REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Université Mohamed Khider – BISKRA
Faculté des Sciences Exactes, des Sciences de la Nature et de la Vie
Département d'informatique

N° d'ordre :……
Série :……..

Thèse

Présentée en vue de l'obtention du diplôme de docteur en sciences

Option : **Informatique**

(Intelligence Artificielle)

Titre

Composition adaptative de services pour l'Internet des objets

Présentée et soutenue publiquement par
AOUDIA Idir

Soutenue le : 23/10/2022

Devant le jury :

REZEG Khaled	Professeur	Université de Biskra	Président
BENHARZALLAH Saber	Professeur	Université de Batna 2	Rapporteur
KAZAR Okba	Professeur	Université de Biskra	Co-Rapporteur
DJEFFAL Abdelhamid	Professeur	Université de Biskra	Examinateur
ARAR Chafik	MCA	Université de Batna 2	Examinateur
KAHLOUL Laid	Professeur	Université de Biskra	Invité

Année universitaire : 2021 – 2022

Dédicace

À mes parents ;

À toute ma famille ;

À tous ceux qui me sont chers.

Remerciement

En premier lieu, je remercie le bon Dieu de m'avoir donné la force, la santé, la volonté et la patience nécessaire pour achever ce travail de thèse.

Je tiens à remercier Pr. BENHARZALLAH Saber et Pr. KAZAR Okba, mes directeurs de thèse, pour tout le soutien, l'aide et l'orientation qu'ils m'ont apporté tout au long de ce travail de recherche ainsi que pour la patience et le temps inconditionnel qu'ils m'ont consacré malgré leurs nombreuses charges.

Je remercie le Pr. REZEG Khaled professeur d'université de Biskra, pour m'avoir fait l'honneur de présider mon jury de thèse.

J'exprime tous mes remerciements à l'ensemble des membres de mon jury ; Pr DJEFFAL Abdelhamid professeur d'université de Biskra, Dr ARAR Chafik maitre de conférences (A) à l'université de Batna 2 de m'avoir fait l'honneur d'accepter de participer à mon jury de thèse.

Je remercie aussi Pr. KAHLOUL Laid professeur d'université de Biskra, pour ses précieux conseils et de m'avoir fait l'honneur d'assister à ma soutenance.

J'adresse toute ma gratitude à tous mes ami(e)s (Mostefa, Hicham, Houssam, Houcine et Saddam) et à toutes les personnes qui m'ont aidé dans la réalisation de ce travail, particulièrement à toi Rahma pour tout ton soutien indéfectible et toutes tes attentions au quotidien qui m'ont donné le courage de continuer.

Une très grande pensée à toi aussi Salsabil, pour tout ton soutien et l'énergie que tu m'as transmis jusqu'à la fin.

Mes derniers remerciements iront évidemment à ma famille. Je pense tout d'abord à ma maman sans qui l'enfant que j'étais ne serait pas devenu l'homme que je suis, à mon frère Hamimi et mes deux sœurs Djohra et Imen.

Pensée à mon père, que Dieu te couvre de sa miséricorde, et t'accueil dans son immense Paradis.

Enfin, je remercie tous ceux qui m'ont soutenu, encouragé et m'ont donné l'envie de mener à terme ce travail.

Résumé

L'internet des objets (IoT) est une technologie émergente, qui représente l'intégration ou la fusion de l'espace d'information et de l'espace physique. Au fil du temps, l'IoT est devenu de plus en plus populaire dans plusieurs endroits. Afin de répondre à la demande compliquée des utilisateurs, la plupart des appareils IoT ne fonctionnent pas seuls, une composition de services multiples doit être effectuée et elle est définie comme la composition de services. Pour des raisons de conductivités, pannes, batterie, charge et autres, la disponibilité des services IoT est imprévisible. Cette imprévisibilité de la disponibilité et l'évolution dynamique des besoins des utilisateurs, font que la composition du service doit gérer cette dynamique et s'adapter à de nouvelles configurations non prévues à la conception. La composition adaptative des services consiste à modifier le système pour lui permettre de se comporter correctement dans différents contextes afin d'assurer la disponibilité des services offerts, afin de répondre à une situation non prévue lors de la phase de conception. De ce fait, notre objectif est de proposer une méthode de composition de services IoT adaptative et sensible au contexte afin de satisfaire les besoins des utilisateurs.

Dans notre travail, nous considérons que la croissance de l'Internet des Objets (IoT) implique la disponibilité d'un très grand nombre de services qui peuvent être similaires ou identiques, la gestion de la Qualité de Service (QoS) permet de différencier un service d'un autre. La composition de services offre la possibilité d'effectuer des activités complexes en combinant les fonctionnalités de plusieurs services au sein d'un seul processus. Très peu de travaux ont présenté une solution de composition de services adaptative gérant les attributs de QoS, en plus dans le domaine de la santé, qui est l'un des plus difficiles et délicats car il concerne la précieuse vie humaine. Dans cette thèse, nous présenterons une approche de composition de services adaptative sensible aux QoS basée sur un algorithme génétique multi-population dans un environnement Fog-IoT. Notre algorithme P-MPGA implémente une méthode de sélection intelligente qui nous permet de sélectionner le bon service. En outre, P-MPGA implémente un système de surveillance qui surveille les services pour gérer le changement dynamique des environnements IoT. Les résultats expérimentaux montrent les excellents résultats du P-MPGA en termes de temps d'exécution, de valeurs de fitness moyennes et de rapport temps d'exécution / meilleure valeur de fitness malgré l'augmentation de la population. P-MPGA peut rapidement obtenir un service composite satisfaisant les besoins de QoS de l'utilisateur, ce qui le rend adapté à un environnement IoT à grande échelle.

Mots-clés : Internet des objets, Composition de services, Adaptabilité, Contexte, Qualité de Service, Fog-IoT computing, Healthcare.

Abstract

The Internet of Things (IoT) is an emerging technology, which represents the integration or fusion of information space and physical space. Over time, the IoT has become increasingly popular in several places. to answer the complicated user request, most IoT devices do not work alone, a multiple service composition must be made, and it is defined as the services composition. For reasons of conductivities, failures, battery, charging and others, the availability of IoT services is unpredictable. This unpredictability of availability and the dynamic evolution of user needs mean that the composition of the service must manage this dynamic and adapt to new configurations not planned in the design phase. The adaptive composition of services consists in modifying the system to allow it to behave correctly in different contexts in order to ensure the availability of the services offered, in order to respond to a situation not foreseen during the design phase. Therefore, our goal is to provide an adaptive IoT and context-aware service composition method to meet user needs.

In our work, we consider that the growth of Internet of Thing (IoT) implies the availability of a very large number of services which may be similar or the same, managing the Quality of Service (QoS) helps to differentiate one service from another. The service composition provides the ability to perform complex activities by combining the functionality of several services within a single process. Very few works have presented an adaptive service composition solution managing QoS attributes, moreover in the field of healthcare, which is one of the most difficult and delicate as it concerns the precious human life. In our thesis, we will present an adaptive QoS-Aware Service Composition Approach (P-MPGA) based on multi-population genetic algorithm in Fog-IoT healthcare environment. Our P-MPGA algorithm implements a smart selection method which allows us to select the right service. Also, P-MPGA implements a monitoring system that monitors services to manage dynamic change of IoT environments. Experimental results show the excellent results of P-MPGA in terms of execution time, average fitness values and execution time / best fitness value ratio despite the increase in population. P-MPGA can quickly achieve a composite service satisfying user's QoS needs, which makes it suitable for a large scale IoT environment.

Keywords: Internet of things, Service composition, Adaptability, Context, Quality of Service, Fog-IoT computing, Healthcare.

ملخص

إنترنت الأشياء هي تقنية ناشئة تمثل تكامل أو اندماج مساحة المعلومات والفضاء المادي. بمرور الوقت، أصبحت إنترنت الأشياء ذات شعبية متزايدة في عدة أماكن. للإجابة على طلبات المستخدم المعقدة، لا تعمل معظم أجهزة إنترنت الأشياء بمفردها، يجب تكوين خدمة متعددة، ويتم تعريفها على أنها تكوين الخدمات. لأسباب تتعلق بالتوصيلات والبطارية والشحن وغيرها، لا يمكن التنبؤ بتوفر خدمات إنترنت الأشياء. يعني عدم القدرة على التنبؤ بالتوافر والتطور الديناميكي لاحتياجات المستخدم أن تكوين الخدمات يجب أن يدير هذه الديناميكية وأن يتكيف مع التكوينات الجديدة التي لم يتم التخطيط لها في مرحلة التصميم. يتكون التكوين التكيفي للخدمات من تعديل النظام للسماح له بالتصرف بشكل صحيح في سياقات مختلفة، من أجل ضمان توافر الخدمات المقدمة، من أجل الاستجابة لحالة لم تكن متوقعة خلال مرحلة التصميم. لذلك يتمثل هدفنا في توفير طريقة تكوين الخدمات متكيفة لإنترنت الأشياء ومراعية للسياق لتلبية احتياجات المستخدم.

في عملنا، نعتبر أن نمو إنترنت الأشياء يعني توفر عدد كبير جدًا من الخدمات التي قد تكون متشابهة، تساعد إدارة جودة الخدمات على التمييز بين خدمة وأخرى. يوفر تكوين الخدمات القدرة على أداء الأنشطة المعقدة من خلال الجمع بين وظائف العديد من الخدمات في عملية واحدة. قدم عدد قليل جدًا من الأعمال حلاً متكيفًا لتكوين الخدمات يدير جودة الخدمات، خاصة في مجال الرعاية الصحية، وهو أحد أكثر المجالات صعوبة وحساسية من حيث صلته بالحياة البشرية الثمينة. في أطروحتنا، سوف نقدم نهج تكوين الخدمات المدرك لجودة الخدمات التكيفية (P-MPGA) بناءً على خوارزمية وراثية متعددة السكان في بيئة الرعاية الصحية الضبابية. تطبق الخوارزمية الخاصة بنا طريقة اختيار ذكية تسمح لنا باختيار الخدمات المناسبة. أيضًا، ينفذ نظام مراقبة يراقب الخدمات لإدارة التغيير الديناميكي لبيئات إنترنت الأشياء. تظهر النتائج التجريبية النتائج الممتازة للخوارزمية من حيث وقت التنفيذ، ومتوسط قيم اللياقة ووقت التنفيذ / أفضل نسبة قيمة لياقة على الرغم من الزيادة في عدد السكان. يمكن أن تحقق الخوارزمية الخاصة بنا بسرعة خدمة مركبة تلبي احتياجات جودة الخدمات للمستخدم، مما يجعلها مناسبة لبيئة إنترنت الأشياء على نطاق واسع

الكلمات المفتاحية: انترنت الأشياء، تكوين الخدمات، القدرة على التكيف، السياق، جودة الخدمات، حوسبة الضباب بإنترنت الأشياء، الرعاية الصحية

Liste des publications

1. Revues internationales

Aoudia Idir, Benharzallah Saber, Kahloul Laid, Kazar Okba, "A Multi-Population Genetic Algorithm for Adaptive Qos-Aware Service Composition in Fog-Iot Healthcare Environment", International Arab Journal of Information Technology (IAJIT), vol. 18, No. 3A, Special Issue 2021. (Le journal est de classe A).

Aoudia Idir, Benharzallah Saber, Kahloul Laid, Kazar Okba, "Service composition approaches for internet of things: a review", Int. J. Communication Networks and Distributed Systems, vol. 23, No. 2, 2019 (Le journal est de classe B).

2. Conférences Internationales

Aoudia Idir, Benharzallah Saber, Kahloul Laid, Kazar Okba, "QoS-aware service composition in Fog-IoT computing using multi-population genetic algorithm", 21th International Arab Conference on Information Technology (ACIT'2020), Giza, Egypt, indexed in IEEE Explore.

Aoudia Idir, Benharzallah Saber, Kahloul Laid, Kazar Okba, "A comparative analysis of IoT service composition approaches",18th International Arab Conference on Information Technology (ACIT'2017), Yassmine Hammamet, Tunisie, indexed in IEEE Explore.

3. Autres

Participation et présentation orale dans Journées d'Etude Informatique Théorique et Appliquée (JDITA'2018), 2018, Université Mohamed Khider, Biskra, Algeria.

Participation dans LINFI Doctoral Day (JDL'2017), November 2017, Université Mohamed Khider, Biskra, Algeria.

Table des matières

Liste des figures

Liste des tableaux

Liste des algorithmes

Abréviations

IoT	Internet of Things	L’internet des objets
QoS	Quality of service	Qualité de service
GA	Genetic Algorithm	Algorithme Génétiques
ACO	ant colony optimization	Algorithmes de colonies de fourmis
PSO	Particle Swarm Optimization	Optimisation de l'essaim de particules
XML	Extensible Markup Language	Langage de balisage extensible
WSDL	Web Services Description Language	Langage de description des services web
SOA	Service Oriented architecture	Architecture orientée service
RFID	Radio-Frequency Identification	Radio-identification
NFC	Near Field communication	Communication en champ proche
M2M	Machine to Machine	Machine à Machine
ISBN	International Standard Book Number	Numéro international normalisé du livre
CPU	Central Processing Unit	Processeur
WSN	Wireless Sensor Network	Réseau de capteurs sans fil
P2P	Peer-to-Peer	Pair-à-pair
ECG	Electro Cardio Graph	Électrocardiographie
ALS	Assistance living system	Système d'assistance à la vie
IA	Artificial intelligence	Intelligence Artificielle
W3C	World Wide Web consortium	Organisme de standardisation
MAMD	Multi-attribute making decision	Prise de décision multi-attributs
OWL	Web Ontology Language	Langage d'ontologie Web
MDP	Markov decision process	Processus décisionnel de Markov
BPEL	Business Process Execution Language	Langage de programmation destiné à l'exécution des procédures d'entreprise
MCGP	Model of multi-criteria goals programming	Modèle de programmation d'objectifs multicritères
QSC	QoS Service composition	Composition des services orientés QoS
PT	Paper-Tape	
DCOP	Distributed constraint optimization problem	Formalisme du problème d'optimisation des contraintes distribuées
APF	Artificial potential fields	Champs potentiels artificiels
AmI	Ambient intelligence environments	Environnements d'intelligence ambiante

Première partie

Introduction et Analyse de l'État de l'art

CHAPITRE 1

Introduction

Sommaire

1. Présentation de la thèse

Les équipements électroniques envahissent progressivement l'univers de notre quotidien qu'on souhaiterait que ces équipements puissent intelligemment réagir à notre activité afin de nous assister dans nos activités de tous les jours. Des objets et des utilisateurs qui interagissent, des bâtiments qui guident leurs visiteurs et les renseignent, configurent leurs appareils électroniques portatifs, leur proposent des services contextualisés ou encore des villes intelligentes gérant de façon autonome leurs ressources, sont quelques-unes des promesses de l'internet des objets (IoT), les ordinateurs doivent être nombreux et interconnectés afin de disparaître de la conscience de l'utilisateur.

L'internet des objets (IoT) est une technologie émergente, qui peut modifier l'industrie, l'environnement, le domaine social et médical. L'IoT est une fusion entre l'espace d'information et l'espace physique, connue sous le nom d'interconnexion d'ordinateurs embarqués, de capteurs, d'appareils mobiles ou d'autres objets identifiables de manière unique, qui sont capables d'interagir les uns avec les autres et de coopérer avec l'environnement pour atteindre des objectifs communs, en exploitant l'infrastructure Internet existante. Avec le développement rapide des puces, des capteurs, des réseaux et des logiciels ces dernières années, l'IoT devient de plus en plus populaire dans plusieurs applications. Un exemple est la ville intelligente, qui est une extension des bâtiments intelligents qui sont eux-mêmes une extension de la maison intelligente, qui est l'une des applications les plus en vue dans la vie quotidienne des gens.

L'IoT consiste à connecter un grand nombre d'objets quotidiens à Internet, en leur donnant leur propre identité, leur permettant ainsi d'offrir des fonctionnalités et de collecter des informations sous la forme d'un service. Afin de répondre à la demande compliquée de l'utilisateur, la plupart des périphériques IoT ne fonctionnent pas seuls, une composition de service multiple doit être faite, et elle est définie comme la composition des services.

La composition des services consiste à combiner la fonctionnalité de plusieurs services au sein d'un seul processus afin de répondre à des demandes complexes qu'un seul service ne peut satisfaire [1]. La composition de services offre la possibilité d'effectuer des activités complexes à partir de services existants, et les services composés forment un nouveau service, qui peut être réutilisé dans une autre composition [2].

2. Exemple illustratif

À titre d'illustration, supposons que nous cherchions à développer une application multi-plateformes (Web, mobile, etc..) pour détecter et signaler une urgence hospitalière. Ainsi, il semble pertinent de bénéficier du concept de l'Internet of Things (IoT) pour récolter les données dans différents endroits géographiquement. Nous nous limitons, dans un premier temps, à la récolte les coordonnées de la personne à secourir et de la disponibilité des ambulances/hôpitaux à intervalles réguliers.

Dans notre exemple, l'information recherchée exige à la fois l'interconnexion et la communication entre l'ensemble des objets disponible installés à différents endroits. Ces objets sont connectés entre eux à travers l'Internet, chacun pouvant demander un service de l'autre, et pouvant à son tour lui offrir un service. De ce fait, il y a un échange de données entre les différents objets. Ainsi, un objet qui ne détient pas une information demandée par l'utilisateur (ou l'application) peut solliciter les autres objets avec lesquels il est connecté pour demander l'information voulue. Grâce à ce mécanisme, l'utilisateur et l'application peuvent avoir l'information demandée, même si le service interrogé ne détient pas cette information dans sa base de données. C'est ainsi que notre exemple d'urgence hospitalière participatif et collaboratif peut devenir une réalité.

3. Problématique

Pour des raisons de conductivités, pannes, batterie, charge et autres, la disponibilité des services IoT est imprévisible [3]. Cette imprévisibilité de la disponibilité et l'évolution dynamique des besoins des utilisateurs, font que la composition de services doit gérer cette dynamique et s'adapter à de nouvelles configurations non prévues à la conception. La composition adaptative des services consiste à modifier le système pour lui permettre de se comporter correctement dans différents contextes afin d'assurer la disponibilité des services offerts, afin de répondre à une situation non prévue lors de la phase de conception.

En bref, une composition adaptative devrait inclure les quatre objectifs suivants :

- Se rétablir des situations inattendues afin que l'application continue l'exécution prévue, ou au moins se termine dans un état cohérent, malgré l'occurrence d'un échec.
- Exploiter les nouvelles opportunités émergentes pour améliorer la qualité de la solution choisie à n'importe quelle étape de l'exécution.
- Prévenir les changements futurs et les fautes en prenant des mesures correctives tôt ; car une réaction tardive (c'est-à-dire après l'exécution de services défectueux ou détériorant la qualité) peut entraîner une incapacité à trouver une récupération appropriée à partir de ce point, ou une nouvelle solution sélectionnée de qualité inférieure à celle qui pourrait être obtenue en réagissant plus tôt aux changements.
- Garder les adaptations déclenchées transparentes pour l'utilisateur final sans temps d'arrêt, car une interruption des performances du service composite pourrait être hautement indésirable, en particulier dans les applications sensibles au temps.

L'IoT devient de plus en plus populaire dans plusieurs applications, cependant de nos jours, si l'on prend par exemple le secteur de la santé, près de 2 millions de personnes perdent leurs vies à travers le monde en raison de l'arriéré des services d'urgence sanitaire (problème de circulation, de localisation, etc.). L'adaptabilité jouera un rôle très important dans cet environnement IoT dynamique où un grand nombre de services volatils sont disponibles, elle offrira la possibilité à l'application (ou service composite) d'évoluer en permanence afin de répondre aux nouvelles contraintes contextuelles.

L'adaptation est importante dans la composition des services IoT, mais pour produire une solution optimale pour toute composition de service, une gestion de la qualité de service (QoS) doit être effectuée. Les attributs QoS nous permettent de faire la différence entre chaque service IoT, il est donc essentiel de les évaluer pour éviter une composition de services de mauvaise qualité.

De nombreuses architectures IoT différentes ont été proposées et l'architecture IoT-Cloud était la méthode la plus pratique, mais à mesure que le nombre d'appareils utilisant le cloud augmente, de plus en plus de problèmes apparaissent. Dans notre cas d'étude, la plupart des données doivent être traitées en temps réel, cela est dû au fait que la plupart des services n'ont pas assez de mémoire pour conserver toutes les données collectées ou qu'ils doivent prendre eux-mêmes la décision (faible unité de calcul).

Ces problèmes rencontrés par le Cloud-IoT computing ont conduit au développement de l'architecture Fog-IoT, le Fog représente un médiateur entre les services IoT et le cloud [4]. L'architecture Fog-IoT a été principalement introduite pour améliorer l'architecture des systèmes Cloud-IoT [5] et elle représente une décentralisation du Cloud-IoT pour surmonter les obstacles de l'architecture Cloud-IoT.

Compte tenu des facteurs tels que les attributs de QoS, la sélection des services IoT pour la composition des services est réduite à un problème d'optimisation multi-objectifs. L'optimisation consiste à utiliser la ressource de la manière la plus efficace. Cela signifie maximiser ou minimiser certains attributs via une fonction objective [6]. Des algorithmes heuristiques tels que les Algorithme Génétiques (GA), Les Algorithmes de colonies de fourmis (ACO) et l’Optimisation de l'essaim de particules (PSO) sont adoptés pour trouver la composition optimale du service IoT.

4. Objectifs et contributions

Actuellement, la communauté de recherche active sur la composition des services dans l'IoT est encore très fragmentée et la plupart se concentre sur des domaines d'application uniques ou des technologies uniques. Nous pensons que cette fragmentation est potentiellement néfaste pour le développement des technologies de composition de services de l'IoT. Aussi, il y a un manque d'études qui passent en revue les différents travaux de la composition des services dans l'IoT.

Dans cette thèse, nous visons à présenter une revue des approches existantes pour la composition des services dans l'IoT, y compris une description et une comparaison entre eux en tenant compte de certains critères. Ce travail aussi présente également une comparaison entre la

composition de services Web traditionnels et la composition de services IoT. L'objectif principal est de donner au lecteur l'occasion de comprendre ce qui a été fait (protocoles, algorithmes, solutions proposées) et ce qui reste encore à traiter, ce travail aussi représente un support pour les chercheurs pour se concentrer sur leurs efforts et à fournir des solutions durables dans ce domaine. Nous espérons que cette étude et synthèse intitulée « *Service composition approaches for internet of things : a review*» [2] qu'on a publié en 2019 dans le journal international « *Int. J. Communication Networks and Distributed Systems* », pourra aider à combler les communautés existantes, à encourager les collaborations croisées et garantir que les défis liés à la composition des services dans l'IoT seront mieux traités, afin que la recherche puisse être exploitée avec succès.

En résumé, les principales contributions de cet article [2] sont :

- L'identification des défis les plus importants liés à la composition de services dans l'internet des objets (IoT). Ces défis consistent à trouver la composition la plus optimale, gérer l'évolution dynamique de l'enivrement IoT et surveiller les appareils avec des ressources de surveillance.
- L'introduction de douze critères spécifiques, qui sont basés sur les défis fondamentaux de la composition des services. On peut citer : l'adaptation, l'optimisation, la performance, etc…
- Une étude comparative exhaustive entre les approches de composition de services IoT basée sur notre ensemble de critères bien ciblés et spécifiques.
- Examiner les tendances de la recherche et à suggérer des orientations futures en mettant l'accent sur la composition des services dans l'IoT.
- Un état de l'art actuel de la composition des services dans les IoT.

En se basant sur notre étude comparative et la synthèse présentée dans [2], où nous avons évalué les approches de composition de services les plus populaires dans l'IoT en se basant sur nos douze critères pertinents, nous avons conclu que très peu de travaux ont présenté une solution adaptative de composition de services IoT qui gère des attributs de QoS, en plus dans le domaine de la santé, qui est l'un des plus difficiles et délicats car ça concerne la précieuse vie humaine. Et cela nous a mené dans la deuxième partie de notre travail a identifié certains attributs de QoS associés aux composants IoT qui quantifient et analysent au mieux les services offerts par les fournisseurs de services IoT.

De nombreuses architectures différentes ont été proposées dans la littérature, on peut citer l'architecture IoT en trois couches [7]–[9], l'architecture en quatre couches [10] etc.., mais en raison du défi de l'IoT concernant la sécurité et la confidentialité, l'architecture à cinq couches [11], [12] a également été proposée. Nous considérons que l'architecture en cinq couches peut répondre aux exigences de l'IoT et satisfaire un maximum de critères. Néanmoins, dans une architecture à cinq couches IoT (appelée aussi IoT-Cloud), tous les traitements (Stockage, Traitement des données, Surveillance) se font dans le cloud. Cette 'architecture IoT-Cloud est une méthode pratique pour quelques appareils. Mais à mesure que le nombre d'appareils utilisant le cloud augmente, l'utilisation de sa bande passante, de sa latence augmente également, et cela implique également autres plusieurs problèmes et ces problèmes rencontrés ont conduit au développement du Fog-IoT pour surmonter au mieux ces obstacles [13].

