

الجمهورية الجزائرية الديمقراطية الشعبية  
وزارة التعليم العالي والبحث العلمي



جامعة سعيدة د. مولاي الطاهر  
كلية التكنولوجيا  
قسم: الإعلام الآلي

## Mémoire de Master

Spécialité : Réseaux Informatiques Et Systèmes Réparties

### Thème

Stratégie de placement adaptative des services pour les architectures fog computing

Présenté par :

Ayoub Nihad Manel

Dalim Imen

Dirigé par :

Dr.Said Limam



Année universitaire 2022-2023

# Remerciements

Nous tenons à exprimer nos sincères remerciements à toutes les personnes qui ont contribué de près ou de loin à la réalisation de cette mission.

Nous tenons tout d'abord à remercier notre encadrant Said LIMAM pour ses conseils avisés, sa disponibilité et sa patience tout au long de cette aventure. Ses précieux conseils ont rendu ce projet de recherche possible.

Nous remercions également les membres du jury pour l'intérêt qu'ils portent à ce travail et pour leurs commentaires constructifs.

Nos remerciements vont également à tous les professeurs d'université de Saïda Dr Tahar MOULAY, pour la qualité de leur enseignement, leur soutien et leurs encouragements tout au long de mes études.

Nous tenons à exprimer notre gratitude à nos amis et à la famille AYOUB et la famille DALIM, qui nous ont encouragés et soutenus tout au long de ces années d'études et qui nous ont soutenus dans les moments difficiles.

Nous tenons également à remercier les personnes qui ont participé à cette étude en répondant aux questionnaires et en acceptant les entretiens. Leurs réponses ont été très utiles pour faire ce travail.

Enfin, nous tenons à remercier toutes les personnes qui ont contribué à la réalisation de ce message, de près ou de loin.

Merci à tous pour votre soutien, votre aide et votre confiance tout au long de ce voyage.

" Médite fréquemment la rapidité avec laquelle passent et se dissipent les êtres et les événements. La substance est, en effet, comme un fleuve, en perpétuel écoulement ; les forces sont soumises à de continuelles transformations, et les causes formelles à des milliers de modifications. Presque rien n'est stable, et voici, tout près, le gouffre infini du passé et de l'avenir, où tout s'évanouit. Comment ne serait-il pas fou, celui qui s'enfle d'orgueil parmi ce tourbillon, se tourmente ou se plaint, comme si quelque chose, pendant quelque temps et même longtemps, pouvait le troubler ? "

*-Marc Aurèle-*

# Table des matières

<b>1</b>	<b><i>Internet des objets(IOT)</i></b>	<b>10</b>
1.1	Introduction : . . . . .	11
1.2	Définition : . . . . .	11
1.3	Les principes de base de l'IOT : . . . . .	12
1.4	Le fonctionnement de l'IOT . . . . .	12
1.5	Comment les appareils IOT se connectent-ils les uns aux autres? . . . . .	13
1.6	Les avantages de l'IOT : . . . . .	13
1.7	Les inconvénients de l'IOT : . . . . .	14
1.8	Conclusion : . . . . .	15
<b>2</b>	<b><i>Cloud et Fog computing</i></b>	<b>16</b>
2.1	Introduction : . . . . .	17
2.2	Cloud computing : . . . . .	17
2.2.1	Définition : . . . . .	17
2.2.2	Les types de cloud computing : . . . . .	17
2.2.3	Les services du cloud computing : . . . . .	18
2.2.4	L'utilisation du cloud computing : . . . . .	19
2.2.5	Les avantages du cloud computing : . . . . .	20
2.2.6	Les inconvénients du cloud computing : . . . . .	22
2.3	Fog computing . . . . .	22
2.3.1	Définition : . . . . .	22
2.3.2	Différenciation avec le cloud-computing : . . . . .	23
2.3.3	Différenciation avec le edge-computing : . . . . .	23
2.3.4	La communication dans une infrastructure fog computing : . . . . .	24
2.3.5	Le fonctionnement du fog computing : . . . . .	25
2.3.6	Les caractéristiques du fog computing : . . . . .	26
2.3.7	Les avantages et les inconvénients d'une architecture fog computing : . . . . .	27
2.4	Conclusion : . . . . .	28
<b>3</b>	<b><i>L'approche propose</i></b>	<b>29</b>
3.1	Introduction : . . . . .	30
3.2	L'architecture proposée : . . . . .	30
3.2.1	Couche périphérique (edge layer) : . . . . .	30
3.2.2	Couche fog (fog layer) : . . . . .	30

---

3.2.3	Couche cloud (cloud layer) : . . . . .	31
3.3	La stratégie de placement des services proposée : . . . . .	31
3.3.1	La phase de traitement des requêtes : . . . . .	31
3.3.2	La phase de réplication des services : . . . . .	33
3.3.3	La phase de suppression : . . . . .	35
3.4	Conclusion : . . . . .	36
<b>4</b>	<b><i>Implémentation et résultats de simulations</i></b>	<b>37</b>
4.1	Introduction : . . . . .	38
4.2	Environnement de Développement : . . . . .	38
4.2.1	Le langage de programmation Java : . . . . .	38
4.2.2	Eclipse IDE : . . . . .	40
4.2.3	IFogSim : . . . . .	41
4.3	L'implémentation : . . . . .	44
4.3.1	Le modèle d'application : . . . . .	44
4.3.2	Caractéristiques détaillées des noeuds de l'infrastructure Fog : . . . . .	45
4.3.3	L'interface principale : . . . . .	46
4.3.4	Résultats : . . . . .	48
4.4	Conclusion : . . . . .	50

# Table des figures

1.1	L'internet des objets [6] . . . . .	11
1.2	La connectivité des objets [6] . . . . .	13
2.1	Les types de cloud computing [28] . . . . .	18
2.2	Les services du cloud computing [5] . . . . .	19
2.3	Edge et fog computing [3] . . . . .	24
2.4	fog computing [2] . . . . .	25
3.1	L'architecture du fog computing [4] . . . . .	30
3.2	L'organigramme de traitement des requêtes . . . . .	31
3.3	L'organigramme de réplication des services . . . . .	34
3.4	L'organigramme de suppression des services inutiles . . . . .	36
4.1	Les classes fondamentales d'iFogSim [1] . . . . .	42
4.2	Les classes de topologie physique iFogSim [1] . . . . .	43
4.3	Modèle d'application . . . . .	44
4.4	L'interface Fog Device . . . . .	47
4.5	L'interface Service . . . . .	47
4.6	L'interface Topology . . . . .	48
4.7	Graphique linéaire de consommation d'énergie moyenne . . . . .	49
4.8	Graphique linéaire de latence moyenne . . . . .	49
4.9	Graphique linéaire de temps de réponse . . . . .	50

# Liste des tableaux

4.1	Caractéristiques détaillées des noeuds de l'infrastructure Fog	46
-----	--	----

# Introduction général

La stratégie de placement adaptative des services pour les architectures Fog Computing vise à garantir une distribution efficace et optimale des services informatiques au sein du réseau fog. Lorsque nous concevons une architecture fog, il est essentiel de prendre en compte la localisation géographique des dispositifs périphériques et les demandes des utilisateurs afin d'offrir une expérience utilisateur fluide et réactive.

L'objectif principal de cette stratégie est de maximiser les performances du système en minimisant la latence et en assurant la disponibilité des services. Nous devons constamment surveiller l'état du réseau, les performances des dispositifs et les besoins des utilisateurs afin de prendre des décisions éclairées sur le placement optimal des services.

L'architecture Fog Computing a été proposée comme une solution pour répondre à ces défis en permettant le traitement et la gestion des données et des services plus proches des objets physiques, dans les couches les plus proches de l'utilisateur. L'architecture Fog Computing permet d'améliorer l'efficacité et les performances des services IoT en offrant des capacités de traitement de données plus proches des utilisateurs et des objets physiques, ce qui réduit les temps de latence et la consommation de la bande passante. En outre, elle permet également une gestion plus efficace des services en offrant des mécanismes de réplication et de synchronisation de services adaptés aux besoins des utilisateurs et des applications. Dans ce contexte, de nombreuses recherches ont été menées pour améliorer l'architecture Fog Computing en proposant de nouvelles méthodes et techniques pour la gestion des données et des services. Ces travaux ont porté notamment sur la réplication des services, le placement des services, la sécurité et la confidentialité des services, la gestion des ressources, etc.

la stratégie de placement adaptative des services pour les architectures Fog Computing nous permet d'assurer une distribution efficace et optimale des services informatiques. En surveillant en temps réel l'état du réseau et les demandes des utilisateurs, nous pouvons prendre des décisions éclairées pour maximiser les performances du système et offrir une expérience utilisateur de haute qualité. Cette approche ouvre la voie à de nombreuses possibilités d'innovation dans le domaine de l'informatique distribuée et de l'Internet des objets.

Dans les chapitres suivants, nous allons explorer plus en détail les différentes techniques et méthodes proposées pour améliorer l'architecture Fog

Computing dans l'environnement de l'IoT.

# Chapitre 1

## *Internet des objets(IOT)*

## 1.1 Introduction :

L'IoT a créé un paradigme qui intègre l'environnement physique nous entourant est intégré, grâce à des capteurs ou actionneurs connectés à Internet. Les capteurs convertissent les paramètres physiques en données numériques, à partir desquelles nous pouvons tirer des informations ou des connaissances relatives à l'environnement physique. Les actionneurs transforment donc les décisions logiques en actions physiques, qui peuvent affecter le monde réel. L'écosystème IoT contient à la fois des dispositifs matériels et des applications logicielles. Les applications IoT analysent les données sensorielles recueillies à partir de capteurs et réalisent des actions au moyen d'actionneurs. Une application IoT peut avoir plusieurs capteurs et actionneurs qui coopèrent afin d'obtenir des informations sur l'environnement et réagir en fonction des informations collectées. L'IoT représente un réseau mondial de dispositifs interconnectés, plus connus sous le nom d'objets intelligents. La coopération des objets entre eux et entre services permet la création de nombreuses applications IoT innovantes.

## 1.2 Définition :

L'Internet des objets (IoT) est un réseau d'appareils physiques (objets) intégrés à des logiciels et des capteurs spéciaux, qui leur permettent de se connecter et de partager des données. N'importe quel objet peut faire partie de l'IoT "des gadgets domestiques intelligents aux outils technologiques industriels " s'il peut être connecté à Internet [27].

La définition de l'IoT inclut également le réseau massif qui connecte les appareils, les personnes et même les animaux via le cloud.

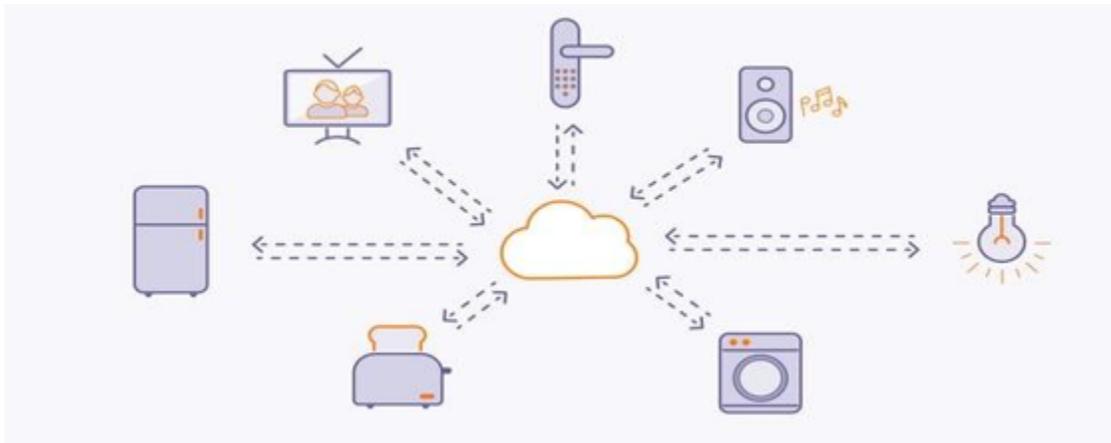


FIGURE 1.1 – L'internet des objets [6]

### 1.3 Les principes de base de l'IOT :

La plupart des appareils IoT ont une sorte de capteur qui leur permet de surveiller diverses conditions. Les appareils IoT peuvent alors « communiquer » entre eux ou partager des données entre eux, de la même manière que les ordinateurs et les smartphones peuvent envoyer et recevoir des informations sans interaction humaine.

De nos jours, les entreprises se précipitent pour ajouter des capteurs et une connectivité Internet à toutes sortes d'articles ménagers (des grille-pains aux porte papiers en passant par les bouteilles de vin), de sorte que vous voyez probablement ces gadgets intelligents de plus en plus lors de vos achats en ligne ou lors d'un voyage au grand magasin [9].

En dehors de la maison, la technologie avancée de l'IoT a de vastes applications commerciales et industrielles, que nous aborderons plus tard. Tout d'abord, comprendre l'Internet des objets nécessite un peu d'histoire.

### 1.4 Le fonctionnement de l'IOT

N'oubliez pas que l'Internet des objets se compose d'appareils ordinaires qui peuvent se connecter à Internet et communiquer entre eux via le cloud. Généralement, cela signifie ajouter des capteurs spéciaux à des articles ordinaires tels que des machines à laver, des systèmes de chauffage, des réfrigérateurs, des montres ou presque tout le reste [15]. Certains appareils utilisent ces capteurs pour collecter des informations et les rapporter. Le tout premier appareil IoT mentionné ci-dessus, un distributeur automatique, utilisait des capteurs pour surveiller son inventaire et mettre ces informations à la disposition du propriétaire.

