

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE
SCIENTIFIQUE
UNIVERSITE ABBES LAGHROUR KHENCHELA
DEPARTEMENT DE MATHEMATIQUE ET INFORMATIQUE



Mémoire de fin d'étude pour l'obtention de diplôme
De master informatique
Sécurité et technologie web

Thème

Proposition d'un modèle de réputation et de
confiance dans le Cloud Computing Volontaire

Réalisé par :

- Aghrou Nezha
- Sebbag Aya

Encadré par :

Dr Hemam Soufian

Année Universitaire 2017/2018

Résumé

Pendant les vingt dernières années, Internet a évolué d'un réseau simple de taille limitée à un système complexe à grande dimension. Alors, qu'il était initialement utilisé pour offrir du contenu statique organisé autour de simples sites web, aujourd'hui il fournit des services complexes ainsi que l'externalisation de calculs, de stockage et d'applications avec le Cloud Computing. Ensuite ce système a été combiné avec le Computing volontaire pour montrer un autre type de système, le Cloud Computing volontaire.

Cependant, en dépit des avantages qu'offre ce nouveau système, la plupart des particuliers hésitent à l'adopter et son évolution a soulevé de nombreuses préoccupations et a été freinée par divers obstacles. La sécurité et la confiance sont les problèmes les plus cruciaux pour ces infrastructures et les risques qui accompagnent le déploiement de services et d'applications deviennent plus complexes avec l'architecture de cet environnement.

En outre, l'interaction des volontaires des services en Cloud volontaire avec les utilisateurs reste souvent basique et la confiance est son maillon faible.

Ce mémoire décrit l'état de l'art pour la gestion de confiance dans les environnements de Cloud Computing Volontaire et propose un système de gestion de confiance globale pour résoudre les problèmes de la confiance pour la sélection des nœuds et la classification des services. De plus, nous utilisons un modèle computationnel pour le calcul de confiance.

ملخص

خلال العشرين سنة الماضية، تطورت الإنترنت من شبكة بسيطة ذات حجم محدود إلى نظام معقد كبير. لذلك، تم استخدامه في البداية لتقديم محتوى ثابت منظم حول مواقع ويب بسيطة، واليوم يوفر خدمات معقدة وكذلك الاستعانة بمصادر خارجية الحسابات والتخزين والتطبيقات مع الحوسبة السحابية. ثم تم دمج هذا النظام مع الحوسبة الطوعية لإظهار نوع آخر من الأنظمة، وهو الحوسبة السحابية الطوعية.

ومع ذلك، وعلى الرغم من فوائد هذا النظام الجديد، فإن معظم الأفراد يترددون في تبنيه، وقد أثار تطوره الكثير من المخاوف وتعثرت بسبب عقبات مختلفة. يمثل الأمن والثقة أكثر القضايا أهمية بالنسبة لهذه البنية التحتية، والمخاطر التي تصاحب نشر الخدمات والتطبيقات تصبح أكثر تعقيداً مع بنية تلك البيئة. بالإضافة إلى ذلك، فإن تفاعل متطوعي الخدمات في السحابة الطوعية مع المستخدمين غالباً ما يكون أساسياً والثقة هي أضعف حلقة.

توضح هذه المذكرة أحدث ما توصلت إليه إدارة الثقة في بيئات الحوسبة السحابية الطوعية وتقتترح نظام إدارة ثقة عالمي لحل مشاكل الثقة في اختيار العقد وتصنيف الخدمات. بالإضافة إلى ذلك، فإننا نستخدم نموذجاً حسابياً لحساب الثقة.



Remerciement

Nous remercions le Bon Dieu Tout Puissant, de nous avoir donné Le bon sens et la grande volonté pour réaliser ce travail « الحمد لله ».

Nos remerciements les plus sincères et notre reconnaissance vont particulièrement à notre encadreur Dr Hamam soufian pour son aide très précieuse qu'il nous a apportée tout au long de notre préparation de ce mémoire, et du fait qu'il a bien voulu diriger ce travail et nous faire profiter de son savoir et de ses expériences.

Par la même occasion, nous tenons à remercier également tous ceux qui ont accepté de juger notre travail et tous les enseignants de l'informatique et surtout les enseignants du département ST.

Enfin, je remercie tous ceux qui ont participé de près ou de loin, de façon directe ou indirecte, à la réalisation de ce projet de fin d'études et surtout nos familles.

Merci A tous...



Dédicace

Je dédie ce mémoire à l'âme de mon cher père « **El Hachimi** » qui a dépensé toute sa vie pour assurer une éducation exemplaire à ses enfants. J'espère être digne de l'éducation et des précieux conseils qu'il m'a toujours prodigué.

Et quelle tristesse je ressens aujourd'hui encore à constater son absence. Puisse Dieu l'accueillir dans son paradis. Repose en paix mon père.

À ma chère mère : « **Mahnia** » qui représente la source de tendresse et l'exemple du dévouement qui n'a pas cessé de m'encourager et de prier pour moi.

À mes frères : « **IzzEssalam** », « **Habib Allah** » et son épouse « **Zina** » et son fils « **Abd El Aziz** ».

A mes chères sœurs : « **Sabah** », « **Sabrina** » et son mari « **Rabeh** »

Et les filles de ma soeur: « **Siradj ,Rahef ,Hadil ,Siri ,Djoury** ».

Aux personnes les plus chères de mon cœur : « **Afef, Amira, Rahma** »

Ma chère amie et mon binôme : « **Aya** ».

A l'homme de ma vie, mon soutien moral, et source de joie et de bonheur :

« **Abd Elhak.** »

À toutes mes collègues d'étude surtout : « **Khaoula ,Fatima ,Rayen ,Amira.** »

À toute ma famille et toutes les personnes que j'aime.



Dédicace

À mon tres cher père « *Ahmed* » tes sacrifices, ton soutien moral, ta gentillesse sans égal, ton profond attachement m'ont permis de réussir mes études. Sans ton aide, tes conseils et tes encouragements ce travail n'aurait vu le jour.

À ma tres chère mère « *Habiba* » tu est la source de tendresse et l'exemple du dévouement qui n'a pas cessé de m'encourager et de prier pour moi.

À mon cher mari « *Hamza* » la personne qui a su guider mes pas égarés vers un horizon plus clair, plus joyeux.

Et À mon petit ange « *Karam Amir* » qui a souffert à cause de mon absence.

À mes chères sœurs « *Sara* », « *Rabab* », « *Fedoua* » et « *Baraa* ».

À mon petit frère « *Islam* ».

À tous mes amis surtout : « *Hadjer* » et « *Nouzha* »

À tous les collègues de promotion d'Informatique (STW).

Aya

Liste des matières

Résumé	I
Remerciement.....	II
Dédicace.....	III
Introduction générale.....	1
Chapitre 01 : le Cloud Computing Volontaire	
Introduction.....	4
1. Cloud Computing	4
1.1.Définition.....	4
1.2.Services du Cloud Computing.....	5
1.3.Virtualisation.....	6
1.4.Modèle de déploiement.....	7
1.5.Avantages et défis.....	7
2. Cloud Computing Volontaire	7
2.1. Définition.....	7
2.2. Architecture.....	8
2.3. Avantages et inconvénients.....	9
Conclusion.....	10
Chapitre 02 : la confiance et la réputation	
Introduction.....	12
1. La confiance	12
1. 1 Définition de la confiance.....	12
1. 2 Caractéristique de confiance.....	13
1. 3 Principes de confiance.....	13
1. 4 Facteurs influant sur la confiance.....	13
1. 5 Types de confiance.....	13
1. 6 Modèle computationnels de confiance.....	14
2. La performance	15
2. 1 Définition de la performance	15
2. 2 Critères d'appréciation de la performance	15
2. 3 La performance dans le Cloud Computing	16
3. La réputation	16
3. 1 La réputation dans le Cloud Computing.....	17
3. 2 Approches d'évaluation de la réputation.....	17
Conclusion.....	19
Chapitre 03 : le modèle proposé	
Introduction.....	21
1. Problématique et objectifs	21
2. Notre modèle proposé	21
3. Conception	22
3. 1 Architecture.....	22
3. 2 Composants de système.....	23
4. Modèle de calcul	24
5. Calculer la confiance des nœuds	24
6. Fonctionnement de Cloud Computing Volontaire	27

6.1. Modélisation avec le diagramme UML.....	27
1) Diagramme de séquence.....	27
2) Diagramme d'activité.....	33
3) Diagramme de classe.....	34
6.2. Algorithme calculer la confiance des nœuds.....	35
Conclusion.....	36
Chapitre 04 : L'implémentation	
Introduction.....	38
1. Langage et environnement de développement.....	38
1.1. Visual Studio.....	38
1.2. Langage de programmation C #	38
1.3. MySQL.....	39
2. Les différentes classes de notre projet.....	39
2.1 La classe interface volontaire	39
2.2 La classe du calculateur.....	42
2.3 La classe du Manager.....	42
2.4 La classe d'interface administrateur.....	43
3. Présentation de différentes interfaces de volontaire.....	45
3.1 L'interface pour l'inscription.....	45
3.2 L'interface pour la connexion	46
3.3 L'interface pour choisir le mode utilisateur	46
3.4 L'interface pour choisir le mode volontaire.....	47
3.5 L'interface pour proposer de services.....	48
3.6 L'interface pour sélectionner des services.....	49
3.7 L'interface d'évaluation des services utilisés.....	50
4. Présentation de différentes interfaces d'administrateur.....	51
4.1 L'interface de connexion	51
4.2 L'interface pour estimer le coefficient des services.....	51
4.3 L'interface pour la modification des services.....	52
4.4 L'interface pour calculer la confiance et gérer les nœuds.....	53
Conclusion.....	53
Conclusion générale.....	54

LISTE DES FIGURES

Figure 1 : Cloud Computing	4
Figure 2: pile de services Cloud	5
Figure 3 : Architecture du Cloud Computing Volontaire	8
Figure 4 : Couche Middleware	9
Figure 5 : Couche physique	9
Figure 6 : gestionnaire de confiance	23
Figure 7 :Diagramme de séquence d'inscription	27
Figure 8 : Diagramme de séquence de connexion	28
Figure 9 :Diagramme de séquence d'une proposition des services	29
Figure 10 : Diagramme de séquence de demande une machine virtuelle	30
Figure 11 : Diagramme de séquence de l'évaluation	31
Figure 12 : Diagramme de séquence d'administrateur	32
Figure 13 :Diagramme d'activité	33
Figure 14 :diagramme de classe	34
Figure 15 :Algorithme calculer la confiance globale des nœuds	35
Figure 16 : Méthode pour demander de la création d'un nouveau compte	39
Figure 17 : Méthode pour Remplir les champs donnée par le manager	39
Figure 18 : Méthode pour demande un formulaire pour connecter au système	40
Figure 19 : Méthode pour la proposition des services	40
Figure 20 : Méthode pour la modification des services proposés	40
Figure 21 : Méthode pour la sélection des services	41
Figure 22 : Méthode pour l'évaluation des services	41
Figure 23 : Les méthodes de la classe calculateur	42
Figure 24: Méthode pour la validation des informations	42
Figure 25: Méthode pour La vérification des informations de la connexion	42
Figure 26 : Méthode pour la fourniture de la machine virtuelle	43
Figure 27 : Méthode d'afficher les nœuds	43
Figure 28 : Méthode pour la connexion au système	43
Figure 29 : Méthode de remplir la liste des hard composants	43
Figure 30 : Méthode d'afficher les nœuds	44
Figure 31 : Méthode pour sélectionner et supprimer les nœuds	44
Figure 32 : l'interface d'inscription	45
Figure 33 : l'interface de connexion	46
Figure 34 : l'interface pour choisir le mode utilisateur	47
Figure 35 : l'interface pour choisir le mode volontaire	47
Figure 36 : Interface proposition des services	48
Figure 37 : Liste des composants fournis par l'administrateur	49
Figure 38 : Interface sélection des services	50
Figure 39 : Interface de notation des services utilisés	51
Figure 40 : Interface de connexion	51
Figure 41 : Interface pour estimer le coefficient d'un service	52
Figure 42 : Interface de la modification des services	52
Figure 43 : Interface pour calculer la confiance et gérer les nœuds	53



LISTE DES FIGURES

La demande croissante pour de nouveaux services informatiques plus économiques a permis l'émergence d'une nouvelle architecture qui 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.

Un nouveau type a émergé de cette nouvelle architecture qui est le Cloud Computing volontaire, dans le quel le concept de Cloud Computing et de volontaire Computing sont combinés pour former le Cloud Computing volontaire. Dans le Cloud Computing volontaire, les ressources inactives ou non dédiées qui ne sont pas conçues pour être une infrastructure de Cloud sont exploitées de manière à fournir des fonctionnalités ou des services à travers le Cloud. Le Cloud Computing volontaire peuvent permet aux utilisateurs de partager tout type de service que ce soit des ressources physiques, des ressources logicielles ou des applications.

Cette technologie offre plusieurs avantages comme un déploiement rapide, Gratuit à utiliser, 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. Tous ces défis présentent un élément commun qui est la problématique de sûreté entre les volontaires et les utilisateurs, puisque le Cloud Computing volontaire exige de faire confiance aux bénévoles 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 ce mémoire, nous procédons à l'identification des défis liés à l'adoption du Cloud Computing volontaire. Nous nous intéressons en particulier aux risques liés à la confiance en proposant un système de gestion de la confiance qui compte sur la performance et la réputation pour la classification des services fiables et la sélection des nœuds efficaces dans un environnement du Cloud Computing volontaire.

Le reste du présent mémoire est organisé comme suit :

Chapitre(01) : Ce chapitre présente en détail le Cloud Computing et le Cloud Computing Volontaire. Tout d'abord, nous définissons le Cloud Computing, ainsi que ses différents types, ses modes de déploiement, la virtualisation. Ensuite, nous présentons le Cloud Computing Volontaire, son l'architecture qui contient de différente couches, et nous concluons ce chapitre par les avantages et les convenaient du Cloud Computing Volontaire.

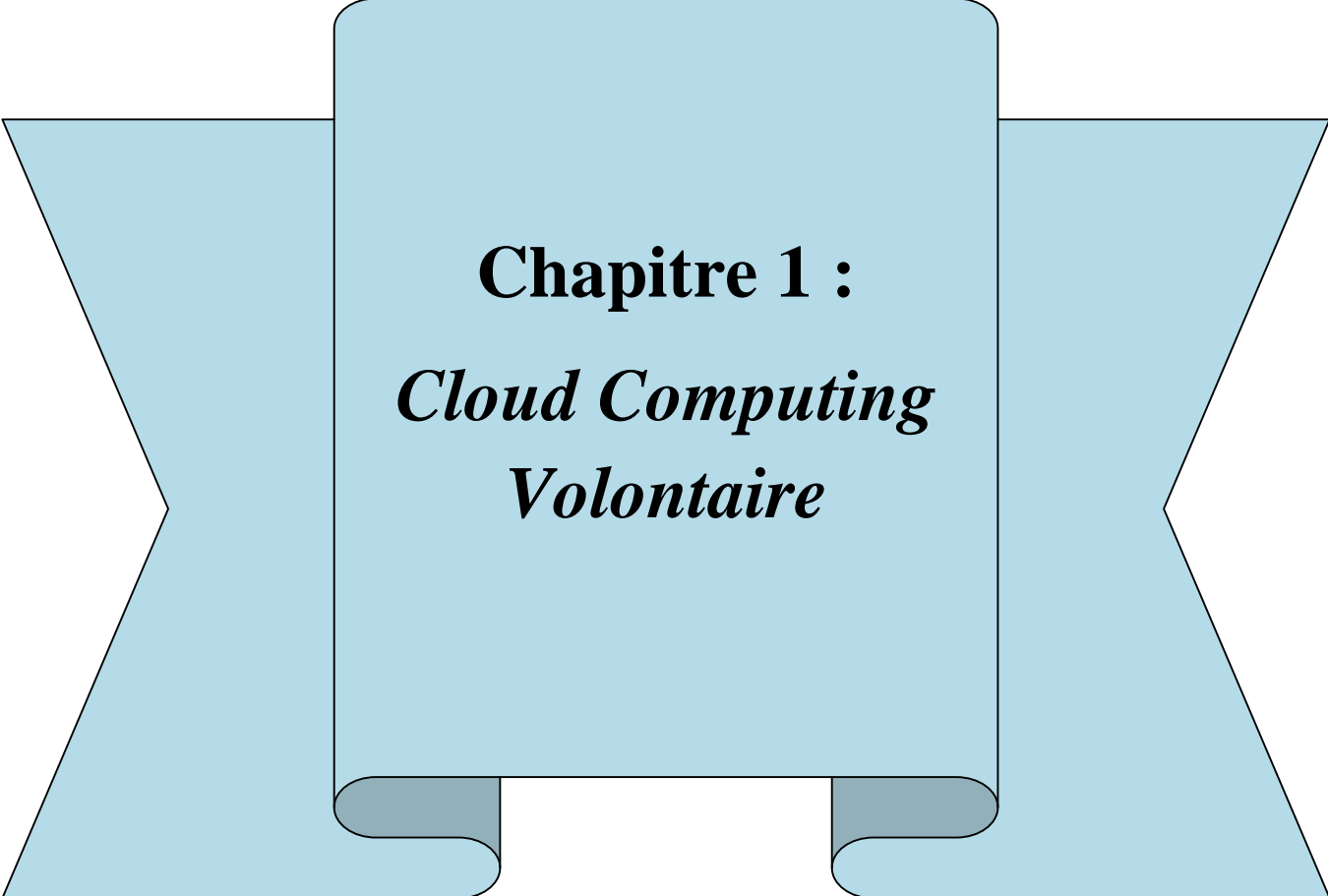
Chapitre(02) : Nous présenterons, dans la première partie de ce chapitre, les différents concepts liés à la confiance, performance et réputation. Puis, nous présenterons dans la deuxième partie, des différents modèles de ses concepts.