Une architecture en cinq couches se compose d'une couche Business, Application, Traitement, Transport et Perception. Dans notre travail nous nous sommes concentrés sur la défragmentation de la couche Traitement en quatre sous-couches (sécurité, stockage, pré-traitement et surveillance) implémentées sur un système Fog-IoT.

La définition de cette bonne architecture Fog-IoT était un aspect important qui nous a permis de concevoir une approche de composition de service adaptative, centrée sur l'utilisateur, qui s'adapte aux différentes situations dans l'environnement IoT et surtout, cette approche peut gérer notre cas d'étude « l'urgence hospitalière ». Ainsi, l'apport majeur de cette approche, qui la rend novatrice, peut être résumé comme suit :

- Compte tenu de facteurs tels que les attributs de QoS, la sélection de services IoT pour la composition de services est réduite à un problème d'optimisation multi-objectifs, ainsi ; pour résoudre ce problème, un algorithme génétique parallèle multi-population (P-MPGA) est proposé.
- L'algorithme P-MPGA implémente une méthode de sélection intelligente qui nous permet de sélectionner le bon service.
- P-MPGA implémente également un système de surveillance qui surveille les services pour gérer le changement dynamique des environnements IoT.
- Enfin, une évaluation expérimentale est effectuée pour vérifier la robustesse du cadre proposé en le comparant à l'GA traditionnel [14] et à l'algorithme génétique (MGA) proposé dans [15].

Ce travail [16] intitulé « *A Multi-Population Genetic Algorithm for Adaptive Qos-Aware Service Composition in Fog-Iot Healthcare Environment* » a été publié en 2021 dans le journal international « *International Arab Journal of Information Technology (IAJIT)* ».

5. Structure de la thèse

Le manuscrit de la thèse est organisé en deux parties soit six chapitres. La première partie est consacrée à l'introduction et l'analyse de l'état de l'art. Dans la deuxième nous présenterons notre contribution et sa validation.

- **Chapitre 1**
  Le premier chapitre présente le cadre général et la problématique de la recherche. Ce chapitre identifie également les objectifs de la thèse et présente brièvement les principales contributions.

- **Chapitre 2**
  Le deuxième chapitre est consacré à la présentation de concepts de base et fondamentaux qui serviront à clarifier les termes utilisés tout au long du document. Aussi, nous parlerons des défis de l'IoT avec une comparaison entre la composition de services Web et celle de l'IoT.

- **Chapitre 3**
  Le troisième chapitre traite certaines approches proposées populaires qui peuvent être considérés pour la composition de services dans l'IoT. Puis dans sa deuxième partie, nous présenterons une étude comparative entre les approches déjà proposées. Nous avons commencé notre enquête en introduisant quelques critères spécifiques, qui sont basés sur les défis fondamentaux de la composition des services cités précédemment dans le chapitre 2.

- **Chapitre 4**
  Dans le quatrième chapitre, nous présentons notre architecture à cinq couches basées sur le concept Fog-IoT, le modèle QoS utilisé et notre algorithme génétique multi-population sensible aux QoS P-MPGA.

- **Chapitre 5**
  Dans le cinquième chapitre, nous évaluons notre algorithme génétique multi-populations (P-MPGA) (algorithme 1) en le comparant aux algorithmes génétiques traditionnels (GA) [14] et à l'algorithme génétique (MGA) proposés dans [15]. Nous exécutons P-MGA, GA et MGA 10 fois et utilisons les valeurs moyennes pour l'évaluation.

- **Conclusion générale**
  Nous terminons cette thèse par une conclusion générale et des quelques perspectives sur des travaux futurs.

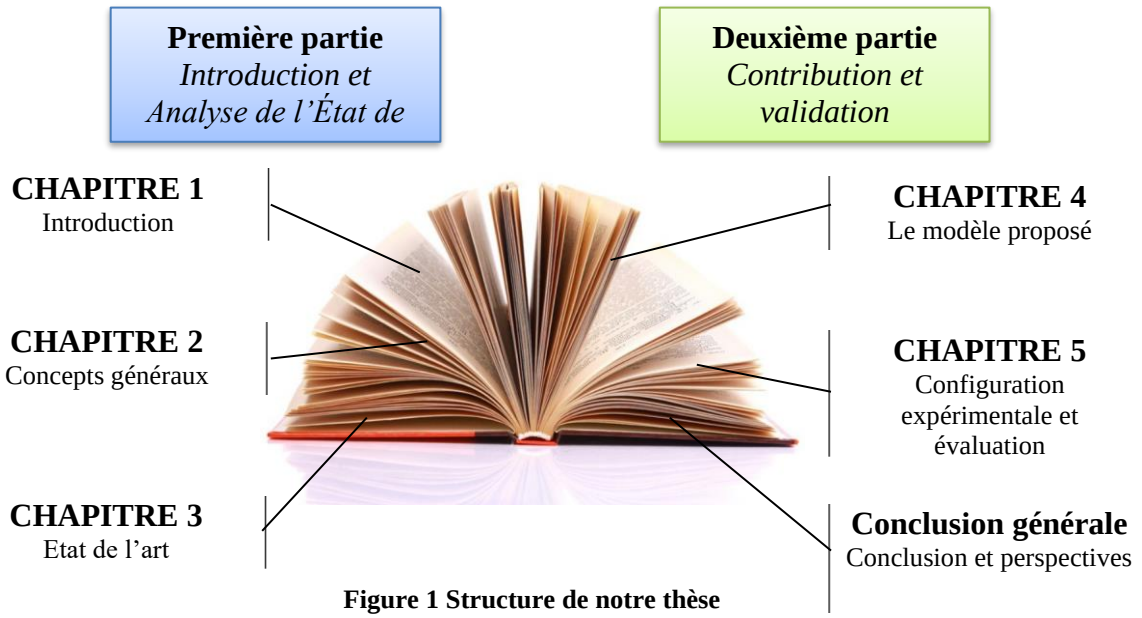


Figure 1 Structure de notre thèse

CHAPITRE 2

Concepts généraux

Sommaire

Dans ce chapitre nous présentons les différents concepts de base qui serviront à clarifier les termes utilisés tout au long du document. Cela aidera le lecteur à se familiariser plus rapidement avec les solutions proposées pour la réalisation de notre approche de composition de service adaptative Fog-IoT. Aussi, nous parlerons des défis de l'IoT avec une comparaison entre la composition de services Web et celle de l'IoT.

Le service Web (1er concept de base) représente la technologie permettant l'échange de données entre les différentes plateformes et leurs applications, bien que ces applications puissent être hétérogènes. L'échange de données nécessite l'utilisation d'un logiciel intermédiaire, Le Middleware (2e concept de base). Ce dernier est un logiciel qui offre une interface d'échange de données entre les objets dans l'Internet. Finalement, tout cela se passe dans l'Internet of Things ou IoT (3e concept de base). Ces trois concepts sont décrits brièvement dans les points suivants.

1. Service web

Étant donné que « les objets connectés ont un talon d'Achille : sans services Web associés, ils meurent ! » [17], dans le modèle [18], l'auteur recoure aux services Web pour contourner les problèmes de normes liés à l'interconnexion, ce qui permet un accès souple à des plateformes logicielles et matérielles via Internet.

Les services Web sont « des applications qui relient des programmes, des objets, des bases de données ou des processus d'affaires à l'aide de XML et des protocoles Internet standards. Les services Web sont des compléments aux programmes et applications existants,

développés à l'aide de langages tel que Visual Basic, C, C++, C# (C sharp), Java ou autre, et servent de pont pour que ces programmes communiquent entre eux » [19].

Plusieurs autres définitions des services Web peuvent être retenues. Nous pouvons citer celle de Abouzaid qui semble la plus simple : « Un service Web est un système logiciel qui permet de soutenir l'interaction entre les machines sur un réseau. Il dispose d'une interface écrite en format exploitable par les machines, spécifiquement le WSDL (Web Services Description Language) » [20]. De ce qui précède, on peut conclure que le service Web est un service électronique, offert par des technologies universelles développées en fonction des protocoles réseautiques mondiaux (essentiellement d'Internet) qui agit comme infrastructure de communication.

2. Architecture des services Web

Les services Web permettent, à travers un espace de communication, d'échanger différentes informations. Dans cette section, on tente de comprendre comment cet échange de données se fait en présentant un modèle d'architecture classique pour les services Web appelée SOA (Architecture orientée service).

La SOA « est une architecture d'application distribuée, de plateforme indépendante et à couplage faible, dans laquelle les différents services communiquent les uns avec les autres par des interfaces à définition simple et précise, et ne tient pas compte de l'interface de programmation sous-jacente et des modèles de communication. » [21]. Comme le montre la (Figure 2) ci-dessous, le modèle SOA met en interaction trois acteurs : un client, un fournisseur et un intermédiaire.

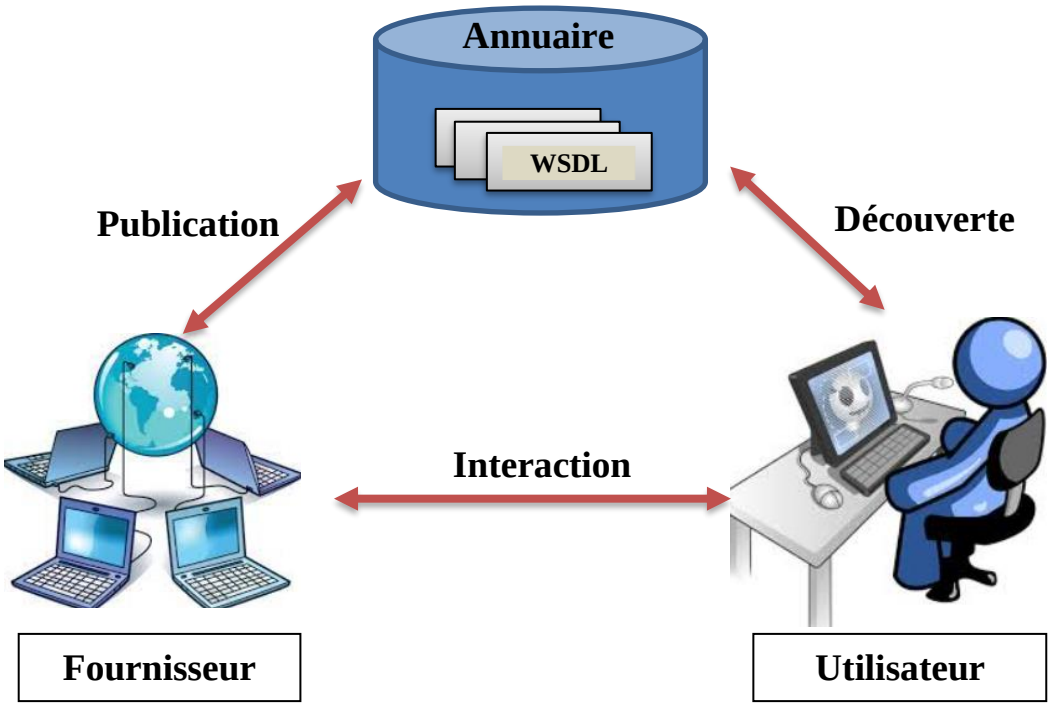


Figure 2 Architecture classique des services Web [22]

Le modèle SOA fait intervenir les trois acteurs suivants :

1) Le client, qui est le demandeur du service représenté par l'application, qui va chercher et consommer le service.
2) Le fournisseur, qui est le propriétaire du service offert dans la plateforme d'accueil
3) L'annuaire des services, un intermédiaire entre le client et le fournisseur, qui offre au fournisseur la possibilité de publier ses services, et au client la capacité de localiser les services répondant à ses besoins.

Le fonctionnement des services Web se base sur un modèle composé de trois phases fondamentales : la publication, la découverte et l'interaction [22].

3. Le Middleware (ou intergiciel)

Nombreux sont les travaux de recherche qui s'intéressent au Middleware. Dans cette section, nous mettons l'accent sur les éléments qui caractérisent ce concept et aussi sur l'intérêt de l'utiliser dans une application informatique.

Zeng définit le Middleware comme étant « une couche de logiciel ou un ensemble de sous-couches interposées entre la technologie et les niveaux d'application. Sa fonction, qui consiste à cacher les détails des différentes technologies, est fondamentale pour éliminer les publications non pertinentes. Le Middleware gagne de plus en plus d'importance en raison de son rôle majeur dans la simplification du développement de nouveaux services » [23].

« La définition du Middleware généralement admise est la suivante : un Middleware est un logiciel de communication qui permet aux processus exécutés sur une ou plusieurs machines d'interagir à travers un réseau. » [24].

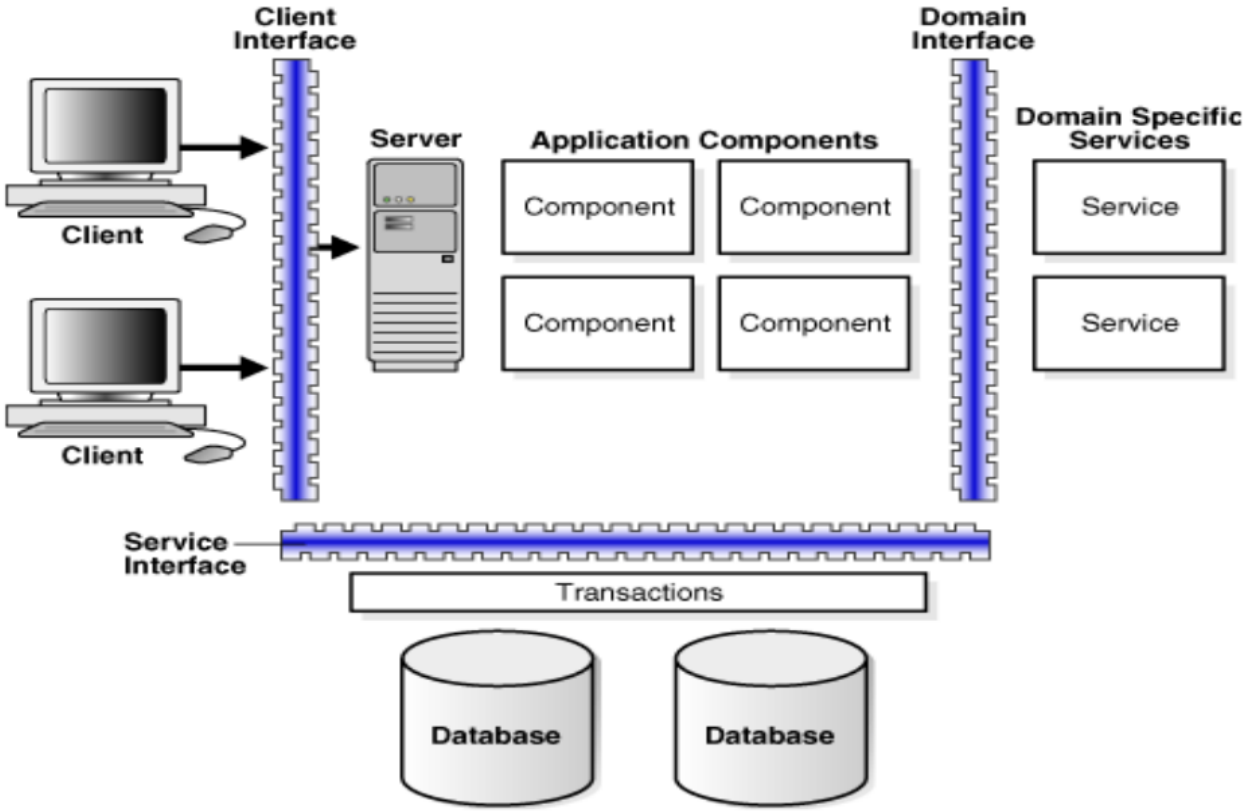


Figure 3 Architecture Middleware [25]

Comme le montre la Figure 3, Le Middleware (la partie en bleu) est « le logiciel qui connecte les composants logiciels ou les applications d'entreprise. C'est la couche logicielle qui se trouve entre le système d'exploitation et les applications de chaque côté d'un réseau d'ordinateurs distribué » [25].

Le Middleware est « une classe de logiciels permettant la communication entre des applications qui n'étaient pas conçues pour dialoguer entre elles. Il homogénéise l'accès à l'ensemble des services proposés par le réseau. Les Middlewares sont utilisés pour supporter des applications complexes et distribuées. Ils permettent aussi de lier deux ou plusieurs logiciels entre eux afin que les applications correspondantes puissent échanger des données. » [26].

« Le Middleware rend le développement d'une application plus facile en fournissant une abstraction de programmation commune, en masquant l'hétérogénéité et la distribution du matériel et du système d'exploitation, et en masquant les détails de programmation de haut niveau. » [25].

4. L'internet des objets (IoT)

L'Internet des objets (IoT) est une concrétisation technique de l'informatique ubiquitaire où la technologie est intégrée naturellement aux objets du quotidien. Très prometteur, ce concept ouvre la voie vers une multitude de scénarios basés sur l'interconnexion entre le monde physique et le monde virtuel : domotique, e-santé, ville intelligente, logistique, sécurité, etc. Cependant, comme d'autres concepts prometteurs, celui-ci fait face à un certain nombre de

problématiques techniques et non techniques qui nécessitent d'être étudiées pour permettre à l'Internet des objets d'atteindre son plein potentiel.

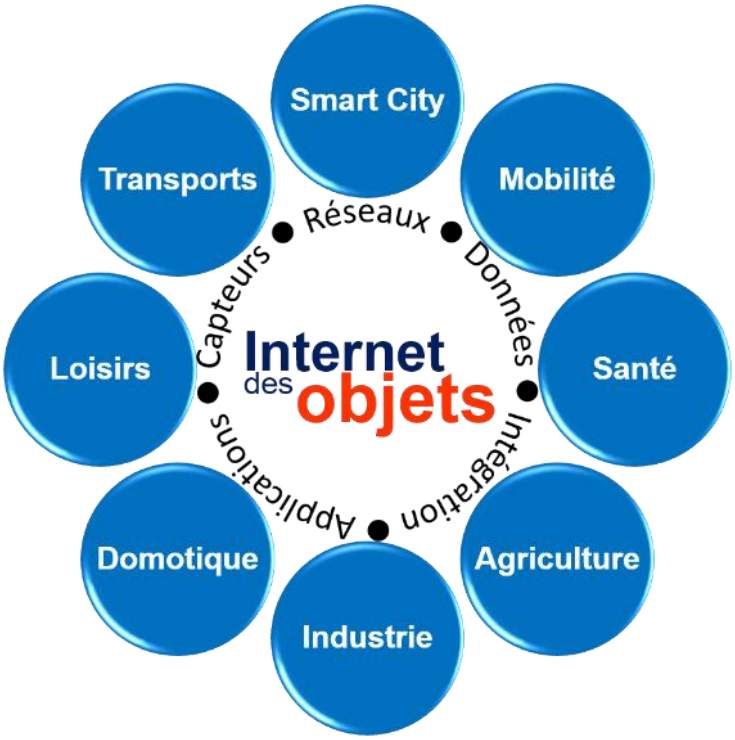


Figure 4 L'internet des objets (IoT) et ses nombreux domaines

L'Internet des objets est un concept concrétisant la vision de l'informatique ubiquitaire telle qu'imaginée en 1991 par Mark Weiser [27], où la technologie s'efface peu à peu dans l'environnement des utilisateurs, intégrée naturellement à l'intérieur des objets du quotidien[1]. La technologie n'est plus alors représentée par un objet unique, l'ordinateur personnel, mais se présente au contraire sous la forme d'appareils spécialisés et simples d'emploi, capables de communiquer au travers de plusieurs types de réseaux sans fil : liseuses numériques, télévisions et montres connectées, ordinateurs de bord, téléphones intelligents, etc.

À l'origine, le terme Internet des objets a été utilisé pour la première fois en 1999 par Kevin Ashton [28] pour décrire des objets équipés de puces d'identification par radiofréquence (ou puce RFID). Chaque objet, identifié de manière unique et universelle, peut alors être rattaché à un ensemble d'informations le concernant, ces dernières étant lisibles par d'autres machines[2]. Caractéristiques, état courant et position sont alors autant de métadonnées échangées entre les objets, formant un nouveau réseau qui leur est dédié : l'Internet des objets.

Le concept a toutefois évolué avec le temps et s'est généralisé vers une approche consistant à connecter un très grand nombre d'objets du quotidien au réseau Internet, les dotant ainsi d'une identité propre et leur permettant, entre autres, d'offrir des services et de collecter des informations de manière autonome.

[1] Mark Weiser : « The most profound technologies are those that disappear. They weave themselves into the fabric of everyday life until they are indistinguishable from it. »
[2] Kevin Ashton : « RFID and sensor technology enable computers to observe, identify and understand the world. »

L’objectif ambitieux derrière cette interconnexion est double, consistant en premier lieu dans la mise en place d’une infrastructure de communication machine à machine (M2M) à grande échelle de façon à permettre à ces machines de mieux « percevoir » le monde qui les entoure [29]. En second lieu réside une volonté d’offrir les abstractions nécessaires aux êtres humains pour interagir avec ces machines, et par extension avec le monde physique, aussi simplement qu’avec le monde virtuel que nous connaissons aujourd’hui [30]. En d’autres termes, les utilisateurs de l’Internet des objets (IoT) devraient être en mesure de manipuler l’environnement physique de la même façon qu’ils manipulent aujourd’hui des abstractions de haut niveau, telles que les fichiers et les dossiers, les pages Web et les hyperliens, ou encore les profils et les relations (p. ex. sur les réseaux sociaux).

Dans le futur Internet, les objets sont connectés et intégrés de manière transparente entre eux et avec l'environnement physique environnant selon la notion informatique omniprésente, fournissant un accès et un support de communication à tout moment/n'importe où/n'importe quel média. La façon dont ces différentes solutions et technologies coexistent conduit au paradigme de l'IoT, qui est le résultat de la convergence de trois principales visions différentes [31] : les perspectives Internet, Sémantique et Objets. L'intégration de telles approches dans le cadre de l'Internet des objets est illustrée à la Figure 5.

Jusqu’à présent, le nouveau paradigme qu’est l’Internet des objets (IoT) émerge des travaux issus de plusieurs communautés scientifiques : les réseaux de capteurs et d’actionneurs sans fil, le Web, le cloud computing, l’identification par radiofréquence (Radio-Frequency IDentification, ou RFID) ou encore la communication en champ proche (Near Field communication, ou NFC). Cette grande diversité, couplée à la popularité croissante du concept auprès des mondes académique et industriel, fait que, s’il existe effectivement des objectifs communs, l’Internet des objets n’est pas clairement défini, comme le montre le Tableau 1 qui recense quelques définitions récentes extraites de la littérature. En combinant ces définitions, on retrouve toutefois les aspects que nous avons évoqués plus haut : identité, autonomie, interconnexion et interaction avec l’environnement (capteurs et actionneurs).

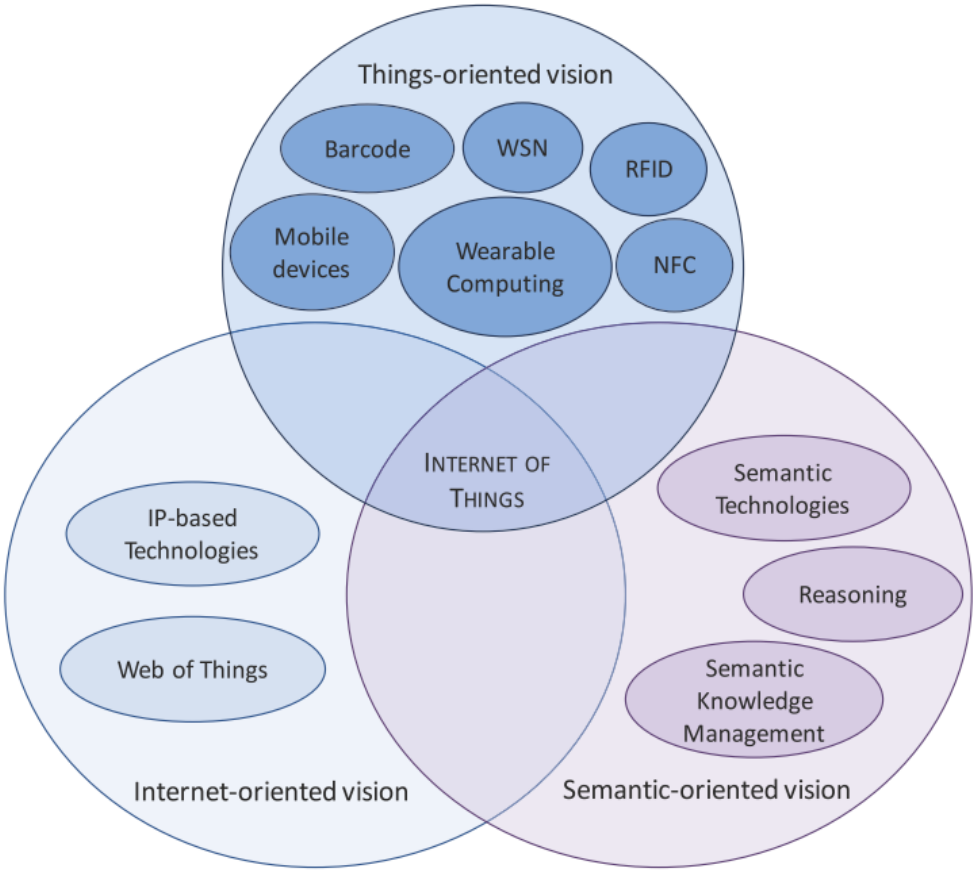


Figure 5 Perspectives du paradigme de l'internet des objets [31]

Tableau 1 Quelques définitions de l'Internet des objets (IoT)

« Un réseau qui relie et combine les objets avec l'Internet, en suivant les protocoles qui assurent leurs communications et échange d'informations à travers une variété de dispositifs. » [32].
« Un réseau de réseaux qui permet, via des systèmes d'identification électroniques normalisés et unifiés, et des dispositifs mobiles sans fil, d'identifier directement et sans ambiguïté des entités numériques et des objets physiques et ainsi, de pouvoir récupérer, stocker, transférer et traiter les données sans discontinuité entre les mondes physiques et virtuels. » [33].
« Une extension de l'Internet actuel envers tout objet pouvant communiquer de manière directe ou indirecte avec des équipements électroniques eux-mêmes connectés à l'Internet. Cette nouvelle dimension de l'Internet s'accompagne avec de forts enjeux technologiques, économiques et sociaux, notamment avec les économies majeures qui pourraient être réalisées par l'ajout de technologies qui favorisent la standardisation de ce nouveau domaine, surtout en matière de communication, tout en assurant la protection des droits et des libertés individuelles » [34].