D'autres appareils peuvent recevoir des informations, puis effectuer une action. Par exemple, les serrures de porte intelligentes reçoivent un signal indiquant que vous souhaitez qu'elles s'ouvrent, puis qu'elles suivent.

Les appareils IoT les plus sophistiqués – et généralement les plus utiles – peuvent faire les deux. Dans l'IoT industriel, il peut s'agir d'une usine surveillant les composants des machines pour détecter d'éventuels des fonctionnements, puis déclenchant l'alarme lorsqu'un problème est détecté. Dans l'IoT à domicile, il peut s'agir de votre thermostat intelligent, qui recueille des informations sur vos préférences et habitudes de température, puis agit en conséquence pour chauffer ou refroidir votre maison à la température souhaitée en fonction de l'heure de la journée.

En général, la technologie intelligente aide les choses à mieux fonctionner, plus efficacement et avec plus de synchronicité. La plupart des gadgets IoT domestiques utilisent votre routeur pour se connecter à une maison intelligente plus grande, ce qui vous permet d'utiliser des commandes à commande vocale ou votre smartphone pour contrôler de nombreuses fonctions de votre maison afin d'économiser du temps, de l'énergie ou les deux.

Sur le plan commercial, la technologie IoT aide les entreprises à surveiller et

à gérer leurs usines, leurs chaînes d’approvisionnement et plus encore. Des capteurs peuvent également être ajoutés à une partie donnée de certaines machines à grande échelle, telles que le forage d’une plate-forme pétrolière, ce qui peut aider à améliorer la production et la sécurité.

## 1.5 Comment les appareils IOT se connectent-ils les uns aux autres ?

L’IoT permet aux appareils connectés au Web de se connecter les uns aux autres et d’agir sur les données qui sont envoyées entre eux. Les appareils utilisent le cloud pour communiquer, en se connectant via Internet via Wi-Fi, une connexion cellulaire (3G ou 4G) ou Bluetooth. Bientôt, la 5G engendrera des possibilités encore plus futuristes, comme les voitures sans conducteur, les villes entièrement interconnectées, les chirurgies à distance et d’autres avancées qui ressemblent actuellement à un film de science-fiction [17].

## 1.6 Les avantages de l’IOT :

**Automatisation** : Les appareils IoT peuvent être programmés pour effectuer des tâches de manière autonome, ce qui peut réduire les coûts et augmenter l’efficacité.

**Surveillance** : Les appareils IoT peuvent être utilisés pour surveiller les processus et les opérations, ce qui peut aider à prévenir les pannes et les problèmes avant qu’ils ne surviennent.

**Connectivité** : L’IoT permet la connectivité entre différents appareils et objets, ce qui permet de créer des systèmes intégrés pour faciliter les processus et les tâches.



FIGURE 1.2 – La connectivité des objets [6]

**Optimisation** : L'IoT peut être utilisé pour optimiser les processus et les opérations, ce qui peut réduire les coûts et augmenter l'efficacité.

**Précision** : Les appareils IoT peuvent collecter et analyser des données en temps réel, ce qui peut aider à améliorer la précision des processus et des opérations.

**Personnalisation** : L'IoT peut être utilisé pour personnaliser les produits et les services en fonction des besoins et des préférences des clients.

**Sécurité** : Les appareils IoT peuvent être utilisés pour renforcer la sécurité des bâtiments, des véhicules et des objets connectés.

**Suivi** : L'IoT peut être utilisé pour suivre la localisation et le mouvement des objets et des personnes, ce qui peut être utile dans les opérations de logistique et de gestion de la chaîne d'approvisionnement.

**Efficacité énergétique** : L'IoT peut être utilisé pour surveiller et contrôler la consommation d'énergie, ce qui peut aider à réduire les coûts et à améliorer l'efficacité énergétique.

**Innovation** : L'IoT peut être utilisé pour développer de nouveaux produits et services, ce qui peut stimuler l'innovation et la croissance économique[7].

## 1.7 Les inconvénients de l'IOT :

**Sécurité** : La sécurité est un problème majeur avec l'IoT. Les appareils IoT peuvent être vulnérables aux cyber attaques, ce qui peut mettre en danger les données et la vie privée des utilisateurs.

**Confidentialité** : Les appareils IoT peuvent collecter de grandes quantités de données sur les utilisateurs, ce qui soulève des préoccupations en matière de confidentialité.

**Coût** : Les appareils IoT peuvent être coûteux à mettre en place et à entretenir, ce qui peut être un obstacle pour les entreprises et les particuliers.

**Complexité** : L'IoT peut être complexe, avec de nombreux appareils et plates-formes différents qui doivent travailler ensemble pour créer des systèmes intégrés.

**Interopérabilité** : L'interopérabilité est un problème majeur avec l'IoT. Les appareils et les plates-formes IoT peuvent ne pas être compatibles les uns avec les autres, ce qui peut rendre difficile l'intégration de différents systèmes.

**Dépendance** : Les utilisateurs peuvent devenir dépendants des appareils IoT, ce qui peut rendre difficile la réalisation de tâches simples sans eux.

**Évolutivité** : L'évolutivité est un problème majeur avec l'IoT. Les systèmes IoT peuvent ne pas être évolutifs, ce qui peut rendre difficile la mise à l'échelle des opérations et des processus [7].

## 1.8 Conclusion :

L'Internet des Objets permet le développement d'un grand nombre d'applications dotant d'intelligence un certain nombre de domaines : santé, maison, ville, télévision, automobile, processus industriels, etc. Le nombre d'objets connectés croît de manière exponentielle. Les solutions techniques se sont développées pour permettre l'interopérabilité entre les différents niveaux : applications, services du nuage, réseaux de communication et composants, du capteur intelligent au système informatique. Les problèmes de sécurité sont un point critique. Plus généralement, l'essor de l'Internet des Objets ne dépend pas uniquement de la possibilité de faire coopérer des objets courants équipés de microélectronique. Il est essentiel qu'existent simultanément des infrastructures fiables et sécurisées, des conditions économiques et légales d'utilisation et un consensus social sur la manière dont les nouvelles opportunités techniques doivent être utilisées.

## Chapitre 2

### *Cloud et Fog computing*

## 2.1 Introduction :

Les infrastructures de l'informatique en brouillard (Fog computing) constituent un élément majeur de l'Internet, Grâce à leur capacité de stockage et de calcul, ainsi que leur placement près des utilisateurs finaux, elles permettent de traiter les données à proximité de l'endroit où elles sont générées. Dans ce chapitre, nous allons introduire des définitions et des explications de certaines notions utilisées dans la présente mémoire.

Par la suite, nous détaillerons les nombreuses évolutions qui ont été nécessaires afin d'aboutir à l'infrastructure d'informatique en brouillard, ou Fog computing, telle que nous la connaissons aujourd'hui.

## 2.2 Cloud computing :

### 2.2.1 Définition :

Le cloud computing ou informatique en nuage est une infrastructure dans laquelle la puissance de calcul et le stockage sont gérés par des serveurs distants auxquels les usagers se connectent via une liaison Internet sécurisée. L'ordinateur de bureau ou portable, le téléphone mobile, la tablette tactile et autres objets connectés deviennent des points d'accès pour exécuter des applications ou consulter des données qui sont hébergées sur les serveurs[26]. Le cloud se caractérise également par sa souplesse qui permet aux fournisseurs d'adapter automatiquement la capacité de stockage et la puissance de calcul aux besoins des utilisateurs.

Pour le grand public, le cloud computing se matérialise notamment par les services de stockage et de partage de données numériques type Box, Dropbox, Microsoft OneDrive ou Apple iCloud sur lesquels les utilisateurs peuvent stocker des contenus personnels (photos, vidéos, musique, documents...) et y accéder n'importe où dans le monde depuis n'importe quel terminal connecté.

### 2.2.2 Les types de cloud computing :

Tous les clouds ne sont pas identiques et aucun type de cloud computing ne convient à tout le monde. Plusieurs modèles, types et services différents ont évolué pour vous aider à trouver la solution adaptée à vos besoins.

Vous devez commencer par déterminer le type de déploiement cloud ou d'architecture de cloud computing sur lequel vos services cloud seront implémentés. Il existe trois modes de déploiement de services cloud : le cloud public, le cloud privé et le cloud hybride. En savoir plus sur les clouds public, privé et hybride.

**Cloud public :**

Un cloud public est détenu et exploité par un fournisseur de services cloud tiers, qui propose des ressources de calcul, telles que des serveurs et du stockage, via Internet. Microsoft Azure est un exemple de cloud public. Dans un cloud public, tout le matériel, tous les logiciels et toute l'infrastructure sont la propriété du fournisseur du cloud. Vous accédez à ces services et vous gérez votre compte par l'intermédiaire d'un navigateur web.

**Cloud privé :**

Le cloud privé est l'ensemble des ressources de cloud computing utilisées de façon exclusive par une entreprise ou une organisation. Le cloud privé peut se trouver physiquement dans le centre de données local de l'entreprise. Certaines entreprises paient également des fournisseurs de services pour qu'ils hébergent leur cloud privé. Le cloud privé est un cloud dans lequel les services et l'infrastructure se trouvent sur un réseau privé.

**Cloud hybride :**

Le cloud privé est l'ensemble des ressources de cloud computing utilisées de façon exclusive par une entreprise ou une organisation. Le cloud privé peut se trouver physiquement dans le centre de données local de l'entreprise. Certaines entreprises paient également des fournisseurs de services pour qu'ils hébergent leur cloud privé. Le cloud privé est un cloud dans lequel les services et l'infrastructure se trouvent sur un réseau privé[31].

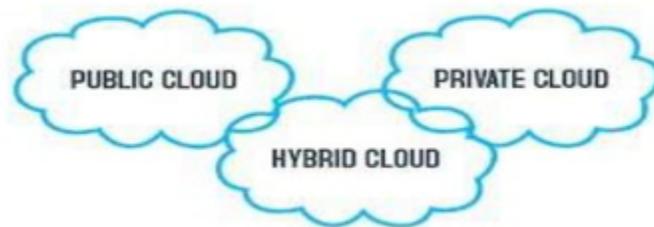


FIGURE 2.1 – Les types de cloud computing [28]

**2.2.3 Les services du cloud computing :**

On distingue plusieurs types de services cloud :

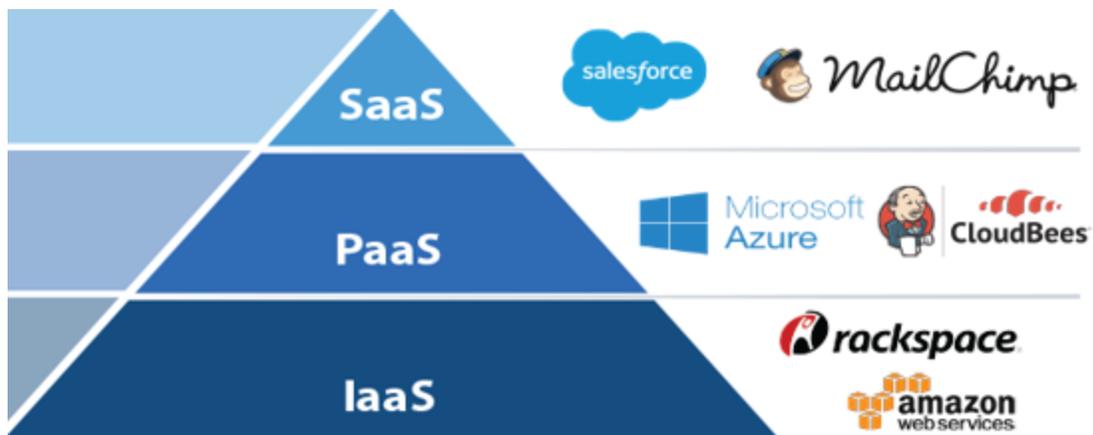


FIGURE 2.2 – Les services du cloud computing [5]

**IaaS (Infrastructure as a service, en anglais) :**

le système d'exploitation et les applications sont installés par les clients sur des serveurs auxquels ils se connectent pour travailler comme s'il s'agissait d'un ordinateur classique.

**PaaS (Platform as a service, en anglais) :**

dans ce mode, c'est le fournisseur du service cloud qui administre le système d'exploitation et ses outils. Le client peut installer ses propres applications si besoin.

**SaaS (Software as a service, en anglais) :**

les applications sont fournies sous forme de services clés en mains auxquels les utilisateurs se connectent via des logiciels dédiés ou un navigateur Internet. Pour le grand public, il s'agit par exemple de messageries électroniques type Gmail, Yahoo, Outlook.com ou de suites bureautiques type Office 365 ou Google Apps [13].

**2.2.4 L'utilisation du cloud computing :**

Vous utilisez probablement en ce moment même le cloud computing sans le savoir. Si vous utilisez un service en ligne pour envoyer des courriers électroniques, modifier des documents, regarder des films ou des programmes de télévision, jouer à des jeux ou stocker des images ou d'autres fichiers, il est probable que le cloud computing intervienne en coulisses. Les premiers services de cloud computing n'ont pas encore dix ans, mais un grand nombre d'organisations, par exemple des start-ups, des multinationales, des services administratifs ou des ONG, adopte cette technologie pour de nombreuses raisons[12].

Vous trouverez ici quelques exemples des possibilités d'utilisation des services cloud d'un fournisseur de cloud :

### **Créer des applications cloud natives :**

Créer, déployer et mettre à l'échelle rapidement des applications (web, mobiles et API). Tirez parti des technologies et approches cloud natives, telles que les conteneurs, Kubernetes, l'architecture de microservices, la communication pilotée par des API et DevOps.

