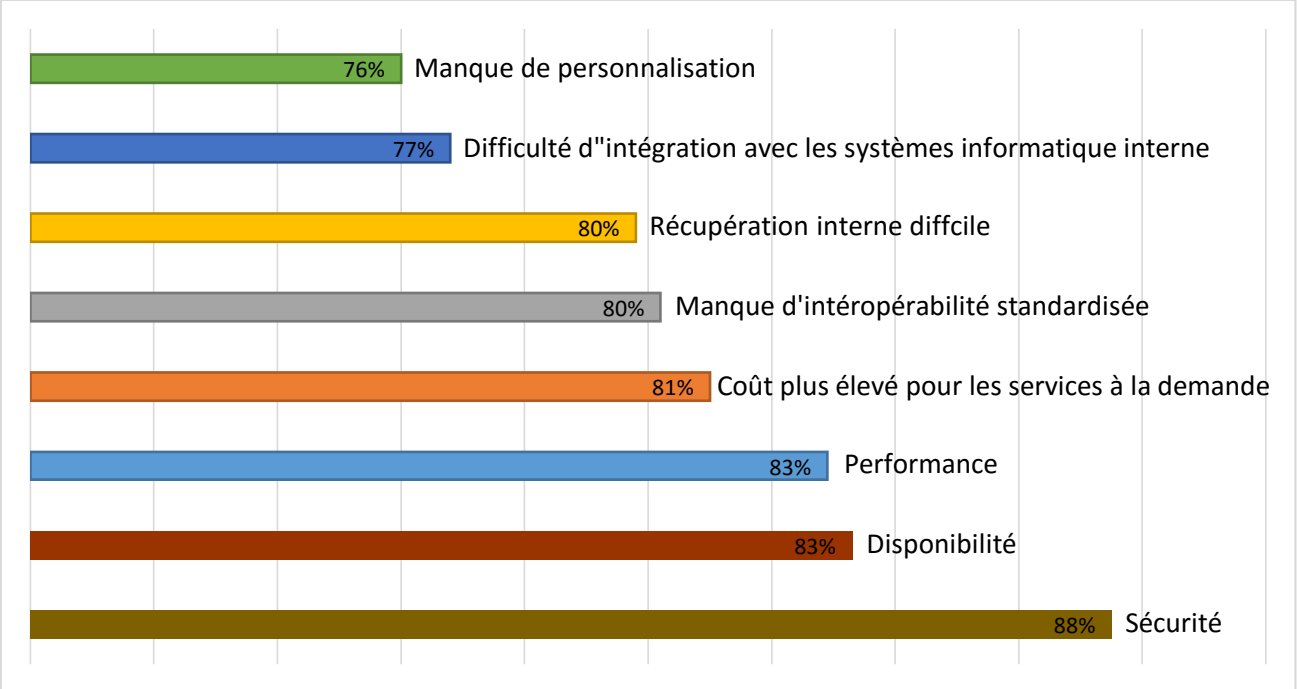

Figure 2.3. Défis du Cloud Computing selon IDC [2]

2.3.1. Défis de sécurité

Les analystes de Gartner [11] ont identifié sept défis de sécurité empêchant les entreprises adoptent le modèle Cloud Computing, énumérés ci-dessous :

a. **accès utilisateurs privilégiés.** Les informations transmises par le client par le biais d'Internet posent un certain degré de risque, en raison de problèmes de propriété des données; les entreprises doivent passer du temps pour connaître leurs fournisseurs et leurs règlements, autant que possible avant d'attribuer certaines applications triviales.
b. **conformité à la réglementation**. S'assurer que le vendeur est d'accord pour avoir des audits externes ainsi que des certifications de sécurité.
c. **localisation des données**. Savoir si le fournisseur offre un contrôle sur la localisation des données.
d. **ségrégation des données**. S'assurer que le chiffrement est possible à tous les niveaux et que les schémas de chiffrement aient été testés et approuvés.
e. **récupération**. Chaque fournisseur doit avoir un protocole de récupération de données pour protéger les données des utilisateurs.
f. **soutient aux enquêtes**. Si un client soupçonne une activité défectueuse du fournisseur, il peut ne pas y avoir beaucoup de moyens juridiques pour poursuivre une enquête.
g. **viabilité à long terme**. Se réfère à la capacité de rétracter un contrat et toutes les données si le fournisseur actuel est racheté par une autre entreprise. [3]

De plus, plusieurs auteurs ont cité différentes problématiques que l'on peut rencontrer dans un environnement Cloud tels que (Subashini & Kavitha, 2011) ; (Shaikh & Mumbai, 2012) ; (Rong, Nguyen, & Jaatun, 2013); (Fernandes, Soares, Gomes, Freire, & Inácio, 2014) ; (Puthal, Sahoo, Mishra, & Swain, 2015). Ces défis peuvent être résumés par :

- sécurité des données ;
- sécurité du réseau ;
- intégrité des données ;
- ségrégation des données ;
- accès aux données ;
- autorisation et authentification ;
- disponibilité des données ;
- sécurité des machines virtuelles.

2.3.2. Mesures de sécurité dans le Cloud Computing:

Diverses mesures de sécurité ont été proposées pour résoudre les problèmes de sécurité et contrer les menaces dans un environnement de Cloud Computing. Ces mesures sont classées de façon générale en :

- **virtualisation.** Chaque locataire peut bénéficier d'un environnement isolé complètement virtuel pour son exécution ;
- **virtual Private Network (VPN).** L'échange de données entre le fournisseur de Cloud et l'utilisateur peut être assuré par l'utilisation de VPN ;
- **identité fédérée (Federated Identity).** C'est la capacité de transporter des données à travers des domaines de sécurité utilisant des allégations et affirmations à partir d'un fournisseur d'identité signé numériquement. Les utilisateurs qui ont déjà été authentifiés dans le réseau de l'organisation devraient être autorisés à utiliser les services en cours d'exécution sur le Cloud. Cela est assuré par un service d'identité fédérée, qui relie ensemble la gestion des identités de l'organisation et le fournisseur de services Cloud ;
- **gestion de politique.** Définit des politiques permettant de décider quel fournisseur choisir en se basant sur des facteurs tels que la fiabilité, la sécurité, etc… ;
- **gestion de Confiance**. Avec l'évolution rapide des nombres de services offerts dans le Cloud Computing, les consommateurs sont confrontés au défi de sélection du meilleur fournisseur et le choix du service le plus approprié dans leur situation.

Chacune de ces solutions implique des techniques diverses pour assurer la sécurité dans un environnement Cloud.

2.3.2.1.la virtualisation:

La virtualisation est le point clé définissant un environnement de Cloud Computing. Dans un environnement multi-locataire, il est nécessaire d'avoir un isolement entre les processus des différentes organisations. Un bug dans l'application ou dans le système d'exploitation peut conduire à un problème. La solution est soit d'allouer des machines physiques distinctes, soit des machines virtuelles distinctes. Bien sûr, la virtualisation dans cette situation devient une solution plus rentable.

En plus de la séparation des processus la virtualisation offre d'autres avantages, tels qu'une rapide élasticité, où les ressources peuvent être ajoutées ou retirées en fonction de la demande

de l'organisation. Un autre avantage de la virtualisation est la portabilité : il est facile de déplacer des machines virtuelles d'une machine physique à une autre.

Les organisations peuvent aussi déployer des solutions de sécurité sur leur espace virtuelle augmentant encore plus le niveau de sécurité. Des solutions comme le pare-feu, la détection et la prévention d'intrusion, la surveillance de l'intégrité, l'inspection des logs, la confiance, etc…

2.3.2.2.Pare-feu:

Un pare-feu est un système conçu pour empêcher l'accès non autorisé à ou à partir d'un réseau privé. Il peut aider en diminuant le domaine d'attaque des serveurs virtuels d'un environnement de Cloud Computing.

Le déploiement de pare-feu sur des machines virtuelles (VM) utilisant des politiques correspondant aux besoins de l'organisation permet l'isolement de la machine virtuelle, le filtrage des données, la ségrégation des données couvrant l'ensemble des protocoles basés sur IP, les types de trames, etc… Les attaques telles que les dénis de service (DoS) peuvent être évitées. Les pare-feu permettent également l'établissement de différentes politiques sur différentes interfaces réseau.

2.3.2.3. Détection et prévention des intrusions (IDS / IPS):

IDS[9] et IPS[10] protègent des vulnérabilités dans les systèmes d'exploitation et les applications jusqu'à ce qu'ils puissent être corrigés et mis-à-jour, pour assurer une protection rapide contre les attaques connues.

Un IDS et IPS peuvent détecter les nouvelles vulnérabilités découvertes dans les applications et le système d'exploitation dans la VM, ce qui fournit une protection contre les tentatives pour compromettre les machines virtuelles. Il existe des IDS et IPS basés sur des techniques d'intelligence artificielle, permettant la découverte de nouvelles vulnérabilités dynamiquement.

2.3.2.4.Surveillance de l'intégrité:

Les fichiers système critiques (fichiers, répertoires, clés de registre et les valeurs, etc…) peuvent être surveillés pour détecter les modifications malveillantes et inattendues qui

[9] IDS: Intrusion Detection System.
[10] IPS: Intrusion Prevention System.

pourraient signaler un compromis des ressources. Le logiciel de surveillance de l'intégrité doit être appliqué au niveau de la machine virtuelle. Une solution de surveillance de l'intégrité devrait permettre:

- une détection à la demande ou programmée ;
- un contrôle complet de propriété de fichier, y compris les attributs ;
- une surveillance au niveau de l'annuaire ;
- une surveillance flexible et pratique par le biais d'inclusion/exclusion ;
- des rapports d'audit.

2.3.2.5.Inspection des journaux:

L'inspection de journaux recueille et analyse les journaux de log du système d'exploitation et des applications pour une analyse de sécurité. Des règles sont définies dans l'inspection des logs permettant l'extraction efficace d'événements liés à la sécurité. Ces journaux peuvent être envoyés à un système de sécurité autonome ou à un système d'information de sécurité et de gestion d'événement (SIEM[11]) ou à un serveur de journalisation centralisée pour l'analyse. Le logiciel d'inspection de logs sur les ressources Cloud permet de détecter tout comportement suspect.

2.3.2.6.Confiance

Dans le cadre du paradigme du Cloud Computing, une organisation renonce à un contrôle direct sur de nombreux aspects de la sécurité et, ce faisant, confère un niveau sans précédent de confiance au fournisseur de services. Divers défis se rapportant à la confiance peuvent être considérés :

- **Accès interne**. La menace pour la sécurité interne est un problème bien connu pour la plupart des organisations et s'applique aussi aux services de Cloud Computing sous-traitant.

  Le déplacement de données et d'applications dans un environnement de Cloud Computing externe élargit le risque pour la sécurité interne, non seulement au personnel du prestataire de services mais aussi potentiellement aux autres clients utilisant le service.

[11] Security Information Management System

- **Services de composites**. Les services Cloud peuvent être composés par d'autres services. Les fournisseurs de services qui sous-traitent certains services à partir d'un tiers fournisseur peuvent rencontrer certaines difficultés comme la portée du contrôle sur la tierce partie, les responsabilités en cause et les remèdes et recours disponibles en cas de problème. La confiance est souvent non transitive, exigeant que les arrangements d'un tiers soient divulgués avant de parvenir à un accord avec le prestataire de services et que les modalités de ces ententes soient maintenues tout au long de l'accord.
- **Visibilité**. L'utilisation de services Cloud laisse le contrôle au fournisseur pour sécuriser les systèmes sur lesquels les données et les applications fonctionnent. Pour éviter de créer des lacunes dans la sécurité, des contrôles de gestion, de procédures et de techniques doivent être appliqués proportionnellement à celles qui sont utilisés pour les systèmes internes de l'organisation. C'est une tâche colossale, puisque les mesures pour comparer la sécurité de deux systèmes informatiques demeurent un domaine de recherche continuel. En outre, le suivi du réseau et du système par l'utilisateur est généralement en dehors du champ d'application de la plupart des services, ce qui limite la visibilité et les moyens de vérification des opérations directement.
- **Gestion des risques**. Avec les services Cloud, certains sous-systèmes ou composants de sous-système sont en dehors du contrôle direct de l'organisation qui détient l'information et autorise l'utilisation du système. Certains se sentent plus à l'aise quand ils ont plus de contrôle sur les processus et les équipements concernés. Au minimum, un degré élevé de contrôle offre la possibilité de considérer d'autres solutions, établir des priorités et agir de façon décisive dans l'intérêt de l'organisation lorsqu'ils sont confrontés à un incident. Lors du choix entre une solution en interne et une mise en œuvre dans le Cloud, les risques associés doivent être évalués en détail. Évaluer et gérer les risques dans les systèmes qui utilisent les services de Cloud peut être un défi. Idéalement, le niveau de confiance est basé sur le taux de contrôle que l'organisation a sur le service externe, en matière d'utilisation de sécurité nécessaires à la protection du service et la preuve de l'efficacité des contrôles. Toutefois, la vérification du fonctionnement correct d'un sous-système et l'efficacité des contrôles de sécurité peut ne pas être réalisable, ainsi le niveau de confiance doit se baser sur d'autres facteurs.

2.4. Confiance

La confiance est liée à tous les défis auxquels fait face l'adoption du Cloud Computing. Elle est grandement influencée par la sécurité de l'information et systèmes. John Chambers

président et PDG de CISCO[12] avait dit que « *le Cloud Computing est un cauchemar de sécurité et ne peut pas être traité de manière traditionnelle* » [8]. La complexité du Cloud Computing rend la problématique de sécurité d'une importance primordiale pour les consommateurs potentiels et les fournisseurs de services. Ces différents points soulèvent la problématique de confiance dans l'utilisation des services de Cloud Computing.

2.4.1. Définitions

La confiance est un concept utilisé dans de nombreuses disciplines. En sciences sociales, l'accent est mis sur l'impact de confiance pour la prise de décision. En informatique, on se concentre sur la conception d'outils pour l'assistance des utilisateurs dans diverses tâches. Souvent, ces outils fonctionnent sur un modèle donné de confiance et fournissent des méthodes pour mesurer la confiance dans un contexte d'application spécifique. Avant d'entrer dans les détails d'une telle notion ou de discuter des différences entre les modèles, nous allons d'abord définir la confiance et les concepts qui s'articulent autour de ce terme.

2.4.1.1.Confiance

Bien que Luhman ait présenté la confiance comme un fait basique de la vie humaine (Luhmann, 1979), la confiance est un terme complexe qui ne connaît pas de définition commune. Cette définition dépend du domaine et du courant de pensée des auteurs. Certains auteurs placent le risque au cœur de la confiance (Deutsch, 1962) tandis que pour d'autres la situation est très importante (Mills, 1983) (Marsh, 1994). D'autres définissent la confiance comme la prévision de comportements. En effet, tout comme (Gambetta, 2000), Rempel voit la confiance comme une probabilité subjective sur de futurs événements, tandis que Stztomka voit la confiance comme un pari sur les futures actions des autres (Rempel & Holmes, 1986).

La littérature est donc très riche sur le concept de la confiance, cependant, certains auteurs se sont distingués dans leurs recherches en publiant des travaux signifiant sur la confiance dans leur domaine. (Deutsch, 1962), se base sur une perception individuelle de la confiance en parallèle à (Luhmann, 1979) qui place ce concept dans une dimension sociale tandis que (Gambetta, 2000) voit la confiance comme la prévision d'un comportement.

[12] http://www.cisco.com/

Ces travaux introduisent la discussion sur la confiance pour ensuite éclaircir les propriétés générales communes à de nombreuses définitions. Ces propriétés générales sont utilisées dans les systèmes à base de confiance afin de mesurer le degré de confiance entre entités.

(Deutsch, 1962) définit la confiance comme une décision par rapport à une perception individuelle des coûts et bénéfices dont dépend cette décision. Lors d'une décision de confiance l'individu est confronté à un chemin ambigu dont les issues dépendantes d'une tierce personne peuvent être perçues positives ou négatives. L'individu perçoit les issues négatives plus importantes que les issues positives. En choisissant de faire confiance l'individu suppose que l'issue positive se produira plutôt que l'issue négative. L'individu est donc confiant aux capacités et intentions de la tierce personne dont dépend l'occurrence de l'issue positive.

Selon (Luhmann, 1979), la confiance permettrait de faire diminuer la complexité de l'environnement, c'est pourquoi il la voit comme un « *fait basique de la vie humaine* ».

Tout comme d'autres auteurs (Deutsch, 1962; Marsh, 1994), Luhman affirme que la confiance se base sur une notion de risque (Luhmann, 1979). En effet, c'est l'acceptation du risque de me faire tuer, qui permet de diminuer la complexité sociale. Selon lui, c'est seulement de par la présence de la possibilité d'un aboutissement négatif que la confiance est présente dans une situation. Luhman oppose l'acte de faire confiance (trust) et le sentiment de faire confiance (confidence), dans une situation de confidence l'alternative n'est pas possible donc il y a absence de risque.

Luhman décrit la confiance comme un phénomène sociale où des individus interagissent dans la société (Luhmann, 1979). Ce phénomène social se traduit par le fait que celui qui fait confiance doit faire face à la liberté de celui à qui il fait confiance. En effet, celui à qui la confiance a été donnée choisit d'effectuer correctement ou non ce qu'on lui demande. Cependant, lorsque quelqu'un choisit de faire confiance en une autre personne, il possède certaines attentes envers cette personne malgré les libertés qu'elle peut avoir. Selon Barber qui se place aussi dans une dimension sociale de la croyance, l'individu qui fait confiance espère que l'autre remplira des compétences techniques ainsi que des responsabilités et des obligations morales (Mills, 1983).

Afin de définir le terme de confiance, Gambetta introduit la notion de probabilité subjective et de circonstances. « *La confiance (ou symétriquement la défiance) est un niveau particulier de la probabilité subjective avec laquelle un agent accomplira une action spécifique, à la fois avant que nous ne puissions suivre chaque action (ou indépendamment de sa capacité*

de même pouvoir la tracer) et aussi dans un contexte dans lequel cela affecte notre propre action. » (Gambetta, 2000)

Dans cette définition, Gambetta fait seulement référence à la confiance entre deux agents et non par exemple à la confiance entre un agent et son environnement. De plus, les actions des agents n'ont pas de conséquence sur la confiance, car elle est évaluée à l'instant présent sans avoir de retour sur les actions des agents. Pour Gambetta, ce niveau particulier, en plus d'être dépendant de l'agent qui l'évalue est aussi dépendant des circonstances. En effet, dans deux circonstances différentes, un agent pourra évaluer différemment la confiance qu'il a envers un autre agent. La confiance n'est donc pas généralisable, elle est spécifique au contexte.

(Mcknight & Chervany, 1996) reprennent les mêmes paramètres que Gambetta dans leur définition en ajoutant le fait que la confiance est présente malgré un aboutissement négatif possible.

Dans le domaine informatique, (J. A. Golbeck, 2005) se rapproche de la définition de Gambetta où la confiance est un engagement à croire au bon déroulement des futures actions d'une autre entité. Selon (Grandison & Sloman, 2003), l'acte de faire confiance se réalise dans un contexte spécifique et se définit par une croyance quantifiée quant aux habilités de l'entité qui est crue. Cette quantification peut être une échelle de valeurs ou une simple classification.

Ces auteurs ont tous des définitions différentes sur la confiance cependant trois éléments sont présents dans chaque définition :

- une personne ou entité faisant confiance (***trustor***) ;
- une cible, une personne ou une entité à qui on fait confiance (***trustee***) ;
- une situation.

De plus, divers autres auteurs ont étudié dans des états de l'art la confiance (Sabater & Sierra, 2002) ; (Audun Josang & Lo Presti, 2004) ; (Viljanen, 2005) ; (Artz & Gil, 2007) ; (A Jøsang, Ismail, & Boyd, 2007) et ont donné des définitions et généralisaient plusieurs concepts.

2.4.1.2.Réputation

Le dictionnaire anglais d'Oxford [12] définit la réputation comme « l'estimation commune ou générale d'une personne avec le respect de la personnalité ou d'autres qualités ».

La réputation peut être liée à un groupe ou à un individu. Une réputation de groupe peut être représentée comme moyenne de toutes les réputations individuelles de ses membres ou comme la moyenne de perception externe du groupe.

La confiance et la réputation sont des termes très proches et extrêmement liées. (Mui, Mohtashemi, & Halberstadt, 2002) voient la réputation comme la perception des intentions et des normes d'une autre entité tandis qu'ils définissent la confiance comme une prévision subjective d'une entité sur le comportement d'une autre entité.

La réputation est la perception d'une entité en fonction des expériences passées, mais ne suppose rien sur les comportements futurs de cette entité tandis que la confiance fait une prévision sur ces comportements futurs en prenant en compte par exemple les expériences passées. En outre, la réputation sert à évaluer la confiance.

2.4.1.3.Risque

Luhman a introduit la notion de risque dans sa définition de la confiance (Luhmann, 1979). Dans la relation entre deux personnes, cette notion de risque vient renforcer le lien de confiance. Plusieurs auteurs (Audun Josang & Lo Presti, 2004; Song, Hwang, & Kwok, 2006) ont proposé des solutions pour mesurer le risque encouru dans une situation de confiance.

Marsh a représenté le risque en un rapport entre les coûts et les bénéfices (Marsh, 1994) :

$$Risque = \frac{Coûts}{Bénéfices} \quad (2.1)$$

(Boyle & Bonacich, 1970) ont utilisé la théorie des jeux avec le dilemme du prisonnier pour une estimation du risque. Toutefois, leur proposition n'est possible que lorsque les gains et les pertes sont connus. En conséquence, elle est difficilement applicable lorsque la situation est incertaine.

2.4.2. Aspects de base

La confiance étant une notion générique comprenant plusieurs définitions et regroupant divers domaines : croyance en psychologie, valeur numérique en informatique. Ainsi, elle compte plusieurs aspects, qui sont discutés dans ce qui suit :

2.4.2.1.Caractéristiques de confiance

Les caractéristiques communes les plus utilisées pour la confiance sont :

1) bienveillance (Rempel, Holmes, & Zanna, 1985), représente la motivation pour agir dans l'intérêt d'une autre personne ;
2) intégrité (Cummings & Bromiley, 1996), conclure un accord avec bienveillance ;
3) compétence, aptitudes ou capacités nécessaires pour faire ce que nous devons faire ;
4) prévisibilité, pouvoir prévoir les situations futures, à partir des actions assez conformes d'autres entités.

2.4.2.2.Principes de confiance

Pour garantir un niveau maximal de confiance certains principes doivent être suivis :

1) la transitivité de confiance, si l'entité A fait confiance à l'entité B et l'entité B fait confiance à l'entité C, alors on peut arriver à la conclusion que A peut faire confiance à l'entité C en se référant à la confiance de l'entité B ;
2) la confiance est une fonction de perception de risque, elle représente une croyance en une personne pour ses actions correctes. Ainsi, la confiance doit aussi évaluer l'incertitude que l'autre partie agit correctement et y intégrer les risques associés ;
3) la confiance est déterminée par le temps, elle est construite au fil du temps en se basant sur les expériences passées ;
4) la confiance peut être mesurée, elle est mesurable par une valeur numérique, généralement dans l'intervalle [0 – 1] ;
5) les outils formels et sociaux sont nécessaires pour l'évolution de la confiance, la confiance peut être modélisée selon divers modèles formels.

2.4.2.3.Facteurs influant sur la confiance

La valeur de la confiance calculée est influencée par quelques facteurs (Divakarla & Sekaran, 2015) :

1) la confiance joue un rôle dans les environnements incertains et à risque ;
2) la confiance est l'élément de base pour la prise de décision ;
3) la confiance est construite par les expériences et connaissances antérieures ;
4) la confiance est une notion subjective basée sur les opinions et valeurs individuelles ;
5) la confiance est dynamique et de nouvelles expériences et connaissances remplacent les anciennes valeurs qui deviennent obsolètes ;
6) la confiance dépend du contexte ; par exemple, une personne A peut faire confiance à une personne B en tant que médecin, mais ne lui fait pas confiance pour réparer son PC.

Ainsi, dans le contexte de médecine, on peut faire confiance à la personne B mais on ne peut plus lui faire confiance dans le contexte de l'informatique ;

7) la confiance a de multiples facettes. Dans le même contexte, il est nécessaire de développer différents aspects de confiance pour un service. Un utilisateur peut évaluer un service selon différentes QoS telles que le temps de réponse, fiabilité, temps d'exécution, etc… La confiance globale correspondra à une combinaison de toutes les facettes de confiance.

2.4.2.4.Types de confiance

Selon la classification de (Grandison & Sloman, 2000), les différentes classes de confiance sont :

1) **Confiance de provision**. Décrit la confiance d'une tierce partie dans un service ou un fournisseur de services. Le projet de la Liberty Alliance utilise la notion de confiance commerciale (Business Trust [9]) pour décrire la confiance mutuelle entre les entreprises issues des contrats SLA qui régissent les interactions entre eux. Cela peut être interprété comme la confiance de provision ; par exemple, quand un contrat spécifie les exigences de qualité pour la prestation des services, alors cette confiance commerciale fournit une confiance dans ce type de classe.
2) **Confiance d'accès**. Décrit la confiance avec le principe d'accès aux ressources propriétaires par ou sous la responsabilité d'une tierce partie. Cela est réalisé avec le principe du contrôle d'accès qui représente un élément capital dans la sécurité. Un exemple de système d'accès basé confiance est (Grandison & Sloman, 2000).
3) **Confiance de délégation**. Décrit la confiance en une entité déléguée qui agit et prend des décisions au nom d'une autre entité.
4) **Confiance d'identité**. Décrit la croyance qu'une identité d'une entité est celle qu'elle prétend être. Les systèmes de confiance qui se basent sur la confiance d'identité utilisent des schémas d'authentification tels que X.509 et PGP (Zimmermann & Zimmermann, 1995).
5) **Confiance de contexte**. Décrit dans quelle mesure, une entité croit que le système soutient la transaction et fournit une solution de secours dans le cas où il y'aurait un problème. Les facteurs pour ce type de confiance peuvent être par exemple des infrastructures critiques, les assurances, les systèmes juridiques.

2.4.3. Représentation de la confiance

Cette approche correspond à la façon de représenter et d'évaluer la confiance. De nombreuses propositions afin de représenter la confiance ont été faites dans la littérature. Selon, (Alfarez Abdul-Rahman & Hailes, 1998), « *les valeurs de confiance sont utilisées pour représenter les différents niveaux de confiance qu'une entité peut avoir envers une autre entité*». Cette définition est la base de nombreux systèmes, mais les différents niveaux peuvent être représentés de différentes façons.

Ils peuvent correspondre à :

- une valeur continue dans un ensemble donné (Marsh, 1994) (Audun Jøsang & Pope, 2005) ;
- une valeur discrète (J. A Golbeck, 2005) ;
- des valeurs floues (Sabater & Sierra, 2002) ;
- des étiquettes discrètes (A. Abdul-Rahman & Hailes, 2000).

Selon Marsh (Marsh, 1994), cette valeur n'a pas d'unité. Il existe seulement un seuil déterminant si l'entité peut être digne de confiance ou non. (Alfarez Abdul-Rahman & Hailes, 1998) proposent aussi de ne pas représenter la confiance comme une valeur absolue mais comme une relation entre entités.

2.4.4. Systèmes de confiance

Dans la littérature, il existe principalement deux types de systèmes de confiance :

- les systèmes de confiance à base de politique ;
- les systèmes de confiance à base de réputation.

Le premier est fondé sur le principe de compétences et d'autorisations requises pour effectuer une action, tandis que le deuxième utilise la réputation d'une entité permettant d'accorder ou non la confiance envers cette entité.

2.4.4.1.Systèmes de confiance à base de politique

Ces systèmes utilisent des politiques pour définir si une entité a l'autorisation ou non d'accéder. Ces systèmes sont basés sur la définition de gestion de la confiance (trust management).

