République algérienne démocratique et populaire
Ministre de l'enseignement supérieur et de la recherche scientifique
Université Saad Dahlab blida 1
Faculté Des Sciences
Département d'Informatique

Mise en œuvre d'un mécanisme de sécurité collaboratif dans Fog computing

Réalisé par
Remmide Mohamed Abdelkarim
Driouech Mohamed Farid ;

Travail présenté devant devant le jury composé de
M^{me} Mancer Yassmine
M^{me} Aroussi Sana
Promotrice : M^{me} Boustia Narimane
Proposé et encadré par : M^{me} Boukhari chahira

10 septembre 2020

Remerciements

Nous remercions ALLAH qui a exaucé nos prières et qui nous a donnés non seulement le courage mais aussi la force et la patience de réaliser ce travail

Nous tenons à remercier notre promoteur, M^{me} Boustia, pour tout le soutien, l'aide et les conseils qu'elle nous a apportés pendant le processus de recherche, ainsi que pour sa patience et son temps inconditionnel.

Nous remercions aussi notre encadreur M^{me} Boukhari, qui est même dans cette période difficile du confinement, elle n'a pas cessé de nous aidé.

Nous tenons à remercier également l'ensemble des enseignants du département d'informatique de notre université, qui ont assurés notre formation tout au long des 5 années d'études, et nous ont transmis leur savoir sans réservé.

Nous présentons nos remerciements aussi aux membres du jury qui nous ont fait l'honneur d'avoir consacré leur temps précieux pour nous jugé.

Dédicace

Je dédis ce travail

A mes chers parents qui m'ont toujours entouré avec tout leur amour et se sont sacrifié pour veiller à m'offrir le meilleur et m'encourager à avancer, j'espère avoir été à la hauteur de tous qu'ils m'ont apporté.Et à mon A mon frère. Toute ma famille et mes amis pour leur présence et tous ces moments de bonheurs passés à leurs cotés.

AbdelKarim

je dédie ce travail a mes parents et surtout a ma mère d'abord, qui est grâce a elle que je suis la maintenant, je ne peux jamais la rendre même un peu de ce qu'elle m'a donné. A ma famille aussi, commençant par mon frère, qui était toujours avec moi, et qu'il ma soutenu dans tous les pas de ma vie et qu'il ma donné beaucoup. A mes sœurs aussi pour leurs soutient qui m'a donné beaucoup de courage. Et terminant par les petites fils et filles de la famille, du plus grand a plus petit, Iyed, Mohamed, Aymen, Kamilia, Milina, Achraf. Ainsi a tous mes amis de l'université.

MohamedFarid,Merci.

Résumé

La confiance peut être définit comme la croyance en la compétence d'une entité pour agir de manière fiable et sécurisée dans un contexte spécifié. Nous allons l'introduire dans le fog pour le sécuriser contre les attaques célèbres du réseau. Il existe plusieurs mécanismes pour le calcul de confiance, chacun d'eux se base sur des méthodes différents, il y en a des solutions qui se base sur l'utilisation de l'apprentissage automatique avec la régression linéaire pour avoir la valeur de confiance, d'autres utilisent la logique floue qui est une extension de la logique booléenne ...etc. Nous nous allons utilisé des formules mathématiques qui se basent sur des métrique de performance et des poids pour donner une importance aux éléments par rapport a d'autres qui dépend de notre besoin de quoi sécuriser dans le calcul de confiance, ces calculs vont être faites dans certains composants du système tels que le broker, autorité de réputation, gestionnaire de connexion... pour les permettre a prendre une décision concernant les clients et les fournisseurs de services (serveurs fog).

Donc l'objectif de notre travail consiste a donner une état de l'art des différents solutions de calcul de confiance qui existent dans la littérature. Ensuite proposer un modèle de confiance qui se base sur le calcul de la réputation et des recommandations en prenant en compte la notion de risque dans le calcul pour estimer le comportement futur des nœuds. Nous allons ensuite valider la solution avec une simulation en utilisant Java. Pour conclure avec des résultats et des améliorations de ce modèle qui peuvent se faire dans le futur.

Mots clés : Fog computing, confiance, risque, réputation.

Abstract

trust can be defined as the belief in the competence of an entity to act reliably and securely in a specified context. We will introduce it into the fog to secure it against famous network attacks. There are several mechanisms for trust calculation, each of them is based on different approachs, some of them are based on the use of machine learning with linear regression to have the trust value, others use fuzzy logic which is an extension of Boolean logic where the truth values of the variables instead of being true or false are real numbers between 0 and 1. We are going to use mathematical formulas which are based on performance metrics and weights to give an importance to some elements compared to others, these calculations will be made in certain components of the system such as the broker, reputation authority , connection manager ... to allow them to make a decision concerning service requester(client) and service providers (fog).

So the objective of our work is to give a study of the different trust computation solutions that exist in the literature. Then propose a trust model which is based on the calculation of the reputation and recommendations taking into account the notion of risk in the calculation to estimate the future behavior of the nodes. We will then validate the solution with a simulation using Java. To conclude with results and improvements of this model that can be done in the future.

Keywords : Fog computing, trust, risk, reputation

الملخص

الثقة يمكن تعريفها على أنها الإيمان بكفاءة الكيان للعمل بشكل موثوق وآمن في سياق محدد. سنقوم بإدخاله في الضباب لتأمينه ضد هجمات الشبكة الشهيرة. هناك عدة آليات لحساب الثقة ، كل منها يعتمد على طرق مختلفة ، هناك حلول تعتمد على استخدام التعلم الآلي مع الانحدار الخطي للحصول على قيمة الثقة ، يستخدم البعض الآخر المنطق الضبابي وهو امتداد للمنطق المنطقي حيث تكون قيم الحقيقة للمتغيرات أرقام حقيقية بين ٠ و ١ بدلاً من أن تكون صحيحة أو خاطئة. سنستخدم الصيغ الرياضية التي تستند إلى مقاييس الأداء والأوزان لإعطاء أهمية للعناصر، سيتم إجراء هذه الحسابات في مكونات معينة من النظام مثل الوسيط، سلطة السمعة ، مدير الاتصال ... للسماح لهم باتخاذ قرار بشأن العملاء ومقدمي الخدمة (خوادم الضباب).

لذا فإن الهدف من عملنا هو تقديم الحلول المختلفة لحساب الثقة الموجودة في الأدبيات. ثم اقترح نموذج ثقة يستند إلى حساب السمعة والتوصيات مع مراعاة فكرة المخاطرة في الحساب لتقدير السلوك المستقبلي للعقد. سنقوم بعد ذلك بالتحقق من صحة الحل من خلال محاكاة باستخدام جافا. لنختتم بالنتائج والتحسينات على هذا النموذج التي يمكن القيام بها في المستقبل

الكلمات المفتاحية الحوسبة الضبابية، الثقة ، تسير الاخطار

Table des matières

Table des figures

Liste des tableaux

Liste des abréviations

IoT : Internet of Things
IdO : Internet des Objets
TN : Terminal Noeud
RA : Réalité augmenté
PKI : Public key infrastructure
MANET : Mobile Ad Hoc Network

Introduction générale

L'Internet des objets changera non seulement notre vie quotidienne, mais aussi le monde de l'industrie et des entreprises. D'ici 2025, près de 75,44 milliard d'objets seront connectés [8] et d'énormes quantités des données seront générées (90% des données mondiales ont été générées au cours des deux dernières années [48]).En supposant que ces informations puissent être directement intégrées dans le centre de données ou le cloud, ce sera très simple. De plus, en termes de latence et de saturation de la bande passante, l'envoi des milliards de données sur Internet peut rapidement devenir un problème. C'est là que le cloud computing cède la place au fog computing [49].

Le fog computing, aussi appelé «informatique dans le brouillard», définit une infrastructure chargée de stocker et de traiter des données issues d'objets connectés. On peut le considérer comme un concurrent du cloud, car il peut fournir les mêmes services que le cloud, ou une solution complémentaire à ce dernier, car il réduit la latence pour les objets connectés et réduit les demandes de services de cloud par les appareils IoT.

Le fog computing étant un nouveau domaine de recherche, plusieurs problèmes ont été soulevées. Outre que les problèmes hérités du cloud, il apporte également de nouveaux défis en matière de confidentialité et de sécurité. Les deux paradigmes ayant des architectures très différentes, implique que les solutions apportées dans le cloud ne peuvent pas être directement appliquées dans le fog.

L'un des problèmes est de savoir comment établir la confiance entre les nœuds du fog computing.

A travers ce mémoire, nous proposons un modèle de confiance qui répond aux exigences de fog computing dans un environnement collaboratif. Ce nouveau modèle permet la collectivité entre les nœuds de confiance dans le fog.
Nous avons introduit dans notre système la notion de risque, la réputation a travers l'accumulation des feedback et l'utilisation de l'architecture basé sur les brokers.

Ce présent mémoire est organisé comme suit :

Chapitre 1 est consacre a la présentation du fog computing, ses exigences, ainsi que quelques notions relatives a la confiance et a la sécurité.

Chapitre 2 présente une taxonomie des solutions proposées dans le calcul de confiance.

Chapitre 3 nous avons décrit le fonctionnement général de notre système de confiance dans Fog computing, ainsi que ses différentes composants.

Chapitre 4 décrit l'implémentions de notre système et se termine par une phase de test.

Chapitre 1

Présentation du Fog computing et la confiance

Introduction :

L'internet des objets désigne les échanges d'informations et de données numériques entre les objets présents dans le monde réel et le réseau Internet, elle est partiellement responsable de l'accroissement du volume de données générées sur le réseau. En conséquence, l'Internet des Objets apporte de nouvelles contraintes informatiques, tels que la latence et la saturation de la bande passante. Le Cloud Computing n'est pas adapté pour répondre aux besoins d'Internet des objets. C'est pourquoi, le nouveau paradigme «Fog computing» a été introduit, ce dernier se distingue par sa proximité avec ses utilisateurs, ce qui lui permet de résoudre pas mal de problèmes du réseau, mais il reste toujours le souci de sécurité de ce réseau.
Pour améliorer la sécurité, plusieurs mécanismes ont été proposés. Nous nous intéressons à un des mécanismes de sécurité basé sur l'indice de confiance.

Dans ce chapitre, nous allons présenter le Fog computing, son architecture et ses domaines d'applications, ainsi les menaces relatives a son déploiement . Par la suite, nous détaillons la confiance, ses dimensions, pourquoi la confiance dans le Fog, les exigences, et les attaques existantes dans ce réseau fog.

1.1 Fog computing

1.1.1 Définition

Les chercheurs de Cisco ont proposé pour la première fois le terme «Fog Computing» ou l'informatique «en brouillard» en 2012. Depuis, de nombreux chercheurs ont donné différentes définitions du Fog Computing. Voici quelques exemples :

- "Le Fog Computing est une plate-forme hautement virtualisée qui fournit des services de calcul, de stockage et de mise en réseau entre les appareils IoT et les centres de données de Cloud computing, elles est situé à la limite du réseau pour réduire la latence, la bande passante et augmenter la fiabilité." [2]
- "Le Fog computing est un scénario dans lequel des nombreux appareils hétérogènes (sans fil et parfois autonomes), omniprésents et décentralisés communiquent et coopèrent potentiellement entre eux pour effectuer des tâches de stockage et de traitement sans l'intervention d'un tiers." [3]
- "Le terme «Fog computing» ou «Edge Computing» signifie que le système de Fog n'est pas hébergé et fonctionne dans un Cloud centralisé, mais s'exécute à l'extrémité du réseau. Le terme fait référence à la mise en place de certains processus et ressources à la limite de Cloud, plutôt qu'à l'établissement des canaux de stockage et d'utilisation du Cloud." [4]
- "Le Fog computing est une plate-forme informatique distribuée dans laquelle la majeure partie du traitement est effectuée par des périphériques finaux ou périphériques virtualisés et non virtualisés. Il est également associé au Cloud pour effectuer un traitement insensible à à la latence et un stockage à long terme de données." [6]

.

Le Fog Computing définit par l'OpenFog Consortium [5] comme "Une architecture horizontale au niveau du système qui distribue les fonctions de calcul, de stockage, de contrôle et de réseau à proximité des utilisateurs dans tout le continuum, du Cloud aux objets". D'autre part, Cisco [1] définit le Fog computing comme une extension de Cloud computing pour rapprocher le Cloud aux objets qui génèrent et agissent sur les données IoT. Ces appareils sont appelés nœuds de Fog, qui peuvent être déployés n'importe où en les connectant à Internet.

1.1.2 Motivation

Le Fog computing a été proposé pour répondre aux limites du Cloud computing.
Selon le rapport Statistica [8] d'ici 2025, le nombre total d'appareils connectés dans le monde

est d'environ 75,44 milliards, et toutes les données doivent être transmises depuis tous les appareils connectés pour être traitées dans le Cloud. Cela nécessite beaucoup de bande passante et d'espace de stockage, ce qui entraînera d'énormes problèmes pour le Cloud. Nous avons aussi la nature centralisée du Cloud computing, donc la communication entre les utilisateurs et le Cloud a un temps de réponse très long, et une fois qu'un grand nombre d'appareils seront connectés, la latence augmentera.

1.1.3 Architecture du Fog computing

L'architecture de Fog computing peut être représentée de deux manières différentes : par une structure hiérarchique selon Sarkar et al. [10] ou une structure en couches par Aazam and Huh [9].

L'architecture à trois niveaux (comme le montre la figure 1.1) est l'une des structures hiérarchiques de base les plus largement utilisées dans le domaine du Fog computing.