### **Stocker, sauvegarder et récupérer des données :**

Protégez vos données à moindre coût et à grande échelle en les transférant via Internet vers un système de stockage cloud hors site, accessible à partir de tout emplacement et appareil.

### **Diffuser du contenu audio et vidéo :**

Communiquez avec votre public en tout lieu, en tout temps et sur tout appareil via un système audio et vidéo haute définition mondialement distribué.

### **Tester et générer des applications :**

Réduisez les coûts et délais de développement d'applications en utilisant des infrastructures cloud dont l'échelle peut être facilement adaptée.

### **Analyser des données :**

Unifiez vos données entre les équipes, les divisions et les emplacements dans le cloud. Utilisez ensuite des services cloud, par exemple de Machine Learning et d'intelligence artificielle, pour extraire des informations analytiques qui vous permettent de prendre des décisions éclairées.

### **Incorporer de l'intelligence :**

Utilisez des modèles intelligents pour interagir avec les clients et fournir des informations analytiques à partir des données capturées.

## **2.2.5 Les avantages du cloud computing :**

Le cloud computing est radicalement différent de l'approche traditionnelle que les entreprises adoptent en matière de ressources informatiques. Voici sept raisons courantes pour lesquelles les organisations optent pour des services de cloud computing :

### **Coût :**

Le cloud computing élimine la nécessité d'investir dans du matériel et des logiciels, et de configurer et de gérer des centres de données sur site : racks de serveurs, alimentation électrique permanente pour l'alimentation

et le refroidissement, experts informatiques pour la gestion de l'infrastructure. La facture est vite salée.

### **Mise à l'échelle mondiale :**

La mise à l'échelle élastique est un des avantages des services de cloud computing. En termes de cloud, cela veut dire qu'il est possible de mettre en œuvre la quantité nécessaire de ressources informatiques, par exemple plus ou moins de puissance de calcul, de stockage ou de bande passante, au moment où elles sont nécessaires, là où elles sont nécessaires.

### **Performances :**

Les plus grands services de cloud computing s'exécutent sur un réseau de centres de données sécurisés, dont le matériel est régulièrement mis à niveau pour assurer des performances rapides et efficaces. Ceci offre plusieurs avantages par rapport à un centre de données classique, y compris un temps de latence réseau réduit pour les applications et de plus grandes économies d'échelle.

### **Sécurité :**

De nombreux fournisseurs de cloud offrent un vaste éventail de stratégies, technologies et contrôles qui renforcent globalement votre situation de sécurité, contribuant ainsi à protéger vos données, vos applications et votre infrastructure contre des menaces potentielles.

### **Vitesse :**

La plupart des services de cloud computing sont fournis en libre-service et à la demande. D'énormes ressources de calcul peuvent donc être mises en œuvre en quelques minutes et en quelques clics, offrant ainsi aux entreprises un haut niveau de flexibilité et les dégageant de la pression liée à la planification de la capacité.

### **Productivité :**

Les centres de données sur site nécessitent en général la manipulation de matériel, la mise à jour des logiciels et d'autres corvées informatiques qui prennent beaucoup de temps. Le cloud computing supprime la plupart de ces tâches et les équipes informatiques peuvent donc passer plus de temps à travailler à la concrétisation des objectifs de l'entreprise.

### **Fiabilité :**

Le cloud computing simplifie la sauvegarde des données, la récupération d'urgence et la continuité des activités. Il rend ces activités moins coûteuses.

teuses, car les données peuvent être mises en miroir sur plusieurs sites redondants au sein du réseau du fournisseur[14].

## **2.2.6 Les inconvénients du cloud computing :**

### **Sécurité :**

Le stockage de données dans le cloud peut poser des problèmes de sécurité, car les données sont stockées sur des serveurs tiers qui peuvent être vulnérables aux attaques de hackers. Les entreprises doivent prendre des mesures de sécurité supplémentaires pour protéger leurs données.

### **Disponibilité :**

Le cloud computing dépend de la disponibilité de l'Internet. Si votre connexion Internet est en panne, vous ne pouvez pas accéder à vos données stockées dans le cloud.

### **Coûts :**

Bien que le cloud computing puisse être rentable, cela dépend de l'utilisation que vous en faites. Si vous utilisez beaucoup de données, les coûts peuvent augmenter rapidement.

### **Dépendance à l'égard du fournisseur :**

Lorsque vous utilisez le cloud computing, vous dépendez du fournisseur de services cloud pour stocker et gérer vos données. Si le fournisseur de services est en panne ou cesse ses activités, vous pourriez perdre l'accès à vos données.

### **Confidentialité :**

Les données stockées dans le cloud sont souvent accessibles à d'autres utilisateurs du même service cloud. Les entreprises doivent prendre des mesures pour protéger leurs données sensibles et confidentielles[14].

## **2.3 Fog computing**

### **2.3.1 Définition :**

Le terme "Fog Computing" a été utilisé pour la première fois en 2012 par Jonathan Bar-Magen Numhauser dans sa proposition de thèse "Fog Computing introduction to a New Cloud Evolution" [22]. L'auteur y présente les limites du Cloud Computing face aux exigences en QoS des nouvelles applications telles que les applications de l'Internet of Things (IoT) et à l'augmentation conséquente du nombre d'utilisateurs ainsi que le trafic réseau

qu'ils génèrent.

Il propose un nouveau paradigme de calcul, qu'il appelle "Fog Computing". Ce dernier a pour but d'étendre les capacités et les services offerts par les infrastructures du Cloud Computing. Par la suite, le terme est repris par Flavio Bonomi en 2012 dans son article "Fog Computing and Its Role in the Internet of Things" [18]. Il met en évidence les avantages et les limites que peut avoir ce nouveau paradigme de calcul dans des environnements de type Internet of Things (IoT).

En 2015, l'université de Princeton en collaboration avec les industriels Cisco Systems, Intel, Microsoft et ARM fondent l'OpenFog consortium afin de promouvoir le Fog computing dans le monde industriel. Ils publient en 2016 "OpenFog Reference Architecture" qui expose les principales caractéristiques des Architectures Fog [23] que nous allons voir dans la section.

En 2017, le National Institute of Standards and Technology (NIST) publie un rapport technique [21] qui présente un modèle conceptuel standardisé du Fog, les caractéristiques du Fog, les stratégies de déploiement des services dans ce type d'infrastructures. Dans le cadre de cette thèse, on considère la définition suivante qui regroupe à la fois celle du NIST [Michaela 2018a] et celle de l'OpenFog consortium [23] : "Le Fog Computing ou Fog Networking est un paradigme de calculs distribué à large échelle et hiérarchique, qui permet d'augmenter les capacités de calcul, de stockage et de communication des serveurs Cloud par celles des noeuds intermédiaires situés entre ces serveurs et les utilisateurs finaux."

### **2.3.2 Différenciation avec le cloud-computing :**

Le fog computing se distingue avant tout du cloud computing par le lieu où les ressources sont mises à disposition et où les données sont traitées. Le cloud computing se fait généralement dans des centres de données centralisés. Des ressources comme la puissance de traitement ou la capacité de stockage sont regroupées au niveau de serveurs principaux et utilisées par les clients via le réseau[24]. La communication entre deux périphériques ou plus s'effectue donc toujours via un serveur en arrière-plan.

Une telle architecture atteint ses limites avec des concepts tels que « l'usine intelligente », où les données doivent être échangées en permanence entre d'innombrables terminaux. Le fog computing recourt ici à un traitement intermédiaire proche de la source de données, afin de réduire le débit de données vers le centre de calcul.

### **2.3.3 Différenciation avec le edge-computing :**

Le débit de données des architectures IdO à grande échelle n'est pas le seul inconvénient du cloud computing, il y a aussi le problème de la latence. Le traitement de données centralisé est à chaque fois associé au retard dû à des chemins de transmission longs[24]. Les terminaux et les capteurs doivent toujours s'adresser au serveur du centre de calcul pour communi-

quer entre eux, attendre le traitement externe de la demande ainsi que la réponse. Ces délais dus à la latence deviennent un problème, notamment dans les processus de fabrication basés sur l'IdO, pour lesquels il faut assurer le traitement de l'information en temps réel, afin que les machines puissent réagir immédiatement aux incidents.

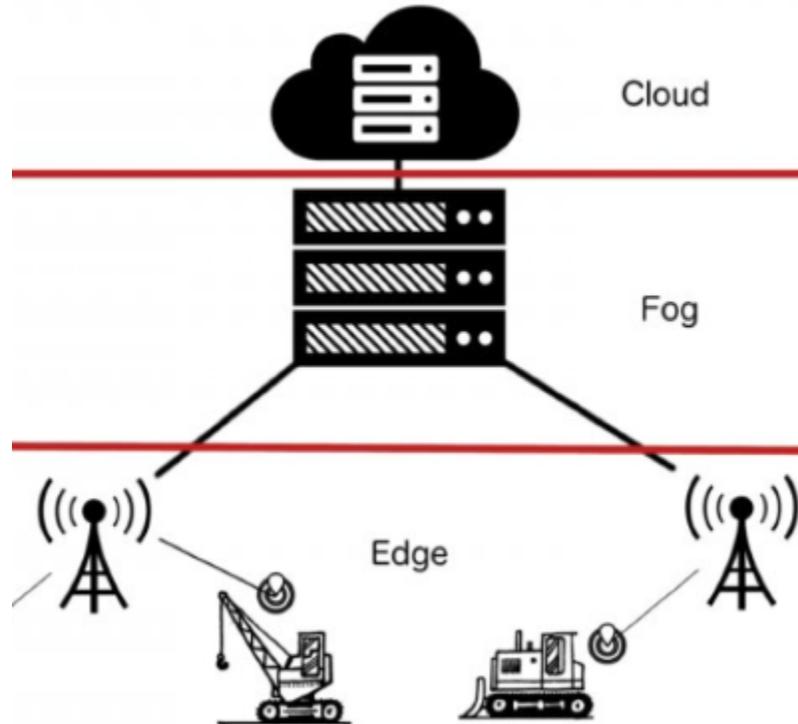


FIGURE 2.3 – Edge et fog computing [3]

Le Edge-computing propose une solution à ce problème de latence. C'est un concept qui, dans le cadre du fog computing, propose un traitement des données non seulement décentralisé, mais directement dans le terminal et donc à la périphérie (edge) du réseau. Chaque « dispositif intelligent » est équipé de son propre microcontrôleur permettant le traitement de données de base et la communication avec d'autres appareils et capteurs de l'IdO.

#### 2.3.4 La communication dans une infrastructure fog computing :

Le Fog est une infrastructure dynamique qui exploite des éléments déjà déployés pour proposer des services de calculs et de communications. Il peut ainsi utiliser toutes les technologies de communication déjà existantes[11]. Cependant, des infrastructures comme le Fog sont majoritairement composées de technologies de communication sans fils aussi bien avec infrastructure (mobile) que sans infrastructure (Peer-to-Peer etc.). On inclut égale-

ment les technologies des communications spécifiques aux réseaux de capteurs (WSN) ou d'une manière plus générique à l'Internet des Objets (IoT).

### 2.3.5 Le fonctionnement du fog computing :

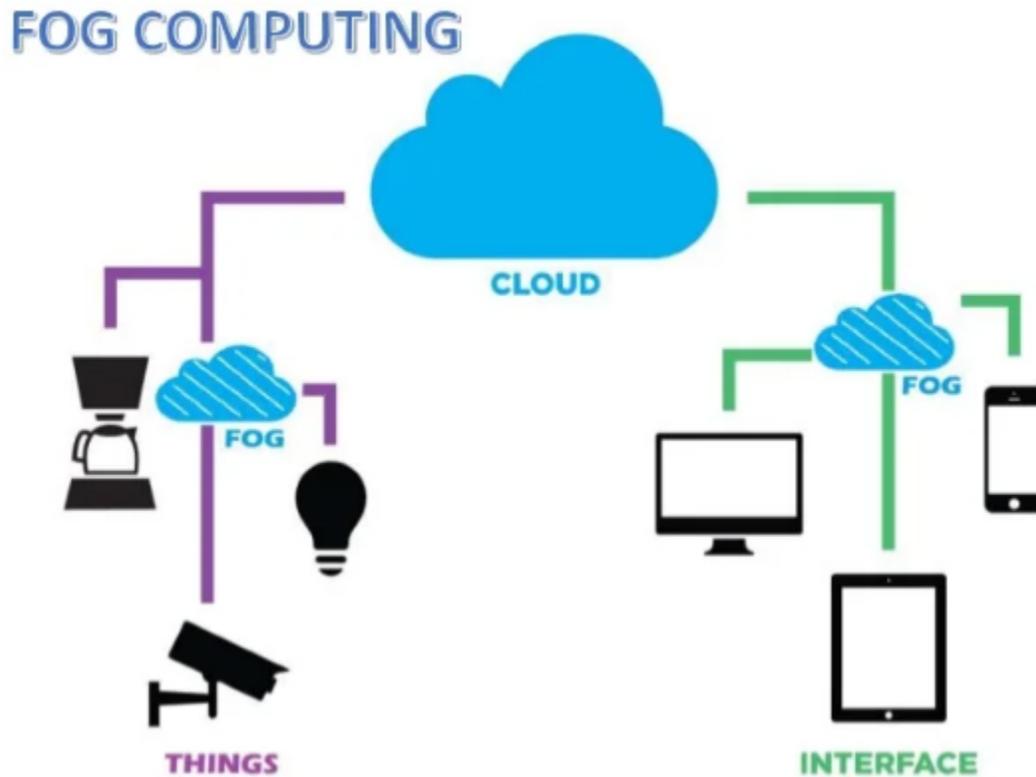


FIGURE 2.4 – fog computing [2]

Les nœuds de Fog sont installés à proximité des appareils IoT (Internet des objets) ou des utilisateurs qui génèrent des données. Ces nœuds peuvent être des routeurs, des commutateurs, des passerelles ou des appareils IoT. Les données sont collectées à partir des capteurs, des dispositifs IoT ou des utilisateurs à proximité et sont traitées sur les nœuds de Fog locaux[30]. Les nœuds de Fog peuvent également stocker des données localement pour réduire la latence et améliorer la disponibilité. Les données traitées peuvent être transférées vers le cloud pour un stockage à long terme, une analyse ou un traitement supplémentaire. Les applications de Fog computing sont conçues pour fonctionner sur les nœuds de Fog locaux, plutôt que sur les serveurs distants dans le cloud. Cela permet une réponse plus rapide et plus efficace aux demandes de traitement de données. Les nœuds de Fog peuvent être gérés et surveillés à distance à partir d'un centre de gestion de Fog, qui peut être situé dans le cloud ou sur site.