Selon (M Blaze, Ioannidis, & Keromytis, 2003) « *la gestion de la confiance est une approche unifiée pour spécifier et interpréter des politiques, des qualifications et des relations permettant d'autoriser ou non des actions* ». Les systèmes se basant sur cette définition de gestion de la confiance gèrent les accès et les actions dans le système grâce à des autorisations. Ces autorisations peuvent se faire sur les qualifications. Dans les systèmes de gestion de la confiance, la confiance a une valeur binaire autorisant ou non une entité d'effectuer une action. Les systèmes KeyNote (Matt Blaze, Feigenbaum, & Keromytis, 1999) et PolicyMaker (M Blaze, Feigenbaum, & Lacy, 1996) reposent sur ce type de systèmes.

2.4.4.2.Système de confiance à base de réputation

Les systèmes de confiance à base de réputation utilisent les interactions ou expériences directes, entre entités ou /et l'expérience des autres pour le choix de faire confiance à une autre entité. La réputation est vue comme un facteur guidant (il peut être unique ou associé à d'autres) le processus de décision de confiance. Plus spécifiquement, la réputation sert à évaluer la confiance.

Les définitions des deux termes sont très proches et dans de nombreux systèmes, seule la réputation est le support de l'évaluation de la valeur de la confiance. La confiance liée ici à la réputation concerne la catégorie « trust Personnel » de (Mcknight & Chervany, 1996) où une personne croit une autre personne dans une situation spécifique.

Dans les systèmes de confiance à base de réputation, il existe des processus pour collecter, réunir ou répandre la réputation dans le système. Selon (Resnick, Kuwabara, Zeckhauser, & Friedman, 2000), ces systèmes possèdent trois propriétés :

- après chaque interaction, il y a l'espérance d'une prochaine interaction ;
- les interactions individuelles sont capturées et distribuées (à une entité centrale ou aux autres entités) ;
- les interactions sont le guide des décisions sur les futures interactions entre entités.

Ces systèmes reposent aussi sur des mécanismes computationnels pour calculer cette réputation.

2.4.5. Modèles computationnels de confiance

Ces dernières années, plusieurs modèles de confiance se basant sur diverses considérations et perspectives pour traiter le problème de sélection de service ont été

proposés. Dans cette section, nous donnons une classification, détaillons et analysons les modèles de confiance qui ont été proposés dans la littérature.

Les systèmes reposant sur la confiance doivent, comme nous l'avons remarqué ci-dessus, posséder une méthode de calcul du degré de confiance. En effet, le calcul de la confiance doit permettre de prendre une décision ; par exemple, un degré élevé de confiance envers une entité permet de juger cette entité fiable et de prendre la décision de lui faire confiance. De nombreux auteurs ont proposé des modèles pour représenter et calculer la confiance dans les systèmes. Les modèles sont classés dans plusieurs catégories : modèles bayésiens, modèles de valeurs discrètes, modèles basés sur la croyance, modèles flous et modèles de flux. Les modèles les plus connus et représentatifs de chaque catégorie sont relatés dans ce qui suit :

2.4.5.1.Modèle de Marsh

Dans son modèle (Marsh, 1994), Marsh se place dans le courant des systèmes multiagents, mais ne traite de la confiance qu'entre deux agents. Pour se faire, il définit trois types de confiance :

- La confiance basique constitue la disposition générale d'un agent x à faire confiance

$$T_x \in [-1, 1[$$

- La confiance générale est la confiance d'un agent x envers un autre agent y sans prendre en compte une situation spécifique. Ce t te connaissance implique la connaissance de l'autre agent notée $K_x(y)$. Marsh accepte la méfiance complète correspondant à −1 mais rejette la confiance complète 1 :

$$T_x(y) \in [-1, 1[$$

- La confiance situationnelle est la confiance d'un agent x envers un autre agent y dans une situation donnée α :

$$T_x(y, \propto) \in [-1, 1[$$

Pour la confiance situationnelle, Marsh utilise la définition de [Rempel and Holmes, 1986], où il affirme que la confiance se trouve dans les gens et dans des situations spécifiques.

Marsh propose une méthode pour calculer la confiance situationnelle qui prend en compte la confiance générale qu'un agent a envers un autre agent ainsi que l'importance et l'utilité de la situation :

$$T_x(y, \propto) = U_x(\propto) * I_x(\propto) * \widehat{T_x(y)} \quad (2.2)$$

Avec :

- $U_x(\propto)$: l'utilité de la situation α pour *x*, c'est-à-dire ce que va lui apporter cette situation,
- $I_x(\propto)$: l'importance de la situation α pour *x*, c'est-à-dire est-ce qu'il y a beaucoup de chance pour que l'entité obtienne des bénéfices ?
- $\widehat{T_x(y)}$: la confiance générale qui prend en compte toutes les situations possibles, correspondant à chaque $T_x(y, \propto)$. Différentes manières de calculer $\widehat{T_x(y)}$ sont proposées. En fonction de :
  - si l'entité est optimiste, il le calculera en fonction du plus élevé de tous les $T_x(y, \propto)$;
  - si l'agent est pessimiste, il le calculera en fonction du moins élevé de tous les $T_x(y, \propto)$;
  - si l'agent est rationnel, il le calculera une moyenne des $T_x(y, \propto)$.

Dans ce modèle, une entité peut coopérer avec un autre si sa confiance situationnelle est supérieure à un seuil de coopération. Selon Marsh, « Le seuil de coopération est une mesure subjective, tempéré par des croyances objectives. » Pour calculer le seuil de coopération, Marsh intègre les risques perçus de la situation, les compétences perçues de l'autre agent dans la situation et la réciprocité :

$$Seui\_Coop_x(\propto) = \frac{Risque\_perçu(\propto)}{Compétanc\_Perçue_x(y, \propto) * \widehat{T_x(y)}} * I_x(\propto) * Pourcentage\_Réduction(y, \propto) \quad (2.3)$$

Dans (Marsh, 1994), Marsh explique comment calculer $Risque\_perçu(\propto)$, $Compétanc\_Perçue_x(y, \propto)$ et $Pourcentage\_Réduction(y, \propto)$.

Les travaux de Marsh sont encore aujourd'hui une référence dans les modèles sur la confiance. En effet, il a réuni dans son modèle de nombreux concepts introduits par différents auteurs : le risque, les compétences, l'utilité, l'importance, la situation, la retroprocité. De plus, il a étudié le facteur de la mémoire et comment calculer la confiance entre une autre entité en fonction de la confiance qu'il attribue grâce à un processus itératif.

2.4.5.2.Modèles basés sur les réseaux bayésiens

Selon (Neapolitan, 1988), un réseau bayésien est un graphe direct acyclique B = (X,A), où X est un ensemble de variables et A est un ensemble d'arcs entre ces nœuds. Chaque variable

de X est un état logique possible du monde. Par exemple, VRAI ou FAUX ou encore FIABLE ou NON FIABLE. S'il y a un lien entre deux variables appartenant à X ; par exemple, si x, y ϵ X et x → y ϵ A alors la valeur de y dépend de x. Certains modèles de confiance se basent sur cette théorie en percevant la confiance d'une entité envers une autre entité comme une probabilité qu'un événement se déroule.

1) Modèle de Jøsang : béta réputation

Le modèle de confiance de Jøsang nommée Subjective logic (Audun Jøsang & Ismail, 2002) repose sur la théorie de probabilité bayésienne ainsi que la théorie de la croyance qui sera présentée ultérieurement. L'approche bayésienne repose sur la bêta distribution permettant d'exprimer la probabilité incertaine qu'une future interaction ou expérience sera positive. En d'autres termes, la bêta distribution représente le fait qu'un résultat futur positif x est quelque chose d'incertain et qu'il peut être estimé. La bêta distribution est utilisée car les valeurs qu'elle prend sont comprises entre 0 et 1.

La bêta distribution est représentée par une fonction de densité de probabilité (PDF) f de la probabilité de la variable p. Cette fonction prend deux paramètres α et β en entrée et est décrite comme suit :

$$f(p| \propto \beta) = \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)} p^{\propto -1}(1-p)^{\beta-1} \quad (2.4)$$

Avec $0 \leq p \leq 1$, $\alpha > 0$, $\beta > 0$, Γ la fonction Gamma et :

$\alpha = r + 1$

$\beta = s + 1$

où r est le nombre d'interactions positives et s le nombre d'interactions négatives.

La probabilité est donc calculée en fonction d'événements précédemment collectés et la représentation de f dans un graphe permet de visualiser la probabilité incertaine que l'événement x se produise dans la prochaine interaction future. Jøsang modélise la confiance par la probabilité subjective qu'une future expérience sera positive. Cette modélisation correspond à la valeur de probabilité d'espérance de la bêta distribution :

$$E(p) = \frac{\propto}{(\propto+\beta)} \quad (2.5)$$

Si, par exemple, on étudie l'événement positif x et que p = 0.8, on peut dire que la probabilité que l'événement x se produise est incertain, mais que la probabilité la plus plausible que x se produise est de 0.8.

Jøsang se base sur ces définitions pour définir une fonction de réputation correspondant à la bêta distribution :

$$\vartheta(p|r_T^X s_T^X) = \frac{\Gamma(r_T^X + s_T^X + 2)}{\Gamma(r_T^X + 1)\Gamma(s_T^X + 1)} p^{r_T^X - 1}(1-p)^{s_T^X} \quad (2.6)$$

Avec $0 \leq p \leq 1$, $r_T^X > 0$, $s_T^X > 0$ et Γ la fonction Gamma.

ϑ est appelé la réputation de T par X. r_T^X représente les expériences positives de l'agent X avec la cible T, tandis que s_T^X représente les expériences négatives de X avec T. Le couple (r_T^X, s_T^X) représente les paramètres de réputation de T par X.

D'après l'équation 2.2 Jøsang modélise la valeur de probabilité d'espérance de la fonction de réputation :

$$E\left(\vartheta(p|r_T^X s_T^X)\right) = \frac{r_T^X + 1}{r_T^X + s_T^X + 2} \quad (2.7)$$

La fonction de réputation est bien subjective puisqu'elle se base sur les expériences passées de X envers T.

Enfin, Jøsang modélise un score de réputation :

$$Rep(r_T^X s_T^X) = E\left(\vartheta(p|r_T^X s_T^X)\right) - 0.5) * 2 = \frac{r_T^X - s_T^X}{r_T^X + s_T^X + 2} \quad (2.8)$$

Ce score de réputation est une mesure de la réputation qui exprime comment un agent est évalué pour ses futures interactions.

De plus, Jøsang introduit deux opérateurs :

- consensus, qui combine les réputations de deux agents à propos d'une même cible ;
- actualisation, qui utilise la transitivité de la croyance. La réputation est calculée en prenant en compte la réputation que l'agent A a envers l'agent B et la réputation que l'agent B a envers la cible T.

2) Autres modèles bayésiens

(Mui et al., 2002) ont aussi réalisé un modèle reposant sur la béta distribution où la décision de coopération entre un agent et un autre dépend de la réputation basée sur le résultat des rencontres précédentes entre ces deux agents.

(Zeng, Alhossaini, Ding, Fikes, & Mcguinness, 2006) ont proposé un système de confiance pour l'encyclopédie en ligne de Wikipédia[13]. Ce système repose sur les réseaux dynamiques bayésiens pour représenter la fiabilité d'un article. Un nœud du réseau correspond à une version i d'un article écrit par l'auteur j, au contenu ajouté par rapport à la version i−1 et au contenu supprimé par rapport à la version i−1. Ce système satisfait la propriété de Markov où l'état de l'article i dépend de l'état de l'article i − 1. La fiabilité d'une version ϵ [0, 1] d'un article dépend alors de la version précédente, de l'auteur de la dernière version, de la quantité de textes ajoutés et de textes supprimés. Si un article obtient la note de 0.7 grâce à ces facteurs, cette note sera plutôt entre 0.65 et 0.75 car cette note peut dépendre d'autres facteurs, par exemple l'intérêt de l'auteur pour le domaine. Afin de représenter l'incertain, la fiabilité est représentée grâce à la bêta distribution.

2.4.5.3.Modèles basés sur les valeurs discrètes

1) Modèle d'Abdul-Rahman

Dans leur modèle, (A. Abdul-Rahman & Hailes, 2000) rejettent le fait de définir le niveau de confiance comme une probabilité, car pour eux une probabilité n'a de sens que pour des événements similaires répétés. De même, ils rejettent la transitivité de la confiance. Dans ce modèle, la confiance est définie comme une mesure subjective (croyance) de l'expérience personnelle dans un contexte particulier et cette mesure subjective est propagée dans le système sous le terme de réputation.

La croyance d'un agent envers un autre agent dans un contexte spécifique est représentée par quatre valeurs : très fiable, fiable, douteux et très douteux. L'expérience directe d'un agent avec un autre agent admet quatre valeurs : très bonne, bonne, mauvaise et très mauvaise.

Le but dans ce modèle est d'obtenir une mesure de la distance sémantique entre la recommandation d'une entité *b* pour une entité *c* faite à l'entité *a* et l'expérience personnelle

[13] www.wikipedia.org

directe que *a* a réellement perçue. Par exemple, si A demande à B à quel degré il peut avoir confiance en C et que B lui répond que C est très fiable et que A s'aperçoit finalement que C n'est pas très fiable mais juste fiable, alors A adaptera les prochaines recommandations de B en les abaissant d'un niveau. Cette distance sémantique se nomme valeur de recommandation de confiance.

Si une entité *x* veut connaître la fiabilité d'un service de stockage s_1, voici la méthodologie à suivre dans ce modèle :

1. *x* demande des recommandations aux entités qui ont eu une expérience directe avec s_1 ;
2. la deuxième étape consiste à obtenir la valeur de recommandation de confiance de chaque entité faisant une recommandation en fonction des recommandations passées ;
3. ensuite, chaque recommandation actuelle de chaque entité reçue par *x* sur le service s_1 est adaptée ;
4. en fonction de sa recommandation adaptée et de sa valeur de recommandation de confiance, chaque entité se voit attribuer un poids contribuant à la valeur de réputation finale de s_1 ;
5. la dernière étape consiste à mettre à jour les expériences entre *x* et chaque entité qui a fait une recommandation ainsi que l'expérience directe entre *x* et s_1.

Dans ce modèle, seules les entités connues participent à la recommandation. Les recommandations des entités non connues serviront à la prochaine recommandation. Les expériences directes servent seulement à calculer la distance sémantique, mais ne sont pas réellement combinées aux recommandations. Comme les auteurs rejettent la transitivité de la confiance, les recommandations de recommandations ne sont pas traitées. Étant donné que les valeurs ne sont pas des valeurs numériques faire des calculs n'est pas facile, mais la plus grosse critique pouvant être faite sur ce modèle est la manière d'attribuer les poids, celle-ci est faite ultérieurement sans aucune justification.

2) TidalTrust

(J. A Golbeck, 2005) propose l'algorithme TidalTrust qui est testé dans le réseau social FilmTrust qui est un système de recommandations de films comprenant 400 utilisateurs. Un utilisateur évalue des films entre une demi-étoile et 4 étoiles et chaque utilisateur peut évaluer sa confiance envers un autre utilisateur avec 10 valeurs discrètes comprises dans l'intervalle [1,10]. Pour Golbeck, représenter la confiance par des valeurs discrètes est plus instinctif que de la représenter dans un intervalle continu. TidalTrust est un algorithme qui permet à une

source d'inférer le score d'un film *m* à travers les scores de recommandations d'évaluateurs. Cet algorithme calcule un score de recommandations qui est une moyenne pondérée. La moyenne pondérée du score d'un film reflète l'opinion ou encore la recommandation des utilisateurs de la communauté. Pour calculer ce score, TidalTrust procède par plusieurs étapes:

1. le système cherche les évaluateurs du film m que la source connaît directement ;
2. s'il n'y a pas de connexions directes entre les évaluateurs de m et la source, le système cherche les utilisateurs connectés avec la source par un chemin de longueur 2, c'est-à-dire les utilisateurs connus par les utilisateurs que la source connait directement. Le processus continue à chercher en augmentant de 1 la longueur du chemin à chaque itération, jusqu'à ce qu'un chemin soit trouvé.
3. le score de confiance est calculé à partir de tous les utilisateurs compris dans le chemin trouvé. Pour un nœud source donné s dans l'ensemble des nœuds S, le score de recommandation r_{sm} inféré par s pour un film m est défini par :

$$r_{sm} = \frac{\sum_{s \epsilon adj(s)} t_{si} * r_{im}}{\sum_{s \epsilon adj(s)} t_{si}} \quad (2.9)$$

où adj(s) ϵ S sont les nœuds directement connectés à s dans le réseau.

Pour éviter des chaînes de trop longues tailles des limites peuvent être fixées. Cependant, la longueur du chemin n'est pas prise en compte. En effet, un score obtenu grâce à un utilisateur en lien direct avec la source dans le réseau aura la même valeur qu'un score obtenu avec un chemin de longueur 3 par exemple. De plus, cette approche ne prend pas en compte l'incertain, c'est-à-dire qu'elle ne tient pas compte des différents chemins possibles, mais calcule le score de recommandation avec le premier chemin trouvé. Le risque n'est pas non plus étudié ici, tout comme la décision de faire confiance ; il n'y a pas de seuil permettant à l'utilisateur de choisir ou non le film.

2.4.5.4.Modèles basés sur la croyance

1) Jøsang : belief model

Jøsang complète son modèle subjective logic (Audun Jøsang & Ismail, 2002) par une théorie de la croyance. Une mesure spécifique de la croyance, appelée l'opinion, est calculée pour représenter une croyance spécifique. La mesure de la croyance est utilisée dans des situations d'ignorance et d'incertain. L'opinion est un triplet défini comme suit :

$$\omega_x^A = (b, d, u) (2.10)$$

où b + d + u = 1 avec b ϵ [0, 1] représente la probabilité de vérité, d ϵ [0, 1] la probabilité de fausseté et u ϵ [0, 1] la probabilité d'incertitude. Grâce à u une entité peut exprimer son incapacité à évaluer la valeur de probabilité de x.

Jøsang a réalisé une combinaison du modèle de bêta probabilité et du modèle de croyance :

$$b = \frac{r}{r + s + 2}$$

$$d = \frac{s}{r + s + 2} \quad (2.11)$$

$$u = \frac{2}{r + s + 2}$$

Le modèle de confiance de Jøsang subjective logique qui vient d'être présenté a été utilisé dans des systèmes de commerces électroniques (A Josang, 1999).

2) Modèle de Barber

(Barber & Kim, 2001) est un modèle multiagents fondé sur la révision de la croyance, il utilise la connaissance à propos de la fiabilité d'une source d'information, la réputation de cette source. La confiance est donc définie ici comme la croyance sur les habilités et l'intention de la source d'information à fournir des informations correctes. Une information est considérée comme fiable si l'agent apportant l'information est considéré comme fiable. Le modèle est basé sur une approche bayésienne, en effet, la croyance de la réputation d'un agent a à propos d'une source d'information se calcule en fonction de la croyance en la réputation d'agents qui sont en lien avec l'agent a et avec cette source d'information. Par exemple, l'agent A demande à l'agent B sa croyance en la réputation de l'agent C. A ce moment-là, l'agent a met en place un processus de révision de la croyance, pour réviser sa croyance en la réputation de C.

Les auteurs ont construit un algorithme de révision de la croyance distribuée pour les systèmes multiagents fondés sur la réputation des sources d'information. Par exemple, si l'on veut réviser notre croyance envers la source d'informations q, on peut demander aux sources d'informations s_1 et s_2 leur croyance en la réputation de q. Cette croyance en la réputation de q est modulée en fonction de notre propre croyance en la réputation de s_1 et s_2. Grâce à ces

critères, le modèle calcule le degré de croyance que q soit vraie et que q soit faux. Le résultat final consiste à chercher la valeur de certitude révisée.

2.4.5.5.Modèles flous

Les modèles flous représentent la confiance ou encore la réputation comme des concepts linguistiques flous. Ils décrivent à quel degré un agent peut être fiable en formalisant des règles reposant sur des mesures floues.

1) Modèle du système REGRET

Dans le système de commerce électronique REGRET (Sabater & Sierra, 2002) la réputation est un concept multidimensionnel. Elle accepte une dimension individuelle, sociale et ontologique. Dans l'environnement, les agents peuvent coopérer, être en compétition ou commercer.

- dimension individuelle, il s'agit de l'expérience directe entre l'agent et l'agent cible. Un contrat entre deux agents est défini par le prix, la qualité et la date de livraison du produit. La confiance est modélisée en fonction de la différence entre le contrat initial et les résultats de la vente. Cette différence va permettre de classer l'agent cible dans une catégorie, par exemple, la catégorie « fait payer trop chère » ;
- dimension sociale, l'agent fait appel à des témoins qui ont déjà un score de réputation pour l'agent cible ou à des voisins de l'agent cible. Cette dimension fait appel à des règles floues. En fonction de la relation du témoin avec la cible et du degré de cette relation, l'agent assignera un degré de croyance aux informations apportées par le témoin ou le voisin. Par exemple, SI le témoin coopère fortement avec l'agent cible ALORS, les informations du témoin sont très mauvaises ;
- dimension ontologique, combine la réputation de différents aspects pour calculer une réputation complexe. Par exemple, mettre l'agent cible dans une nouvelle catégorie qui est une combinaison de deux catégories.

2.4.5.6.Modèles de flux

Les modèles de flux calculent la confiance grâce à des itérations transitives à travers des boucles ou encore de longues chaînes. Dans certains modèles la confiance totale est constante à travers toute la communauté donc, l'augmentation du score de confiance d'une des entités se fait au détriment des autres.

1) Modèle de Google PageRank

L'algorithme PageRank de Google (Ranking, Ranking, Order, & Order, 1998), permet de classer les résultats d'une recherche de pages web. Cette classification est basée sur la confiance, puisqu'elle est basée sur les liens entre pages. Dans une page, le lien vers une autre page est vu comme une preuve de confiance. Le PageRank d'une page mesure en fait la probabilité qu'à un surfeur, qui est sur une page donnée, d'arriver sur cette page. C'est une représentation de chaîne de Markov avec le suivi d'un très grand nombre de liens. Cette méthode repose sur l'hypothèse qu'un surfeur qui se retrouve sur une page, choisit aléatoirement de suivre un lien de cette page. Voici la méthode de calcul du PageRank :

$$R(u) = c \sum_{v \in B_u} \frac{R(v)}{N_v} \quad (2.12)$$

Avec *R(u)* le pageRank de la page *u*, *c* un facteur de normalisation et N_v le nombre de liens présents dans la page *v*.

Le PageRank d'une page est d'autant plus important que les PageRank des pages qui pointent vers elles sont importantes. En effet, chaque page qui pointe vers cette page contribue à une fraction du PageRank de cette page. Dans ce calcul, l'attribution du PageRank d'une page est récursif. On peut résumer ce calcul par trois affirmations :

- plus une page a un PageRank important, plus elle contribue au PageRank des pages vers lesquelles elle pointe ;
- plus une page contient de lien, moins elle contribue au PageRank des pages vers lesquelles elle pointe ;
- plus une page reçoit de liens, plus son PageRank est important.

2) EigenTrust

Le modèle (Kamvar, Schlosser, & Garcia-Molina, 2003) a été réalisé pour les systèmes Pair-à-Pair ; il permet d'assigner à chaque pair une valeur globale unique dérivée grâce à l'histoire de ce dernier. Pour être constant, ce score ne requiert pas la somme de tous les scores de confiance. Les scores de réputation dans ce modèle sont calculés à travers une longue chaîne transitive de calculs répétés et itératifs jusqu'à ce que les scores de confiance des agents convergent vers une valeur stable.

L'interaction d'un pair i avec un pair j peut être positive ou négative. Le nombre d'interactions positives qu'a eu i avec j est noté *sat(i, j)* et le nombre d'interactions négatives qu'a eu i avec j est noté *unsat(i, j)*.

Le score local de croyance normalisé est noté :

$$c_{ij} = \frac{\max(s_{ij}, 0)}{\sum_{l \in L} \max(s_{ij}, 0)} \in [0,1] \quad (2.14)$$

Un score de confiance hors du groupe local de i peut ensuite être calculé :

$$t_{ik} = \sum_{j \in L} c_{ij} c_{jk} \quad (2.15)$$

Ce score permet de réunir dans un score unique la confiance fonctionnelle et la confiance de référence.

Dans le modèle, $C = [c_{ij}]$ représente la matrice de tous les scores de confiance normalisée dans la communauté. Tandis que $\vec{c_\iota}$ représente le vecteur des valeurs de confiances locales de i.

$\vec{t_\iota}$quant à lui représente le vecteur contenant les valeurs de confiance t_{ik} où *i* et *k* sont séparés par n nœuds intermédiaires :

$$\vec{t_\iota} = C^n \vec{c_\iota} \quad (2.16)$$

Lorsque n est important, le vecteur $\vec{t_\iota}$ converge vers la même valeur quel que soit le pair i. Le vecteur $\vec{t_\iota}$ correspond à un vecteur de confiance globale et correspond à la valeur de confiance globale qu'a la communauté envers k. Les limites de ce modèle correspondent au fait que le score normalisé supprime les scores de satisfaction négative. De plus, un nouvel arrivant aura la même valeur qu'un mauvais pair.

2.5. Confiance pour le Cloud Computing

Plusieurs modèles de confiance ont été proposés pour le Cloud Computing.

1) **Modèle pour l'établissement de la confiance** (Khan & Malluhi, 2010)

Les auteurs ont abordé la confiance dans les systèmes Cloud à partir de la perspective des utilisateurs. Ils ont analysé les défis de confiance à partir des attentes des utilisateurs en respectant la sécurité et la confidentialité. Ils ont aussi abordé les points qu'un fournisseur de services doit améliorer afin de gagner la confiance des utilisateurs. Ils ont identifié les concepts

de *contrôle*, *possession*, *prévention* et *sécurité* comme les points-clés qui définissent le niveau de confiance pour un service. La diminution du contrôle et manque de transparence représentent les principaux aspects de perte de confiance des utilisateurs dans les systèmes de Cloud Computing. Les auteurs ont proposé de renforcer la confiance des utilisateurs par l'utilisation de solution de contrôle d'accès distants des ressources utilisateurs, l'utilisation de la transparence à l'égard des actions des fournisseurs sous la forme de traçabilité automatique et l'utilisation de la certification des propriétés et des fonctions de sécurité Cloud par une autorité de certification indépendante.

2) **Modèle basé sur la sécurité** (Sato, Kanai, & Tanimoto, 2010)

Les auteurs ont proposé un modèle de confiance pour le Cloud en terme de sécurité sociale. Les auteurs ont identifié le principal problème de sécurité comme un problème d'insécurité sociale. Ils ont essayé de l'aborder en utilisant une approche à trois volets. Ils ont subdivisé le problème de l'insécurité sociale par trois sous-domaines, à savoir le problème de multiples partis, le problème de la sécurité de l'espace ouvert et le problème de gestion de données critiques. Le problème de multiples parties aborde les défis de sécurité lorsque de multiples partis interagissent dans le système Cloud. Selon les auteurs, trois partis peuvent être clairement identifiés, le client, les fournisseurs de services et les tiers tels que les concurrents. Le problème de sécurité de l'espace ouvert aborde la question de la perte de contrôle sur l'emplacement des données et la façon par laquelle elles sont gérées une fois que le contrôle des données est délégué au fournisseur. Ils conseillent de crypter les données avant de les transférer ce qui transforme le problème en un problème de gestion des clés. Le problème de la gestion des données critiques examine la question de délégation du contrôle des données critiques à un fournisseur de services. Ils proposent pour cela de ne pas déléguer le contrôle de ces données, mais de les garder dans un Cloud privé avec une configuration hybride. Cependant, la mise en place d'un Cloud privé peut ne pas être une option pour les petites et moyennes organisations en raison des coûts élevés impliqués.

3) **Modèle basé sur les gènes de famille** (T. Wang, Li, & Zhu, 2011; T. Wang, Ye, Li, & Yang, 2010)

Le modèle de confiance basé sur les gènes de famille est basé sur l'infrastructure à clés publiques. Les auteurs ont étudié les opérations basiques telles que l'authentification des utilisateurs, la gestion des autorisations et le contrôle d'accès et ont proposé un modèle de

confiance pour le Cloud basé sur les gènes de famille (FBCT – Family-gene Based model for Cloud Trust) qui intègre ces opérations en utilisant les algorithmes génétiques.