Le terme « Internet des objets » est utilisé comme mot-clé générique pour couvrir divers aspects liés à l'extension d'Internet et du Web dans le domaine physique, au moyen du déploiement généralisé de dispositifs répartis dans l'espace avec identification, détection et/ou capacités d'actionnement. L'Internet des objets envisage un avenir dans lequel les entités numériques et physiques peuvent être liées, au moyen de technologies de l'information et de la communication appropriées, pour permettre une toute nouvelle classe d'applications et de services [30].

« Internet des objets » signifie « un réseau mondial d'objets interconnectés adressables de manière unique, basé sur des protocoles de communication standard », ou, plus largement : « des objets ayant des identités et des personnalités virtuelles opérant dans des espaces intelligents utilisant des interfaces intelligentes pour se connecter et communiquer au contextes sociaux, environnementaux et utilisateurs" [35].

Une infrastructure de réseau mondiale, reliant des objets physiques et virtuels grâce à l'exploitation de capacités de capture de données et de communication. Cette infrastructure comprend les développements Internet et réseau existants et évolutifs. Il offrira des capacités spécifiques d'identification d'objets, de capteurs et de connexion comme base pour le développement de services et d'applications coopératifs indépendants. Ceux-ci seront caractérisés par un degré élevé de capture de données autonome, de transfert d'événements, de connectivité réseau et d'interopérabilité [36].

L'idée de base de ce concept est la présence omniprésente autour de nous d'une variété de choses ou d'objets - tels que des étiquettes d'identification par radiofréquence (RFID), des capteurs, des actionneurs, des téléphones portables, etc. - qui, grâce à des schémas d'adressage uniques, sont capables interagir les uns avec les autres et coopérer avec leurs voisins pour atteindre des objectifs communs [31].

---

5. Qu’est-ce qu’un objet ?

Un objet est, avant toute chose, une entité physique ; par exemple, un livre, une voiture, une machine à café électrique ou un téléphone mobile. Dans le contexte précis de l’Internet des objets (IoT), et ce quelle que soit la vision, cet objet possède au minimum un identifiant unique attaché à une identité [31] exprimant d’une part ses propriétés immuables (type, couleur, poids, etc.) et son état c’est-à-dire l’ensemble de ses caractéristiques pouvant évoluer au cours du temps (position, niveau de batterie, etc.).

De nombreuses définitions s’accordent à dire qu’un objet possède des capacités de calcul, d’acquisition (capteur) et d’action (actionneur) [30], [36], [37], cependant cette définition exclut les objets inertes identifiés par RFID. En effet, une puce RFID classique ne peut pas être considérée comme un dispositif de calcul, celle-ci se résumant à un identifiant stocké dans une mémoire. De même, si l’on considère les cas extrêmes, un simple code-barre peut être employé pour identifier un objet, ce dernier étant exempt d’une quelconque partie électronique ; un livre et son code ISBN, par exemple. Ainsi, il est raisonnable de considérer que l’Internet des objets

est composé d'objets actifs, capables d'accomplir des calculs, d'effectuer des mesures sur l'environnement ou d'influer sur celui-ci, et d'objets passifs qui n'ont pas d'autres aptitudes que celles d'être suivis et détectés par des objets actifs. Par extension, l'identité d'un objet passif n'est pas directement stockée dans celui-ci, à l'exception de l'identifiant, et nécessite l'utilisation d'une infrastructure tierce capable de stocker ces informations. Au contraire, un objet actif peut stocker tout ou partie de son identité et échanger directement ces informations avec d'autres objets actifs. Cependant, cette capacité à stocker sa propre identité n'est pas obligatoire pour un objet actif et dépend (i) de ses ressources matérielles, notamment la mémoire, et (ii) de la complexité et du volume de ladite identité.

De manière générale, les ressources matérielles des objets actifs ont une importance cruciale en cela qu'elles définissent quels traitements pourront être effectués par l'objet. Par exemple, un algorithme de reconnaissance faciale nécessite plus de ressources qu'un algorithme de recherche d'éléments dans un ensemble ordonné, ce dernier nécessitant lui-même plus de ressources qu'une simple opération arithmétique telle que l'addition. Par ailleurs, les systèmes embarqués offrent souvent un ensemble d'opérations spécifiques implémentées directement au niveau du matériel, notamment sous la forme de circuits logiques programmables. Ainsi, même des objets aux ressources CPU et mémoires limités peuvent être dotés d'un module matériel de sécurité (hardware security module) ou d'une puce d'encodage ou de décodage vidéo. De la même façon, les objets actifs sont aussi des objets communicants et, de fait, embarquent diverses interfaces leur permettant d'échanger des informations sur divers réseaux filaires ou sans fil.

Au-delà des ressources matérielles, les objets physiques possèdent des caractéristiques variées dues à leurs usages. Certains objets sont mobiles et, de fait, caractérisés par une position qui évolue au cours du temps. Il peut s'agir d'objets transportables, tels qu'un téléphone mobile, un livre ou des vêtements, ou d'objets mobiles autonomes dotés de capacités motrices, tels qu'une voiture, un drone ou un animal domestique[3]. À l'inverse, de nombreux objets sont fixes la plupart du temps : réfrigérateurs, meubles, compteurs électriques, etc. De la même façon, certains objets sont alimentés en continu depuis une source permanente d'énergie électrique (typiquement, les appareils électroménagers) tandis que d'autres sont alimentés par une batterie et possèdent donc une durée de vie limitée en fonction de celle-ci. Ce facteur a une influence variable selon si l'objet est facilement rechargeable, comme c'est le cas avec un téléphone mobile ou un baladeur numérique, ou au contraire abandonné sur le terrain, par exemple dans le cadre d'études des mouvements migratoires d'une population animale. De la même façon, certains objets sont en mesure d'extraire de l'énergie depuis leur environnement (energy harvesting) par exemple en tirant parti de l'énergie solaire ou d'un champ électromagnétique. En outre, certains appareils dotés de capacités motrices sont capables de rechercher des points de rechargement et de s'y rendre sans intervention humaine. C'est notamment le cas des aspirateurs robots.

6. Fonctionnement de l'IoT L'Internet

L'Internet of Things (IoT) permet l'interconnexion des différents objets intelligents via l'Internet. Ainsi, pour son fonctionnement, plusieurs systèmes technologiques sont nécessaires. Citons quelques exemples de ces technologies.

[3] Un animal est un objet au sens restreint d'ensemble de caractéristiques mesurables et évoluant au cours du temps.

« L'IoT désigne diverses solutions techniques (RFID, TCP/IP, technologies mobiles, etc.) qui permettent d'identifier des objets, de capter, stocker, traiter, et transférer des données dans les environnements physiques, mais aussi entre des contextes physiques et des univers virtuels. » [33].

En effet, bien qu'il existe plusieurs technologies utilisées dans le fonctionnement de l'IoT, nous mettons l'accent seulement sur quelques-unes qui sont selon la littérature les technologies clés de l'IoT. Ces technologies sont les suivantes : RFID, WSN et M2M, et sont définies ci-dessous.

- ***RFID*** (Radio Frequency Identification) : le terme RFID englobe toutes les technologies qui utilisent les ondes radio pour identifier automatiquement des objets ou des personnes. C'est une technologie qui permet de mémoriser et de récupérer des informations à distance grâce à une étiquette qui émet des ondes radio [38]. Il s'agit d'une méthode utilisée pour transférer les données des étiquettes à des objets, ou pour identifier les objets à distance. L'étiquette contient des informations stockées électroniquement pouvant être lues à distance [32].
- ***WSN*** (Wireless Sensor Network) : c'est un ensemble de nœuds qui communiquent sans fil et qui sont organisés en un réseau coopératif. Chaque nœud possède une capacité de traitement et peut contenir différents types de mémoires, un émetteur-récepteur RF et une source d'alimentation, comme il peut aussi tenir compte des divers capteurs et des actionneurs [39]. Comme son nom l'indique, le WSN constitue alors un réseau de capteurs sans fil qui peut être une technologie nécessaire au fonctionnement de l'IoT.
- ***M2M*** (Machine to Machine) : c'est « l'association des technologies de l'information et de la communication avec des objets intelligents dans le but de donner à ces derniers les moyens d'interagir sans intervention humaine avec le système d'information d'une organisation ou d'une entreprise » [40].

7. Domaines d'application de l'IoT

Nous constatons que le concept de l'Internet of Things (IoT) est en pleine explosion vu que nous avons de plus en plus besoin dans la vie quotidienne d'objets intelligents capables de rendre l'atteinte de nos objectifs plus facile. Ainsi, les domaines d'applications de l'IoT peuvent être variés.

Plusieurs domaines d'application sont touchés par l'IoT. Dans [41], les auteurs ont classé les applications en quatre domaines : le domaine personnel, le domaine du transport, l'environnement et les services publics.

Comme la Figure 6 ci-dessous le montre, on trouve alors l'IoT dans notre vie personnelle quotidienne et également dans les services publics offerts par le gouvernement.

Figure 6 Les domaines d'Internet of Things [41]

Nous pouvons affirmer que l'Internet peut être connecté à n'importe quel objet. Ainsi, les domaines d'applications de l'IoT sont multiples. On cite, à titre d'exemples, l'industrie, la santé, l'éducation et la recherche. Cependant, il sera possible dans le futur de trouver le concept de l'IoT n'importe où, n'importe quand et à la disposition de tout le monde.

Tableau 2 Quelques scénarios pour l'Internet des objets

Suivi et logistique	Rendre possible le suivi dans le temps et l'espace de n'importe quel objet, pour peu que celui-ci soit identifié (p. ex. avec une étiquette RFID). Ce scénario est aujourd'hui implémenté par les industriels pour le suivi des objets dans une chaîne de production (fabrication, transport, etc.) ou dans un stock, le suivi d'objets perdus ou volés, ou encore la classification de documents [31].
Soins médicaux et	Utiliser des capteurs pour obtenir des informations en temps réel sur l'état des patients (rythme cardiaque, pression sanguine, etc.), même

aide à la personne	lorsque ceux-ci sont en extérieur (capteurs portatifs), pour pouvoir réagir rapidement en cas d'anomalie [42].
Environnements intelligents	Autoréguler les paramètres de l'environnement dans les logements et les bureaux (température, luminosité, humidité, consommation énergétique, etc.) et contrôler l'accès à certaines ressources en fonction du contexte (p. ex. présence ou non d'un responsable). Dans le cadre des villes intelligentes, analyser en temps réel les différents paramètres (bruit, pollution, trafic routier) et connecter les différents éléments de l'infrastructure urbaine pour en améliorer la gestion [43].
Social	Utiliser les capteurs portatifs pour collecter un ensemble d'informations concernant, par exemple, la position, l'activité en cours (courir, faire du vélo, etc.), l'humeur ou l'environnement (bruyant, peuplé, pluie, etc.) d'un individu dans le but de les partager automatiquement avec des groupes sociaux [44]. Par extension, permettre aux utilisateurs et aux organisations de partager les informations de leurs systèmes à grande échelle pour contribuer collaborativement à de nombreuses tâches d'analyse [45] : trafic routier, flux de population, conditions climatiques, etc.

8. La composition de service dans l'IoT

Le concept d'objet étant posé, les différentes définitions présentées dans le Tableau 2 et la littérature que nous avons étudiée à ce sujet nous conduisent à conclure que l'Internet des objets (IoT) consiste à connecter un grand nombre d'objets du quotidien à internet, leur donnant leur propre identité, leur permettant, entre autres, d'offrir des fonctionnalités et de collecter des informations sous forme de service.

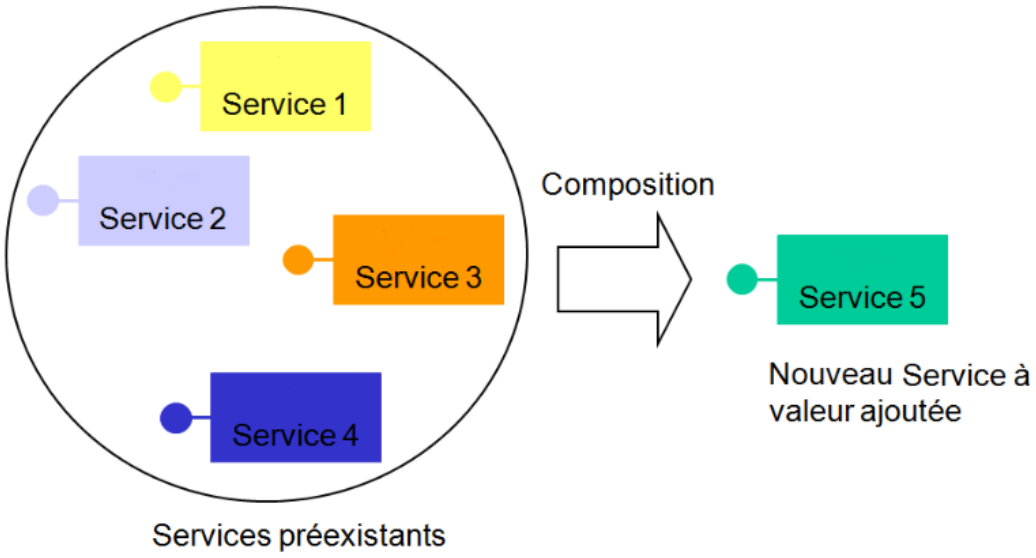


Figure 7 Composition de service

Afin de répondre à la demande compliquée de l'utilisateur, la plupart des périphériques IoT ne fonctionnent pas seuls, une composition de service multiple doit être faite (Figure 7), et elle est définie comme la composition des services. La composition des services consiste à combiner la fonctionnalité de plusieurs services au sein d'un seul processus afin de répondre à des demandes complexes qu'un seul service ne peut satisfaire [1]. La composition de service offre la possibilité d'effectuer des activités complexes à partir de services existants, et les services composés forment un nouveau service, qui peut être réutilisé dans une autre composition [2].

9. Catégories de composition de services

Ces dernières années, plusieurs communautés, aussi bien dans la recherche académique que dans l'industrie, se sont intéressées à la problématique de la composition de services. Cette problématique est à l'origine d'un nombre considérable de travaux notamment dans les communautés de l'intelligence artificielle [46], [47].

La (figure 8) présente notre classification des catégories de composition de services rencontrées dans la littérature. Cette classification se base sur trois axes principaux :

9.1. Degré d'automatisation

En fonction du degré de participation de l'utilisateur dans la définition du schéma de composition, trois types de composition de services sont distingués : compositions manuelles, semi-automatiques ou automatiques.

9.1.1. Composition manuelle

Les techniques de composition manuelles reposent sur un expert en charge de définir et générer le schéma de la composition de services. L'utilisateur joue un rôle central en définissant manuellement l'ordre d'invocation et d'exécution des services composants. Tout d'abord,

l'utilisateur définit la requête qui reflète ses besoins, ensuite le schéma de la composition de service qui repend à cette requête. Enfin, ce schéma de composition de services est soumis à un moteur d'exécution afin de le réaliser.

9.1.2. Composition semi-automatique

Dans les approches semi-automatiques, des outils graphiques sont développés afin de guider l'utilisateur étape par étape durant le processus de composition. Ces outils mettent à disposition un certain nombre d'opérateurs graphiques pour aider l'utilisateur voulant construire un service composite. Aussi, ces outils proposent à l'utilisateur des conseils et des suggestions quant à la sélection des services à composer. Une fois le service composite est défini, le schéma de composition est enfin soumis à un moteur d'exécution. Certains outils permettent également de mémoriser les schémas pour faciliter leurs réutilisations.

9.1.3. Composition automatique

Dans ce type de composition, l'utilisateur ne fait que spécifier ses besoins sous forme d'une requête de services. Par la suite, c'est le système qui prend en charge tout le processus de composition et le réalise automatiquement de manière tout à fait transparente, sans qu'aucune autre intervention de l'utilisateur ne soit requise.

9.2. Moment de la sélection des services

La réalisation concrète d'un schéma de composition de services est réalisée par la sélection de services qui consiste à identifier les services concrets à invoqués lors de l'exécution. Cette sélection peut être faite au moment de la conception, ou bien au moment de l'exécution, la composition est dite alors statique ou dynamique.

9.2.1. Composition statique

La composition statique est une composition qui est spécifiée aux moments de la conception et du déploiement de l'architecture du système. Les services à composer sont identifiés, sélectionnés, interconnectés, compilés et déployés par le concepteur. Cela donne lieu à des compositions figées, avec des services participants et des liaisons entre eux prédéfinis [48]. Ce type de composition n'est rentable que lorsque le système et les services sont peu évolutifs. Les approches statiques de composition de services sont en général adoptées par l'industrie.

9.2.2. Composition dynamique

Par opposition à la composition statique, la composition dynamique se caractérise par le fait que les services candidats sont localisés par le système à l'arrivé de la requête, puis reliés à la demande de l'utilisateur pour obtenir un service composite [48]. Les étapes de construction du service composite sont réalisées au moment de l'exécution en fonction des services disponibles à ce moment. Par conséquent, la composition dynamique de services est idéale pour répondre aux exigences des environnements dynamiques où les services disponibles changent constamment et les attentes des utilisateurs sont variables et personnalisées. Cependant sa mise en œuvre reste difficile à cause des changements fréquents de contextes des utilisateurs et des services dans de tels environnements.

9.3. L'exécution de la composition de services

L'exécution de la composition de services peut être soit confiée à une seule entité calcule ou bien partagée entre différentes entités de calcule, une composition est dite centralisée ou distribuée.

9.3.1. Composition centralisée

Cette catégorie nécessite un nœud central qui agit comme un coordinateur de composition de services et imposant de ce fait une structure centralisée. Ce nœud central, appelé aussi coordinateur central ou moteur d'exécution. Son rôle principal consiste à gérer l'invocation des différents services impliqués dans le schéma de composition de services.

9.3.2. Composition distribuée

Contrairement à la composition centralisée, la composition distribuée n'a pas de coordinateur central. Ce type de composition impose une structure décentralisée dans laquelle, tous les services participants à une composition de services doivent collaborer afin de réaliser cette composition. C'est ainsi que la tâche de réalisation d'un service composite est, cette fois-ci, répartie entre tous les services concernés.

9.3.3. Composition hybride

A défaut de considérer une structure purement centralisée ou purement distribuée, la composition hybride combine ces deux structures. Dans ce cas de figure, la topologie est partitionnée en plusieurs sous-ensembles. Chacun de ces ensembles possède son propre coordinateur local de composition de services. Les différents coordinateurs locaux sont, à leurs tours, rattachés à un autre coordinateur de niveau hiérarchique supérieur. [48].

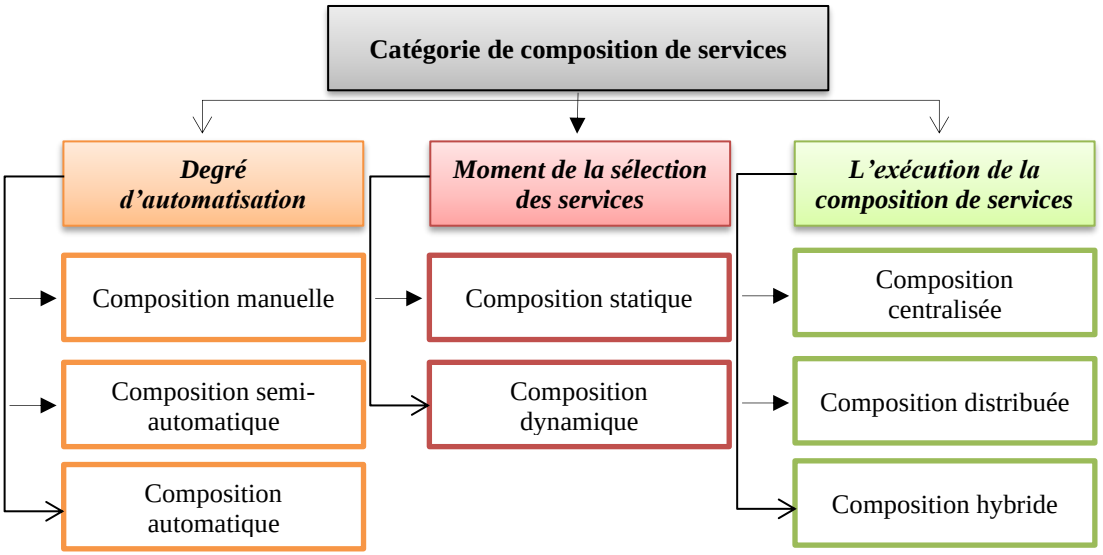


Figure 8 Catégories de composition de services

10. Les challenges de la composition de service dans l'internet des objets (IoT)

La composition des services dans l'IoT présente de nombreux défis, les services de l'IoT sont principalement déployés dans des appareils à ressources limitées avec une capacité de stockage et de traitement d'énergie limitée et qui se comparent aux services Internet déployés dans des ordinateurs dotés de ressources riches et de fonctions puissantes [49]. En outre, avec l'augmentation rapide du nombre d'appareils, la composition doit être conçue de centralisée à distribuée, et le coût de la composition du service dans l'IoT doit être pris en compte.

L'environnement IoT est dit dynamique, les informations du service IoT sont instables (son état bascule entre disponibilité/indisponibilité) par rapport aux services Internet [49]. De plus, en raison de l'indépendance de chaque entité, des problèmes d'interaction et de communication peuvent apparaître, en plus la probabilité d'erreurs telles que des pannes et des changements de service dans l'IoT est élevée. Un mécanisme de surveillance doit être utilisé pour garantir la tolérance aux pannes et surveiller le système IoT ; cependant, cela peut entraîner un coût élevé [50].

En raison de l'évolution dynamique des besoins des utilisateurs et du contexte, la composition doit être adaptative (Figure 9) et effectue une mise à jour en temps réel, pour assurer la continuité des services à l'utilisateur et améliorer l'interaction de l'utilisateur avec l'environnement.

Il convient au système de composition de services de l'IoT d'introduire une stratégie de paramètres non fonctionnels comme les attributs de qualité de service (QoS) pour choisir et composer des services selon leurs informations sensorielles, parce que la composition de service IoT vise à fournir à l'utilisateur des informations sensorielles [49]. Contrairement à la composition des services Internet, qui vise à résoudre les problèmes de programmation, se concentre sur la démonstration de la faisabilité et de la logique.

Un système IoT connecte le monde physique dans le cyberespace [51], ainsi l'un des défis du système de composition de services de l'IoT est la gestion d'un système de protection et de sécurité pour résister aux attaques malveillantes afin de survivre dans des environnements hostiles.