### 2.3.6 Les caractéristiques du fog computing :

Le National Institute of Standards and Technology (NIST) et l'OpenFog consortium ont identifié un certain nombre de caractéristiques définissant les infrastructures Fog et que nous résumons dans la liste non exhaustive suivante :

#### **Le temps de latence :**

Les applications métiers d'aujourd'hui exigent un temps de réponse de l'ordre de quelques secondes ou millisecondes, surtout quand il faut des actions correctives. Dans ce contexte, le fog computing élimine le besoin d'envoyer des données pour traitement vers le cloud. Par conséquent, le calcul s'effectue plus rapidement et le réseau de brouillard peut traiter de gros volumes de données avec un délai minimal. Autrement dit, la latence entre l'entrée et la réponse est minimisée. L'objectif est de fournir une réactivité au niveau de la milliseconde, permettant le traitement des données en temps quasi réel.

#### **Conservation de la bande passante du réseau :**

Le transfert de données entre la périphérie du réseau vers un serveur cloud nécessite beaucoup de bande passante. Imaginez une situation où chacun de vos appareils IoT génère 100 Go de données par jour, alors que vous en avez des milliers. Le transfert de cette grande quantité de données peut prendre des jours. En fait, étant donné que le fog computing traite les données localement, il conserve au maximum la bande passante du réseau.

#### **Distribution géographique :**

L'infrastructure Fog peut s'étendre sur une très large zone géographique à l'échelle d'une ville ou d'une région. Cette infrastructure va héberger des applications dont les composants peuvent être distribués dans n'importe quelle région.

#### **Hétérogénéité et hiérarchisation des nœuds :**

L'architecture Fog la plus répandue dans la littérature et qui est aussi la plus générique (également utilisée par le NIST) définit l'architecture Fog comme une architecture en couches 3 tiers[8]. Ce modèle représente une hiérarchie entre les nœuds de calcul, allant des équipements utilisateurs jusqu'au centre de données en passant par des équipements Edge et Fog Computing. Le principe est que les capacités de calcul, de stockage et la consommation énergétique des nœuds augmentent en allant de la première couche de nœuds utilisateurs à la dernière représentant les centres de données.

**Sensibilité à la mobilité et à la géolocalisation des utilisateurs :**

Le Fog est un environnement hétérogène et dynamique. Ceci est principalement dû à la mobilité des utilisateurs et des équipements à la périphérie du réseau. Il est donc important d'avoir les positions géographiques des utilisateurs et de connaître la portée des applications qu'ils utilisent.

**Sécurité :**

Le Fog Computing peut pallier les failles de sécurités des réseaux de capteurs en offrant des nœuds intermédiaires avec des protocoles de communication plus sécurisés que ceux des objets connectés. Le Fog permet également de gérer les services et applications au plus proche des utilisateurs. Les opérations de stockage de données, de filtrage et traitement ayant lieu au plus proche des utilisateurs évitent de faire transiter des informations sensibles vers les serveurs Cloud.

**Interopérabilité et fédération :**

Une infrastructure Fog étendue regroupe plusieurs domaines réseau et de calcul gérés par différents organismes. Ces domaines doivent coopérer pour assurer un déploiement fluide et une gestion transparente des services pour les utilisateurs.

**Autonomie et programmabilité :**

L'une des principales caractéristiques des infrastructures Fog est l'autonomie de décision pour la gestion du cycle de vie des applications qui y sont déployées. La facilité de programmabilité et de reconfiguration des équipements grâce à la virtualisation simplifie la gestion et assure l'évolutivité de l'infrastructure face à la dynamique de l'environnement.

**2.3.7 Les avantages et les inconvénients d'une architecture fog computing :**

Le fog computing offre des solutions à divers problèmes des infrastructures informatiques de type cloud. L'accent est mis sur les approches qui promettent des voies de communication courtes et minimisent le téléchargement vers le cloud. Le traitement décentralisé en périphérie du réseau présente également des inconvénients[29]. Ceux-ci sont avant tout le fait des coûts de maintenance et d'administration d'un système distribué.

**Les avantages :**

- Moins de trafic réseau : le fog computing réduit le trafic entre les dispositifs IdO et le cloud.
- Disponibilité hors ligne : les appareils IdO sont également disponibles hors ligne quand ils sont intégrés dans une architecture de fog computing.

- Latence réduite : le fog computing raccourcit les voies de communication et accélère ainsi les processus automatisés d'analyse et de décision.
- Sécurité des données : quand on a recours au fog computing, le pré-traitement des données de l'appareil a lieu sur le réseau local. Cela permet une implémentation dans laquelle les données sensibles restent dans l'entreprise ou peuvent être cryptées ou rendues anonymes avant d'être téléchargées sur le cloud.
- Réduction des coûts liés à l'utilisation de réseaux tiers : les opérateurs de réseau font payer cher le téléchargement rapide vers le cloud ; une partie de ces coûts peut être économisée grâce au fog computing.

#### **Les inconvénients :**

- Coûts de matériel plus élevé : les terminaux et capteurs IdO de l'architecture fog computing doivent être équipés d'une unité de traitement supplémentaire pour permettre le traitement de données local et la communication entre appareils.
- Faible protection contre les pannes ou les abus : les entreprises qui ont recours au fog computing doivent équiper les appareils et les capteurs IdO de contrôleurs situés à la périphérie du réseau, p.ex. dans des installations de production qui sont difficiles à protéger contre les défaillances ou une mauvaise utilisation.
- Augmentation des besoins de maintenance : le traitement décentralisé des données implique des besoins accrus de maintenance. La raison en est que les contrôleurs et les éléments de stockage sont répartis sur l'ensemble du réseau et que, contrairement aux solutions cloud, ils ne peuvent être ni maintenus ni administrés de manière centralisée.

## **2.4 Conclusion :**

Le fog computing a été introduit pour répondre à un besoin de traitement avec une faible latence que les infrastructures de cloud computing ne sont pas capables de fournir, mais également pour une question de passage à l'échelle. De fait, le cloud ne pouvait faire face à l'explosion du nombre d'utilisateurs. Il est également intéressant de noter que les approches de fog computing ou d'edge computing sont très semblables et ne se distinguent que sur des aspects mineurs. Autrement dit, le fog considère des ressources organisées de façon continue entre le cloud et les utilisateurs alors que l'edge computing ne considère qu'un niveau intermédiaire. En pratique, la plupart des solutions logicielles déployées pour de l'edge computing fonctionnent dans une infrastructure fog et inversement. Aujourd'hui, les infrastructures de fog sont introduites dans le réseau 5G afin de fournir une puissance de calculs aux téléphones portable.

# Chapitre 3

## *L'approche propose*

### 3.1 Introduction :

Le placement des services dans une architecture Fog Computing est un défi important à relever pour optimiser les performances et la qualité de service de l'ensemble du système. En effet, le choix des nœuds Fog sur lesquels les services doivent être déployés à un impact direct sur les temps de réponse, la bande passante et la consommation d'énergie. Dans cette section, nous allons présenter une approche afin d'améliorer la répliation et le placement des services dans l'architecture Fog Computing.

### 3.2 L'architecture proposée :

La figure suivante montre une représentation schématique des trois couches de notre architecture de fog computing :

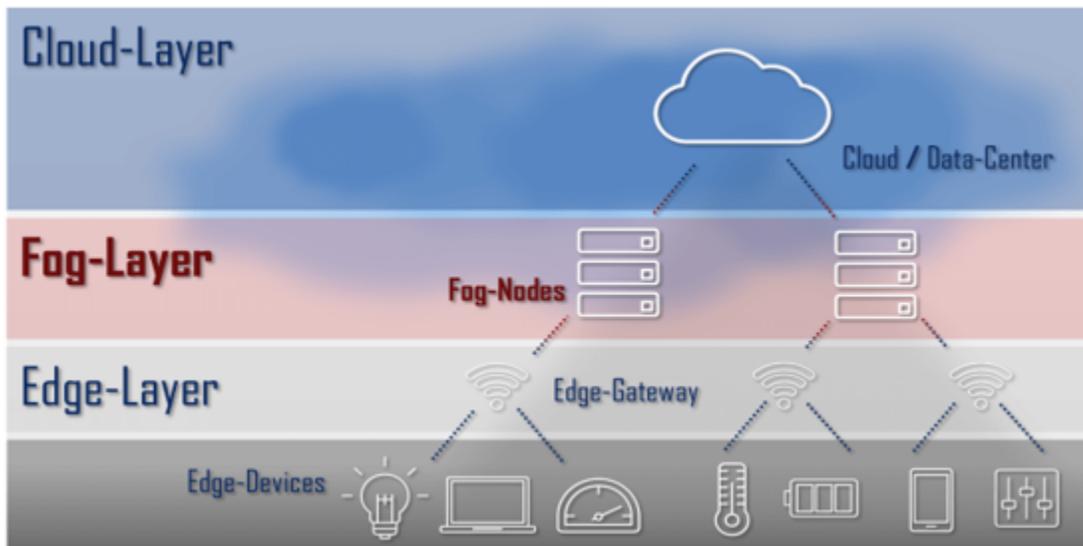


FIGURE 3.1 – L'architecture du fog computing [4]

#### 3.2.1 Couche périphérique (edge layer) :

la couche périphérique comprend tous les dispositifs « intelligents » (dispositifs périphériques) d'une architecture IdO. Les données générées au niveau de la couche périphérique sont soit traitées directement sur le périphérique, soit transmises à un serveur (fog node) de la couche fog.

#### 3.2.2 Couche fog (fog layer) :

la couche fog comprend un certain nombre de serveurs performants qui reçoivent les données de la couche périphériques, les prétraitent, et les téléchargeent dans le cloud si nécessaire.

### 3.2.3 Couche cloud (cloud layer) :

la couche cloud est le terminal de données central d'une architecture de cloud computing. Dans le fog computing, les ressources de stockage et de prétraitement des données sont externalisées du cloud public ou d'un centre de données central et fournies de manière décentralisée sur le réseau par des nœuds fog d'une couche intermédiaire.

## 3.3 La stratégie de placement des services proposée :

Dans cette section, nous présentons notre stratégie pour le placement des services, qui comprend trois phases importantes : la première phase permet l'analyse des requêtes des utilisateurs en fonction de leur proximité et de la disponibilité des services demandés, la deuxième phase consiste à traiter le placement réel des services et de déclencher le processus de réplication des services manquants, la dernière phase concerne la suppression des services inutiles.

### 3.3.1 La phase de traitement des requêtes :

La phase de traitement des requêtes est une étape clé dans notre stratégie proposée de placement des services dans l'architecture Fog Computing .

L'organigramme suivant détaille le principe de traitement des requêtes :

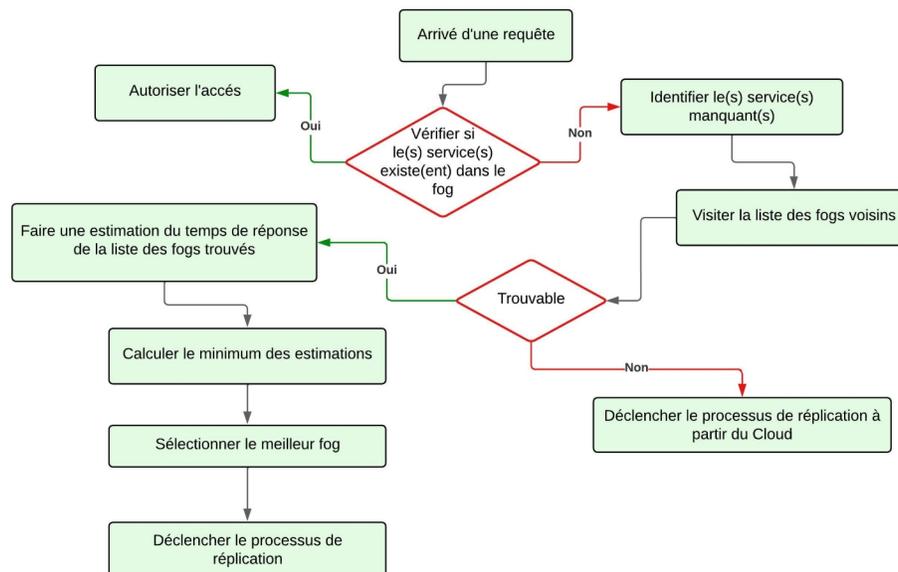


FIGURE 3.2 – L'organigramme de traitement des requêtes

Lorsqu'une requête est émise par un utilisateur, elle est reçue par le nœud

Fog le plus proche de l'utilisateur. Le nœud Fog examine alors la requête et détermine le type des services requis. Si les services sont disponibles sur ce nœud Fog, alors cette requête est traitée immédiatement. Sinon, le nœud Fog doit rechercher dans les nœuds Fog les plus proches disposants des services requis. Une estimation du temps de réponse ( $R_s$ ) est calculé en se basant sur la formule suivante, les Fogs qui ont la plus petite estimation du  $R_s$  seront choisis :

$$R_s = \text{latence d'envoi d'un message} + \left( \frac{\text{taille de service}}{\text{bande passante}} \right) + \text{latence de répondre au message}$$

Une fois que le nœud Fog approprié pour le traitement de la requête est trouvé, un processus de réplication sera déclenché sur le nœud qui a le minimum temps de réponse.