4) **Modèle intégré au broker de CARE**(P. D. Manuel, Thamarai Selve, & Barr, 2009)

Les auteurs ont proposé un modèle de confiance intégré au broker de ressources CARE. Ce modèle de confiance peut être utilisé dans un système de Cloud et de grille. Le modèle proposé calcule la confiance en utilisant trois composants principaux, à savoir :

- un évaluateur de niveau de sécurité qui prend en compte l'authentification, l'autorisation et des mécanismes de sécurité automatique ;
- un évaluateur des avis (feedback) qui passe par trois différentes phases collecte d'avis, leur vérification et leur mise à jour ;
- un évaluateur de confiance et de réputation qui calcule les valeurs de confiance pour les ressources Cloud/Grille.

Enfin, la valeur globale est calculée en tant que somme des valeurs fournies à partir des trois évaluateurs.

5) **Modèle construit avec TPS-Trusted Platform Service** (Z. Shen, Li, Yan, & Wu, 2010) (Z. S. Z. Shen & Tong, 2010)

Les auteurs ont analysé la sécurité pour un environnement Cloud et ont décrit la fonction de plate-forme de calcul sûre pour le Cloud Computing. Ils ont aussi proposé une méthode pour améliorer la sécurité et la fiabilité en intégrant la plate-forme de calcul sûr (Trusted Computing Platform – TCP) dans les systèmes Cloud. Le TCP[14] a été utilisé pour l'authentification, la confidentialité et l'intégrité. Finalement, le modèle a été développé comme un intergiciel nommé TSS (Platform Software Stack).

6) **Modèle MTCEM-Multi-Tenancy Trusted Computing Environment Model** (Li, Zhou, Shi, & Guo, 2010)

Un modèle appelé MTCEM (Multi-Tenancy Trusted Computing Environment Model) pour le Cloud a été proposé par Li. Ce modèle a été proposé pour fournir des services IaaS sécurisés avec un mécanisme de confiance à double niveau puisque les ressources Cloud appartiennent à de multiples entités telles que le fournisseur de services et les utilisateurs, alors

[14] TCP : Trusted Computing Platform

ils appartiennent à de multiples domaines de sécurité qui définissent différentes politiques de sécurité simultanément.

7) **Modèle basé sur un module de confiance collaborative** (Yang, Qiao, Liu, Yang, & Wan, 2010)

Les auteurs ont étudié les modèles de confiance existants et ont révélé que ces modèles ignorent l'existence d'un pare-feu dans le réseau, puisque le pare-feu fait partie intégrante de chaque architecture de sécurité. Les auteurs ont proposé un modèle de confiance collaborative basé sur le pare-feu.

8) **Modèle Watermark basé sécurité** (Fu, Wang, Yu, Wang, & Sun, 2010)

Les auteurs ont étudié les problèmes de sécurité associés à l'exécution de programme dans le Cloud. Ils ont proposé un environnement d'exécution sécurisé dans le Cloud. Le modèle proposé est composé de deux parties : le centre d'administration et l'environnement serveur.

9) **Modèle basé SLA** (M. Alhamad, Dillon, & Chang, 2010)

Les auteurs ont proposé un modèle de confiance basé sur les SLA (Service Level Agreement) pour le Cloud Computing. Le modèle est composé d'un agent SLA, de module pour l'utilisateur et de répertoire pour les services Cloud. L'agent SLA est le composant de base de l'architecture : Il désigne les métriques SLA, négocie les termes avec le fournisseur, sélectionne le fournisseur en se basant sur des paramètres de QoS et surveille les activités des utilisateurs. Le module de l'utilisateur demande l'exécution externe d'un ou plusieurs services. Le répertoire des services consiste en un annuaire pour répertorier des services.

2.6. Conclusion

Ce chapitre a permis de présenter l'environnement des systèmes distribués en général et l'environnement de Cloud Computing en particulier. Le Cloud Computing se présente comme une réponse satisfaisante à la problématique de stockage et de calcul des données. Il propose d'assurer le traitement et l'hébergement des informations via une infrastructure externalisée ce qui permet aux utilisateurs de bénéficier de nombreux services en ligne sans avoir à se soucier des aspects techniques de leurs usages, tout en amortissant les coûts engendrés par la prise en charge de toutes ces données. Le Cloud Computing apparaît donc comme une opportunité intéressante mais pose logiquement la question de la sécurité des données quand elles sont hébergées par un fournisseur externe.

Les interrogations liées à la sécurité et à la fiabilité sont le principal frein à son adoption. Par conséquent, l‘enjeu essentiel du Cloud revient à recueillir la confiance des utilisateurs. À cet effet, plusieurs modèles de calcul de confiance ont été proposés afin de résoudre cette problématique.

Les différents modèles étudiés qui permettent de rendre le Cloud digne de confiance peuvent différer légèrement de ceux qui sont applicables à d’autres systèmes, mais l’objectif fixé reste le même : augmenter la performance et la compétitivité en s’appuyant sur de nouvelles technologies, lesquelles construisent graduellement leur réputation sur la base du retour d’expérience des utilisateurs. Les problèmes de sécurité rencontrés par certains fournisseurs de Cloud ont conduit les clients à prendre d’avantage conscience des risques encourus.

Ainsi, les inconvénients relatifs aux divers modèles présentés et les problèmes de sécurité rencontrés par les fournisseurs de Cloud nous ont conduits à évaluer les risques encourus et détailler les points non abordés dans les modèles existants. De ce fait, nous allons exposer la solution envisagée et notre contribution pour assurer la confiance des utilisateurs dans le chapitre suivant.

Chapitre 3 : Système proposé

3.1. Principe de base

Ce chapitre introduit *Global Trust* (Filali & Yagoubi, 2015a), un système de confiance pour la sélection de fournisseur de services dans un environnement Cloud. Nous commençons par une présentation des approches suivies et travaux connexes au modèle proposé. La section 2.2 décrit la conception et les concepts de base de l'architecture proposée. La section 2.3 se focalise sur le modèle computationnel proposé, les différents facteurs de performance ainsi que les facteurs de confiance utilisés pour le calcul de la confiance. La section 3.2.4 décrit quelques modèles d'attaques dans un environnement Cloud et présente le modèle d'attaque intégré au système de confiance.

3.1.1. Approche

Cette partie décrit l'approche suivie pour la proposition d'un système de gestion de confiance pour la sélection de service. Nous présentons d'abord un ensemble d'hypothèses nécessaire au fonctionnement du système proposé, ensuite quelques exigences à prendre en compte et enfin nous définissons les travaux de recherche sur lesquels nous nous sommes basés pour notre travail.

3.1.2. Hypothèses

Nous avons établi un certain nombre d'hypothèses qui servent de base au système proposé. Elles concernent la gestion des relations entre les entités, le fournisseur de services, le système de gestion de confiance et le consommateur :

1) on suppose que les fournisseurs sont honnêtes durant l'enregistrement, c'est-à-dire que les différentes informations fournies sont correctes ;
2) aucune information fournie par le consommateur n'est sûre ;
3) les informations de performances récupérées du fournisseur sont considérées comme sûres.

3.1.3. Exigences

Cette partie décrit les exigences qui doivent être respectées pour réaliser un système de gestion de confiance dans un environnement Cloud. Bien que ces exigences ne soient pas toutes primordiales, elles constituent les principes clés pour avoir un système complet et fiable.

Le système proposé va servir pour la sélection de fournisseurs sûrs et fiables. Le but n'étant pas de contrôler et de surveiller l'exécution des services, mais d'offrir un moyen fiable et sécurisé pour choisir un service crédible. Ainsi, le point-clé pour un système de gestion de confiance est la fiabilité du degré de confiance final. D'autres exigences viennent s'ajouter à ce point, telles que la prise en compte des performances du fournisseur de services Cloud, le filtrage des avis des consommateurs malhonnêtes, la prise en compte de diverses sources d'informations pour prendre une décision finale.

La liste suivante énumère et détaille les principales exigences pour le système proposé.

E1 : la fiabilité du degré de confiance : la valeur finale doit être fiable afin de fournir une sélection de services crédible et sûre ;

E2 : le calcul de confiance doit se baser sur divers aspects de sécurité afin de sécuriser le degré de confiance finale ;

E3 : le calcul de réputation d'un service doit provenir de plusieurs sources d'informations afin d'avoir une note complète et globale, ce qui n'est pas le cas pour plusieurs systèmes de gestion de confiance ;

E4 : la prise en compte de l'hétérogénéité et la complexité d'un environnement Cloud pour la conception du système de gestion de confiance ;

E5 : la confidentialité des données de confiance sauvegardées par le système de gestion de confiance conçu. Les données doivent être protégées des accès utilisateurs non autorisés.

3.1.4. Investigation de travaux existants

Pour modéliser un système de gestion de confiance deux parties doivent être intégrées :

1) représentation de confiance, les mesures de confiance utilisées. La confiance peut être mesurée en utilisant des valeurs discrètes ou continues ;
2) calcul de la confiance, le modèle mathématique pour l'agrégation des notes de confiance. Les métriques peuvent être :

- somme ou moyenne des notes (Resnick & Zeckhauser, 2002) (A Jøsang et al., 2007), est la méthode la plus basique pour calculer le degré de confiance. Le degré de confiance peut être calculé comme le nombre total d'évidences positives moins le nombre d'évidences négatives. Cette méthode de calcul est utilisée dans le système eBay[15] (Boyd, 2002). Une autre technique plus avancée de calcul consiste à calculer la moyenne de toutes les notes, cette méthode est utilisée par Amazon[16] et Epinions[17]. Les deux techniques présentent un avantage, puisqu'il est facile de comprendre le principe derrière la note finale. Toutefois, cette méthode reste primitive et fournit une médiocre représentation de la confiance ;
- logique floue (Bharadwaj & Al-Shamri, 2009)(Iltaf & Ghafoor, 2013)(Song et al., 2006)(Song, Hwang, Zhou, & Kwok, 2005) (Manchala, 1998)(Sherchan, Loke, & Krishnaswamy, 2006), la confiance est représentée comme un concept flou et le degré, à partir duquel l'entité est considérée comme sûre, est mesuré à base de la logique floue ;
- modèles à base de flux (Brin & Page, 2012)(Huang, Nie, Huang, & Tu, 2012)(Kamvar, Schlosser, & Garcia-Molina, 2003)(Lempel & Moran, 2000)(SIMONE, ŠKORIĆ, & ZANNONE, 2012), pour ce modèle, les systèmes évaluent les entités par des boucles itératives ou des chaînes. Par exemple, dans Google PageRank (Ranking et al., 1998), chaque hyperlien vers une page web, augmente son rang et chaque hyperlien à partir d'une page web réduit son rang ;
- modèles discrets (Manchala, 1998), les êtres humains sont souvent capables de noter facilement avec des mesures discrètes qu'en mesures continues. Par exemple, la confiance d'une entité peut être élevée, moyenne ou basse ;
- modèles probabilistes :
    - Les réseaux bayésiens (Tavakolifard & Knapskog, 2009)(Teacy, Luck, Rogers, & Jennings, 2012)(Whitby, Jøsang, & Indulska, 2005)(Zhang & Fang, 2007)(Halberstadt, Mui, Mohtashemi, Ang, & Szolovits, 1998)(Mui et al., 2002), le degré de confiance est calculé par des mises à jour des fonctions de densité de probabilité PDF. Plus spécifiquement, il prend une entrée binaire positive ou négative et calcul la réputation en se basant sur les anciennes notes et les nouvelles. Le résultat est représenté par (α,β), le nombre de valeurs

[15] http://www.ebay.com/
[16] http://www.amazon.com/
[17] http://www.epinions.com/

positives et négatives respectivement et la valeur de probabilité de béta PDF. Cette méthode est complexe et difficile pour un simple utilisateur à comprendre.

  - densité de probabilité beta (van Deursen, Koster, & Petković, 2008).
- logique subjective (Audun Jøsang, Hayward, & Pope, 2006)(Y. Wang & Singh, 2006).

Un bon modèle de calcul doit être facile à comprendre puisque les utilisateurs doivent comprendre ce que le taux de confiance signifie. Le degré de confiance doit être aussi simple que la somme des notes, on peut aussi utiliser des fonctions complexes ou encore appliquer des techniques d'optimisation et utiliser un réseau de neurones. Cependant, un modèle complexe pourrait être difficile à comprendre pour l'utilisateur.

Divers chercheurs ont utilisé ces méthodes pour représenter et calculer la confiance.

1. Logique Subjective : un modèle a été proposé par Jøsang (Audun Jøsang, 2001) (Audun Jøsang, Gray, & Kinateder, 2006). Dans ce modèle l'auteur décrit la représentation de la croyance incertaine par la logique subjective. Dans la logique subjective, une opinion est représentée par un triplet (b, d, u) avec b, d, u représentant respectivement le taux de croyance, de non croyance et d'incertitude. Leur somme est égale à 1 (a+b+u = 1). La formalisation de l'incertitude u permet d'exprimer et d'expliquer les degrés de non confiance, méfiance et confiance, qui prend en compte une connaissance partielle du fournisseur.
2. Logique floue : (Song et al., 2006) ont développé un système de gestion de réputation basé sur l'approche de la logique floue. Ils ont utilisé les capacités de la logique floue pour gérer l'incertitude, le flou et les informations incomplètes. Le système proposé utilise les règles d'inférence de la logique floue pour calculer les degrés de confiance locaux et la réputation globale.
3. Réseaux Bayésiens : Dans (Zhang & Fang, 2007) un modèle de réputation bayésien a été proposé pour le calcul des valeurs de confiance en se basant sur les fonctions de densité béta. La valeur de réputation est calculée par $\alpha+\beta+2$, où α et β sont deux paramètres représentant le nombre de résultats positifs et négatifs respectivement ;
4. Modèle de croyance : (Y. Wang & Singh, 2006) ont modélisé la réputation comme une croyance a trois dimensions (b ; d ; u) représentant respectivement les valeurs probabilistes positives, négatives et incertaines.
5. Théorie de graphe ; (Huang et al., 2012) ont proposé une méthode pour agréger les réseaux sociaux hétérogènes et ont utilisé la topologie améliorée du graphe de confiance pour prédire la réputation.

6. Logique certaine : Dans (Ries, Habib, Mühlhäuser, & Varadharajan, 2011) les auteurs ont proposé un modèle pour évaluer les termes de logique propositionnelle avec l'incertitude. Le modèle a été prouvé pour être conforme à une évaluation probabiliste des termes de la logique propositionnelle et de la logique subjective. L'approche proposée est plus expressive que les approches probabilistes et elle est aussi expressive que la logique subjective en matière d'incertitude. Elle fournit une représentation plus simple puisqu'elle est basée sur des paramètres indépendants contrairement à la logique subjective. De plus, les paramètres d'évaluation de la confiance peuvent être dérivés à partir de multiples approches et sources. Enfin, les auteurs ont évalué la fiabilité de leur système dans un scénario pour le Cloud Computing.

Les différentes approches pour le calcul de la confiance présentent plusieurs inconvénients. D'abord, les approches probabilistes sont supposées connues, ce qui est vraiment difficile dans un environnement Cloud.
Les approches basées sur les probabilités bayésiennes supposent l'utilisation d'une fonction de densité de probabilité, ce qui résulte en des distributions mathématiques complexes et des interprétations difficiles.
Les approches basées sur la logique floue représentent un type différent d'incertitude, plus orienté vers l'incertitude linguistique et floue.
Les approches basées sur la logique subjective sont plus appropriées pour notre cas, mais les paramètres de croyance, non-croyance et certitude sont dépendants les uns des autres.
Enfin, l'approche basée sur la logique certaine est plus appropriée pour modéliser notre solution puisqu'elle n'est pas sujette aux inconvénients cités.
À cet effet, nous allons détailler l'approche de la logique certaine dans la partie 3.1.1.1.

3.1.1.1.Logique certaine

Dans ce modèle, la confiance d'un service représente la croyance qu'une proposition (ou une combinaison de propositions) est vraie. Par exemple, un service est un service digne de confiance s'il est attendu qu'il fournisse une certaine qualité de service.

Ce modèle est basé sur le principe de l'opinion. Chaque opinion d'une proposition x représenté par O(x) = (r, c, f) est modélisé comme un triplet de valeurs (r, c, f) respectivement la note moyenne, la certitude et l'expectation initiale.

- Note moyenne r(x) représente le degré auquel les observations passées soutiennent la vérité à propos d'une proposition.
- Certitude c(x), degré auquel la croyance est supposée représenter le futur.

- Attente initiale f(x), degré qui fournit le poids de la certitude ou l'incertitude d'une proposition.

Le modèle de logique certaine est calculé comme suit :

$$O(x) = (rx, cx, fx) \qquad (3.1)$$

$$r(x) = \begin{cases} 0.5 & if\ p+n=0 \\ \frac{p}{p+n} \end{cases} \qquad (3.2)$$

$$c(x) = \frac{N*(p+n)}{w*NA+N*(p+n)} \qquad (3.3)$$

$$f(x) = 0.99 \qquad (3.4)$$

Avec :

w : représente un poids pour les évidences neutres ;

p : nombre d'évidences positives ;

n : nombre d'évidences négatives ;

NA : nombre d'évidences neutres ;

N : nombre d'évidences totales avec N = p+n+NA ;

L'espérance de probabilité d'une opinion est utilisée pour fournir la valeur de confiance finale. L'espérance d'une opinion est donnée par la formule suivante :

$$E(O(x)) = r*c + (1-c)*f \qquad (3.5)$$

où $E \in [0,1]$

3.2. Conception

Tous ces modèles permettent de quantifier la confiance comme une valeur probabiliste basée sur la confiance directe ou la confiance recommandée. Toutefois, ils ignorent les facteurs objectifs du service fourni (les facteurs de qualité de service QoS).

3.2.1. Considérations de conception

Dans cette partie, nous présentons plusieurs dimensions de classification des principaux problèmes de confiance pour les systèmes Cloud.

Ces considérations ont servi de base pour la conception du système développé.

3.2.1.1.Défis de recherche

La sélection de service Cloud est parmi les problèmes les plus difficiles pour la sécurité dans le Cloud Computing (Fernandes et al., 2014), (T. Noor, Sheng, Zeadally, & Yu, 2013), (Pearson & Benameur, 2010). En fait, sélectionner le meilleur service revient à sélectionner le service le plus fiable. Ainsi, plusieurs études se concentrent sur le problème de confiance pour l'adoption de service Cloud, ce qui conduit à faire surgir plusieurs défis de recherche tels que :

Récupération d'informations sur la confiance :

Lorsqu'un consommateur a besoin d'informations de confiance sur un potentiel fournisseur de services, deux choix s'offrent à lui, soit il se base sur son expérience directe avec ce fournisseur soit la confiance est évaluée à partir de sa réputation à partir d'une tierce partie. Cependant, en général le consommateur n'a pas encore d'interaction directe avec le fournisseur. De plus connaître la réputation d'un fournisseur de services pose quelques problèmes tels que :

- comment rechercher des informations de confiance utiles dans le système, puisque la confiance est calculée à partir de divers consommateurs et dans différentes catégories ;
- est-ce que les autres consommateurs sont prêts à partager les informations sur la confiance ;
- est-ce que les informations offertes correspondent à des valeurs réelles reflétant le comportement du fournisseur de services.

Les systèmes de gestion de confiance peuvent utiliser des informations explicites et implicites pour la prise de décision. Ces informations peuvent être collectées à partir de différentes sources :

- utilisateurs, les évidences à partir d'interactions directes avec le service ;
- fournisseur de services Cloud, les informations fournies lors de l'enregistrement des services ;
- recommandations à partir d'autres sources et feedbacks donnés par les autres utilisateurs ;
- les valeurs de QoS et de performance du service à partir de contrat SLA ou service de contrôle.

Agrégation d'information de confiance

Si un consommateur récupère plusieurs valeurs de confiance d'un fournisseur de services, comment regrouper toutes ces informations pour générer une valeur de confiance pour

un fournisseur de services, puisque différents consommateurs peuvent avoir différents points de vue.

Représentation d'un contexte complet

Dans la plupart des modèles de confiance, un service est représenté par une seule valeur attribuée par le consommateur. Cependant, dans le monde réel, il est difficile d'utiliser une valeur unique pour représenter des contextes complexes liés à un service. (Sensoy, M. and Yolum, 2006).

Fiabilité des évidences et feedbacks des informations

Les systèmes de confiance représentent une tendance importante dans le processus décisionnel de services Cloud. Le principe de base est de laisser les consommateurs noter les services utilisés, par exemple à la fin d'une transaction et utiliser les évaluations agrégées sur un service donné pour obtenir une valeur de confiance. Cette valeur peut être utilisée pour soutenir d'autres utilisateurs afin de décider d'interagir avec ce service. Le problème qui se pose est la fourniture de diverses notes malhonnêtes résultant dans des évaluations incorrectes des services.

Autres défis

Un bon système de gestion de la confiance pour les services de Cloud Computing devrait prendre en considération les exigences suivantes :

1. La confiance doit couvrir plusieurs facteurs de qualité de service au sein de plusieurs services qui offrent des fonctionnalités similaires. Les utilisateurs ont besoin de connaître non seulement le but du service mais aussi les qualités du service.

2. La confiance doit être calculée en fonction des préférences de l'utilisateur puisque les différents utilisateurs peuvent être intéressés par des caractéristiques différentes et un service adapté à un utilisateur particulier peut ne pas l'être pour d'autres utilisateurs.

3. Il est nécessaire de combiner les aspects subjectifs (les feedbacks des utilisateurs) et aspects objectifs (performance QoS) pour évaluer la confiance. La confiance est une notion subjective qui prédit l'action future d'une entité en se basant sur les actions passées. Ainsi, les évaluations des utilisateurs représentent un facteur important pour partager des connaissances sur l'expérience directe de l'utilisation d'un service Cloud.

3.2.1.2.Motivation

La motivation de cette thèse est de résoudre quelques problèmes de confiance en développant un système de gestion de confiance comprenant un modèle de calcul de confiance pour le Cloud Computing.

Le modèle de confiance proposé contribuera à résoudre les questions suivantes :

- agrégation d'information de confiance en fournissant une formule d'agrégation de la confiance qui considère les expériences des utilisateurs directes, les recommandations d'autres consommateurs, qualité de service du fournisseur, le temps de la référence de confiance, etc… ;
- représentation de la confiance dans un contexte complet en proposant une méthode formelle pour la description d'un contexte complet du service ;
- structuration de la confiance en prenant en compte plusieurs paramètres pour calculer le degré de confiance ;
- efficacité en proposant un concept de taux de couverture de fonctionnalité ;
- évaluation de la qualité de service du fournisseur de services.

3.2.2. Architecture

La plate-forme nommée *Global Trust* est une plate-forme pour la gestion de confiance et la sélection de fournisseur dans un environnement Cloud Computing.

La figure.3.1 décrit les composants principaux du système de confiance, qui consistent en 3 différents niveaux : à savoir demandeur de service, gestionnaire de service et fournisseur de services.

A) Le gestionnaire de service

Ce niveau représente le noyau du système. Il comprend les services de gestion de confiance où un utilisateur peut donner un avis sur un service utilisé et où un fournisseur peut enregistrer un service.

- Le calculateur de confiance constitue l'élément principal de ce niveau. Il est responsable de la sélection du fournisseur de services en se basant sur le modèle de calcul proposé.
- Le moniteur de confiance global est constitué de deux moniteurs : moniteur de performance et moniteur de confiance. Il est responsable de la supervision de l'exécution de l'application.

- Le broker de confiance global est en charge de l'échange des ressources entre les consommateurs et fournisseurs de services. Il sélectionne le fournisseur le plus approprié pour fournir le service demandé.

B) Demandeur de service

Ce niveau est constitué de différents demandeurs de service pour la consommation des services offerts par les fournisseurs du Cloud. Par exemple, une nouvelle organisation ayant un budget limité peut faire appel à des services Cloud tels que l'hébergement de données et d'applications dans Amazon S3[18].

Les consommateurs de service peuvent donner leur avis et feedbacks pour l'utilisation d'un service donné en invoquant le système de gestion de confiance.

C) Fournisseur de service

Le niveau fournisseur de service contient les différents services Cloud, les infrastructures IaaS, les plate-formes PaaS et les applications SaaS.

[18] https://aws.amazon.com/fr/s3/

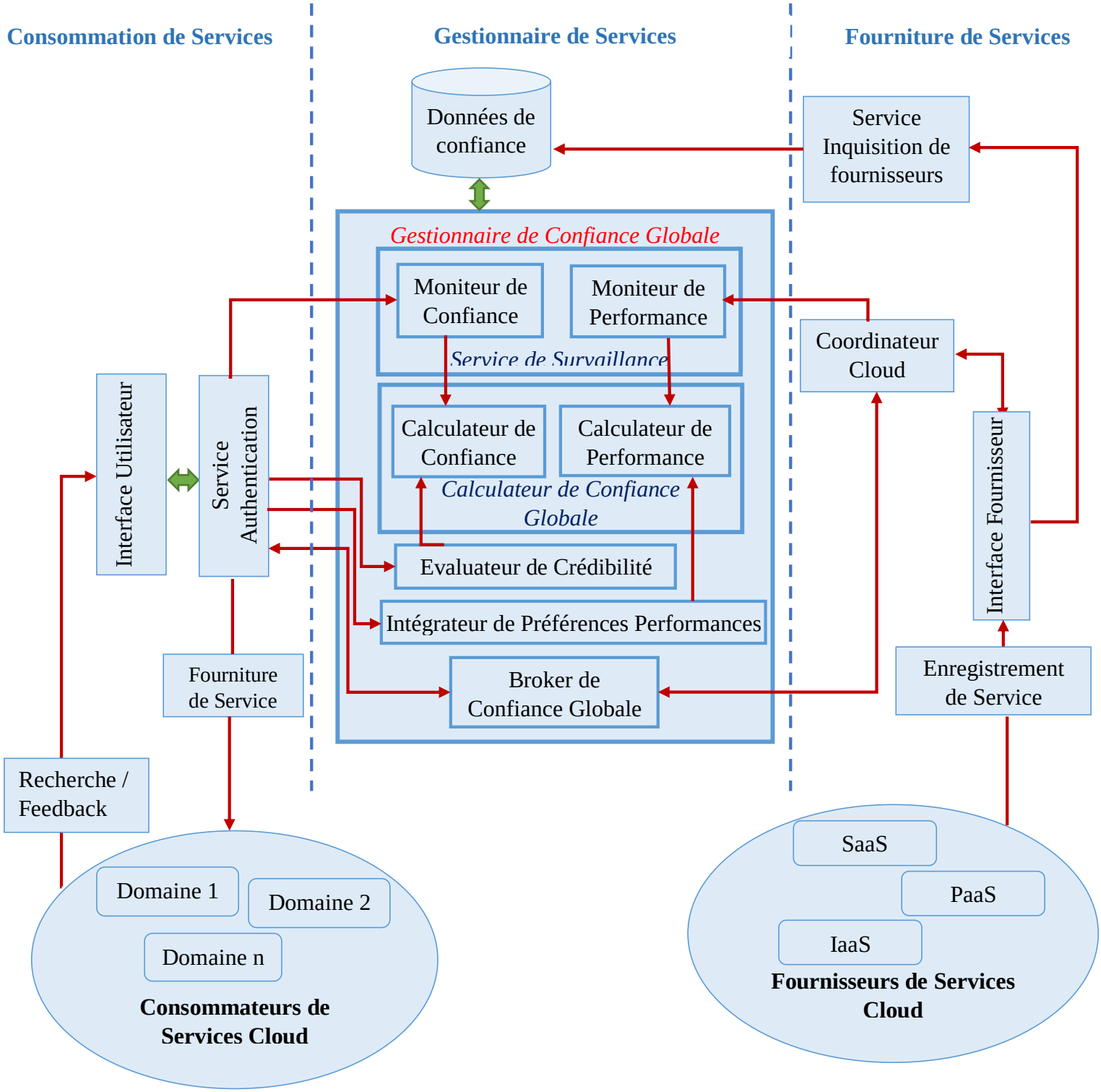

Figure.3.1. Architecture du système de gestion de confiance « Global Trust »

3.2.3. Composants du système

Le système est composé de divers composants à savoir :

1) **Services utilisateurs** : l'utilisateur se verra offrir un ensemble de services pour s'enregistrer au système de confiance afin de pouvoir trouver le service le plus crédible et approprié pour son cas et afin de fournir une note et des feedbacks du service utilisé.
   - Interface Utilisateur (IU) : l'utilisateur pourra interagir avec la plate-forme à travers une interface web et ergonomique.
   - Service d'authentification (SA): comme nous l'avons déjà précisé la sécurité et la confidentialité constitue un aspect important pour gagner la confiance des utilisateurs. À cet effet, nous avons inclus dans la plateforme un service pour gérer les accès utilisateurs à la plateforme. Ainsi, l'utilisateur doit être authentifié afin d'accéder à la plateforme.
   - Fourniture de services (FS) : permet l'interaction de l'utilisateur avec la plateforme pour l'utilisation d'un service.
2) **Services de base de la plateforme** : les principaux services de la plateforme constituent le gestionnaire General Trust (GGT) qui est responsable de la surveillance, de l'évaluation et du calcul de la confiance ainsi que la sélection des services.
   - Moniteur de cerformance (MP) : responsable de la collecte des données de performances et qualité de services à partir des services utilisés.
   - Moniteur de confiance (MC) : responsable de la gestion des notes utilisateurs et de leurs feedbacks.
   - Calculateur de Confiance Globale (CCG) : constitué de deux composants : le calculateur de confiance (CC) et le calculateur de performance (CP). Le CC est basé sur le modèle CertainTrust (Ries et al., 2011) : Le CP intègre une valeur de performance pour fournir une évaluation plus objective sur la qualité des services utilisés. Pour la sélection finale de services, les valeurs des deux calculateurs sont agrégées afin de donner une valeur de confiance globale.
   - Calculateur de Performance (CP) : le CP est responsable du calcul de la valeur de performance. Pour cela, le calculateur utilise une fonction objective basée sur plusieurs attributs représentant des valeurs de QoS pour des services de Cloud Computing qui sont : puissance, temps de réponse, efficacité, transparence, interopérabilité, efficacité, disponibilité et sécurité.