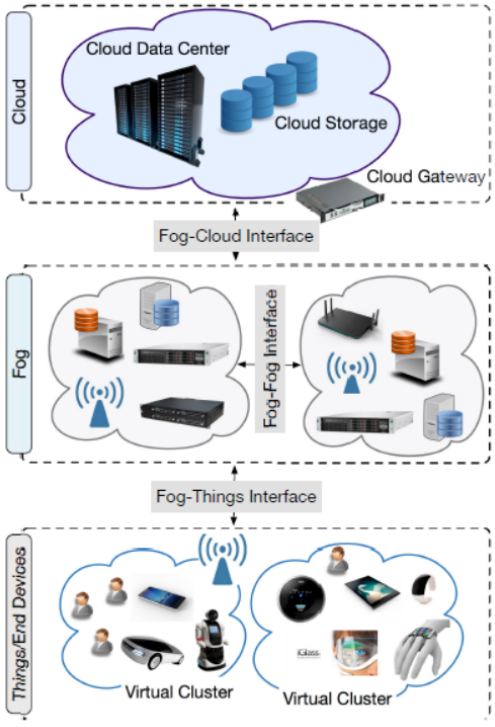

FIGURE 1.1 – L'architecture à trois niveaux dans le Fog computing

— Le premier niveau comprend les dispositifs terminaux, les dispositifs compatibles d'Internet des Objets (Id0), y compris les nœuds capteurs, les dispositifs intelligents, etc. Ces dispositifs terminaux peuvent également être appelés "nœuds terminaux" (TN).

— Le deuxième niveau représente la couche de Fog computing. Il est composé des routeurs, des passerelles et de commutateurs... etc, qui peuvent partager des ressources de stockage et de calcule.
— Le troisième niveau est le Cloud, qui se compose des centres de données et fournit des ressources de calcul et de stockage suffisantes.

Le principal avantage de l'architecture à trois niveaux est une meilleure gestion du service. Les services sensibles au temps sont traités par des nœuds Fog les plus proches de l'utilisateur. En outre, les services de confidentialité et de protection de la vie privée exigent également que de nombreux types de données et d'informations soient traités localement, plutôt que les envoyés sur des réseaux publics, ce qui souligne également l'importance de l'architecture de trois niveaux. Comme nous pouvons le voir sur la figure 1.1 il existe plusieurs modèles hiérarchiques de déploiement de Fog (y compris le déploiement de Fog a N niveau) recommandés pour divers cas d'utilisation du Fog computing.

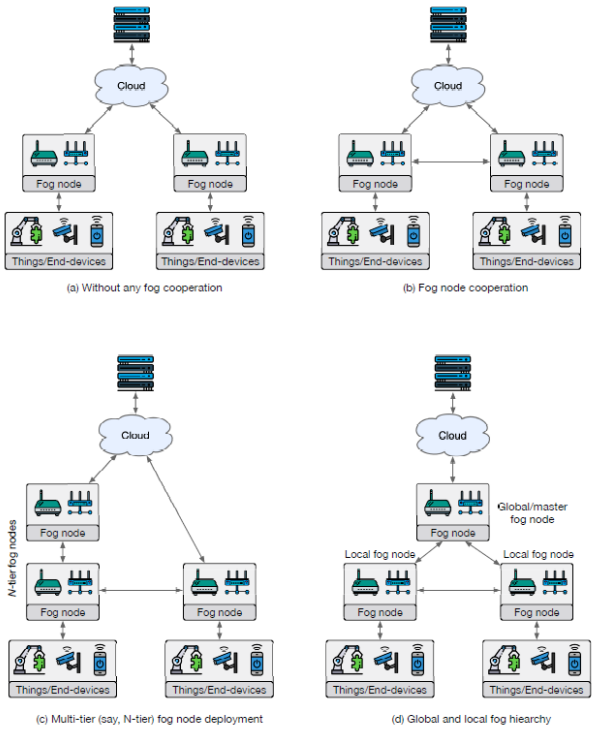

FIGURE 1.2 – Illustration des modèles hiérarchiques de déploiement du Fog [5]

L'architecture en couches se compose des couches suivantes : couche physique et de virtualisation, couche de surveillance, couche de pré-traitement, couche de stockage temporaire, couche de sécurité et couche de transport.

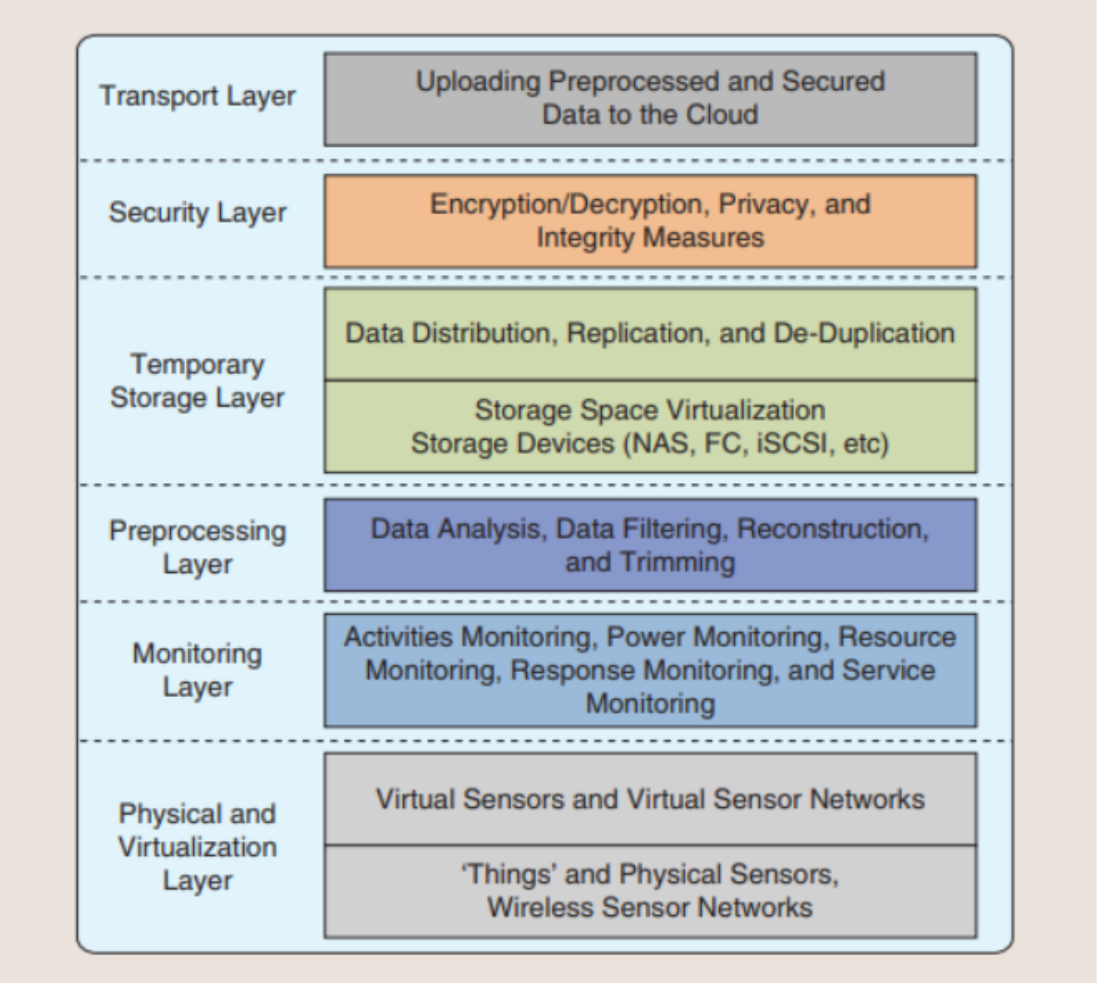

FIGURE 1.3 – L'architecture en couches du Fog [5]

Les nœuds physiques et les capteurs virtuels sont situés et gérés au niveau de la couche physique et de la couche de virtualisation. La couche de surveillance est responsable d'analyser la mise à disposition des tâches demandées et la consommation d'énergie de l'équipement physique. Les tâches liées à la gestion des données sont effectuées dans la couche de pré-traitement. La couche de stockage temporaire est responsable du stockage des données pendant une durée limitée. Les problèmes liés à la sécurité sont gérés dans la couche de sécurité. Le transfert de données vers le Cloud s'effectue au niveau de la couche de transport.

1.1.4 Différence entre Fog et Cloud

Le Fog computing et le Cloud computing sont interdépendants dans la fourniture de ressources de stockage et de calcul. Cependant, ce sont des paradigmes informatiques différents. Le tableau 1.1 résume les principales différences dont nous citions :

	Cloud computing	Fog computing
Gestion	Centralisée	Distribuée
Taille	Très grands centres de données	Un grand nombre de petits nœuds
Latence	Élevée	Faible
Capacité de calcul	Très élevée	Faible
Évolutivité	Moyenne	Élevée
Nature de l'échec	Prédictible	Divers
Coût de déploiement	Élevé	Faible

Table 1.1 – Comparaison entre le Cloud et Fog computing

— La proximité du Fog avec les nœuds sous-jacents.
— La gestion : Le Cloud est géré de manière centralisée, en revanche, la gestion dans le Fog est distribuée.
— La latence : En raison de la position du Fog, son délai de traitement est réduit par rapport au cloud.
— En raison de la connectivité sans fil dominante et de la gestion décentralisée, le taux d'échec dans le Fog est élevé que celui du Cloud.

Il convient de noter que le Fog ne peut pas remplacer le Cloud. Les deux technologies ont des contributions différentes pour améliorer les performances.

1.1.5 Fonctionnement du Fog computing

En fonction de la sensibilité des données, le Fog computing dirigera les données vers le meilleur emplacement [1] :

— Les données les plus sensibles au temps sont analysées sur le nœud Fog le plus proche des éléments qui ont généré les données.
— Les données dont l'action peut prendre quelques secondes ou quelques minutes seront transmises au nœud d'agrégation pour l'analyse.
— Les données moins sensibles au temps sont envoyées au Cloud pour l'analyse ou le stockage a long terme.

1.1.6 Domaines d'application

D'après Cisco et IBM le Fog computing convient aux situations suivantes : où les données sont collectées à les limite du réseau, Applications nécessitant une latence extrêmement faible et prévisible et les applications réparties géographiquement. C'est pourquoi le Fog computing a sa place dans plusieurs domaines.

Dans la figure 1.4 nous présentons quelques domaines auxquels le Fog apportera d'énormes bénéfices :

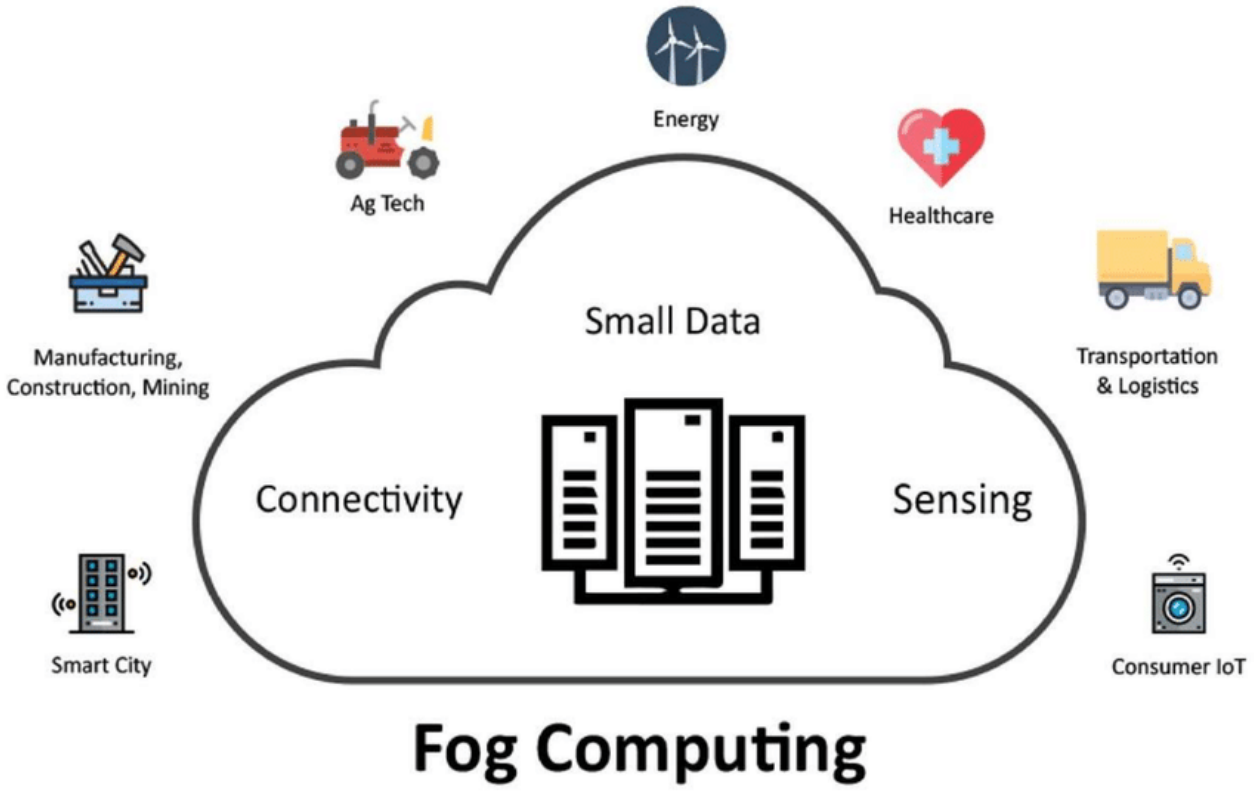

FIGURE 1.4 – Le domain d'application de Fog computing [53]

Villes intelligentes : Dans le cadre des «villes intelligentes», [12] a été proposé d'utiliser les sites Fog comme une plateforme de traitement pour collecter les données des capteurs de la ville. Le Fog computing est aussi utilisé pour contrôler le trafic routier, l'idée est de transmettre l'état des feux de circulation aux voitures à proximité pour les ralentir [11].

Domaine de la sante : Dans le domaine de la santé, Dubey et al. [50] proposent d'utiliser le Fog dans la surveillance médicale. L'idée est que des capteurs mesurent différents paramètres de la personne. L'idée proposée est d'introduire un Fog entre l'utilisateur

et le Cloud afin d'agréger les données. Seul un agrégat des données rejoint le Cloud. L'infrastructure de Fog est également utilisée pour détecter des motifs dans les données qui signifient que la personne est en danger. L'objectif est de pouvoir être alerté le plus rapidement possible sans attendre que le calcul soit traité sur une plateforme de Cloud. D'autres auteurs [51], [52] proposent quant à eux d'utiliser le Fog pour détecter les accidents vasculaires cérébraux.