Chapitre(03) : Dans ce chapitre nous allons proposer une approche pour ce type de problème, cette approche est basée sur un modèle computationnel d'opinion, c'est-à-dire que nous allons calculer la valeur de la performance qui est basée sur l'évaluation de la QoS (Quality of service), et calculer la valeur de la confiance qui est basée sur l'évaluation de la sécurité des données

INTRODUCTION GENERALE

Chapitre(04) : nous allons décrire, au sein de ce chapitre l'implémentation de ce système proposé qui sera réalisé sous l'environnement c#. En outre, ce chapitre détaille les classes et les interfaces de notre système.

Enfin, ce manuscrit se termine par une conclusion générale qui récapitule le travail réalisé et propose quelques perspectives.



Chapitre 1 :
Cloud Computing
Volontaire

Introduction

Le Cloud Computing volontaire est un nouveau type de Cloud visant à transférer le Computing volontaire vers le Cloud. Ce nouveau type de Cloud est motivé par le fait que la construction d'un Cloud, à partir de ressources non dédiées, peut être utile pour des projets scientifiques qui ne peuvent pas supporter le coût de la consommation des services Cloud fournis par les fournisseurs de services Cloud tels qu'Amazon. Cependant, les Clouds Volontaires en sont à leurs débuts avec quelques défis et problèmes qui devraient être abordés. Ce chapitre présente en détail le Cloud Computing et le Cloud Computing Volontaire. Tout d'abord, nous définissons le Cloud Computing, ainsi que ses différents types, ses modes de déploiement, la virtualisation. Ensuite, nous présentons le Cloud Computing Volontaire, son l'architecture qui contient de différente couches, et nous concluons ce chapitre par les avantages et les convenaient du Cloud Computing Volontaire.

1. Cloud Computing

1.1. Définition

La plus part des définitions relatives au Cloud Computing incluent la notion de services disponibles à la demande, et l'extensibilité. En contradiction avec les systèmes actuels, les services sont virtuels et illimités et les détails des infrastructures physiques sur lesquels les applications reposent ne sont plus dures sort de l'utilisateur [1]. Ainsi, le NIST (National Institute of Standards and Technology) définit Le Cloud Computing comme étant « un modèle qui permet un accès réseau à la demande et pratique à un pool partagé de ressources informatiques configurables (telles que réseaux, serveurs, stockage, applications et services) qui peuvent être provisionnées rapidement et distribuées avec un minimum de gestion ou d'interaction avec le fournisseur de services [2]», et CISCO le défini comme étant «une plateforme de mutualisation informatique fournissant aux entreprises des services à la demande avec l'illusion d'une infinité des ressources[3]».

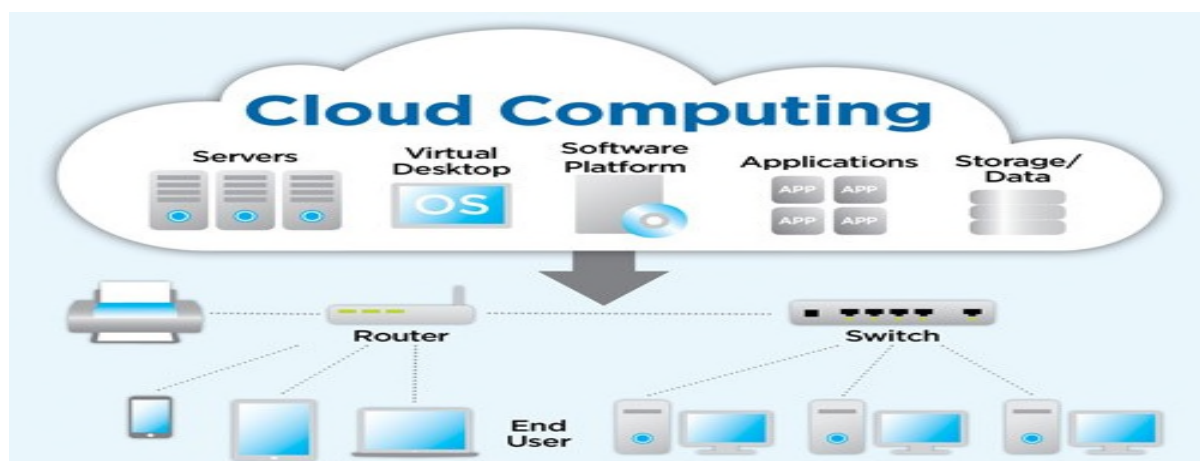


Figure 1 : Cloud Computing [6]

Comme le montre la figure (1), le Cloud Computing est un concept qui consiste à déporter sur des serveurs distants des stockages et des traitements informatiques traditionnellement localisés sur des serveurs locaux ou sur le poste de l'utilisateur. Il consiste à proposer des services informatiques sous forme de service à la demande, accessible de n'importe où, n'importe quand et par n'importe qui. Ainsi, le Cloud n'est pas un ensemble de technologies, mais un modèle de fourniture, de gestion et de consommation de services et de ressources informatiques localisées dans des Datacenter [1].

1.2. Services du Cloud Computing

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) 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.[4]




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

Figure 2 : pile de services Cloud [9]

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 [4].

Salesforce, qui s'appuie sur le modèle SaaS, offre des applications de productivité d'entreprise CRM (Customer Relationship Management) qui réside 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 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 [4].

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 offre principalement des services IaaS, qui dans le cas de son service 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]

1.3. Virtualisation

La Virtualisation a été la première pierre vers l'ère du Cloud Computing, c'est une technologie qui permet une gestion optimisée des ressources matérielles en disposant de plusieurs machines virtuelles sur une machine physique. Certains noms tels que VMware ou Virtual Box sont désormais relativement familiers.

Aujourd'hui la plupart des serveurs possèdent des ressources matérielles importantes, mais celles-ci sont rarement utilisées à 100%. Il est fréquent qu'une entreprise dispose d'une quinzaine de serveurs fonctionnant chacun à 15% de ses capacités [5]. C'est pourquoi les prestataires d'offres dans le Cloud y installent des outils de virtualisation qui vont permettre de disposer, sur une machine, de plusieurs systèmes d'exploitation.

Le principe de virtualisation permet aussi d'être plus flexible dans l'allocation des ressources, par exemple si une machine virtuelle manque de ressources car le serveur physique sur lequel elle repose n'est pas ou plus assez puissant il est dès lors très simple de virtualiser cette machine virtuelle sur un autre serveur plus puissant.

1.4. 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 [4]:

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 (virtual private network).

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.

1.5. Avantages et défis

Le Cloud offre plusieurs avantages tel que : le 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 [4].

2. Cloud Computing Volontaire

2.1. Définition

Le VCC (Cloud Computing Volontaire) combine les concepts de Cloud Computing [6] et de Computing volontaire [7] en offrant tout ou partie des services de Cloud sans frais. VCC a quelques avantages par rapport aux Clouds commerciaux. Le premier avantage est la rentabilité de Cloud de volontaires puisque toutes les ressources sont offertes volontairement, ce qui peut être très utile pour les projets qui ne peuvent pas se permettre des services commerciaux de Cloud [8]. Deuxièmement, il réduit la consommation d'énergie et les émissions de gaz parce qu'il utilise des ressources informatiques qui, autrement, resteraient inutilisées. Ainsi, le pourcentage moyen de ressources locales inactives dans une organisation est d'environ 80% [9]. En revanche, les Clouds commerciaux mettent en place un grand nombre de ressources dédiées dans leurs centres de données, ce qui a un impact négatif sur l'environnement puisque leurs centres de données consomment des quantités massives d'électricité [10]. Enfin, les Clouds commerciaux sont inefficaces en termes de mobilité des données et accordent peu d'attention à la localisation des clients [11].

2.2. Architecture

Le niveau abstrait de l'architecture, figure (3), divise VCC en trois couches : une couche de service; une couche de middleware; et une couche physique.

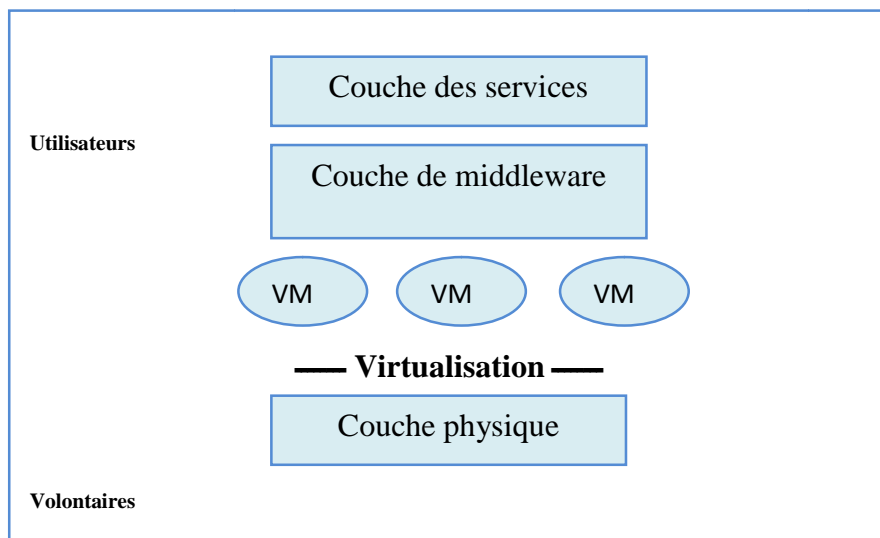


Figure 3 : Architecture du Cloud Computing Volontaire [10]

1) Couche de service

La couche de service est concernée par la fourniture de services aux clients, d'une manière similaire aux Cloud Computing, via une interface basée sur l'approche SOA (Service Oriented Architecture). Les contributeurs de VCC donnent volontairement leurs ressources pour former un VCC pendant un certain temps, et ils peuvent être des consommateurs de services en même temps s'ils le souhaitent.

2) Couche Middleware

Le but de cette couche est de fournir des ressources à la couche des services tels qu'elles seraient fournies par un Cloud Computing. La couche, illustrée dans la figure (4), comprend la gestion des tâches, la gestion de la qualité de service et l'auto-automatisation. Le planificateur de tâches organise les tâches provenant de la couche de service en les transmettant aux ressources appropriées qui sont proposées dans la couche physique. Alors que L'équilibrage de charge garantit que la charge est distribuée de manière appropriée, ce qui minimise le temps requis pour traiter une tâche. L'auto-automatisation aide à fournir l'élasticité rapide dans les VCC. Il permet aux utilisateurs d'augmenter ou de réduire les services en fonction de leurs besoins. La gestion de la qualité de service garantit qu'un niveau de qualité minimum est maintenu.

Le moniteur de performances de la gestion de la qualité de service s'assure que les performances de chaque tâche sont maintenues à un niveau acceptable signalé dans le composant de rapport SLA (Service Level Agreement).

Dans le VCC la volatilité des nœuds est assez élevée. Ainsi, le moniteur de performances doit donc coopérer avec la gestion des ressources pour trouver des nœuds fiables parmi les ressources disponibles pour chaque tâche.

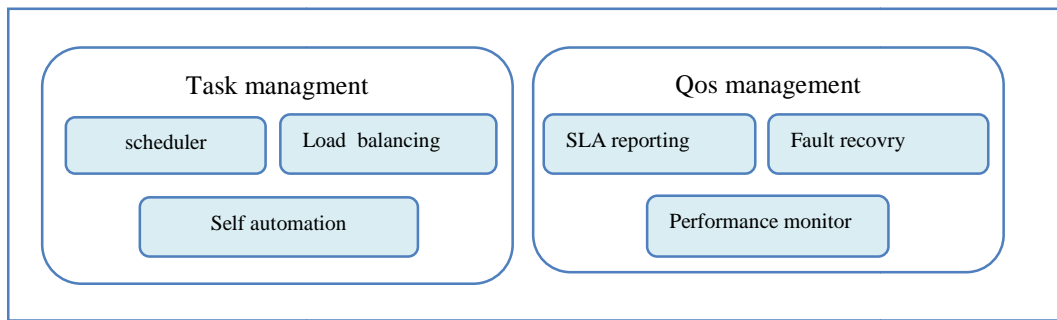


Figure 4 : Couche Middleware [10]

3) Couche physique

Le gestionnaire de ressources, Figure (5), est responsable de l'agrégation des ressources, de l'allocation des ressources et de la surveillance des ressources. Il regroupe les nœuds volontaires désignés par le public. Un mécanisme d'agrégation permet de classer les ressources selon un certain nombre de critères dans le but d'optimiser la qualité du service. Par exemple, l'historique de chaque nœud volontaire peut être utile en termes de reconnaissance du nœud qui doit être sélectionné par l'allocateur de ressource pour chaque tâche. L'allocateur reçoit des tâches de la gestion des tâches et les alloue aux ressources requises. L'allocateur peut diminuer l'interruption des services en attribuant des tâches aux nœuds avec une fiabilité plus élevée. Le composant de surveillance observe régulièrement les ressources allouées au cas où l'une d'entre elles deviendrait indisponible. Dans ce cas, le moniteur informe la récupération d'erreur dans le middleware afin de récupérer la tâche à partir d'un nœud répliqué.

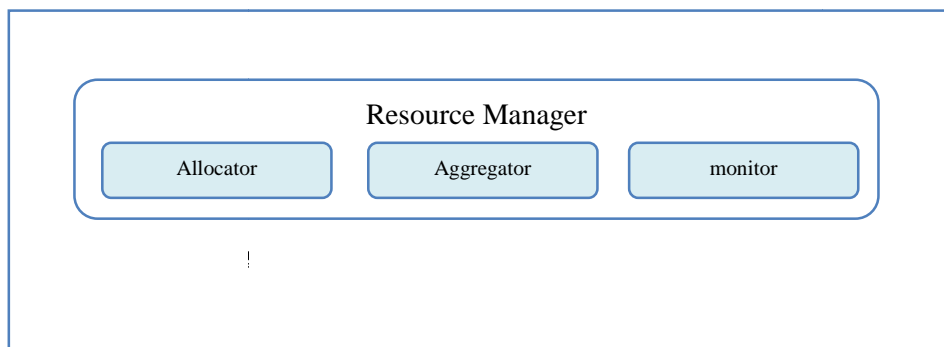


Figure 5 : Couche physique

2.3. Avantages et inconvénients

1) Avantages

- La possibilité de déployer et de rendre disponibles des services de manière immédiate.
- Les données partagées, puisque tout utilisateur du Cloud Computing Volontaire peut rendre disponibles ses données à un ou plusieurs autres utilisateurs du VCC.
- Un accès libre et ouvert au utilisateur, qui peut établir sa connexion de n'importe où et avoir accès à ses données immédiatement, sans passer par la mise en place d'un VPN dans l'entreprise.

- Une liberté totale, puisque l'utilisateur n'est lié à les volontaires de service par aucun engagement à long terme. libre de mettre un terme à un service à tout moment, s'il juge n'en avoir plus besoin, ou s'il désire simplement changer de service.
- Gratuit à utiliser, le volontaire fournit ses services via le Cloud volontaire et peut demander d'autres services gratuits.

2) Inconvénients

- la confidentialité et de la sécurité des données d'utilisateur, Le risque de voir les données des utilisateurs finir en situation de vol ou de mauvaise utilisation demeure donc une possibilité.
- la confiance aux services fournis, possibilité de fournir des services non fiables et non efficaces.
-

Conclusion

Dans ce chapitre, nous avons présenté l'environnement Cloud Computing en général et l'environnement de Cloud Computing Volontaire en particulier. Le Cloud Computing Volontaire 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 qui permet aux utilisateurs de bénéficier de nombreux services fournis par les machines de volontaires. Le Cloud Computing Volontaire apparaît donc comme une opportunité intéressante mais pose logiquement la question de la confiance et performance des services quand elles sont fournies par les volontaires.

Les interrogations liées à la confiance et à la performance des services sont le principal frein à son adoption. Par conséquent, l'enjeu essentiel du Cloud volontaire revient à recueillir la confiance des utilisateurs. À cet effet, plusieurs modèles de calcul de confiance ont été proposés afin de résoudre cette problématique. Ainsi, le chapitre suivant sera réservé aux modèles de la confiance et de la réputation.



Chapitre 2 :

*La Confiance ,la
performance &
La réputation*

Introduction

La gestion de la confiance, la réputation et la performance attire de plus en plus l'attention des experts en sécurité. C'est un moyen important qui est utilisé pour améliorer la sécurité dans les systèmes critiques. Les notions de confiance, réputation et performance proviennent des sciences sociales qui étudient la dynamique des sociétés humaines. Le concept de confiance a été étudié dans différentes disciplines de l'économie à la psychologie, de la sociologie à l'information et à l'informatique.[12]

1. La confiance

La confiance est un concept utilisé dans de nombreuses disciplines. En sciences sociales, l'accent est mis sur l'impact de la confiance pour la prise de décision. En informatique, on se concentre sur la conception d'outils pour l'assistance des utilisateurs dans diverses tâches [4].

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 les différences entre les modèles, nous allons d'abord définir la confiance et les concepts qui s'articulent autour de ce terme.

1.1.Définition de la confiance

Le concept de la confiance a attiré beaucoup de chercheur. Ainsi, dans la littérature certains auteurs se sont distingués dans leurs recherches en publiant des travaux signifiant sur la confiance dans leur domaine [13]. Ainsi, dans le chercheur se base sur une perception individuelle de la confiance, en parallèle à d'autre recherche où il place ce concept dans une dimension sociale [14] tandis que dans[15] la confiance est vue comme la prévision d'un comportement[4].

La confiance est définit comme une décision par rapport à une perception individuelle des coûts et bénéfices dont dépend cette décision [13]. 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 [4].

Bien que dans [14] la confiance est présentée comme un fait basique de la vie humaine, elle 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 comme dans [13], tandis que pour d'autres la situation est très importante comme dans [16] et dans [17]. D'autres définissent la confiance comme la prévision de comportements. En effet, dans [18] la confiance est vue comme une probabilité subjective sur les futurs événements.