- Le calculateur de confiance (CC) : le calculateur de confiance permet de calculer les valeurs subjectives sur les opinions des utilisateurs. À cet effet, le calculateur procède au calcul d'une valeur de confiance directe. Ensuite, ces valeurs sont utilisées avec le modèle d'opinion afin d'avoir une valeur de confiance pour un service donné.
- Évaluateur de Crédibilité (EC) : ce composant permet de filtrer les opinions biaisées. Il consiste à ignorer les avis des utilisateurs que le système considère malhonnêtes ou faux. Le plus simple cas serait d'utiliser une distance euclidienne pour calculer la distance entre l'opinion d'un utilisateur sur un service donné et la moyenne des opinions des autres utilisateurs à propos du même service. Ce composant est basé sur un nouveau modèle que nous avons proposé qui sera abordé dans la dernière section.
- Intégrateur de Préférences de Performance (IPP) : ce composant est responsable de la gestion des préférences données par les utilisateurs à travers l'interface graphique et de l'intégration de ces préférences dans le processus de calcul de performances.
- Broker de Confiance Globale (BCG) : responsable de la sélection du service le plus fiable en se basant sur la valeur finale de confiance globale qui est calculée par le composant CCG.

3) Services fournisseurs :

- Inquisition du fournisseur de services (IFS) : procède à la recherche des différents paramètres des fournisseurs de services à travers la base de données globale. Il donne un accès limité d'informations et de statistiques pour les fournisseurs à propos des services fournis, via l'interface web.
- Authentification du Fournisseur (AF) : permet d'authentifier le fournisseur afin d'avoir un accès aux informations de confiance pour leur service.
- Enregistrement de Service (ES) : responsable de l'indexation de la liste de services offerts par les différents fournisseurs.
- Coordinateur Cloud (CC) : permet de gérer les différents fournisseurs pour accéder au système. Il récupère les informations à partir du fournisseur pour la surveillance. Il est aussi responsable de la coordination du broker de la plateforme avec les services des fournisseurs.

3.3. Modèle de calcul

Notre modèle se concentre sur l'estimation de la valeur de confiance pour un fournisseur dans le Cloud Computing et est basé sur des informations directes et d'autres informations.

L'approche de modélisation que nous avons suivie se base sur la confiance certaine (Ries, 2009) et combine les opérateurs de la logique certaine (Ries et al., 2011). Le modèle d'opinion est basé sur la logique subjective (Audun Jøsang, Hayward, et al., 2006) et est utilisé pour modéliser l'incertitude de la confiance.

Dans la plupart des travaux présentés, la confiance a été calculée à partir des avis d'utilisateurs selon divers formalismes. Toutes ces méthodes sont basées sur les notes d'utilisateurs, cependant ils négligent le fait que la performance fournie par un service représente une part importante pour gagner la confiance de l'utilisateur. Ainsi, nous avons proposé un modèle qui intègre ces deux moyens pour calculer une valeur globale de confiance.

3.3.1. Sources des évidences pour le calcul de confiance

Pour ce modèle, les évidences sont collectées à partir de différentes sources à savoir :

- l'interface utilisateur ;
- le fournisseur de services ;
- les recommandations à partir d'autres sources ;
- les valeurs de QoS.

3.3.2. Modélisation de la confiance

Il existe plusieurs façons de modéliser la confiance, allant d'une simple représentation binaire jusqu'à la représentation dans un intervalle de valeurs réelles.

Dans notre cas, les valeurs de confiance vont d'abord être calculées comme des valeurs réelles dans un intervalle [0, 1]. Ensuite et ceci afin de l'intégrer au modèle d'opinion nous allons les représenter en des valeurs qualitatives représentant des évidences positives, des évidences négatives et des évidences neutres.

3.3.3. Calcul de la confiance

Le système de confiance proposé se base sur un modèle conceptuel pour la sélection de fournisseurs fiables. Ce modèle est basé sur la logique certaine présentée dans la section 3.1.1.1 et est basé sur deux parties importantes :

- Le modèle de performance, qui sert à intégrer les paramètres objectifs des services Cloud dans un système de confiance ;
- Le modèle de confiance, qui est basé sur les avis et recommandations des utilisateurs.

Les valeurs de ces deux modèles seront agrégées par un opérateur de logique subjective afin de donner la valeur de confiance globale.

3.3.3.1.Calcul de valeur de Performance

Plusieurs chercheurs ont étudié les modèles de confiance existants. Ils ont défini la confiance comme « *la croyance ferme en la capacité d'une entité à agir de manière cohérente, sécurisée et fiable dans un contexte spécifié* ». D'autres chercheurs ont affirmé que la confiance est une composition de multiples caractéristiques telles que : efficacité, honnêteté, sincérité, fiabilité, sécurité, compétence, rapidité, QoS par rapport au contexte de l'environnement.
L'auteur (P. Manuel, 2013) a proposé un modèle de confiance basé sur la qualité de service pour le Cloud Computing. Le modèle utilise quatre attributs de performances à savoir disponibilité, efficacité, intégrité des données et efficacité globale. Toutefois, calculer la performance en se basant sur ces quatre attributs est insuffisant pour valider un modèle. Il doit se baser sur des mesures standardisées et approuvées dans le contexte du Cloud Computing.
Ainsi, afin de proposer un modèle de performance valide, nous allons utiliser des attributs définis par l'Indice de Mesure de service SMI (Service Measurement Index).
Cloud Service Measurement Index Consortium (CSMIC) (Garg, Versteeg, & Buyya, 2013) propose une plateforme basée sur les caractéristiques communes des services Cloud. L'objectif de ce consortium est d'exprimer chaque attribut de QoS à travers la plateforme et d'offrir ainsi une méthodologie pour le calcul d'un indice standard pour les différents services Cloud. Pour cela, CSMIC a conçu l'Indice de Mesure de Service SMI, qui consiste en un ensemble d'indicateurs clés de performance KPI (Key Performance Indicators) afin de standardiser l'évaluation de services Cloud.
Pour notre modèle de performance, nous avons sélectionné les attributs : puissance, temps de réponse, efficacité, transparence, interopérabilité, fiabilité, disponibilité et sécurité. Le calcul de certains de ces facteurs est inclus dans l'ensemble KPI.

Nous présentons dans ce qui suit les attributs utilisés ainsi que les formules de mesure de chaque attribut, en sachant que x représente un service donné.

- **Puissance** : représente la puissance des différents composants de l'infrastructure physique. Elle est calculée selon :

$$\boldsymbol{Puissance}\ (\boldsymbol{x}) = w_1 * p + w_2 * r + w_3 * s \quad (3.6)$$

Avec

w_1, w_2, w_3 : les poids des différents composants tels que $w_1+w_2+w_3 = 1$;

p : la puissance du processeur normalisée ;

r : la puissance du réseau normalisée ;

s : la puissance de stockage normalisée.

- **Temps de réponse** : la rapidité de la disponibilité d'un service peut être mesurée en terme de temps de réponse. Par exemple, si un utilisateur demande une machine virtuelle d'un fournisseur Cloud, alors le temps de réponse représente le temps pris par le fournisseur Cloud afin de répondre à la demande de l'utilisateur.

  Le temps de réponse pour un service donné dépend de plusieurs facteurs, mais dans notre cas, nous nous sommes focalisés sur le temps de réponse moyen qui est calculé selon :

$$\boldsymbol{Temps\ de\ Réponse}\ (\boldsymbol{x}) = \frac{\sum_i T_i}{n} \quad (3.7)$$

Avec

T_i : durée de demande d'un service donné pour un utilisateur i et lorsqu'il est effectivement disponible ;

n : nombre total de requête pour un service Cloud.

- **Efficacité** : l'efficacité d'un service Cloud représente l'utilisation effective d'un service loué. Ainsi, une valeur plus élevée d'efficacité indique que la surcharge sera moins importante. L'efficacité est donnée par :

$$Efficacité\ (x) = \frac{T_e(n,m)}{T_e(n,m) + T_0} \quad (3.8)$$

Avec

$T_e(n,m)$: temps d'exécution de n taches par m machine ;

T_0 : temps total dû au retard d'exécution (tel que l'initialisation des machines, intercommunication entre les machines, etc…).

- **Transparence** : la transparence représente un facteur important pour les services Cloud. La transparence indique jusqu'à quel point l'utilisateur de services sera affecté par des changements dans le service. Ainsi, la transparence peut être représentée par le temps durant lequel la performance de l'utilisateur sera affectée par les changements d'un service. Il peut être représenté par :

$$Transparence(x) = \sum_{i} \frac{\sum \frac{t_i}{n}}{u} \quad (3.9)$$

Avec

t_i : temps d'affectation lors d'un changement pour un utilisateur i ;

n : nombre de fois où l'utilisateur i est affecté par un changement ;

u : nombre d'utilisateurs utilisant le service x.

- **Interopérabilité** : l'interopérabilité est la capacité d'un service à interagir avec les autres services offerts que ce soit par le même fournisseur ou par d'autres fournisseurs. ce paramètre peut être calculé par :

$$\text{Interopérabilité (x)} = \frac{p}{n} \quad (3.10)$$

Avec

p : nombre de plate-formes offertes par le fournisseur ;

n : nombre de plate-formes requises par les utilisateurs pour l'interopérabilité.

- **Fiabilité** : la fiabilité représente la façon dont un service fonctionne sans défaillance pendant une certaine durée et contexte. Ainsi, la fiabilité est définie par le temps moyen de défaillance promis par le fournisseur Cloud et les défaillances antérieures rencontrées par les utilisateurs :

$$\text{Fiabilité (x)} = \left(1 - \frac{f}{n}\right) * p_{mttf} \quad (3.11)$$

Avec

f : nombre d'utilisateurs ayant rencontrés une défaillance avec le service x dans un intervalle de temps inférieur à celui promis par le fournisseur ;

n : nombre total d'utilisateurs du service x ;

p_{mttf} : temps moyen de défaillance promit par le fournisseur.

- **Disponibilité** : la disponibilité est le pourcentage de temps pour lequel l'utilisateur peut accéder au service.

$$\text{Disponibilité(x)} = \frac{t - u}{t} \quad (3.12)$$

Avec

t : temps de service total ;

u : temps total pour lequel le service x n'a pas été disponible.

- **Sécurité** : représente le niveau de sécurité offert par le fournisseur de services. C'est une valeur plutôt qualitative, mais nous pouvons la calculer par :

$$Sécurité\ (x) = \frac{\sum_i \frac{n}{t_i}}{u} \quad (3.13)$$

Avec

n : nombre de niveaux de sécurité offerts par le fournisseur de services ;

t_i : nombre total de niveau de sécurité requis par l'utilisateur i ;

u : nombre total d'utilisateurs du service x.

La valeur finale de performance d'un service x sera calculée par une fonction objective avec les attributs décrits dans les équations 3.6, 3.7, 3.8, 3.9, 3.10, 3.11, 3.12, 3.13 :

$$P(x) = \sum_{i=0}^{9} w_i * x_i \quad (3.14)$$

Avec

w_i : poids attribué à chaque attribut, ces poids étant définis par les préférences utilisateurs avec $\sum w_i = 1$;

x_i : attribut i pour le service x.

3.3.3.2. Calcul des valeurs de confiance

La confiance est un terme qui décrit le degré de croyance dans quelque chose. Évaluer la confiance pour un service permet aux utilisateurs de prédire son comportement futur (Muller & Schweitzer, 2013).

Afin d'évaluer la confiance directe pour un service Cloud, nous nous sommes basés sur trois paramètres principaux qui sont le coût, le temps et la satisfaction.

Pour chaque transaction effectuée, une valeur de confiance est calculée comme combinaison de ces trois facteurs.

$$C\ (x) = \begin{cases} 0.5\ initial \\ \propto \dfrac{\sum_{k=1}^{n}\left(f(t_k) * f(c_k) * f(s_k)\right)}{n} \end{cases} \quad (3.15)$$

k : k^éme utilisation du service ;

α : ***facteur d'ajustement*** :

$$\alpha = \sqrt{\frac{n}{T+1}} \quad (3.16)$$

Avec

n : Nombre d'utilisations satisfaisantes du service x ;

T : Nombre total d'utilisations du service x ;

$f(t_k)$: ***facteur d'atténuation*** :

$$f(t_k) = \frac{e^{-(t_0 - t_{k-1})}}{T} \quad (3.17)$$

Avec

t_0 : temps de la 1ére utilisation du service x ;

t_{k-1} : temps de la dernière utilisation du service x ;

T : temps total d'utilisations du service x ;

$f(c_k)$: ***facteur de coût*** :

$$f(c_k) = \frac{S}{p^a * r^b * s^c * m^d} \quad (3.18)$$

Avec

a, b, c, d : poids des différents facteurs de coût et a+b+c+d=1 ;

S : coût du service ;

p : coût d'une unité de processeur ;

r : coût d'une unité de réseau ;

s : coût d'une unité de stockage ;

m : coût d'une unité de RAM ;

$f(s_k)$: ***facteur de satisfaction*** :

$$f(s_k) = \begin{cases} 1 & si\ utilisateur\ satisfait \\ \dfrac{\sum c}{t} & sinon \end{cases} \quad (3.19)$$

Avec

c : critères de satisfaction du service x ;

t : nombre total de critères de satisfaction du service x.

Les critères de satisfaction sont définis par les attributs de performances. L'utilisateur donne une note à chaque attribut défini dans le modèle de performance. Ces valeurs sont normalisées afin de calculer le facteur de satisfaction.

Cependant des utilisateurs malintentionnés peuvent donner de fausses notes aux services. Afin de protéger le système contre ce type d'attaque, nous avons intégré un modèle d'attaque qui sera abordé dans la section 3.4.

3.3.3.3.Calcul de la valeur globale

La confiance globale finale est calculée en fonction des valeurs de confiance et de performance en appliquant le modèle d'opinion de logique certaine.

Une fois les valeurs de performance ou de confiance calculées, ces valeurs vont représenter des évidences positives, négatives ou neutres. Une évidence est positive si la valeur de performance ou de confiance est supérieure à la moyenne des valeurs. Elle est neutre si elle est égale à la moyenne des valeurs. Elle est négative si elle est inférieure.

Une fois les évidences rassemblées. Nous appliquons aux évidences de performance le modèle d'opinion ainsi qu'aux évidences de confiance.

Ces deux valeurs seront agrégées par un opérateur de logique certaine qui est l'opérateur de consensus.

$$R = O(O(P(x) \oplus O(T(x))) \quad (3.20)$$

Les opérateurs logiques présentés par (Ries et al., 2011) sont utilisés pour combiner de multiples opinions afin de former une seule opinion en utilisant des opérateurs tels que la conjonction, disjonction ou le consensus afin d'effectuer des opérations logiques sur les opinions. L'opérateur de consensus ($\oplus$) permet d'agréger des opinions sur une même évidence mais provenant de sources différentes et indépendantes.

Cet opérateur est défini tel que suit (S. M. Habib, Varadharajan, & Muhlhauser, 2013) :

$$O(A,B) = O(A) \oplus O(B) \quad (3.21)$$

$$O(A,B) = F(t_{A,B}, c_{A,B}, f_{A,B})$$

$$t_{A,B} = \frac{c_A * t_A * (1 - c_B) + c_B * t_B * (1 - c_A)}{c_A * (1 - c_B) + c_B * (1 - c_A)}$$

$$C_{A,B} = \frac{c_A + c_B - 2 * c_A c_B}{1 - c_A c_B}$$

$$f_{A,B} = 0.99$$

3.3.4. Modèle de sélection

La sélection d'un service fait partie des problèmes les plus connus du Cloud Computing en particulier et des services Internet en général.

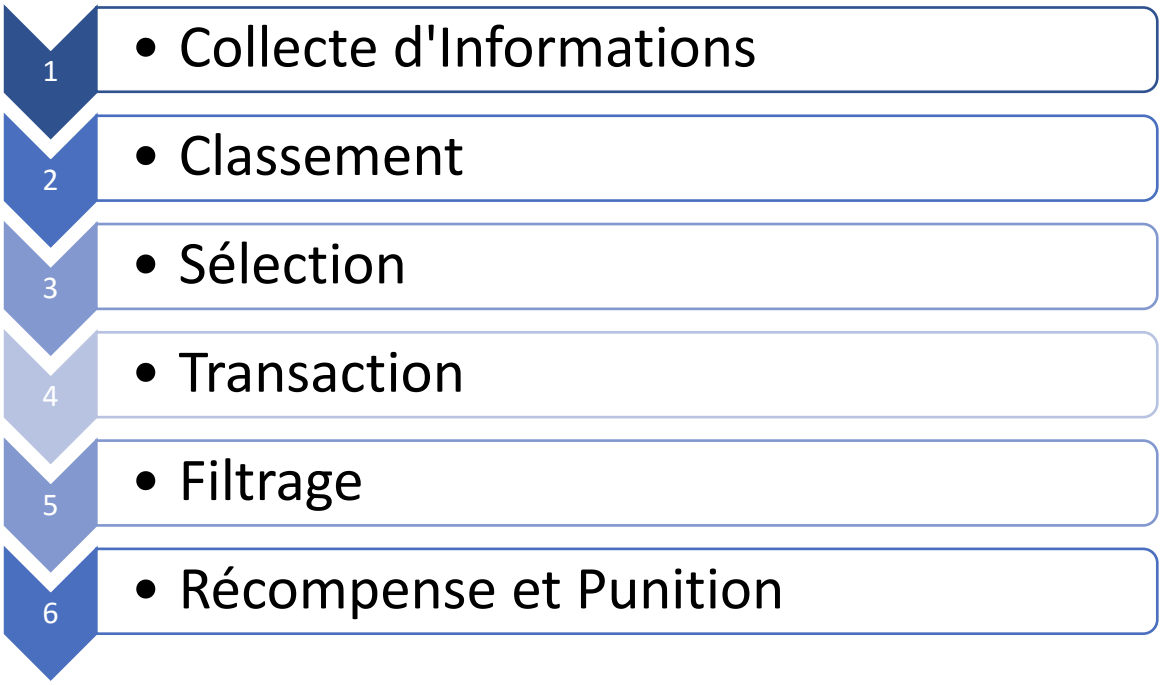

Figure.3.2. Modèle de sélection

La méthodologie suivie afin de proposer un modèle de sélection de services est la suivante :

1. Collecte d'informations sur un service donné en se basant sur son expérience directe, sur les avis d'autres utilisateurs de services, sur la qualité de services offertes, etc… pour notre modèle, nous avons utilisé les sources suivantes :
    - fournisseurs du service ;
    - performances des services :
    - feedbacks des utilisateurs ;
    - recommandations des utilisateurs ;
    - préférences des utilisateurs :
2. Classement : consiste en l'agrégation de toutes les informations reçues correctement et calculer une valeur finale en se basant sur le modèle de calcul présenté précédemment ;
3. Sélection de l'entité : consiste en la sélection du service le plus fiable et le plus sûr à partir des valeurs fournies par le modèle de calcul ;

4. Transaction : consiste à interagir avec le service sélectionné ;
5. Filtrage : à la fin d'utilisation d'un service donné, l'utilisateur sera amené à donner son avis sur le service et à lui attribuer une note. Cette valeur sera d'abord évaluée par le modèle d'attaque afin de garantir sa fiabilité avant de l'intégrer au système ;
6. Récompense et Punition : selon la satisfaction obtenue, une dernière étape consiste à récompenser ou à punir les utilisateurs, allant jusqu'à bannir les utilisateurs malhonnêtes.

3.3.4.1.Formulation du problème de sélection

Considérant un ensemble F = {F1, F2, …, Fm} de m fournisseurs offrant q services Cloud S = {S1, S2, …, Sq} et un ensemble C = {C1, C2, …, Cn} de n Consommateurs. Les consommateurs ont besoin de choisir parmi les services les plus appropriés des fournisseurs les plus fiables. Chaque service étant associé à une valeur de confiance globale. G = {G1, …. Gq}. Pour chaque service, le consommateur peut choisir parmi L services de mêmes types :

$L = \{S1, S2, \ldots, Sl\} / L \subseteq S$ et $l<=q$

Le problème étant de trouver le service le plus fiable R parmi l'ensemble des fournisseurs offrant ce type de service pour un utilisateur donné et avec une valeur de confiance maximale.

$R = Si / Gi = \max \{G\} / Si \in L$

3.3.4.2.Processus de sélection

Le processus de sélection implémenté dans le système de gestion de confiance que nous avons proposé est constitué de six étapes :

Etape 1 : (initialisation des valeurs de performance)

Les valeurs initiales de performances représentent la QoS fournie par les fournisseurs lors de l'enregistrement des services.

Elles sont calculées en fonction des informations d'identification des ressources du fournisseur et validées à partir du contrat SLA.

Etape 2 : (initialisation des valeurs de confiance)

Les valeurs de confiance initiales sont calculées à partir du modèle d'opinion présenté dans la section 3.1.1.1.

Etape 3 : (calcul des valeurs de performance)

À la fin de chaque transaction, une valeur de performance est calculée pour chaque utilisation de service.

Le consommateur Cloud potentiel qui cherche à sélectionner un service donne un poids pour chaque attribut de performance qui sera converti en une valeur normalisée. À partir de ces préférences la valeur de performance est calculée selon la fonction objective de l'équation 3.14 présentée dans la partie 3.3.3.1. Cette valeur est ensuite modélisée sous forme d'assertion positive, négative ou neutre et utilisée dans le modèle d'opinion suivant :

$$O(P) = E(r, c, f) \quad (3.22)$$

$$r = \begin{cases} 0.5 & if\ p + n = 0 \\ \frac{p}{p+n} \end{cases}$$

$$c = \frac{N * (p + n)}{2 * NA + N * (p + n)}$$

$$f = P_0(x)$$

Avec

r : moyenne des notes ;

c : certitude ;

f : attente initiale ;

p : avis positif ;

n : avis négatif ;

N : nombre total d'évidences ;

NA : évidences neutres ;

$P_0(x)$: valeur de performance précédente.

Le modèle d'opinion correspond au modèle de la logique certaine présenté précédemment.

Cependant, nous avons procédé à un certain changement pour ce modèle par rapport à l'attente initiale f. Dans la logique certaine, cette valeur est initialisée par 0.99 qui corresponde à une attente initiale maximale, mais nous avons jugé que cette valeur ne reflète pas la réalité pour un modèle de confiance. À cet effet, nous avons proposé de l'initialiser avec $P_0(x)$ qui est la valeur de performance initiale (QoS du fournisseur) si le consommateur n'a encore jamais utilisé le service sinon elle est égale à la dernière valeur de performance calculée en date.

Ainsi, après chaque transaction, nous recueillons les valeurs de performance pour l'utilisateur. Ensuite, nous appliquons le modèle d'opinion pour le nombre total d'évidences. Une évidence étant positive si elle est supérieure à la moyenne des valeurs de performance, elle est neutre si elle est égale à la moyenne des valeurs de performance et elle est négative si elle est inférieure. La moyenne des notes r est calculée comme la probabilité d'avoir des évidences positives par rapport au total d'évidences négatives et positives.

La certitude *c* est calculée en fonction du nombre total d'évidences N et le nombre d'évidences positives et négatives. La certitude c est égal à 1 lorsque toutes les assertions sont « positives » ou « négatives » et égales à 1 s'il n'y a pas de réponse.

Etape 4 : (filtrage des avis de satisfaction biaisés pour les valeurs de confiance)

Pour le calcul de confiance, les fonctions de satisfaction dépendent de l'avis de l'utilisateur cependant cet avis peut être biaisé, malhonnête ou même exagéré. Pour cela, nous avons proposé un modèle pour filtrer les avis malhonnêtes. Si l'avis est considéré comme biaisé, cet avis est ignoré et nous le remplaçons par la moyenne des avis des autres utilisateurs honnêtes.

Etape 5 : (calcul des valeurs de confiance)

De la même façon que pour le modèle de performance, les valeurs de confiance d'un utilisateur sont calculées après chaque transaction. Ensuite, le modèle d'opinion est appliqué aux évidences résultantes de ces valeurs. Le modèle de confiance est comme suit :

$$O(T) = E(r, c, f) \qquad (3.23)$$

$$r = \begin{cases} 0.5 & if\ p + n = 0 \\ \frac{p}{p+n} \end{cases}$$

$$c = \frac{N*(p+n)}{2*NA + N*(p+n)}$$

$$f = T_0(x)$$

$T_0(x)$: valeur de confiance précédente.

De la même manière que pour le modèle de performance, nous avons modifié l'attente initiale de façon qu'elle soit égale à la valeur de confiance calculée pour la transaction précédente et non pas 0.99 comme le modèle de logique certaine.

Etape 6 : (détermination des valeurs de confiance globale) :

Le modèle de calcul de valeur globale utilise l'opinion pour représenter les valeurs de performance et les valeurs de confiance. La valeur finale est calculée par un opérateur de logique subjective.

Ainsi, la valeur finale est calculée en terme de :

- opinion de performance ;
- opinion de confiance.

Ces deux valeurs sont agrégées selon l'opérateur de consensus ($\otimes$) de la logique subjective. Et les valeurs des différents services sont classifiées pour la sélection du service le plus fiable.

$$R = O(O(P(x) \otimes O(T(x))) \quad (3.24)$$

3.3.4.3.Algorithmes de sélection

Lorsqu'un utilisateur x recherche le service le plus fiable i, il fournit au système un ensemble de préférences de qualité requises Q = {qj, fj}.

qj: ID de l'attribut de qualité appartenant à l'ensemble de performance (puissance, temps de réponse, efficacité, transparence, interopérabilité, fiabilité, disponibilité, sécurité). Chaque attribut est calculé selon l'indice de mesure de service SMI.

fj: préférence d'un utilisateur, qui représente le niveau d'intérêt de l'utilisateur en l'attribut j avec fj ∈ [1 – 10].

Ces valeurs sont utilisées pour calculer les valeurs de performance et de confiance du service. Lorsqu'un utilisateur effectue une transaction, il donne un ensemble de valeurs de satisfaction représentant son avis sur le service utilisé S = {S1, S2, …, Sn} avec Si ∈ [1 – 10]. Les détails pour le calcul sont présentés dans les algorithmes suivants :

Algorithme 1 : décrit le processus de sélection de service pour un utilisateur x.
Algorithme 2 : décrit les étapes de calcul de la valeur de confiance globale d'un service pour un utilisateur x.
Algorithme 3 : décrit le calcul de l'opinion.
Algorithme 4 : définit le processus après chaque transaction et la mise à jour des valeurs.
Algorithme 5 : définit le calcul des valeurs de performance.
Algorithme 6 : définit le calcul de valeur de confiance.