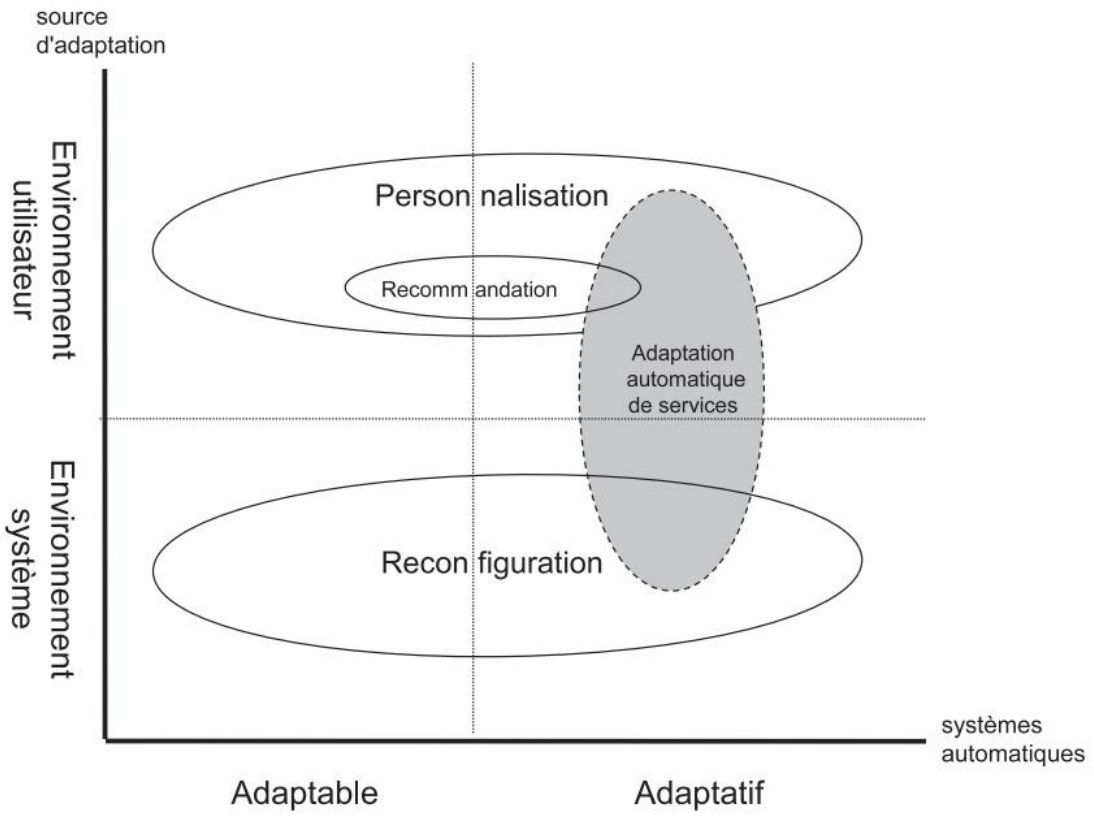


Figure 9 Relation entre les concepts d'adaptation [52]

Parmi les défis ci-dessus (résumés dans la Figure 10), les trois aspects suivants sont considérés comme les plus difficiles :

- Trouver la composition la plus optimale, en utilisant par exemple des paramètres non fonctionnels pour améliorer le comportement de la composition et de ses composants
- Protéger et surveiller les appareils avec des ressources de surveillance restreintes dans un environnement dynamique
- Gérer l'évolution dynamique de l'environnement et des ressources.

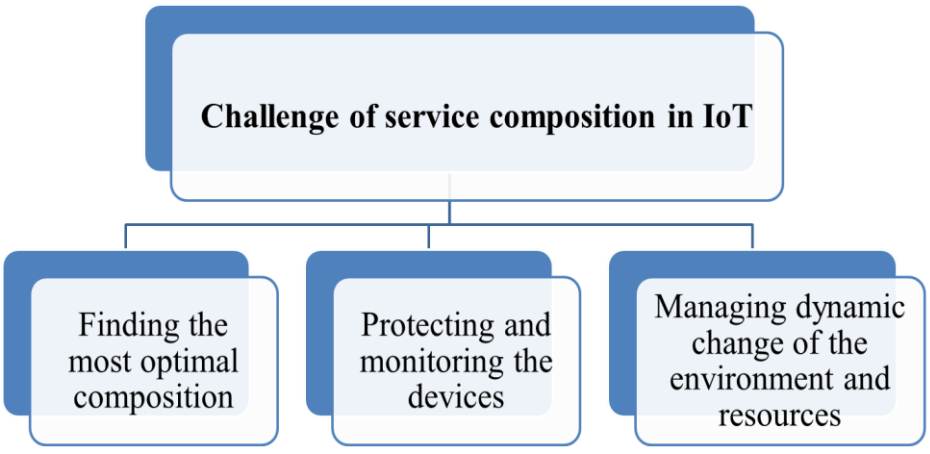


Figure 10 Défi de la composition des services dans l'IoT

11. L'application de la composition de services Web standard sur la composition de services de l'IoT

Il existe deux modes pour composer des services web, le mode médiation qui suit le modèle d'orchestration, où toutes les dépendances globales sont connues pour au moins un service (appelé le médiateur) avant exécution. De l'autre côté on parle du mode P2P qui suit le modèle de la chorégraphie.

La composition de service par orchestration consiste à faire l'assemblage de service selon un ordre et un flux d'exécution. L'exécution d'une composition par orchestration est réalisée par un coordinateur de services. L'utilisation d'un tel coordinateur implique une composition, avec une logique d'exécution qui sera interprétée par un coordinateur (Figure 11).

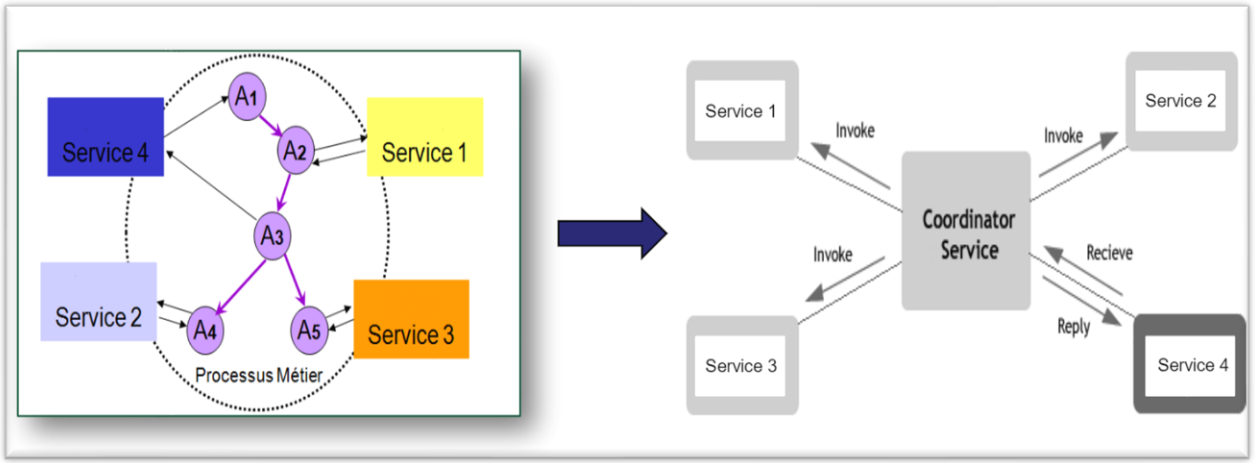


Figure 11 La composition de service par orchestration

La Chorégraphie de service est un effort de collaboration dans lequel chaque participant du processus décrit l'interaction qui l'appartient. Le comportement global du processus est basé sur l'interaction des différentes parties, chacune suivant son propre rôle de manière autonome (Figure 12).

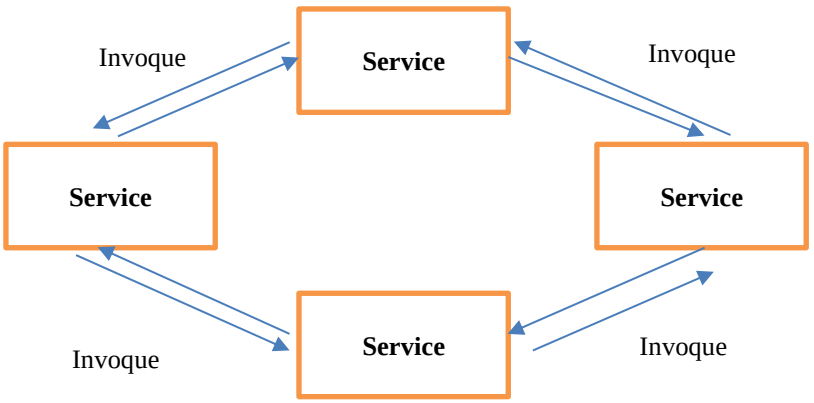


Figure 12 La Chorégraphie de service

Tout d'abord, en architecture orientée services (SOA), la composition des services nécessite que les services soient décrits de manière standard dans un annuaire, et offrent des formats d'échange (ou encapsulation d'échange) riches, précis et thématiques soumis à des règles strictes mais facilement évolutive, pour permettre l'exploitation des ressources disponibles. D'autant que cette démarche prend tout son sens dans la construction d'un service, qui sera proposé à de très nombreux utilisateurs, au fur et à mesure que dans l'IoT les relations évoluent, et l'organisation est amenée à être repensée. Les applications de l'IoT impliquent des éléments personnels, qui doivent répondre à des problèmes individuels. L'intervention d'un architecte est alors requise pour la construction d'un ensemble solide de services destinés à une population d'utilisateurs la plus large possible.

Chaque application IoT ne peut être utilisée que par son concepteur et sa famille, car elle est adaptée plus spécifiquement à leur usage, et/ou à certains de leurs objets. Dans l'IoT, le concepteur/consommateur de circuits est réduit, et parfois ces rôles sont joués par la même personne [53]. Dans les environnements IoT et fortement contraints, il semble approprié de disqualifier le protocole simple d'accès aux objets (SOAP), notamment par sa lourdeur pour le service qu'il est censé rendre. La plupart des opérations actuelles peuvent être effectuées via HTTP, de manière plus simple, plus légère et concise. La brièveté et la légèreté des échanges sont exactement ce dont l'IoT a besoin, et c'est ce que propose l'approche RESTful [54].

La chorégraphie et l'orchestration des services sont deux approches différentes que l'on s'interroge sur leurs récupérations ; s'ils expriment les mêmes règles et satisfirent les mêmes besoins. Une étude [55] s'est intéressée à l'utilisation de la chorégraphie sous SOAP et REST, et compare leurs forces et faiblesses face aux rouages de la chorégraphie. REST permet un couplage faible et la clarté de la définition des opérations souhaitées qui apparaissent dans le message, tandis que SOAP permet de décrire la complexité de certaines transactions et assure un suivi des échanges, particulièrement intéressant pour déboguer la cinématique de dialogue. L'approche chorégraphique est adaptée à l'IoT ; il permet de réduire l'utilisation des ressources réseau impliquées dans l'IoT. La chorégraphie respecte les contraintes énergétiques, car les interactions directes entre les nœuds nécessitent moins de réseau qu'un retour de données vers un point central. De plus, les capacités de traitement ajoutées aux objets (certes limitées, mais présentes) permettent la mise en œuvre des algorithmes décrivant la tâche à effectuer sur l'objet lui-même [53].

12. Conclusion

Dans ce chapitre, nous avons présenté un aperçu global sur le concept de l’internet des objets. Pour clarifier le concept de l’IoT, nous avons donné un ensemble de définitions montrant les différentes visions de ce concept. Ensuite, nous avons parlé des différents challenges de l’IoT, ensuite on a introduit le concept de la comparaison service avec une petite comparaison entre la composition des services web standard et celle dans l’IoT.

La pratique nous a montré que l'application directe de services Web standard sur les services de l'IoT ne convient pas et apportera de nouveaux problèmes ; car les services de l'IoT sont différents des services traditionnels et leur environnement est dynamique. La plus grande différence entre les services traditionnels et IoT est que le service traditionnel est une entité virtuelle mais les services IoT sont directement liés au monde physique [56], fournissant des services qui satisfont l'utilisateur et qui constituent l'une des problématiques colossales de l'IoT.

Dans le prochain chapitre, nous abordons certaines approches proposées populaires qui peuvent être considéré pour composer les services de l'IoT. Et nous proposerons une étude comparative de ces propositions d'approches.

CHAPITRE 3

Analyse de l’état d’art

Sommaire

Ce chapitre traite certaines approches proposées populaires qui peuvent être considérées pour la composition de services dans l'IoT. Puis dans la deuxième partie, nous présenterons une étude comparative entre les approches déjà proposées dans la littérature. Nous avons commencé notre enquête en introduisant quelques critères spécifiques, qui sont basés sur les défis fondamentaux de la composition des services cités précédemment dans le chapitre 2.

1. Critères fondamentaux pour la composition de service dans l’IoT

1.1. Composition dynamique

La composition peut avoir lieu pendant l'exécution de l'application pour intégrer de nouvelles caractéristiques et fonctionnalités, à la suite de l'évolution dynamique de l'environnement et des ressources.

1.2. L'adaptation

Suite au dynamique des changements de l'environnement, des ressources et de l'évolution des besoins des utilisateurs et du contexte, un changement peut avoir lieu pendant l'exécution de l'application pour intégrer de nouvelles fonctionnalités, et le système doit faire une mise à jour en temps réel et une décision sur les choix de composition peut donc être prise lors de l'exécution en accord avec cela.

Le critère d'adaptation est basé sur les sous-critères suivants :

- Type (moment) : il existe une technique d'adaptation hors ligne (lors de la décomposition et de la précision de l'objectif de l'utilisateur, ou lors de la sélection et de la découverte) et une technique d'adaptation online (réalisée à l'exécution).
- La technique utilisée dans le système d'adaptation.
- Les lieux de déploiement du système d'adaptation.
- La description : le comportement du système d'adaptation après détection des changements non acceptés.
- L'optimisation utilisée pour améliorer l'efficacité du système d'adaptation.

1.3. Indépendance et extensibilité

Les entités de la composition doivent être décrites indépendamment sans supposer l'existence ou l'absence des autres. Le mécanisme de composition doit également permettre l'ajout et la suppression d'entités à l'exécution, la composition doit donc être recalculée automatiquement.

1.4. Identification et résolution automatique des pannes et problèmes d'interaction

A cause de l'indépendance de chaque entité, des problèmes d'interaction et de communication peuvent apparaître (conductivité, pannes, charge de batterie et autres...) ; cela perturbera le résultat de la composition. La probabilité d'erreurs telles que des pannes et des changements de service dans l'IoT est élevée. Si une exception se produit, le système sera en mesure de recomposer la composition de service afin que la nouvelle composition puisse probablement fonctionner avec différents services sans défaillance.

1.5. Composition distribuée et décentralisée

L'exécution de la composition doit se faire de manière décentralisée et distribuée, du fait du nombre croissant d'appareils, ainsi que des contraintes de ressources (énergie et capacité de traitement) de chaque appareil.

1.6. Protocole de confiance, de protection et de sécurité

Un système IoT évolue avec de nouveaux nœuds et des nœuds existants, les entités du système IoT sont principalement portées par des humains ou des appareils actionnés par l'homme, le système se compose essentiellement d'appareils IoT non censurés offrant une grande variété de services, beaucoup d'entre eux (propriétaires) seront malveillants pour leur propre profit, un protocole de gestion de confiance est donc nécessaire pour résister aux attaques malveillantes et survivre dans des environnements hostiles.

1.7. L'optimisation

L'optimisation utilise des paramètres non fonctionnels pour améliorer le comportement de la composition et de ses composants. Bien que la satisfaction des exigences fonctionnelles soit importante pour la construction du système, l'optimisation des préférences non fonctionnelles peut être tout aussi cruciale. Par exemple, une composition qui minimise les temps de réponse et les coûts, et garantit la disponibilité, est requise.

Ces paramètres peuvent être dynamiques ou statiques. Les paramètres statiques sont généralement connus au moment du déploiement et ne sont généralement pas mis à jour lors de l'exécution. Cependant, les paramètres dynamiques représentent les caractéristiques variables d'un service donné telles que le temps de réponse, le niveau

d'énergie, la disponibilité et leurs valeurs sont fournies au moment de l'appel du service par un processus de surveillance.

1.8. Le modèle utilisé

Quel modèle est utilisé pour modéliser les relations entre les services IoT.

1.9. La performance

Le processus de composition nécessite des calculs. Ces calculs sont nécessaires pour composer un service qui satisfasse l'objectif de l'utilisateur, cela pourrait entraîner une grande consommation de temps, et cela est dû aux nombres de services disponibles dans l'IoT qui augmente énormément.

1.10. Résultats obtenus

Évaluation et résultats expérimentaux pour chaque approche proposée.

1.11.Les normes et le protocole utilisés

Indique les normes et le protocole utilisés les dans le système de composition des services.

1.12.**La représentation des services**

Indique quel style architectural est utilisé pour la représentation des services dans le système IoT.

2. Aperçu des propositions

Dans cette partie, nous allons présenter plusieurs approches déjà proposées pour la composition de services dans l'IoT.

Dans [57], une approche basée sur les réseaux sociaux distribués pour la gestion des appareils IoT et la composition des services a été proposée. Les auteurs ont encapsulé les appareils IoT dans des services Web à l'aide du style RESTful ; ils ont modélisé les relations entre les services IoT par les réseaux sociaux, et les ont classés en trois dimensions qui sont l'emplacement, le type et la corrélation. Les schémas d'enregistrement, de sélection et de gestion des services sont soigneusement conçus pour chacune de ces trois dimensions, et ainsi le réseau social est divisé en trois sous-réseaux distincts pour une gestion efficace et une recherche de service accélérée. Les auteurs ont proposé un mécanisme flexible et évolutif pour la composition des services IoT ; prenant en compte les relations sociales entre les services IoT, ce mécanisme permet la collaboration automatique d'appareils IoT hétérogènes afin de répondre aux besoins des utilisateurs. Toutes les expériences ont abouti en faveur des propositions des auteurs, les résultats montrent également que les algorithmes de recherche peuvent être appliqués à la sélection de services avec une grande efficacité.

Dans [49], une méthode QoS pour la composition des services a été proposée, les auteurs ont utilisé une prise de décision multi-attributs (MAMD) [58] pour calculer les performances QoS et évaluer chaque service individuellement. Selon les performances de QoS et la valeur fonctionnelle du service, plusieurs services sont choisis pour être dans la composition. Les auteurs divisent la composition des services en : analyse de la demande des utilisateurs, recherche et correspondance de services, sélection de services et enfin composition de services.

Un algorithme génétique (GA) amélioré est utilisé pour trouver les solutions optimales de composition de services. Par rapport à l'optimisation de la composition des services qui repose simplement sur l'évaluation des performances QoS moyennes, les critères de l'optimisation globale de la valeur de la fonction contribuent à une meilleure préservation des informations sensorielles importantes. Les résultats expérimentaux démontrent la faisabilité de l'optimisation basée sur le GA amélioré, tandis que l'efficacité du critère d'optimisation basé sur la QoS et la valeur de la fonction de service, ce qui apporte un schéma de composition plus adapté aux demandes de l'utilisateur.

Dans [51] et [59] une approche sociale adaptative basée sur la confiance pour la composition des services a été proposée, cette approche vise à résoudre la conception et la validation d'un protocole de gestion de confiance adaptatif et de survie pour SOA des systèmes IoT basés sur les réseaux sociaux [60]. Un utilisateur effectue une évaluation de la confiance sur la base de ses expériences de satisfaction directes passées et des retours de confiance des autres utilisateurs partageant des intérêts sociaux similaires. Les auteurs divisent les relations sociales en trois listes et chaque utilisateur dispose d'au moins un appareil haut de gamme désigné (c'est-à-dire un téléphone intelligent et un ordinateur portable) stockant ces listes dans le profil de l'utilisateur. Les autres appareils du même utilisateur ont le privilège d'accéder au profil. En déléguant le stockage et le calcul des réseaux sociaux à un appareil haut de gamme pour chaque utilisateur, de nombreux appareils bas de gamme (c'est-à-dire des capteurs) sont capables de partager et d'utiliser les mêmes informations sociales pour maximiser leurs performances. Pour l'évolutivité, une stratégie de gestion du stockage est utilisée pour les petits appareils IoT afin d'utiliser efficacement un espace de stockage limité. Les auteurs ont développé une technique de filtrage adaptatif pour trouver le meilleur moyen de combiner dynamiquement la confiance directe et la rétroaction de confiance indirecte, permettant à chaque nœud de sélectionner de manière adaptative son meilleur paramètre de confiance pour minimiser le temps de convergence et le biais de confiance. Les expériences ont évalué l'exactitude et la robustesse des modèles et des algorithmes. Dans [61], les auteurs ont étendu leur gestion adaptative de la confiance pour être résiliente aux attaques malveillantes. Les résultats montrent qu'il est capable d'approcher les performances idéales lors de la convergence et peut considérablement surpasser le protocole de sélection aléatoire non basé sur la confiance. Cependant, pour le système de surveillance, l'ajout d'autres paramètres entraîne un problème d'optimisation plus complexe et peut ne pas convenir aux appareils IoT à ressources limitées.

Dans [56], une méthode contextuelle pour la composition de services a été proposée, où les auteurs ont utilisé le langage d'ontologie Web (OWL) pour construire l'ontologie de contexte dans l'environnement de l'IoT. Afin de réduire la portée des informations contextuelles, le nombre de concepts et d'attributs communs dans différents sous-domaines, les auteurs divisent le champ de découverte de services en différents sous-domaines. Dans la composition des services, le processus de sélection des services a été divisé en deux sous-processus. Premièrement, les auteurs utilisent le contexte de calcul pour sélectionner les services appropriés. Ensuite, en fonction de la qualité de service des attentes des utilisateurs, ils choisissent les meilleurs services pour la composition des services afin de répondre aux besoins de l'utilisateur. Ainsi, les auteurs ont utilisé les informations contextuelles pour optimiser la composition du service dans un contexte particulier, pour trouver le plus compatible avec les besoins des utilisateurs du service tout en satisfaisant les contraintes contextuelles. Cette approche peut améliorer efficacement la satisfaction des utilisateurs, plus important encore, elle réduit la charge de traitement du contexte des appareils de terrain et adapte les ressources aux changements dynamiques de l'IoT. Les

résultats montrent que l'utilisation d'informations contextuelles a optimisé la composition des services, tout en satisfaisant les contraintes contextuelles. Un service de surveillance de périphérique est présenté comme un test sans aucune implémentation.

Dans [62] et [63], une approche middleware pour la composition des services logistiques dans l'IoT a été proposée. Un middleware qui se compose d'agents de ressources et d'agents de tâches est utilisé pour désigner au moment de l'exécution des services composants candidats pour participer à la composition d'une transaction logistique. Chaque service composant est représenté par un agent de ressource, qui est responsable du maintien des informations de QoS du service correspondant. Les agents de tâche représentent les instances de service composite et sont chargés de trouver des services composants qui satisfont aux exigences de QoS de bout en bout de l'utilisateur. Les auteurs n'utilisent que quelques moyens de surveillance pour suivre et assurer les opérations et garantir la robustesse du système logistique. Les expériences ont évalué l'exactitude et la robustesse des modèles et des algorithmes.

Dans [64], une approche probabiliste a été proposée pour décrire et analyser formellement la fiabilité et les propriétés liées aux coûts de la composition des services dans l'IoT. Les auteurs ont développé une approche pour modéliser la fiabilité et le coût de la composition des services sur la base du processus décisionnel de Markov (MDP) avec structure de coûts. Ainsi, la vérification de modèle probabiliste peut être appliquée pour vérifier et analyser les propriétés de qualité de la composition du service. En utilisant cette approche, la probabilité de succès du service composite peut être calculée avec les paramètres IoT actuels des appareils. De plus, la valeur de coût pour chaque service composite candidat peut être obtenue. Le développeur de services IoT peut utiliser les résultats pour prendre la décision concernant la sélection du service et les résultats peuvent l'alerter pour décider s'il doit déployer un service candidat alternatif en cas d'échec du service sélectionné lorsque la probabilité de succès du service sélectionné n'est pas très élevée. Bien que les auteurs vérifient et analysent également les propriétés des modèles de composition proposés, la façon de surveiller le système et ce qu'il faut faire en cas de défaillance n'y sont pas gérés. Les expériences qui en ont résulté montrent que cette approche est expressive.

Dans [65], une approche de composition dynamique des services a été proposée, basée sur la génération en ligne et la reprogrammation des plans optimaux pour l'orchestration les services en utilisant une technique de planification heuristique. Le mécanisme de la sélection proposé est basé sur une technique d'optimisation qui détecte les situations où certaines exigences de composition ne sont pas satisfaites ou certains services deviennent indisponibles ou lorsque des défaillances/exceptions se produisent. Il a la capacité de fonctionner malgré les changements qui se produisent dans le comportement des services sélectionnés et réduit par conséquent le besoin d'intervention humaine dans la reconfiguration de la composition. Les résultats expérimentaux obtenus sont encourageants et surpassent les techniques apparentées qui sont proches de leur portée. Aussi, la simulation a montré l'efficacité de l'approche proposée en présence d'échec grâce à la méthode de replanification.

Dans [66], une approche de composition de services Web RESTful asynchrone légère pour la composition des services dans l'IoT, qui est basée sur l'extension BPEL (Business Process Execution Language) a été proposé. Les auteurs ont divisé l'architecture en six couches et ajouté une interaction asynchrone pour gagner du temps et améliorer les performances.

L'expérimentation montre que la combinaison asynchrone peut en effet réduire le temps. Cela satisfera plus de demandes et améliora les performances sous une forte concurrence.

Dans [67], un Framework de composition des services REST léger pour l'IoT basé sur la programmation fonctionnelle avec des monades (une structure pour manipuler des langages fonctionnels purs avec des traits impératifs) a été proposé. Pour gérer les besoins dynamiques et le grand nombre de ressources, les auteurs ont traité les ressources comme des objets distribués avec des états locaux [68] accessibles par une interface uniforme via l'hypertexte, comme le HTTP 1.1 GET, PUT, POST, et Méthodes DELETE. Étant donné que la sortie de ces méthodes dépend des états des ressources, elles ne peuvent pas être modélisées comme des fonctions pures dont la sortie ne dépend que de l'entrée. Pour cette raison, les auteurs utilisent des monades pour accéder aux ressources REST à composer. Les résultats des tests montrent que l'approche proposée est réalisable et qu'il est possible d'affiner et d'améliorer les implémentations.