Dans le domaine informatique, la confiance est définie comme un engagement à croire au bon déroulement des futures actions d'une autre entité[19]. Selon [20], 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 les trois éléments ci-dessous sont présents dans chaque définition [4]:

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

1.2. Caractéristique de confiance

Les caractéristiques communes les plus utilisées pour la confiance sont [4]:

- Bienveillance : qui, représente la motivation pour agir dans l'intérêt d'une autre personne.
- Intégrité : qui, consiste à conclure un accord avec bienveillance.
- Compétence : qui représente les aptitudes ou capacités nécessaires pour faire ce que nous devons faire.
- Prévisibilité : c'est pourvoir prévoir les situations futures, à partir des actions assez conformes d'autres entités.

-

1.3. Principes de confiance

Pour garantir un niveau maximal de confiance certains principes doivent suivis [4]:

- 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.
- 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 intégrer les risques associés.
- La confiance est déterminée par le temps : elle est construite au fil du temps en se basant sur les expériences passées.
- La confiance peut être mesurée : elle est mesurable par une valeur numérique, généralement dans l'intervalle [0 - 1].

1.4. Facteurs influant sur la confiance

La valeur calculée de la confiance est influencée par quelques facteurs [21]. Ainsi:

- 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.
- 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 différents 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[4].

1.5. Types de confiance

Les différentes classes de confiance sont [22] :

- **Confiance de provision** : Décrit la confiance d'une tierce partie à un service ou à un fournisseur de service.

- **Confiance d'accès** : Décrit la confiance avec le principe d'accès aux ressources propriétaires par ou sous la responsabilité d'une tierce partie.
- **Confiance de délégation** : Décrit la confiance en une entité déléguée qui agit et prend des décisions au nom d'une entité.
- **Confiance d'identité** : Décrit la croyance qu'une identité d'une entité et celle qu'elle prétend être.
- **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.

1.6. Modèles computationnels de confiance

Ces dernières années, plusieurs modèles de confiance ont été proposés se basant sur diverses considérations et perspectives pour traiter le problème de sélection de service. Dans cette section nous présentons quelques modèles :

1) TidalTrust

TidalTrust [23] est un algorithme qui, est testé dans le réseau social (FilmTrust), afin de recommander des films. Ce système est utilisé par 400 utilisateurs, et chacun d'eux évalue les 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]. 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.

2) Logique certaine

Dans [24] les auteurs ont proposé un modèle pour évaluer les termes de logique propositionnelle avec l'incertitude. Il fournit une représentation plus simple puisqu'il est basé sur des paramètres indépendants. 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.

3) Modèle de Jøsang : bêta réputation

Le modèle de confiance **bêta réputation** [25] repose sur la théorie de probabilité bayésienne ainsi que la théorie de la croyance. 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 dans l'intervalle [0..1].

4) Modèles basés sur la croyance: belief model

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 [25].

2. Performance

La notion de performance est une notion polysémique, rarement définie de façon claire par les auteurs. Elle est un concept englobant et intégrateur, donc difficile à définir de façon précise [26].

Nous avons identifié plusieurs définitions acceptables relatives à la performance en fonction du domaine considéré et du contexte d'utilisation. Néanmoins, nous allons essayer de présenter les définitions les plus répandues [27].

2.1. Définition de performance

Dans [28] la performance peut être définie comme « la réalisation des objectifs organisationnels, quelles que soient la nature et la variété des objectifs ». Cette réalisation pouvant se comprendre au sens strict (résultat, aboutissement) ou au sens de processus qui mène au résultat (action).

A travers la définition de [29], la performance correspond à l'efficacité, l'efficience et l'économie. On peut définir l'efficacité comme l'atteinte de l'objectif visé, tandis que l'efficience est le fait d'atteindre l'objectif visé avec le minimum de quantité de ressources possible. L'économie est l'acquisition des ressources au coût le plus faible. En plus de ces trois autres notions prises en compte dans la définition de performance, le terme pertinence est rajouté dans [30], il est défini comme étant l'articulation entre les objectifs et les moyens.

2.2. Critères d'appréciation de la performance

Le contrôleur de gestion doit fournir un ensemble de critères d'appréciation qui, contribueront au pilotage de la performance globale de l'organisation en fonction de ses objectifs stratégiques. Les critères d'appréciation de la performance informent sur le résultat obtenu et sur la façon dont il a été obtenu. De façon générale, la performance peut être appréciée suivant un certain nombre de critères qui sont [27] :

- L'efficacité ;
- L'efficience ;
- L'économie ;
- L'effectivité ;
- L'équité.

1) L'efficacité

L'efficacité est la capacité à atteindre l'objectif, c'est-à-dire à atteindre un résultat conforme à l'objectif [31]. Elle représente le rapport entre les résultats atteints par un système et les objectifs fixés.

2) L'efficience

L'efficience est définie comme la mise en œuvre du minimum de ressources nécessaires pour le résultat obtenu [31].

3) L'économie

L'économie est le fait de se procurer au moindre coût les ressources nécessaires. Elle est définie comme étant l'acquisition de ressources financières, humaines et matérielles appropriées, tant sur le plan de la qualité que celui de la quantité au moment, au lieu et au coût le moindre [32].

4) L'effectivité

L'effectivité va plus loin dans l'évaluation en se prononçant sur le trio : objectif, moyens et résultats [29]. Il s'agit de la proportion d'objectifs atteints par rapport aux objectifs visés. L'effectivité permet de voir si ce qui avait été prévu de faire est effectivement fait.

5) L'équité

Ce concept favorise à toute organisation le respect et la cohésion sociale [33]. C'est le principe qui conduit à corriger des inégalités que subissent des personnes défavorisées au sein d'une organisation. L'équité doit être un critère d'appréciation de la performance, en ce sens qu'elle permet d'apprécier la capacité d'une entreprise, à gérer son climat social de la meilleure façon qu'il soit, pour créer de bonnes conditions de travail et ainsi permettre à ses employés d'être le plus performant possible.

2.3. La performance dans le Cloud Computing

Le Cloud Computing est l'un des systèmes distribués qui traite des ressources dispersées et repose sur le traitement de données volumineuses. Ce besoin de contrôle de performance et d'analyse est nécessaire pour les développer, c'est-à-dire que le contrôle du Cloud est important pour les utilisateurs et les fournisseurs. Il fournit des informations et des indicateurs de performance, lorsque ce niveau de performance affecte l'adoption des services Cloud par les utilisateurs. Les performances font partie des avantages qui devraient être disponibles dans les services Cloud, car les performances ont un impact sur les utilisateurs et les fournisseurs de services. Pour évaluer les performances, nous devons prendre en compte plusieurs critères pour évaluer les facteurs affectant les performances du Cloud Computing, notamment la réponse moyenne par unité de temps et le temps d'attente moyen par unité de temps et autres dans les propriétés des mesures de performance du Cloud [34].

- (SaaS) L'évaluation est faite par les utilisateurs directement en fonction des mesures de performance, de la rapidité de réponse, de la fiabilité des services techniques et de la disponibilité.
- (PaaS) L'évaluation est effectuée directement ou indirectement par les utilisateurs en fonction des mesures de performance en fonction de la technicité, de la productivité, de la fiabilité, du service technique et de la capacité du middleware.
- (IaaS) Les mesures de performance sont déterminées en fonction des performances de l'infrastructure, de la capacité, de la fiabilité, de la disponibilité et de l'évolutivité.

3. La réputation

3.1. Définition de la réputation

La réputation est de notoriété publique et représente l'opinion collective des membres d'une communauté. La réputation est basée sur l'opinion de confiance agrégée d'un groupe d'agents. Comme la confiance est hautement subjective, ce résultat agrégé peut ne pas être d'égale utilité pour tous les agents [12].

La différence entre la confiance et la réputation c'est que la confiance est finalement un phénomène personnel et subjectif qui repose sur divers facteurs ou évidences. L'expérience personnelle a généralement plus de poids que les renvois de confiance ou la réputation de seconde main, mais en l'absence d'expérience personnelle, la confiance doit souvent être basée sur les références d'autres personnes. Alors que la réputation est généralement définie comme la quantité

de confiance inspirée par un membre particulier d'une communauté dans un cadre ou un domaine d'intérêt spécifique. Les membres qui ont une bonne réputation, car ils contribuent utilement à la communauté de vie, peuvent utiliser les ressources, tandis que les membres ayant une mauvaise réputation, parce qu'ils refusaient de coopérer, sont progressivement exclus de la communauté. Ainsi, les systèmes de réputation peuvent être appelés systèmes de sanction collaborative pour refléter leur nature collaborative, et sont liés aux systèmes de filtrage collaboratif. Ces systèmes sont déjà utilisés dans des applications commerciales en ligne réussies [12].

3.2. La réputation dans le Cloud Computing

Les développements rapides du Cloud Computing signifient que les services Cloud sont devenus le principal mode de calcul sur Internet. De nombreux services ont été déployés pour fournir des fonctionnalités similaires. Cependant, le problème de l'identification de services fiables a attiré l'attention des chercheurs. Ainsi, les concepts de confiance et de réputation ont été introduits pour évaluer la fiabilité des services Cloud.

La réputation est une évaluation subjective d'un service de Cloud, basée sur l'expérience individuelle ou les recommandations d'autres utilisateurs. La réputation et la confiance sont dynamiques, ce qui rend la construction d'une norme d'évaluation difficile. En outre, la survenance d'évaluations malveillantes, d'éloges délibérés et de préférences personnelles des utilisateurs signifie qu'une valeur de réputation standardisée peut différer de la véritable valeur.

Récemment, divers systèmes de révision de réputation ont été proposés pour relever les défis posés par les environnements de services Cloud ouverts et dynamiques. La plupart de ces systèmes sont axés sur le calcul des notations de réputation, la gestion de la réputation, l'expérience et d'autres caractéristiques des environnements dynamiques qui pourraient constituer une référence appropriée pour les utilisateurs. Cependant, l'existence de notations injustes affecte grandement l'exactitude des évaluations de confiance. Actuellement, ces modèles de réputation tendent principalement à l'exactitude des évaluations de confiance ; Cependant, ces méthodes existantes sont limitées par la préférence de la personnalité. [35]

3.3. Approches d'évaluation de la réputation

Plusieurs approches d'évaluation de la réputation ont été proposées pour les systèmes distribués. Dans cette section nous présentons quelques approches [35]:

- 1) Dans [36], les auteurs ont proposé un mécanisme de réputation distribuée basé sur un certain nombre de centres de réputation spéciaux. Chaque centre de réputation spécial a recueilli des informations sur la réputation pour des services prédéterminés offerts par différents fournisseurs de services. Cependant, cette méthode était encore quelque peu centralisée.
- 2) Les auteurs dans [37] ont proposé un mécanisme de contrôle des ressources décentralisé, qui a introduit une architecture de Cloud Computing orientée vers le marché. Cependant, le système de réputation était vulnérable au blanchiment d'argent, au retour d'informations incorrectement rapporté et aux attaques de collusion (où plusieurs utilisateurs coordonnent leurs commentaires pour manipuler les informations de réputation).
- 3) Le mécanisme de confiance distribuée a été proposé dans [38]. Il calcule deux valeurs de réputation (pour un vendeur et un acheteur) pour chaque nœud de manière itérative sur la base de l'historique des transactions. Ce mécanisme pourrait rapidement réduire les valeurs de réputation

des nœuds malveillants et empêcher les attaques par collusion. Pour construire un environnement informatique fiable.

4) Une méthode de révision des valeurs de réputation qui calcule la réputation en fonction de la différence entre la qualité de service annoncée QoS (Quality of Service) fournie par les fournisseurs de services et les évaluations faites par les consommateurs a été proposée dans [39]. Ensuite, les consommateurs ont été triés en fonction des notations de réputation qu'ils ont fournies et les consommateurs qui pourraient être impliqués dans la collusion ont été exploités en utilisant un algorithme de règles d'association. Enfin, la réputation mise à jour a été recalculée et enregistrée dans le centre de réputation.

Conclusion

Dans ce chapitre nous avons établi une relation entre la confiance, la performance et la réputation est positive. La réputation et le niveau de performance affectent la confiance, et la performance et la réputation affectent la confiance du client dans l'adoption du Cloud Computing, et ceci est un indicateur qui affecte le développement et la progression du Cloud Computing. Par conséquent, l'enjeu essentiel du Cloud revient à recueillir la confiance des utilisateurs d'une part, et le progrès et le développement du Cloud Computing d'autre part.

Le chapitre suivant présentera le modèle proposé relatif à la confiance et à la réputation dans le Cloud Computing Volontaire.



Chapitre 3 :

L'approche proposé

Introduction

La complexité du Cloud Computing Volontaire rend le problème de la confiance lors de l'utilisation des services fournis par les volontaires, d'une importance primordiale pour les utilisateurs et les Volontaires de services. Dans ce chapitre nous allons proposer une approche pour ce type du problème, cette approche est basée sur un modèle computationnel d'opinion, c'est-à-dire que nous allons calculer la valeur de la performance qui est basée sur l'évaluation de la QoS et calculer la valeur de la confiance qui est basée sur l'évaluation de la sécurité des données.

1. Définition du Problème

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

Bien que le Cloud Computing volontaire ait connu une évolution rapide, il reste encore plusieurs barrières telles que la confiance dans l'utilisation des services, qui découragent les utilisateurs à adopter ce système, où la confiance a été répertoriée comme une préoccupation plus importante dans le Cloud Computing volontaire.

De plus, lorsque l'on considère d'adopter un service en Cloud volontaire, la première question que tout utilisateurs se pose est « Jusqu'à quel point l'efficacité des services proposés par les volontaires ? ». Même si l'erreur est non-intentionnelle, il est capable de détruire la confiance des utilisateurs.

2. Objectifs

Notre objectif est de proposer une approche pour la gestion de la confiance basée sur l'évaluations de la performance d'un service afin de sélectionner les services les plus fiables et la sélection des nœuds en fonction de leur efficacité, en examinant les considérations de qualité de service. Pour cela, plusieurs questions sont posées:

Question 1 : Quelles qualités de service influent sur la fiabilité et la sûreté de service?

Question 2 : Comment amener les utilisateurs à faire confiance aux volontaires de service?

Question 3 : Est-il possible que les volontaires se fassent 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 volontaire de services dans un environnement du Cloud Computing volontaire? »

3. Notre modèle proposé

Dans la plupart des travaux présentés dans le deuxième chapitre, la confiance a été calculée à partir des avis d'utilisateurs selon divers méthodes. 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 proposons, dans ce projet de fin d'étude, une approche qui intègre ces deux moyens pour calculer une valeur globale de la confiance.

L'objectif de notre approche est de se concentrer sur l'estimation de la valeur de la confiance globale d'une machine virtuelle, de classer les services les plus fiables, et d'autre

part de sélectionner les nœuds les plus efficaces dans le Cloud Computing volontaire. Cette approche est basée sur la performance et la confiance des services à travers les opinions des utilisateurs.

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

- 1) représentation de la confiance Globale des services: les mesures de la performance et de confiance utilisées.
- 2) calcul de la confiance globale des nœuds: le modèle mathématique pour l'agrégation des notes de confiance globale.

Afin de répondre à cet objectif, l'approche proposée se compose de deux étapes afin de calculer la confiance globale qui dépend du calcul du taux d'évaluation de la performance et la confiance du service :

etape1 - calculer la performance: Elle dépend du calcul du taux l'évaluation de la QoS (quality of services) à savoir : le temps de réponse(CPU), le stockage mémoire vive(RAM), traitement graphique (GPU), et les applications.

Etape2 - calculer la confiance : dépend du calcul du taux l'évaluation de la sécurité des données contenues dans disque dure (HDD).

4. Conception

4.1. Architecture

La figure (8) décrit les composants principaux du système de confiance, qui consistent en trois différents niveaux à savoir : l'administrateur, volontaire de service, gestionnaire de service.

1) L'administrateur

Ce niveau est responsable de remplir une liste des services dans le système, ainsi que de sélectionner les nœuds les plus effectifs et supprimer les moins efficaces.

2) Le volontaire de service

Ce niveau est constitué de deux types de volontaires : nouveau volontaire et ancien volontaire.

Le volontaire fourni des services, et peut demander des services. Dans ce dernier cas il donne son avis sur les services utilisés.

3) Le gestionnaire de service

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

Le calculateur de la confiance globale constitue l'élément principal de ce niveau. Il est responsable de calculer la confiance globale d'un nœud en se basant sur le modèle de calcul proposé.

Le manager est en charge de l'échange des ressources entre les volontaires et le manager. Il classifie les services les plus fiables.