Algorithme 3-1. service_selection (utilisateur x)

Début
Donner les attributs de préférences P = { q_j , p_j }
//SC : liste de Services Cloud
Pour $SC_i \in SC$ **faire**
CG(i) ← calcul_confiance_globale(x , i)
Fin pour
Retourner la liste SC triée en ordre décroissant par CG(i)
Fin

Algorithme 3-2. calcul_confiance_globale (utilisateur x , service i)

Début
// Evidences positives, négatives et neutres pour la performance.
$p_P \leftarrow 0$; $n_P \leftarrow 0$; $na_P \leftarrow 0$;
// Evidences positives, négatives et neutres pour la confiance.
$p_C \leftarrow 0$; $n_C \leftarrow 0$; $na_C \leftarrow 0$;
Pour Transaction u du service i **faire**
Pour utilisateur y ≠ x **faire**
// $P_{yu}(i)$: Performance de la transaction u pour l'utilisateur y.
Lire $P_{yu}(i)$;
Si ($P_{yu}(i) = \overline{\overline{P(\iota)}}$) **alors**
// Incrémenter les évidences neutres.
$na_P \leftarrow na_P + 1$;
Sinon Si ($P_{yu}(i) > \overline{\overline{P(\iota)}}$) **alors**
// Incrémenter les évidences positives.
$p_P \leftarrow p_P + 1$;
Sinon
// Incrémenter les évidences négatives.
$n_P \leftarrow n_P + 1$;
Fin Si
// $C_{yu}(i)$: Confiance de la transaction u pour l'utilisateur y.
Lire $C_{yu}(i)$;
Si ($C_{yu}(i) = \overline{\overline{C(\iota)}}$) **alors**
// Incrémenter les évidences neutres.
$na_C \leftarrow na_C + 1$;
Sinon Si ($P_{yu}(i) > \overline{\overline{C(\iota)}}$) **alors**
// Incrémenter les évidences positives.
$p_C \leftarrow p_C + 1$;
Sinon
// Incrémenter les évidences négatives.
$n_C \leftarrow n_C + 1$;
Fin Si
Fin Pour
Fin Pour
// Calculer le nombre d'évidences pour le consensus d'opinion
$p \leftarrow p_P + p_C$;
$n \leftarrow n_P + n_C$;
$na \leftarrow na_P + na_C$;
// n_{ux} : nombre total de transactions du service i pour l'utilisateur x
Lire n_{ux}
Si ($n_{ux} = 0$) **alors**
// 1ère transaction
$P \leftarrow QoS(i)$;
$C \leftarrow 0.5$;
Sinon
$P \leftarrow P_x(i)$;
$C \leftarrow C_x(i)$;
Fin Si
$G \leftarrow$ Opinion (p , n , na , P , C) ;
Retourner G ;
Fin

Algorithme 3-3 opinion (positives p , négatives n , neutres na , performance P , confiance C)

Début
N ← p + n + na ;
// Calculer la note moyenne
Si (p + n = 0) **alors**
r ← 0.5 ;
Sinon
$r \leftarrow \frac{p}{p+n}$;
Fin Si
// Calculer la certitude
$c \leftarrow \frac{N*(p+n)}{2*na+N*(p+n)}$;
// Calculer l'attente initiale
$f \leftarrow \frac{P+C}{2}$;
// Calculer l'opinion
E ← r c + (1 – c) f ;
Retourner E ;
Fin

Algorithme 3-4 mise_a_jour_apres_transaction (utilisateur x , service i , transaction u)

Début
// Lire les notes de satisfaction sur le service S = {S_1 , S_2 , … , S_n}
Lire S ;
Pour chaque attribut de performance de i de x faire
Mesurer et sauvegarder les valeurs délivrées de u (*puissance, stockage, bande passante, …*)
Fin Pour
P ← calcul_performance(x , i) ;
// Mettre à jour le nombre total de transaction pour l'utilisateur x sur le service i
n ← n + 1 ;
// Durée de la transaction
t_u ← temps (x , i , u);
// Coûts de la transaction
c_u ← cout(x , i , u) ;
c_{cpu} ← cout_cpu(x , i , u) ;
c_{vm} ← cout_vm(x , i , u) ;
c_{ram} ← cout_ram(x , i , u) ;
$c_{réseau}$ ← cout_reseau(x , i , u) ;
$c_{stockage}$ ← cout_stockage(x , i , u) ;
C ← caclul_confiance(x , i) ;
Fin

Algorithme 3-5 calcul_performance(utilisateur x , service i)

Début
// Q : liste des attributs j de qualité de service requis
// $Q_j = \{q_j , f_j\}$ avec q_j : valeur de la qualité , f_j ; valeur de préférence
Lire Q ;
Pour transaction u du service i faire
$P_x(i) \leftarrow 0$;
// $j \in \{ 1 , \dots , 9\}$
Pour attribut a_j de i faire
Calculer la valeur de a_j ;
// Calculer le poids de a_j selon les préférences définis par l'utilisateur
$w_j \leftarrow \frac{f_j}{\sum f_j}$;
// Calculer la valeur de performance de l'attribut a_j
$P_{xj}(i) \leftarrow a_j * w_j$;
// Calculer la performance globale pour tous les attributs
$P_x(i) \leftarrow P_x(i) + P_{xj}(i)$;
Fin Pour
Fin Pour
Retourner $P_x(i)$;
Fin

Algorithme 3-6 calcul_confiance(utilisateur x , service i)

Début

// n : nombre total de transaction du service i pour l'utilisateur x

Si (n = 0) **alors**

// 1ère transaction

$C_x(i) \leftarrow 0.5$;

Sinon

$C_x(i) \leftarrow 0$;

Pour transaction u de i pout x **faire**

// t : temps des transactions, t = {t_0 , t_1 , … , t_n}

t ← temps(x , i , u) ;

// Calculer le temps total de l'utilisation du service i pour l'utilisateur x

$TT \leftarrow \sum t_j$

//Calculer le facteur d'atténuation

$ft_{ux}(i) \leftarrow \frac{e^{-(t_0 - t_{n-1})}}{TT}$

// F : préférences de coûts, F = {f_{cpu} , f_{vm} , f_{ram} , f_{reseau} , $f_{stockage}$}

F ← preferences_couts(x , i , u) ;

Pour préférence de coût $f_i \in F$ / $i \in \{1 , \dots , 5\}$ **faire**

// Calculer le poids de préférences

$w_i \leftarrow \frac{f_i}{\sum f_i}$;

Fin Pour

// C : les différents coûts de la transaction u

// C = { c_i , c_{cpu} , c_{vm} , c_{ram} , c_{reseau} , $c_{stockage}$}

C ← cout(x , i , u) ;

// Calculer la facteur de coût avec $w_1 + w_2 + w_3 + w_4 + w_5 = 1$

$fC_{ux}(i) \leftarrow \frac{c_i}{c_{cpu}^{w_1} * c_{ram}^{w_2} * c_{vm}^{w_3} * c_{reseau}^{w_4} * c_{stockage}^{w_5}}$

// S : les notes de satisfaction sur le service i de l'utilisateur x

// S = {S_{x1} , … , S_{xn}} / $S_{xn} \in [1 – 10]$

S ← satisfaction(x , i , u);

// Calculer le facteur de satisfaction

$FS_{ux}(i)$ ← calcul_satisfaction(x , i , S) ;

// Calculer la confiance globale pour la transaction u

$C_{ux}(i)$ ← ft(i) * fC(i) * fS(i) ;

// Calculer la somme des confiances de transactions u

$C_x(i) \leftarrow C_x(i) + C_{ux}(i)$;

Fin Pour

// Calculer le facteur d'ajustement

$\alpha \leftarrow \sqrt{\frac{n}{TT+1}}$;

// Calculer la confiance globale

$C_x(i) \leftarrow \frac{\alpha C_x(i)}{n}$;

Fin Si

Retourner $C_x(i)$;

Fin

3.4. Modèle de menaces

Cette partie décrit les menaces associées à la gestion de la confiance, particulièrement dans un environnement de Cloud Computing. Ensuite, nous proposons un modèle pour catégoriser les utilisateurs selon quatre différentes classes de menaces sans aucune connaissance a priori, en utilisant des mesures et distances statistiques. Une fois les utilisateurs catégorisés, nous procédons au filtrage des utilisateurs biaisés afin de fiabiliser le calcul du degré de confiance.

3.4.1. Menaces pour un système du Cloud Computing

Plusieurs chercheurs ont discuté les menaces et attaques pour des systèmes de gestion de confiance (Audun Jøsang & Golbeck, 2009)(Tavakolifard & Almeroth, 2012)(Mármol & Pérez, 2009). Ces différentes menaces contre les systèmes de confiance présentent des caractéristiques de sécurité communes pour un environnement de Cloud Computing :

1) **Problème de départ (Cold-start),** représente le problème où de nouveaux services ou de nouvelles recommandations d'utilisateurs rencontrent des difficultés pour fournir des notes adéquates pour le système de confiance. Un service ne peut être recommandé à moins qu'il y ait un nombre suffisant de consommateurs. Comme les utilisateurs ont tendance à interagir avec les services les plus répétés et de ce fait, les plus notés, les chances pour qu'un nouveau service soit sélectionné sont généralement rarissimes. Une des méthodes proposées pour régler ce type de problème est l'utilisation de mesures statistiques. (Ahn, 2008)
2) **Feedbacks malhonnêtes,** représentent le problème où les utilisateurs déclarent de faux avis, rendant le système non crédible. Ces avis malhonnêtes peuvent être soit en rapport à un seul individu lorsqu'un un seul utilisateur fournit toujours des avis malhonnêtes pour les services ou bien en rapport à un groupe d'utilisateurs lorsque deux utilisateurs ou plus collaborent afin d'améliorer ou de dégrader la réputation d'un service. Ainsi, un système de gestion de confiance doit exclure ces avis malhonnêtes. Une solution générale serait d'utiliser des méthodes statistiques (Chrysanthos, 2000) (Chen & Singh, 2001).
3) **Rejeu (Playbook)**. un re-jeu consiste à maximiser le profit d'un service selon certains critères. Par exemple, un fournisseur peut travailler de façon honnête et fournir des services de qualité pour une certaine durée de temps afin de gagner une bonne réputation, ensuite avec ce niveau de réputation élevé, fournit des services de basse qualité (Kerr & Cohen,

2006). Ainsi, un système de gestion de confiance doit considérer la propriété d'auto-ajustement du service à travers le temps.

4) **Attaque Sybil**. Des utilisateurs malintentionnés peuvent acquérir plusieurs identités. Chaque fois qu'ils fournissent un faux avis, ils prennent une nouvelle identité. Pour cela, un bon système de confiance doit intégrer un service d'authentification fiable et sécurisé basé non seulement sur la sécurité mais aussi pouvant détecter les identités fantômes.

3.4.2. Vulnérabilités et risques

Cloud Security Alliance (Zhao, Rong, Jaatun, & Sandnes, 2010) ont identifié diverses menaces pour le Cloud Computing. les auteurs de (Bamiah & Brohi, 2011) ont classifié plusieurs vulnérabilités et attaques qui peuvent être rencontrées dans un environnement Cloud à différents niveaux de sécurité tels que :

- Utilisation abusive et néfaste du Cloud ;
- Interfaces et APIs non sécurisées ;
- Entités internes malhonnêtes ;
- Technologie virtualisée.

Cependant, ces menaces représentent divers aspects de la sécurité, ce qui recouvre un large domaine. Pour assurer l'implémentation correcte d'un système de gestion de confiance, des menaces spécifiques doivent être considérées. Ainsi, nous nous concentrons ici sur les attaques de comportement en rapport avec la gestion de confiance pour le Cloud Computing.

Pour un système de gestion de confiance, une vulnérabilité est une faiblesse dans le système qui pourrait être considéré afin d'influencer la recommandation et la confiance des services proposés.

Le Cloud Computing est tout aussi vulnérable comme n'importe quelle technologie utilisant l'Internet. Ses vulnérabilités incluent :

- Attaques malhonnêtes ;
- Attaque man-in-the-middle ;
- Attaque Sybil ;
- Attaque déni de service.

Plus particulièrement, on peut rencontrer des attaques contre les systèmes de confiance telles que (Sun, Han, Yu, & Liu, 2006) (Audun Jøsang & Golbeck, 2009) (Tavakolifard & Almeroth,

2012) (T. H. Noor, Sheng, & Alfazi, 2013) (Fan & Perros, 2013) (Thirunarayan, Anantharam, Henson, & Sheth, 2014) :

- Attaque de médisances ;
- Connivence d'avis ;
- Attaque de retard de réputation ;
- Attaque Sybil.

3.4.3. Travaux existants

Dans plusieurs recherches, il a été démontré que les utilisateurs ayant des avis malhonnêtes partagent des caractéristiques communes. Dans (Trevathan & Read, 2007), il a été identifié que les utilisateurs malhonnêtes ayant des caractéristiques similaires tels que un taux de demande élevé afin de dépasser ceux des utilisateurs honnêtes, puisque les utilisateurs qui essaient d'augmenter ou de réduire la popularité d'un service envoient leurs avis régulièrement. Ils ont également tendance à engager le minimum de coût dans une transaction, uniquement pour répondre aux exigences et envoyer un avis. De plus, les avis malhonnêtes sont en général soit très bas soit très élevés que la majorité des autres avis d'utilisateurs.

Il existe plusieurs approches pour évaluer la crédibilité des utilisateurs qui se base sur les opinions des utilisateurs tels que le filtrage béta des avis. Cette approche fonctionne tant que la majorité des utilisateurs ne sont pas d'un groupe d'utilisateurs conspirant ensemble afin de modifier leurs notes.

Une autre approche (Audun Jøsang & Quattrociocchi, 2009) consiste à utiliser la fonction béta de densité de probabilité pour estimer la réputation d'un fournisseur. Cette approche a été étendue ensuite de façon à considérer un avis comme honnête s'il est inclus dans l'intervalle de tous les avis (Audun Jøsang & Golbeck, 2009). La limite de cette stratégie est que les utilisateurs peuvent comploter en tant que groupe pour manipuler la majorité des avis. Donc, nous pouvons déduire qu'une approche qui se base sur la majorité des avis n'est pas suffisante pour évaluer les différents avis utilisateurs. Les auteurs de (Yu & Singh, 2003) ont proposé un modèle basé sur l'hypothèse que tous les clients dans le système fournissent des évaluations pour une période de temps donnée. Par exemple, les nouveaux utilisateurs pourraient être traités comme des utilisateurs malhonnêtes et leurs avis auront moins de poids dans l'évaluation de la confiance.

La plupart de ces approches sont basées sur la technique de filtrage par similarité tels que (Audun Jøsang & Golbeck, 2009) (Whitby et al., 2005). Pour ces techniques, les utilisateurs

ayant une faible note de similarité sont considérés comme moins fiables. Ces approches se sont révélées très efficaces en terme de filtrage des utilisateurs malveillants. Par conséquent, nous allons utiliser une technique basée sur la similarité afin de procéder à une première phase qui consistera à prédire les valeurs manquantes pour les notes des services.

3.4.4. Modèle proposé

Le modèle proposé consiste à filtrer les utilisateurs malhonnêtes pour traiter les problèmes discutés précédemment. Pour cela, nous proposons de grouper les utilisateurs selon la fiabilité de leur avis en utilisant la méthode d'apprentissage de k-means. Le problème avec cette méthode est les valeurs manquantes pour de multiples utilisateurs. Ainsi, avant de procéder à la classification des groupes, nous traitons le problème de départ comme il a été souligné par (Mármol, Sorge, Ugus, & Pérez, 2013) et ceci en calculant la similarité et distance entre les différents utilisateurs par des méthodes statistiques afin de remplacer les valeurs manquantes. Enfin, nous intégrons un service pour récompenser les utilisateurs honnêtes et punir les utilisateurs malhonnêtes.

3.4.4.1.Classes d'utilisateurs

À partir des attaques de vulnérabilités et risques discutés, nous remarquons que généralement quatre groupes d'utilisateurs peuvent être catégorisés, que ce soit pour un utilisateur ou un fournisseur de services. Ainsi, nous proposons de classifier les utilisateurs en quatre groupes :

1) Positive Honnête (PH), lorsqu'une entité fournit un avis honnête sur un service de haute qualité ;
2) Négative Honnête (NH), lorsqu'une entité fournit un avis honnête sur un service de basse qualité ;
3) Positive Malhonnête (PM), lorsqu'une entité fournit un avis malhonnête sur un service de haute qualité (fournit des avis déraisonnablement augmentés) ;
4) Négative Malhonnête (NM), lorsqu'une entité fournit un avis malhonnête sur un service de basse qualité (fournit des avis faibles).

Nous observons aussi qu'un des principaux intérêts d'un système de gestion de confiance est d'identifier correctement les avis malhonnêtes. Pour cela, nous devons identifier dans quel groupe l'utilisateur est classifié.

3.4.4.2.Les mesures et paramètres du modèle

Les utilisateurs des divers services peuvent fournir leurs avis pour exprimer leur satisfaction ou insatisfaction sur le service utilisé. En se basant sur ces avis, le système refait l'évaluation des taux de confiance. Cependant, plusieurs utilisateurs peuvent fournir des avis malhonnêtes alors il est primordial de détecter de tels avis pour fiabiliser le système.

Le filtrage des avis est un composant important pour résoudre de telles menaces. Plusieurs techniques pour détecter les avis malhonnêtes ont été présentées à travers diverses études (Tavakolifard & Almeroth, 2012)(Trevathan & Read, 2007)(Kerr & Cohen, 2009)(Whitby et al., 2005). Bien qu'il existe plusieurs algorithmes, l'idée de base est d'utiliser différentes mesures de similarité ou de distance entre les utilisateurs.

Il existe plusieurs algorithmes de similarités (Lee, Jun, Lee, & Kim, 2005) (Gong, 2010) qui ont été utilisés pour le domaine du filtrage tel que la corrélation de Pearson, similarité du vecteur de cosinus, distance euclidienne et distance de Minkowski. Ces mesures peuvent être utilisées efficacement afin d'équilibrer l'algorithme de prédiction des notes manquantes et accroître ainsi leur exactitude.

1. Corrélation de Pearson

La corrélation de Pearson mesure la corrélation linéaire entre deux séquences de notes pour l'utilisateur x et l'utilisateur y à propos de l'ensemble des services m notés à la fois par l'utilisateur x et y.

$$Sim\_Pearson(x,y) = \frac{\sum_{i=1}^{m}(R_{ix} - \overline{R_x})(R_{iy} - \overline{R_y})}{\sqrt{\sum_{i=1}^{m}(R_{ix} - \overline{R_x})^2 \sum_{i=1}^{m}(R_{iy} - \overline{R_y})^2}} \quad (3.25)$$

Où R_{ix} représente les notes du service i par l'utilisateur x, $\overline{R_x}$ est la moyenne des notes de l'utilisateur x et R_{iy} représente les notes du service i par l'utilisateur y, $\overline{R_y}$ est la moyenne des notes de l'utilisateur y.

2. Mesure de Cosinus

La mesure du cosinus s'intéresse à l'angle entre deux séquences de notes où une plus grande similitude implique un angle plus petit, tel que l'indique la formule suivante :

$$Sim\_Cosine(x,y) = \frac{\sum_{i=1}^{m} R_{ix}\, R_{iy}}{\sqrt{\sum_{i=1}^{m} {R_{ix}}^2 \sum_{i=1}^{m} {R_{iy}}^2}} \quad (3.26)$$

Où R_{ix} représente les notes du service i par l'utilisateur x, $\overline{R_x}$ est la moyenne des notes de l'utilisateur x et R_{iy} représente les notes du service i par l'utilisateur y, $\overline{R_y}$ est la moyenne des notes de l'utilisateur y.

3. Distance Euclidienne

Une distance euclidienne représente la distance entre deux points dans un espace euclidien. Nous allons utiliser cette formule pour mesurer la distance entre deux séquences de notes pour deux utilisateurs x et y sur m services notés par les deux utilisateurs.

$$Dis\_Euclidean(x,y) = \sqrt{\sum_{i=1}^{m} (R_{ix} - R_{iy})^2} \quad (3.27)$$

Où R_{ix} est la note du service I par l'utilisateur x et R_{iy} est la note du service i par l'utilisateur y.

4. Distance de Minkowski

La distance de Minkowski peut être considérée comme une généralisation de la distance euclidienne et la distance de Manhattan. Nous allons utiliser cette métrique pour mesurer la distance entre deux séquences de notes.

$$Dis\_Minkowski(x,y) = \left(\sum_{i=1}^{m} (R_{ix} - R_{iy})^p\right)^{1/p} \quad (3.28)$$

Où R_{ix} représente les notes du service i par l'utilisateur x, and R_{iy} représente les notes du service i par l'utilisateur y.

5. Distance de Hamming:

La Distance de Hamming est utilisée en théorie de l'information pour mesurer la différence entre deux ensembles de chaînes ou deux séquences de bits. En d'autres termes, elle mesure le nombre minimum d'erreurs qui auraient pu transformer une chaîne en une autre.

Nous allons utiliser cette distance pour mesurer la différence entre deux séquences de notes, comme décrit dans la formule suivante :

$$Distance\_Hamming(x,y) = \sum_{i=1}^{m} [R_{ix} \neq R_{iy}] \quad (3.29)$$

Dans l'équation Dis_hamming (x, y) représente la distance de Hamming entre l'utilisateur x et l'utilisateur y, i est l'indice du service pour un total de n services. La distance de Hamming donne l'écart entre le classement de l'utilisateur x et l'utilisateur y.

3.4.4.3.Solution proposée

Pour évaluer un degré de confiance fiable, il est important de filtrer les notes injustes des différents services. La littérature démontre que les méthodes de filtrage fournissent un moyen fructueux pour y arriver. Cependant, la majorité de ces méthodes sont limitées par les problèmes de départ.

Dans cette solution, nous proposons de classifier les utilisateurs en quatre groupes. Pour cela, nous allons utiliser l'algorithme de regroupement k-means (Gu, Zhou, & Chen, 2009), pour former ces groupes en se basant sur leurs avis.

Cependant, les techniques d'apprentissage supposent la présence de connaissances complètes sur les notes d'utilisateurs pour tous les services. Malheureusement, dans un environnement de Cloud Computing et en particulier pour un système de gestion de confiance, ces suppositions ne sont pas adéquates pour avoir des résultats précis. Beaucoup de travaux de recherche traitent ce problème en remplaçant les données manquantes aléatoirement, ce qui engendrera un estimateur inexact et partiel.

Pour cette raison, nous proposons de prétraiter l'ensemble des notes avant de procéder à la classification et puisque les techniques statistiques offrent de meilleurs résultats, nous proposons une technique hybride basée sur les différentes mesures discutées dans la section 3.4.4.2.

Nous employons le principe du voisinage pour aider à identifier les classes d'utilisateurs.

Enfin, les classes qui en résultent sont soient récompensées pour les groupes honnêtes ou pénalisées pour les groupes malhonnêtes.

3.4.4.4.Algorithme de détection d'attaque

Nous considérons un ensemble d'utilisateurs avec des notes honnêtes positives, honnêtes négatives, malhonnêtes positives et malhonnêtes négatives. Pour chaque utilisateur i, nous avons une séquence de notes Ni sur les différents services.

L'algorithme de classification implique que l'utilisateur doit noter tous les services utilisés. Chaque utilisateur est représenté par la note (Rij) dans la séquence des listes d'utilisateurs qui contient les notes données par un utilisateur i pour le service j, tel que décrit dans le tableau 3.1.

Service / ***Utilisateur***	**S_1**	**S_2**	**…**	**S_m**
U_1	R_{11}	R_{12}	…	R_{1m}
U_2	R_{21}	R_{22}	…	R_{2m}
…	…	…	…	…
U_n	R_{n1}	R_{n1}	…	R_{nm}

Tableau 3.1. Matrice des notes utilisateurs

Comme évoqué précédemment, dans un environnement réel, il est impossible pour un utilisateur d'utiliser chaque service proposé. Par conséquent, le tableau présenté contiendra de nombreuses valeurs manquantes des différents services, ce qui se traduira par une classification non précise ou biaisée.

Pour cette raison, nous procédons à une phase de prétraitement où nous prédisons les valeurs manquantes en utilisant les mesures statistiques abordées dans la section 3.4.4.2.

Les valeurs de la matrice résultante sont ensuite classifiées en quatre groupes en utilisant l'algorithme des k-means. Nous employons le principe de voisinage pour aider à identifier les classes d'utilisateurs. Nous supposons que l'ensemble des notes dont la moyenne est plus proche de la moyenne des notes des voisins sont des utilisateurs honnêtes. Parmi ces quatre ensembles, l'ensemble de note avec la moindre moyenne type sont les utilisateurs malhonnêtes.

Nous remarquons que cette solution est proposée dans le cadre où l'avis de l'utilisateur dépend de la qualité du service fourni.

Etape 1 : Sélectionner les voisins de chaque utilisateur

Algorithme 3-7 Sélection des voisins

```
fonction selection_voisins (utilisateur x)
{
    pour utilisateur y ≠ x
        d ← Dis_Hamming(x,y)        //Formule 3.29
        si (d < 0.5)
            ajouter y liste_voisins
        finsi
    finpour
    retourner liste_voisins
}
```

Etape 2 : Calculer les similarités entre les utilisateurs

Algorithme 3-8 Calcul de similarité

```
fonction get_utilisateurs_similaires (utilisateur x)
{
    pour utilisateur y ≠ x
        s1 ← Sim_Pearson(x,y)        //Formule 3.25
        s2 ← Sim_Cosine(x,y)         //Formule 3.26
        d1 ← Dis_Euclidean(x,y)      //Formule 3.27
        d2 ← Dis_Minkowski(x,y)      //Formule 3.28
        si (s1 >= 0.5) ET (s2 >= 0.5) ET (d1 <= 0.5) ET (d2 <= 0.5)
            ajouter y à liste_utilisateurs_similaires
        finsi
    finpour
    retourner liste_utilisateurs_similaires
}
```

Etape 3 : Calculer les valeurs manquantes pour chaque utilisateur

Algorithme 3-9 Calcul des valeurs manquantes

fonction *predire_notes_manquantes* (utilisateur *x*)

{

for chaque service i non note par x

p ← taille (*get_utilisateurs_similaires* (*x*))

$R_{xi} \leftarrow \frac{\sum_{k=1}^{p} Rki}{p}$ //R_{ki} : notes de l'utilisateur k pour le service i

//avec k ϵ utilisateurs_similaires(x)

endfor

}

Etape 4: Classifier les utilisateurs

Les valeurs de la matrice résultante des notes sont classifiées en quatre groupes en utilisant l'algorithme de classification k-means : PH, NH, PM, NM.

3.4.4.5.Autres propriétés

Suite au modèle proposé, d'autres propriétés ont été intégrées au système de gestion de confiance pour renforcer la robustesse du système contre les attaques discutées.

Auto-ajustement de la classification : L'auto-ajustement est une propriété importante où les groupes d'utilisateurs sont établis dynamiquement pour refléter les interactions et notations récentes. La solution proposée permet de détecter les utilisateurs malhonnêtes, les changements des degrés de notes, ainsi que la reconstitution de nouveaux groupes d'utilisateurs et ceci en effectuant périodiquement le regroupement et la détection de nouveaux groupes d'utilisateurs.

Punition et Récompense : Selon les groupes résultants, une dernière étape de punition ou de récompense doit être effectuée. Ce mécanisme fonctionne en diminuant ou en augmentant, respectivement, les poids attribués à chaque source de notation, qui dépendra des groupes d'utilisateurs.

Pour les utilisateurs malveillants, après k punition, les notes de l'utilisateur devraient être bannies du processus de notation et de calcul de similitudes.

3.5. Conclusion

Ce chapitre a présenté la solution adoptée pour traiter la problématique rencontrée. Pour cela, nous avons proposé une plate-forme de gestion de confiance dans un environnement de Cloud Computing. Cette plate-forme consiste à sélectionner les fournisseurs les plus fiables en se basant sur deux modèles computationnels : le modèle de performance afin de traiter les paramètres objectifs des services Cloud et le modèle de confiance qui consiste à traiter les notations et recommandations des différents services. Nous avons aussi proposé un modèle de filtrage pour détecter les attaques ciblées contre les systèmes de gestion de confiance.