Dans [69], un Framework pour la composition des services IoT dans l'IoT a été proposé. Les auteurs prennent en compte la réputation du prestataire des services et la fiabilité du prestataire de réputation afin d'identifier les meilleurs candidats à la création d'un service composé, pour faire face à une tâche donnée. La réputation d'un fournisseur est calculée localement, de sorte que chaque fournisseur peut calculer une réputation différente en fonction de ses expériences personnelles antérieures avec différents fournisseurs (asymétrie). Cette approche est coûteuse en calcul et ne gère pas les échecs de connexion, ce qui n'est pas très adapté à un grand environnement comme un scénario de composition de service IoT.

Dans [15], une approche de composition de services Web orientée QoS basée sur un algorithme génétique multi-population pour l'Internet des objets a été introduite. Les auteurs formulent le problème de composition des services orientés QoS (QSC) en tant que modèle de programmation d'objectifs multicritères (MCGP) et développent un algorithme génétique multi-population (MGA) pour résoudre le modèle. MCGP non seulement attribue automatiquement des services Web de haute qualité pour combiner un service composite, mais trouve également des services composites non inférieurs en assouplissant les contraintes de QoS pour satisfaire les exigences de QoS des utilisateurs. Leurs expériences indiquent que MGA est capable de résoudre le problème QSC à grande échelle en termes d'efficacité et d'évolutivité en raison des excellentes performances de MGA en termes de capacité de recherche puissante et d'excellente capacité de convergence. De plus, les résultats illustrent une excellente performance en termes de capacité de recherche puissante et d'excellente capacité de convergence.

Dans [70], une composition de service adaptable pour l'IoT a été proposée. Les auteurs ont utilisé les concepts d'orchestration et de chorégraphie de services, le processus d'orchestration fonctionne au niveau local des ressources IoT pour intégrer les résultats obtenus à partir de plusieurs capteurs ou appareils intelligents tandis que le module de chorégraphie globale est utilisé pour invoquer plusieurs autres services pour étendre la coopération entre virtuel et mondes physiques. Cependant, aucun test expérimental n'a été présenté.

Dans [71], les auteurs ont présenté une architecture, une méthode et un algorithme de composition de services basés sur une ontologie sémantique dans le contexte du Web d'objets. Dans leur approche, les auteurs ont utilisé des ontologies pour décrire la relation entre les objets, les services et les règles afin de composer dynamiquement de nouveaux services. Cependant, dans le système proposé actuel, les utilisateurs doivent écrire leur propre exigence manuellement

dans l'interface Web en tant que langage normal pour le service composite souhaité. Évalué sur un centre commercial en tant que cas d'utilisation de composition des services, le système proposé est capable de traiter le langage naturel de l'utilisateur, ce qui permet aux utilisateurs de déterminer leurs besoins en matière d'inspection et de réparation.

Dans [72], une approche middleware décentralisée pour la composition des services dans l'IoT a été proposée, qui met l'accent à la fois sur la flexibilité et la réactivité des applications résultantes. Les auteurs ont utilisé un système multi-agents où les agents sont interconnectés via des relations qui leur permettent de se découvrir et d'interagir les uns avec les autres de manière flexible. Le mécanisme de composition repose sur des arbres de décomposition d'objectifs, dans lesquels un objectif est décomposé en sous-objectifs et ne peut être atteint qu'une fois tous ses sous-objectifs atteints. Un arbre de décomposition de buts est distribué sur tous les agents qui participent à sa réalisation et il est composé dynamiquement à l'aide des décompositions des buts préprogrammées. Les résultats de l'évaluation montrent que, dans le pire des cas, la surcharge de composition augmente linéairement avec le nombre d'agents dans la STN et suggèrent que les applications restent réactives lorsqu'elles s'adaptent à de nombreux appareils et pour des compositions de mashup relativement importantes.

Dans [73], une sélection des services Skyline parallèle pour la composition des services sensibles à la QoS dans l'IoT a été proposée. Les auteurs ont appliqué le modèle MapReduce pour la sélection des services Skyline parallèle afin d'améliorer l'efficacité de la sélection, et ils ont proposé une méthode de sélection de services Skyline dynamique basée sur un modèle Paper-Tape (PT), qui pourrait rapidement localiser le service variable et estimer l'influence de la variation des services Skyline d'origine. Des expériences basées sur des données réelles démontrent l'efficacité des propositions des auteurs.

Dans [74], un nouveau modèle architectural cloud multicouche est développé pour la maison intelligente basée sur l'IoT, qui fournit un degré considérablement amélioré d'interactions/interopérations entre les appareils domestiques hétérogènes et les services fournis par différents fournisseurs. Pour mieux résoudre les problèmes d'hétérogénéité, les auteurs ont utilisé l'ontologie comme un moyen prometteur d'aborder la représentation des données, les connaissances et l'hétérogénéité des applications. Pour prendre en charge la sécurité et la préservation de la confidentialité dans le processus d'interactions/interopérations, un Framework de service de sécurité basé sur une ontologie est conçu. Les résultats expérimentaux démontrent que dans le modèle architectural de cloud en couches proposé, les valeurs de test du temps de réponse moyen sont justifiées pour les exigences des applications de manipulation à domicile, en particulier pour les interactions/interopérations entre plates-formes hétérogènes.

Dans [75], une nouvelle approche d'exécution adaptative est proposée pour les environnements de service dynamiques, qui gère efficacement les changements des services qui se produisent au moment de l'exécution, à des fins de réparation et d'optimisation. Dans les situations où il n'est pas possible d'empêcher un comportement indésirable avant son exécution, l'approche permet une récupération efficace (avec presque aucune interruption), tout en raisonnant efficacement sur le meilleur remplacement en avant disponible. Chaque fois qu'une opportunité d'optimisation est identifiée (par exemple en raison de la disponibilité de nouveaux services de meilleure qualité), l'adaptation est déclenchée pour améliorer la solution actuelle, par opposition aux approches existantes où l'adaptation est principalement corrective. L'adaptation est effectuée dès que possible et en parallèle avec le processus d'exécution, réduisant ainsi le

temps d'interruption, augmentant les chances d'une récupération réussie et produisant la solution la plus optimale en fonction de l'état actuel de l'environnement. Cependant, la version actuelle de ce travail ne fournit aucun support pour raisonner sur le degré de fiabilité des offres de services lors de la sélection des services. Tous les services sont supposés avoir la même crédibilité en ce qui concerne les valeurs de qualité promises. L'étude d'évaluation montre l'efficacité de l'approche proposée dans le cadre de la composition des objets d'apprentissage, les résultats montrent que, même avec des changements fréquents, ou dans les cas où l'interférence avec l'exécution n'est pas évitable, l'approche parvient à se remettre de la situation avec interruption minimale.

Dans [76], une approche d'optimisation distribuée est proposée pour résoudre le problème de sélection des services dans le contexte de l'architecture de chorégraphie des services. Les auteurs ont proposé une approche de sélection QoS distribuée et optimale basée sur le paradigme multi-agent et le formalisme du problème d'optimisation des contraintes distribuées (DCOP). L'algorithme proposé prend en compte les spécificités du contexte de composition du service et la satisfaction des contraintes globales des utilisateurs. Les résultats des expérimentations montrent une performance très satisfaisante de l'approche en termes de rapidité et de nombre de messages échangés, ce qui en fait une solution appropriée aux problèmes de sélection de services distribués en temps réel. Cependant, la taille des hypercubes générés à chaque nœud doit être réduite.

Dans [77], un nouveau Framework de composition des services dynamiques et adaptatifs est proposé, qui prend en charge le raisonnement dynamique sur les tâches des utilisateurs et les comportements de service pour faire face à l'hétérogénéité des domaines IoT. Ce Framework est basé sur wEASEL [78] un modèle de service abstrait représentant les services et les tâches des utilisateurs en termes de signature, de spécification et de conversation. Cependant, le processus de composition des services implique la sélection et la coordination, et non le processus d'exécution de la composition. Pour faire face aux réalisations dynamiques des tâches utilisateur, un processus d'adaptation hors ligne est proposé pour modifier la structure de la tâche utilisateur afin d'augmenter la probabilité de trouver une composition. L'expérimentation montre que cette approche permet une composition plus précise et permet aux utilisateurs finaux de découvrir et d'étudier plus d'opportunités de composition que les autres approches.

Dans [79], un nouvel algorithme de composition de service IoT multi-Cloud sensible à l'énergie appelé (E2C2) est proposé, où les auteurs visent à créer un plan de composition sensible à l'énergie en recherchant et en intégrant le moins possible des services IoT, afin de répondre aux besoins spécifiques des utilisateurs. Ce travail vise à aborder un problème clé et émergent de l'IoT qui consiste à adopter des approches écoénergétiques pour les services et applications basés sur le cloud. Une modélisation et une analyse formelles de la traduction et de la transformation des besoins des utilisateurs sont adoptées pour l'algorithme proposé. L'algorithme a été évalué par rapport à quatre algorithmes de composition de service établis dans plusieurs environnements cloud (tous les Cloud, Cloud de base, Cloud intelligent et COM2). Les résultats démontrent une performance favorable de l'algorithme en termes d'obtention du moins des services recherchés pour arriver à une composition optimale et efficace en énergie.

Dans [14], un mécanisme efficace en énergie pour composer des services IoT est proposé, où ces services sont configurables sur le co-hébergement d'objets intelligents si nécessaire. Les auteurs ont proposé un cadre de service où les fonctionnalités fournies par les

objets intelligents sont encapsulées dans des services IoT. En outre, la composition des services IoT peut être réduite à un problème d'optimisation multi-objectifs et multi-contraintes (telles que les contraintes spatiales et temporelles, l'efficacité énergétique et la configurabilité), où des solutions optimales sont dérivées en adoptant des méthodes heuristiques, y compris l'algorithme génétique (GA), optimisation des colonies de fourmis (ACO) et optimisation des essaims de particules (PSO). Les résultats expérimentaux montrent que des compositions de services IoT approximativement optimales peuvent être dérivées et que le PSO fonctionne mieux que le GA et l'ACO en ce qui concerne la condition physique et la consommation d'énergie.

Dans [80], un algorithme de sélection de services centré sur l'énergie et sensible à la QoS (EQSA) est proposé pour la composition des services IoT. L'approche de sélection proposée consiste à présélectionner les services offrant le niveau de QoS requis pour la satisfaction de l'utilisateur en utilisant une stratégie d'optimisation lexicographique et une technique de relaxation des contraintes de QoS. Il est basé sur l'idée qu'il est toujours possible de faire des économies d'énergie en réduisant légèrement le niveau de QoS sans affecter la satisfaction de l'utilisateur. La sélection des services est formulée et résolue comme un problème d'optimisation multi-objectifs. Afin de réduire la consommation énergétique d'un service composite sans affecter la satisfaction de l'utilisateur, les services les plus adaptés parmi ceux présélectionnés sont alors sélectionnés en utilisant la notion de dominance relative des services au sens de Pareto. La dominance relative d'un service candidat dépend de son profil énergétique et de ses attributs QoS, ainsi que des préférences de l'utilisateur. L'algorithme proposé a été évalué à travers plusieurs scénarios de simulation. Les résultats obtenus montrent clairement les bonnes performances de l'algorithme EQSA en termes de temps de sélection, d'efficacité énergétique, de durée de vie de la composition, d'optimalité et sa valeur ajoutée par rapport aux algorithmes traitant séparément la QoS et la consommation d'énergie.

Dans [81], une approche de composition d'événements est proposée, où les auteurs intègrent des événements distribués dans la SOA pour créer une infrastructure SOA pilotée par les événements (EDSOA) pour les services IoT distribués et unifier l'accès à différents types d'appareils. Les auteurs adoptent des principes orientés services pour concevoir un middleware de publication/abonnement, et ses opérations réseau fonctionnent sur le protocole de service (SOAP) pour prendre en charge le routage de service, comme l'adressage des points de terminaison de service et la livraison d'appels de service. Les applications et les expérimentations ont montré l'efficacité et l'applicabilité de la solution.

Dans [82], une approche de composition des services automatisée et adaptable est proposée, dans laquelle la génération du schéma de composition est effectuée au moment de l'exécution grâce à l'utilisation de services abstraits fournis au moment de la conception. Le processus de composition qui prend en entrée une structure des besoins utilisateurs matérialisés par un graphe d'intentions et enrichit ce graphe pour expliciter les relations implicites. Le graphe enrichi est utilisé pour générer un schéma de composition initial en construisant le flux de contrôle et en sélectionnant les services abstraits appropriés. La sélection de ces services est basée sur la correspondance sémantique et le degré d'affinité sémantique entre les services abstraits. Ensuite, le schéma de composition final est généré à l'aide d'un mécanisme de raffinement des services abstraits utilisant des techniques de correspondance sémantique et prenant en compte le contexte et les contraintes de l'utilisateur. Les résultats expérimentaux montrent que le mécanisme de composition proposé permet de générer les compositions

correspondant aux besoins de l'utilisateur, grâce à l'utilisation de la correspondance sémantique entre intention et service abstrait et grâce à l'utilisation de l'affinité sémantique entre services.

Dans [83], une approche de sélection des services QoS distribuée et optimale basée sur le paradigme multi-agents et le formalisme d'optimisation des contraintes distribuées (DCOP) pour le Web d'objets est proposée. Les auteurs visent à résoudre le problème de sélection de service dans le cadre d'un service Chorégraphie au lieu d'un service central d'Orchestration. L'objectif est de mettre en œuvre une approche pour sélectionner les meilleurs services candidats qui maximisent la qualité de service globale tout en satisfaisant les contraintes globales de l'utilisateur de QoS, pour cela les auteurs ont revisité un algorithme DCOP basé sur la technique bien connue d'optimisation Branch and Bound pour développer de nouveaux algorithme « SynchBB4QoS » cherchant à maximiser ou à minimiser les attributs de QoS, tels que la minimisation du temps de réponse et la maximisation de la disponibilité. Et pour traiter le compromis entre les différents objectifs, une technique de pondération additive simple (SAW) et une méthode de fonction d'utilité ont été utilisées. SynchBB4QoS a une structure de contrôle simple qui le rend facile à implémenter sur des systèmes à faible capacité de calcul (mémoire limitée) mais nécessite des échanges de messages intensifs, les auteurs ont amélioré le "SynchBB4QoS" en proposant deux techniques : élaguer les services non prometteurs de chaque service d'agent défini à l'aide de la relation de Pareto-Dominance et d'élagage de la recherche en ordonnant les services candidats selon une fonction de coût local. Les résultats expérimentaux montrent que SynchBB4QoS amélioré a une meilleure scalabilité des messages échangés et donc un meilleur temps d'exécution.

Dans [84] et [85], un modèle de composition des services décentralisés pour l'environnement omniprésent de l'IoT est proposé, qui effectue une sélection émergente des services atomiques basée sur des champs potentiels artificiels (APF). Les auteurs ont utilisé des champs de potentiel artificiel (APF) pour diriger le processus de composition des services à travers l'équilibre des forces appliquées entre les demandes des services et les nœuds des services. Les APF sont formées en tenant compte du pourcentage des services demandés par l'utilisateur qui peuvent être offerts par les nœuds de fourniture des services, ainsi que de la disponibilité des nœuds des services. L'applicabilité de l'approche proposée est discutée dans un scénario exemplaire concernant la composition dynamique et personnalisée d'un service de guide virtuel audiovisuel dans un réseau IoT d'un lieu de salon professionnel. Cependant, aucun test expérimental n'a été présenté.

Dans [86], un Framework orienté service, centré sur l'utilisateur et sensible aux événements nommé FASEM, pour la surveillance des services ambiants et des événements dans les espaces ambiants est présenté, en d'autres termes, le Framework est capable d'effectuer la surveillance des services. Pour gérer automatiquement les événements pouvant survenir dans les environnements ambiants. Cette surveillance est basée sur un processus dynamique de découverte et de sélection des services pour améliorer l'auto-adaptation aux changements imprévus et assurer la continuité des services avec la meilleure qualité. De plus, ce Framework est basé sur quatre aspects principaux, à savoir : (1) la découverte des services ambiants ; une approche de découverte qui est responsable du maintien d'un état cohérent de tous les services ambiants disponibles qui peuvent apparaître et disparaître dynamiquement dans les environnements d'intelligence ambiante (AmI). (2) Invocation des services ambiants ; une approche d'invocation qui est responsable de la préparation, de l'invocation et de la mise à jour des informations sur un service ambiant. (3) sélection de services ambiants locaux ; une

démarche de sélection locale qui est chargée de choisir le meilleur service ambiant, en termes de qualité estimée, à partir d'une classe de service ambiant donnée. Et (4) sélection globale de services ambiants ; la stratégie de sélection globale qui est chargée de contrôler la composition des services, la sélection et l'invocation des services locaux afin d'atteindre efficacement l'objectif souhaité lorsqu'un événement se produit dans un environnement ambiant. Les auteurs ont utilisé un contrôle prenant en charge l'auto-adaptation aux changements imprévus en utilisant le remplacement des services et la replanification en cas d'échec. De plus, les auteurs étudient comment des changements imprévus peuvent être détectés dans l'étape de découverte de services en considérant le degré de dynamicité de l'environnement. Les résultats expérimentaux et les performances obtenues à partir d'essais poussés montrent l'efficacité et la faisabilité de l'approche proposée dans le cas d'un environnement à grande échelle.

Dans [87], un nouvel algorithme de composition des services IoT est proposé, qui produit et génère automatiquement un service IoT composite avec un temps de réponse minimal et optimal. Premièrement, les auteurs trouvent le premier moyen de générer chaque élément de données dans une demande d'utilisateur, et la deuxième phase ne récupère que les services IoT nécessaires et produit un service IoT composite pour satisfaire la demande de l'utilisateur. Cependant, cette approche ne traite que du temps de réponse alors qu'il existe différents critères de QoS. Les expérimentations montrent des résultats prometteurs.

Dans [88], un algorithme de colonie d'abeilles artificielles à modification croisée (CMABC) est proposé, où une solution optimale globale a été utilisée pour accélérer le taux de convergence en imitant l'opération croisée de l'algorithme génétique (GA) pour surmonter le problème et l'espace de solution chaotique. Les auteurs visent à fournir la tâche de service qui satisfait l'utilisateur le plus rapidement possible. Le résultat de l'expérience de simulation montre que CMABC présente une vitesse de convergence plus rapide et une meilleure précision de convergence que certains autres algorithmes d'optimisation intelligents. Cependant, le modèle d'algorithme n'est pas parfait. L'optimisation séquentielle des nœuds de tâches n'est pas prise en compte.

3. Une étude comparative entre les approches

Dans cette section, nous présentons une étude comparative entre les différentes approches déjà présentées. Notre comparaison est basée sur les critères cités précédemment. Nous avons essayé de connaître leurs limites ainsi que la meilleure catégorie d'approches répondant le plus complètement possible aux critères. Cette étude comparative [2] a été publiée dans le journal international « Int. J. Communication Networks and Distributed Systems, vol. 23, No. 2, 2019 ».

Remarque : dans la série de tableau qui va suivre le (+) veut dire : Satisfait, (-) veut dire : Insatisfait, (/) veut dire : non mentionné et (N/A) veut dire : non disponible

Tableau 3 Comparaison entre les approches avec le critère "Composition dynamique"

L'approche	Composition Dynamique
[57]	-
[49]	+ Méthode Top-down [89] pour composer les services dynamiquement
[50] , [62] et [63]	+
[51] et [59]	+ Gestion dynamique de la confiance
[56]	+ Raisonnement : modélisation de contexte avec OWL
[64]	+
[65]	+
[66]	-
[67]	+
[69]	-
[15]	-
[70] et [90]	+ (Uniquement au moment de la conception)
[71]	+
[72]	+
[73] & [91]	+
[74]	+
[75]	+

Tableau 4 Comparaison entre les approches avec le critère "L'adaptation"

L'approche	L'adaptation					
		Le type	***Technique***	***Les lieux de déploiement***	***La description***	***L'optimisation***
[57]	-	N/A				
[49]	-	N/A				
[50] , [62] & [63]	+	Online & Offline	Système de surveillance	Un agent est affecté au service atomique	Lorsqu'une zone est surveillée, si elle comporte une erreur, ils considèrent que son erreur est supprimée, le détail omis. (Pas réaliser)	+Gestion de la Scalabilité + Utiliser un algorithme pour utiliser le moins de ressources de surveillance, mais en tirer le meilleur parti (Plus grande valeur d'utilité)
[51] & [59]	+	Offline	Adaptation des poids	Le service (ressource)	Chaque utilisateur met à jour dynamiquement les "paramètres de poids" dans un intervalle optimisé	-L'ajout d'autres paramètres entraîne un problème d'optimisation plus complexe et peut ne pas convenir aux appareils IoT à ressources limitées.
[56]	-	N/A				
[64]	+	Online & Offline	Candidat alternatif avec un système de surveillance	/	Candidat alternatif en cas d'échec (lorsque la probabilité de succès du service sélectionné n'est pas très élevée.)	/
[65]	+	Online	Système de surveillance centralisé avec notification (Moniteur d'état de l'appareil)	Un annuaire d'agents et de services (cloud)	Resélectionner les meilleurs services efficaces et refaire l'opération de replanification	+ Chaque service enregistré envoie une notification sur son statut + Il est plus facile de gérer une liste de services indisponibles qu'autrement.

L’approche	L’adaptation					
		Le type	***Technique***	***Les lieux de déploiement***	***La description***	***L'optimisation***
						- Une liste des services indisponibles (centralisée)
[66]	-	N/A				
[67]	-	N/A				
[69]	-	N/A				
[15]	-	N/A				
[70] & [90]	+	Online	Système de surveillance fonctionnant avec des messages de battements de cœur	Moniteur d'état de l'appareil	Chaque fois qu'un périphérique ne parvient pas à envoyer un message de pulsation après un délai d'attente donné, *Device Status Monitor* signale son indisponibilité et avertit Service Replacement Manager afin de remplacer le service indisponible.	- Resélectionnez au hasard un service qui peut réellement remplacer le service indisponible à partir d'une liste de service demandée à partir du registre des services + Assurer la synchronisation d'état du BP afin qu'il reprenne son exécution à partir du même état qu'avant le blocage de ses activités suite à une panne de service
[71]	-	N/A				
[72]	-	N/A				
[73] & [91]	+	Offline (Méthode de sélection)	Paper-Tape	/	S'il y a de nouveaux services nécessitant un enregistrement, des services invalides supprimés et des changements de qualité de service, maintenez les services Skyline au lieu de les recalculer une fois que certains services varient.	+ Maintenir les services Skyline au lieu de les recalculer + Paper-Tape peut être traité comme une matrice, ou l'auteur enregistre les informations de service sur chaque attribut non fonctionnel dans un Paper-Tape

L’approche	L’adaptation					
		Le type	***Technique***	***Les lieux de déploiement***	***La description***	***L'optimisation***
						correspondant.
[74]	-	N/A				
[75]	+	Online (Gère les changements de service survenant au moment de l'exécution)	Méthode parallèle	Niveau middleware	Dans les situations où il n'est pas possible d'empêcher un comportement indésirable avant son exécution, l'approche permet une récupération efficace (avec presque aucune interruption), tout en raisonnant efficacement sur le meilleur remplacement disponible. Chaque fois qu'une opportunité d'optimisation est identifiée (par exemple en raison de la disponibilité de nouveaux services de meilleure qualité), l'adaptation est déclenchée pour améliorer la solution actuelle, par opposition aux approches existantes où l'adaptation est principalement corrective.	L'adaptation est effectuée dès que possible et en parallèle avec le processus d'exécution, réduisant ainsi le temps d'interruption, augmentant les chances d'une récupération réussie et produisant la solution la plus optimale en fonction de l'état actuel de l'environnement.