Robotique : Dans le domaine de la robotique, le Fog computing est utilisé pour avoir des capacités de prise de décision plus rapides.

La mise en cache de contenus : est un cas d'usage très fréquemment proposé. L'idée générale est d'utiliser l'infrastructure de Fog pour réduire le temps d'accès aux données en copiant le contenu à proximité de l'utilisateur.

Réalité augmente (RA) : La technologie de réalité augmentée est une technologie qui utilise la technologie de l'infographie pour combiner la réalité virtuelle avec le monde réel sous la forme d'images visuelles améliorées numériquement en temps réel. Les applications qui reposent sur la technologie RA doivent inévitablement nécessiter une bande passante élevée pour transmettre des données et des calculs à haute puissance pour fournir une diffusion en temps réel. ceci est principalement dû au fait que même un délai très court ou une mise en tampon (tels que des interruptions) peuvent endommager la présentation de l'utilisateur..

Il existe d'autres domaines, tels que la défense, le transport, le traitement vidéo et les grilles intelligentes..etc

1.1.7 Problèmes de sécurité et de confidentialité dans le Fog computing

Dans cette section, nous abordons certains problèmes de sécurité et de confidentialité dans le Fog computing :

Confiance : Le réseau IoT doit fournir des services fiables et sécurisés aux utilisateurs finaux. Cela nécessite que tous les appareils faisant partie du réseau Fog se fassent confiance. Dans ce cas, la confiance joue un rôle important pour la promotion des relations basées sur des interactions antérieures.

Authentification : L'une des principales exigences du réseau Fog est l'authenfication. Pour accéder aux services du réseau Fog, l'appareil doit d'abord faire partie du réseau en en authentifiant au réseau Fog. Ceci est essentiel pour empêcher l'entrée non autorisée des nœuds. En raison des contraintes de ressources des appareils IoT, les

mécanismes d'authentification traditionnels utilisant des certificats et l'infrastructure à clé publique (PKI) ne conviennent pas.

Attaques malicieuses : L'environnement de Fog computing peut être soumis à de nombreux attaques malveillantes, et sans mesures de sécurité appropriées, les fonctions du réseau peuvent être gravement endommagées.

La privacy (la protection de la vie privée des utilisateur) : Dans le Fog computing, le maintien de la confidentialité devient difficile. Comme le nœud de Fog est très proche de l'utilisateur final, un compromis de nœud mal sécurisé peut être le point d'entrée pour voler et extraire des donnes sensible.

Communication sécurisée : Pour protéger la communication dans le Fog computing, les connexions suivantes entre les nœud du réseau doivent être sécurisées :

1- La communication entre les nœuds IoT et les nœuds Fog.

2- Communication des noeuds Fog entre eux.

1.2 La confiance

1.2.1 Définition

En informatique, la confiance est un terme largement utilisé et sa définition varie selon les chercheurs et les domaines d'application.

— Mui et al. [15], Définissent la confiance comme «l'attente subjective d'un agent par rapport au comportement futur d'un autre agent sur la base de son historique des rencontres» Dans cette définition, la confiance est basée sur l'expérience passée des agents, c'est-à-dire les interactions qu'ils ont été réalisée.

— "La conviction qu'une entité a la capacité d'agir de manière fiable et sûre dans un contexte défini." [16] dans cette deuxième définition de Grandison and Sloman, la confiance fait référence à la compétence d'une entité dans un contexte unique.

— "La confiance de la partie A dans le service X de la partie B est une croyance mesurable, c'est-à-dire que la performance de la partie B dans le contexte défini est fiable dans la période définie." [17] Olmedilla et al. on fait référence aux actions plutôt que la compétence comme la définition précédente.

— Selon Neisse et al. [18] la confiance est "la mesure de la croyance du point de vue d'une partie qui fait confiance (trusteur) par rapport à une partie de confiance (trustée) axée sur un aspect spécifique de la confiance qui peuvent impliquer des gains ou des risques."

1.2.2 Modèle de confiance

Le modèle de confiance peut être défini comme un modèle mathématique de tous les aspects de la relation de confiance. Ces modèles sont généralement utilisés pour établir et gérer la relation de confiance entre les nœuds du réseau afin d'atteindre les objectifs de sécurité [22].

1.2.3 Dimensions de confiance

Selon Guo et al. [19] le modèle de confiance se compose de 5 dimensions de base : trust composition, trust propagation, trust update, trust formation et trust aggregating.

Trust composition : Trust Composition définit les éléments à prendre en compte dans le calcul de la confiance. Nous avons la confiance de la qualité de service (QoS) et la confiance sociale.

Trust propagation : Cette partie du calcul de confiance décrit comment stocker les valeurs de confiance et comment la propagation se faite dans le réseau. Deux scénarios principaux sont décrits dans la littérature. Distribué et centralisé.

Trust update : Concerne la mise à jour de la confiance. En général, il y a deux méthodes l'une basée sur les événements et l'autre sur le temps.

Trust formation : Dans la littérature, la formation de la confiance est considérée sous l'angle de la confiance unique ou de la confiance multiple.

Trust aggregating : désigne l'agrégation des données recueillies par le biais d'observations ou de rétroactions de pairs. Principales techniques d'agrégation de la confiance étudiées par Josang et al. [20] sont : La somme pondérée, la théorie de la croyance, l'inférence bayésienne, la logique floue et l'analyse de régression.

1.2.4 La confiance dans Fog computing

L'un des objectifs du Fog computing est d'améliorer la fiabilité du réseau, qui peut être attend en utilisant le calcul de la confiance. L'utilisation de la confiance dans le Fog computing permet la prédiction du comportement future des objets ce qui permet de sélectionner le meilleur.

Dans la figure 1.5 ci dessous nous allons mettre en position l'emplacement de la confiance dans l'architecture de Fog computing.

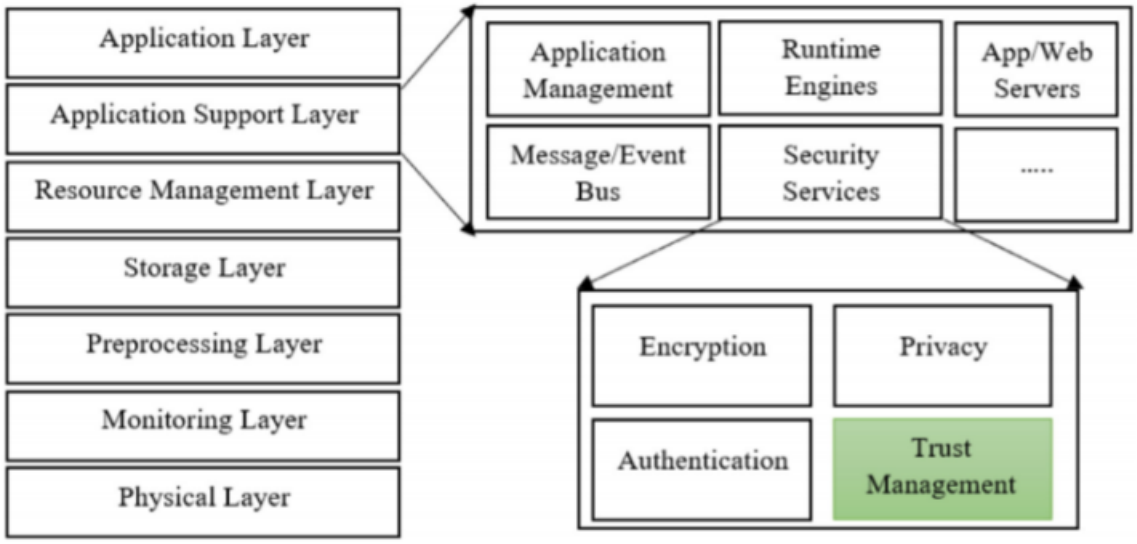

FIGURE 1.5 – La position de la confiance architecture en couches du Fog computing [5]

1.2.5 Exigences de confiance dans le Fog computing

Cho, A. Swami et I Chen [21] ont définit cinq exigences de confiance pour les réseaux mobiles ad-hoc (MANET), qui seront également incluses dans le Fog computing.

La confiance est dynamique : En raison de comportements variables des objets et le changement continu de la topologie du réseau, la confiance doit être calculé en continu.

La confiance est asymétrique : Si le client Fog B trouve que le nœud Fog A est digne de confiance, le nœud Fog A peut trouver que le client Fog B n'est pas digne de confiance.

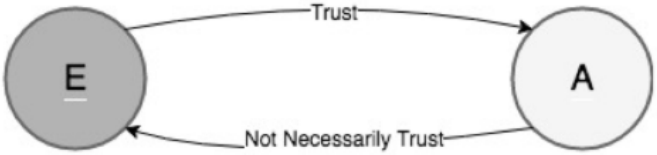

FIGURE 1.6 – La confiance n'est pas asymétrique.

La confiance n'est pas nécessairement transitive : Chaque objet a sa propre politique de sécurité. Si le client de Fog B délègue un nœud Fog A et le nœud Fog A délègue un nœud Fog C, le client de Fog B ne fait pas nécessairement confiance à un nœud Fog C. (C'est-à-dire que si E fait confiance à A, nous ne pouvons pas être sûrs que E fait confiance à B voir figure 1.7).

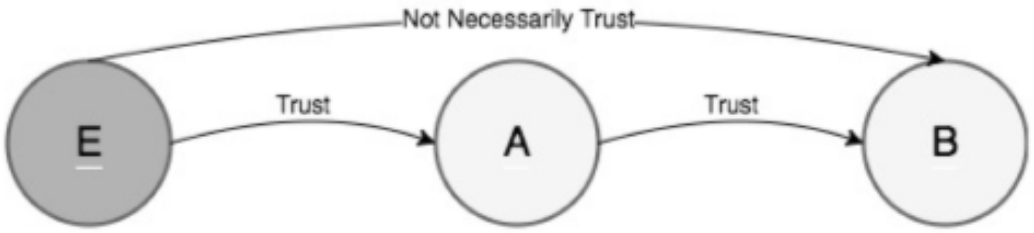

FIGURE 1.7 – Exemple de la non transitivité de la confiance

La confiance est subjective : Chaque objet a des exigences de sécurité différentes de celles des autres objets du réseau. En d'autres termes la confiance est subjective signifie que différentes applications ont des points de vue différents sur les nœuds.

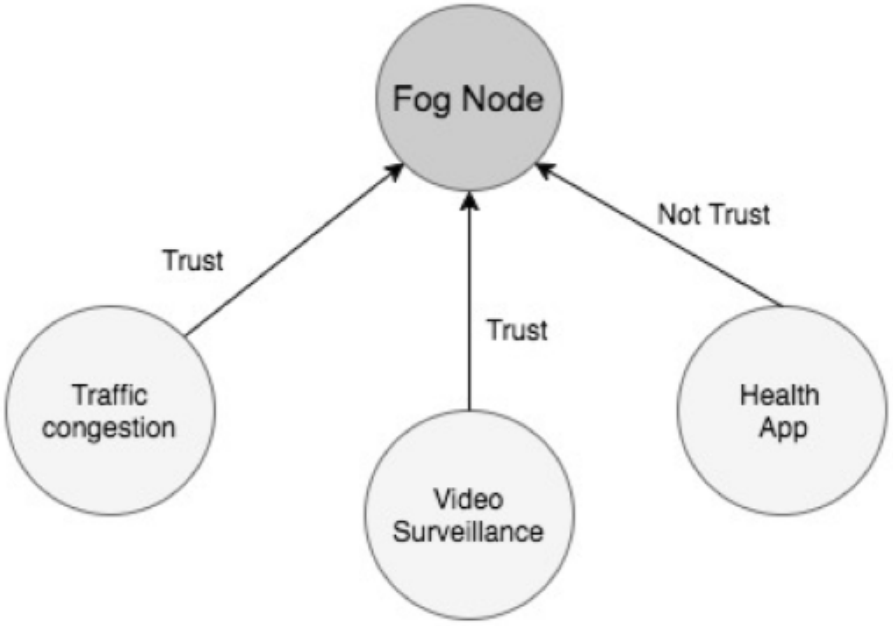

FIGURE 1.8 – Exemple réal sur comment la confiance peut être subjective

La confiance dépend du contexte : Le contexte est important pour calculer la confiance ; nous faisons confiance aux entités dans un contexte spécifique.

1.2.6 Attaques sur le calcul de confiance

L'objectif principal de système de gestion de confiance dans Fog computing est la prise de décision pour le choix de collaborateurs. Cependant, une entité malicieuses peut exécutée

une variété d'attaques afin de perturber les services assurés par le système. Les attaques les plus courantes sont les suivantes :

L'attaque Self-promotion : un nœud malveillant peut augmenter son importance (en se donnant des bons recommandations) pour être sélectionné en tant que fournisseur de services, mais peut ensuite fournir un service mauvais ou mal fonctionné.

L'attaque de mauvaise bouche : Il s'agit d'une forme d'attaque de collusion de recommandation pour détruire la confiance des bons nœuds(en lui fournissant des recommandations erronées), afin de réduire la possibilité que le nœud soit choisit pour les services.

L'attaque de ballot-stuffing : Il s'agit d'une autre forme d'attaque de collusion de recommandations. Dans ce genre d'attaques un nœud malveillant peut renforcer la confiance d'un nœud malveillant (en fournissant de bonnes recommandations) ce qui augmente la possibilité de choisir des nœuds malveillants comme fournisseurs de services.

L'attaques de service opportunistes : lorsqu'un nœud malveillant détecte que sa réputation a diminuée, il peut fournir des bons services pour retrouver sa réputation. Avec une bonne réputation, il peut collaborer efficacement avec d'autres nœuds mauvais pour effectuer des attaques de mauvaise bouche et de ballot-stuffing.