Le chapitre suivant présentera l'implémentation du système proposé, les différents cas de simulation et de validation du système et l'évaluation des résultats obtenus.

Chapitre 4 : Implémentation et Simulation

4.1. Implémentation

Après avoir détaillé les concepts de notre système de confiance pour la sélection de fournisseur dans un environnement Cloud et de son architecture et avoir proposé un modèle computationnel pour le calcul de confiance ainsi qu'un modèle d'attaque pour filtrer les utilisateurs malhonnêtes, nous décrivons, au sein de ce chapitre son implémentation qui a été réalisée à l'aide de la plate-forme J2EE sous l'environnement Eclipse. En outre, ce chapitre détaille les différentes simulations et comparaisons menées pour valider le système proposé.

4.1.1. Environnement de travail

Nous allons présenter dans cette section, le langage utilisé, les différents outils utilisés pour l'implémentation du système de gestion de confiance, ainsi que l'environnement matériel choisit pour effectuer les analyses.

4.1.1.1.Langage de programmation

1) Java

Nous avons choisi JAVA pour implémenter le système de gestion de confiance. C'est un langage multi plateforme qui possède de nombreuses caractéristiques qui en font un des langages de choix pour la programmation. Il a été conçu pour mettre en œuvre des applications susceptibles de s'exécuter sur n'importe quelle plate-forme. C'est un langage fortement typé, il permet plusieurs niveaux de protection des données, ainsi qu'un mécanisme de traitement des exceptions. C'est un langage orienté objet.

2) JavaEE

Nous avons aussi décidé d'utiliser la version entreprise de JAVA, plus exactement la programmation web dynamique basée sur l'approche orientée service qui est Java Entreprise Edition (J2EE). L'objectif majeur de Java EE est de faciliter le développement d'applications web robustes et distribuées, déployées et exécutées sur un serveur d'application.

3) Les langages de script web

Dans cette partie, nous allons détailler tous les langages de programmation utilisés pour le développement du service conçu à savoir : HTML, CSS, Javascript, XML, servlet et JSP.

4.1.1.2.Outils de développement

Dans cette partie, nous présentons les outils utilisés pour le développement de la plateforme de gestion de confiance.

1) **Dreamweaver**

Adobe Dreamweaver est un éditeur HTML professionnel destiné à la conception, au codage et au développement de sites, de pages et d'applications Web.

Nous l'avons utilisé pour programmer les interfaces utilisateur et les interfaces du fournisseur du système de confiance permettant l'accès à la plateforme proposée.

2) **L'IDE Eclipse JavaEE**

Un environnement de développement intégré (EDI ou IDE - Integrated Development Environment) est un programme regroupant un éditeur de texte, un compilateur, des outils automatiques de création et souvent un débogueur. Pour le langage Java, il existe plusieurs EDI tels que NetBeans (de Sun), JBuilder (de Borland), JCreator ou Eclipse (d'IBM).

L'EDI que nous devons utiliser doit être extensible, universel, polyvalent et libre. Notre choix s'est porté sur ECLIPSE parce qu'il répond à tous les critères énumérés.

Nous avons utilisé l'IDE Eclipse JavaEE pour coder les différents composants et modules du système proposé, à savoir CGT, MP, MC, CCG, CP, CC, EC, etc…

4.1.1.3.Environnement matériel

Le développement du système proposé ainsi que les différentes simulations et analyses de performance effectuées ont été réalisés en utilisant l'ordinateur « Dell Inspiron 3737 » dont la configuration est comme suit :

- Processeur Intel Core i7-4500U à 3GHz;
- 8 Go de mémoire RAM ;
- 1000 Go d'espace disque dur.

4.1.2. Implémentation du système

Nous présentons, au sein de cette section, le système de gestion de confiance développé. Nous définissons le calculateur de confiance qui constitue le composant le plus important du système puisqu'il s'occupe du calcul de confiance et de la sélection de fournisseurs.

Dans ce qui suit, nous présentons les diagrammes de classes des principaux composants du gestionnaire de confiance décrits au chapitre précédent.

La figure 4.1 représente le digramme de classe du calculateur de confiance du composant CC décrit dans la section 3.2.3. Cette classe est constituée de trois attributs à savoir la certitude, la note moyenne et l'attente initiale. Elle définit plusieurs méthodes pour le calcul de l'opinion, la mise à jour de la valeur de confiance, le calcul de la valeur de confiance totale, etc…

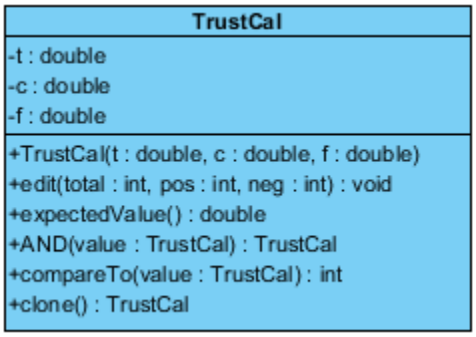

Figure.4.1. Diagramme de classe du composant calculateur de confiance

La figure 4.2 représente le digramme de classe du calculateur de performance du composant CP décrit dans la section 3.2.3. Cette classe est constituée des divers attributs de performance décrits à savoir la puissance, le coût, le temps de réponse, l'efficacité, la transparence, l'interopérabilité, la fiabilité, la disponibilité et la sécurité. Elle définit les différentes méthodes pour le calcul de chaque attribut ainsi que la méthode de calcul de la valeur de performance totale.

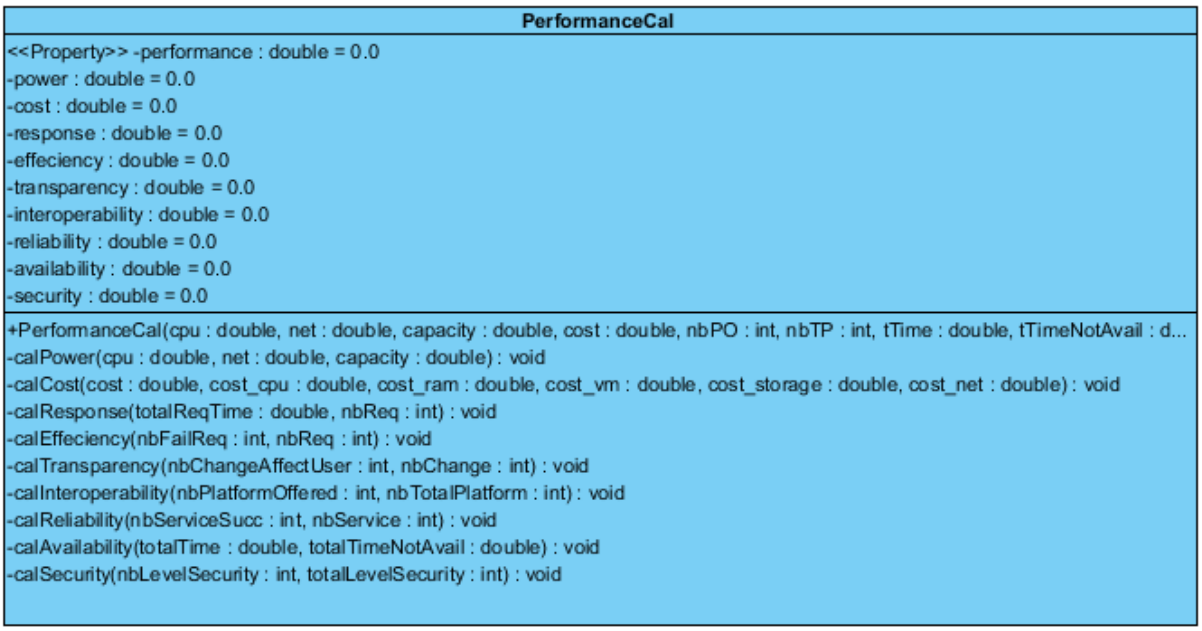

Figure.4.2. Diagramme de classe du composant calculateur de performance.

La figure 4.3 représente le digramme de classe du calculateur de confiance globale CCG décrit dans la section 3.2.3. Cette classe est constituée de plusieurs attributs tels que les objets contenant les valeurs de confiance et de performance. Elle définit plusieurs méthodes pour le calcul de confiance globale, le consensus et l'agrégation des valeurs d'opinions, la mise à jour des valeurs de confiance, etc…

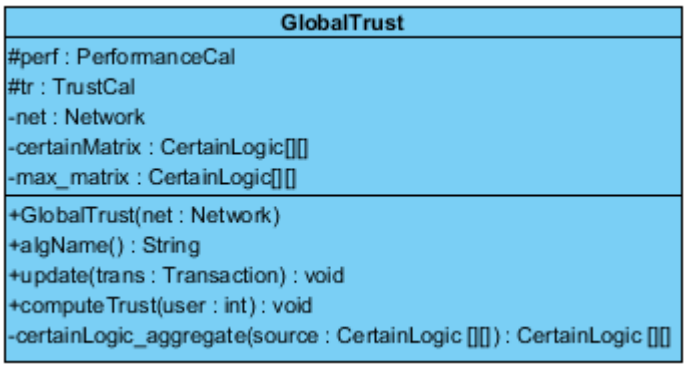

Figure.4.3. Diagramme de classe du calculateur de confiance globale

4.2. Simulation et analyses

Une simulation est une technique pour réaliser des expériences sur un système sans avoir recours à la création d'un véritable système. C'est une approche plus simple et efficace pour analyser et évaluer les mécanismes conçus, les protocoles ainsi que les algorithmes.

Sur la base de l'implémentation décrite dans la section précédente, nous avons réalisé une série de simulations sous un simulateur basé sur CloudSim afin de déterminer les valeurs appropriées des paramètres utilisés dans le modèle et conduire une évaluation des performances.

Cloud-Sim représente une boîte à outils de simulation développée par (Calheiros, Ranjan, Beloglazov, Cesar, & Buyya, 2011) pour créer un environnement de simulation pour le Cloud Computing. L'environnement de Cloud simulé contient diverses ressources pour intégrer la notion d'hétérogénéité. Chaque ressource a différentes caractéristiques et facteurs tels que la vitesse du processeur, la taille de la mémoire, la capacité de stockage, les valeurs de bande passante et de latence pour le réseau, etc… La simulation a été effectuée en utilisant la dernière version de cloudsim-3.0.3.

Dans cette section, nous allons discuter la fiabilité, la performance et la robustesse de notre protection vis-à-vis des attaques présentées antérieurement.

4.2.1. Evaluation et analyse du modèle de calcul

Dans ce qui suit, nous allons valider le modèle de calcul développé à travers les expérimentations présentées ci-dessous. Pour cela, nous allons comparer le modèle de confiance par rapport à deux approches connues dans la gestion de confiance : EigenTrust (Kamvar et al., 2003) et logique subjective (Audun Jøsang, Hayward, et al., 2006).

Les différents scénarios ont été testés avec les paramètres suivants :

Nombre d'utilisateurs	Nombre de fournisseurs	Nombre de Data Center/Fournisseur	Nombre de VM /Data Center	Nombre de services
100-1000	25	1-5	5-25	5-50

Tableau 4.1. Paramètres de Simulation

Le nombre d'utilisateurs représente les consommateurs de services Cloud. Ce nombre est compris entre 100 et 1000 afin de refléter la réalité dans un environnement de Cloud Computing. Cependant, ces valeurs ne sont pas significatives et peuvent être facilement réduites ou augmentées pour tester l'évolutivité du système.

Le nombre de fournisseurs a été délimité à 25. De la même façon, ce nombre peut être augmenté ou réduit ce qui va limiter le choix des services, mais ceci n'influence en rien la crédibilité du système.

Le nombre de DC a été défini entre 1 et 5 DC pour chaque fournisseur afin de refléter la réalité pour la plupart des fournisseurs Cloud.

Le nombre de VM a été défini entre 5 et 25 VM pour chaque DC. Ce nombre dépendra des performances du centre de données propre à chaque fournisseur.

Enfin, nous avons défini le nombre de services offerts pour chaque fournisseur entre 5 et 50 services.

Nous avons choisi ces différentes mesures suite à une étude des différents services réels de Cloud Computing proposés, puis nous avons pris les valeurs moyennes pour chaque paramètre.

Scénario 1

Pour cette expérimentation, 100 tours de simulation ont été réalisés. Pour chaque tour, de 100 à 150 demandes de divers types de services sont générées aléatoirement. Puis, trois méthodes pour sélectionner des fournisseurs de confiance sont utilisées :

1. Sélection avec Eigen : basé sur l'algorithme EigenTrust comme décrit par Hector Garcia-Molina (Kamvar et al., 2003) ;

2. Sélection avec Logique Subjective : basé sur la sélection par l'approche de logique subjective (Audun Jøsang, Gray, et al., 2006) ;

3. Solution proposée : procède à la sélection du service le plus fiable avec la plateforme développée.

Pour les différents tours de simulation, l'exactitude de la solution a été calculée. L'exactitude de chaque méthode est égale au nombre total de transaction réussie par le tour de simulation.

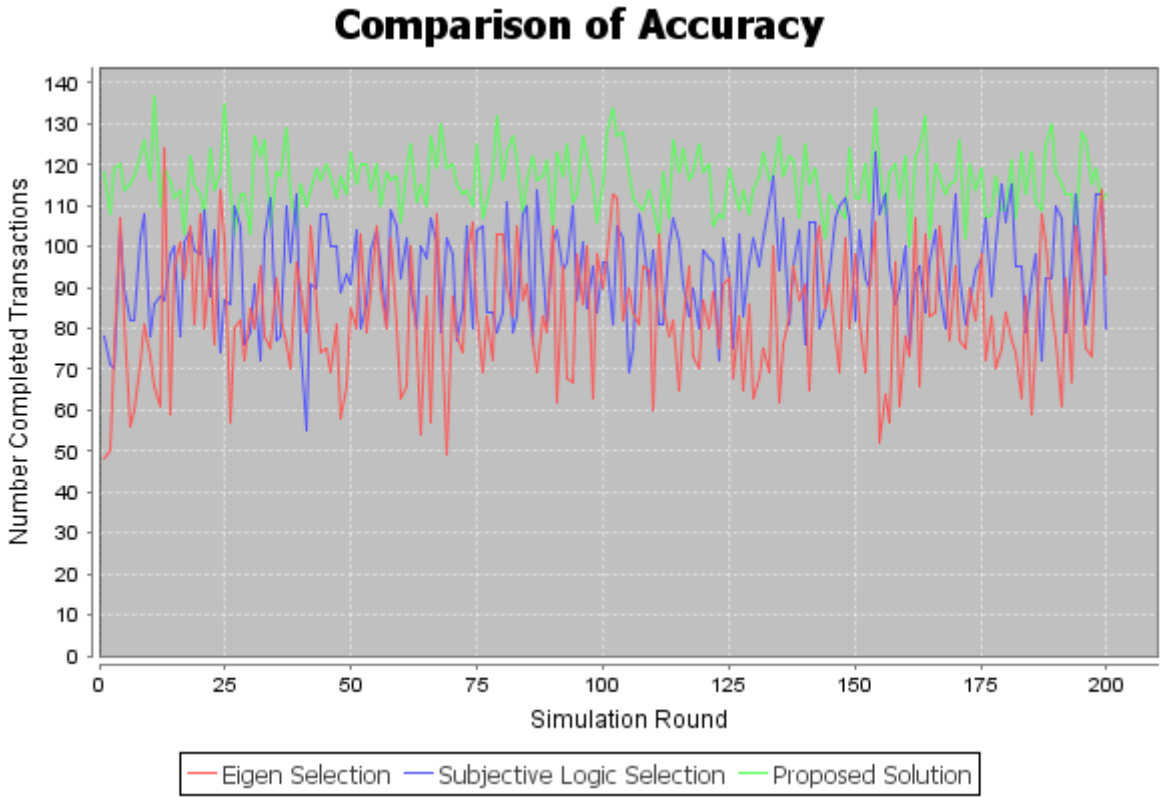

Figure 4.4. Exactitude

Interprétations

À partir des résultats présentés dans la figure 4.4, nous pouvons noter que notre modèle de confiance permet d'augmenter le nombre de transactions réussies. Au cours des 100 tours de la simulation, le nombre de transactions réussies avec la solution proposée augmente de manière significative et reste stable à un certain point, par rapport à l'algorithme Eigen et la logique subjective.

Cela revient au fait que la solution proposée est basée sur les notes d'utilisateurs mais aussi sur les valeurs de performance des services, ainsi le nombre de transactions réussies a un ratio d'exactitude plus élevé par rapport aux autres solutions.

À chaque tour de simulation, nous avons procédé au calcul de l'exactitude (nombre de transactions réussies). Nous considérons qu'une transaction est réussie si la valeur de confiance calculée pour chaque sélection est supérieure au seuil défini (c'est-à-dire la moyenne des valeurs de confiance, décrit dans l'algorithme 2). Pour la sélection avec Eigen, elle représente la valeur de confiance calculée par l'algorithme de (Kamvar et al., 2003), pour la sélection avec la logique subjective, elle représente la valeur de confiance calculée avec l'opinion. Pour notre solution, nous avons effectué le calcul des valeurs de confiance mais aussi les valeurs de performance, ce qui permet d'orienter les demandes des différents consommateurs vers les fournisseurs les plus appropriés, augmentant de ce fait la confiance du consommateur ainsi que l'exactitude de la transaction.

Scénario 2

Pour ce scénario, 250 tours de simulation ont été réalisés. Pour chaque tour, les demandes sont émises au hasard. Trois méthodes pour sélectionner des fournisseurs fiables sont utilisées:

1. Sélection aléatoire : procède à la sélection d'un fournisseur aléatoirement parmi les fournisseurs qui répondent à toutes les exigences fonctionnelles de contrat SLA ;

2. Sélection basée sur la performance : permet de sélectionner le service qui convient le mieux à toutes les exigences fonctionnelles de service SLA ;

3. Solution proposée : permet de sélectionner un service selon le modèle proposé.

Pour chaque méthode, le temps d'exécution est calculé afin de valider la performance du modèle proposé.

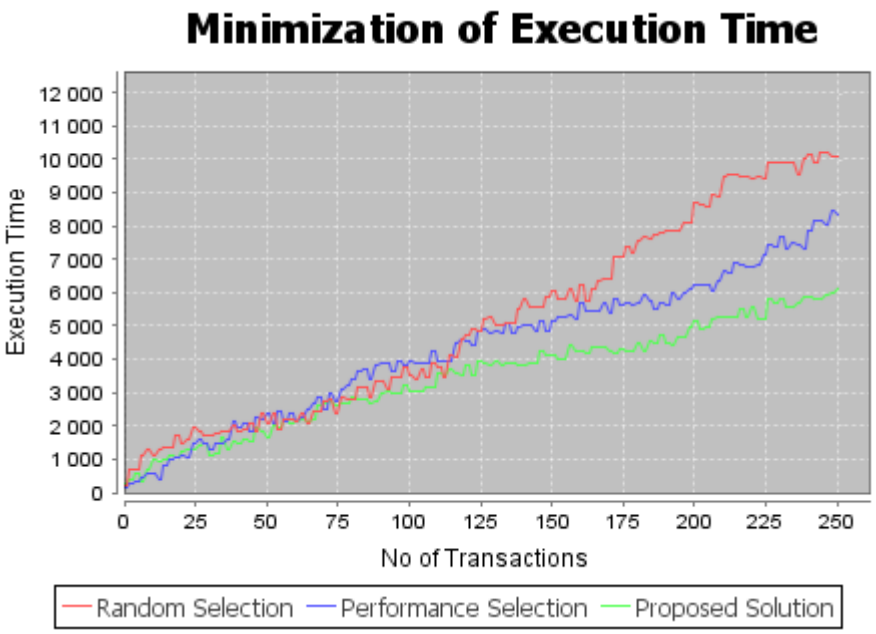

Figure 4.5. Temps d'exécution

Interprétations

Les résultats de l'expérimentation sont présentés sur la figure 4.5. Compte tenu d'un certain nombre de transactions, toutes les propositions présentent un temps d'exécution similaire. Cependant, avec la surcharge de demande de services, le temps d'exécution de la sélection aléatoire et la sélection basée performance augmente où la solution proposée maintiens une certaine stabilité. La raison majeure pour laquelle la solution présente de meilleurs résultats est que la solution proposée effectue la sélection du fournisseur en se basant non seulement sur la qualité de service, mais aussi sur l'ensemble des notes et avis des utilisateurs, ce qui augmente la fiabilité du service fourni et minimise ainsi le temps d'exécution.

Scénario 3

Pour ce scénario, nous avons validé le modèle de calcul en se basant sur la mesure de la satisfaction d'utilisateur. La satisfaction de l'utilisateur représente le nombre de requêtes servies par rapport au temps d'exécution. L'étude comparative a été faite par rapport à la solution proposée et le modèle de performance.

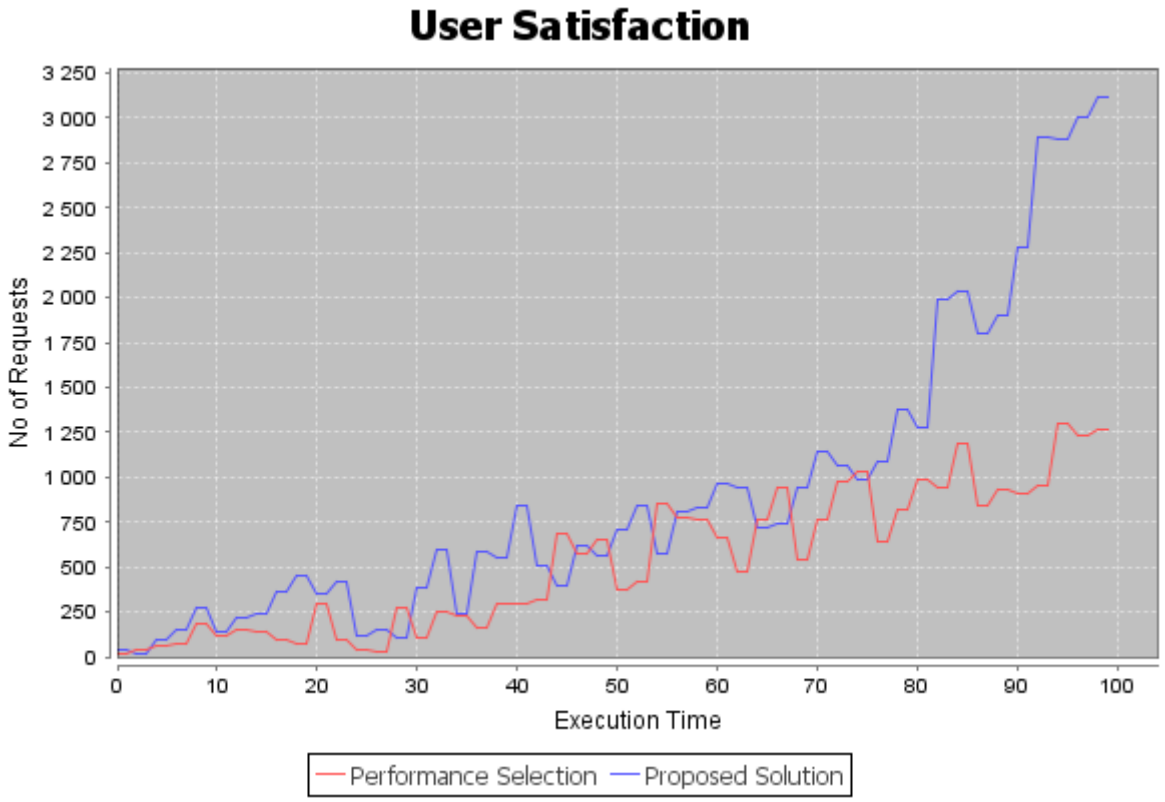

Figure 4.6. Satisfaction de l'utilisateur

Interprétations

La figure 4.6 présente les résultats du scénario présenté. À partir de ces résultats, nous pouvons remarquer qu'avec la réduction du temps de création de machines virtuelles dans le modèle proposé, des demandes supplémentaires peuvent être servies par rapport au modèle de performance. La réduction du temps de création des VM est due au fait que pour le modèle proposé, les services ne sont pas seulement choisis en fonction des performances mais aussi en fonction des valeurs de confiance associées ainsi que les préférences utilisateurs, ce qui va alléger les services débordés et répartissant ainsi la charge. Par conséquent, la satisfaction de l'utilisateur et le profit économique peuvent être atteints. Ici, comme le temps de service augmente, la différence de traitement des requêtes entre notre proposition et le modèle de performance augmente également.

4.2.2. Evaluation et analyse du Modèle de filtrage

Dans cette partie, nous allons procéder à la validation du modèle de filtrage des avis malhonnêtes intégrés au système proposé.

Avant de décrire les différentes simulations, nous allons présenter le modèle de note sur lequel est basé l'ensemble des expérimentations.

Modèles de notes

Nous supposons que toutes les notes sont soumises à une distribution normale, lorsqu'un utilisateur positif honnête note un service de haute qualité de façon objective, un utilisateur négatif honnête note un service de faible qualité de façon objective. De même, un utilisateur malhonnête note les services de façon opposée par rapport aux utilisateurs honnêtes.

	Notes des utilisateurs honnêtes		**Notes des utilisateurs malhonnêtes**	
	Positif Honnête	*Négatif Honnête*	*Positif Malhonnête*	*Négatif Malhonnête*
Service de haute qualité	note [7 – 10] toutes les transactions	NA	NA	note [1 - 3] toutes les transactions
Service de qualité moyenne	NA	NA	NA	NA
Service de basse qualité	NA	note [1 - 3] toutes les transactions	note [7 – 10] toutes les transactions	NA

Tableau 4.2. Modèle de notes

Dans cette étude, nous avons fourni trois sortes de qualité de service : haute, moyenne et faible, (cependant, nous pourrions facilement l'étendre à un intervalle plus grand). Pour chaque service et en fonction de sa qualité, nous avons généré pour les utilisateurs un taux basé sur son groupe, comme détaillé dans le tableau 4.2. Dans le tableau, la valeur "NA" indique une note générée aléatoirement selon une distribution normale.

Si l'utilisateur est un évaluateur positif honnête, les services de haute qualité auront une valeur supérieure à 7. Pour les services avec d'autres qualités, les taux sont générés aléatoirement. Si l'utilisateur est un évaluateur négatif honnête, le service à faible qualité aura une valeur inférieure à 3. Pour les services avec d'autres qualités, les taux sont générés aléatoirement.

Si l'utilisateur est un évaluateur positif malhonnête, les services de faible qualité auront une valeur supérieure à 7. Pour les services avec d'autres qualités, les taux sont générés aléatoirement. Si l'utilisateur est un évaluateur négatif malhonnête, le service avec haute qualité aura une valeur inférieure à 3. Pour les services avec d'autres qualités, les taux sont générés aléatoirement.

Mesures de Performance

Plusieurs indicateurs ont été proposés pour évaluer la précision de méthodes de filtrage des avis. Pour évaluer la précision des mesures statistiques, l'erreur moyenne absolue (MAE) et l'erreur quadratique moyenne (RMSE) sont les indicateurs les plus couramment utilisés pour mesurer l'exactitude de la prédiction d'un modèle (Gong, 2010) (Gu et al., 2009). Ainsi, nous allons utiliser ces indicateurs afin de comparer l'exactitude et la précision de l'approche proposée.

Si le nombre des notes d'utilisateurs pour un ensemble de services est n, alors la valeur MAE est la moyenne de la différence entre les n utilisateurs, comme décrite :

$$MAE = \frac{\sum_{i=1}^{n} |r_i - c_i|}{n} \quad (4.1)$$

Avec r_1, r_2, r_3, ..., r_n représente les notes classifiées et c_1, c_2, c_3, ..., c_n est l'ensemble de notes réelles des utilisateurs correspondants.

La métrique RMSE est définie comme suit:

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n} (r_i - c_i)^2}{n}} \quad (4.2)$$

Plus les valeurs des métriques MAE et RMSE sont basses, plus la classification ne sera pas précise. Ainsi, nous avons utilisé ces deux indicateurs pour évaluer l'exactitude des groupes classifiés.