Tableau 5 Comparaison entre les approches avec les deux critères "Indépendance et extensibilité / Identification et résolution automatique des pannes et problèmes d'interaction"

L’approche	Indépendance et extensibilité	Identification et résolution automatique des pannes et problèmes d'interaction
[57]	+	-

L’approche	Indépendance et extensibilité	Identification et résolution automatique des pannes et problèmes d'interaction
[49]	+	-
[50]	+	+ Système de surveillance (FbasedMonitor)
[51] et [59]	+	-
[56]	+	-
[62] et [63]	+	+ Système de surveillance (Mécanisme de coordination décentralisé)
[64]	+	+ Système de surveillance (Candidat alternatif en cas d'échec)
[65]	+	+ Suivi de l'orchestration des services Tolérance aux pannes
[66]	+	-
[67]	+	- (Traitement des erreurs et surveillance des événements pour les travaux futurs)
[69]	+	- (Système de suivi des travaux futurs)
[15]	+	-
[70]	+	+
[71]	+	-
[72]	+	-
[73] et [91]	+	-
[74]	+	-
[75]	+	+
[76]	+	-
[77]	+	-
[79]	+	+

L'approche	Indépendance et extensibilité	Identification et résolution automatique des pannes et problèmes d'interaction
[14]	+	-
[80]	+	-
[81]	+	-
[82]	+	-
[83]	+	-
[84] et [85]	+	-
[86]	+	+ Monitoring system
[87]	+	-
[88]	+	-

Tableau 6 Comparaison entre les approches avec les deux critères "Composition distribuée et décentralisée / Protocole de confiance, de protection et de sécurité"

L'approche	Composition distribuée et décentralisée	Protocole de confiance, de protection et de sécurité
[57]	+	-
[49]	- (Utilise un référentiel de services)	-
[50]	+	-
[51] et [59]	+ (Pas d'autorité de confiance centralisée)	+ (Protocole de gestion de confiance)
[56]	+ (Divise le champ de découverte de service en différentes sous-zones)	-
[62] et [63]	+ (Approche middleware avec agents)	-
[64]	- (Orchestration)	-
[65]	+	-

L’approche	Composition distribuée et décentralisée	Protocole de confiance, de protection et de sécurité
	(Orchestration avec architecture multi-agents)	
[66]	+ (Divise l'architecture en six couches)	+ (Plateforme de sécurité)
[67]	+	-
[69]	- (Décentraliser le calcul de la réputation pour les travaux futurs)	-
[15]	+ Utilise un modèle de programmation multicritères	-
[70]	+ Processus d'orchestration locale et processus de chorégraphie globale	-
[71]	-	-
[72]	+ Un arbre de décomposition des objectifs est distribué sur tous les agents	-
[73] et [91]	+ Divise l'espace de données en N partitions (plusieurs blocs de données)	-
[74]	+ Environnement basé Internet distribué	+ Un Framework de service de sécurité basé sur une ontologie
[75]	+	-
[76]	+ Architecture de chorégraphie	L'utilisation d'un paramètre de QoS statique (Sécurité), qui Représente le niveau de sécurité assuré par un service (authentification, chiffrement, etc.)
[77]	-	-
[79]	+	-

L'approche	Composition distribuée et décentralisée	Protocole de confiance, de protection et de sécurité
[14]	+	-
[80]	+	-
[81]	+ (Événements distribués)	+ (Une politique de sécurité est efficacement intégrée à un événement)
[82]	/	+ (Intention de l'utilisateur)
[83]	+ (Chorégraphie)	- (Juste un paramètre de niveau de sécurité QoS)
[84] et [85]	+	-
[86]	+	-
[87]	- (approche Middleware)	-
[88]	+	-

Tableau 7 Comparaison entre les approches avec les deux critères "L'optimisation / Le modèle utilisé"

L'approche	L'optimisation	Le modèle utilisé
[57]	+ Position géographique	Réseau social a 3 dimensions
[49]	+ Évaluation de la QoS (Prix, temps de réponse, fiabilité, réputation et localisation géographique) - Aucune contrainte de ressources	Algorithme génétique amélioré
[50]	+ Évaluation utilisant la fiabilité, le temps de réponse et le coût du dispositif atomique	Modèle de Petri
[51] et [59]	+ Contrainte de ressources (Appareil haut de gamme pour chaque utilisateur) + Position géographique	Réseaux sociaux (Un environnement social IoT basé sur SOA)

L’approche	L'optimisation	Le modèle utilisé
[56]	+ Contrainte de ressources : réduire la charge de traitement du contexte de l'appareil de terrain le plus faible. + Réduire la portée des informations contextuelles + L'utilisation de la QoS.	Modèle basé sur le contexte
[62] et [63]	+ Contrainte de ressources + Informations QoS (Prix, Délai, Fiabilité, Degré de Réputation, Disponibilité)	Système multi-agents (Modèle middleware de coordination)
[64]	+ Fiabilité et coût + Contrainte de ressources - Aucune contrainte de temps réel (pour les travaux futurs)	Markov Decision Process (MDP)
[65]	+ Contexte (localisation, niveau de batterie etc.) +Gestion de la QoS (temps de réponse, fiabilité, disponibilité et réputation) - Extrêmement exigeant - Traitement intensif	Architecture Multi-agents
[66]	-	Extension BPEL asynchrone
[67]	+ Contrainte de ressources	Programmation fonctionnelle avec monade comme Framework
[69]	+ Une approche basée sur la réputation du prestataire et la fiabilité du prestataire de réputation	Framework de réputation

L’approche	L'optimisation	Le modèle utilisé
[15]	+ L'utilisation de trois propriétés QoS (Délai d'exécution, Fiabilité et Coût d'exécution)	Modèle de programmation multicritères pour le problème QSC (À l'aide d'un algorithme génétique multi-populations)
[70]	+ Contraintes de ressources + Contraintes temps réel	Processus d'orchestration locale et processus de chorégraphie globale
[71]	Les utilisateurs écrivent leur propre exigence dans l'interface Web comme un langage normal, comme rechercher quelque chose dans un moteur de recherche Web	Modèle basé sur une ontologie sémantique
[72]	+ Temps de réponse	Multi-agent (Modèle décentralisé) Réseau sociotechnique (RTC)
[73] & [91]	Les contraintes de QoS sont définies par les exigences de l'utilisateur (latence, temps de réponse, débit, disponibilité, fiabilité et capacité de réussite)	MapReduce basé un Modèle Skyline
[74]	/	Un nouveau modèle architectural cloud multicouche
[75]	/	Une nouvelle approche adaptative basée sur un modèle de sélection de service réactif
[76]	Contraintes de l'utilisateur QoS (minimiser le temps de réponse et maximiser la disponibilité en même temps) Énergie, disponibilité, temps de réponse, fiabilité, prix et sécurité.	Paradigme multi-agents
[77]	/	Un Framework de composition basé sur WEASEL
[79]	Énergie, coordonnées géographiques	Une nouvelle approche de composition de services IoT multicloud sensible à l'énergie
[14]	Contraintes spatiales, temporelles et énergétiques	Un mécanisme efficace en énergie avec des méthodes heuristiques
[80]	QoS (coût, temps de réponse, réputation, fiabilité et disponibilité) et énergie	Un algorithme de sélection de services centré sur l'énergie et sensible à la QoS (comme problème d'optimisation multi-objectifs) Méthode d'optimisation lexicographique et solution

L’approche	L'optimisation	Le modèle utilisé
		optimale de Pareto
[81]	-	Une infrastructure SOA événementielle (EDSOA)
[82]	+ (Intention de l'utilisateur : une combinaison d'un objectif et d'un ensemble de contraintes)	Une approche de composition de service automatisée et adaptable basée sur un modèle d'intention
[83]	Attributs QoS (le temps de réponse, le niveau d'énergie, la disponibilité, le prix)	Multi-Agents
[84] & [85]	/	Multi-agents et champs de potentiel artificiel (APF)
[86]	Contraintes de qualité de service (QoS) Disponibilité, temps de réponse et probabilité de réponse	Framework sensible aux événements
[87]	Temps de réponse	Un modèle basé sur le temps de réponse en exploitant le parallélisme
[88]	Attributs QoS, tels que le temps de réponse, le coût, la fiabilité, la disponibilité et la réputation, etc.	Un algorithme de colonie d'abeilles artificielles à modification croisée (CMABC)

Tableau 8 Comparaison entre les approches avec les deux critères "Les normes et les protocoles utilisés/ La représentation des services"

L’approche	Les normes et les protocoles utilisés	La représentation des services
[57]	WSDL	RESTful style Décrit par un fichier configurable XML
[49]	/	/
[50]	/	/
[51] & [59]	/	/
[56]	OWL	/
[62] et [63]	/	/
[64]	/	Machine à états finis étiquetée (FSM)
[65]	/	/

L’approche	Les normes et les protocoles utilisés	La représentation des services
[66]	/	Style RESTful
[67]	XML	Style RESTful
[69]	/	/
[15]	/	/
[70]	BPMN to BPEL	/
[71]	XML/RDF OWL	/
[72]	RDF	Style RESTful
[73] et [91]	/	/
[74]	Ontologie	/
[75]	OWL-S or WSDL-S	/
[76]	/	/
[77]	contExt Aware Web Service dEscription Language OWLS-TC4-based	Modèle wEASEL
[79]	/	/
[14]	Profil de périphériques pour les services Web (DPWS) standard [92]	Modèle DPWS (Style RESTful)
[80]	/	Concrete Versus Abstract Service
[81]	SOAP	/
[82]	OWL-S	Abstract services
[83]	HTTP, XML, XMPP	/
[84] et [85]	/	/
[86]	WSDL	Ambient services
[87]	WSDL, OWL, WSLA, WSDL & BPEL	/
[88]	/	/

Tableau 9 Comparaison entre les approches avec les deux critères "Les performances / Résultats obtenus"

L'approche	Les performances	Résultats obtenus
[57]	+ Les processus de recherche et de sélection sont mis en œuvre de manière parallèle. + Utilise 3 algorithmes pour chaque dimension (Emplacement, type et corrélation)	*Expériences de simulation faisant varier le nombre d'appareils IoT de 100 à 1200.* + Avec la croissance de l'espace de recherche, le temps d'exécution augmente beaucoup plus lentement par rapport à l'alternative exponentielle. + La complexité temporelle est O (1) pour l'algorithme de type (utilisé une table de hachage) + Les algorithmes de recherche peuvent être appliqués à la sélection de services avec une grande efficacité
[49]	+ Utiliser GA comme algorithme d'optimisation globale + GA peut aider à obtenir une solution optimale ou quasi-optimale à un coût de calcul relativement faible + Tant que la fonction de fitness est modélisée mathématiquement, GA peut être utilisé pour résoudre un problème d'optimisation à grande échelle	+ Par rapport à l'optimisation de la composition des services basée simplement sur l'évaluation des performances QoS moyennes, les critères de l'optimisation globale de la valeur de la fonction contribuent à une meilleure préservation des informations sensorielles importantes. + Les résultats expérimentaux montrent que l'utilisation de la valeur de la fonction de service dans l'optimisation de la composition du service apporte un schéma de composition plus adapté aux demandes du demandeur. + Les résultats expérimentaux démontrent la faisabilité de l'optimisation basée sur le GA amélioré, aussi, l'efficacité de critère d'optimisation basé sur la qualité de service et la valeur de la fonction de service.
[50]	+ Utilise « FindTOptimal » comme algorithme de recherche de composition, qui utilise une fonction de performance complète nommée (rtc) pour évaluer la rentabilité.	+ "FindTOptimal" montre un bon résultat concernant la satisfaction des utilisateurs. + Les résultats expérimentaux ont prouvé la solidité et l'exactitude des

L’approche	Les performances	Résultats obtenus
		algorithmes.
[51] et [59]	+ Utilise 3 listes de relations : une liste d'amis, une liste de localisation et une liste d'appareils (services) avec lesquels on interagit directement. + Stockage dans un appareil haut de gamme. + Pour l'évolutivité, une stratégie de gestion du stockage est utilisée pour les petits appareils IoT afin d'utiliser efficacement l'espace de stockage limité + Utilisation d'une méthode basée sur le filtrage adaptatif pour ajuster dynamiquement les valeurs afin d'améliorer les performances d'évaluation de la confiance.	+ Les résultats montrent qu'il est capable d'approcher les performances idéales lors de la convergence et peut considérablement surpasser le protocole de sélection aléatoire non basé sur la confiance. + Les auteurs ont démontré par simulation la supériorité du protocole de confiance IoT adaptatif sur EigenTrust et PeerTrust en termes de convergence de confiance, de précision et de résilience contre les nœuds malveillants effectuant des attaques de service d'auto-promotion, de dénigrement, de bourrage de bulletins de vote et de service opportuniste. - Pour le système de surveillance, l'ajout d'autres paramètres entraîne un problème d'optimisation plus complexe et peut ne pas convenir aux appareils IoT à ressources limitées.
[56]	+ L'utilisation d'une méthode de pondération simple.	+ L'utilisation d'informations contextuelles a optimisé la composition des services, tout en satisfaisant les contraintes contextuelles
[62] et [63]	+L'utilisation de l'algorithme UFBasedMonitor pour résoudre le manque de ressources de surveillance.	+ Les expériences ont évalué l'exactitude et la robustesse des modèles et des algorithmes.
[64]	+Basé sur MDP. +L'application des techniques de modélisation probabiliste pour vérifier et analyser automatiquement les propriétés - Basé sur le modèle d'orchestration qui se concentre uniquement sur un seul service composite.	+Cette approche est expressive
[65]	+ Technique d'optimisation du mécanisme de sélection, qui peut détecter des situations où certaines exigences de composition ne sont pas satisfaites ou	Pour évaluer la performance de l'approche proposée, trois métriques ont été utilisées : le temps de génération du plan, le temps de réponse et le taux de réussite. Ces métriques sont affectées par différents facteurs tels

L’approche	Les performances	Résultats obtenus
	certains services deviennent indisponibles ou lorsque des échecs/exceptions se produisent. + Technique de planification heuristique.	que le nombre de services dans les scénarios de simulation, les échecs et le nombre de tâches de planification simultanées. Ils ont comparé leurs résultats avec trois autres approches : HTN-DL, CDSC et FCoSC. Les résultats obtenus sont encourageants et surpassent les techniques apparentées proches de leur portée La simulation a également montré l'efficacité de l'approche proposée en présence d'échec grâce à la méthode de replanification.
[66]	+ Interaction asynchrone pour gagner du temps et améliorer les performances.	L'expérimentation montre que la combinaison asynchrone peut en effet réduire le temps. Cela satisfera plus de demandes, améliorant les performances sous une forte concurrence.
[67]	Utilisation d'une collection de nomades (programmation fonctionnelle) agencée pour accéder à un ensemble de ressources REST à composer.	Les résultats des tests montrent que l'approche proposée est réalisable et qu'il est possible d'affiner et d'améliorer les implémentations.
[69]	/	- L'approche ne se concentre pas sur les mécanismes de découverte de service ou les langages de spécification de service. - Extrêmement exigeant
[15]	+ L'utilisation d'un modèle de programmation à buts multicritères (MCGP) et d'un algorithme génétique multi-populations pour résoudre le modèle	+ Les expériences indiquent que MGA est capable de résoudre le problème QSC à grande échelle en termes d'efficacité et d'évolutivité +Excellentes performances en termes de capacité de recherche puissante et d'excellente capacité de convergence
[70]	+ Parallélisme + Précision et efficacité -Humain dans la boucle pour initier le processus de chorégraphie	/

L’approche	Les performances	Résultats obtenus
[71]	+ Réutiliser et partager les propriétés et les attributs de l'objet virtuel + Les services sont décrits dans la base de connaissances de la couche ontologie, et des règles sont élaborées pour effectuer des tâches de raisonnement pour la découverte et le traitement automatiques des services.	*Les auteurs ont considéré le scénario du centre commercial comme cas d'utilisation de la composition des services* Le système proposé est capable de traiter le langage naturel de l'utilisateur, ce qui permet aux utilisateurs de déterminer leurs besoins en matière d'inspection et de réparation.
[72]	+ L'utilisation de systèmes multi-agents, chaque agent coopère entre eux lors de l'exécution + Le mécanisme de composition repose sur des arbres de décomposition de buts	Les résultats de l'évaluation montrent que, dans le pire des cas, la surcharge de composition augmente linéairement avec le nombre d'agents dans la STN et suggèrent que les applications restent réactives lorsqu'elles s'adaptent à de nombreux appareils et pour des compositions de mashup relativement importantes.
[73] et [91]	+Traitement parallèle +L'utilisation d'une méthode de partitionnement par angle (réduit de nombreux calculs redondants et équilibre la charge de travail)	Des expériences basées sur des données réelles démontrent l'efficacité des propositions des auteurs.
[74]	/	Les résultats expérimentaux démontrent que dans le modèle architectural du cloud en couches proposé, les valeurs de test du temps de réponse moyen sont justifiées pour les exigences des applications de manipulation à domicile, en particulier pour les interactions/interopérations entre plates-formes hétérogènes.
[75]	/	L'efficacité de l'approche proposée est démontrée à la fois analytiquement et empiriquement à travers une évaluation d'études de cas appliquée dans le cadre de la composition d'objets d'apprentissage. En particulier, les résultats montrent que, même avec des changements fréquents (par exemple 20 changements par exécution de service), ou dans les cas où l'interférence avec l'exécution n'est pas évitable (par

L’approche	Les performances	Résultats obtenus
		exemple, lorsqu'un service exécuté délivre des valeurs de qualité imprévues), l'approche parvient à récupérer de la situation avec une interruption minimale.
[76]	Technique de pondération additive simple (SAW) et utilisation de la méthode de la fonction d'utilité, qui permet une mesure unifiée de plusieurs objectifs.	Les résultats des expérimentations montrent une performance très satisfaisante de l'approche en termes de rapidité et de nombre de messages échangés, ce qui en fait une solution appropriée aux problèmes de sélection de services distribués en temps réel.
[77]	/	Le système basé sur wEASEL effectue une composition plus précise et permet aux utilisateurs finaux de découvrir et d'étudier plus d'opportunités de composition que d'autres approches.
[79]	/	L'algorithme a été évalué par rapport à quatre algorithmes de composition de service établis dans plusieurs environnements cloud (tous les clouds, cloud de base, cloud intelligent et COM2). Les résultats démontrent une performance favorable de l'algorithme en termes d'obtention du moins de services recherchés pour arriver à une composition optimale et économe en énergie.
[14]	Pour résoudre le problème d'optimisation multi-objectifs et multi-contraintes, trois algorithmes d'optimisation, dont l'algorithme génétique (GA), l'optimisation des colonies de fourmis (ACO) et l'optimisation des essaims de particules (PSO), sont adoptés. L'utilisation de la classe de service pour représenter la fonctionnalité des appareils	Les résultats expérimentaux montrent que des compositions de services IoT approximativement optimales peuvent être dérivées et que le PSO fonctionne mieux que le GA et l'ACO en ce qui concerne la forme physique et la consommation d'énergie
[80]	L'utilisation d'une stratégie d'optimisation lexicographique, d'une technique de relaxation des contraintes de QoS et d'une solution optimale de Pareto.	L'algorithme proposé a été évalué à travers plusieurs scénarios de simulation. Les résultats obtenus montrent clairement les bonnes performances de l'algorithme EQSA en termes de temps de sélection, d'efficacité énergétique, de durée de vie de la composition, d'optimalité et sa valeur ajoutée par rapport aux algorithmes traitant séparément la

L’approche	Les performances	Résultats obtenus
		QoS et la consommation d'énergie. L'algorithme EQSA est évolutif dans le temps pour les environnements IoT à grande échelle et est capable de trouver des solutions très proches de l'optimum (environ 98%). Une durée de vie de la composition proche du cas optimal en consommant une quantité d'énergie supplémentaire raisonnable (environ 13 à 31%).
[81]	Un middleware de publication/abonnement orienté service pour établir un bus de service basé sur les événements pour connecter ensemble des ressources et des services IoT hétérogènes sur une large zone pour des interactions transparentes sans référence aux emplacements de service et à l'état en ligne, appelé Unified Message Space (UMS)	Les applications et les expérimentations ont montré l'efficacité et l'applicabilité de la solution.
[82]	La génération du schéma de composition est réalisée en partie au moment de l'exécution tout en ayant des services abstraits comme composants au moment de la conception. Cela offre une certaine flexibilité et adaptabilité dans la composition sans avoir à gérer la composition à partir de zéro au moment de l'exécution.	Les résultats expérimentaux montrent que le mécanisme de composition proposé permet de générer les compositions correspondant aux besoins de l'utilisateur, et ce grâce à l'utilisation de la correspondance sémantique entre intention et service abstrait et grâce à l'utilisation de l'affinité sémantique entre services.
[83]	Un algorithme DCOP révisé basé sur la technique d'optimisation bien connue Branch and Bound pour développer un nouvel algorithme complet "SynchBB4QoS" cherchant à maximiser ou à minimiser les attributs de QoS, tels que la minimisation du temps de réponse et la maximisation de la disponibilité.	Les résultats expérimentaux montrent que SynchBB4QoS amélioré a une meilleure évolutivité des messages échangés et donc un meilleur temps d'exécution.

L’approche	Les performances	Résultats obtenus
	Amélioration de la « SynchBB4QoS » en proposant deux techniques : l'élagage des services non prometteurs de chaque ensemble de services d'agent en utilisant la relation de Pareto-Dominance et l'élagage de la recherche en ordonnant les services candidats selon une fonction de coût local.	
[84] et [85]	Les champs potentiels artificiels exerçant des forces sur les demandes des utilisateurs. D'un point de vue structurel, l'approche proposée n'est pas basée sur une organisation hiérarchique, ce qui signifie que tous les agents logiciels du système sont considérés comme égaux. La découverte et la sélection de services ne sont pas basées sur les informations stockées des recherches précédentes. Chaque SPN ne conserve que des informations concernant la structure actuelle du réseau local	Les résultats de la simulation indiquent que l'approche proposée maintien des performances et une évolutivité satisfaisante en tant que nombre de SPN
[86]	Assure l'exécution des services composites résultants en prenant en compte les changements de services au niveau de la découverte et de la sélection	Les résultats expérimentaux et les performances obtenues à partir d'essais poussés montrent l'efficacité et la faisabilité de l'approche proposée dans le cas d'un environnement à grande échelle.
[87]	Exploite le maximum de parallélisme (exécution des services IoT en parallèle chaque fois que les services sont invocables)	L'expérience a montré des résultats prometteurs
[88]	Une solution optimale globale a été utilisée pour accélérer le taux de convergence en imitant l'opération croisée de l'algorithme génétique (GA)	Le résultat de l'expérience de simulation montre que CMABC présentait une vitesse de convergence plus rapide et une meilleure précision de convergence que certains autres algorithmes d'optimisation intelligents

4. Discussion

Le (Tableau 3) nous montre que la plupart des approches sont dynamiques, au lieu de [57], [66], [69], [15] and [80]. Le (Tableau 4) montre que des approches comme [50], [51] , [59], [62] , [63], [65],[67], [70], [73], [75], [77], [79],[90] and [91] sont adaptatifs.

Le (Tableau 5) montre que seuls quelques auteurs mettent en œuvre un système d'identification et de résolution automatique des problèmes de défaillance et d'interaction, comme dans [50], [62], [63], [65], [70], [75], [79], [86] et [64]. Aussi que toutes les approches satisfont aux critères d'indépendance et d'extensibilité et d'après le (Tableau 6), la plupart d'entre elles sont distribuées/décentralisation en dehors [49], [67], [69] et [71], [77], [82], [87] and [64].

Concernant l'optimisation de la composition du service dans le (Tableau 7), la plupart des approches ont utilisé des paramètres non fonctionnels pour améliorer le comportement de la composition et de ses composants, à part [66], [74], [75], [77], [81], [84] et [85]. Enfin, la plupart des auteurs ont utilisé différentes méthodes pour réaliser leurs approches et chacune d'entre elles a ses avantages et ses inconvénients.

En se basant sur le (Tableau 8) et (Tableau 9), nous avons remarqué que la plupart des approches ne satisfont pas à tous nos critères et certains aspects fondamentaux sont couramment ignorés ou mal discutés, tels que l'adaptabilité, la protection, la sécurité, l'identification et la résolution automatique des pannes et les problèmes d'interaction. De plus, au cours de cette étude comparative nous avons remarqué que la plupart des recherches se concentrent sur la façon de détecter s'il y a un échec/changement de contexte que sur le QUOI FAIRE (la réaction) après que des changements inattendus se soient produits.

De cette comparaison des approches présentées qui a exposé les principaux avantages et inconvénients de chaque travail, nous déduisons que les approches comme ; *Approche réseau de Petri* [50], *Approche contextuelle* [56] (qui utilise la méthode de raisonnement OWL), *l'approche Middleware* [62] & [63], *L'approche Event-Aware Framework* [86] et [75] qui représente une nouvelle approche d'exécution adaptative basée sur un modèle de sélection de service réactif, sont les meilleures approches adaptées à l'environnement IoT, cependant toutes ces approches ne satisfassent pas les critères des protocoles de confiance, de protection et de sécurité.

L'approche sociale basée sur la confiance adaptative [51] [59], a démontré qu'avec sa conception de protocole de confiance adaptative, l'application est capable d'approcher les performances idéales lors de la convergence et peut surpasser de manière significative la non-confiance méthode basé sur un protocole de sélection aléatoire, cependant l'auteur ne s'est pas concentré sur les autres aspects gérés par les trois approches mentionnées ci-dessus.

5. Autres études comparatives dans la littérature

À notre connaissance, seules deux revues [93] & [94] ont été produites dans le domaine de la composition des services pour l'Internet des objets. Les autres travaux les plus proches qui ont été réalisés se situent du côté de la composition des services Web.