L'attaque on-off : les nœuds peuvent exécuter de manière aléatoire des bons et des mauvais services pour éviter d'être marqués comme des mauvais nœuds.

Nous concluons que les attaques sur la confiance sont classées comme suit :

— Les attaques de type self-promotion et service opportunistes sont basées sur l'intérêt personnel.

— Les attaques mauvaise bouche et ballot-stuffing sont d'une forme de coopération pour attaquer la réputation.

— Tandis que les attaque on-off sont souvent utilisées par des nœuds malveillants pour s'échapper à la détection.

Conclusion :

Dans ce chapitre, nous avons décrit l'environnement Fog computing, son architecture ainsi les domaines potentiels de son utilisation. Par la suite, nous avons abordé les problèmes de sécurité rencontré dans le Fog. Il est donc primordiale de mettre en place un mécanisme de sécurité pour la détection des nœuds malicieux. Pour cela, nous avons définit la gestion de confiance ainsi ses différents aspects, afin de contrarier les différentes attaques sur la confiance.

Dans le prochain chapitre, nous allons présenter les différents modèles de confiance proposés dans la littérature afin de faire une étude comparative.

Chapitre 2

Taxonomie des solutions proposées dans le calcul de confiance

Introduction :

Pour assurer les objectives de sécurité dans Fog computing, les chercheurs ont appliqué le concept de confiance afin d'établir des relations de confiance entre les nœuds du réseau.

Dans ce chapitre, nous allons présenter les différents modèles de confiance proposés dans la littérature pour une études comparatives.

2.1 Critères de comparaison des solutions

Afin d'évaluer les travaux que nous avons étudié dans le contexte de confiance, nous avons définît quelques critères de comparaison pour procéder a leur classification.

- La résilience : C'est la capacité d'un système d'information à résister aux cyberattaques et aux pannes accidentelles, puis à revenir à un état de fonctionnement et de sécurité satisfaisant. [37].
- La scalabilité : La capacité d'un système, ou de ses composants, à être utilisé sur des plates-formes de tailles très inférieures ou très supérieures [37].
- La consommation d'énergie : Certains objets font plusieurs traitements complexes qui consomment beaucoup d'énergie. De ce fait, le mécanisme de gestion de confiance doit être à moindre consommation d'énergie [38].
- L'exactitude : Représente le degré de similarité entre la valeur de confiance d'un nœud calculé par le système de gestion de confiance avec la confiance effective qui doit être attribuée au nœud [38].

2.2 Modèles de confiance

2.2.1 Solution proposé par Tuva Dybedokken (Trust management in Fog computing) [34]

L'auteur de [34] a utilisée deux méthodes pour l'estimation de confiance : subjective logique et la régression logistique.

Subjective logique est utilisé pour assurer qu'un objet est dans un état fiable avant d'établir une collaboration. Pour cela le tuple (b,d,u) est utilisé où b désigne croyant belief, d non croyance (disbelief) et u est l'incertitude (uncertainty). Il y a quelques défis lors de la collection des recommandations des nœuds, quelques recommandeur ne disent pas la vérité, c'est pour cela les recommandations sont calculés à partir de plusieurs critères (anciens recommandations positifs, négatifs ...) pour prédire la fiabilité d'un nœud.

Régression logistique Dans cette technique le calcule est plus lourd par rapport à subjective logique, mais elle donne des résultats précises.

Régression logistique est utilisée pour le calcul de confiance, où cette dernière est une décision binaire, soit le nœud est fiable, ou non fiable.
Avec cette méthode, on peut trouver la probabilité d'un nœud à partir d'un ensemble de variables appelles condition,comme par exemple : le comportement du réseau, les ressources nécessaires. En se basant sur ces variables (conditions d'environnement), le modèle peut prédire la fiabilité de nœud.

Le calcul de la confiance passe par deux étapes : L'estimation de la confiance de nœud Fog et l'estimation de la confiance du nœud Fog au client.

Dans l'estimation de la confiance de client Fog au noeud Fog, le modele utilie la methode «logistique regression» et passe par plusieurs etapes : Le client demande du nœuds du Fog des recommandations en lui envoyant un message du format m=j,x où j représente le nœud et x les conditions. Ensuite, les nœuds calculent la confiance du nœud j en utilisant les variables x, et renvoient la réponse au client. Le client estime la confiance du nœud j en combinant ces résultats, avec les valeurs de confiance de chaque recommandeur pour conclure une seule valeur de confiance.

L'estimation de la confiance du nœud Fog au client Fog qu'elle est basée sur la technique «Subjective logique», le principe est comme suit : D'abord le nœud demande des recommandations des autres nœuds par rapport un client. Ensuite, les nœuds répondront en envoyant des tuples (b,d,u). Par la suite, il les combinent avec les valeurs de confiance qu'il a pour chaque nœud recommandeur. Pour avoir le résultat final, le nœud combine l'ensemble des valeurs calculées dans l'étape précédente avec son propre tuple (b,d,u) du client.

Résultat et discussion :

Cette solution est efficace contre plusieurs attaques et la valeur de confiance a une exactitude forte, mais elle consomme beaucoup d'énergie lors de calcul de confiance, et elle prend beaucoup de temps à cause de calcul de confiance qui ce fait des deux sens, du client au nœud Fog et du nœud au client, et aussi a cause d'utilisation de la méthode de la regression logistic qui donne des valeurs précises mais elle est trop lourde.

2.2.2 A Risk-Aware Reputation-Based Trust Management in Wireless Sensor Networks [47]

Dans cette solution, la mesure de confiance est composée de deux parties : la valeur de confiance et la valeur de risque. La valeur globale de confiance du nœud est évaluée en soustrayant la valeur de risque de la valeur de confiance.

Pour chaque nœud, la valeur de confiance est calculée à partir de la valeur de confiance directe et de la valeur de confiance indirecte.La valeur du risque est dérivée des expériences négatives existantes.

Dans ce modèle, les nœuds n'interagissent qu'avec leurs voisins directs. Par conséquent, ils ne conservent pas d'informations de confiance sur chaque nœud du réseau. Cette conservation qui se faite sur le voisinage implique une consommation d'énergie considérablement réduite.

La valeur de confiance globale du nœud j par rapport à nœud i est :

$$Trust_{i,j} = \alpha * T_{i,j} - (1 - \alpha) * R_{i,j}, (1 \geq \alpha \geq 0) \tag{2.1}$$

Dans la formule 2.1 , $T_{i,j}$ et $R_{i,j}$ désignent respectivement la valeur de confiance et la valeur de risque du nœud j.
α et $(1 - \alpha)$ sont les pondérations des valeurs de confiance et de risque respectivement. Si

nous choisissons un plus grand α, alors la valeur globale de la confiance est moins affectée par la valeur du risque. D'un autre côté, si nous choisissons un α plus petit, la valeur globale de la confiance est plus sensible à la valeur du risque. Si nous définissons $\alpha = 1$, cela signifie que le poids du risque est 0.

Le calcul d'une valeur de confiance nécessite deux parties d'informations : la valeur de confiance directe et la valeur de confiance indirecte. La valeur de confiance directe peut être obtenue lorsqu'un nœud a des transactions directes avec un autre nœud. En réalité, un nœud peut ne pas avoir traité avec un nœud auparavant ; il devra donc s'appuyer sur la recommandation d'autres nœuds, c'est-à-dire la valeur indirecte, pour évaluer dans quelle mesure il peut faire confiance au nouveau nœud. La valeur de confiance indirecte est mesurée en agrégeant toutes les références provenant d'autres nœuds. Soit $T_{i,j}$ la valeur de confiance du nœud i concernant le nœud j. Elle est définie comme suit :

$$T_{i,j} = \Omega * DT_{i,j} - (1 - \Omega) * IT_{i,j} \tag{2.2}$$

$DT_{i,j}$ est la valeur de la confiance directe, $IT_{i,j}$ est la valeur de confiance indirecte, Ω est le facteur de confiance ($1 \geq \Omega \geq 0$) qui décrit le niveau de confiance du nœud i concernant sa valeur de confiance directe sur le nœud j.

La valeur de risque est introduite dans le calcul de confiance afin de décrire les comportements imprévisibles et incertains des nœuds malveillants. Dans ce modèle la valeur du risque est calculée en appliquant le concept d'entropie de l'information, l'entropie est un nombre qui mesure l'incertitude d'une donnée à partir de celui qui la précède, on a le nœud maintient localement une liste de proportions de deux types d'interactions S et U (interactions réussies et échouées) qu'on les utilisent pour avoir la valeur de $R_{i,j}$.

Après avoir calculé la valeur de confiance Trust i, j, un nœud j est classé par un nœud i parmi trois états comme suit :

$$Mp(Trust_{i,j}) = \begin{cases} T : trusted & \textbf{if } 1 \geq Trust_{i,j} \geq \text{THR1} \\ U : uncertain & \textbf{if } \text{THR1} \geq Trust_{i,j} \geq \text{THR2} \\ M : malicious & \textbf{if } \text{THR2} \geq Trust_{i,j} \geq 0 \end{cases} \tag{2.3}$$

Où $THR2 < THR1 < 1$, et THR1 et THR2 peuvent être réglés en fonction des exigences du système et de sécurité pour déterminer l'état du nœud.

Résultat et discussion :

L'évaluation des risques est un élément très utile pour traiter efficacement les comportements de nœuds conflictuels. Cette méthode quantifiant ainsi la crédibilité d'un recommandataire pour réduire les recommandations malhonnêtes, elle est efficace contre les attaques On-Off, mais pas pour d'autres attaques.

2.2.3 TRFIoT : Trust and Reputation Model for Fog-Based IoT

Yasir Hussain et Zhiqiu Huang [40], ont proposé un modèle basé sur la confiance et la réputation. Dans ce modèle, chaque nœud est identifié par un id unique. Le fonctionnement de ce modèle se déclenche quand un nœud IoT se communique avec un nœud de Fog, où il va partager son id avec la liste des réputations de tous les nœuds qu'il a y vais communiquer avec, cette liste sera utilisé dans le processus d'évaluation de confiance des autres nœuds. Après l'établissement de la connexion avec le nœud IoT, le nœud de Fog va calculer la valeur de confiance en collaborant avec les nœuds voisins réputé qui seront trouvé par un processus de calcule de réputation qui utilise une version modifiable de l'algorithme de Page Rank. L'équation d'évaluation de confiance est définit comme suit :

$$TV_{D_{id}} = \alpha * TV_{N_{id}} + \beta * \sum_{j=1}^{n} \frac{TV_{N_{id}^{RV^j}}}{J} + \gamma * \sum_{k=1}^{n} \frac{TV_{N_{id}^{RV^k}}}{K} \tag{2.4}$$

Où TV_N est la valeur de confiance du nœud N local.RV^j etRV^k Sont les nœuds IoT et Fog réputé. TV_N^{RV} Est la valeur de confiance du nœud N depuis un autre nœud (recommandation).J, K sont le nombre total des nœuds IoT et Fog respectivement, TV est la valeur de confiance, et α, β et γ sont les facteurs bondrée.

Sinon si ni le nœud actuel ni les nœuds voisins ont la valeur de confiance de ce nœud IoT le système va l'attribué une valeur par défaut égale 0.5 et le surveille afin de garder trace de son comportement.

Après la complétion de processus d'évaluation de la confiance, le système envoi une valeur à tous les nœuds participe dans le calcul afin de les informer si le nœud a gain ou perd de la confiance.

A la fin, la valeur de la confiance en plus des données collectées depuis le nœud IoT seront transmet au cloud qui va ensuite envoyer une récompense / punition pour le nœud.

Résultat et discussion :

Le modèle proposé a résolu le problème de cold star ainsi il assure la fiabilité, efficacité, et il est objectif dans l'évaluation de confiance et la prise de décision. Mais dans ce travail ils ont considéré que les nœuds de Fog sont des nœuds réputé mais ces nœuds peuvent êtres malveillants et doivent être évalués.

2.2.4 A Fog-based Hierarchical Trust Mechanism for Sensor-Cloud Underlying Structure [46]

Confiance directe entre les nœuds :

Il existe de nombreuses caractéristiques de comportement qui peuvent être observées pour évaluer l'état de confiance des nœuds. Cependant, plus les caractéristiques sont concernées, plus la mise en œuvre du système devient difficile en raison de certaines restrictions, telles que la consommation d'énergie, la charge du réseau, etc. Dans cette solution ils ont choisi le taux de perte de paquets, le taux d'échec de route et le délai de traitement comme indicateurs d'évaluation pour évaluer la confiance des nœuds.

Le taux de perte de paquets fait référence à la proportion des paquets perdus du total des paquets de données dans un cycle de communication. Le taux d'échec de routage fait référence à la proportion des paquets rejetés du total des paquets de routage envoyés par les expéditeurs pendant un intervalle de temps.

Le délai de traitement fait référence à l'intervalle de temps entre la réception des données et la transmission des données pour les nœuds relais.

Ces 3 caractéristiques sont causées par les nœuds de panne, les nœuds attaqués et les nœuds malveillants. Le nœud source peut les utiliser pour établir une relation de confiance directe avec les nœuds coopératifs. La formule de la confiance directe est indiquée comme (2.5).

$$Trust_{direct} = (w_1 Trust_{packet} + w2 Trust_{history}) * Delay_{forwarding} \tag{2.5}$$

Lorsque l'intervalle de temps est supérieur au seuil, la valeur de $Delay_{forwarding}$ est définie sur 0, sinon 1.
$Trust_{packet}$ est la confiance de paquet dans la plage 0 à 1, qui est liée au taux de perte de paquets.

$Trust_{history}$ (l'historique de confiance) est ajoutée au calcul de confiance direct afin de réduire le taux d'erreur de jugement des nœuds normaux.