Pour valider l'exactitude de notre approche, nous l'avons comparé avec les deux approches suivantes :

- Aléatoire : avant de procéder à la classification des groupes, les valeurs manquantes sont générées aléatoirement

- Moyenne : la valeur manquante, pour un utilisateur x sur un service i, est remplacée par la valeur de la moyenne des notes pour l'utilisateur x.

Scénario

L'environnement de simulation se compose d'utilisateurs de services de Cloud Computing et des fournisseurs de services de Cloud Computing. La simulation se déroule en cycles de simulation. Pour chaque cycle de simulation, un utilisateur Cloud note un service utilisé selon le modèle détaillé dans le tableau 4.2.

Les résultats présentés dans cette section ont été obtenus en supposant que 30% des utilisateurs soient positifs honnêtes, 30% soient positifs malhonnêtes, 20% soient positivement malhonnêtes et 20% soient négativement malhonnêtes. En outre, nous avons supposé que le nombre de notes NR = 10, avec 1 étant le plus faible et 10 étant le plus élevé. (La sélection de la NR = 10 est non significative et tout autre intervalle de valeurs peut être facilement utilisé). Le nombre total d'utilisateurs est de 500 et le nombre total de services est de 25. La réduction du nombre de services n'a aucune influence sur le modèle proposé. Par contre, la réduction du nombre d'utilisateurs à moins de 5 pourrait résulter en des valeurs incorrectes puisque le modèle se base sur les opinions des utilisateurs afin de classifier les différents groupes.

		Classes de références						
Technique de filtrage	***Classes d'utilisateurs***	***PH***	***NH***	***PM***	***NM***	***Prédiction Total***	***Exactitude***	
							% Exactitude	*MAE*
Classes prédites	***PH***	**136**	2	10	1	149	91%	0.180
	NH	13	**146**	0	5	164	97%	0.147
	PM	1	0	**86**	10	97	86%	0.250
	NM	0	2	4	**84**	90	84%	0.220
	Total	150	150	100	100	500	**90%**	0.200
Exactitude totale	452/500 = 90%							

Tableau 4.3. Matrice d'erreur

Interprétations

Les résultats indiquent que la solution proposée a obtenu des valeurs de précision de classification (84% - 97%) plus élevées pour chaque catégorie d'utilisateurs (tableau 4.3). Le tableau montre que la précision globale de classification de l'approche proposée était de 90%, ce qui représente une valeur élevée.

Pour plus de validation du modèle, nous avons comparé la solution proposée avec le filtrage aléatoire et le filtrage par moyenne pour 100 tours de simulation. Les résultats expérimentaux de l'exactitude sont présentés dans la figure 4.7, pour les utilisateurs positifs honnêtes, dans la figure 4.8, pour les utilisateurs négatifs malhonnêtes, dans la figure 4.9, pour les utilisateurs positifs malhonnêtes et dans la figure 4.10, pour les utilisateurs négatifs malhonnêtes.

Les figures montrent que la solution proposée a obtenu une meilleure précision de la classification dans tous les cas de simulations.

D'abord, le filtrage aléatoire consiste à remplacer les valeurs manquantes par des valeurs aléatoires pour les notes d'un service donné. Cela pourrait être acceptable dans le cas où il y aurait peu de valeurs manquantes, mais pour un environnement réel, cela ne reflète pas la réalité et résulte en une classification incorrecte et biaisée.

Le filtrage de moyenne consiste à remplacer les notes manquantes par la moyenne des notes, ce qui ne reflètent pas toujours la réalité, puisqu'un groupe d'utilisateurs malhonnêtes peuvent se mettre d'accord afin de noter négativement un service donné, ce qui résultera en une fausse classification.

Enfin, le filtrage proposé assure une très bonne classification puisque les différentes notes sont remplacées par la moyenne de quatre différentes mesures de similarité et de distance selon un certain voisinage, calculé par la distance de Hamming, ceci garantit une classification juste, fiable et non biaisée.

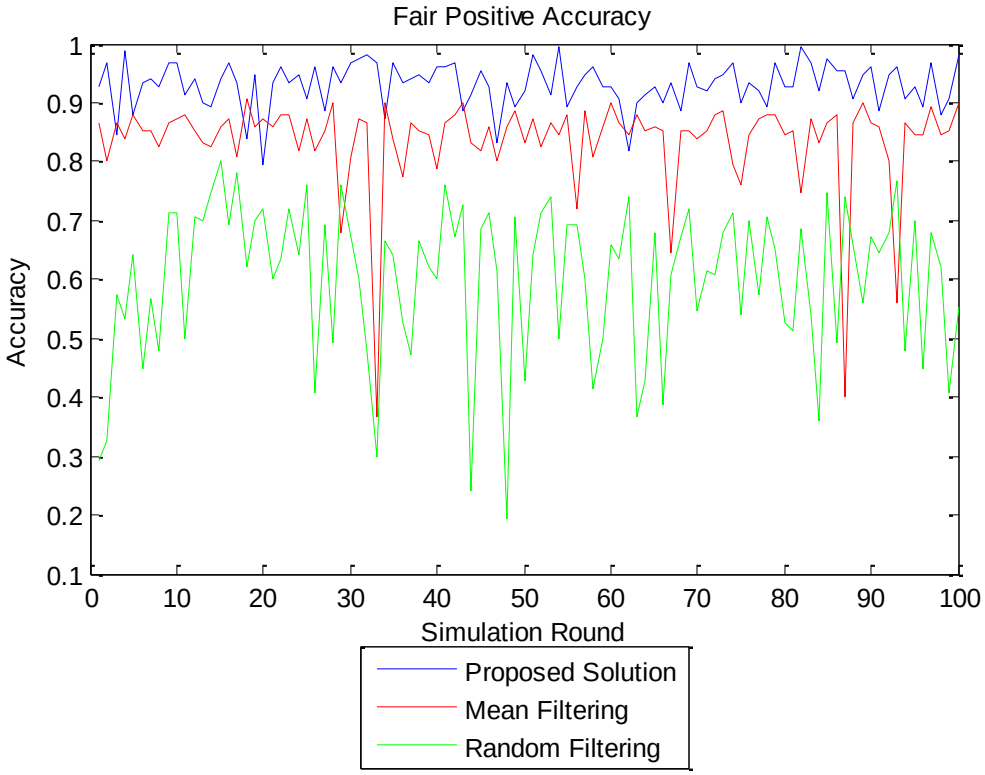

Figure.4.7. Exactitude pour les utilisateurs PH

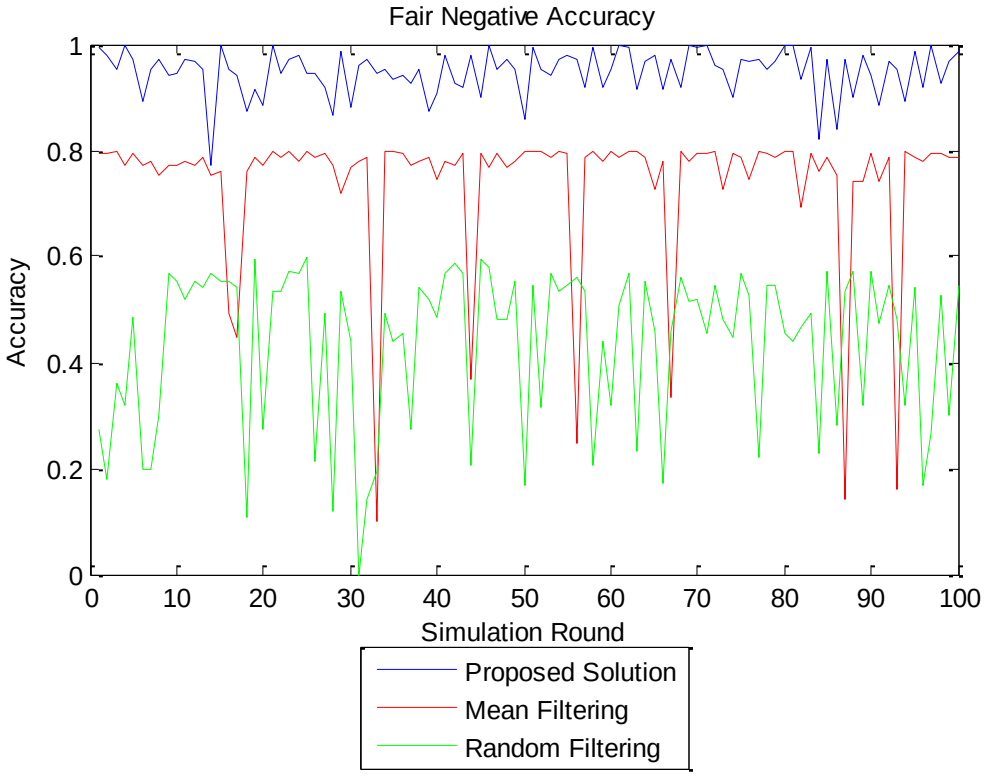

Figure.4.8. Exactitude pour les utilisateurs NH

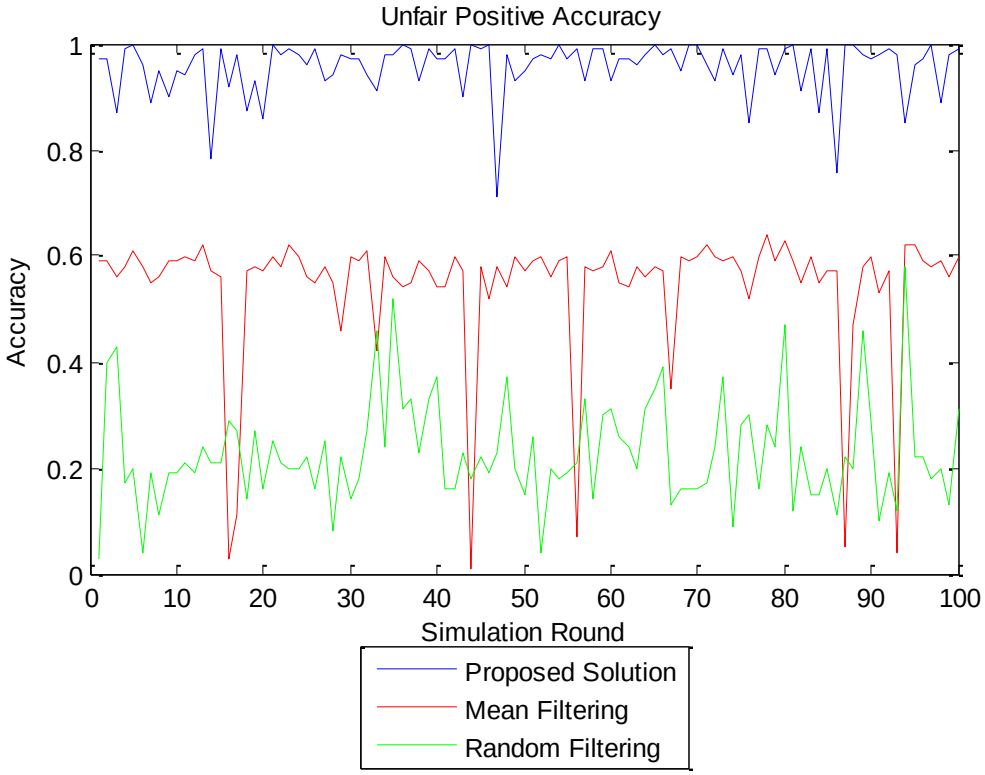

Figure.4.9. Exactitude pour les utilisateurs PM

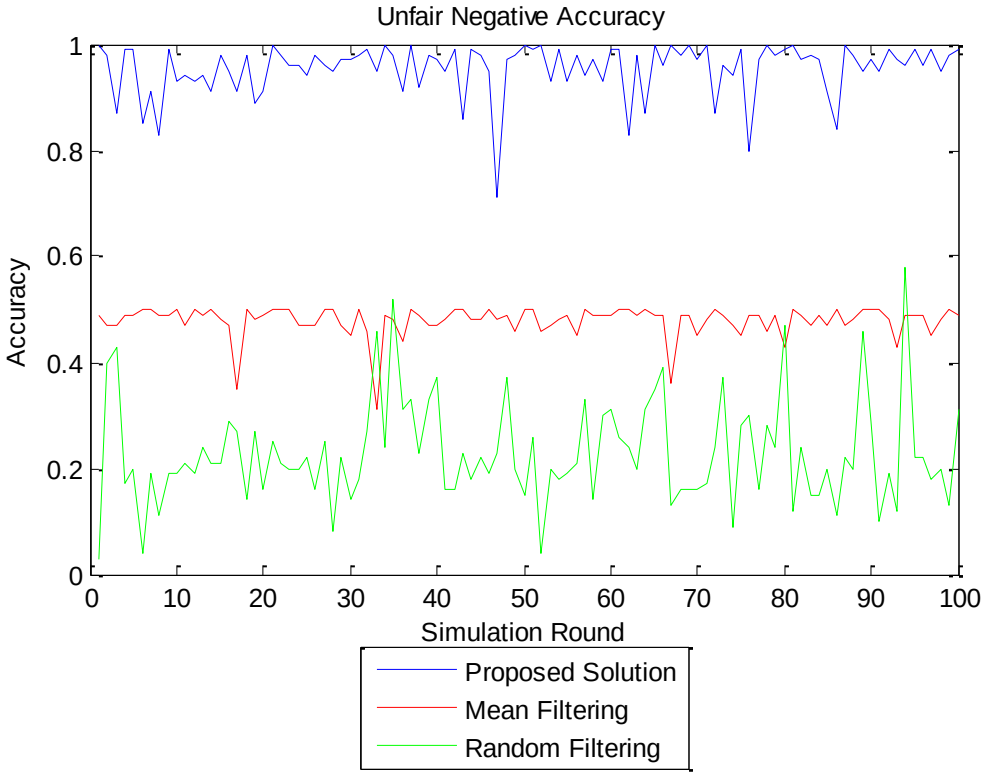

Figure.4.10. Exactitude pour les utilisateurs NM

Nous présentons également dans le tableau 4.4 les meilleurs, moyens et pires des cas pour chaque technique évaluée.

Cas	Métriques	Solution proposée	Aléatoire	Moyenne
Meilleur	MAE	***0.162***	0.365	0.190
	RMSE	***0.220***	0.419	0.311
Moyen	MAE	***0.181***	0.383	0.239
	RMSE	***0.245***	0.464	0.398
Pire	MAE	***0.193***	0.498	0.348
	RMSE	***0.297***	0.586	0.416

Tableau 4.4. Exactitude des techniques de filtrage

À partir des résultats des trois groupes d'expérimentations, nous pouvons remarquer que, dans le meilleur des cas, notre solution améliore par rapport aux techniques de filtrage aléatoire et filtrage par moyenne respectivement de 56% -15% en terme de MAE et de 47% -29% en terme de RMSE. Dans le pire des cas, les améliorations augmentent à 61% -45% en terme de MAE et 49% -29% en terme de RMSE. Ce résultat signifie que notre modèle a une meilleure robustesse. En d'autres termes, non seulement il se comporte bien dans le meilleur des cas, mais surmonte aussi les pires des cas avec une précision légèrement inférieure.

Par conséquent, nous pouvons conclure que les résultats expérimentaux ont démontré que la technique proposée augmente de manière significative la prédiction des valeurs manquantes pour les utilisateurs et cela afin de procéder à une classification précise des différents groupes. Cela est dû aux raisons suivantes. Premièrement, la solution proposée prend en compte plusieurs mesures de distance et de similarité. Deuxièmement, la technique de filtrage va améliorer le degré de confiance et la sélection de service, puisque la classification des groupes est effectuée sans aucune connaissance de la qualité du service offert.

4.2.3. Evaluation et analyse de la Plateforme de gestion de confiance

Dans ce qui suit, nous avons effectué des expérimentations afin de tester le système de confiance proposé.

Métriques

Afin de procéder à l'analyse du système et sa validation, nous avons utilisé différentes métriques afin d'évaluer la précision de la confiance.

- *Efficacité,* représente le nombre de services de confiance fourni par le total des services ;
- *Confiance moyenne*, représente le rapport de la somme des valeurs de confiance au nombre des valeurs de confiance.

Scénario 1

Pour ce scénario, le système proposé a été testé avec 20 services (10 services fiables et 10 services non fiables). Nous avons effectué 200 demandes d'utilisateurs dans 10 cycles de temps. Ensuite, nous avons compté le nombre de demandes d'utilisateurs affectées aux services de confiance, avec et sans (sélection basée sur la performance du service) le système de gestion de la confiance proposé. Les résultats sont présentés dans la Fig. 4.11 ci-dessous.

Figure.4.11. Sélection de Service de confiance

Interprétations

Selon les résultats obtenus dans la figure.4.11, nous constatons que le système de confiance proposé a une plus grande efficacité par rapport à la sélection de la performance. Pour le premier cycle de temps, le système prend un certain temps pour se stabiliser en raison de l'initialisation des premières transactions. En outre, la sélection de la performance choisit le service par sa valeur de rendement et ne considère pas la confiance du fournisseur par rapport au système développé.

Scénario 2

Pour la dernière expérimentation, nous avons comparé le système proposé à une solution basée sur la confiance seulement (sans inclure la performance dans le processus de sélection). Nous générons un ensemble de requêtes utilisateurs sur des cycles de temps. Ensuite, nous calculons la valeur moyenne de confiance pour les deux approches. Les résultats sont présentés dans la Figure.4.12.

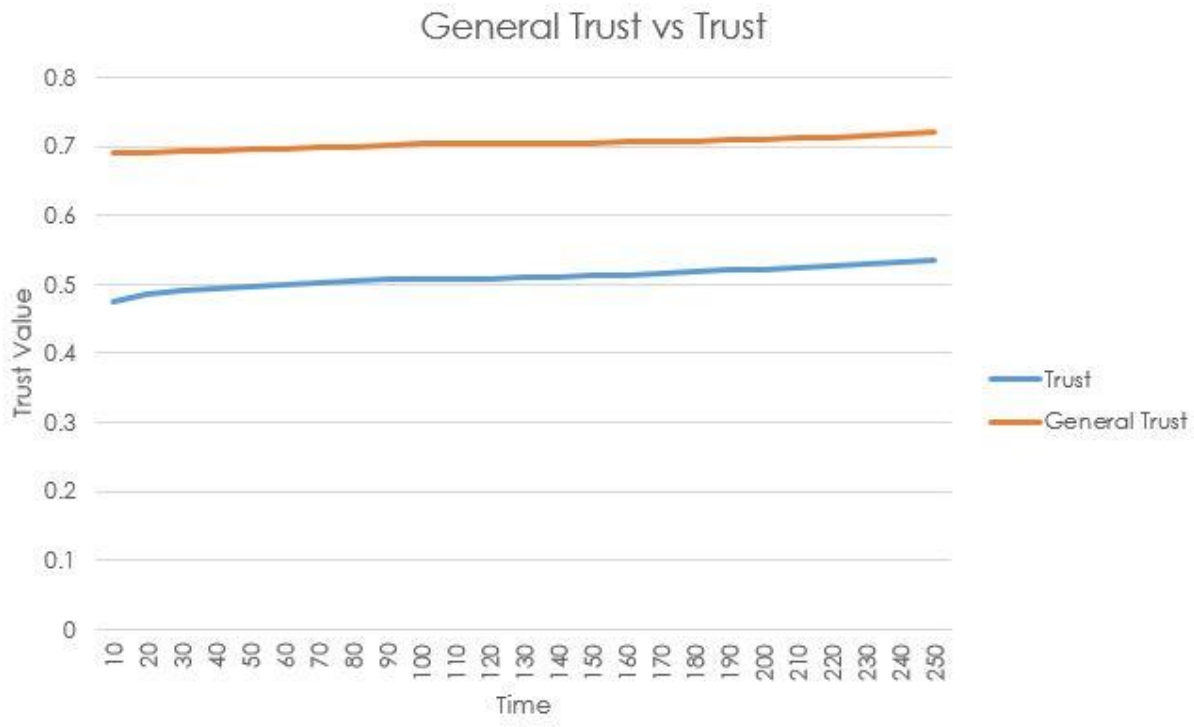

Figure.4.12. Sélection à base de moyennes de confiance

Interprétations

À partir du résultat, nous pouvons remarquer la différence dans les valeurs moyennes de confiance, entre la solution proposée et l'approche basée uniquement sur la confiance. Cela étant dû au fait, que le système proposé prend en compte les paramètres de performance dans la sélection du service. De ce fait, il donne un résultat plus fiable en comparaison à l'évaluation basée seulement sur la confiance. Par conséquent, il augmente le nombre de services fiables sélectionnés.

4.3. Conclusion

Le problème de la confiance des services Cloud qui est un problème sous-jacent à celui de la sécurité informatique n'a jamais été totalement résolu. Notre système de confiance ne

prétend pas l'avoir totalement résolu, car il présente certains aspects qui n'ont pas été intégrés au système développé. Cependant, Global Trust est un système de gestion de confiance qui a apporté des solutions à plusieurs points pour faire confiance à des services Cloud. Il permet d'abord de vérifier la crédibilité et la fiabilité des fournisseurs. Ensuite, il s'est penché sur la protection du système contre les attaques de confiance et filtrage des avis malhonnêtes. Enfin, il offre une sélection des fournisseurs les plus crédibles en prenant en compte l'hétérogénéité et la complexité d'un environnement Cloud.

L'étude des performances réalisée a permis d'avoir une idée sur sa fiabilité, son exactitude et sa robustesse. Toutefois, elle a été conçue en égard aux problèmes actuels ; ce qui pourrait le rendre vulnérable à de nouvelles attaques

Chapitre 5 : Conclusion et Perspectives

L'utilisation massive d'applications web, employant notamment des services à la demande, a été engendrée par des phénomènes récents tels que l'évolution des technologies et des infrastructures matérielles, la mobilité des utilisateurs ou la révolution de l'Internet. Ces applications évoluent dans des environnements de plus en plus dynamiques, imprévisibles et hostiles et engendrent donc un important problème de sélection d'un service fiable et de confiance. Ce défi est expliqué par l'importance et l'étendue d'utilisation des services Cloud pour les entreprises et les particuliers afin de pouvoir pleinement profiter de leurs avantages.

Le travail effectué dans cette thèse s'inscrit dans le cadre de cette problématique. Il vise particulièrement à proposer un système générique pour la sélection d'un service fiable et sûr dans un environnement hétérogène comme le Cloud. Pour résoudre ce problème, nous avons commencé par une étude de l'environnement de Cloud et une étude des approches et des modèles de confiance existants. Cette analyse a montré l'inexistence d'un système de gestion de confiance fiable pour un environnement de Cloud Computing. C'est la raison pour laquelle nous nous sommes penchés sur le problème de développement d'un modèle computationnel pour le calcul de confiance. Ce modèle étant basé sur deux paramètres importants qui sont la performance des services, ainsi que les recommandations et avis des différents utilisateurs. De plus, afin de fiabiliser l'approche proposée, il a été nécessaire d'intégrer un modèle de filtrage des avis des utilisateurs malhonnêtes.

5.1. Apports de notre travail

Nous avons présenté un système pour permettre la sélection de fournisseur Cloud fiables et sûrs et cela afin de traiter un des problèmes de sécurité se rapportant au Cloud Computing. Aucune approche, à notre connaissance, n'a utilisé les performances des services Cloud comme modèle d'opinion afin de l'agréger aux recommandations et confiance des utilisateurs pour résoudre ce problème. Cette technique repose sur la crédibilité des avis fournis et sur la QoS offerte. Elle permet une sélection basée sur des valeurs objectives, tout en l'intégrant aux paramètres subjectifs.

Comparée aux systèmes usuels de confiance, notre proposition fournit plusieurs avantages :

- Elle se base sur plusieurs sources pour calculer le degré de confiance finale.

- L'évaluation de la confiance est basée sur des valeurs objectives, qui représentent une dimension importante pour la comparaison de services Cloud.
- Le modèle de confiance proposé est flexible, l'utilisateur de service peut adapter la sélection selon ses préférences et ses besoins.
- Le processus de filtrage des avis fournit un intérêt certain puisqu'il permet de valider la fiabilité du système et ce sans aucune connaissance a priori.
- Le modèle computationnel est validé par rapport à deux méthodes de gestion de confiance connues et approuvées (EigenTrust, logique subjective).

5.2. Limites et perspectives

Toutefois, dans cette thèse nous n'avons pas abordé le problème de la gestion des contrats SLA entre les utilisateurs et les fournisseurs Cloud. Le prolongement immédiat de ce travail serait la définition d'une politique de gestion de contrats afin de l'intégrer au processus de sélection final.

Une autre perspective comme continuité à ce travail serait de tester le système développé dans un environnement réel afin de prouver son efficacité à long terme. À ce sujet nous pourrons y intégrer un module de surveillance et monitoring des services Cloud générique afin de pallier aux problèmes d'hétérogénéités des plateformes de déploiement des services Cloud.

Dans notre approche de filtrage, nous avons considéré la technique des k-means pour la classification des groupes d'utilisateurs toutefois, il serait intéressant de voir le résultat par rapport à d'autres approches de classification telles que l'Espérance-Maximisation (EM), les chaînes de Markov, les cartes auto-adaptative (SOM), etc…

Enfin, rappelons que les problèmes de confiance et de sécurité dans un environnement ouvert et hétérogène tel que le Cloud Computing demeurent toujours des problèmes ouverts, en conséquence beaucoup de pistes restent à explorer.

Références Bibliographiques

Abbas, A. (2004). Grid Computing: A Practical Guide to Technology and Applications. 406p. Charles River Media, 2éme édition, 2004.

Abdul-Rahman, A., & Hailes, S. (1998). A distributed trust model. *Proceedings of the 1997 workshop on New security paradigms*, p.48–60, 1998.

Abdul-Rahman, A., & Hailes, S. (2000). Supporting trust in virtual communities. *Proceedings of the 33rd Annual Hawaii International Conference on System Sciences*, p.9, 2000.

Ahmad, S., Ahmad, B., Saqib, S. M., & Khattak, R. M. (2012). Trust Model: Cloud's Provider and Cloud's User. *International Journal of Advanced Science and Technology*, Vol. *44*, p.69–80, 2012.

Ahn, H. J. (2008). A new similarity measure for collaborative filtering to alleviate the new user cold-starting problem. *Information Sciences, Vol. 178, N1*, p.37–51, 2008.

Alhamad, M. (2011). SLA-Based Trust Model for Secure Cloud Computing, thèse de doctorat, *Curtin University*, 202p, 2011.

Alhamad, M., Dillon, T., & Chang, E. (2010). SLA-Based Trust Model for Cloud Computing. *Proceedings of the 13th International Conference on Network-Based Information Systems (NBiS)*, p.321–324, 2010.

Andrei, T., & Jain, R. (2009). Cloud computing challenges and related security issues. Rapport, *washington university*, 10p, 2009.

Armbrust, M., Stoica, I., Zaharia, M., Fox, A., Griffith, R., Joseph, A. D., Rabkin, A. (2010). A view of cloud computing. *Communications of the ACM, Vol. 53, N4*, p.50-58, 2010.

Artz, D., & Gil, Y. (2007). A survey of trust in computer science and the Semantic Web. *Web Semantics, Vol. 5, N2*, p.58–71, 2007.

Bamiah, M., & Brohi, S. (2011). Seven deadly threats and vulnerabilities in cloud computing. *International Journal of Advanced Engineering Sciences and Technologies (IJAEST), Vol.9, N1*, p.87–90, 2011.

Barber, K. S., & Kim, J. (2001). Belief Revision Process Based on Trust: Agents Evaluating Reputation of Information Sources. *Trust in Cybersocieties, LNCS Vol. 2246*, p.73–82, 2001.

Baun, C. (2011). Cloud computing web-based dynamic IT services. 100p. Springer Science & Business Media, 1ére édition, 2011.

Bharadwaj, K. K., & Al-Shamri, M. Y. H. (2009). Fuzzy computational models for trust and reputation systems. *Electronic Commerce Research and Applications, Vol.8, N1*, p.37-47, 2009.