Dans [93], une enquête sur les modèles de calcul de la confiance des systèmes IoT pour la gestion des services a été présentée, où les auteurs classent les modèles de calcul de la confiance existants pour la gestion des services dans les systèmes IoT sur la base de cinq dimensions de conception pour un modèle de calcul de la confiance : la confiance composition, propagation de confiance, agrégation de confiance, mise à jour de confiance et formation de confiance. Ils mettent en évidence l'efficacité des mécanismes de défense contre les attaques malveillantes du modèle de chaque dimension et résument les techniques de calcul de confiance les plus et les moins visitées dans la littérature et donnent un aperçu de l'efficacité des techniques de calcul de confiance appliquées aux systèmes IoT. Enfin, les auteurs identifient les principaux inconvénients de la recherche sur le calcul de la confiance IoT et suggèrent des orientations de recherche futures. Par rapport à notre étude, l'article se concentrait uniquement sur le calcul de la confiance et ne gérait pas les autres aspects de la composition des services dans l'IoT.

Dans [94], une revue de la littérature sur la composition des services dans l'IoT a été présentée, où les auteurs introduisent des problèmes de composition de services et de nouveaux défis de recherche. Ils fournissent un examen des études préliminaires sur la composition des services dans l'IoT classés par les technologies d'objets intelligents (RFID, WSN et autres) et les approches de composition (composition SOC et Mashup). Cet article explique la faisabilité du futur IoT full-IP avec des protocoles Web en temps réel pour énoncer formellement le problème de la composition des services pour les objets intelligents IP ainsi que ses exigences (contrainte de ressources, faible consommation, efficacité énergétique, services axés sur les données/événements, Asynchrone, Découverte, Vérification des exigences de gestion et Sensibilisation à la QoS) et certains cas d'utilisation. Cependant, toute leur étude et le problème abordé dans cet article étaient basés sur des protocoles Web.

Dans [31], les auteurs ont étudié les aspects les plus importants de l'IoT en mettant l'accent sur ce qui est fait et quels sont les problèmes qui nécessitent des recherches plus approfondies. Dans une autre enquête sur les technologies, les applications et les défis de recherche pour l'IoT [30], les auteurs ont présenté les domaines d'application les plus pertinents, et un certain nombre de défis de recherche ont été identifiés. Toujours dans [95], les auteurs ont fourni une brève description de l'état de l'art actuel de l'IoT, avec un accent particulier sur les concepts et technologies liés à la mobilité et aux communications sans fil et réseautage. Cependant, ils ne se sont pas concentrés sur la composition des services.

Dans [96], une revue des articles non techniques sur l'IoT a été présentée. Les auteurs examinent l'état actuel des discours scientifiques et des modèles de processus de hiérarchie analytique appliquée pour évaluer les priorités des futures recherches sur l'IoT. Cette étude a analysé de manière sélective les options possibles pour la recherche future, et elle illustre que la recherche sur l'IoT s'est concentrée que sur certains domaines avec des sujets et des méthodologies limités. Aucun détail sur la composition du service n'est discuté.

Dans [97], les auteurs ont analysé le phénomène IoT d'un point de vue évolutif. Ils visent à fournir une analyse minutieuse des technologies qui ont contribué à la naissance de l'IoT et à sa croissance au fil du temps. À la suite de cette analyse, trois générations d'IoT ont été identifiées : la première c’est les objets étiquetés ; la second c’est l'interconnexion des choses à travers les technologies Web ; la troisième c’est les objets sociaux de la représentation sémantique des données et du nuage des choses. Cependant, l'objectif principal de l'article est de

présenter les étapes évolutives qui ont caractérisé le développement de l'IoT, ainsi que les motivations de leur déclenchement. Aucun détail sur la composition du service n'est discuté.

Dans [98], un modèle d'architecture IoT a été décrit dans lequel les objets, les personnes et les services cloud sont combinés pour faciliter les tâches applicatives. Les auteurs décrivent les composants clés de l'architecture en pensant à l'application des villes intelligentes. En outre, un aperçu des plates-formes logicielles IoT et des technologies habilitantes, certains des défis IoT provenant de l'immaturité des logiciels et du matériel IoT ont été décrits (disponibilité, interopérabilité, évolutivité, gestion des performances, sécurité et confidentialité, analyse des big data, services cloud et Smart conception des appareils). Cependant, aucun détail sur la composition du service n'est discuté.

Dans [99], une enquête sur l'IoT a été présentée, où l'auteur se concentre sur les architectures spécifiques à un domaine des applications IoT. Dans cet article, le contexte et la définition de l'IoT sont donnés avec plusieurs domaines clés dans lesquels des travaux de recherche basés sur l'IoT sont en cours. L'auteur a également présenté des analyses détaillées des défis de la recherche avec le graphique résultant, et met en évidence les défis et les opportunités de recherche possibles pour les futurs chercheurs en IoT qui travailleraient dans l'architecture ainsi que dans l'IoT dans son ensemble. Cependant, l'auteur ne s'est pas concentré sur la composition des services.

Dans [100], un aperçu des schémas de sécurité de la couche physique de faible complexité qui conviennent à l'IoT est présenté, où les auteurs proposent une étude complète des avancées actuelles et des défis restants dans le codage de la confidentialité des ressources limitées et la génération de clés secrètes adaptées pour les applications dans l'Internet des objets. L'état de l'art de la sécurité de la couche physique pour les réseaux de capteurs est passé en revue, suivi d'un aperçu des techniques de sécurité des réseaux de communication. Ensuite, les auteurs identifient les techniques de sécurité les plus efficaces en énergie et les moins complexes qui conviennent le mieux aux applications de détection IoT. Ceci est suivi d'une discussion sur les schémas candidats de faible complexité pour la sécurité des communications. L'article se termine par une discussion sur les questions de recherche ouvertes et les pistes de travaux futurs. Aucuns détails sur la composition du service n'est discuté.

Dans [101], une étude comparative des approches existantes pour la composition de services Web a été décrite, les auteurs les comparent entre elles en ce qui concerne certaines exigences clés. En outre, les auteurs donnent un aperçu des méthodes et approches de composition des services, et ils discutent des avantages et des inconvénients de chaque article étudié.

Dans [102], un aperçu et une comparaison des progrès récents dans la composition des services Web ont été fournis. Les auteurs classent ces approches en trois catégories (basées sur les workflows, basées sur XML et basées sur les ontologies). Dans chaque catégorie, ils présentent et comparent des approches sélectionnées sur la base de certains critères (comme la qualité de service, l'évolutivité et l'exactitude).

Dans [103], une étude de différentes techniques de composition de services Web basées sur des plateformes et des Framework de composition actuellement existants a été réalisée. Les auteurs comparent ces techniques en fonction des avantages et des inconvénients. Ils discutent

de ce qui rend la composition des services Web si spéciale et en tirent des défis pour la communauté des affaires.

Dans [104], un aperçu de l'état de l'art de la recherche dans les approches de composition du Web sémantique a été discuté. Les auteurs ont comparé et classé ces approches en deux catégories ; approches de composition Web sémantique avec prise en charge QoS et approches de composition Web sémantique sans prise en charge QoS. Six critères d'évaluation ont été proposés dans le but de comparer systématiquement les deux catégories d'approches à la composition des services du Web sémantique ; évolutivité, non-déterminisme, aspect dynamique, adaptabilité, indépendance du domaine, exactitude, capacité sémantique et conscience de la QoS. Dans chaque catégorie, ils donnent l'introduction et la comparaison des approches sélectionnées. L'objectif de l'article était d'identifier la meilleure approche pouvant être utilisée pour la composition de services Web sémantiques.

Dans [105], un aperçu des progrès récents réalisés dans les approches de découverte de services Web a été fourni. De plus, les auteurs ont effectué une analyse de ces approches et ont souligné certains de leurs mérites ainsi que leurs lacunes. Après avoir introduit une taxonomie qui catégorise les systèmes de découverte de services Web selon différents points de vue, ils ont présenté les avantages et les inconvénients de chaque groupe.

Dans [103], une étude des différentes techniques de sélection de services Web disponibles dans la littérature a été réalisée. Les auteurs comparent ces méthodologies de sélection de services Web en utilisant différents paramètres (QoS et fiabilité).

Dans [106], un Framework articulé pour analyser et comparer les approches de composition de services Web a été proposé. Les auteurs ont proposé une taxonomie composée des dimensions (langage, réutilisation des connaissances, automatisation, support d'outils, plateforme d'exécution, utilisateur cible) qui caractérisent et comparent les approches de composition de services. Ils ont fourni une analyse systématique des approches de composition de services les plus représentatives en les évaluant et en les classant par rapport à la taxonomie proposée.

Dans [107], une revue des enquêtes de composition automatique des services Web a été présentée. Les auteurs ont utilisé les préoccupations de recherche précédemment identifiées par eux pour organiser leur présentation et indiquer la portée de chaque enquête automatique sur la composition des services Web incluse.

Dans [108], une revue systématique de la littérature sur la composition des services de cloud computing a été proposée. Les auteurs ont divisé la recherche en quatre groupes principaux en fonction des approches de résolution de problèmes et en identifiant les paramètres de qualité de service étudiés, les objectifs visés et les environnements en développement, des résultats bénéfiques et des statistiques sont obtenus qui peuvent contribuer aux recherches futures.

6. Conclusion

Nous concluons cette comparaison en identifiant que la protection, la sécurité, l'identification et la résolution automatique des problèmes de défaillance et d'interaction, sont les inconvénients

majeurs de toutes les approches. Le manque d'implémentations disponibles est un problème, car ces aspects sont très importants. Cela ouvre des opportunités pour l'amélioration des stratégies existantes d'approches de composition de services pour l'IoT, ainsi que pour les recherches futures dans le domaine. Nous pensons que les mécanismes de composition de services pour l'IoT peuvent bénéficier d'une recherche plus approfondie sur les thèmes susmentionnés.

Pour résumer, comme suit, nous identifierons les lacunes dans la composition de service dans l'IoT et suggérerons des orientations de recherche futures.

- Il existe plusieurs problèmes ouverts qui nécessitent des efforts de recherche et de développement supplémentaires afin d'exploiter pleinement le potentiel de l'IoT, l'un de ces défis fait référence à l'énorme quantité d'énergie consommée lors de la diffusion d'informations produites par un grand nombre d'appareils interconnectés à ressources limitées, qui doivent être stockées, traitées et présentées sous une forme efficace et facilement interprétable.
- De plus, l'IoT est extrêmement vulnérable aux attaques pour plusieurs raisons. Premièrement, ses composants passent souvent la plupart du temps sans surveillance ; et ainsi, il est facile de les attaquer physiquement. Deuxièmement, la plupart des communications sont sans fil, ce qui rend l'écoute clandestine extrêmement simple. Enfin, la plupart des composants IoT se caractérisent par de faibles capacités en termes de ressources énergétiques et informatiques (c'est notamment le cas pour les composants passifs) et ne peuvent donc pas mettre en œuvre des schémas complexes soutenant la sécurité.
- De nombreuses informations privées sur une personne peuvent être collectées à son insu. Le contrôle de la diffusion de toutes ces informations est impossible avec les techniques actuelles.
- Toutes les informations collectées sur une personne par l'IoT peuvent être conservées indéfiniment à mesure que le coût du stockage diminue. En outre, les techniques d'exploration de données peuvent être utilisées pour récupérer facilement n'importe quelle information, même après plusieurs années.
- Il existe plusieurs efforts de normalisation mais ils ne sont pas intégrés dans un cadre global.
- Un autre problème que l'on peut également rencontrer est que certaines des données doivent être traitées en temps réel dans l'IoT. La raison pourrait être que le capteur ne dispose pas de suffisamment de stockage pour conserver toutes les données collectées ou que le capteur doit prendre une décision par lui-même.

Dans ce qui suit, on se focalisera le plus sur l'aspect du traitement de donnée en temps réel avec notre architecture proposé Fog-IoT en cinq couches et notre algorithme P-MPGA sensible aux QoS, qui gère la sélection des services IoT de manière adaptative grâce à son module de surveillance. Ce travail va être plus détaillé dans le chapitre qui suit.

Deuxième partie

Contribution et validation

CHAPITRE 4

Notre approche proposée

Sommaire

Dans ce chapitre, nous présentons notre architecture en cinq couches basées sur le concept Fog-IoT, le modèle QoS utilisé et notre algorithme génétique multi-population sensible aux QoS P-MPGA. Cette architecture Fog-IoT nous a permis de décentraliser le processus de la composition de services comme le montre la (Figure 16) et de bénéficier de plusieurs avantages (qu'on va invoquer ci-dessous). Aussi, notre algorithme adaptatif P-MPGA permet d'utiliser les ressources de la manière la plus efficace pour l'optimisation en maximisant ou minimisant certains attributs QoS [16].

1. Approches proches de notre travail dans la littérature

D'après notre étude dans le chapitre 3, où nous avons évalué les approches de composition de services les plus populaires dans l'IoT en se basant sur plusieurs critères, nous avons vu que très peu de travaux ont présenté une solution de composition de services adaptative gérant les attributs de QoS, de plus dans le domaine de la santé, qui est l'un des plus difficiles et délicats car il concerne la précieuse vie humaine, nous pouvons citer les travaux suivants :

Dans [70], [90], une composition de service adaptative a été proposée pour un système d'assistance à la vie (ALS) [109], le système est censé soutenir les personnes atteintes d'une maladie chronique ou celles qui en ont besoin pour un suivi médical constant (par exemple, les personnes âgées) afin qu'elles puissent continuer à vivre de manière autonome à la maison. L'architecture proposée utilise un mécanisme de gestion des événements pour propager les informations contextuelles des appareils IoT, une fonction de remplacement de service dynamique est utilisée en cas de panne de service, ce qui rend le système adaptatif et un candidat potentiel pour les environnements IoT où des pannes de service fréquentes sont observées, cependant, le la latence due à la gestion des événements augmente linéairement avec l'augmentation du nombre de capteurs. De plus, il n'y a pas de gestion de la qualité de service et leur système de surveillance ne signale que les pannes matérielles et ne détecte pas les données de capteur incorrectes ou les pannes logicielles.

Dans [110], une sélection de services IoT basée sur la QoS est proposée pour le système de surveillance Electro Cardio Graphy (ECG). Les auteurs ont considéré le cadre IoT comme une composition de trois composants principaux : les objets, l'informatique et la communication, les attributs de QoS sont associés à chacun des trois composants. Le cadre de sélection évalue l'importance relative des critères de QoS pour classer les services IoT. Cependant, seule la partie sélection de la composition du service a été présentée, aucun cadre d'adaptabilité n'a été illustré.

En dehors du domaine de la santé, d'autres approches adaptatives ont été proposées dans les approches qui vont suivre.

Dans [50], un nouvel algorithme de composition de service basé sur le réseau de Petri a été proposé. Pour gérer le changement dynamique des environnements, les auteurs ont proposé un algorithme de surveillance pour surveiller le système IoT de manière moins coûteuse. Une évaluation QoS est utilisée, mais ne concerne que trois propriétés de qualité, à savoir la fiabilité, le temps de réponse et le coût. Les résultats expérimentaux ont prouvé la solidité et l'exactitude des algorithmes, mais il n'y a eu aucune évaluation dans le système de service du monde réel dans l'IoT.

Dans [62], [63], une approche middleware pour la composition de services IoT a été proposée. Les auteurs utilisent un système de surveillance avec seulement quelques ressources de surveillance pour surveiller et assurer le fonctionnement et la robustesse du système. Chaque service composant est représenté par un agent, qui est responsable du maintien des informations QoS (prix, temps, fiabilité, degré de réputation, disponibilité) de chaque service. Les expériences ont évalué l'exactitude et la robustesse des modèles et des algorithmes, mais il n'y a eu aucune évaluation dans le système de service du monde réel dans l'IoT.

Dans [64], une approche probabiliste de composition de services a été proposée. Pour surveiller le système et rendre cette approche adaptative, les auteurs utilisent des candidats alternatifs en cas de défaillance (lorsque la probabilité de succès du service sélectionné n'est pas très élevée). Aussi, lors de la sélection du service, les auteurs décrivent et analysent les attributs QoS (fiabilité et coût). Cependant, cette approche est centralisée et ne traite pas la contrainte du temps réel.

Dans [65], une approche adaptative pour la composition de services dans l'IoT a été proposée. Le mécanisme de sélection proposé est basé sur une technique d'optimisation pour détecter les situations où certaines exigences de composition ne sont pas satisfaites ou certains services deviennent indisponibles ou lorsque des défaillances/exceptions se produisent. Cette méthode essaie de réduire le besoin d'intervention humaine dans la reconfiguration de la composition. Cependant, le système de surveillance est centralisé et utilise une liste de services indisponibles. L'auteur gère également certains attributs QoS (temps de réponse, fiabilité, disponibilité, localisation, niveau de batterie et réputation) mais c'est extrêmement exigeant et demande un calcul intensif.

Dans [82], une approche de composition de service adaptative pour l'IoT est proposée. La génération du schéma de composition est partiellement réalisée au moment de l'exécution à l'aide de services abstraits fournis au moment de la conception, ce qui permet une flexibilité et une adaptabilité sans avoir à créer un ensemble de services à partir de zéro. Cependant, aucun

système de suivi pour l'identification et la résolution n'a été proposé. De plus, aucune gestion spécifique de la QoS n'a été citée dans ce travail.

Dans [86], une approche de composition de service sensible aux événements pour l'IoT est proposée. Les auteurs présentent un système de surveillance avec un mécanisme de dérogation basé sur une stratégie d'événement, dans cette approche, la détection des changements est effectuée au niveau de la découverte pour économiser plus de temps et d'énergie lors de la sélection et pour s'assurer que les services fonctionneront avec succès, certains changements peuvent se produire à l'exécution. L'apprentissage bayésien est utilisé pour mettre à jour la QoS dynamique (disponibilité, temps de réponse et temps de réponse) et la recomposition automatique du service en cas de panne du service. Cependant, il n'y a pas eu d'évaluation dans le système de service du monde réel dans l'IoT.

Dans [75], une approche de composition de service adaptative pour l'IoT est proposée. L'approche gère efficacement les changements de services pendant l'exécution, une adaptation est effectuée dès que possible et en parallèle du processus d'exécution, réduisant ainsi les temps d'arrêt, augmentant les chances de réussite de la récupération et fournissant la solution la plus optimale en fonction de l'état actuel de l'environnement. Cependant, aucune gestion de QoS spécifique n'a été citée dans cette version actuelle de ce travail.

Dans [111], une nouvelle composition de services multi-objectifs pour l'IoT est proposée. Les auteurs au lieu d'optimiser un seul objet, ils prennent la maximisation de la QoS et la minimisation des coûts comme deux objets. Pour résoudre ce problème d'optimisation complexe, un algorithme de colonie d'abeilles artificielles multi-objectifs est utilisé. Cependant, la méthode proposée est conçue pour la composition des services cloud, et aucun cadre d'adaptabilité n'a été illustré.

Dans [112], [113], une méthode de sélection de service hybride pour l'IoT a été proposée. Les auteurs utilisent une approche d'optimisation et des attributs de QoS pour trouver le meilleur service candidat, le problème de sélection de service IoT est transformé en un problème d'optimisation à objectif unique adoptant une méthode de pondération simple. L'expérimentation montre que l'algorithme proposé peut satisfaire les besoins de l'utilisateur, cependant, la consommation d'énergie n'est pas prise en compte dans la sélection du service et aucun cadre d'adaptabilité n'a été illustré.

En comparaison avec les travaux ci-dessus, nous avons identifié certains attributs de QoS associés aux composants IoT qui quantifient et analysent au mieux les services offerts par les fournisseurs de services IoT. De plus, des problèmes d'architecture IoT-cloud nous ont amenés à utiliser l'architecture à 5 couches implémentée sur un système Fog-IoT. Nous pensons que cet aspect important du cadre de sélection et d'architecture de service doit être nécessairement défini avant de concevoir un algorithme de composition de service.

2. Architecture en cinq couches

Il n'y a pas d'accord unique ou général sur l'architecture IoT qui soit accepté dans le monde de la recherche [114]. De nombreuses architectures différentes ont été proposées et selon certains chercheurs, l'architecture IoT comporte trois couches [7]–[9], mais certains chercheurs

privilégient l'architecture à quatre couches [10]. Les auteurs pensent qu'en raison du développement de l'IoT, l'architecture à trois couches est basique et ne peut pas répondre aux exigences des applications.

L'architecture à quatre couches a joué un rôle important dans le développement de l'IoT, mais en raison d'un autre défi de l'IoT concernant la sécurité et la confidentialité, l'architecture à cinq couches [11], [12] a également été proposée. Cette architecture comporte trois couches comme les architectures précédentes dont les noms sont couche de perception, couche de transport et couche d'application. Elle a également deux couches supplémentaires. Les noms de ces nouvelles couches proposées sont Couche de traitement et Couche « Business ». On considère que cette architecture a la capacité de répondre aux exigences de l'IoT, elle a également la capacité de sécuriser les applications IoT. Voici les 5 couches de cette architecture :

- *Couche « Business »* : Gère l'ensemble du système, y compris les applications ainsi que la confidentialité des utilisateurs. (Gestion & suivi)
- *Application* : Responsable de la fourniture de services spécifiques à l'application pour l'utilisateur (Interface).
- *Traitement* : Stocke, analyse et traite l'énorme quantité de données à l'aide de bases de données etc.… selon les besoins des utilisateurs.
- *Transport* : Transfert des données des capteurs entre les différentes couches via des réseaux tels que sans fil, 3G, LAN, Bluetooth, RFid et NFC
- *Perception* : Détecte et collecte les informations sur l'environnement (par des capteurs…)

La hiérarchie de toutes les architectures de couches proposées de l'Internet des objets (IoT) est illustrée à la (Figure 13), qui montre les architectures de couche IoT composées respectivement de trois couches, quatre couches et cinq couches. Nous considérons également que l'architecture à 5 couches peut répondre aux exigences de l'IoT et satisfaire au maximum les critères cités dans le chapitre 3.

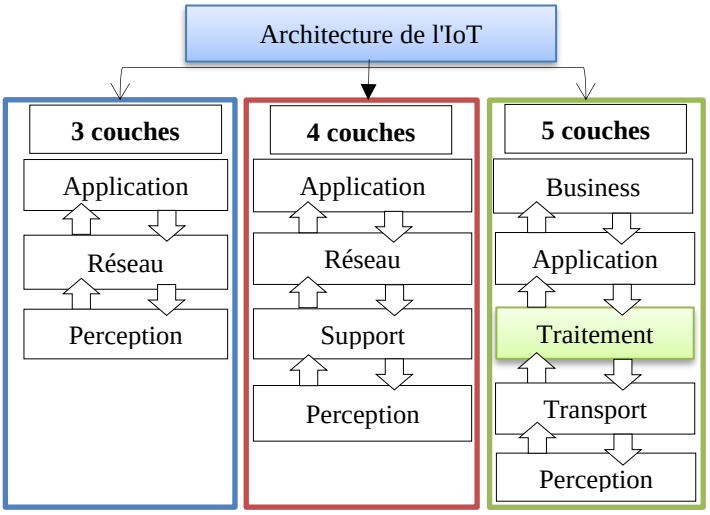


Figure 13 Architectures en couches IoT dans la littérature

Dans notre travail, nous nous intéressons plus particulièrement à la couche de traitement (Figure 14), cette couche est également appelée couche middleware. Elle collecte les informations envoyées par la couche de transport, en effectuant un traitement sur les

informations collectées. Elle est chargée de supprimer les informations supplémentaires qui n'ont pas de sens et d'extraire les informations utiles. Toutefois, cela supprime également le problème des BigData dans l'IoT, ou une grande quantité d'informations est reçue, ce qui peut affecter les performances de l'IoT.

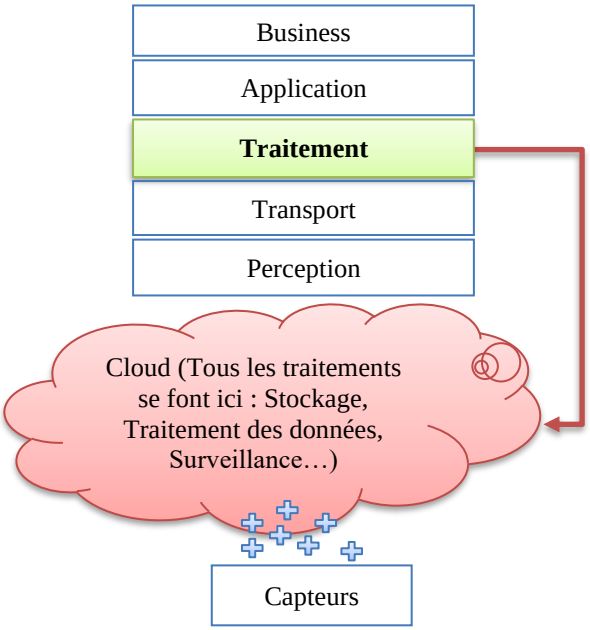


Figure 14 La relation entre la couche de traitement et le Cloud

En général, les informations sont envoyées du stockage local au stockage en cloud où tous les objets envoient les informations collectées (Figure 13). Enfin, en utilisant les informations collectées, les mesures appropriées sont prises. Il n'est pas nécessaire que l'action soit toujours effectuée à l'aide de ces informations, mais nous pouvons également gérer et contrôler à distance des objets et des machines et utiliser les informations pour conserver des enregistrements pour une utilisation future.