Sinon, le processus sera déclenché à partir du cloud. Afin d'améliorer le processus de recherche, les Fogs échangent leurs listes des services entre eux. Le partage de la liste des services entre les nœuds Fog Computing fait référence à la communication et à la synchronisation des informations sur les services disponibles entre les différents nœuds de périphérie. Le choix du cas de partage de la liste des services entre les nœuds Fog dépend des besoins spécifiques de l'application et de l'architecture. Nous distinguons 3 solutions, chaque solution présente ses avantages et ses inconvénients en termes de réactivité, de surcharge du réseau et de gestion des ressources.

#### **1. Informer les autres nœuds fog après un certain nombre d'ajout de service :**

Lorsqu'un service est ajouté à un nœud fog, celui-ci incrémente un compteur NbSer qui compte le nombre de service qui ont été ajouté depuis la dernière mise à jour ; si la valeur de ce compteur dépasse le seuil de mise à jour, alors le fog informe immédiatement les autres nœuds de la mise à jour de la liste des services disponibles. La valeur du Compteur NbSer est réinitialisé après chaque lancement de mise à jour. .

##### Avantages :

Permet une réactivité immédiate en informant les autres nœuds dès qu'un certain nombre de services sont ajoutés. Assure une disponibilité rapide des services, offrant aux utilisateurs un accès rapide à une multitude de nouveaux services ajoutés.

##### Inconvénients :

- Peut entraîner une surcharge du réseau de périphérie si de nombreux services sont ajoutés fréquemment.
- Peut provoquer une utilisation inefficace des ressources si les autres nœuds doivent maintenir constamment une liste à jour.

#### **2. Informer les autres nœuds fog après un intervalle de temps :**

Les mises à jour de la liste des services ajoutés à un nœud fog sont transmises périodiquement aux autres nœuds après un certain intervalle de temps prédéfini par exemple, toutes les heures ou toutes les 24 heures.

##### Avantages :

- Réduit la surcharge du réseau en limitant la fréquence des mises à jour de la liste des services.
- Économise les ressources en évitant une mise à jour constante des listes des services sur les autres nœuds.

**Inconvénients :**

- Les autres nœuds peuvent ne pas être immédiatement informés des nouveaux services ajoutés, entraînant un délai pour les utilisateurs qui souhaitent accéder à ces services.
- La périodicité des mises à jour peut nécessiter un réglage précis pour équilibrer la réactivité et la surcharge du réseau.

**3. Informer les autres nœuds fog d'une manière hybride :**

Une combinaison des deux approches précédentes est utilisée, où certains services sont immédiatement signalés aux autres nœuds fog, tandis que d'autres sont transmis périodiquement après un intervalle de temps.

**Avantages :**

- Permet une réactivité immédiate pour les services critiques ou populaires, tandis que les autres services sont mis à jour périodiquement.
- Équilibre la réactivité et la surcharge du réseau en choisissant sélectivement les services qui nécessitent une notification immédiate.

**Inconvénients :**

- Peut nécessiter une gestion plus complexe pour différencier les services qui nécessitent une notification immédiate de ceux qui peuvent être mis à jour périodiquement.
- Une mauvaise configuration de l'approche hybride peut entraîner des retards dans la notification des nouveaux services importants.

**3.3.2 La phase de réplication des services :**

La phase de réplication des services consiste à créer des copies des services sur d'autres nœuds Fog pour améliorer la disponibilité des services et réduire les temps de latence. Cette phase peut se dérouler en étapes :

Réplication sur le nœud actuel : Dans cette étape, on vérifie si le service est disponible dans le nœud actuel, s'il se trouve, nous autorisons l'accès et nous traitons la requête, sinon nous faisons une deuxième vérification :

Nous comptons pour chaque service le nombre d'accès, nous vérifions si le nœud dispose d'un espace de stockage suffisant pour créer une copie de ce service. Si c'est le cas, la copie de ce service est créée sur ce nœud. Cela permet de garantir que le service est disponible même si le nœud original tombe en panne ou est surchargé.

Sinon, le processus de suppression sera déclenché.

Dans le cas où le nombre d'accès à ce service ne dépasse pas le SEUIL de réplication, nous autorisons un accès à distance à ce service et nous incrémentons le nombre d'accès à ce dernier.

Après la suppression nous répétons la vérification sur la capacité du stockage, si nous obtenons un espace, nous créons la réplique. Dans le cas contraire, où le stockage reste insuffisant alors nous créons la réplique sur

un autre nœud voisin du nœud actuel.

Suppression des services inutiles : Si le nœud actuel n'a pas suffisamment d'espace de stockage pour répliquer le service, un processus de suppression est déclenché. Ce processus vise à identifier et supprimer les services inutiles présents sur le nœud.

Nous détaillons les principes de fonctionnement de ce processus dans la prochaine phase.

Le fonctionnement de processus de réplication est décrit dans l'organigramme suivant :

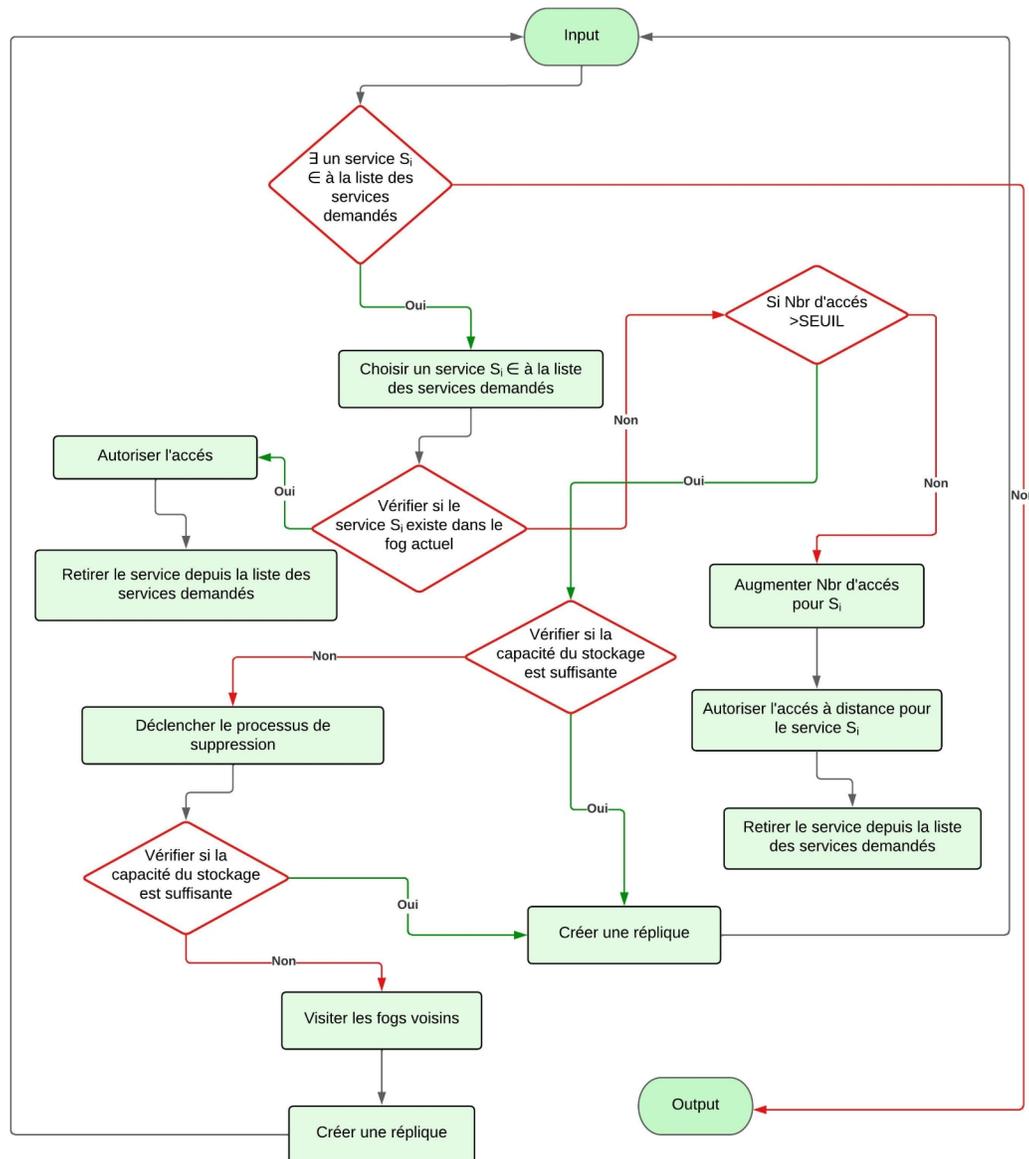


FIGURE 3.3 – L'organigramme de réplication des services

Cette approche de réplication des services permet d'améliorer la disponibilité des services en assurant leur présence sur plusieurs nœuds Fog. Elle

permet également de réduire les temps de latence en rapprochant les services des utilisateurs, puisque les copies des services sont réparties sur plusieurs nœuds fog dans le réseau.

**Les avantages de la gestion des répliques :**

- Amélioration de la disponibilité : Les répliques des services garantissent que les utilisateurs peuvent accéder aux services même en cas de panne d'un nœud ou de conditions défavorables.
- Réduction des temps de latence : En plaçant les répliques des services à proximité des utilisateurs finaux, les temps de réponse sont réduits, ce qui permet d'améliorer les performances.
- Équilibrage de charge : La répartition des services sur plusieurs nœuds de périphérie permet de répartir la charge de manière équilibrée, évitant ainsi les goulets d'étranglement et les ralentissements.

**Les inconvénients potentiels de la gestion des répliques :**

- Complexité de gestion : La gestion des répliques et des décisions de placement peut être complexe, nécessitant une surveillance continue, des mécanismes de détection de pannes et une coordination entre les nœuds de périphérie.
- Surcharge des ressources : La réplication de services sur plusieurs nœuds de périphérie peut entraîner une utilisation inefficace des ressources, notamment en termes de puissance de calcul, de stockage et de bande passante réseau.

### 3.3.3 La phase de suppression :

Dans cette phase, les services qui ne sont plus utilisés seront supprimés des nœuds Fog pour libérer des ressources et améliorer la performance du système.

Nous parcourons chaque nœud Fog dans le réseau, puis pour chaque service présent sur le nœud Fog, nous vérifions si le nombre d'accès au service est inférieur à un seuil prédéfini. Si c'est le cas, cela signifie que le service est considéré comme inutile et peut être supprimé du nœud Fog. Cette opération est effectuée pour chaque nœud Fog dans le réseau, permettant ainsi la suppression des services inutiles sur l'ensemble du système.

En classifiant les services qui ont un nombre d'accès inférieur au SEUIL par ordre croissant, puis en sélectionnant qui a le minimum, nous le marquons comme désactiver puis nous le supprimons si la capacité de stockage est insuffisante pour créer des nouvelles répliques des services plus demandés. L'organigramme ci-dessous détaille le principe de fonctionnement de la suppression des services inutiles :