Une grande partie de l'énergie des nœuds est consommée dans la transmission de données, de sorte que la période de détection de confiance entre les nœuds doit être maximisée dans une plage raisonnable. Ils ont considérés donc le facteur de pondération pour $Trust_{history}$ et $Trust_{packet}$ dans le calcul de la confiance directe. La valeur de poids de l'historique Trust dépend du temps et la formule est représentée par la formule (2.6).

$$w_2 = real_1 * Period_{network} * exp(-real_2 * Period_{network}) \tag{2.6}$$

$Period_{network}$ est l'intervalle de temps entre la dernière mise à jour et maintenant. $real_1$ et $real_2$ sont deux nombres réels définis à l'initialisation. Plus la valeur du $Period_{network}$ est grande, plus la valeur de la confiance direct converge rapidement vers $Trust_{packet}$. Dans la formule, la valeur de $w1 + w2$ est 1.

Confiance totale entre les nœuds :

À ce niveau, le nœud source demande des valeurs de recommandation à ses nœuds adjacents de confiance.
La formule générale de calcul de confiance des recommandations est représentée par la formule (2.7).

$$Trust_{recommandation} = \sum_{i \in set(neighbor)} w_{i(i,j)} * Trust_{(j,k)} \tag{2.7}$$

set (neighbor) est un ensemble de nœuds de confiance du nœud source. $Trust_{(j,k)}$ est la valeur de confiance que le nœud j fait confiance au nœud k.
Cependant, le nœud source affiche différentes valeurs de confiance pour différents nœuds adjacents. Dans ce cas, il devrait y avoir des mécanismes pour réduire correctement l'impact des nœuds à faible performance. Ici, ils ont trié la table de confiance du nœud source par valeurs de confiance de petite à grande. Ensuite ils ont calculés le poids de chaque nœud adjacent par une progression arithmétique, cette formule est représentée par (2.8).

$$w_{i(i,j)} = i / \sum 1^n i = 2 * i / n(n+1) \tag{2.8}$$

i est l'emplacement d'un nœud dans la séquence ordonnée.
n est le nombre de nœuds qui répondent au paquet de confiance de recommandation. $Trust_{recommandation}$ donne au nœud source un avis consultatif, donc la confiance est mise à jour par $Trust_{direct}$

et $Trust_{recommandation}$. La formule est représentée par (2.9) et la valeur de $w3 + w4$ est 1.

$$Trust_{synthesis} = w_3 * Trust_{direct} + w_4 * Trust_{recommandation} \tag{2.9}$$

Résultat et discussion :

Ce mécanisme de confiance peut réduire la consommation d'énergie du réseau, garantir l'état de confiance du réseau et des nœuds périphériques et récupérer les nœuds de jugement erroné dans une plage de retard autorisée. Mais il est vulnérable a quelques attaques comme On-off attack lorsqu'un nœud effectuer des bons et de mauvais services de manière aléatoire, et Opportunistic service attacks qu'elle se fait lorsqu'un nœud malicieux détecte que sa valeur de confiance a diminué, il peut fournir de bons services pour la remontée.

2.2.5 A two-way trust management system for Fog computing

Alemneh et al [39] ont proposé un système de gestion de confiance basé sur la logique subjective bidirectionnelle.

Quand le serveur Fog reçoit une requête de la part de client, il va vérifie la fiabilité de client en calculant la confiance direct depuis les métriques tel que friendship, honesty et ownership.ainsi qu'une confiance indirect en agrégeant la recommendations consulte depuis les serveurs voisins en utilisant un logique subjectif . enfin il va combine les deux valeurs de confiance en fonction de leur poid pour déterminer la valeur finale de confiance.Si ce dernier soit supérieur ou égale à un seuil la connexion est permet sinon il soit refusé.

Dans le cas de permission, le client aussi va assurer la fiabilité de serveur en trouvant sa valeur de confiance en calculant la somme bondrée de la confiance direct et indirect ou la confiance direct est calculé en utilisant la latence, PDR et les informations des relations social, ownership. la confiance indirect soit calculé par l'agrégation des recommandations des voisins. si la valeur de confiance de serveur est acceptable la connexion soit établie. si l'un des deux n'est pas fiable le client cherche un autre serveur.

Résultat et discussion :

Cette solution est la première à introduire le confiance bidirectionnelle dans le Fog.Ils y avaient démontré que la solutions est résilient aux attaques de confiance, précise et converge rapidement c'est-à-dire que le calcule de confiance consomme un nombre de cycle.

2.2.6 Detection of hidden data attacks combined Fog computing and trust evaluation method in sensor-cloud system

dans le but de détecter les attaque cachées dans les données dans un système sensor-cloud Zhang et al [41], en propose d'utiliser un mechanism d'évaluation de confiance hiérarchique ou ce dernier soit divisé en trois partis : une couche de confiance directs dans WSN , une couche de décision préliminaire parmi les dispositif Fog sous-jacent, et une couche d'analyse des données dans le Fog.

La première est aussi composé en trois parties comme l'équation (2.11) présente : le calcul de confiance du paquet($Trust_{packet}$) selon l'équation (2.10) qui représente la valeur de confiance de la preuve général où $packet_{success}$ est le nombre de paquet reçue et $packet_{total}$ est le nombre de paquet envoie , la confiance du taux d'échec de routage ($Trust_{routing}$) est utilisé pour un type de preuves qui peuvent être utilisées pour surveiller la charge du réseau et fournir des preuves de prise de décision pour le chemin de transmission des données , et La valeur de confiance du délai de transmission ($Trust_{forwarding}$) qui est une valeur booléen représente la présence d'un anomalies grave ou non par exemple quand la différence entre la nouvel et l'ancien valeur de confiance direct est grand.

$$Trust^{new}_{packet} = \begin{cases} \frac{Packet_{success}}{Packet_{total}} & \textbf{if } |Trust^{old}_{packet} - \frac{Packet_{success}}{Packet_{total}}| \leq Threshold \\ w * \frac{Packet_{success}}{Packet_{total}} + (1-w) * Trust^{old}_{packet} & \textbf{else } |Trust^{old}_{packet} - \frac{Packet_{success}}{Packet_{total}}| < Threshold \end{cases} \tag{2.10}$$

$$Trust_{direct} = (Trust_{packet} - Trust_{routing}) * Trust_{forwarding} \tag{2.11}$$

Pour réduire la consommation d'énergie les informations sont envoie au Fog sous-jacent quant un exception émerge.mais dans le cas de certain exception comme l'exception de retard de transmission,exception des valeurs différent, l'exception de statuts de réseau. la confiance intégrative est calculé comme l'équation (2.12) .qui se baser sur la liste de confiance des capteurs et la topologie du réseau

$$Trust^{(m,j)}_{integrative} = \sum_{i=1}^{n} weight_{(m,i)} * Trust_{(i,j)} \tag{2.12}$$

où i est le numéro de capteur adjacent confiance du capteur anormal. j,n est le nombre des capteur sélectionné et m est un des dispositif Fog sous-jacent.$trust_{i,j}$ est la valeur de confiance de capteur i pour un capteur j et $weight$ est calculé à base de la fréquence de

communication comme l'equation (2.13)

$$weight_{(m,i)} = \frac{communication^{i}_{frequency}}{\sum_{x=1}^{n} communication^{x}_{frequency}} \tag{2.13}$$

dans cet partie les jugements sont préliminaire et simple. les décisions sont temporaire.

dans la troisième parties L'analyse, le traitement et la décision des données sont les principales tâches de cette couche.en se basant sur la list de confiance, les données de capteur et le statut de réseau ils ont arrive a calcule la confiance global de réseau, la supervision de l'échec des capteurs, récupération en cas d'erreur de jugement, la détection d'attaque de données cachées.

Résultat et discussion :

Dans cet solution il avait introduit la notions de récupération en cas d'un erreur de jugement ainsi que il avoient fait d'un manière à économise la consommation d'énergie.Mais la solution n'est pas exacte dans tous les cas, les attaque des données caches ne peuvent être trouvées que dans certaine mesure.

2.2.7 TACRM : trust access control and resource management mechanism in Fog computing

Dans le but de sécurisé l'accès et géré les ressources Daoud et al [42], ont proposé le modale TACRM base sur l'idée d'avoir un agent de surveillance dans chaque cluster afin de faire l'évaluation et la mise a jours de niveau de confiance des utilisateurs connectes. Ainsi que la mise en place d'un gestionnaire de ressource base sur la priorité pour garantir la QoS.

Le fonctionnement de ce modal est divise en trois étapes :

1. La première étape est le regroupement (clustering) en utilisant un algorithme de machine learning afin de deviser les nœuds en des groupes qui vont contenir un nombre de nœud aléatoire dans chaque groupe où ils vont avoir un nœud manager qui joue le rôle d'un lien entre le client IoT et le nœud de Fog.Dans le but de gérer l'accès des nœuds IoT en calculent le niveau de confiance de requête. Apres le nœud manager va autoriser la requête ou non a bases de son niveau de confiance.Ainsi qu'il joue gère les ressources en ordonnancement les requetés des clients. Avec la surveillance du système.En donnant la responsabilité de calcule de confiance au nœud manager qui sera

attribué selon le comportement de client et peut changer à base de de profile utilisateur. Chaque tache utilisateur est attribué avec un pois qui est donnes par le système pour montre l'importance d'activité donc l'équation de calcule de confiance est (2.14) :

$$Trust = \sum_{i=1}^{n} w_i x_i \tag{2.14}$$

Où x est le vecteur des métriques des utilisateurs et w les poids Et la décision sera (2.15)

$$Ac_Des = \begin{cases} 1 & \textbf{if } \sum_{i=1}^{n} w_i x_i \geq \theta \\ 0 & \textbf{else} \end{cases} \tag{2.15}$$

Avec θ est un seuil prédéfinie.Les métriques identifiant les utilisateurs peuvent être par exemple l'historique d'accès, type de host ...

2. La survivance des processus ou l'agent de survivance va détecte les comportements malicieux des utilisateurs et les reports au Fog nœud de l'infrastructure en utilisent les messages IDMEF.
3. La gestion des ressources qui été responsable pour la gestion d'allocation et de l'ordonnancement par rapport a la priorité déterminer par elle .

Résultat et discussion :

La solution propose est fiable et a un niveau de sécurité très élevé ainsi qu'il garantit les contraint de la latence, la scalabilite.

2.3 Tableau comparatif des solutions

Le tableau 2.3 illustre la comparaison entre les travaux présentés précédemment.

	Résilience	Scalabilité	Consommation énergétique	Exactitude
Solution proposé par Tuva Dybedokken (Trust management in Fog computing)	Élevé	Pas mentionnée	Élevé	Élevé
A Risk-Aware Reputation-Based Trust Management in Wireless Sensor Networks	Élevé	Oui	Moyenne	Élevé
TRFIoT : Trust and Reputation Model for Fog-Based IoT	Moyene	Non étudié	Non étudié	Élevé
A Fog-based Hierarchical Trust Mechanism for Sensor-Cloud Underlying Structure	Faible	Oui	Faible	Moyenne
A two-way trust management system for Fog computing	Élevé	Non étudié	Faible	Élevé
Detection of hidden data attacks combined Fog computing and trust evaluation method in sensor-cloud system	moyenne(que si les informations sont d'un degré de crédibilité grand)	No	Faible	moyenne
TACRM : trust access control and resource ma-nagement mechanism in Fog computing	Non étudié	Élevé	Moyenne	Moyenne

TABLE 2.1 – Comparaison des solutions basées sur la confiance dans le Fog

Nous pouvons aussi cité les travaux de Khattak et al [43] , qu'ont proposé un modèle générique de confiance en fournissant une architecture base sur les composant d'un système de gestion de confiance pour aider les fournisseurs de services de Fog à préserver la confiance des utilisateurs dans l'environnement. Où ils avoient distinguer entre la confiance et la fiabilité (La fiabilité est un aspect qui nécessite une évaluation continue pour permettre aux utilisateurs de choisir de faire confiance à un certain fournisseur de Fog).

Le modèle proposé vise à fournir la base d'un système de gestion de la confiance, qui est chargé de fournir la confiance et la fiabilité du système Fog.

Ils ont discuté des composants de base du calculateur de confiance, qui permettent la gouvernance de la confiance, l'établissement de la confiance, l'évaluation de la confiance, la journalisation des événements et le partage des statistiques.

Et les travaux de Rahman et al [44] où ils ont identifier les configurations de la logique flou qui affecte les valeurs de confiance d'un nœud de Fog, en considérant la distance,la latence, et la fiabilité comme métrique. Ainsi qu'il donne un aperçu des définitions de la confiance et de l'avantage de la logique floue pour l'évaluation de la confiance dans le Fog computing.Les même auteurs ont proposé un framework pour l'evaluation de la confiance base sur les broker pour l'allocation des service dans le Fog mais dans leur solution l'utilisation de broker est issentiel ce que le rendre le point faible de systeme [45].

Synthèse :

De cette étude nous avons remarqué que ces modèles représentent des solutions partielles aux problématiques rencontrées dans la gestion de confiance et non pas une solution globale pour le Fog en général, il ne faut toujours un compromis entre les critères tout dépend de nos besoins.

Dans le prochain chapitre, nous allons concevoir l'architecture du système de confiance, et définir les fonction qui calculent la réputation et le risque des nœuds pour former la valeur total de confiance.

Chapitre 3

Conception de système de confiance a base de réputation et de risque dans Fog computing

Introduction

Le chapitre précédent contient l'étude de quelques solutions proposées dans la littérature concernant les mécanismes de calcul de confiance.

Dans ce chapitre nous allons proposer une combinaison de la solution de confiance basée sur les brokers et les recommendations avec l'évaluation des risque.

3.1 Architecture

La figure 3.3 présente le systèmes en démontrant qu'il a une nature décentralisée, plus qu'il est composé de trois types de composants : utilisateurs, brokers et réputation authority. Dans ce modèle, les utilisateurs peuvent être classés en deux catégories : clients Fog qui sont les nœuds qui génèrent des demandes pour initier des transactions avec d'autres utilisateurs . Les serveurs Fog qui peuvent fournir des service aux clients.

Dans cette architecture, les utilisateurs comptent sur leurs brokers pour collecter les informations de réputation. Tous les brokers gèrent une base de données de réputation, qui recueille la réputation de tous les serveurs ayant effectués des transactions avec ses utilisateurs. Généralement les brokers sont des ressources partageable dans le réseau qui peuvent être utilisés par plusieurs utilisateurs.