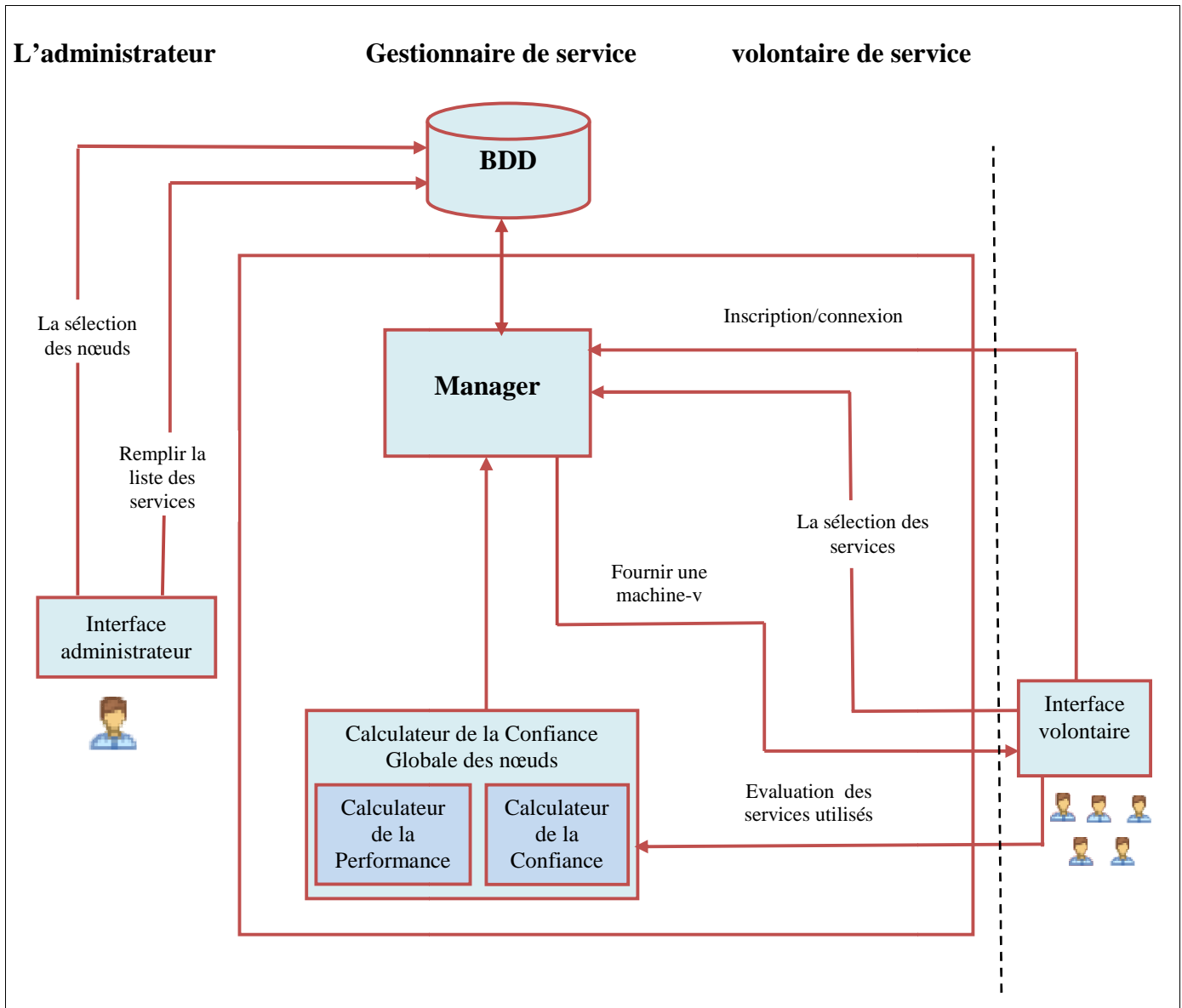


Figure 6 : gestionnaire de confiance

4.2. Composants de système

Le système est composé de divers composants à savoir :

1) Volontaire de service :

Ce composant contient une interface à travers laquelle le nouveau volontaire peut s'inscrire, puis il peut connecter pour proposer des services à fournir. Les anciens volontaires se connectent au système pour proposer d'autres services, supprimer des services, ou bien sélectionner des services afin de les utiliser. Les services choisis forment une machine virtuelle. Après l'utilisation de la machine virtuelle, le volontaire doit évaluer les services utilisés.

1) Services de base du modèle:

Les principaux services du modèle constituent le gestionnaire de la confiance qui est responsable de calculer de la confiance globale des nœuds et il est responsable de classer les services selon leurs fiabilités. Ainsi, le gestionnaire de la confiance se compose de composants suivants:

- Évaluation des services : ce composant permet une évaluation basée sur les opinions des utilisateurs en ce qui concerne la QoS, le temps de réponse, Taille mémoire vive, le traitement graphique, les applications, et d'autre part les opinions sur la sécurité des données stockées sur le disque dur. L'évaluation est entre [1-9].
- Calculateur de confiance globale d'un nœud: ce calculateur permet de calculer les valeurs subjectives à partir des opinions des utilisateurs sur les services fournis par le même volontaire. Le calculateur prend en considération le calcul du taux de l'évaluation de la QoS, et le taux de l'évaluation de la sécurité des données. Ce calculateur est basé sur le modèle d'opinion.
- Manager : ce composant permet de :
 - Stocker et vérifier les informations et les services fournis par les volontaires dans la base de données.
 - Fournir les machines virtuelles aux volontaires après la sélection des services souhaités.
 - Stocker les valeurs finales des évaluations et de la confiance globale des nœuds dans la base de données.
 - Classer les services selon leurs fiabilités.
 - sélectionner les nœuds les plus efficaces.
- Base de données : responsable de stocker les informations des volontaires et les services qu'ils fournissent, et de stocker les évaluations des services et la confiance globale des nœuds.

2) L'administrateur :

- Interface administrateur : l'administrateur se connecte au système.
- Remplir la liste des services : l'administrateur du système remplit une liste des services que les utilisateurs sont censés demander.
- La sélection des nœuds : après la classification des nœuds l'administrateur peut supprimer les nœuds les plus faibles et empêcher le volontaire de se connecter.

5. Modèle de calcul

Le modèle de la gestion de la confiance proposé se base sur un modèle computationnel pour la classification des services et la sélection des nœuds les plus efficaces.

Dans ce modèle :

- Les services sont évalués à travers les opinions des volontaires ;
- Les évaluations sont basé sur la QoS ;
- Les notes des évaluations sont entre [0-9] ;

5.1. Calculer la confiance des nœuds

Pour calculer la confiance globale des nœuds, nous proposons les étapes suivantes:

- calculer la valeur de la performance en se basant sur les notes de l'évaluation des volontaires relatives à la vitesse du CPU, le traitement graphique du GPU, le stockage de la RAM, et les applications;
- calculer la valeur de la confiance en se basant sur les notes de l'évaluation des volontaires relatives à la sécurité des données enregistrées dans le disque dur.

Les valeurs de ces deux calculs seront agrégées afin de donner la valeur de confiance globale.

1) Calcul de valeur de la performance des hard composants

- **Total des notes de la performance** : représente la valeur de la performance des différents composants de l'infrastructure physique. Ces composants représentent les services proposés par le même volontaire (CPU, GPU, RAM). Elle est calculée selon la formule ci-dessous:

$$P_m(s) = \sum_{i=1}^n (\text{note}(i) * \text{coef}(i))$$

P_m : la valeur de la performance des services (composants matériels);

s : le nœud volontaire ;

n : nombre total des services fournis par le volontaire s ;

Note(i) : Note de l'évaluation du service utilisé i fourni par le volontaire s . Elle appartient à l'intervalle $[0, 9]$;

coef : chaque service a un coefficient ;

- **Total des coefficients pour la performance**: représente le coefficient de la performance des différents composants de l'infrastructure physique (CPU, GPU, RAM) du même nœud. Elle est calculée selon :

$$P_c(s) = \sum_{i=1}^n \text{coef}(i)$$

P_c : la somme de coefficients des services fournis ;

s : le nœud volontaire ;

n : nombre total des services fournis par le volontaire s ;

Coef(i) : Le coefficient du service i .

- **La valeur de la performance totale** :

$$P_1(s) = P_m(s) / P_c(s) ;$$

2) Calculer la valeur de la confiance

- **Total des notes pour la confiance** : représente le montant de la performance des différents composants de l'infrastructure physique (disque dure) proposés par le même volontaire. Elle est calculée selon :

$$C_m(s) = \sum_{i=1}^n (\text{note} * \text{coef})$$

C_m : la valeur de la confiance des services (Disques Dure) fournis par le volontaire s ;

s : le nœud volontaire ;

n : nombre total des services ;

note(i): l'évaluation du service utilisé i fourni par le volontaire s . La note $\in [0, 9]$;

Coef(i): Le coefficient du service i .

- **Total des coefficients pour la confiance** : représente le coefficient de la confiance des différents composants de l'infrastructure physique (disque dur) proposés par le même volontaire. Elle est calculée selon :

$$C_c(s) = \sum_{i=1}^n \text{coef}(i)$$

C_c : la somme des coefficients de la confiance des services ;

s : le nœud volontaire ;

n : nombre total des services ;

coef(i): Le coefficient du service i .

- **La valeur de la confiance totale** :

$$C(s) = C_m(s) / C_c(s) ;$$

3) Calcul de valeur de la performance des softs composants

- **Total des notes de la performance** : représente la valeur de la performance des différents composants soft, ces services proposés par le même volontaire (applications). Elle est calculée selon :

$$P_m(s) = \sum_{i=1}^n (\text{note}(i) * \text{coef}(i))$$

P_m : la valeur de la performance des services (applications) fournis par le volontaire s ;

i : les services (applications) fournis par nœud volontaire s ;

s : le nœud volontaire ;

n : nombre total des services ;

note(i): l'évaluation du service utilisé **i** fourni par le volontaire **s**. La note $\in [0, 9]$;

coef(i): Le coefficient du service **i**

- **Total des coefficients pour la performance**: représente le coefficient de la performance des différents composants soft (applications) du même nœud. Elle est calculée selon :

$$P_c(s) = \sum_{i=1}^n \text{coef}(i)$$

P_c : la valeur de la performance du volontaire **s** relative aux services logiciels;

s : le nœud volontaire ;

n : nombre total des services ;

coef(i) : Le coefficient du service **i**.

- **La valeur de la performance totale** :

$$P_2(s) = P_m(s) / P_c(s) ;$$

4) Calcul la valeur globale

La confiance globale finale est calculée en fonction des valeurs de la confiance et de la performance.

Une fois les valeurs de la performance ou de la confiance sont calculées, ces valeurs vont représenter des résultats positifs, négatifs ou neutres. Un résultat est positif si la valeur de performance ou de confiance est supérieure à (6). Il est neutre si elle est entre (4-6). Il est négatif si elle est inférieure à (4). Après, ces deux valeurs seront agrégées.

$$CG(S) = (P_1(s) + P_2(s) + C(s)) / 3;$$

CG : la confiance globale d'un nœud ;

S : le nœud;

6. Fonctionnement de Cloud Computing Volontaire

6.1. Modélisation avec le diagramme UML

UML (Unified Modeling Language): se définit comme un langage de modélisation graphique et textuel destiné à comprendre et décrire des besoins, spécifier, concevoir des solutions et communiquer des points de vue.

UML unifie à la fois les notations et les concepts orientés objet. Il ne s'agit pas d'une simple notation, mais les concepts transmis par un diagramme ont une sémantique précise et sont porteurs de sens au même titre que les mots d'un langage. UML permet de modéliser de manière claire et précise la structure et le comportement d'un système indépendamment de toute méthode où de tout langage de programmation. [50]

1) Diagramme de séquence

- Diagramme de séquence d'inscription

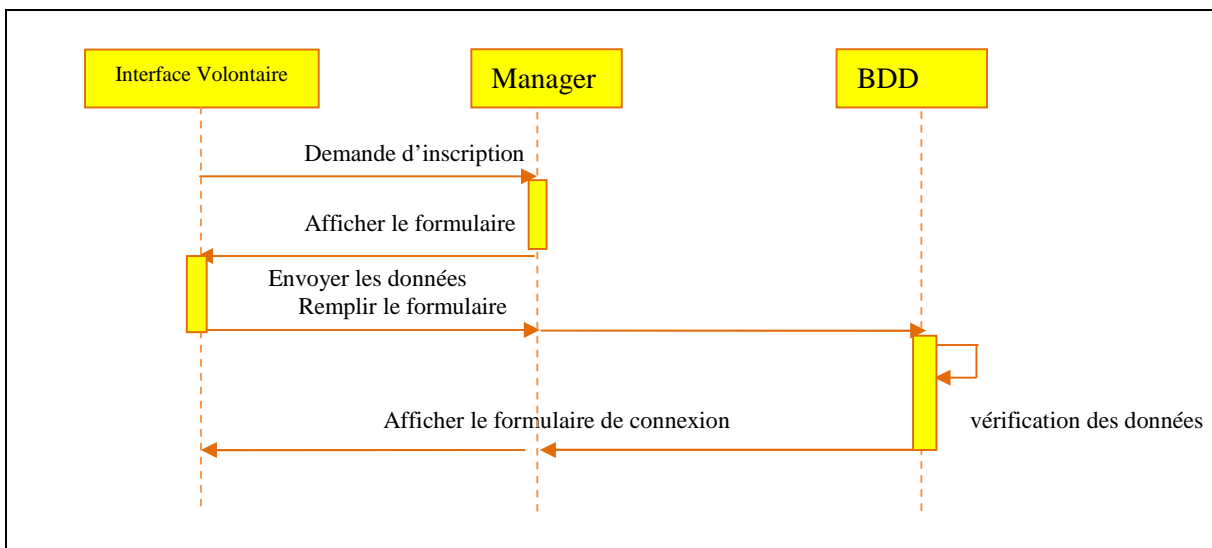


Figure 7 : Diagramme de séquence d'inscription

Le diagramme représenté dans la figure (7) décrit les scénarios possibles lors d'une inscription de volontaire. En effet :

- Le volontaire demande d'inscrire au système.
- Le manager lui affiche un formulaire qui contient des champs vides.
- Le volontaire remplit les champs vides.
- Le manager crée le compte.
- Le manager affiche le formulaire de connexion.

- Diagramme de séquence de connexion

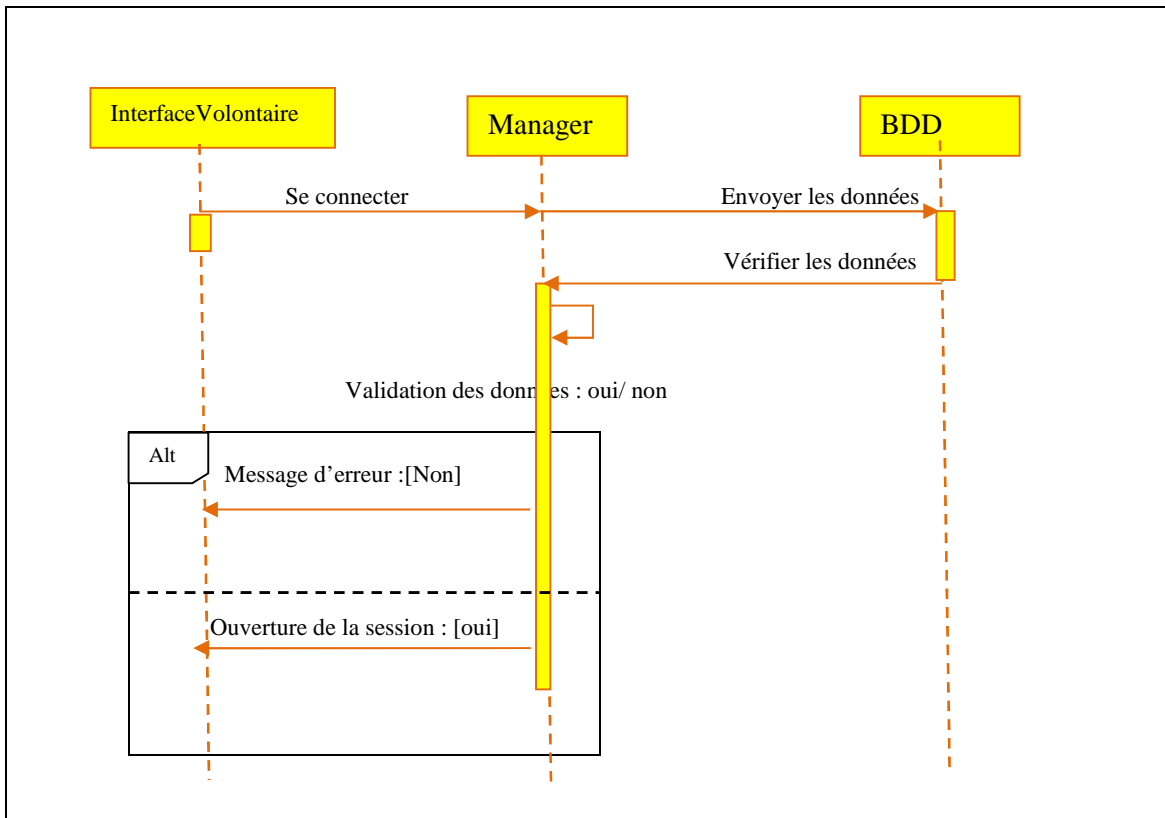
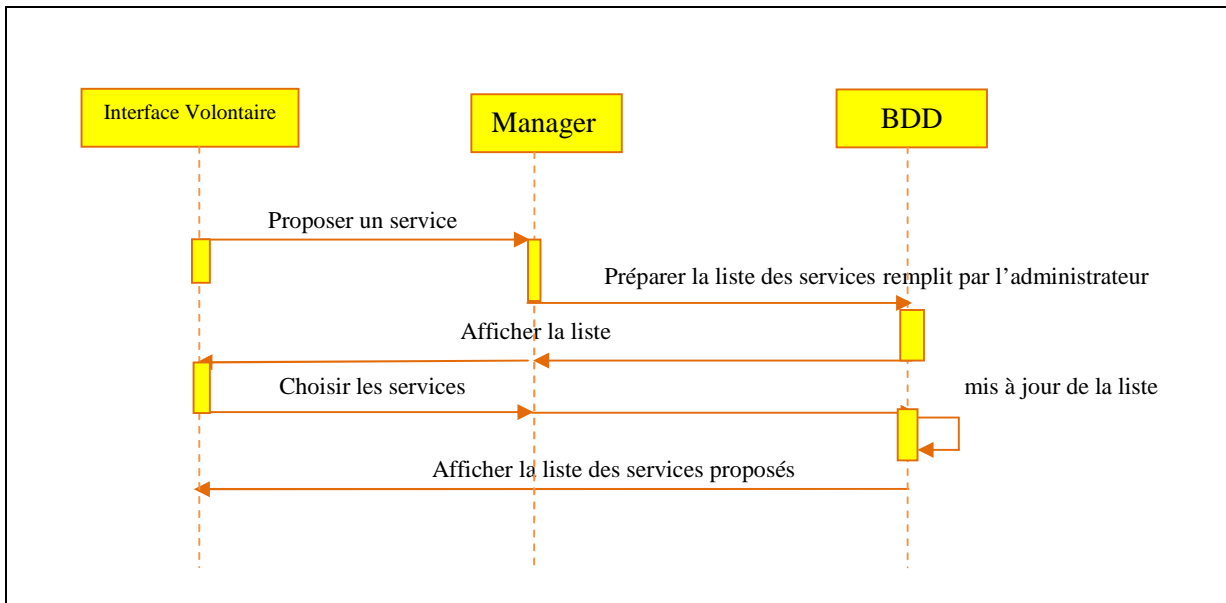


Figure 8 : Diagramme de séquence de connexion

Le diagramme représenté dans la figure (8) décrit les scénarios possibles lors d'une connexion de volontaire. En effet :

- Le volontaire se connecte au système.
 - Le manager vérifie les données de volontaire sur la base de données.
 - Le volontaire remplit les champs vides.
 - Une série de tests doit être réalisée (login existe, tester non de volontaire, tester mot de passe).
- Si tous les champs sont corrects, alors le manager prend en charge les informations introduites et les enregistre dans la base de données et permet à le volontaire d'accéder à la totalité du système.
- Si la connexion n'est pas valide, le volontaire doit soit reconnecter soit quitter le système.