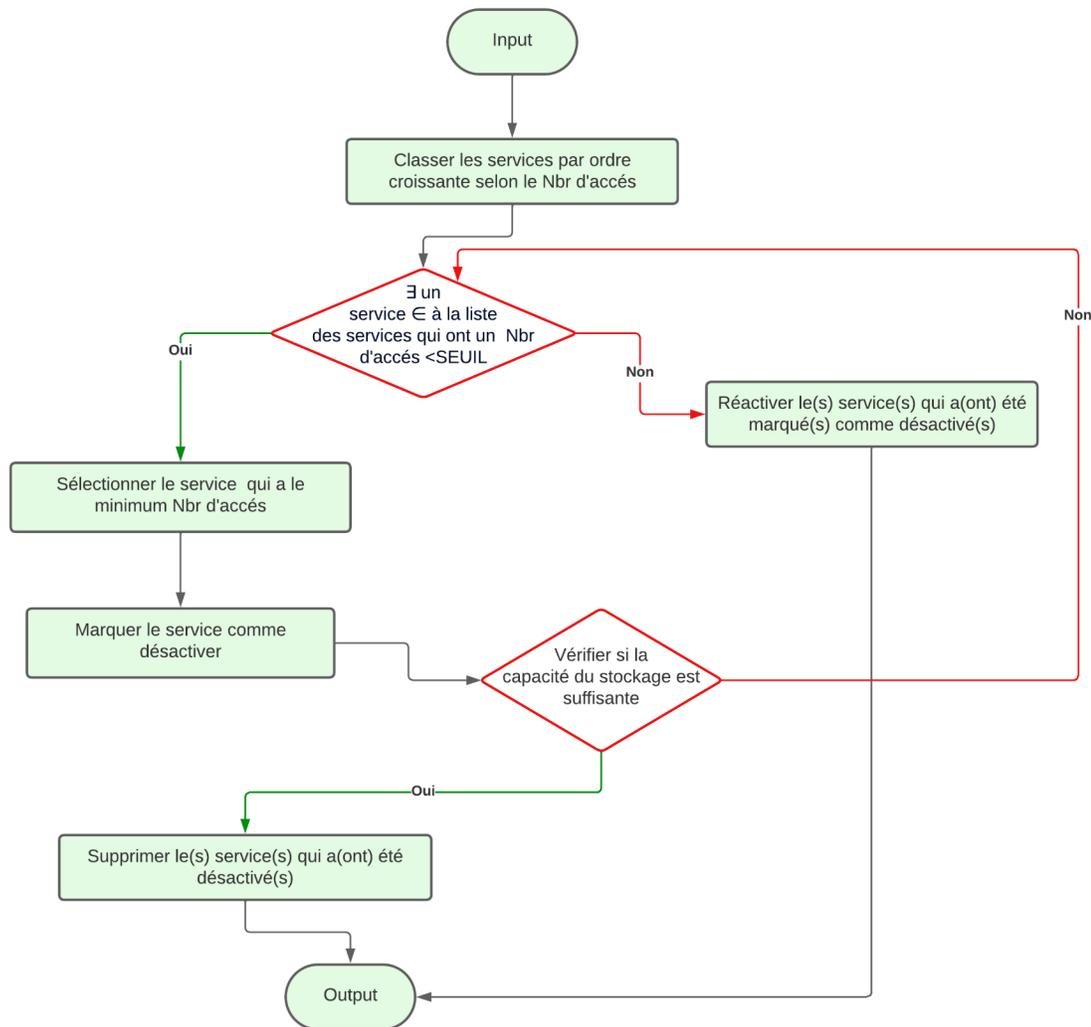


FIGURE 3.4 – L'organigramme de suppression des services inutiles

### 3.4 Conclusion :

En conclusion, notre approche proposée pour le placement des services dans l'architecture Fog computing prend en compte plusieurs critères tels que le temps de réponse, la disponibilité et la capacité de stockage des nœuds Fog. Nous avons également proposé une stratégie de réplication des services pour améliorer la disponibilité et la résilience du système, les services sont placés sur les nœuds Fog les mieux équipés pour les traiter, ce qui garantit des temps de réponse plus rapides et une utilisation plus efficace des ressources. De plus, la réplication des services assure une disponibilité continue des services, même en cas de panne d'un nœud Fog.

## Chapitre 4

### *Implémentation et résultats de simulations*

## 4.1 Introduction :

Dans ce chapitre, nous allons présenter les outils que nous avons utilisés pour mettre en œuvre notre stratégie de placement des services dans le Fog computing. Nous commencerons par le langage de programmation Java et l'environnement de développement Eclipse. Ensuite, nous parlerons du simulateur IfogSim et des classes fondamentales qu'il utilise.

Nous illustrerons aussi notre approche à travers une étude de cas et des expérimentations réalisées sur la plateforme de simulation. Enfin, nous discuterons des résultats obtenus et des perspectives de cette approche pour l'optimisation du placement des services dans les architectures Fog Computing.

## 4.2 Environnement de Développement :

Pour implémenter et tester notre stratégie de placement adaptative des services dans les architectures Fog Computing et internet of things , nous avons développé le simulateur IFogSim afin de supporter notre proposition. Nous avons utilisé l'environnement du travail suivant pour cette implémentation :

— Caractéristiques matérielles et logicielles de l'ordinateur utilisés : nous avons développé notre application sur une machine avec un processeur Intel(R) Core(TM) i7-7700Q CPU @ 2.80GHz et d'une capacité mémoire de 8Go.

Système d'exploitation Windows 10 ,64 bits.

Simulateur utilisé : iFogSim.

Langage utilisée : Java.

Version Java Développement Kit utilisée : JDK 19.

IDE utilisée : Eclipse.

### 4.2.1 Le langage de programmation Java :

Java est un langage de programmation orienté objet développé par Sun Microsystems en 1995. Depuis lors, il est devenu l'un des langages de programmation les plus populaires au monde.

Java est souvent utilisé pour développer des applications Web, des applications mobiles et des logiciels d'entreprise. Il est également utilisé pour créer des jeux vidéo et des applications embarquées. Java est un langage de programmation orienté objet, ce qui signifie qu'il utilise des objets pour représenter des données et des fonctionnalités. Les objets sont créés à partir de classes, qui sont des modèles pour la création d'objets [19].

Java est également portable, ce qui signifie que les programmes écrits en Java peuvent être exécutés sur différents systèmes d'exploitation sans avoir besoin de modifications importantes. Cela est possible grâce à la machine virtuelle Java (JVM), qui interprète le code Java et le traduit en instructions

pour le système d'exploitation sous-jacent.



**Les avantages de Java :**

Java est un langage de programmation très populaire, ce qui signifie qu'il dispose d'une grande communauté de développeurs et de nombreuses ressources en ligne pour aider les nouveaux programmeurs à apprendre. De plus, Java est un langage de programmation sûr et sécurisé. Il dispose d'un mécanisme de gestion de la mémoire qui empêche les erreurs de corruption de la mémoire et les fuites de mémoire. Il dispose également de fonctionna-

lités de sécurité intégrées pour protéger les programmes contre les attaques malveillantes [10].

### **Les outils de développement Java :**

Il existe de nombreux outils de développement disponibles pour Java, tels que Eclipse, NetBeans et IntelliJ IDEA. Ces outils offrent des fonctionnalités telles que la coloration syntaxique, la complétion automatique du code et le débogage.

En outre, Java dispose de sa propre plateforme de développement, Java Development Kit (JDK). JDK comprend tous les outils nécessaires pour développer, tester et déployer des applications Java [15].

### **L'avenir de Java :**

Java continue d'évoluer avec de nouvelles versions publiées régulièrement. La dernière version majeure de Java est Java 16, publiée en mars 2021. Java est également utilisé dans des domaines émergents tels que l'Internet des objets (IoT) et l'apprentissage automatique. Avec sa portabilité et ses fonctionnalités de sécurité, Java est bien positionné pour continuer à être l'un des langages de programmation les plus populaires et les plus utilisés dans les années à venir [15].

#### **4.2.2 Eclipse IDE :**

Eclipse est un environnement de développement intégré (IDE) open source pour le développement de logiciels. Il est principalement utilisé pour le développement de logiciels Java, mais il prend également en charge d'autres langages de programmation tels que C++, Python et PHP [16]



Eclipse est un outil puissant qui offre une gamme de fonctionnalités avancées telles que la complétion automatique de code, la refonte de code, la

détection d'erreurs de syntaxe et l'intégration avec des outils de gestion de version comme Git et SVN.

### **Installation et configuration d'Eclipse IDE :**

L'installation d'Eclipse est simple et peut être effectuée en téléchargeant le package approprié pour votre système d'exploitation depuis le site officiel d'Eclipse. Une fois installé, vous pouvez personnaliser l'environnement selon vos besoins en ajoutant des plugins et en configurant les paramètres. Il est important de noter que la configuration d'Eclipse dépend du projet sur lequel vous travaillez. Par exemple, si vous travaillez sur un projet Java, vous devrez configurer le chemin d'accès aux bibliothèques Java et au JDK.

### **Fonctionnalités clés d'Eclipse IDE :**

Eclipse offre une multitude de fonctionnalités avancées pour faciliter le développement de logiciels. L'une des fonctionnalités les plus utiles est la complétion automatique de code, qui permet de gagner du temps en proposant des suggestions de code en temps réel. Eclipse dispose également d'un débogueur intégré qui permet de détecter et de corriger les erreurs de code plus rapidement.

En outre, Eclipse prend en charge l'intégration avec des outils de gestion de version tels que Git et SVN, ce qui facilite la collaboration avec d'autres développeurs. Enfin, Eclipse dispose d'une grande communauté de développeurs qui publient régulièrement des plugins pour ajouter des fonctionnalités supplémentaires à l'IDE.

### **4.2.3 IFogSim :**

IFogSim est une plateforme de simulation open-source pour les environnements de l'Internet des objets (IoT) et de l'informatique en nuage. Elle permet aux chercheurs et développeurs de tester et évaluer les performances des systèmes IoT et de l'informatique en nuage dans des environnements réalistes avant leur déploiement [20].

### **Fonctionnalités de IFogSim :**

IFogSim offre une variété de fonctionnalités pour simuler des scénarios IoT et de l'informatique en nuage, y compris la modélisation des capteurs, des actuateurs, des nœuds et des applications.

Elle prend également en compte les contraintes de communication, les politiques de gestion de données, les stratégies de placement de services et les techniques d'optimisation de la consommation d'énergie.

### Applications de IFogSim :

IFogSim peut être utilisé pour simuler des applications IoT et de l'informatique en nuage dans divers domaines tels que la santé, l'agriculture, la ville intelligente, l'industrie 4.0 et l'énergie. Elle peut aider à concevoir des systèmes efficaces, à identifier les goulots d'étranglement, à évaluer les performances et à optimiser les coûts avant le déploiement sur le terrain[20].

### Avantages de IFogSim :

IFogSim offre plusieurs avantages par rapport aux autres outils de simulation, notamment sa flexibilité, sa modularité, sa facilité d'utilisation et son support multiplateforme.

Elle permet également d'accélérer le processus de développement, de réduire les coûts de test et de minimiser les risques liés au déploiement de systèmes IoT et de l'informatique en nuage [25].

### Architecture de IFogSim :

IFogSim utilise une architecture de type fog computing qui permet de déployer des services de calcul, de stockage et de réseau plus près des utilisateurs finaux.

Cette approche réduit la latence, améliore la qualité de service et optimise l'utilisation des ressources disponibles dans le réseau. Dans la bibliothèque iFogSim, les éléments d'architecture de l'Internet des objets nébuleux (fog computing) sont modélisés sous forme de classes [20]. Les principales classes de iFogSim sont les suivantes :

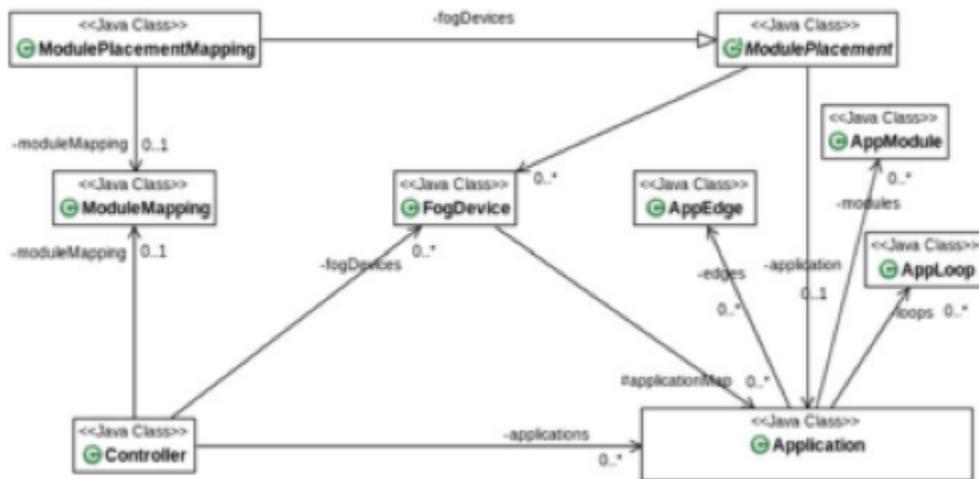


FIGURE 4.1 – Les classes fondamentales d'iFogSim [1]

1. Sensor (capteur) : la classe Sensor représente un capteur dans le modèle de simulation iFogSim. Les capteurs sont utilisés pour collecter des données à partir de l'environnement ou des périphériques connectés, et transmettent

ces données aux nœuds de la brume pour le traitement ultérieur.

2. Actuator (actionneur) : la classe Actuator représente un actionneur dans le modèle de simulation iFogSim. Les actionneurs sont utilisés pour contrôler les dispositifs connectés ou effectuer des actions physiques en réponse aux signaux émis par les nœuds de la brume.

3. FogDevice (dispositif brume) : la classe FogDevice représente un nœud de la brume dans le modèle de simulation iFogSim. Les nœuds de la brume sont des dispositifs de traitement distribué utilisés pour fournir des services de calcul et de stockage dans des environnements de l'Internet des objets.

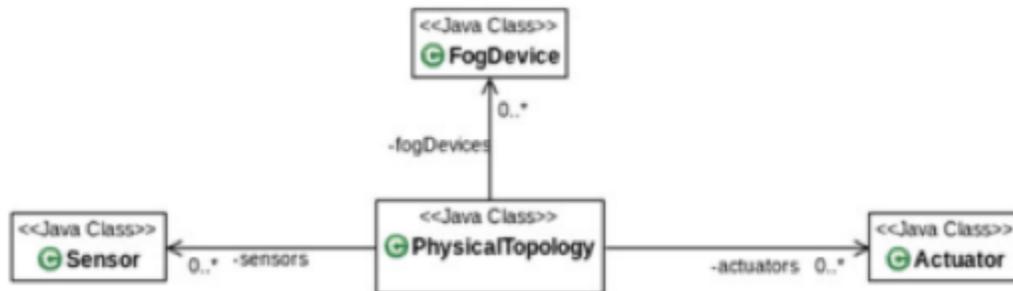


FIGURE 4.2 – Les classes de topologie physique iFogSim [1]

4. Tuple (n-uplet) : la classe Tuple représente un n-uplet dans le modèle de simulation iFogSim. Les n-uplets sont utilisés pour représenter des données structurées dans les environnements informatiques distribués, et sont souvent utilisés pour la transmission de données entre les nœuds de la brume.