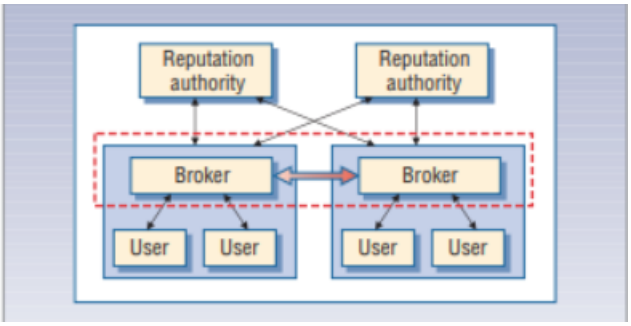

FIGURE 3.1 – La architecture de la solution.

Après toute transaction, chaque utilisateur va envoyer une évaluation (note) de l'autre partie. Après cette étape, la base de données des valeurs de réputation sera remplie en cas ou un utilisateur demande. Cependant, si un broker constate que sa base de données de réputation locale ne convient pas pour fournir des recommandations à ses utilisateurs, il contactera d'autres brokers ou les autorités de réputation.

Les brokers communiquent entre eux pour trouver la valeur de réputation du nœud. Nous supposons que les brokers sont des bons nœuds dans le réseau qui ne peuvent pas fournir des informations fausses ou erronées. Si tous les brokers ne trouvent pas suffisamment d'informations sur un utilisateur, l'autorité de réputation est un dernier recours. L'autorité de réputation maintient une base de données mondiale sur tous les serveurs. Cependant, en raison de la grande échelle, les notes obtenues par les autorités de réputation peuvent être incomplètes ou supérieurs.

3.2 Broker

Le broker est l'élément essentiel dans cette solution où presque tous les calculs sont faites. Il est divisé en deux composants, reputation manager et connection manager. Reputation manager reçoit des requêtes depuis les utilisateurs et les brokers, et il a le pouvoir de demander au connection manager de collecter des données auprès d'autres brokers et des autorités de réputation.

3.2.1 Reputation Manager :

Reputation Manager remplit trois fonctions essentiels. Premièrement, il traite les demandes des utilisateurs clients et des autres brokers. Deuxièmement, il transmet les demandes au connection manager si nécessaire. Troisièmement, il est chargé de sauvegarder les notes des

ces utilisateurs.

Tout enregistrement de réputation contient les champs suivantes :
1- UserID : l'identifiant de l'utilisateur
2- Rating : la valeur de réputation entre 0 et 1
3- Size : le nombre de transactions utilisées pour générer la réputation
4- Timestamp : le temps de la dernière modification.

3.2.2 Connection manager :

Connection manager a le rôle de maintenir la liste des brokers de confiance et aussi les autorités de réputation fiables et jouer le rôle d'interface entre le broker et d'autres brokers.

Tout broker a une valeur de confiance dynamique qui est basée sur le nombre de suggestions précises fournies, en commençant par la valeur $x = 0,5$ et ils vont êtres mises à jour après chaque recommandation reçu et comparer avec le résultat de la transaction en utilisant la formule :

$$X = X + F * (1 - X) \tag{3.1}$$

Si la Recommandation est corrects,sinon

$$X = X * (1 - F) \tag{3.2}$$

3.3 Reputation Authority

L'autorité de réputation est conçu comme une base de données universelle. Elle collecte volontairement des informations de confiance et les stocks depuis le publique pour tous les utilisateurs. Toutefois, la base de données n'est pas mise à jour fréquemment, les informations peuvent donc être erronées ou obsolètes.

3.4 Fonctionnement du protocole

Lorsqu'un utilisateur a besoin des renseignements de confiance d'un autre utilisateur, il communique d'abord avec le broker. Le gestionnaire de réputation(reputation manager) du broker traite les demandes des utilisateurs. S'il ne peut pas traiter une demande, il transmet la demande au gestionnaire de connexion(connection manager) du broker qui va envoyé une

requête aux autres brokers. Si aucun broker ne peut fournir les informations nécessaires, il enverra la demande à l'autorité de réputation.

La note se trouve dans la base de données local ou depuis les autres brokers et combiné avec le risque afin d'être envoyé au client qui aura le choix de continuer ou non la transaction.

Après chaque transaction, l'utilisateur soumet sa note ou son évaluation de l'autre utilisateur au reputation manager. Reputation manager collecte les notes et calcule une valeur de confiance pour l'utilisateur.

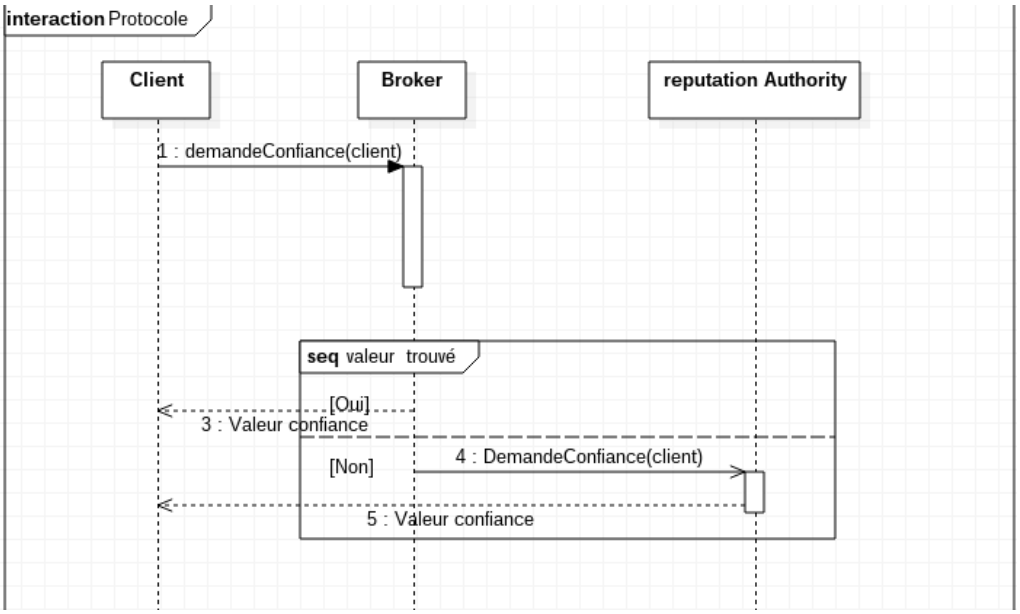

FIGURE 3.2 – Diagramme de séquence du protocole.

3.5 Dimension de confiance

Dans notre solution, nous allons définir les dimensions de confiance comme suit :

Trust composition : pour la composition de confiance, les composants de QoS (Qualité de service).

Trust propagation : comme nous l'avons déjà expliqué, cette partie décrit comment les valeurs de confiance sont stockées dans le réseau. Pour cela, l'approche de la propagation distribué est utilisée et chaque objet est responsable du calcul et du stockage des valeurs de confiance des autres objets du réseau. il stocke la valeur des noeuds avec lesquels il a déjà communiqué.

Trust update : la mise à jour des valeurs de confiance se faite suite à un événement.

Trust formation : la formation de confiance se faite en combinant plusieurs propriétés pour le calcul de la confiance.

Trust aggregating : l'agrégation de confiance se faite à la base des recommandations des autres objets.

3.6 Processus de calcul de confiance

Notion	Description
Trust	Confiance
R_e	Réputation
R_i	Risque
W_{Re}	Poid de réputation
W_{Ri}	Poid de risque
h(x)	Qualite de service
G	Bien (Good)
L	Grade bas (low grade)
N	Pas de reponse (no response)
B	Réponse fausse ou malveillance (Byzantine Behavoir)

TABLE 3.1 – La liste des abréviations utilisées

Dans notre solution, nous avons utilisé une combinaison entre la réputation et le risque, en utilisant une formule qui renvoie une valeur réelle entre 0, qui signifie un manque de confiance total, et 1, qui signifie une confiance totale.

$$Trust = \alpha * R_e + (1 - \alpha) * R_i, 0 \leq \alpha \leq 1 \tag{3.3}$$

Dans cette équation, nous notons l'apparition des poids de la réputation qui est égal a $W_{R_e} = \alpha$ et de risque égal à $W_{R_i} = 1 - \alpha$. En fonction de la valeur de α , nous pouvons paramétrer la solution et utiliser la notion la plus adéquate dans le réseau, par exemple si nous sommes dans un réseau confiant où on peut faire confiance aux recommandeurs, la valeur de α va être très grande pour minimiser l'effet de la valeur de risque dans le calcule de la confiance totale. Sinon un α petit pour mettre une grande importance au risque dans le calcul de confiance [23].

3.6.1 Calcul de la réputation

La valeur de réputation peut être définit comme le point de vue de l'observateur sur l'observé en se basant sur l'historique du comportement passé de l'observé. La réputation

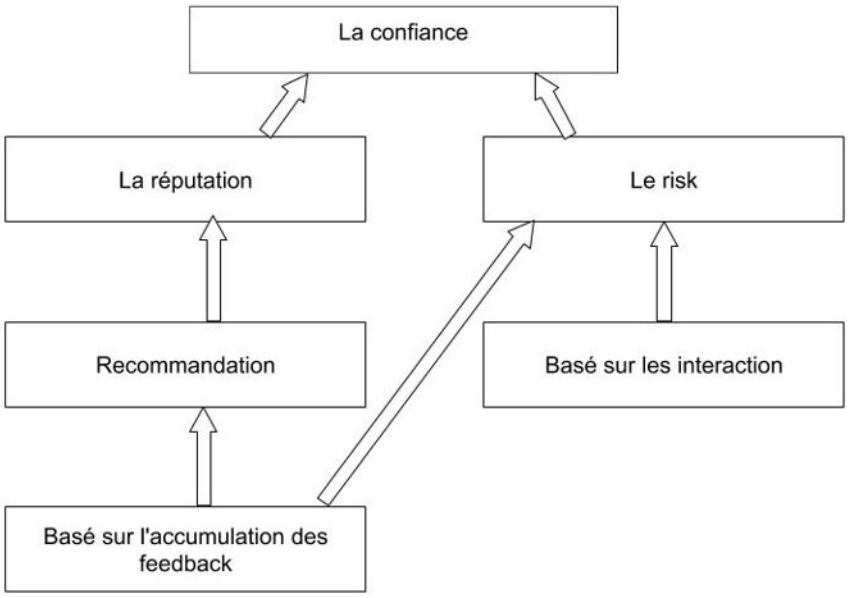

FIGURE 3.3 – Les composant de la solution.

est calculée en cumulant les évaluations des utilisateurs après chaque transaction selon la formule suivante [24] :

$$T_{new} = e^{-\beta.\Delta t}\frac{N}{N+1}T_{old} + \left(1 - e^{-\beta.\Delta t}\frac{N}{N+1}\right)r \tag{3.4}$$

où T_{new} est la nouvelle valeur de réputation d'utilisateur, T_{old} est l'ancienne valeur de réputation, N est le nombre de transactions actuels , r la note de l'utilisateur ,t est la différence entre le temps de note et Told , $e^{-.t}$ est un facteur de réduction de T_{old}

Si la réputation n'est pas trouvée dans la base de données de réputation manager locale, la demande est envoyée au connection manager, et le connection manager transmettra la demande aux m premiers brokers dont le niveau de confiance est supérieur au seuil T. après qu'il reçoit les recommandations des autres broker, il va les calculer avec la formule suivante :

$$R = \sum \frac{X_i * N_i * R_i * F_i}{\sum X_i * N_i * F_i} \tag{3.5}$$

où X_i est la valeur de confiance de broker i, N_i est le nombre de fois où connection manager a demandé des recommandations au broker, R_i est la recommandation du broker i, et F_i est un facteur différentiel de temps. Si la différence de temps par rapport à la dernière recommandation est inférieure à une valeur seuil alors $F_i = 1$; sinon $F_i = e^{-.t}$.

Connection manager envoi la recommandation calculée R au réputation manager.

3.6.2 Calcul de risque

La réputation n'est pas assez sensible pour percevoir les nœuds qui s'endommagent soudainement car il lui faut du temps pour diminuer le score cumulé. L'évaluation des risques peut aider à résoudre ce problème.

Chaque nœud a ses propres vues personnalisées sur le réseau, donc les recommandations ne sont pas fiables même si elles proviennent des nœuds fiables. Par conséquent, pour rendre la confiance plus précise, nous utilisons uniquement les informations dérivées de l'interaction pour calculer la valeur du risque.

Pour calculer la valeur de risque on va classifier les services en quatre catégories présentées comme suit $Q = \{G, L, N, B\}$ où les services L, N, B sont mauvaises. Tel que si le service est bon comme prévue sa classe est G mais s'il y a une certaine dégradation sa classe devient L. Si pas de réponse reçu elle devient N, sinon si la réponse est fausse ou malveillante sa classe sera B. Nous utilisons une fonction de mappage $h(x)$ depuis Q(les classes) aux scores qui sont définis auparavant par l'administrateur ($S1, S2, S3, S4$ en notant que $S2, S3, S4$ ont des valeurs négatives)[47].

$$h(x) = \begin{cases} S_1 & x = G, S_1 > 0 \\ S_2 & x = L, S_2 < 0 \text{ and } |S_2| > S_1 \\ S_3 & x = N, S_3 < S_2 \\ S_4 & x = B, S_4 < S_3 \end{cases} \tag{3.6}$$

$$R = \frac{\sum_{i=B,N,L}(N_i * h(i))}{h(B) * \sum_{j=G,B,N,L}(N_j)} \tag{3.7}$$

N_i : le nombre de services de qualité i.

$h(i)$: le score pour la coopération avec la qualité de service i

.

Conclusion

Dans ce chapitre, nous avons cité les formules mathématiques qui nous ont permis le calcul de la confiance a partir des différentes valeurs de risque et de recommendations des noeuds du réseau pour conclure avec la valeur totale de confiance.