- Diagramme de séquence d'une proposition des services



- **Figure 9 : Diagramme de séquence d'une proposition des services**

Le diagramme représenté dans la figure (9) décrit les scénarios possibles lors d'une proposition des services. En effet :

- Le nouveau volontaire propose des services, et l'ancien volontaire peut proposer d'autres services ou bien demander l'utilisation de la machine virtuelle.
- Le manager prépare la liste des services rempli par l'administrateur.
- Le manager affiche la liste des services demandés.
- Le volontaire choisit les services pour les proposés.
- Le manger affiche la liste des services proposés.

- Diagramme de séquence de demande une machine virtuelle

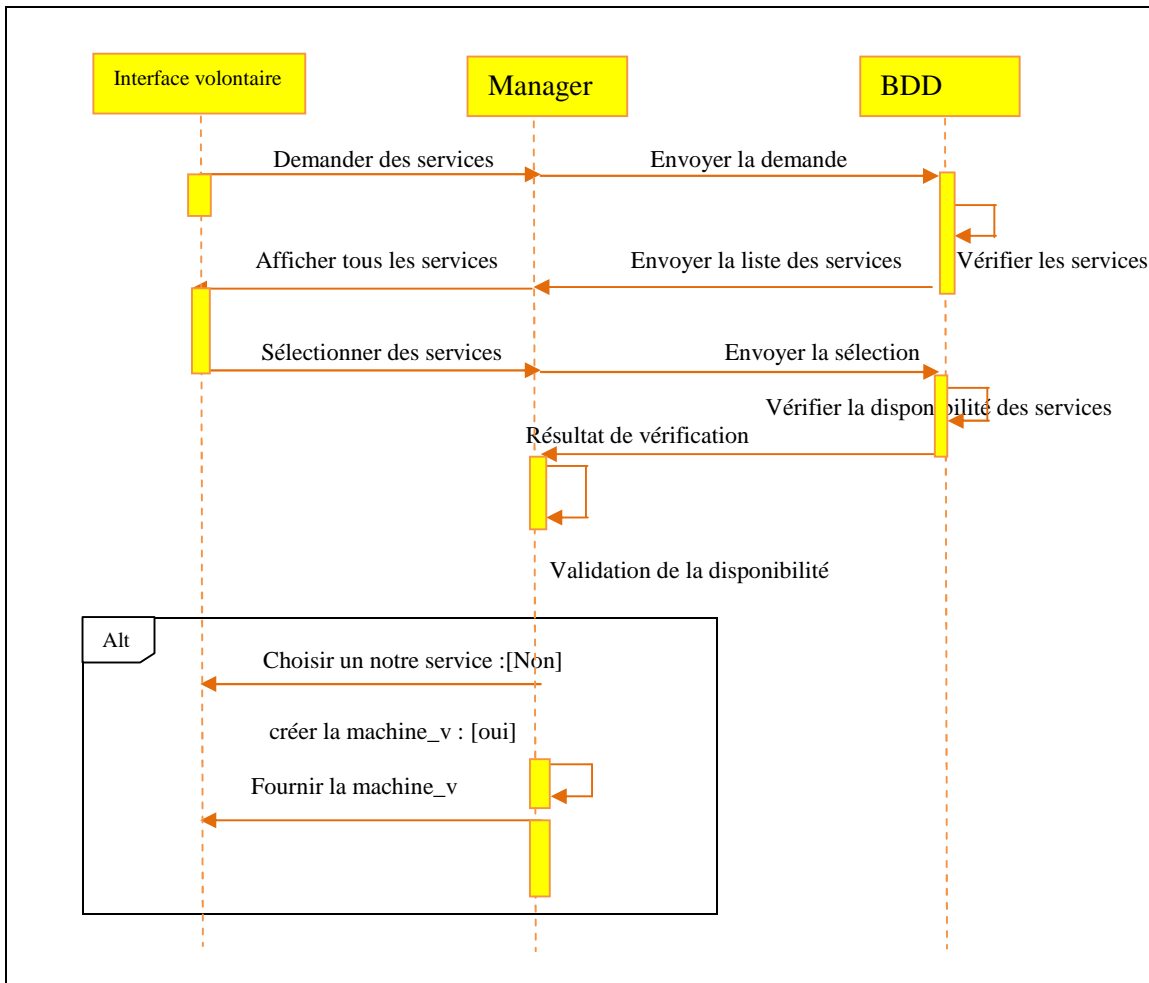


Figure 10 : Diagramme de séquence de demande une machine virtuelle

Le diagramme représenté dans la figure (10) décrit les scénarios possibles lors d'une demande une machine virtuelle. En effet :

- Le volontaire demande une machine virtuelle.
- Le manager vérifie les services et envoyé la liste des services au volontaire.
- Le volontaire sélectionne les services qu'il veut.
- Le manager vérifie la disponibilité des services sélectionnés dans la base de données.
- Si tous les services sélectionnés disponibles, alors le manager crée la machine virtuelle et la fournit au volontaire.
- Si les services ou quelques services non pas disponibles, le volontaire doit choisir d'autres services.

-Diagramme de séquence de l'évaluation

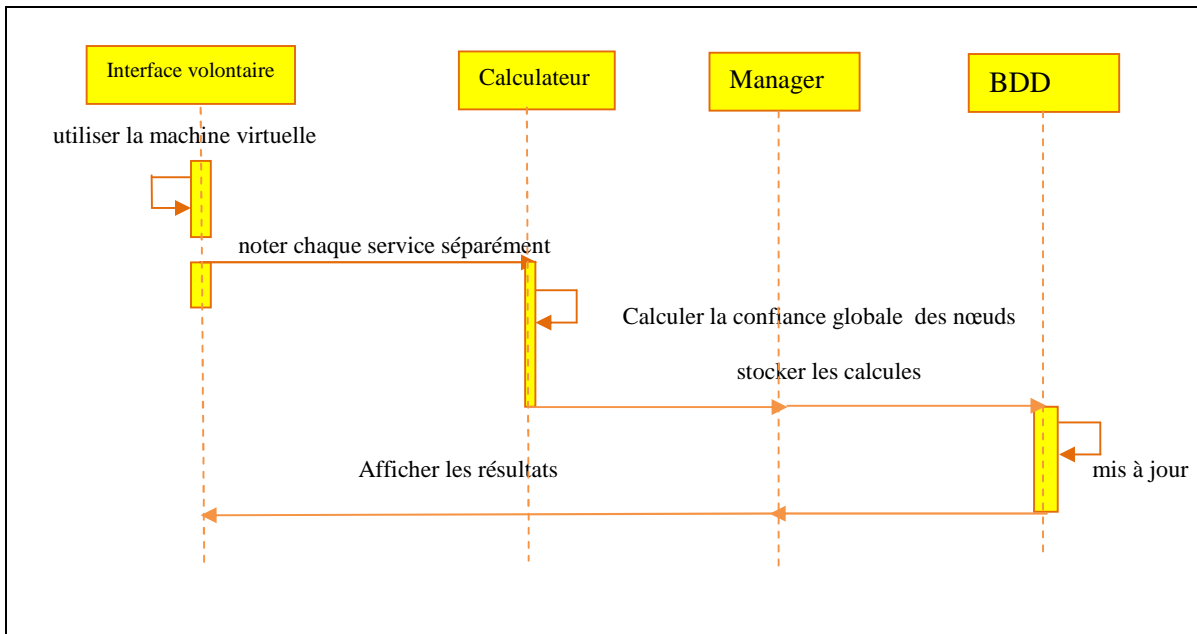


Figure 11 : Diagramme de séquence de l'évaluation

Le diagramme représenté dans la figure (11) décrit les scénarios possibles lors d'une évaluation. En effet :

- Après l'utilisation de la machine virtuelle, le volontaire doit évaluer les services sélectionnés.
- Le calculateur calcule la confiance globale de chaque nœud à partir de les évaluations des services.
- Le manager stocker les résultats dans la base de données et afficher les valeurs de la confiance de chaque nœud.
- Après chaque évaluation la base de données fait la mise à jour.

- Diagramme de séquence d'administrateur

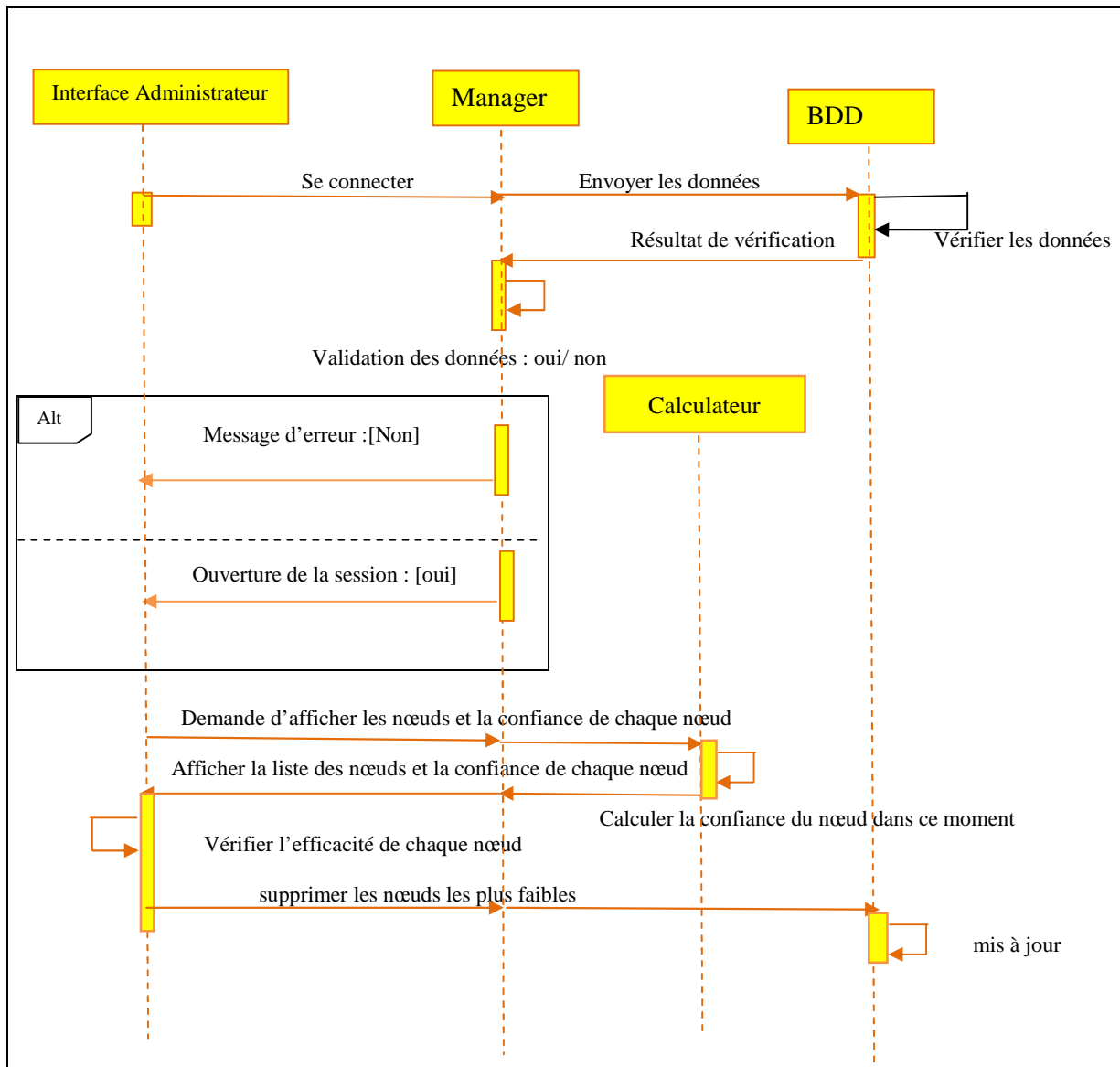


Figure 12 :Diagramme de séquence d'administrateur

Le diagramme représenté dans la figure (12) décrit les scénarios possibles de l'administrateur.

En effet :

- L'administrateur se connecte au système.
- Le manager vérifie les données d'administrateur sur la base de données.
- L'administrateur remplit les champs vides.
- Une série de tests doit être réalisée (login existe, tester non d'administrateur, tester mot de passe). Si tous les champs sont corrects, alors le manager prend en charge les informations introduites et les enregistrent dans la base de données et permet à l'administrateur d'accéder à la totalité du système.
- Si la connexion n'est pas valide, l'administrateur : soit reconnecter soit quitter le système.
- Après la connexion l'administrateur demande au manager d'afficher la liste des nœuds et chaque nœud a une valeur de confiance globale.
- L'administrateur vérifie la liste des nœuds et supprimer les nœuds les plus faibles.
-

2) Diagramme d'activité

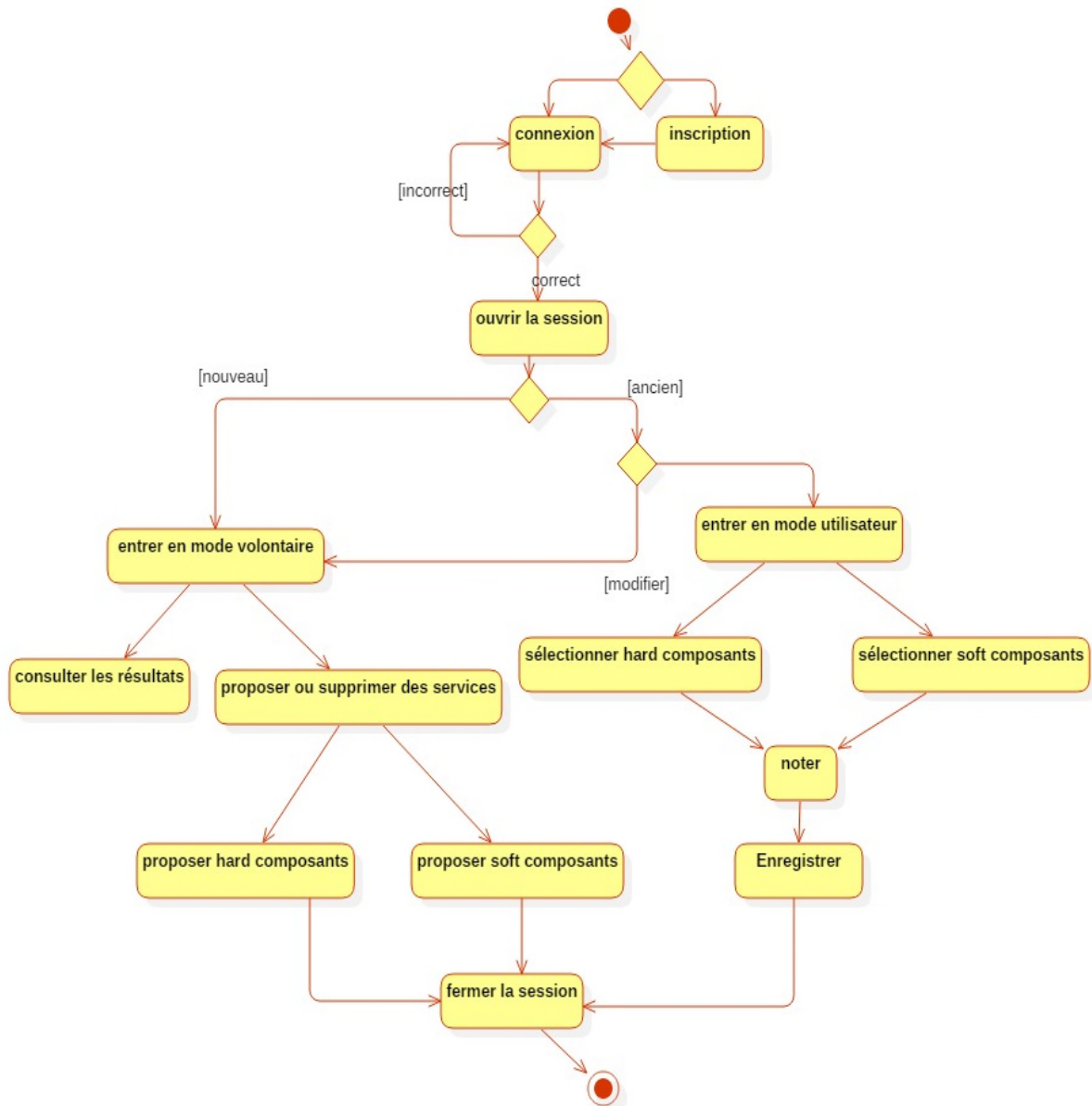


Figure 13 : Diagramme d'activité

Le diagramme d'activité représenté dans la figure (13) décrit les scénarios possibles. En effet :

Un nouveau volontaire inscrire au système, après il se connecte, avant de sélectionner les services pour utiliser la machine virtuelle, il doit proposer des composants, mais l'ancien volontaire se connecte directement au système, il peut proposer ou supprimer des services ou bien utiliser la machine directement, aussi il peut consulter les évaluations des utilisateurs pour ces services. A fin de l'utiliser la machine l'utilisateur de la machine doit noter chaque composant individuellement;