5. ResourceManagementService (service de gestion des ressources) : la classe ResourceManagementService représente un service de gestion des ressources dans le modèle de simulation iFogSim. Les services de gestion des ressources sont utilisés pour allouer et gérer les ressources informatiques, telles que le processeur, la mémoire et la bande passante réseau, dans les environnements de l'Internet des objets.

6. Application (application) : la classe Application représente une application dans le modèle de simulation iFogSim. Les applications sont des programmes informatiques conçus pour effectuer des tâches spécifiques, et peuvent être exécutées sur des nœuds de la brume pour fournir des services aux utilisateurs, Ces entités sont réalisées à l'aide des classes suivantes :

- AppEdge : dénote la dépendance des données entre une paire d'application modèle.
- AppLoop : une classe supplémentaire, utilisée pour spécifier le processus de contrôle des boucles d'intérêt pour l'utilisateur.
- AppModule : Représente les éléments de traitement des applications de brouillard.

7. MonitoringService (service de surveillance) : la classe MonitoringService représente un service de surveillance dans le modèle de simulation iFogSim. Les services de surveillance sont utilisés pour surveiller les performances

et la santé des nœuds de la brume et des applications, et pour signaler les problèmes ou les défaillances aux administrateurs pour qu'ils puissent y remédier.

En utilisant ces classes, iFogSim permet de simuler le comportement du réseau fog et d'évaluer les performances de différentes configurations du réseau. Les interactions entre ces classes sont gérées par des services simulés qui implémentent les fonctions de traitement des données, de routage des messages et de gestion des ressources.

## 4.3 L'implémentation :

### 4.3.1 Le modèle d'application :

Modèle d'application. Comme illustré dans la Figure 3, l'application se compose de deux modules principaux qui effectuent le traitement : Client et Calculateur de concentration. Les modules de l'application sont modélisés dans iFogSim en utilisant la classe AppModule.

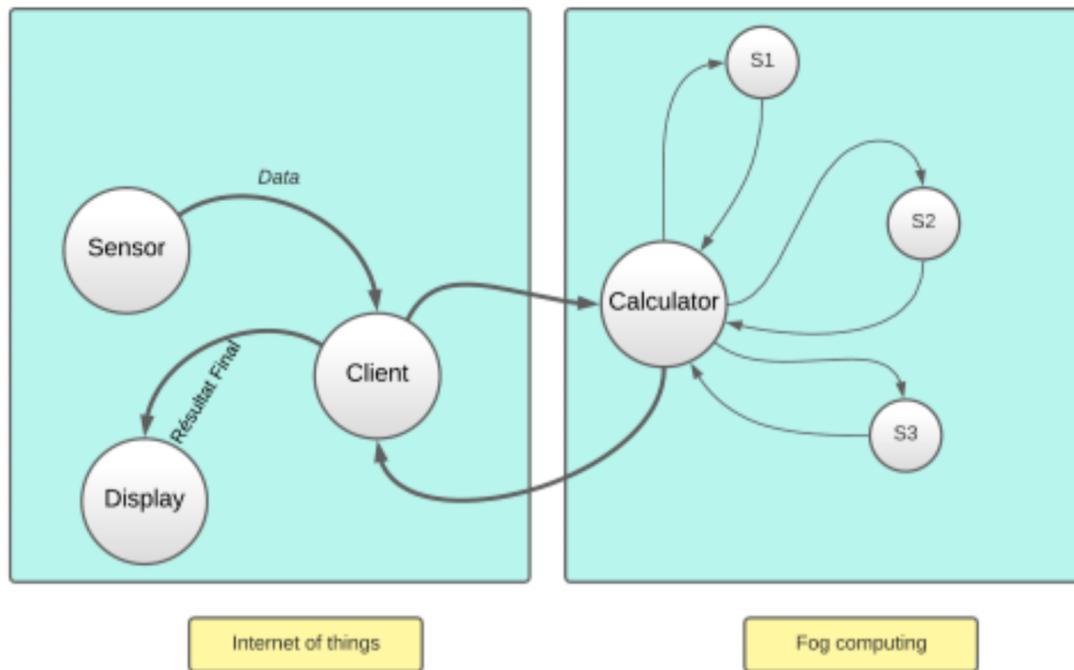


FIGURE 4.3 – Modèle d'application

#### Client :

Le module Client est responsable de l'interaction avec l'utilisateur ou d'autres composants externes. Il peut recevoir des entrées, envoyer des re-

quêtes et afficher des résultats. Il agit comme une interface utilisateur pour l'application.

### Calculateur :

Le module Calculateur est chargé de réaliser des calculs ou des traitements spécifiques liés à la requête d'un utilisateur. Il peut prendre des données en entrée provenant du module Client ou d'autres sources, effectuer des opérations de calcul ou de traitement sur ces données, puis générer des résultats ou des sorties. Si la requête d'un utilisateur a besoin de plusieurs services, le calculateur va interroger ces services avant de générer des résultats.

Ces deux modules interagissent entre eux ou avec d'autres parties du système via des dépendances de données ou des échanges d'informations. Les détails spécifiques de ces interactions peuvent être modélisés en utilisant la classe `AppEdge` fournie par `iFogSim`.

Enfin, la boucle de contrôle d'intérêt pour l'application est modélisée dans `iFogSim` en utilisant la classe `AppLoop`. L'application reçoit des signaux via un capteur et un actionneur `DISPLAY` affiche les résultats.

### 4.3.2 Caractéristiques détaillées des nœuds de l'infrastructure Fog :

Les nœuds de l'infrastructure Fog sont des dispositifs informatiques situés en périphérie du réseau, plus proches des utilisateurs finaux que le Cloud centralisé.

***nodeName*** : Nom du nœud Fog.

***level*** : Niveau hiérarchique du nœud Fog dans l'architecture Fog (par exemple, Fog niveau 1, Fog niveau 2, etc.).

***mips*** : Millions d'instructions par seconde (MIPS) disponibles pour le traitement des tâches.

***uplink Bandwidth*** : Capacité de bande passante montante pour envoyer des données du nœud Fog vers le Cloud ou d'autres nœuds Fog.

***downlink Bandwidth*** : Capacité de bande passante descendante pour recevoir des données du Cloud ou d'autres nœuds Fog.

***RAM*** : Capacité de mémoire vive (Random Access Memory) disponible pour le stockage temporaire des données.

***rate Per Mips*** : Taux d'utilisation de l'énergie par unité de MIPS.

***busyPower*** : Consommation d'énergie lorsque le nœud Fog est en mode de traitement intensif (actif).

***idlePower*** : Consommation d'énergie lorsque le nœud Fog est en mode d'attente (inactif).

Voici les caractéristiques détaillées des nœuds de l'infrastructure Fog et le cloud dans notre stratégie :

Paramètre	Type de nœud		
	Cloud	Fog	Mobile
nodeName	cloud	dept	m
level	0	1	2
mips	44800	2800	1000
downlink Bandwidth	40000	4000	1000
uplink Bandwidth	100	10000	10000
RAM	10000	10000	270
rate Per Mips	0.01	0.0	0
busyPower	16*103	107.339	87.53
idlePower	16*83.25	83.4333	82.44

TABLE 4.1 – Caractéristiques détaillées des nœuds de l'infrastructure Fog

### 4.3.3 L'interface principale :

L'utilisation d'interfaces graphiques permet à l'utilisateur d'interagir facilement avec le système de simulation. Cela rend l'expérience utilisateur plus agréable et intuitive et parce-que Le simulateur IfogSim n'a pas une interface graphique, la configuration et l'exécution se fait sur console , Nous avons créé une interface qui facilite l'accès au simulateur .

L'interface graphique principale de notre application comprendra différents éléments tels que des boutons, des zones de texte, etc.

#### L'interface Fog Device :

- **les paramètres des nœuds Fog :**

Des champs de saisie pour spécifier les paramètres spécifiques à chaque nœud Fog : nodeName, level, mips ,downlink Bandwidth,uplink Bandwidht, RAM, rate Per Mips .

- **Un bouton "Ajouter" :**

Une fois que l'utilisateur a terminé de remplir les paramètres du nouveau nœud Fog, il peut cliquer sur un bouton "Ajouter" pour ajouter le nœud Fog à la topologie existante.Les caractéristiques des noeuds ajoutés sera afficher dans un tableau .



Lorsque vous utilisez l'interface Service, vous pouvez remplir les champs de texte en spécifiant le nom du service, la taille du service et le fog auquel vous souhaitez le lier. Ensuite, en cliquant sur le bouton "Ajouter", vous pouvez ajouter ce service au nœud du brouillard.

- **Un bouton "Répliquer" :**

pour créer une copie de service un nœuds Fog. Lorsque l'utilisateur clique sur ce bouton, il peut sélectionner un service spécifique à répliquer et spécifier les nœuds Fog sur lesquels la réplication doit être effectuée.

- **Un bouton "Supprimer" :**

pour supprimer un service spécifique d'un nœud Fog. Lorsque l'utilisateur clique sur ce bouton, il peut sélectionner le service à supprimer et confirmer son action.

### L'interface topology :

l'interface topologie vous permet d'accéder aux fichiers existants et d'afficher la topologie associée à ces fichiers. Cela vous offre une représentation visuelle de la structure et des relations entre les éléments de la topologie.

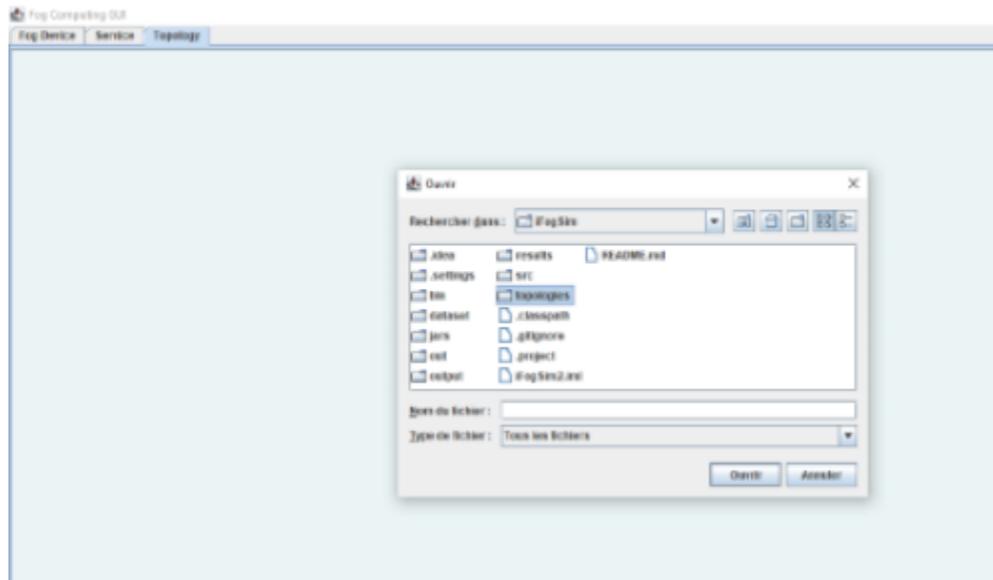


FIGURE 4.6 – L'interface Topology

### 4.3.4 Résultats :

Dans le cadre de notre projet de développement d'une stratégie de placement adaptatif des services, nous avons réalisé une série d'expérimenta-

tions afin de valider et d'évaluer l'efficacité de notre module. Les résultats obtenus ainsi que leurs interprétations sont présentés dans cette section.

**Métriques utilisées :**

- **Consommation énergétique :**

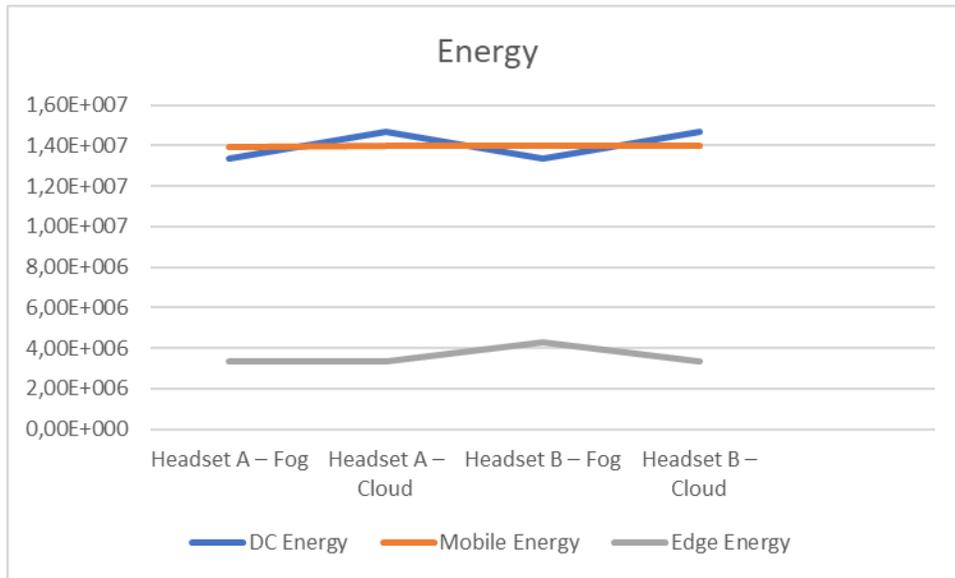


FIGURE 4.7 – Graphique linéaire de consommation d'énergie moyenne

Cette métrique mesure la quantité d'énergie consommée par les nœuds de fog pour fournir les services demandés. Une stratégie de placement efficace devrait chercher à minimiser la consommation énergétique en tenant compte des ressources disponibles et de la charge de travail.