Dans le prochain chapitre, nous allons faire une implémentation de la solution proposée, ensuite conclure avec des résultats sous formes des graphes.

Chapitre 4

L'implémentions

Introduction

Dans ce chapitre, nous allons parler de l'environnement de développement et le langage utilisés pour la mise en œuvre de notre solution, les raisons pour laquelle nous les a choisit, aussi des parties essentiels des algorithmes sous forme de code java, et les résultats de la simulation sous forme de graphes.

4.1 Déploiement de la solution

Il existe plusieurs simulateurs pour le Fog computing, on peut citer quelques uns : iFogSim pour mesurer l'impact des techniques de gestion des ressources. Discrete Event System Specification (DEVS) qui est un outil basé sur l'évaluation de l'impact du déploiement du Fogging.EmuFog qui est un cadre d'émulation extensible pour les environnements de Fog computing sans fonction de mobilité. FogTorch pour les déploiements prenant en charge la QoS d'applications IoT multi-composants aux infrastructures de Fog. dans notre solution, nous avons utilisé au départ le simulateur omnet++ qui est un simulateur pour les réseaux internet, mais nous avons rencontrés beaucoup de problèmes durant l'utilisation de ce simulateur, parmi eux, le package Fognetsim qui implémente le Fog est obsolète et ne fonctionne pas dans la nouvelle version de omnet++ et de son frameword inet, même quand nous avons installé la version antérieur, les problèmes restent toujours a cause des versions obsolètes. Nous avons éssayés d'utiliser YAFS (Yet Another Fog Simulator) qui est basé sur python, mais il est obsolète aussi parcequ'il fonctionne que sur python 2.0.

Vu q'il n'y a pas d'outil de simulation à part entière pour le nouveau paradigme informatique donc nous avons développé notre propre modèle de simulation basé sur java pour

le scénario, ce modèle contient des classes comme Broker, Fog, ReputationAuthority comme composants les plus importants, plus de détails dans la partie suivante.

4.2 Configurations matérielle et logiciel utilisés

Les tableaux 4.2 et 4.2 la configuration qui a été utilisé pour la mise on œuvre de la solution :

Système d'exploitation	Windows 10
RAM	8Go
CPU	2.7GHz 4 core
Langage de programmation	Java 8
IDE	Eclipse

TABLE 4.1 – Les caractéristiques de l'environnement de simulation.

Temps de simulation	30 min
Nombre des requêtes	1340
Threshold de confiance	0.4
$\alpha_1, \alpha_2, \beta_1, \beta_2$	0.2, 0.8 , 0.2, 0.8
β	2.7*10-7

TABLE 4.2 – Les paramètres utilise lors de simulation.

4.3 Pourquoi java a la place d'omnet++

Vu les limites de java, nous ne pouvions pas calculé la latence, le nombre de paquets rejetés... donc notre simulation aurait pu être mieux avec omnet++ qui contient tout ces fonctionnalités réseau.

Malheureusement, nous avons perdu beaucoup de temps environ 3 mois avec son installation de utilisation, vu que le package fognetsim existe dans l'ancienne version de omnet (4.6) qui fonctionne sous ubuntu 16.04, après l'installation, nous avons vécu trop de soucis dans l'utilisation de cette bibliothèque, et le manque de documentation nous a rendu les choses flou, donc nous étions obligés d'utilisé java a la fin.

4.4 Fonctionnement du système

nous avons notre système constitué des classes qui représentent les composants de l'architecture, classe pour le broker, classe client, serveur, reputation autority, et chaque classe contient des méthodes a l'intérieur pour le calcul et la propagation de confiance, pour le calcul on utilise les fonctions mathématiques de calcul de risque et de la réputation qui sont précisés dans la solution, et pour la simulation, nous avons crée des objets de ces classes et faire les interactions entre ces objets sans prendre en compte les notions de réseau tel que le temps perdu, le nombre des paquets envoyés et reçus ...etc

Donc nous avons les clients qui font des requêtes aléatoires, après la transmission des valeurs de confiance entre les nœuds se faite avec des fonctions.

4.5 Simulation et résultats

4.5.1 L'interface d'entrée des données

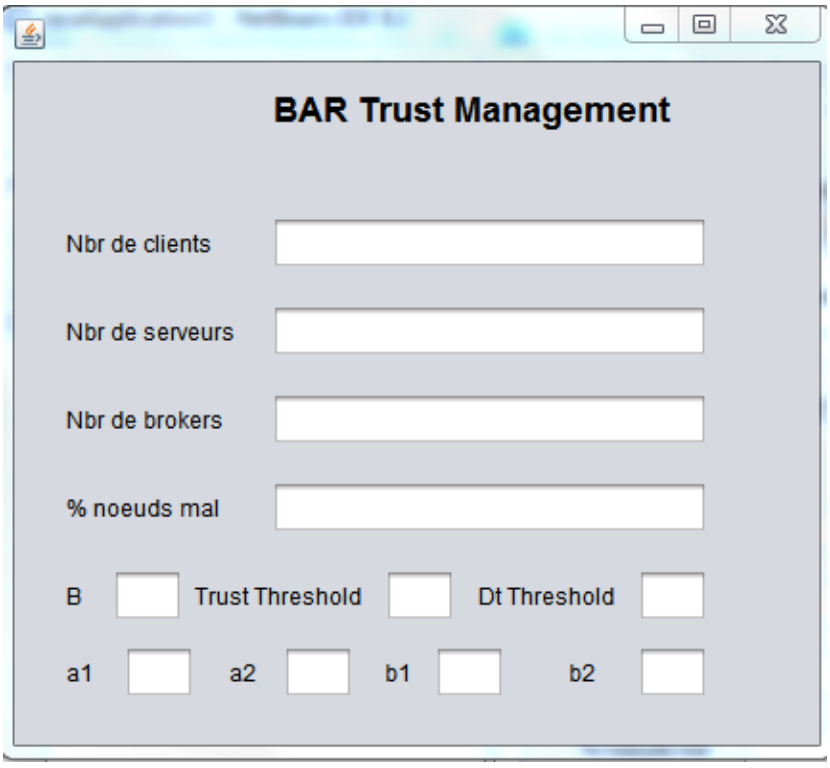

FIGURE 4.1 – L'interface d'entrée des paramètres.

Cette image montre l'interface d'entrée des paramètres initiales pour le démarrage de la simulation de solution. Les paramètres d'entrée ont été spécifiés dans le tableau des configurations matériels et logiciels.

nous avons le nbr de clients qui représente le nombre total des clients dans le système Nbr de serveurs représente le nombre des fournisseurs de services dans le système Nbre de brokers c'est le nombre nécessaire des brokers pour le calcul de confiance % noeuds mal représente le pourcentage de chaque nœud qu'il soit malveillant ou pas Trust Threshold c'est le seuil dont chaque nœud qui a une valeur de confiance supérieur a ce seuil nous allons le considéré comme nœud confiant les autres entrées B,a1,a2,b1,b2 Dt Threshold sont des coefficients qui rentrent dans le calcul de confiance pour donner une importance a des propriétés par rapport aux autres, par exemple donner une grande importance a la valeur du risque dans le calcul

4.5.2 L'interface des résultats de simulation

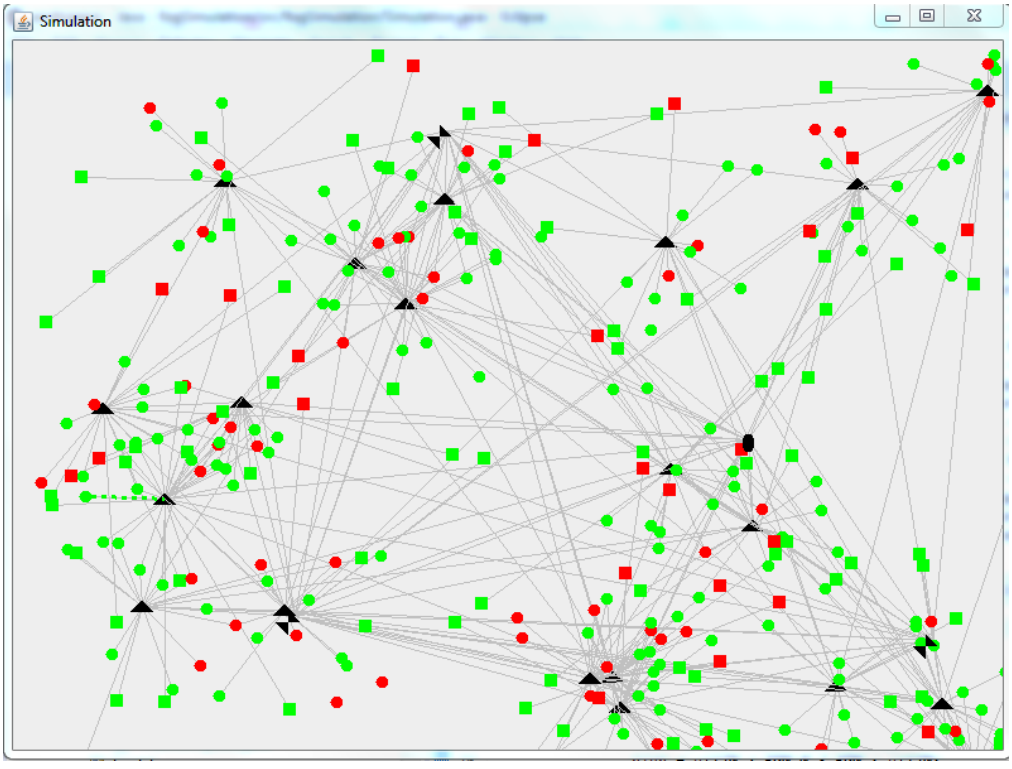

FIGURE 4.2 – L'interface de l'architecture de réseau.

— carré : fog serveur
— triangle : broker
— ellipse : cloud

— triangles collés : autorité de réputation

cette photo représente les résultats de simulation après le calcul de la valeur de confiance de chaque nœud, les liens gris entre les nœuds représentent les transferts de valeur de confiance, donc après le calcul de confiance nous avons comme résultat les nœuds fiables qui sont en vert, et les nœuds malveillants en rouge.

4.5.3 Graphes des résultats

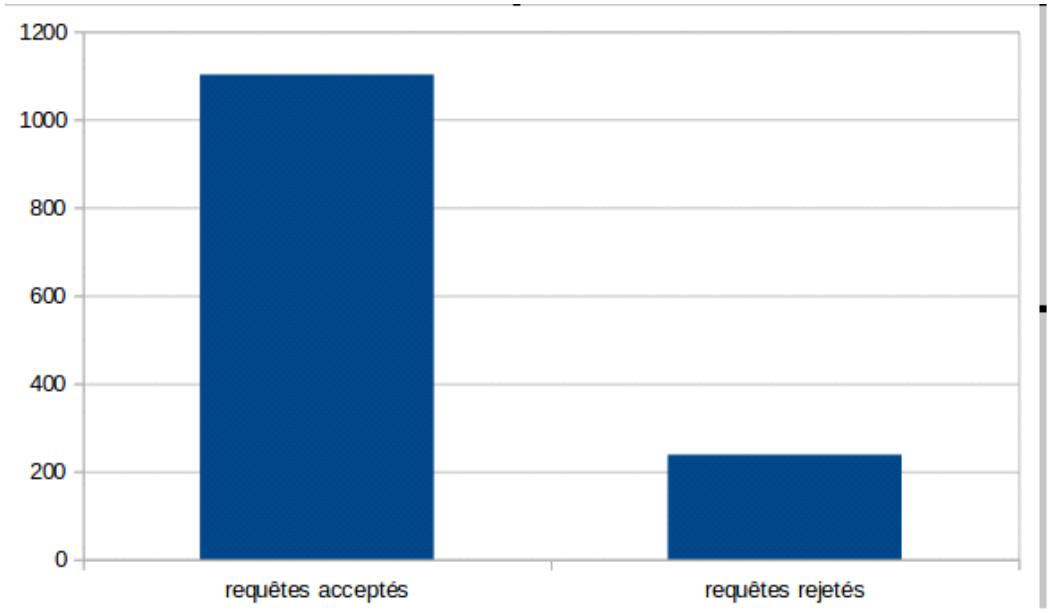

FIGURE 4.3 – Diagramme de comparaison entre les requêtes acceptés et rejetés.

ce diagramme représente le nombre des requêtes acceptés et rejetés dans le système, nous avons en total 1340 requêtes, qui ont été divisé entre 1102 requêtes acceptés, et 238 requêtes rejetés par le système après la simulation.

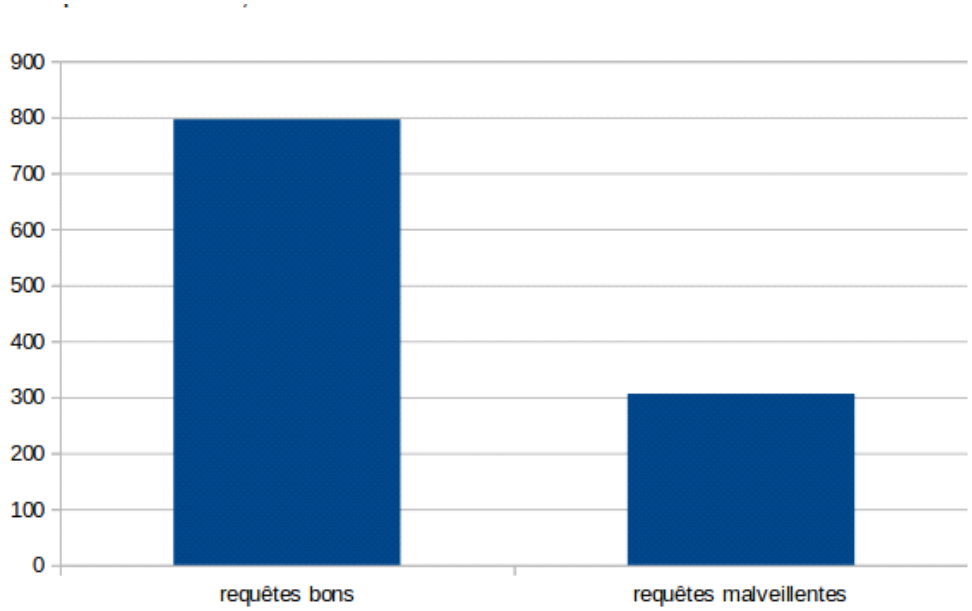

FIGURE 4.4 – Diagramme de comparaison entre les bons requêtes et les mauvaises

ce diagramme représente le nombres des requêtes bons et malveillante, donc après simulation, le système a considéré que 306 requêtes sont malveillantes c-a-d qui ont une valeur de confiance inférieur a le seuil, et 796 requêtes bons (c-a-d nœuds fiables qui ont une valeur de confiance supérieur aux seuil), parmi les 1102 requêtes acceptés avant.