3) Diagramme de classe

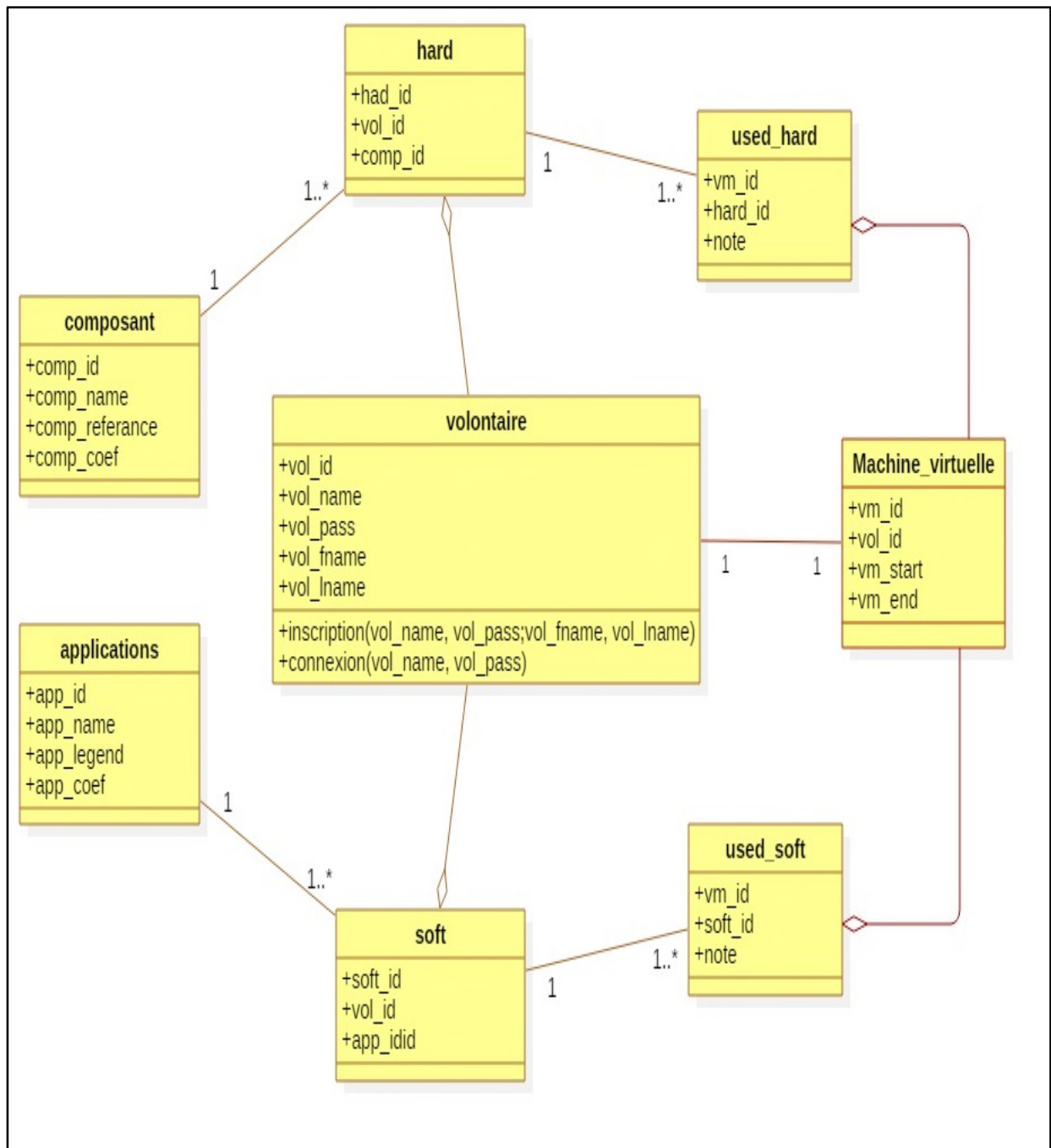


Figure 14 : diagramme de classe

- Ce diagramme figure (14) représente la classe métier de notre système proposé :
- Le volontaire propose un composants hard ou un composants soft ou bien les deux ;
 - Composant Hard contient un ou plusieurs composants (CPU, GPU, RAM, Disque dure) ;
 - Soft composants contient une ou plusieurs applications (des différents applications) ;
 - A partir de ces composants proposés, le manager crée une machine virtuelle sélectionné par un autre volontaire pour l'utiliser ;
 - Le volontaire peut utiliser une seule machine virtuelle, et la machine peut utiliser par un seul volontaire.

6.2. Algorithme de la méthode proposé

Algorithme calculer la confiance des nœuds

Algorithme calcul_confiance_globale

Début

Structure composant {

 Non : **chaîne de caractères** ;

 Note : **entier** ;

 Coef : **entier** ;

}

Structure application {

 Nom : **chaîne de caractère** ;

 Note : **entier** ;

 Coef : **entier** ;

}

Var

T_Perf_N ← 0 ; // Total des notes pour la performance

T_Perf_C ← 0 ; // Total des coefficients pour la performance

T_Conf_N ← 0 ; // Total des notes pour la confiance

T_Conf_C ← 0 ; // Total des coefficients pour la confiance

Performance ← 0

Confiance ← 0 ;

Services_com = **Tableau de composants** ; // chargé depuis le serveur de la base des données

Services_app = **Tableau de applications** ; // chargé depuis le serveur de la base des données

Début

Pour i=1 **Jusqu'à** Longueur(Services_com) **Faire**

Début

Si services[i].nom **dans** ["CPU","RAM","GPU"] **Alors**

Début

 T_Perf_N ← T_Perf_N + (Services_com[i].note * Services_com[i].coef) ;

 T_Perf_C ← T_Perf_C + Services_com[i].coef;

Fin Sinon

Début

 T_Conf_N ← T_Conf_N + (Services_com[i].note * Services_com[i].coef) ;

 T_Conf_C ← T_Conf_C + Services_com[i].coef;

Fin

Fin pour

Pour i=1 **Jusqu'à** Longueur(Services_app) **Faire**

Début

 T_Perf_N ← T_Perf_N + (Services_app[i].note * Services_app[i].coef) ;

 T_Perf_C ← T_Perf_C + Services_app[i].coef;

Fin

Performance ← T_Perf_N / T_Perf_C;

Confiance ← T_Conf_N / T_Conf_C;

Confiance_globale ← (Performance + Confiance) / 2 ;

Fin.

Figure 15 : Algorithme calculer la confiance globale des nœuds

Conclusion

Dans ce chapitre avons présenté le problème de réputation et de confiance dans le Cloud Computing Volontaire. Pour cela, nous avons proposé une méthode de gestion de confiance dans un environnement de Cloud Computing volontaire basé sur le modèle d'opinion.

Cette méthode consiste à classer les services selon leurs fiabilités et de sélectionner les nœuds les plus efficaces en se basant sur deux calculs. Le premier calcul est celui de la valeur de la performance basé sur l'évaluation de la QoS , et le deuxième calcul est celui de la valeur de la confiance basé sur la sécurité des données. Le chapitre suivant présentera l'implémentation de la méthode proposée.

Introduction

Après avoir détaillé les composants de notre système de confiance pour la classification des services et la sélection des nœuds dans un environnement Cloud volontaire et de son architecture et avoir proposé un modèle computationnel pour le calcul de confiance, nous décrivons, au sein de ce chapitre son implémentation qui a été réalisée sous l'environnement c#. En outre, ce chapitre détaille les classes et les interfaces de notre approche proposé.

1. Langage et environnement de développement

1.1. Visual studio

Microsoft Visual Studio est un environnement de développement intégré de Microsoft. Il est utilisé pour développer des programmes informatiques, ainsi que des sites Web, des applications Web, des services Web et des applications mobiles. Visual Studio utilise des plates-formes de développement logiciel Microsoft telles que Windows API, Windows Forms, Windows Présentation Fondation, Windows Store et Microsoft Silverlight.

Visual Studio prend en charge 36 langages de programmation différents et permet à l'éditeur de code et au débogueur de prendre en charge (à des degrés divers) presque n'importe quel langage de programmation, à condition qu'un service spécifique au langage existe. [40]

1.2. Langage de programmation C

C# est un langage de programmation orientée objet, fortement typé, dérivé de C et de C++, ressemblant au langage Java. Il est utilisé pour développer des applications web, ainsi que des applications de bureau, des services web, des commandes, des widgets ou des bibliothèques de classes. En C#, une application est un lot de classes où une des classes comporte une méthode Main, comme cela se fait en Java.

C# est destiné à développer sur la plateforme .NET, une pile technologique créée par Microsoft pour succéder à COM.

Les exécutables en C# sont subdivisés en assemblées, en namespaces, en classes et en membres de classe. Un assemblé est la forme compilée, qui peut être un programme (un exécutable) ou une bibliothèque de classes (une dll). Un assemblé contient le code exécutable en MSIL, ainsi que les symboles. Le code MSIL est traduit en langage machine au moment de l'exécution par la fonction just-in-time de la plateforme.NET. [41]

1.3. Les caractéristiques de C

Comme la définition de C # a évolué, les buts utilisés dans sa conception étaient les suivants:

- C # est conçu pour être un langage de programmation simple, moderne, polyvalent et orienté objet.
- Le langage et ses implémentations devraient fournir un support pour l'ingénierie logicielle principes tels que la vérification de type forte, la vérification des limites des tableaux, la détection des tentatives d'utilisation variables non initialisées et collecte automatique de place.
- Le langage est destiné à être utilisé dans le développement de composants logiciels pouvant être déployés dans environnements distribués.
- La portabilité du code source est très importante, tout comme la portabilité du programmeur, en particulier pour ceux programmeurs déjà familiers avec C et C ++.

- Le soutien à l'internationalisation est très important.
- C # est conçu pour être adapté à l'écriture d'applications pour les systèmes hébergés et embarqués, de la très grande qui utilise des systèmes d'exploitation sophistiqués, jusqu'au très petit. [42]

1.4.MySQL

MySQL est une base de données relationnelle libre qui a vu le jour en 1995 et très employée sur le Web, souvent en association avec PHP (langage) et Apache (serveur web). MySQL fonctionne indifféremment sur tous les systèmes d'exploitation (Windows, Linux, Mac OS notamment).

Le principe d'une base de données relationnelle est d'enregistrer les informations dans des tables, qui représentent des regroupements de données par sujets (table des clients, table des fournisseurs, table des produits, par exemple). Les tables sont reliées entre elles par des relations.

Le langage SQL (acronyme de Structured Query Language) est un langage universellement reconnu par MySQL et les autres bases de données et permettant d'interroger et de modifier le contenu d'une base de données. Les autres bases de données utilisées en informatique sont essentiellement Microsoft SQL Server et Oracle. [43]

2. Les différentes classes de notre projet

2.1.La classe interface volontaire

On a créé la classe interface volontaire pour faire plusieurs méthodes telle :

- 1) Cette figure (16) permet de demander de la création d'un nouveau compte pour faire la connexion.

```
private void linkLabel1_Click(object sender, EventArgs e)
{
    create_account cc = new create_account();
    if (cc.ShowDialog() == DialogResult.OK)
    {
```

Figure 16: Méthode pour demander de la création d'un nouveau compte

- 2) Cette figure (17) permet de Remplir les champs données par le manager avec les informations de chaque nouveau volontaire.

```
private void button1_Click(object sender, EventArgs e)
{
    if (f1.Text != "" && f2.Text != "" && f3.Text != "" && f4.Text != "")
    {
        if (cloud.create_account(f1.Text, f2.Text, f3.Text, f4.Text))
        {
            DialogResult = DialogResult.OK;
        }
    }
}
```

Figure 17 : Méthode pour Remplir les champs donnée par le manager

3) Cette figure(18) permet de demander un formulaire par un volontaire soit nouveau ou ancien, pour connecter au système, cette opération se fait à partir le remplissage des champs données.

```
private void button1_Click(object sender, EventArgs e)
{
    cloudv.USER = cloud.login(textBox1.Text, textBox2.Text);

    if (cloudv.USER != null)
    {
        DialogResult = DialogResult.OK;
    }
    else
    {
        MessageBox.Show("Erreur de login!", "Ereur", MessageBoxButtons.OK, MessageBoxIcon.Error);
    }
}
```

Figure 18 : Méthode pour demande un formulaire pour connecter au système

4) Cette figure(19) permet de proposer un Matériel (hard composant) et un Logiciel (application) par un Volontaire à partir la liste remplit par l'administrateur.

```
private void Propositions_Load(object sender, EventArgs e)
{
    cloudv.USER.load_hard_composants();
    cloudv.USER.load_soft_composants();
    laod_hard();
    laod_soft();
    cancule();
}
```

Figure 19 : Méthode pour la proposition des services

5) Cette figure (20) permet de modifier les services proposés par un volontaire (ajouter ou bien supprimer un service).

```
private void button4_Click(object sender, EventArgs e)
{
    if (listView2.SelectedItems.Count == 1)
    {
        if (MessageBox.Show("Supprimer?", "Confirmer", MessageBoxButtons.YesNo, MessageBoxIcon.Question) == DialogResult.Yes)
        {
            cloudv.USER.remove_soft_composant(listView2.SelectedItems[0].Text);
            cloudv.USER.load_soft_composants();
            laod_soft();
            cancule();
        }
    }
}
```

Figure 20 : Méthode pour la modification des services proposés

6) Cette figure (21) permet la sélection des services (Matériel et Logiciel), sélectionner des services veut dire l'utilisation de ses services.

```
private void button1_Click(object sender, EventArgs e)
{
    string sql = "SELECT * FROM composants, hard WHERE (hard.com_id = composants.com_id) AND (composants.com_name = ?)";
    MySqlCommand cmd = new MySqlCommand(sql, cloudv.link);
    cmd.Parameters.Add("CA", MySqlDbType.String).Value = comboBox1.Text;
    MySqlDataReader reader = cmd.ExecuteReader();
    listView1.Items.Clear();
    while (reader.Read())
    {
        proposed_hard ph = new proposed_hard();
        ph.id = reader.GetInt32("har_id");
        ph.uid = cloudv.USER.user_id;
        ph.hard = new hard_composant();
        ph.hard.id = reader.GetInt32("com_id");
        ph.hard.name = reader.GetString("com_name");
        ph.hard.reference = reader.GetString("com_reference");
        ph.hard.coef = reader.GetInt32("com_coef");
    }
}
```

Figure 21 : Méthode pour la sélection des services

7) Cette figure (22) permet d'évaluer les services utilisés, l'évaluation entre [0,9].

```
private void set_note_Load(object sender, EventArgs e)
{
    comboBox1.SelectedIndex = 6;
}

private void button1_Click(object sender, EventArgs e)
{
    DialogResult = DialogResult.OK;
}
```

Figure 22: Méthode pour l'évaluation des services

2.2. La classe du calculateur

La classe calculateur figure (23) c'est le cœur de notre approche proposé, son rôle est de calculer la performance à partir l'évaluation des volontaires sur (CPU,GPU,RAM), et calculer la confiance à partir l'évaluations de (HDD), ensuite il calcul la confiance globale avec la combinaison du performance et du confiance.

```

public void calcule()
{
    decimal total_performance_note_avg = 0;
    decimal total_performance_coefficient = 0;
    decimal total_confiance_note_avg = 0;
    decimal total_confiance_coefficient = 0;

    foreach (cloud_service ser in com_services)
    {
        if (ser.name == "HDD" )
        {
            total_confiance_note_avg += ser.note_avg * ser.coef;
            total_confiance_coefficient += ser.coef;
        }
        else
        {
            total_performance_note_avg += ser.note_avg * ser.coef;
            total_performance_coefficient += ser.coef;
        }
    }
    foreach (cloud_service ser in app_services)
    {
        total_performance_note_avg += ser.note_avg * ser.coef;
        total_performance_coefficient += ser.coef;
    }
    decimal performance_avg = 0;
    decimal confiance_avg = 0;
    decimal indice_global_avg = 0;
    if (total_performance_coefficient > 0 )
    {
        performance_avg = (decimal)total_performance_note_avg / (decimal)total_performance_coefficient;
        confiance_avg = (decimal)total_confiance_note_avg / (decimal)total_confiance_coefficient;
        indice_global_avg = (performance_avg + confiance_avg) / 2;
    }
    label9.Text = string.Format("{0:0.000}", performance_avg);
    label8.Text = string.Format("{0:0.000}", confiance_avg);
    label7.Text = string.Format("{0:0.000}", indice_global_avg);
}

```

Figure 23 : Les méthodes de la classe calculateur

2.3. La classe du manager

Le rôle de la classe manager est :

1) La validation des informations d'inscription du volontaire et le stockage de ces informations dans la base de données.

```

private void create_account_Load(object sender, EventArgs e)
{
    ,

```

Figure 24 : Méthode pour la validation des informations

2) La vérification des informations de la connexion s'elles sont correctes, il confirme la connexion sinon il envoie un message d'erreur pour reconnecter.

```

private void Login_Load(object sender, EventArgs e)
{
    ,

```

Figure 25 : Méthode pour La vérification des informations de la connexion

3) La fourniture de la machine virtuelle sélectionnée par un volontaire.

```
private void pick_hard_Load(object sender, EventArgs e)
{
    comboBox1.Items.Clear();
    MySqlCommand cmd = new MySqlCommand("SELECT com_name FROM composants", cloudv.link);
    MySqlDataReader reader = cmd.ExecuteReader();
    while (reader.Read()) { comboBox1.Items.Add(reader.GetString("com_name")); }
    reader.Close();
    reader.Dispose();
    cmd.Dispose();

    dateTimePicker1.Value = DateTime.Parse("01/01/2018");
    dateTimePicker2.Value = DateTime.Parse("31/12/2018");
    comboBox1.SelectedIndex = 0;
}
```

Figure 26 : Méthode pour la fourniture de la machine virtuelle

4) L'affichage de la liste des nœuds et la valeur de la confiance globale de certain moment, chaque nœud a sa confiance globale.

```
private void Statistiques_Load(object sender, EventArgs e)
{
```

Figure 27 : Méthode d'afficher les nœuds

2.4.La classe d'interface administrateur

Le but de la création de cette classe est :

1) Faire la connexion pour accéder au système à partir de remplir un formulaire donnée, si les informations sont correctes il accède sinon il doit reconnecter.

```
private void button1_Click(object sender, EventArgs e)
{
    Form1.ADMIN = cloud_admin.login(textBox1.Text, textBox2.Text);

    if (Form1.ADMIN != null)
    {
        DialogResult = DialogResult.OK;
    }
    else
    {
        MessageBox.Show("Erreur de login!", "Ereur", MessageBoxButtons.OK, MessageBoxIcon.Error);
    }
}
```