- **La latence :**

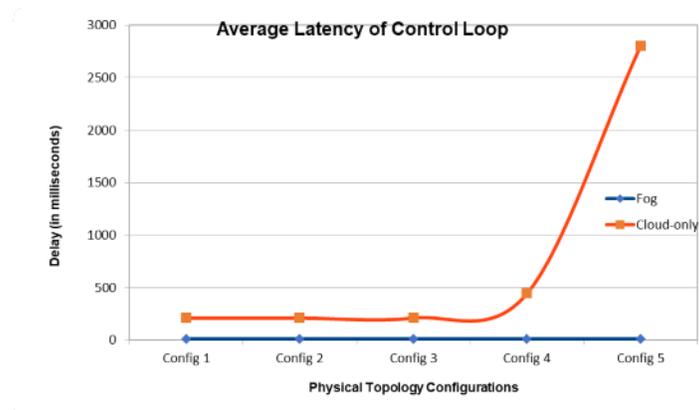


FIGURE 4.8 – Graphique linéaire de latence moyenne

La métrique de latence est importante dans la conception, le déploiement et

l'optimisation des architectures de Fog Computing. Les décisions relatives à la localisation des ressources de Fog, au routage des données et à la distribution des charges de travail sont prises en tenant compte de cette métrique afin de fournir des services plus performants et une meilleure expérience utilisateur.

- **Temps de réponse :**

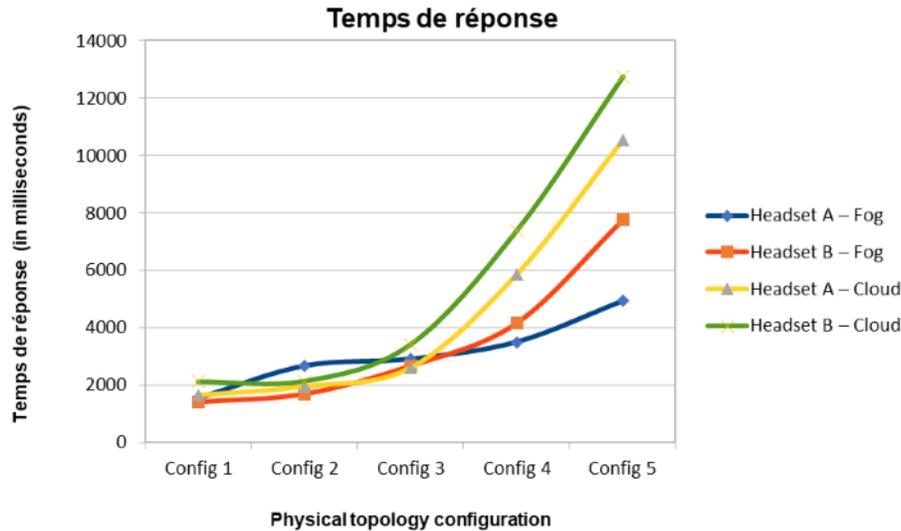


FIGURE 4.9 – Graphique linéaire de temps de réponse

Cette métrique mesure le temps nécessaire pour répondre aux demandes des utilisateurs. Un temps de réponse court est essentiel pour offrir une expérience utilisateur fluide et réactive. Les stratégies de placement de services doivent prendre en compte la latence de communication entre les utilisateurs et les nœuds de fog afin de minimiser le temps de réponse global.

## 4.4 Conclusion :

Dans ce chapitre, nous avons présenté les outils que nous avons utilisés pour mettre en œuvre notre stratégie de placement des services dans l'architecture Fog computing. Nous avons discuté du langage de programmation Java, de l'environnement de développement Eclipse et du simulateur Ifog-Sim. Nous avons également expliqué les différentes étapes nécessaires pour configurer la simulation de notre approche.

# Conclusion général

Notre mémoire a abordé l'importance du placement adaptatif des services dans l'architecture Fog Computing et l'Internet des Objets (IoT). Nous avons constaté que les défis liés à la répartition des services dans ces environnements nécessitent des solutions novatrices afin de garantir des performances optimales.

Pour répondre à ces défis, nous avons proposé une solution reposant sur une approche de réplication dynamique des services. Cette approche permet de réduire les temps de réponse et d'améliorer l'utilisation de la bande passante en plaçant les services de manière optimale dans le réseau Fog. En répartissant les services de manière efficace, nous avons pu minimiser les goulots d'étranglement et améliorer la qualité de service pour les utilisateurs finaux.

Il est important de souligner que le Fog Computing et l'IoT sont des technologies en constante évolution. Les recherches et les développements continus sont nécessaires pour relever les défis actuels et futurs liés à la gestion de la répartition des services dans ces environnements. Les améliorations technologiques, les nouvelles normes et les nouvelles architectures doivent être explorées pour optimiser davantage les performances du Fog Computing et de l'IoT.

En conclusion, nous avons proposé des pistes de recherche futures pour améliorer notre solution et explorer d'autres approches innovantes afin de résoudre les défis associés à la gestion de la répartition des services dans le Fog Computing et l'IoT. Ces pistes incluent l'utilisation de techniques d'apprentissage automatique pour l'optimisation du placement des services, l'intégration de mécanismes de sécurité robustes et l'exploration de nouvelles méthodes de gestion des ressources.

# Bibliographie

- [1] An extension to ifogsim to enable the design of data placement strategies. *website* [www.researchgate.net](http://www.researchgate.net).
- [2] fog computing and cloud computing. *website* [techgenix.com](http://techgenix.com).
- [3] iot edge fog computing. *website* [www.simform.com](http://www.simform.com).
- [4] peripherie et brouillard que sont edge et fog computing. *website* [www.nextinpact.com](http://www.nextinpact.com).
- [5] tapping saas paas iaas. *website* [blog.crozdesk.com](http://blog.crozdesk.com).
- [6] what is the internet of things. *website* [www.avast.com](http://www.avast.com).
- [7] Rostand Affogbolo, Claire Gauzente, and Pascale Kuntz. Fiabilité du système de production industrielle grâce à l'iot–proposition théorique et exploration empirique. In *XXIVème Conférence de l'AIM, Nantes, France, 2019*.
- [8] Jungyeon Baek. *Artificial intelligence-empowered resource management system for fog computing networks*. PhD thesis, École de technologie supérieure, 2022.
- [9] Yassine Banouar. *Gestion autonome de la QoS au niveau middleware dans l'IoT*. PhD thesis, Université Paul Sabatier-Toulouse III, 2017.
- [10] Nicolas Berthier. Gestion hybride de la mémoire dynamique dans les systèmes java temps-réel. *Rapport de magistère, 2* :115–139, 2007.
- [11] El Houssine Bourhim. La communication et le placement de conteneurs docker dans le fog computing. 2020.
- [12] Stéphane Calé, Matthieu Chaveyriat, and Stéphane Lenco. Cloud externe : les meilleures pratiques pour assurer son utilisation en toute sécurité. *Sécurité et stratégie, (1)* :48–58, 2017.
- [13] Mehdi Chouiten. *Architecture distribuée dédiée aux applications de Réalité Augmentée mobile*. PhD thesis, Université d'Evry-Val d'Essonne, 2013.
- [14] Cédric Denuit and Hervé Jacquemin. " cloud computing et gratuité : quel coût pour le consommateur? qualification juridique, étude des modes de rémunération et examen de la relation contractuelle.
- [15] Didier Donsez. Les outils de développement pour java.
- [16] I Eclipse. Eclipse ide. *Website* [www.eclipse.org](http://www.eclipse.org) *Last visited : July*, pages 1–20, 2017.

- [17] KHALIFA Fatiha. Réseaux avancés. 2023.
- [18] Jiang Zhu et Sateesh Addepalli Flavio Bonomi, Rodolfo Milito. Fog computing and its role in the internet of things. dans proceedings of the first edition of the mcc workshop on mobile cloud computing. pages 13—16, 2017.
- [19] Thierry Groussard. *Java 6 : les fondamentaux du langage Java*. Editions ENI, 2009.
- [20] Harshit Gupta, Amir Vahid Dastjerdi, Soumya K Ghosh, and Rajkumar Buyya. ifogsim : A toolkit for modeling and simulation of resource management techniques in the internet of things, edge and fog computing environments. *Software : Practice and Experience*, 47(9) :1275–1296, 2017.
- [21] Barton Robert J.Martin Michael Goren Nedim et Mahmoudi Charif Iorga Michaela, Feldman Larry. Fog computing conceptual model recommendations of the national institute of standards and technology. *Rapport technique*, pages 2, 24, 25, 26, 29, 2018.
- [22] Bar-Magen Numhauser Jonathan. *Fog Computing introduction to a New Cloud Evolution*. PhD thesis, University of Alcalá, Spain,, 2012.
- [23] Bar-Magen Numhauser Jonathan. Openfog consortium architecture working. *Rapport technique, OpenFog Consortium*, pages 24–25, 2016.
- [24] Mithun Mukherjee, Lei Shu, and Di Wang. Survey of fog computing : Fundamental, network applications, and research challenges. *IEEE Communications Surveys & Tutorials*, 20(3) :1826–1857, 2018.
- [25] Fatin Hamadah Rahman, Thien Wan Au, SH Shah Newaz, and Wida Susanty Haji Suhaili. A performance study of high-end fog and fog cluster in ifogsim. In *Computational Intelligence in Information Systems : Proceedings of the Computational Intelligence in Information Systems Conference (CIIS 2018)* 3, pages 87–96. Springer, 2019.
- [26] Boujemaa REHALI et al. Cloud computing ou informatique à la demande : enjeux et perspectives. *Revue Marocaine de Gestion et d’Economie*, 3(7), 2016.
- [27] Imad Saleh. Internet des objets (ido) : Concepts, enjeux, défis et perspectives. *Revue Internet des objets*, 2(10.21494), 2018.
- [28] SarveshMeena. cloud computing. *website www.slideshare.net*.
- [29] Luiz Angelo Steffanel and Manuele Kirsch-Pinheiro. Accès aux données dans le fog computing : le cas des dispositifs de proximité. In *Conférence d’informatique en Parallélisme, Architecture et Système (Compas)*, 2019.
- [30] Luiz Angelo Steffanel and Manuele Kirsch Pinheiro. Stratégies multi-échelles pour les environnements pervasifs et l’internet des objets. In *Journées Francophones Mobilité et Ubiquité (Ubimob)*, 2016.
- [31] KHAIZAR YUCEF. Une approche hybride pour le contrôle d’accès.

## ملخص

تتناول هذه المذكرة تحسين مواقع الخدمات في هندسة الضباب المحوسبة وإنترنت الأشياء. الهدف الرئيسي لهذا العمل هو اقتراح طريقة لتحديد المواقع المتكيفة للخدمات من أجل تقليل وقت الاستجابة وتقليل استهلاك عرض النطاق الترددي.

في البداية، تم إجراء تحليل شامل للأدبيات المتاحة لفهم التحديات والمشكلات المرتبطة بإدارة الخدمات في بيئة الضباب المحوسبة وإنترنت الأشياء. ثم تم اقتراح بنية مرجعية لتنفيذ طريقة متكيفة لتحديد مواقع الخدمات.

تتيح هذه الطريقة تحديد الموقع الأمثل للخدمات في الوقت الحقيقي وفقاً لاحتياجات المستخدمين، وذلك من خلال تقليل أوقات الاستجابة.

للتحقق من صحة الطريقة المقترحة، تم إجراء تجارب باستخدام بيئة محاكاة تسمى iFogSim

في الختام، اقترحت هذه المذكرة طريقة مبتكرة لتحديد مواقع الخدمات المتكيفة في هندسة الضباب المحوسبة وإنترنت الأشياء، مما يتيح الاستجابة لاحتياجات المستخدمين في الوقت الحقيقي. أظهرت نتائج التجارب فاعلية الطريقة المقترحة، مما يجعلها حلاً واعدًا لتطبيقات إنترنت الأشياء في المستقبل.

## Abstract

This dissertation focuses on the optimization of service placement in Fog Computing and the Internet of Things (IoT) architecture. The main objective of this work is to propose an adaptive placement method for services to minimize response time and reduce bandwidth consumption.

Initially, a comprehensive analysis of existing literature was conducted to understand the challenges and issues related to service management in a Fog Computing and IoT environment. Subsequently, a reference architecture was proposed for implementing an adaptive service placement method.

This method enables the determination of the optimal location for services in real-time based on user requirements, thereby minimizing response times. To validate the proposed method, experiments were conducted using a simulation environment called iFogSim.

In conclusion, this dissertation has proposed an innovative method for adaptive service placement in Fog Computing and IoT architecture, which caters to real-time user needs. The results of the experiments

## Résumé

Ce mémoire porte sur l'optimisation de l'emplacement des services dans l'architecture Fog Computing et l'Internet des Objets (IoT). L'objectif principal de ce travail est de proposer une méthode d'emplacement adaptative pour les services afin de minimiser le temps de réponse et de réduire la consommation de la bande passante.

Dans un premier temps, une analyse approfondie de la littérature existante a été réalisée pour comprendre les enjeux et les défis liés à la gestion des services dans un environnement Fog Computing et IoT. Ensuite, une architecture de référence a été proposée pour la mise en place d'une méthode d'emplacement adaptative des services.

Cette méthode permet de déterminer l'emplacement optimal pour les services en temps réel en fonction des besoins des utilisateurs, en minimisant les temps de réponse.