Conclusion

Nous pouvons donc conclure a partir de notre simulation que la confiance dépend de sa valeur initiale et du threshold, ils ont une Relation de corrélation directe, si ces 2 propriétés augmentent, forcément on va avoir une valeur de confiance meilleur et donc une meilleur sécurité.

Conclusion générale

Un nombre important des chercheurs travaillent toujours sur ce nouveau paradigme, ils cherchent a développé une solution optimale pour l'appliqué dans ce réseau de fog, mais chaque solution a des avantages et des inconvénients, nous avons essayés de trouver une solution qui couvre la majorité des propriétés nécessaires du fog dans le calcul de confiance.

Notre solution consiste a proposer un modèle de confiance qui se base sur le calcul de la réputation et des recommandations en utilisant une architecture composé de broker, autorité de réputation, et gestionnaires de connexion. Nous avons valider la solution avec une simulation en utilisant Java vu qu'il n'y a pas d'outil de simulation à part entière pour ce nouveau paradigme informatique.

Pour conclure avec des résultats et des améliorations de ce modèle qui peuvent se faire dans le futur comme le feedback, notre solution fait confiance a feedback reçu des clients, ce qui n'est pas très bon et met notre système vulnérable a des attaques de fog, même pour les transferts de la valeur de confiance entre les clients et le broker ralentissent les transactions et augmentent la latence, donc il nous faut un nombre minimal de transferts de valeur de confiance dans le calcul pour que notre solution soit efficace sinon nous allons perdre une des caractéristiques du fog computing qui est a la base conçu pour réduire la latence, donc nous allons essayé dans le futur d'amélioré cette solution.

Bibliographie

[1] Cisco Systems, “Fog Computing and the Internet of Things :Extend the Cloud to Where the Things Are,” www.Cisco.Com,p. 3, 2020. [Online]. Available : https://www.cisco.com/c/dam/en_us/solutions/trends/iot/docs/computing-overview.pdf

[2] F. Bonomi, R. Milito, J. Zhu, and S. Addepalli, “Fog computing and its role in the internet of things,” in Proceedings of the first edition of the MCC workshop on Mobile cloud computing. ACM, 2012, pp. 13–16.

[3] :L. M. Vaquero and L. Rodero-Merino, “Finding your way in the fog : Towards a comprehensive definition of fog computing,” ACM SIGCOMM Computer Communication Review, vol. 44, no. 5, pp. 27–32, 2014.

[4] IBM, “What is fog computing ?” Sep 2020.[Online].Available : https://www.ibm.com/blogs/cloud-computing/2014/08/fog-computing/

[5] “Definition of Fog computing,” accessed on 12 May, 2020. [Online]. Available : https://www.iiconsortium.org/pdf/OpenFog_Reference_Architecture_2_09_17.pdf

[6] Naha, Ranesh & Garg, Saurabh & Georgakopoulos, Dimitrios & Jayaraman, Prem Prakash & Gao, Longxiang & Xiang, Yong & Ranjan, R.. (2018). Fog Computing : Survey of Trends, Architectures, Requirements, and Research Directions.

[7] Alemneh, Esubalew & Senouci, Sidi-Mohammed & Brunet, Philippe & Tegegne, Tesfa. (2019). A two-way trust management system for fog computing. Future Generation Computer Systems. 106. 10.1016/j.future.2019.12.045.

[8] Accessed on 12 May, 2020. [Online]. Available : https://www.statista.com/statistics/471264/iot-number-of-connected-devices-worldwide/

[9] M. Aazam and E. Huh, "Fog Computing : The Cloud-IoT/IoE Middleware Paradigm," in IEEE Potentials, vol. 35, no. 3, pp. 40-44, May-June 2016, doi : 10.1109/MPOT.2015.2456213.

[10] Sarkar, S., Chatterjee, S., Misra, S. (2018). Assessment of the Suitability of Fog Computing in the Context of Internet of Thing. IEEE Transactions on Cloud Computing, 6 (1), 46-59.

[11] Proceedings of the 2014 Federated Conference on Computer Science and Information Systems, M. Ganzha, L. Maciaszek, M. Paprzycki (eds). ACSIS, Vol. 2, pages 1–8 (2014)

[12] Tang, Bo & Chen, Zhen & Hefferman, Gerald & Wei, Tao & He, Haibo & Yang, Qing. (2015). A hierarchical distributed fog computing architecture for big data analysis in smart cities. 10.1145/2818869.2818898.

[13] M. Ahmad, M. B. Amin, S. Hussain, B. H. Kang, T. Cheong, S. Lee. Health fog : a novel framework for health and wellness applications. Journal of Supercomputing. 2016, 72 (10), 3677-3695, doi : doi :10.1007/s11227-016-1634-x.

[14] Neware, R., & Shrawankar, U. (2020). Fog Computing Architecture, Applications and Security Issues. International Journal of Fog Computing (IJFC), 3(1), 75-105. doi :10.4018/IJFC.2020010105

[15] L. Mui, M. Mohtashemi, A. Halberstadt, A computational model of trust and reputation, in : Proceedings of the 35th International Conference on System Science, 2002, pp. 280–287.

[16] T. Grandison, M. Sloman, A survey of trust in internet applications, IEEE Commun. Surv. Tutorials 4 (4) (2000) 2–16.

[17] D. Olmedilla, O. Rana, B. Matthews, W. Nejdl, Security and trust issues in semantic grids, in : Proceedings of the Dagsthul Seminar, Semantic Grid : The Convergence of Technologies, vol. 05271, 2005.

[18] R. Neisse, M. Wegdam, and M. van Sinderen. Trust management support for context-aware service platforms. In User-Centric Networking, pages 75-106. Springer, 2014.

[19] Jia Guo, Ing Ray Chen, and Jeffrey J P Tsai. A survey of trust computation models for service management in internet of things systems. Computer Communications, 97 :1–14, 2017.

[20] A. Jøsang, R. Ismail, and C. Boyd, "A survey of trust and reputation systems for online service provision," Decision Support Systems, vo. 43, no. 2, 2007, pp. 618-644.

[21] Jin-Hee Cho, Ananthram Swami, and Ing-Ray Chen. A survey on trust management for mobile ad hoc networks. IEEE Communications Surveys & Tutorials Tutorials, 13(4) :562–583, 2011.

[22] Proposition d'un modèle de confiance pour l'Internet des Objets. Mémoire soutenu le 21/06/2015 par : AIT MOUHOUB Younes, BOUCHEBBAH Fatah

[23] Liang, Zhengqiang & Shi, Weisong. (2005). PET : A PErsonalized Trust Model with Reputation and Risk Evaluation for P2P Resource Sharing.. 10.1109/HICSS.2005.493.

[24] K.-J. Lin et al., "A Reputation and Trust Management Broker Framework for Web Applications," Proc. IEEE Int'l Conf. e-Technology, e-Commerce, and e-Services, IEEE CS Press, 2005, pp. 262-269

[25] Dastjerdi, A., Gupta, H., Calheiros, R., Ghosh, S., Buyya, R. : Chapter 4 - fog computing : principles, architectures, and applications. In Buyya, R., Dastjerdi, A.V., eds. : Internet of Things : Principles and Paradigms. Morgan Kaufmann (2016) 61 – 75

[26] Branka Mikavica, Aleksandra Kostic-Ljubisavljevic : FOG COMPUTING IN LOGISTICS SYSTEMS, Conference Paper · May 2019

[27] Taamourt feriel évaluation des performances d'une flotte de drones dans un environnement Fog computing

[28] Shubha Brata Nath, Harshit Gupta, Sandip Chakraborty, Soumya K. Ghosh. A Survey of Fog Computing and Communication : Current Researches and Future Directions

[29] A survey of trust in computer science and the Semantic Web Donovan Artz, Yolanda Gil

[30] Proposition d'un modèle de confiance pour l'Internet des Objets. Mémoire soutenu le 21/06/2015 par : M r AIT MOUHOUB Younes M r BOUCHEBBAH Fatah

[31] SECURITY AND PRIVACY SOLUTIONS FOR FOG COMPUTING : EMERGING TRENDS, ISSUES, AND CHALLENGES

[32] Neware, R. (2019). Fog Computing Architecture, Applications and Security Issues : A Survey.

[33] Cisco. (2015). Fog Computing and the Internet of Things : Extend the Cloud to Where the Things Are [White paper]. Récupéré le 03 janvier 2020 de Cisco : `https://www.cisco.com/c/dam/en_us/solutions/trends/iot/docs/computing-overview.pdf`

[34] Dybedokken, T.S. (2017). Trust Management in Fog Computing.

[35] Jia Guo, Ing Ray Chen, and Jeffrey J P Tsai. (2016) . A survey of trust computation models for service management in internet of things systems. Computer Communications

[36] Huaizhi Li and Mukesh Singhal. Trust management in distributed systems. Computer, 40(2) :45–53, 2007.

[37] Accessed on 5 spetembre, 2020. [Online]. Available : http://www.dicofr.com/cgi-bin/n.pl/dicofr/definition/20190724073521/

[38] Vers un modèle de confiance pour l'Internet des Objets. Mémoire soutenu en 2016 par HADDAD Syphax.

[39] E. Alemneh, S.-M. Senouci, P. Brunet et al., A two-way trust management system for fog computing, Future Generation Computer Systems (2019), doi : https://doi.org/10.1016/j.future.2019.12.045

[40] Hussain, Yasir & Huang, Zhiqiu. (2018). TRFIoT : Trust and Reputation Model for Fog-based IoT : 4th International Conference, ICCCS 2018, Haikou, China, June 8-10, 2018, Revised Selected Papers, Part VI. $10.1007/978 - 3 - 030 - 00021 - 9_18$.

[41] Zhang, G, Wang, T, Wang, G, Liu, A, Jia, W. Detection of hidden data attacks combined fog computing and trust evaluation method in sensor-cloud system. Concurrency Computat Pract Exper. 2018 ;e5109. https://doi.org/10.1002/cpe.5109.

[42] Daoud, W.B., Obaidat, M.S., Meddeb-Makhlouf, A. et al. TACRM : trust access control and resource management mechanism in fog computing. Hum. Cent. Comput. Inf. Sci. 9, 28 (2019). https://doi.org/10.1186/s13673-019-0188-3.

[43] Khattak H.A., Imran M., Abbas A., Khan S.U. (2019) Maintaining Fog Trust Through Continuous Assessment. In : Xia Y., Zhang LJ. (eds) Services – SERVICES 2019. SERVICES 2019. Lecture Notes in Computer Science, vol 11517. Springer, Cham

[44] F. H. Rahman, T. W. Au, S. Newaz, W. S. Suhaili, Trustworthiness in fog :A fuzzy approach, in : Proceedings of the 2017 VI International Conference on Network, Communication and Computing, ACM, 2017, pp. 207–211.

[45] F. H. Rahman, T.-W. Au, S. S. Newaz, W. S. Suhaili, G. M. Lee, Find my trustworthy fogs : A fuzzy-based trust evaluation framework, Future Generation Computer Systems.

[46] G. Zhang, T. Wang, M. Z. A. Bhuiyan and G. Wang, "A Fog-Based Hierarchical Trust Mechanism for Sensor-Cloud Underlying Structure," 2017 IEEE International Symposium on Parallel and Distributed Processing with Applications and 2017 IEEE International Conference on Ubiquitous Computing and Communications (ISPA/IUCC), Guangzhou, 2017, pp. 481-485.

[47] Labraoui, N., Gueroui, M. & Sekhri, L. A Risk-Aware Reputation-Based Trust Management in Wireless Sensor Networks. Wireless Pers Commun 87, 1037–1055 (2016). https://doi.org/10.1007/s11277-015-2636-3.

[48] AAse Dragland. Big data, for better or worse : 90% of world's data generated over last two years – sciencedaily. https://www.sciencedaily.com/releases/2013/05/130522085217.htm, 05 2013. (Accessed on 5 septembre 2020).

[49] Accessed on 5 spetembre, 2020. [Online]. Available : https://www.vinci-energies.com/le-fog-computing-la-solution-pour-gerer-des-milliards-dobjets-connectes/

[50] Dubey, Harishchandra & Constant, Nicholas. (2015). Fog Data : Enhancing Telehealth Big Data Through Fog Computing. ACM BigData 2015 The Fifth ASE International Conference on Big Data. 10.1145/2818869.2818889.

[51] Yi, Shanhe Hao, Zijiang & Qin, Zhengrui Li, Qun. (2015). Fog Computing : Platform and Applications. 10.1109/HotWeb.2015.22.

[52] A.V. Dastjerdi, H. Gupta, R.N. Calheiros, S.K. Ghosh, R. Buyya, Chapter 4 - Fog Computing : principles, architectures, and applications, Editor(s) : Rajkumar Buyya, Amir Vahid Dastjerdi, Internet of Things, Morgan Kaufmann, 2016, Pages 61-75, ISBN 9780128053959, https://doi.org/10.1016/B978-0-12-805395-9.00004-6.

[53] Atlam, Hany & Walters, Robert & Wills, Gary. (2018). Fog Computing and the Internet of Things : A Review. Big Data and Cognitive Computing. 2. 10.3390/bdcc2020010.