Figure28 : Méthode pour la connexion au système

2) Le remplissage d'une liste des services (Matériels et Logiciels), cette liste sert les volontaires à proposer.

```
private void hard_Load(object sender, EventArgs e)
{
    if (hard != null)
    {
        in_name.SelectedIndex = in_name.Items.IndexOf(hard.name);
        in_ref.Text = hard.reference;
        in_coef.SelectedIndex = hard.coef - 1;
    }
}
```

Figure29 : Méthode de remplir la liste des hard composants

3) Faire une demande d'afficher les nœuds et la confiance globale de chaque nœud de ce moment.

```

public void cancel(Int32 id, Int32 lv)
{
    decimal total_performance_note_avg = 0;
    decimal total_performance_coefficient = 0;
    decimal total_confiance_note_avg = 0;
    decimal total_confiance_coefficient = 0;

    load_hard();
    foreach (cloud_service ser in com_services)
    {
        if (ser.name == "HDD")
        {
            total_confiance_note_avg += ser.note_avg * ser.coef;
            total_confiance_coefficient += ser.coef;
        }
        else
        {
            total_performance_note_avg += ser.note_avg * ser.coef;
            total_performance_coefficient += ser.coef;
        }
    }

    load_soft();
    foreach (cloud_service ser in app_services)
    {
        total_performance_note_avg += ser.note_avg * ser.coef;
        total_performance_coefficient += ser.coef;
    }

    decimal performance_avg = 0; if (total_performance_coefficient > 0) { performance_avg = (dec:
    decimal confiance_avg = 0; if (total_confiance_coefficient > 0) { confiance_avg = (decimal)t

    listView1.Items[lv].SubItems[3].Text = string.Format("{0:0.000}", performance_avg);
    listView1.Items[lv].SubItems[4].Text = string.Format("{0:0.000}", confiance_avg);
}
    
```

Figure30 : Méthode d'afficher les nœuds

4) Sélectionner les nœuds les plus efficaces et supprimer les plus faibles.

```

private void button2_Click(object sender, EventArgs e)
{
    if (listView1.SelectedItems.Count != 1) { return; }

    if (MessageBox.Show("Supprimer?", "Confirmation.", MessageBoxButtons.YesNo, MessageBoxIcon.Question) == System.Windows.Forms.DialogResult.Yes)
    {
        Int32 id = (Int32) listView1.SelectedItems[0].Tag;
        MySqlCommand cmd = new MySqlCommand("", Form1.link);
        MySqlDataReader reader;

        List<Int32> hard = new List<int>();
        cmd.CommandText = "select * from hard where user_id=" + id;
        reader = cmd.ExecuteReader();
        while (reader.Read()) { hard.Add(reader.GetInt32(0)); }
        reader.Close();

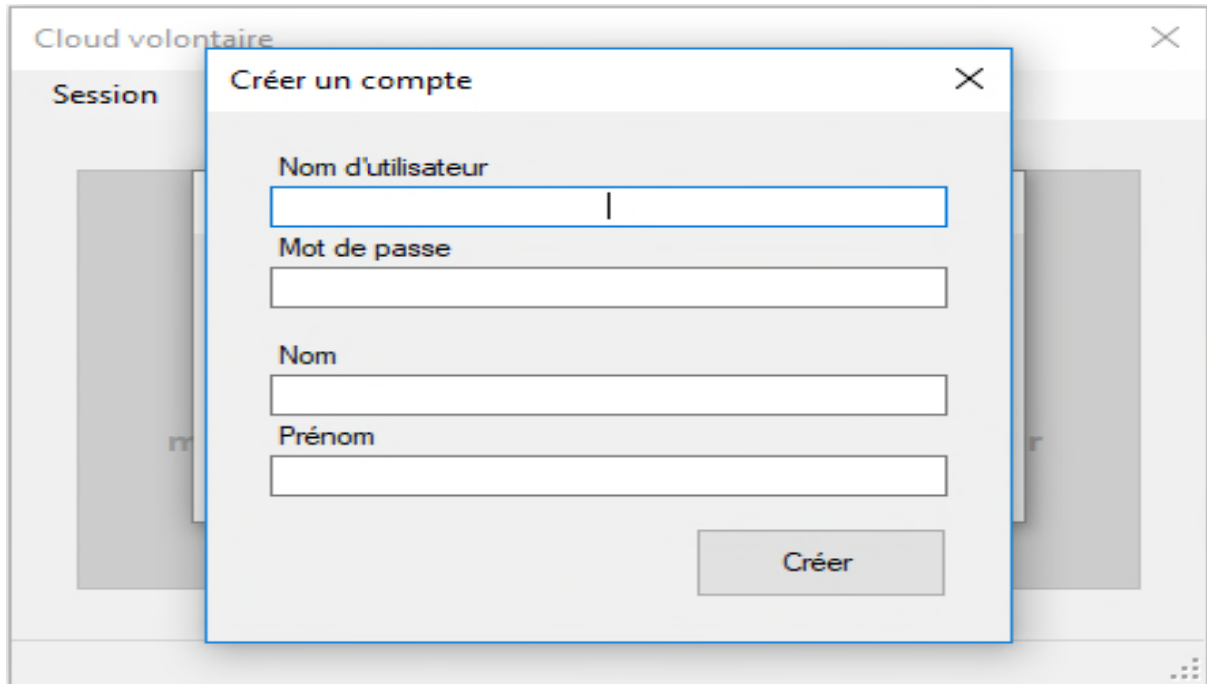
        List<Int32> soft = new List<int>();
        cmd.CommandText = "select * from soft where user_id=" + id;
        reader = cmd.ExecuteReader();
        while (reader.Read()) { soft.Add(reader.GetInt32(0)); }
        reader.Close();
        reader.Dispose();
    }
}
    
```

Figure31 : Méthode pour sélectionner et supprimer les nœuds

3. Présentation de différentes interfaces de volontaire

3.1.L'interface d'inscription

- Cette interface figure (32) permet à un nouveau volontaire de s'inscrire, et ouvrir une nouvelle session. Ainsi, l'interface permet à l'utilisateur (Volontaire) de saisir son nom d'utilisation et son mot de passe puis son nom et prénom. Une fois que le volontaire est inscrit, il doit se connecter encore au système.



The image shows a screenshot of a software interface for creating an account. The main window is titled 'Cloud volontaire' and has a 'Session' section. A modal dialog box titled 'Créer un compte' is open in the foreground. The dialog box contains the following fields and buttons:

- Input field for 'Nom d'utilisateur' (Username)
- Input field for 'Mot de passe' (Password)
- Input field for 'Nom' (Name)
- Input field for 'Prénom' (First name)
- A 'Créer' (Create) button at the bottom right.

Figure 32 : l'interface d'inscription

3.2.L'interface de connexion

Cette interface figure (33) permet au volontaire (ancien et nouveau) de se connecter au système par la saisit de son nom d'utilisateur et son mot de passe, puis il clique sur l'onglet connexion.

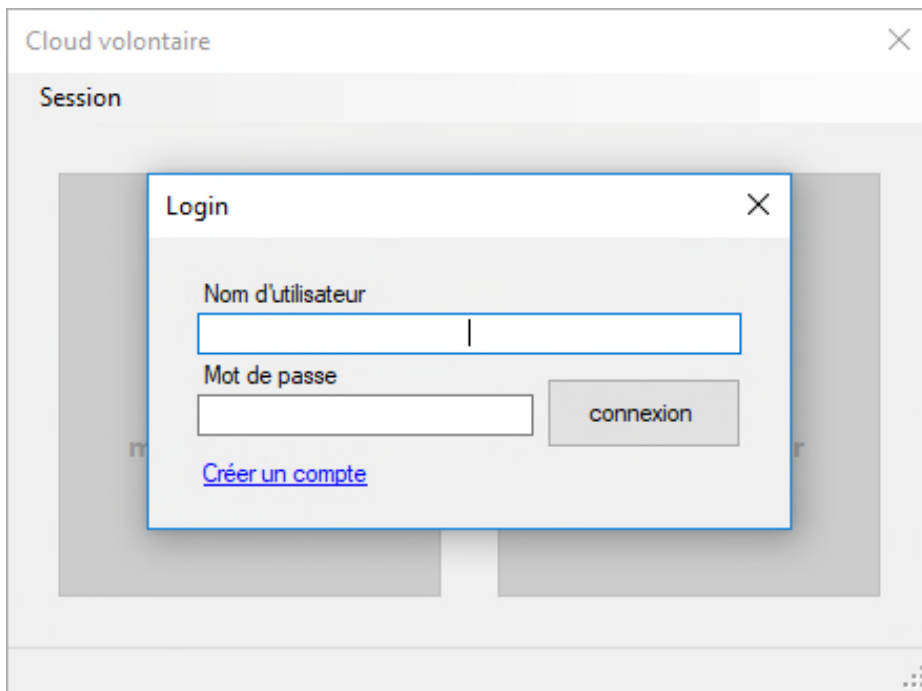


Figure 33 : l'interface de connexion

3.3.L'interface de choisir le mode

Nous avons prévu deux modes d'accès aux volontaires qui sont : Nouveau volontaire et Ancien volontaire.

1) Nouveau volontaire

Cette interface Figure (34) permet à un nouveau volontaire de choisir, le mode volontaire afin qu'il puisse fournir gracieusement ses services (matériels et logiciels). Une fois qu'il a proposé ses services, il peut choisir le mode utilisateur pour pouvoir sélectionner les services qu'il veut utiliser sous forme de machine virtuelle.

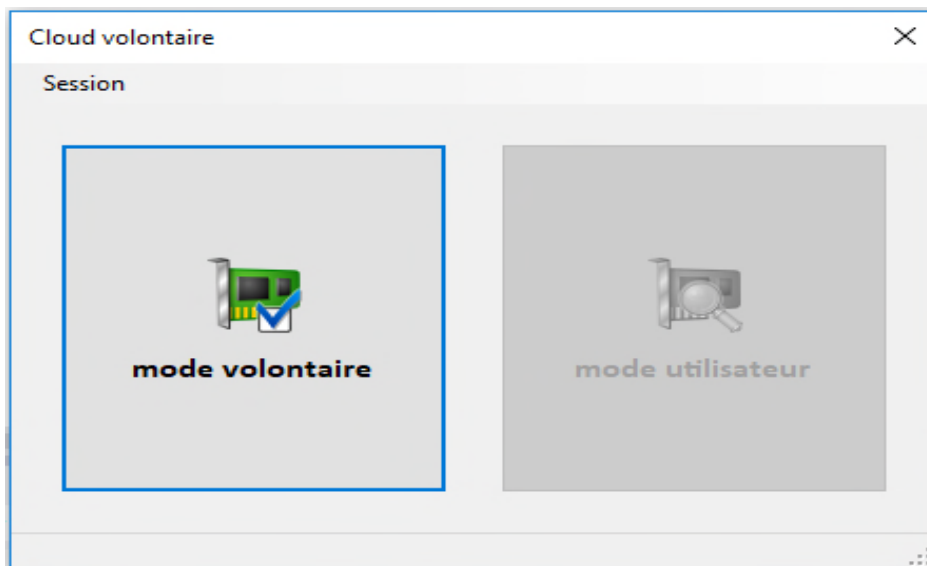


Figure 34 : l'interface pour choisir le mode utilisateur

2) Ancien volontaire

Cette interface Figure (35) permet à l'ancien volontaire de choisir le mode volontaire pour pouvoir proposer d'autres services ou bien supprimer quelques uns. Il a aussi la possibilité de choisir le mode utilisateurs pour la sélection des services à utilisés.

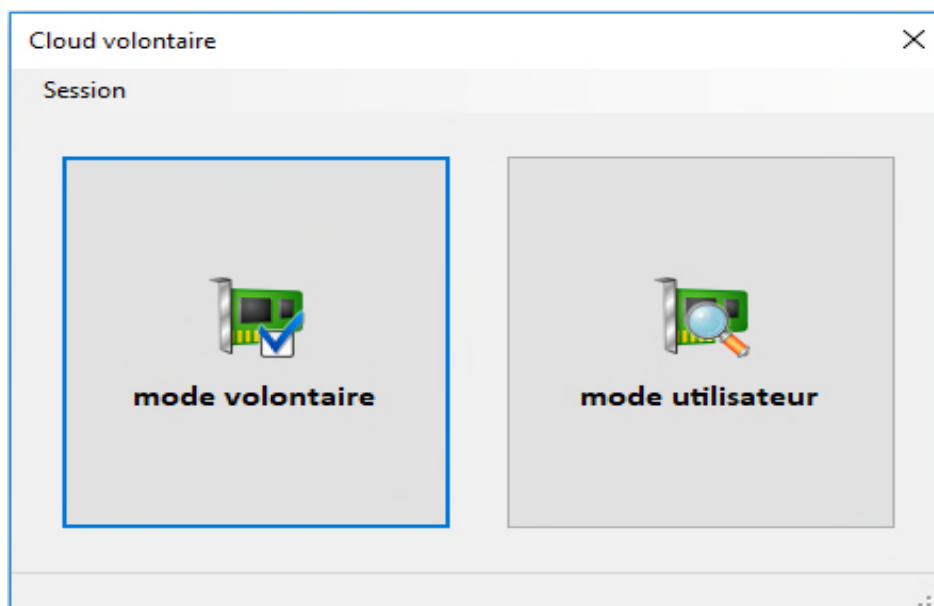


Figure 35 : l'interface pour choisir le mode volontaire

3.4. Interface proposition des services

Cette interface Figure (36) permet au volontaire de proposer ses services à partir d'une liste de services fournis par l'administrateur de système. Ainsi, cette interface permet au volontaire de proposer des services ou supprimer d'autres. Elle lui permet aussi de consulter ses résultats (performance, confiance, confiance globale)

Pour proposer un hard ou un soft composant :

Pour supprimer un hard ou soft composant :

proposer un hard composant

Supprimer un hard composant

code	Composant	Référence	Coefficient	MIN	AVG	MAX
1	CPU	I3	2	9	9,0000	9
3	RAM	8GB	5	3	3,0000	3
4	HDD	SSD 512	9	7	7,0000	7

proposer un soft composant

Supprimer un soft composant

code	Application	Légende	Coefficient	MIN	AVG	MAX
1	app 1	legend 01	5	jamais utilisé	jamais utilisé	jamais utilisé
2	app 2	legend 02	2	6	6,0000	6

Statistiques

Indice de performance minimal	3,214	Indice de performance moyen	3,214	Indice de performance maximal	3,214
Indice de confiance minimal	7,000	Indice de confiance moyen	7,000	Indice de confiance maximal	7,000
Indice global minimal	5,107	Indice global moyen	5,107	Indice global maximal	5,107

Consulter les résultats (performance, confiance et confiance globale)

Figure 36: Interface proposition des services

Une fois que le volontaire clique sur le bouton proposer un composant matériel ou logiciel, une liste des composants sera affichée figure (37). Ainsi, cette interface permet aux volontaires de proposer des services à partir d'une liste fournis par l'administrateur. Le volontaire doit choisir un composant (matériel ou logiciel) qu'il veut proposer à partir de cette liste.

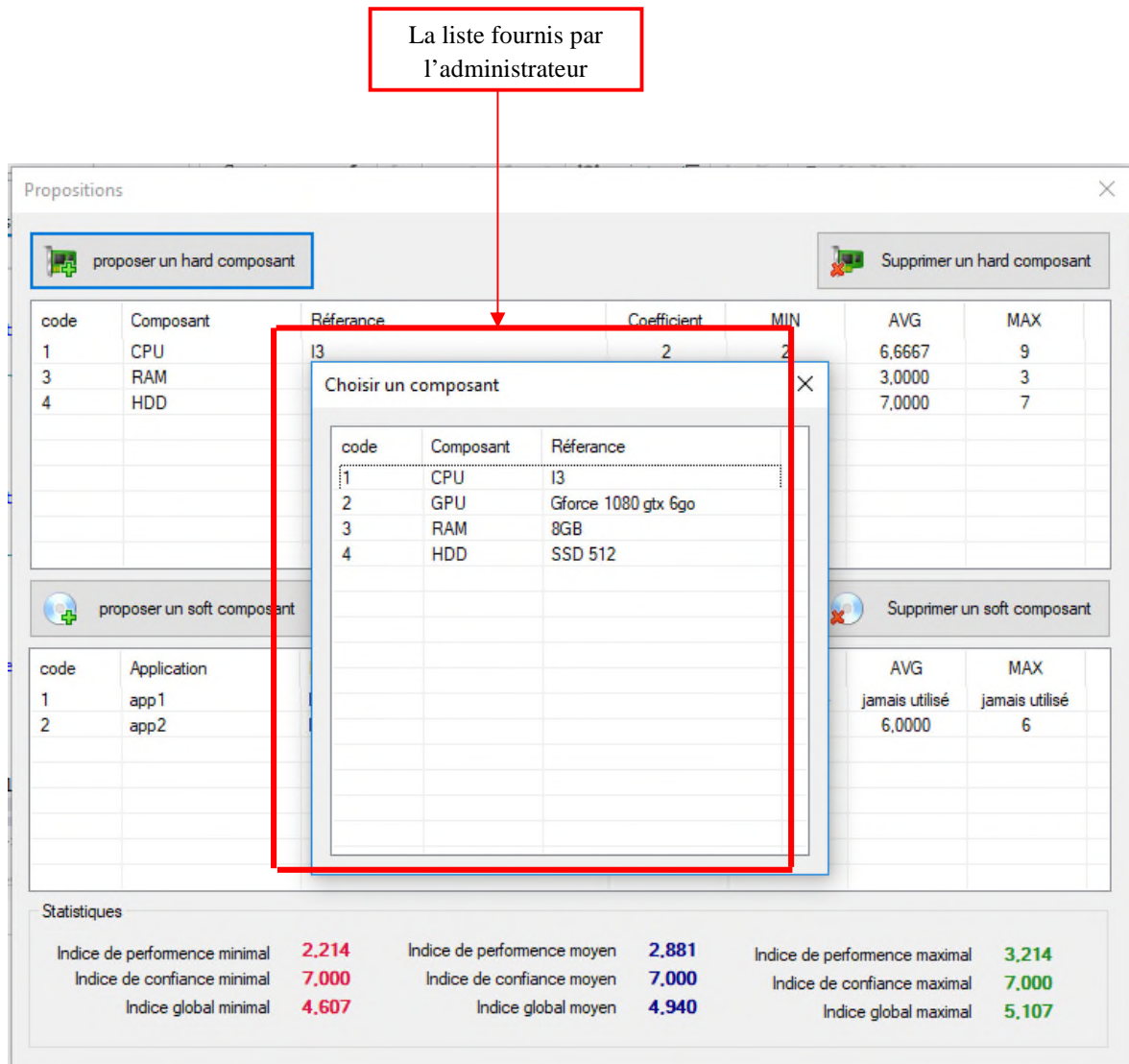


Figure 37 : Liste des composants fournis par l'administrateur

3.6 Interface sélection des services

Cette interface figure (38) permet au volontaire de :

- sélectionner les services (matériel ou logiciel) afin de les utiliser ou bien les supprimer de sa liste de sélection. Dans le cas d'une sélection, une autre interface apparaît où le volontaire peut rechercher et sélectionner les services voulus.
- noter les services utilisés après l'utilisation de ces derniers.