Bhatti, T. R. (2005). Critical Success Factors for the Implementation of Eentreprise Resource Planning (ERP): Empirical Validation. 2nd *International Conference on Innovation in Information Technology*, Dubai, UAE, Septembre 26-28, 2005.

Blaze, M., Feigenbaum, J., & Keromytis, A. D. (1999). Trust Management for Public-Key Infrastructures. *Proceedings of 6th International Workshop Cambridge on Security Protocols, LNCS, Vol. 1550*, p.59–63, 1999.

Blaze, M., Feigenbaum, J., & Lacy, J. (1996). Decentralized trust management. *Proceedings of Security and Privacy on IEEE Symposium*, p.164–173, 1996.

Blaze, M., Ioannidis, J., & Keromytis, A. D. (2003). Experience with the KeyNote trust management system: Applications and future directions. *Proceedings on Trust Management, LNCS, Vol. 2692*, p.284–300, 2003.

Boyd, J. (2002). In Community We Trust: Online Security Communication at eBay. *Journal of Computer-Mediated Communication, Vol. 7, N3*, p.0-0 , 2002.

Boyle, R., & Bonacich, P. (1970). The development of trust and mistrust in mixed-motive games. *Sociometry, Vol. 33, N2*, p.123–139, 1970.

Brin, S., & Page, L. (2012). Reprint of: The anatomy of a large-scale hypertextual web search engine. *Computer Networks, Vol. 56, N*18, p.3825–3833, 2012.

Buyya, R., Abramson, D., & Venugopal, S. (2005). The grid economy. *Proceedings of the IEEE, Vol. 93, N3*, p.698–714, 2005.

Buyya, R., Yeo, C. S., Venugopal, S., Brandic, I. (2009). Cloud computing and emerging IT platforms: Vision, hype, and reality for delivering computing as the 5th utility. *Future Generation Computer Systems, vol 25, N*6, p.599-616, 2009.

Buyya, R., Yeo, C. S., & Venugopal, S. (2008). Market-oriented cloud computing: Vision, hype, and reality for delivering IT services as computing utilities. *Proceedings of 10th IEEE International Conference on High Performance Computing and Communications, HPCC*, p.5–13, 2008.

Calheiros, R. N., Ranjan, R., Beloglazov, A., Cesar, a F. D. R., & Buyya, R. (2011). CloudSim:a toolkit for modeling and simulation of cloud computing environments and evaluation of resource. *Software - Practice and Experience, Vol. 41, N1*, p.23–50, 2011.

Catteddu, D., & Hogben, G. (2009). Cloud Computing: Benefits, risks and recommendations for information security. *Web Application Security, CCIS, Vol. 72*, p.17-17, 2009.

Chen, M., & Singh, J. P. (2001). Computing and using reputations for internet ratings. *Proceedings of the 3rd ACM Conference on Electronic Commerce - EC '01*, p.154–162, 2001.

Chrysanthos, D. (2000). Immunizing online reputation reporting systems against unfair ratings and discriminatory behavior. *Proceedings of the 2nd ACM Conference on Electronic Commerce*, p.150–157, 2000.

Coulouris, G., Dollimore, J., & Kindberg, T. (2012). Distributed Systems: Concepts and Design. Computer, 1047p, Addison-Wesley, 5éme édition, 2012.

Cummings, L. L., & Bromiley, P. (1996). The organizational trust inventory. *Trust in organizations: Frontiers of theory and research*, p.302-330, 1996.

Czajkowski, K., Foster, I., Kesselman, C., Martin, S., Smith, W., & Tuecke, S. (1998). A Resource Management Architecture for Metasystems. *Proceedings on Job Scheduling Strategies for Parallel Processing LNCS,* p.62-82, 1998.

Deutsch, M. (1962). Cooperation and trust: Some theoretical notes. *Nebraska Symposium on Motiviation, vol10, N4*, p.230–275, 1962.

Divakarla, U., & Sekaran, K. C. (2015). Trust Models in Cloud : A Survey on Pros and Cons. *New Trends in Networking, Computing, E-learning, Systems Siences and Engineering, LNCS, Vol. 312,* p.335-341, 2015.

Fan, W., & Perros, H. (2013). A reliability-based trust management mechanism for cloud services. *Proceedings of 12th IEEE International Conference on Trust, Security and Privacy in Computing and Communications, TrustCom 2013*, p.1581–1586, 2013.

Fernandes, D. a B., Soares, L. F. B., Gomes, J. V., Freire, M. M., & Inácio, P. R. M. (2014). Security issues in cloud environments: A survey. *International Journal of Information Security, Vol. 13, N*2, p.113–170, 2014.

Filali, F. Z., & Yagoubi, B. (2015a). A General Trust Management Framework for Provider Selection in Cloud Environment. *Advances in Databases and Information Systems, Lecture Notes in Computer Science, LNCS, Vol. 9282*, Poitiers, France, Springer International Publishing, p.446–457, 2015.

Filali, F. Z., & Yagoubi, B. (2015b). Classifying and Filtering Users by Similarity Measures for Trust Management in Cloud Environment. *Scalable Computing: Practice and Experience*, *Vol. 16, N*3, p.289–301, 2015.

Filali, F. Z., & Yagoubi, B. (2015c). Global Trust: A Trust Model for Cloud Service Selection. *International Journal of Computer Network and Information Security*, *Vol.7, N5*, p.41–50, 2015.

Foster, I., & Kesselman, C. (1997). Globus: a Metacomputing Infrastructure Toolkit. *International Journal of High Performance Computing Applications*, *Vol. 11, N2,* p.115-128, 1997.

Foster, I., Kesselman, C., Nick, J. M., & Tuecke, S. (2002). Grid services for distributed system integration. *Computer*, *Vol. 35, N6*, p.37–46, 2002.

Fu, J., Wang, C., Yu, Z., Wang, J., & Sun, J. G. (2010). A watermark-aware trusted running environment for software clouds. *Proceedings of 5th Annual China Grid Conference, ChinaGrid 2010*, p.144–151, 2010.

Gambetta, D. (2000). Can we trust trust? *Trust: Making and Breaking Cooperative Relations*, p.213-237, 2000.

Garg, S. K., Versteeg, S., & Buyya, R. (2013). A framework for ranking of cloud computing services. *Future Generation Computer Systems*, *Vol. 29, N4*, p.1012–1023, 2013.

Golbeck, J. A. (2005). Computing and applying trust in web-based social networks. Thèse de doctorat, *Univeristy of Maryland*, 185p, 2005.

Gong, S. (2010). An efficient collaborative recommendation algorithm based on item clustering. *Proceedings of Advances in Wireless Networks and Information Systems, LNEE, Vol. 72*, p.381–387, 2010.

Grandison, T., & Sloman, M. (2000). A survey of trust in internet applications. *IEEE Communications Surveys & Tutorials, Vol. 3, N4*, p.2-16, 2000.

Grandison, T., & Sloman, M. (2003). Trust Management for Internet Applications. Thèse de doctorat, *University of London*, 252p, 2003.

Gu, J. G. J., Zhou, J. Z. J., & Chen, X. C. X. (2009). An Enhancement of K-means Clustering Algorithm. *Proceedings of International Conference on Business Intelligence and Financial Engineering*, *Vol. 2, N2*, p.2–5, 2009.

Habib, S. M., Varadharajan, V., & Muhlhauser, M. (2013). A Trust-Aware Framework for Evaluating Security Controls of Service Providers in Cloud Marketplaces. *Proceedings of the 12th IEEE International Conference on Trust, Security and Privacy in Computing and Communications*, p.459–468, 2013.

Habib, S., & Ries, S. (2013). Towards a trust management system for cloud computing marketplaces: using CAIQ as a trust information source. *Security and Communication Networks, Vol. 7, N11*, p.2185-2200, 2013.

Halberstadt, A., Mui, L., Mohtashemi, M., Ang, C., & Szolovits, P. (1998). Ratings in Distributed Systems : A Bayesian Approach. *In Proceedings of the Workshop on Information Technologies and Systems (WITS)*, p.1–7, 1998.

Huang, J., Nie, F., Huang, H., & Tu, Y.-C. (2012). Trust prediction via aggregating heterogeneous social networks. *Proceedings of the 21st ACM International Conference on Information and Knowledge Management - CIKM '12*, p.1774-1778, 2012.

Iltaf, N., & Ghafoor, A. (2013). A fuzzy based credibility evaluation of recommended trust in pervasive computing environment. *IEEE 10th Consumer Communications and Networking Conference*, p.617–620, 2013.

Josang, A. (1999). Trust-based decision making for electronic transactions. In *Proceedings of the Fourth Nordic Workshop on Secure Computer Systems (NORDSEC'99)*, p.496-502, 1999.

Jøsang, A. (2001). a Logic for Uncertain Probabilities. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems, Vol. 9, N3*, p.279–311, 2001.

Jøsang, A., & Golbeck, J. (2009). Challenges for robust trust and reputation systems. *5th International Workshop on Security and Trust Management (SMT'09),* Saint Malo, France, p.1–12, 2009.

Jøsang, A., Gray, E., & Kinateder, M. (2006). Simplification and Analysis of Transitive Trust Networks. *Web Intelligence and Agent Systems*, *Vol. 4, N*2, p.1–26, 2006.

Jøsang, A., Hayward, R., & Pope, S. (2006). Trust network analysis with subjective logic. *Proceedings of Conferences in Research and Practice in Information Technology Series, Vol. 48*, p.85–94, 2006.

Jøsang, A., & Ismail, R. (2002). The Beta Reputation System. *Proceedings of the 15th Bled Electronic Commerce Conference, Vol. 160*, p.324–337, 2002.

Jøsang, A., Ismail, R., & Boyd, C. (2007). A survey of trust and reputation systems for online service provision. *Decision Support Systems, Vol. 43*, p.618–644, 2007.

Josang, A., & Lo Presti, S. (2004). Analysing the Relationship Between Risk and Trust. *Proceedings of Second International Conference iTrust on Trust Management, LNCS, Vol. 2995*, p.135–145, 2004.

Jøsang, A., & Pope, S. (2005). Semantic constraints for trust transitivity. *Proceedings of Cnferences in Research and Practice in Information Technology, LNCS, Vol. 43*, p.59–68, 2005.

Jøsang, A., & Quattrociocchi, W. (2009). Advanced features in Bayesian reputation systems. *Proceedings of the 6th International Conference TrustBus on Trust, Privacy and Security in Digital Business, LNCS, Vol. 5695*, p.105–114, 2009.

Kamvar, S. D., Schlosser, M. T., & Garcia-Molina, H. (2003). The Eigentrust algorithm for reputation management in P2P networks. *Proceedings of the Twelfth International Conference on World Wide Web - WWW '03*, p. 640-651, 2003.

Kerr, R., & Cohen, R. (2006). Modeling Trust Using Transactional , Numerical Units. *Proceedings of the 2006 International Conference on Privacy Security and Trust Bridge the Gap Between PST Technologies and Business Services*, p.1–11, 2006.

Kerr, R., & Cohen, R. (2009). Smart Cheaters Do Prosper : Defeating Trust and Reputation Systems. *Proceedings of The 8th International Conference on Autonomous Agents and Multiagent Systems-Vol. 2*, p.993-1000, 2009.

Khajeh-Hosseini, A., Greenwood, D., & Sommerville, I. (2010). Cloud migration: A case study of migrating an enterprise IT system to IaaS. *Proceedings - 2010 IEEE 3rd International Conference on Cloud Computing, CLOUD 2010*, p.450–457, 2010.

Khan, K., & Malluhi, Q. (2010). Establishing trust in cloud computing. *IT Professional, Vol. 12, N5*, p.20-27, 2010.

Kourpas, E. (2006). Grid computing: Past, present and future. An innovation perspective. *IBM White Paper*, p.1–22, 2006.

Lamport, L. (1978). Time, clocks, and the ordering of events in a distributed system. *Communications of the ACM, Vol. 21, N7*, p.558–565, 1978.

Lee, J. S., Jun, C. H., Lee, J., & Kim, S. (2005). Classification-based collaborative filtering using market basket data. *Expert Systems with Applications, Vol. 29, N3*, p.700–704, 2005.

Lempel, R., & Moran, S. (2000). Stochastic approach for link-structure analysis (SALSA) and the TKC effect. *Computer Networks, Vol. 33, N*1, p.387–401, 2000.

Li, X.-Y., Zhou, L.-T., Shi, Y., & Guo, Y. (2010). A trusted computing environment model in cloud architecture. *Proceedings of 2010 International Conference on Machine Learning and Cybernetics (ICMLC), Vol. 6*, p.2843–2848, 2010.

Luhmann, N. (1979). Trust and Power. 208p, John Wiley & Sons, 1ére édition, 1979.

Macías, M., & Guitart, J. (2015). Analysis of a trust model for SLA negotiation and enforcement in cloud markets. *Future Generation Computer Systems, Vol. 55*, p.460–472, 2015.

Manchala, D. W. (1998). Trust metrics, models and protocols for electronic commerce transactions. *Proceedings of 18th International Conference on Distributed Computing Systems*, p.312-321, 1998.

Manuel, P. (2013). A trust model of cloud computing based on Quality of Service. *Annals of Operations Research, Vol. 233, N1*, p.281–292, 2013.

Manuel, P. D., Thamarai Selve, S., & Barr, M. I. A. E. I. (2009). Trust management system for grid and cloud resources. *1st International Conference on Advanced Computing, ICAC 2009*, Chennai, 13-15 Dec. 2009, p.176–181, 2009.

Mármol, F. G., & Pérez, G. M. (2009). Security threats scenarios in trust and reputation models for distributed systems. *Computers & Security, Vol. 28, N7*, p.545–556, 2009.

Mármol, F. G., Sorge, C., Ugus, O., & Pérez, G. M. (2013). WSANRep, WSAN reputation-based selection in open environments. *Wireless Personal Communications, Vol. 68, N3*, p.921–937, 2013.

Marsh, S. P. (1994). Formalising Trust as a Computational Concept. Thèse de doctorat, *University of Stirling*, 1994.

McCarthy, J. (1960). Recursive functions symbolic expressions and their computation by machine, Part I. *Communications of the ACM, Vol. 3, N*4, p.184–195, 1960.

Mcknight, D. H., & Chervany, N. L. (1996). The meanings of trust. Thèse de doctorat, *University of Minnesota*, 86p. 1996.

Mell, P., & Grance, T. (2009). The NIST Definition of Cloud Computing. *National Institute of Standards and Technology*, *Vol. 53, N*6, p.50-50, 2009.

Mills, D. H. (1983). The Logic and Limits of Trust. *Business and Professional Ethics Journal*, *Vol. 2, N*3, p.77–78, 1983.

Mui, L., Mohtashemi, M., & Halberstadt, A. (2002). A computational model of trust and reputation. *Proceedings of the 35th Annual Hawaii International Conference on System Sciences*, p.2431-2439, 2002.

Muller, T., & Schweitzer, P. (2013). On beta models with trust chains. *Trust Management VII, IFIP, Vol. 401*, p.49–65, 2013.

Neapolitan, R. (1988). Probabilistic reasoning in intelligent systems. *Journal of Philosophy, Vol. 88, N8, p.434-437*, 1988.

Noor, T. H., Sheng, Q. Z., & Alfazi, A. (2013). Reputation attacks detection for effective trust assessment among cloud services. *Proceedings of 12th IEEE International Conference on Trust, Security and Privacy in Computing and Communications, TrustCom 2013*, p.469–476, 2013.

Noor, T., Sheng, Q., Zeadally, S., & Yu, J. (2013). Trust management of services in cloud environments: Obstacles and solutions. *ACM Computing Surveys (CSUR), Vol. 46, N1*, p.1–30, 2013.

Pavlidis, M. ., Mouratidis, H., Kalloniatis, C. ., Islam, S. ., & Gritzalis, S. . (2013). Trustworthy selection of cloud providers based on security and privacy requirements: Justifying trust assumptions. *Proceedings of the 10th International Conference, TrustBus 2013 on Trust, Privacy, and Security in Digital Business, LNCS, Vol. 8058*, p.185–198, 2013.

Pearson, S., & Benameur, A. (2010). Privacy, security and trust issues arising from cloud computing. *Proceedings of IEEE Second International Conference on Cloud Computing Technology and Science (CloudCom)*, p.693–702, 2010.

Puthal, D., Sahoo, B. P. S., Mishra, S., & Swain, S. (2015). Cloud Computing Features , Issues and Challenges : A Big Picture. *A Big Picture. Proceedings of 2015 International Conference on Computational Intelligence and Networks (CINE),* p.116-123, 2015.

Qu, C., & Buyya, R. (2014). A Cloud Trust Evaluation System Using Hierarchical Fuzzy Inference System for Service Selection. *Proceedings of IEEE 28th International Conference on Advanced Information Networking and Applications*, p.850–857, 2014.

Ranking, T. P. C., Ranking, T. P. C., Order, B., & Order, B. (1998). The PageRank Citation Ranking: Bringing Order to the Web. Rapport technique, *Stanford University*, 17p, 1998.

Rempel, J. K., & Holmes, J. G. (1986). How do I trust thee? *Psychology Today, Vol. 20, N2*, p.28–34, 1986.

Rempel, J. K., Holmes, J. G., & Zanna, M. P. (1985). Trust in close relationships. *Journal of Personality and Social Psychology, Vol. 49, N1*, 1985.

Resnick, P., Kuwabara, K., Zeckhauser, R., & Friedman, E. (2000). Reputation systems. *Communications of the ACM, Vol. 43, N2,* p.45-48, 2000.

Resnick, P., & Zeckhauser, R. (2002). Trust among strangers in Internet transactions: Empirical analysis of eBay's reputation system. *The Economics of the Internet and E-commerce, Vol. 11, N2,* p.23-25, 2002.

Ries, S. (2009). Extending Bayesian Trust Models Regarding Context-Dependence and User Friendly Representation. *Proceedings of the 2009 ACM Symposium on Applied Computing SAC '09*, p.1294-1301, 2009.

Ries, S., Habib, S. M., Mühlhäuser, M., & Varadharajan, V. (2011). CertainLogic: A logic for modeling trust and uncertainty (Short paper). *Proceedings of 4th International Conference, TRUST 2011 on Trust and Trustworthy Computing, LNCS, Vol. 6740,* p.254–261, 2011.

Rittinghouse, J., & Ransome, J. (2009). Cloud Computing - Implementation, Management and Security. 340p, CRC press, 1ére édition, 2009.

Rong, C., Nguyen, S. T., & Jaatun, M. G. (2013). Beyond lightning: A survey on security challenges in cloud computing. *Computers and Electrical Engineering, Vol. 39, N1*, p.47–54, 2013.

Sabater, J., & Sierra, C. (2002). Reputation and social network analysis in multi-agent systems. *Proceedings of the First International Joint Conference on Autonomous Agents and Multiagent Systems Part 1 - AAMAS '02*, p.475-482, 2002

Sato, H., Kanai, A., & Tanimoto, S. (2010). A cloud trust model in a security aware cloud. In *Proceedings of 2010 10th Annual International Symposium on Applications and the Internet, SAINT 2010*, p.121–124, 2010.

Schikuta, E., Weishaeupl, T., Donno, F., Stockinger, H., Vinek, E., Wanek, H., Haq, I. U. (2009). Business in the Grid, Rapport, 26p, 2009.

Sensoy, M. and Yolum, P. (2006). A Context-Aware Approach For Service Selection Using Ontologies. *Proceedings of the Fifth International Joint Conference on Autonomous Agents and Multiagent Systems*, p.931–938, 2006.

Shaikh, R., & Mumbai, N. (2012). Security Issues in Cloud Computing : A survey. *International Journal of Computer Applications, Vol. 44, N19*, p.4–10, 2012.

Shen, Z., Li, L., Yan, F., & Wu, X. (2010). Cloud Computing System Based on Trusted Computing Platform. *Proceedings of 2010 International Conference on Intelligent Computation Technology and Automation*, p.942–945, 2010.

Shen, Z. S. Z., & Tong, Q. T. Q. (2010). The security of cloud computing system enabled by trusted computing technology. *Proceedings of 2010 2nd International Conference on Signal Processing Systems (ICSPS), Vol. 2*, p.2-11, 2010.

Sherchan, W., Loke, S. W., & Krishnaswamy, S. (2006). A Fuzzy Model for Reasoning about Reputation in Web Services. *Scenario*, p.1886–1892, 2006.

Shi, Y., & Guo, Y. U. (2010). a Trusted Computing Environment Model in Cloud. *Machine Learning*, (July), p.2843–2848, 2010.

Simone, A., Škoric, B., & Zannne, N. (2012). Flow-Based Reputation: More than just Ranking. *International Journal of Information Technology & Decision Making*, *Vol. 11*, *N3*, p.551-578, 2012.

Song, S., Hwang, K., & Kwok, Y. K. (2006). Risk-resilient heuristics and genetic algorithms for security-assured grid job scheduling. *IEEE Transactions on Computers, Vol. 55, N*6, p.703–719, 2006.

Song, S., Hwang, K., Zhou, R., & Kwok, Y. K. (2005). Trusted P2P transactions with fuzzy reputation aggregation. *IEEE Internet Computing, Vol. 9, N6*, p.24–34, 2005.

Subashini, S., & Kavitha, V. (2011). A survey on security issues in service delivery models of cloud computing. *Journal of Network and Computer Applications, Vol. 34, N*1, p1–11, 2011.

Sun, Y. L., Han, Z., Yu, W., & Liu, K. J. R. (2006). A trust evaluation framework in distributed networks: Vulnerability analysis and defense against attacks. *Proceedings IEEE INFOCOM 2006. 25TH IEEE International Conference on Computer Communications*, p.1–13, 2006.

Tanenbaum, A. S., & Van Steen, M. (2007). Distributed Systems: Principles and Paradigms, 704p, Pearson, 2^{éme} édition, 2007.

Tavakolifard, M., & Almeroth, K. C. (2012). A Taxonomy to Express Open Challenges in Trust and Reputation Systems. *Journal of Communications, Vol. 7, N*7, p.538–551, 2012.

Tavakolifard, M., & Knapskog, S. J. (2009). A Probabilistic Reputation Algorithm for Decentralized Multi-Agent Environments. *Electronic Notes in Theoretical Computer Science, Vol. 244*, p.139–149, 2009.

Teacy, W. T. L., Luck, M., Rogers, a., & Jennings, N. R. (2012). An Efficient and Versatile Approach to Trust and Reputation using Hierarchical Bayesian Modelling, *Artificial Intelligence, Vol. 193*, p.149-185., 2012.

Thirunarayan, K., Anantharam, P., Henson, C., & Sheth, A. (2014). Comparative trust management with applications: Bayesian approaches emphasis. *Future Generation Computer Systems, Vol. 31, N*1, p.182–199, 2014.

Trevathan, J., & Read, W. (2007). A simple shill bidding agent. *Proceedings of International Conference on Information Technology-New Generations, ITNG 2007*, p.766–771, 2007.

Van Deursen, T., Koster, P., & Petković, M. (2008). Hedaquin: A Reputation-based Health Data Quality Indicator. *Electronic Notes in Theoretical Computer Science*, *Vol. 197, N*2, p.159–167, 2008.

Viljanen, L. (2005). Towards an Ontology of Trust. *Proceedings of Second International Conference, TrustBus on Trust, Privacy, and Security in Digital Business, LNCS, Vol.3592,* p.175–184, 2005.

Wang, T., Li, Y., & Zhu, L. (2011). Study on Enhancing Performance of Ad Hoc Trust Model with Family Gene Technology. *Proceedings of 2011 7th International Conference on Wireless Communications, Networking and Mobile Computing*, p.1–4, 2011.

Wang, T., Ye, B., Li, Y., & Yang, Y. (2010). Family gene based Cloud Trust model. *Proceedings of ICENT 2010 - 2010 International Conference on Educational and Network Technology*, p.540–544, 2010.

Wang, Y., & Singh, M. P. (2006). Trust representation and aggregation in a distributed agent system. In *Proccedings of the National Conference on Artificial Intelligence, Vol. 21,* p.1425-1430, 2006.

Whitby, A., Jøsang, A., & Indulska, J. (2005). Filtering out unfair ratings in bayesian reputation systems. *Proceedings of 7th International Workshop on Trust in Agent Societies, Vol. 6,* p. 106-117, 2005.

Yang, Z., Qiao, L., Liu, C., Yang, C., & Wan, G. (2010). A collaborative trust model of firewall-through based on Cloud Computing. *Proceedings of the 2010 14th International Conference on Computer Supported Cooperative Work in Design, CSCWD 2010*, p.329–334, 2010.

Yu, B., & Singh, M. P. (2003). Detecting Deception in Reputation Management. *Proceedings of the Second International Joint Conference on Autonomous Agents and Multiagent Systems*, p.73–80, 2003.

Zeng, H., Alhossaini, M. A., Ding, L., Fikes, R., & Mcguinness, D. L. (2006). Computing Trust from Revision History. *Proceeding of the International Conference on Privacy, Security and Trust*, p.1–8, 2006.

Zhang, Y., & Fang, Y. (2007). A fine-grained reputation system for reliable service selection in peer-to-peer networks. *IEEE Transactions on Parallel and Distributed Systems*, *Vol. 18, N*8, p.1134–1145, 2007.

Zhao, G., Rong, C., Jaatun, M. G., & Sandnes, F. E. (2010). Deployment models: Towards eliminating security concerns from cloud computing. *Proceedings of the 2010 International Conference on High Performance Computing and Simulation, HPCS 2010*, p.189–195, 2010.

Zimmermann, P. R., & Zimmermann, P. R. (1995). The official PGP user's guide. 146p, MIT press, 1ére édition, 1995.

Sites Web

[1] **F. Gens**. IT Cloud Services User Survey, pt.2: Top Benefits & Challenges. A partir de: http://blogs.idc.com/ie/?p=210. Octobre 2008. Consulté Novembre 2015.

[2] **F. Gens**. New IDC IT Cloud Services Survey: Top Benefits and Challenges. A partir de: http://blogs.idc.com/ie/?p=730. Décembre 2009. Consulté Novembre 2015.

[3] **W. Jansen et T. Grance**. Guidelines on security and privacy in public cloud Computing. Rapport Technique. National Institute of Standards and Technology. Janvier 2011. Draft Special Publication 800 -144. Disponible sur :http://csrc.nist.gov/publications/drafts/800-144/ Draft-SP-800-144_cloud-Computing.pdf. Consulté Novembre 2015.

[4] **D. Talbot**. Security in the Ether. Technology Review, pages 36–42, Février 2010.

[5] **IBM.** The Benefits of Cloud Computing. Pages 1-16. Juillet 2009. Disponible sur: http://www.ibm.com/ibm/files/H300444G23392G14/13Benefits_of_Cloud_Computing_634KB.pdf. Consulté Novembre 2015.

[6] **Daryl Plummer**. Experts Define Cloud Computing: Can we get a Little Definition in our definitions? Janvier 2009. Disponible sur: http://blogs.gartner.com/daryl_plummer/2009/01/27/experts-define-cloud-Computing-can-we-get-a-little-definition-in-our-definitions/. Consulté Novembre 2015.

[7] **Frank Gens**. Defining “Cloud Services” and “Cloud Computing”. Septembre 2008. Disponible sur : http://blogs.idc.com/ie/?p=190. Consulté Novembre 2015.

[8] **Robert McMillan**. Cloud Computing est un « cauchemar de la sécurité » selon le PDG de Cisco. http://www.pcworld.com/article/163681/article.html, 2009. Consulté Novembre 2015.

[9] **Liberty-Alliance**. Liberty Trust Models Guidelines. Disponible sur: http://www.projectliberty.org/specs/liberty-trustmodels-guidelines-v1.0.pdf, Draft Version 1.0-15, 2003. Consulté Novembre 2015.

[10] **United Device**. “The History of Distributed Computing”. Disponible sur: http://www.ud.com/products/dc/history.htm. Consulté Mars 2015.

[11] **Jay Heiser, Mark Nicolett**. Assessing the Security Risks of Cloud Computing. Juin 2008. Disponible sur : https://www.gartner.com/doc/685308/assessing-security-risks-cloud-computing. Consulté Septembre 20014.

[12] **Oxford Dictionnary.** Reputation. Disponible sur : http://www.oxforddictionaries.com/definition/english/reputation. Consulté Avril 2015.