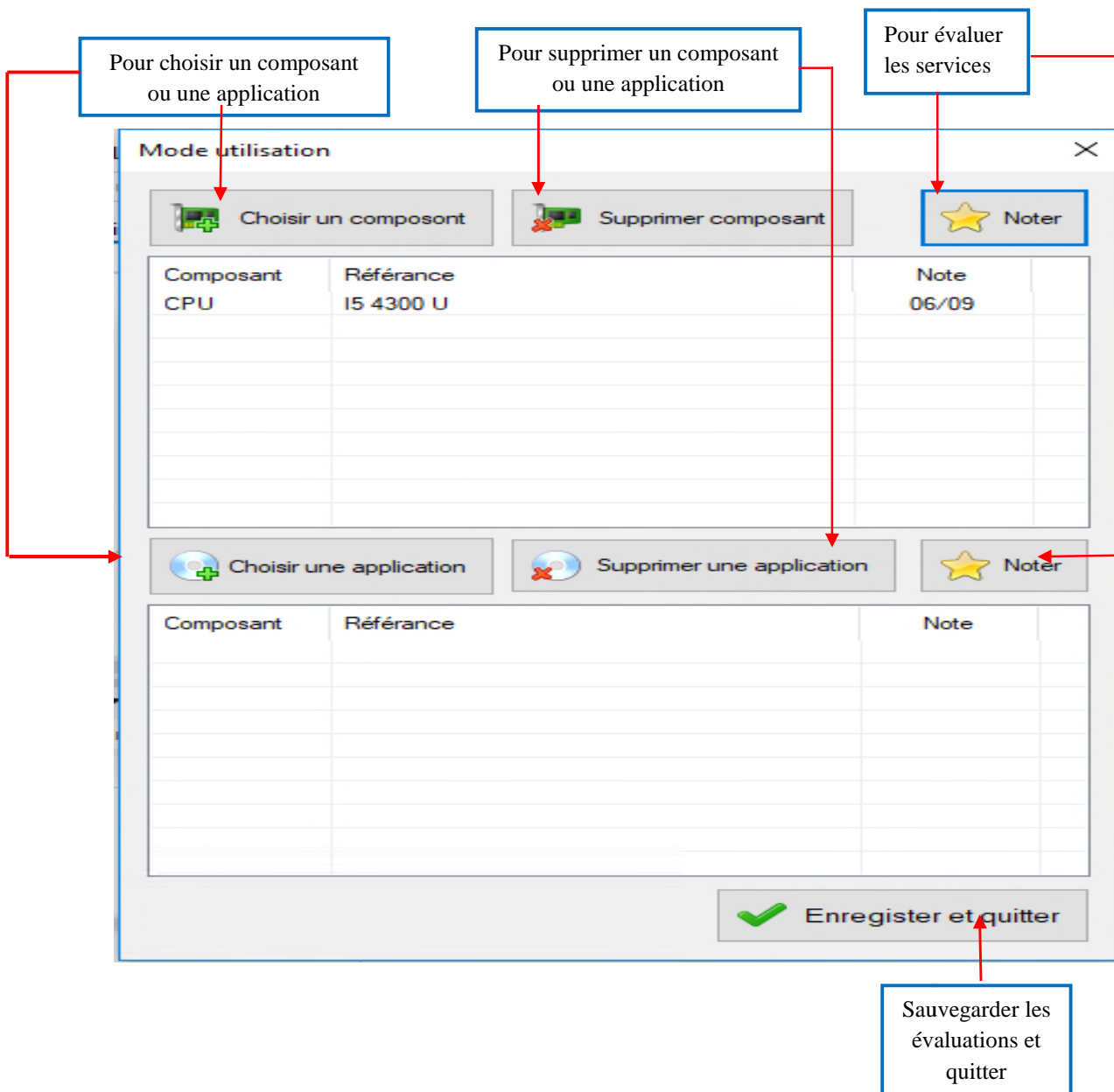


Figure 38 : Interface sélection des services

3.5. Interface évaluation des services utilisés

Cette interface figure (39) permet d'évaluer les services utilisés. Ainsi, à travers cette interface l'utilisateur évalue le service qu'il a utilisé. La note de l'évaluation est entre [0-9]. Une fois l'évaluation est terminée, le volontaire peut enregistrer et quitter.

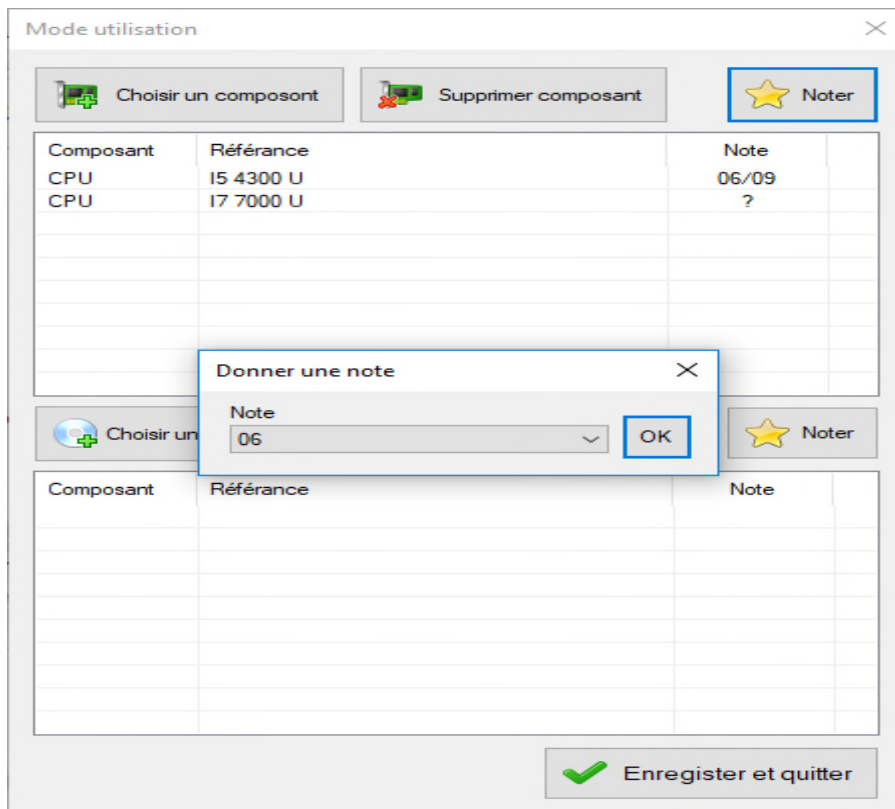


Figure 39 : Interface de notation des services utilisés

4. Présentation de différentes interfaces d'administrateur

4.1.L'interface de connexion

Cette interface Figure (40) permet à l'administrateur de se connecter au système.

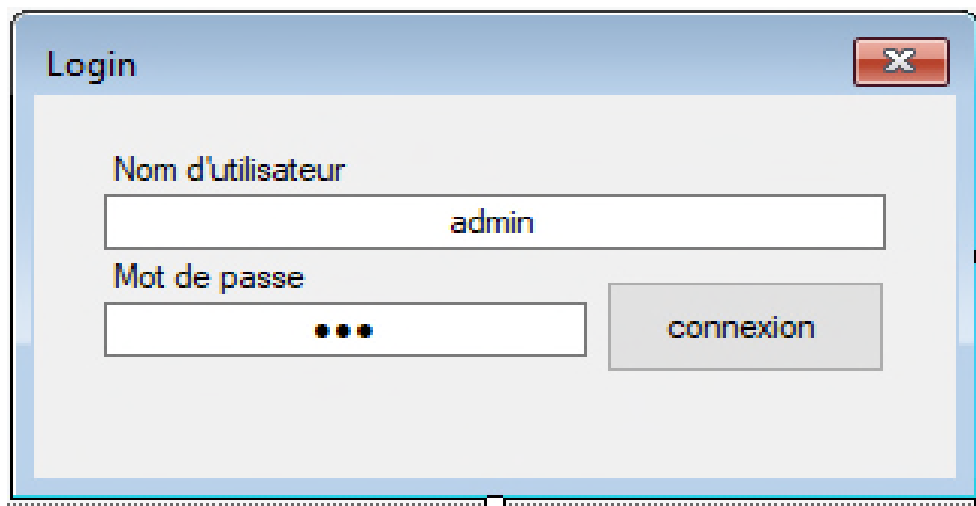


Figure 40: Interface de connexion

4.2. L'interface pour estimer le coefficient d'un service

Cette interface Figure (41) permet à l'administrateur de saisir les noms des services (Matériel ou Logiciel) dont les volontaires auront besoin, puis il estime et saisie le coefficient de chaque type de service. Dans le cas où le service est un composant matériel, l'administrateur

REFERENCES BIBLIOGRAPHIQUES

- [1] Akbikhalil, Zehrime hammed ,Mémoire MASTER ACADEMIQUE, Etude et mise en place d'une solution cloud computing privé, (2012-2013)
- [2] NIST- www.nist.gov/publications/nist-definition-cloud-computing (2011)
- [3] CISCO- www.cisco.com/c/dam/m/fr_fr/internet-of-everything/iac/assets/pdfs/cloud/C11-733578-00-Intercloud-WP-v5a-FR.pdf
- [4] Fatima Zohra filali, thèse doctoral, gestion de confiance dans le Cloud Computing,(2015-2016)
- [5] Wikipedia. (2012e, Janvier 2). Virtualisation. Consulté le Janvier 5, 2012, sur Wikipedia: <http://fr.wikipedia.org/wiki/Virtualisation>
- [6] Armbrust, M., Fox, A., Griffith, R., Joseph, A. D., Katz, R., Konwinski, A., Lee, G., et al. (2010). A view of cloud computing. *Communications of the ACM*, 53(4), 50–58. ACM. Retrieved from <http://portal.acm.org/citation.cfm?id=1721672>
- [7] Anderson, D. P., & Fedak, G. (2006). The Computational and Storage Potential of Volunteer Computing. *Sixth IEEE International Symposium on Cluster Computing and the Grid (CCGRID'06)*, 73-80. Ieee. doi:10.1109/CCGRID.2006.101
- [8] Chandra, A., & Weissman, J. (2009). Nebulas: Using distributed voluntary resources to build clouds. *Proceedings of the 2009 conference on Hot topics in cloud computing* (pp. 2–2). USENIX Association. Retrieved from <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.148.7267>
- [9] Arpaci, R. H., Dusseau, A. C., Vahdat, A. M., Liu, L. T., Anderson, T. E., & Patterson, D. A. (1995). The Interaction of Parallel and Sequential Workloads on a Network of Workstations. *Science* (Vol. 23). ACM. Retrieved from <http://portal.acm.org/citation.cfm?id=223618>
- [10] Gupta, A., & Awasthi, L. K. (2009). Peer enterprises: A viable alternative to Cloud computing? *Internet Multimedia Services Architecture and Applications (IMSAA), 2009 IEEE International Conference on* (Vol. 2, pp. 1–6). IEEE. Retrieved from http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=5439456
- [11] Weissman, J. B., Sundarrajan, P., Gupta, A., Ryden, M., Nair, R., & Chandra, A. (2011). Early experience with the distributed nebula cloud. *Proceedings of the fourth international workshop on Data-intensive distributed computing* (pp. 17–26). ACM. Retrieved from <http://portal.acm.org/citation.cfm?id=1996019>
- [12] Amira Bradai, thèse doctorat, Secured trust and reputation system: analysis of malicious behaviors and optimization, (2014)
- [13] Deutsch, M. (1962). Cooperation and trust: Some theoretical notes. *Nebraska Symposium on Motivation*, vol10, N4, p.230–275, 1962.
- [14] Luhmann, N. (1979). *Trust and Power*. 208p, John Wiley & Sons, 1^{ère} édition, 1979.
- [15] Gambetta, D. (2000). Can we trust trust? *Trust: Making and Breaking Cooperative Relations*, p.213-237, 2000.
- [16] Mills, D. H. (1983). The Logic and Limits of Trust. *Business and Professional Ethics Journal*, Vol. 2, N3, p.77–78, 1983.
- [17] Marsh, S. P. (1994). *Formalising Trust as a Computational Concept*. Thèse de doctorat, University of Stirling, 1994
- [18] Rempel, J. K., & Holmes, J. G. (1986). How do I trust thee? *Psychology Today*, Vol. 20, N2, p.28–34, 1986.
- [19] Golbeck, J. A. (2005). *Computing and applying trust in web-based social networks*. Thèse de doctorat, University of Maryland, 185p, 2005.

REFERENCES BIBLIOGRAPHIQUES

- [20] Grandison, T., & Sloman, M. (2003). Trust Management for Internet Applications. Thèse de doctorat, University of London, 252p, 2003.
- [21] Divakarla, U., & Sekaran, K. C. (2015). Trust Models in Cloud : A Survey on Pros and Cons. *New Trends in Networking, Computing, E-learning, Systems Sciences and Engineering*, LNCS, Vol. 312, p.335-341, 2015.
- [22] Grandison, T., & Sloman, M. (2000). A survey of trust in internet applications. *IEEE Communications Surveys & Tutorials*, Vol. 3, N4, p.2-16, 2000.
- [23] 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.
- [24] Ries, S., Habib, S. M., Mühlhäuser, M., & Varadharajan, V. (2011). Certain Logic: 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.
- [25] Jøsang, A., & Ismail, R. (2002). The Beta Reputation System. *Proceedings of the 15th Bled Electronic Commerce Conference*, Vol. 160, p.324-337, 2002.
- [26] VOYER Pierre (1999), *Tableaux de bord de gestion et indicateur de performance*, Edition Presse de l'université de Québec, Québec, page 439.
- [27] Chafika Madougousariki, (2014). Evaluation de la performance d'un service de gestion : cas de la loterie nationale du Benin (LNB), p.26-27, 2014.
- [28] BOURGUIGNON Alain (2000), *Performance et contrôle interne*, Edition d'organisation, Paris, pages 1025.
- [29] JACQUOT Thierry & MILKOFF Richard (1999), *Comptabilité de gestion : analyse et maîtrise des coûts*, Pearson Education, Paris, 319 pages.
- [30] TAHON Christian (2003), *Evaluation des performances des systèmes de production*, Hermès science publications, Paris, 304 pages
- [31] LÖNING Hélène, MALLERET Véronique, MERIC Jérôme, PESQUEUX Yvon, CHIAPELLO Ève, MICHEL Daniel, SOLE Andreu (2008), *Le contrôle de gestion : organisation, outils et pratiques*, 3^{ème} édition, Dunod, Paris, 304 pages.
- [32] VOYER Pierre (2002), *Tableaux de bord de gestion et indicateurs de performance*, 2^{ème} édition, Presses de l'Université de Québec, Québec, 446 pages.
- [33] PRESQUEUX Yvon (2002), *Organisation : modèles et représentations*, Edition Presses Universités de France, Collection Gestion, Paris, 397 pages.
- [34] Nabeel Zanoon (2015), *Toward Cloud Computing: security and performance*, *International Journal on Cloud Computing: Services and Architecture (IJCCSA)*, Vol. 5, No. 5/6, page.18-19, 2015
- [35] Qingtao Wu, Xulong Zhang, et al..., *Reputation Revision Method for Selecting Cloud Services Based on Prior Knowledge and a Market Mechanism*, *The Scientific World Journal* Volume 2014, Article ID 617087, 9 pages
- [36] A. Ghaffarinejad et M. K. Akbari, «Un mécanisme de réputation incitatif compatible et distribué basé sur la similitude de contexte pour les systèmes axés sur les services», *Future Generation Computer Systems*, vol. 29, non. 3, pp. 863-875, 2013.
- [37] F. Faniyi et R. Bahsoon, «La conformité SLA autogérée dans les architectures cloud: une approche basée sur le marché», dans *Actes du 3^{ème} symposium international ACM SIGSOFT sur Architecting Critical Systems*, pp. 61-70, ACM, 2012
- [38] J. Gorner, J. Zhang et R. Cohen, «Amélioration de la modélisation de la confiance à travers la limite de la taille du réseau de conseillers et l'utilisation des renvois», *Electronic Commerce Research and Applications*, vol. 12, non. 2, pp. 112-123, 2013.

REFERENCES BIBLIOGRAPHIQUES

- [39] S. Zhao, G. Wu, G. Chen et H. Chen, «Sélection de service en fonction de la réputation fondée sur la similarité QOS», Journal of Networks, vol. 6, non. 7, pp. 950-957, 2011.
- [40] https://fr.wikipedia.org/wiki/Microsoft_Visual_Studio
- [41] https://fr.wikipedia.org/wiki/C_sharp.
- [42] <http://www.ecma-international.org/publications/files/ECMA>.
- [43] <http://www.mosaique-info.fr/glossaire-web-referencement-infographie-multimediainformatique/m-glossaire-informatique-et-multimedia/448-mysql-definition.html>