L'architecture IoT-Cloud est une méthode pratique pour quelques appareils. Mais à mesure que le nombre d'appareils utilisant le cloud augmente, l'utilisation de sa bande passante, de sa latence augmente également, et cela implique également :

- Ne rentre pas dans les systèmes qui doivent recevoir des actions immédiates et en temps réel.
- Il y aura de la latence pendant les transferts de données.
- Risque de surcharge du Cloud (nombre d'objets en croissance...).
- Nécessite une bande passante élevée (trop de données à transmettre).
- Nécessite d'être toujours connecté à Internet.
- Problème de sécurité Cloud (Vie privée).
- Problème d'évolutivité.

Les problèmes rencontrés par l'architecture IoT-Cloud ont conduit au développement du Fog pour surmonter au mieux ces obstacles [13]. Le Fog Computing est une couche qui réside entre le service cloud et les appareils locaux. Les nœuds de Fog seront répartis géographiquement pour fournir des services à leurs appareils locaux requis. Chaque nœud pourra effectuer des tâches de calcul et pourra fournir des services liés aux données collectées dans la zone sous leur contrôle. En résumé, le Fog Computing représente une décentralisation du Cloud Computing (Figure 15).

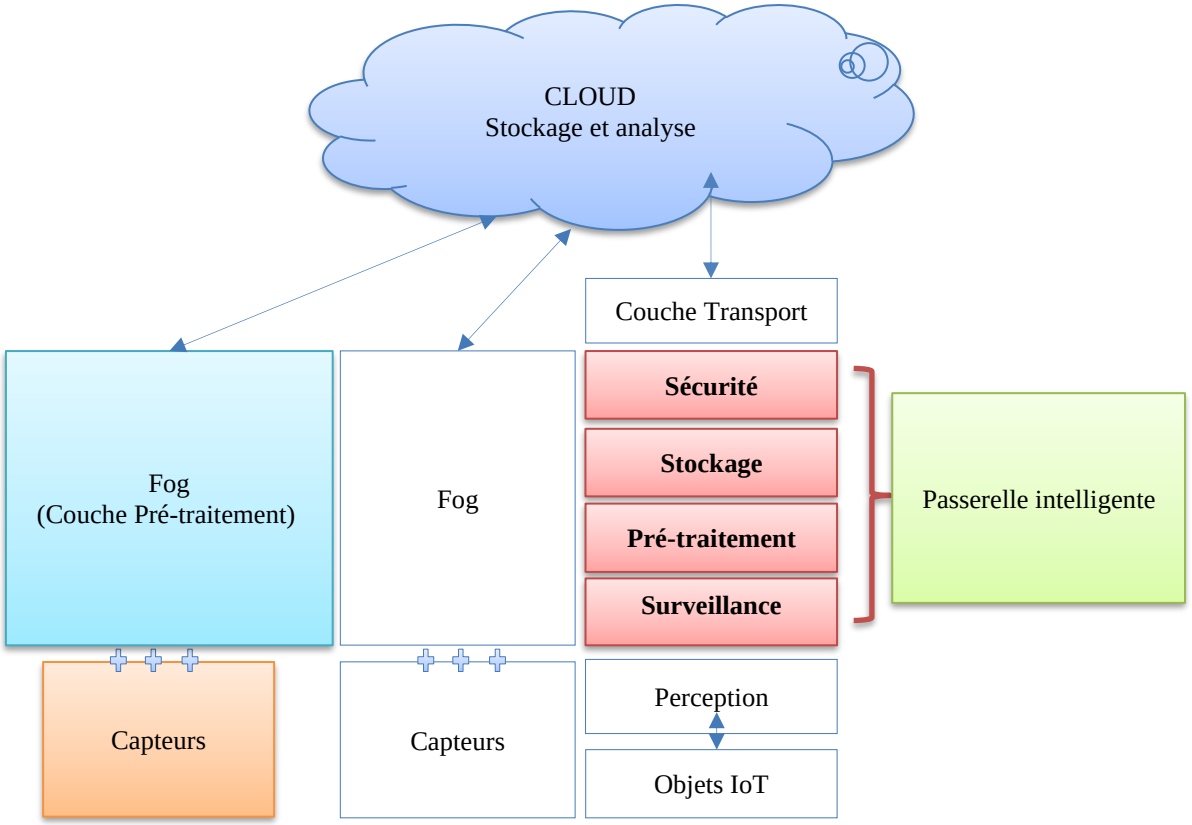


Figure 15 Notre Architecture Cloud-Fog-IoT adoptée après la défragmentation de la couche traitement

Comme nous l'avons dit dans les chapitres précédents, le secteur de la santé est l'un des plus difficiles et des plus délicats car il concerne la vie des personnes. L'IoT avec le Cloud Computing a amélioré la qualité de vie des patients ; cependant, cette architecture est souvent trop réductrice et inadaptée à de nombreuses applications de santé émergentes avec des exigences critiques. Le Fog Computing peut être la solution au problème [115], il permet des temps de réponse faibles et prévisibles, ce qui peut souvent faire la différence entre la vie et la mort des patients ; il garantit qu'au moins la partie la plus critique du service global est toujours disponible pour le patient, même en présence d'environnements hostiles avec une connectivité réseau intermittente ou inexistante au Cloud ; il protège les données sensibles liées à la santé en les stockant localement plutôt que de les envoyer dans le cloud via Internet.

L'utilisation du Fog computing et la défragmentation de la couches Traitement en quatre sous-couches (sécurité, stockage, pré-traitement & surveillance) comme le montre la (Figure 14) nous permet d'avoir les avantages suivants :

- Travaillez directement sur les réseaux locaux donc c'est plus rapide (faible latence).
- Pas besoin de consulter le cloud et d'être connecté à Internet.
- Interactions en temps réel (presque).
- Mobilité.
- Distribution et décentralisation.
- Localisation.
- Sécurité (Réduire le risque d'attaques).
- Système Intelligent et efficace (efficace).
- Seules les informations nécessaires sont envoyées au Cloud pour être analysées.

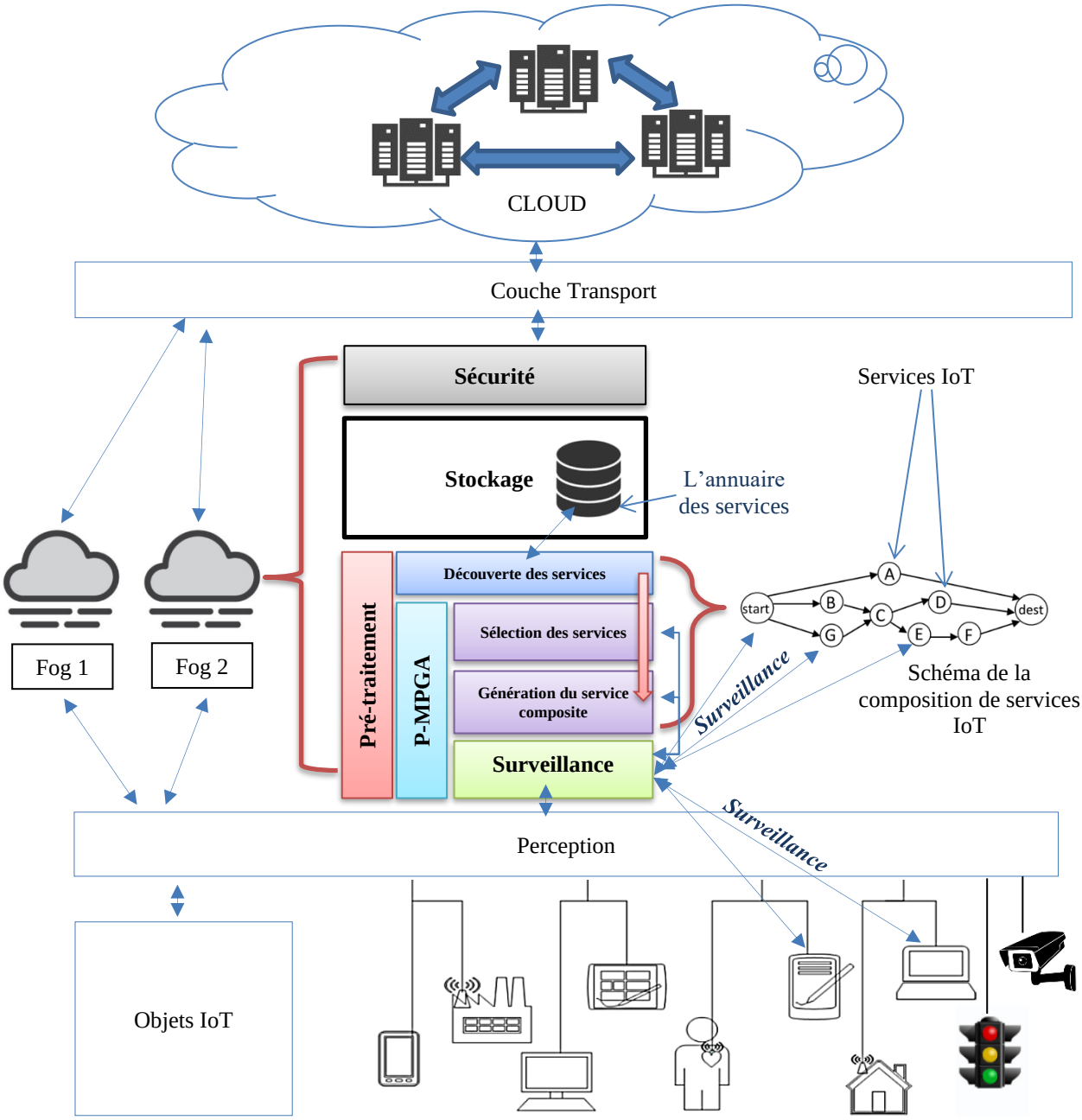


Figure 16 Notre architecture Fog-IoT adoptée en cinq couches pour la composition de services dans l'IoT (détaillée)

La (figure 16) détaille la couche « Pré-Traitement » de notre architecture adoptée Fog-IoT, tout le processus de la composition de service se fait dans cette couche, de la découverte de service jusqu'à l'exécution du service composite. La couche surveillance est responsable de surveiller tous les services disponibles pour gérer le changement dynamique des environnements IoT et assurer que notre approche proposée soit adaptative.

3. Le modèle QoS utilisé

Chaque objet de l'IoT peut fournir un certain nombre de services spécifiques. Certains services IoT peuvent être similaires ou identiques. Cependant, les performances QoS peuvent être différentes les unes des autres. Par conséquent, l'évaluation de la qualité de service (QoS) permet de différencier un service d'un autre. Ainsi, il aide les demandeurs de service à reconnaître le meilleur service IoT pour son application.

Pour traiter les services IoT séparément, l'exploration de ces trois composants de l'IoT ; l'objet, la communication et le calcul sont nécessaires, avec leurs métriques de qualité de service respectives [110]. Les attributs clés de la QoS dans l'IoT peuvent être dynamiques ou statiques [83]. Les services dans l'IoT sont liés au monde physique, donc les informations de localisation géographique des appareils affectent la satisfaction des utilisateurs, sans oublier que cet environnement dynamique, qui affecte la disponibilité de ces services [2]. Fournir un niveau acceptable de QoS est un problème important dans Fog-IoT [116] et pour la sélection de services du système de santé basé sur Fog-IoT, nous avons pris douze attributs QoS qui ont le plus d'impact sur ce cas d'étude, et ces attributs peuvent être modélisé comme suit :

3.1. Disponibilité (A)
C'est un pourcentage de temps, et indique quand le service est disponible.

3.2. Coût (C)
Représente le coût que l'utilisateur doit payer pour l'acquisition du service.

3.3. Documentation (D)
Mesure de la documentation (c.-à-d. balises de description)

3.4. Emplacement / Portée (L)
C'est la distance entre le service et la destination spécifiée par l'utilisateur.

3.5. Précision (P)
C'est la capacité des capteurs à mesurer l'écart obtenu dans la sortie lorsque le même signal ou les mêmes données sont mesurés à plusieurs reprises dans des conditions similaires [117]. Les auteurs dans [118] définissent la précision pour décrire exactement comment les informations de contexte fournies reflètent la réalité.

3.6. Fiabilité ou Reliability (R)
C'est le taux de réussite d'une demande de service. Les auteurs dans [119] définissent la Fiabilité pour indiquer dans quelle mesure le contexte peut être considéré comme crédible.

3.7. Temps de réponse (Rt)
C'est le temps moyen entre la soumission d'une demande de l'utilisateur et l'acceptation de la réponse du service.

3.8. Réputation (Rp)
C'est le score moyen de multi-évaluation du service par l'utilisation de n'importe quel système de notation.

3.9. Sécurité (S)

Représente le niveau de sécurité assuré par un service (authentification, cryptage, etc.). La sécurité permet de protéger les utilisateurs contre les accès interdits illégaux [120].

3.10. Classification de service (Sc)

La classification de service représente divers niveaux de qualité d'offre de service. Il existe quatre classifications de service : Platine (Haute qualité), Or, Argent et Bronze (Basse qualité).

3.11. Taux de réussite (Sr)

C'est le nombre de réponses/nombre de messages de demande (%)

3.12. Débit (T)

Le débit d'un service désigne la quantité de données traitables par unité de temps. Habituellement, le débit est donné en octets/sec et est interprété comme une dimension croissante

La différence de performance QoS entre chaque service d'information IoT est la clé de la sélection et de la composition du service, il est donc essentiel de faire une évaluation quantitative de la performance QoS de chaque service [121]. On suppose qu'un service composite est constitué de (n) services abstraits, notés : CS= {S1, S2, ..., Sn}. Pour chaque service abstrait Si, il possède plusieurs services concrets candidats, notés : Si= {Si1, Si2, ..., Sim} (m représente le nombre de services candidats de Si).

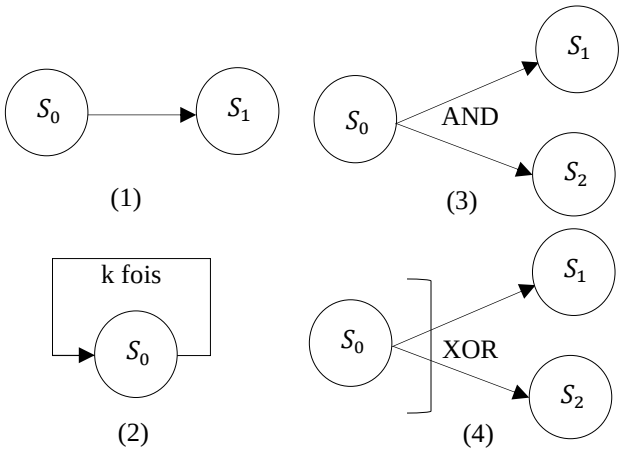


Figure 17 Modèles structurels en composition de service (séquence, boucle, parallèle-AND et parallèle-XOR) [122]

Il existe quatre modèles structurels dans la composition des services (Figure 17), c'est-à-dire séquence (1) , boucle (2), parallèle-AND (3) et parallèle-XOR (4) [122]. Nous identifions cinq types différents de types d'attributs qui sont catégorisés par leur fonction d'agrégation (Tableau 11).

Le modèle séquence (1) est un modèle d'exécution dans lequel les services sont exécutés les uns après les autres et il n'y a pas de chevauchement entre les périodes d'exécution des services. Pour calculer la valeur d'agrégation du temps de réponse et du coût d'exécution, chaque valeur de service doit être ajoutée l'une à l'autre. De plus, afin de calculer la valeur d'agrégation de la disponibilité et taux de réussite, les valeurs des services doivent être multipliées les unes par les autres car les services sont indépendants les uns des autres. La valeur globale de la réputation est obtenue en prenant la moyenne des valeurs de réputation des services.

Le modèle boucle (2) est une sorte de modèle séquentiel dans lequel le service s'exécute pendant des cycles limités.

Le modèle parallèle-AND est un modèle d'exécution dans lequel, après l'achèvement du service précédent, tous les services suivants sont exécutés simultanément. Notez que par exemple, pour obtenir un temps de réponse agrégé, nous utilisons la fonction Max, car tous les composants suivants sont exécutés simultanément.

Le modèle parallèle-XOR est un modèle d'exécution dans lequel, après l'achèvement du service précédent, l'un des services suivants s'exécute. Dans ce modèle, l'exécution de chaque composant est non déterministe ; par conséquent, pour calculer l'effet QoS d'agrégation de ce modèle, le pire des cas doit être calculé.

Tableau 10 Classes d'attributs QoS

Type	Fonction d'agrégation
Addition	$q^* = \sum_{i=1}^{n} q_i$
Multiplication	$q^* = \prod_{i=1}^{n} q_i$
Moyenne	$q^* = \frac{1}{n} . \sum_{i=1}^{n} q_i$
Minimum	$q^* = \min(q_1, q_2, \dots, q_n)$
Maximum	$q^* = \max(q_1, q_2, \dots, q_n)$

Les fonctions d'agrégation [15] du service composite pour les 12 attributs de QoS sont formulées dans le (Tableau 11) respectivement.

Tableau 11 Fonction d'agrégation des propriétés QoS basée sur le modèle du groupe de travail W3C [123]

	Attribue QoS	Séquentiel	Boucle	Parallèle-AND	Parallèle-XOR
f_1	**Disponibilité (A)**	$A_{cs} = \prod_{i=1 \,\Delta\, S_{ij} \in S_i}^{n} A_{ij}$	$A_{cs} = A_{ij}$	$A_{cs} = \prod_{i=1 \,\Delta\, S_{ij} \in S_i}^{n} A_{ij}$	$A_{cs} = \min(A_{1j}, A_{2j}, \dots, A_{ij})$
f_2	**Coût (C)**	$C_{cs} = \sum_{i=1 \,\Delta\, S_{ij} \in S_i}^{n} C_{ij}$	$C_{cs} = k * C_{ij}$	$C_{cs} = \sum_{i=1 \,\Delta\, S_{ij} \in S_i}^{n} C_{ij}$	$C_{cs} = \max(C_{1j}, C_{2j}, \dots, C_{ij})$
f_3	**Documentation (D)**	$D_{cs} = \prod_{i=1 \,\Delta\, S_{ij} \in S_i}^{n} D_{ij}$	$D_{cs} = D_{ij}$	$D_{cs} = \prod_{i=1 \,\Delta\, S_{ij} \in S_i}^{n} D_{ij}$	$D_{cs} = \min(D_{1j}, D_{2j}, \dots, D_{ij})$
f_4	**Emplacement (L)**	$L_{cs} = \sum_{i=1 \,\Delta\, S_{ij} \in S_i}^{n} L_{ij}$	$L_{cs} = k * L_{ij}$	$L_{cs} = \sum_{i=1 \,\Delta\, S_{ij} \in S_i}^{n} L_{ij}$	$L_{cs} = \min(L_{1j}, L_{2j}, \dots, L_{ij})$
f_5	**Précision (P)**	$P_{cs} = \prod_{i=1 \,\Delta\, S_{ij} \in S_i}^{n} P_{ij}$	$P_{cs} = P_{ij}$	$P_{cs} = \prod_{i=1 \,\Delta\, S_{ij} \in S_i}^{n} P_{ij}$	$P_{cs} = \min(P_{1j}, P_{2j}, \dots, P_{ij})$
f_6	**Fiabilité (R)**	$R_{cs} = \prod_{i=1 \,\Delta\, S_{ij} \in S_i}^{n} R_{ij}$	$R_{cs} = R_{ij}$	$R_{cs} = \prod_{i=1 \,\Delta\, S_{ij} \in S_i}^{n} R_{ij}$	$R_{cs} = \min(R_{1j}, R_{2j}, \dots, R_{ij})$

	Attribue QoS	Séquentiel	Boucle	Parallèle-AND	Parallèle-XOR
f_7	**Temps de réponse (Rt)**	$Rt_{cs} = \sum_{i=1 \, \Delta \, S_{ij} \in S_i}^{n} Rt_{ij}$	$Rt_{cs} = k * Rt_{ij}$	$Rt_{cs} = \max(Rt_{1j}, Rt_{2j},, Rt_{ij})$	$Rt_{cs} = \max(Rt_{1j}, Rt_{2j},, Rt_{ij})$
f_8	**Réputation (Rp)**	$Rp_{cs} = \frac{1}{n}. \sum_{i=1 \, \Delta \, S_{ij} \in S_i}^{n} Rp_{ij}$	$Rp_{cs} = Rp_{ij}$	$Rp_{cs} = \frac{1}{n}. \sum_{i=1 \, \Delta \, S_{ij} \in S_i}^{n} Rp_{ij}$	$Rp_{cs} = \min(Rp_{1j}, Rp_{2j}, \dots, Rp_{ij})$
f_9	**Sécurité (S)**	$S_{cs} = \min(S_{1j}, S_{2j}, \dots, S_{ij})$	$S_{cs} = S_{ij}$	$S_{cs} = \min(S_{1j}, S_{2j}, \dots, S_{ij})$	$S_{cs} = \min(S_{1j}, S_{2j}, \dots, S_{ij})$
f_{10}	**Classification des services (Sc)**	$Sc_{cs} = \prod_{i=1 \, \Delta \, S_{ij} \in S_i}^{n} Sc_{ij}$	$Sc_{cs} = Sc_{ij}$	$Sc_{cs} = \prod_{i=1 \, \Delta \, S_{ij} \in S_i}^{n} Sc_{ij}$	$Sc_{cs} = \min(Sc_{1j}, Sc_{2j}, \dots, Sc_{ij})$
f_{11}	**Taux de réussite (Sr)**	$Sr_{cs} = \prod_{i=1 \, \Delta \, S_{ij} \in S_i}^{n} Sr_{ij}$	$Sr_{cs} = Sr_{ij}$	$Sr_{cs} = \prod_{i=1 \, \Delta \, S_{ij} \in S_i}^{n} Sr_{ij}$	$Sr_{cs} = \min(Sr_{1j}, Sr_{2j}, \dots, Sr_{ij})$
f_{12}	**Débit (T)**	$T_{cs} = \min(T_{1j}, T_{2j}, \dots, T_{ij})$	$T_{cs} = T_{ij}$	$T_{cs} = \min(T_{1j}, T_{2j}, \dots, T_{ij})$	$T_{cs} = \min(T_{1j}, T_{2j}, \dots, T_{ij})$

Remarque : k est le nombre de répétition de la boucle.

Le problème est de trouver une composition optimale des services, avec un faible coût et temps de réponse, une haute classification des services, disponibilité, documentation, précision, fiabilité, réputation, taux de réussite, débit et sécurité en en moins de temps que possible, compte tenu de l'emplacement des services.

Par conséquent, en accordant un poids égal à chaque propriété de QoS décrite dans le (Tableau 11) ; Par conséquent, le problème de composition de service consiste à optimiser les 12 objectifs suivants.

- **Disponibilité (A)** : il est censé être le plus maximal que possible.
$$maximiser \, \boldsymbol{f_1} = aggregation \, (A) \quad (1)$$

- **Coût (C)** : il est censé être aussi minimal que possible.
$$minimiser \, \boldsymbol{f_2} = aggregation \, (C) \quad (2)$$

- **Documentation (D) :** il est censé être le plus maximal que possible.
$$maximiser \, \boldsymbol{f_3} = aggregation \, (D) \quad (3)$$
- **Emplacement (L) :** il est censé être le plus maximal que possible.
$$maximiser \, \boldsymbol{f_4} = aggregation \, (L) \quad (4)$$

- **Précision (P) :** il est censé être le plus maximal que possible.
$$maximiser \, \boldsymbol{f_5} = aggregation \, (P) \quad (5)$$

- **Fiabilité (R) :** il est censé être le plus maximal que possible.
$$maximiser \, \boldsymbol{f_6} = aggregation \, (R) \quad (6)$$

- **Temps d'exécution (Rt)** : il est censé être aussi minimal que possible.
$$minimiser \, \boldsymbol{f_7} = aggregation \, (Rt) \quad (7)$$

- **Réputation (Rp) :** il est censé être le plus maximal que possible.

$$maximiser\ \boldsymbol{f_8} = aggregation\ (Rp) \quad (8)$$

- **Sécurité (S) :** il est censé être le plus maximal que possible.

$$maximiser\ \boldsymbol{f_9} = aggregation\ (S) \quad (9)$$

- **Classification des services (Sc) :** il est censé être le plus maximal que possible.

$$maximiser\ \boldsymbol{f_{10}} = aggregation\ (Sc) \quad (10)$$

- **Taux de réussite (Sr) :** il est censé être le plus maximal que possible.

$$maximiser\ \boldsymbol{f_{11}} = aggregation\ (Sr) \quad (11)$$

- **Débit (T) :** il est censé être le plus maximal que possible.

$$maximiser\ \boldsymbol{f_{12}} = aggregation\ (T) \quad (12)$$

Les 12 fonctions objectives ci-dessus peuvent être combinées en une seule fonction objective « f », comme suit :

$$f = \text{maximiser}(w_1 f_1 + w_3 f_3 + w_4 f_4 + w_5 f_5 + w_6 f_6 + w_8 f_8 + w_9 f_9 + w_{10} f_{10} + w_{11} f_{11} + w_{12} f_{12} - w_2 f_2 - w_7 f_7) \quad (13)$$

Ici, w_i est le poids de chaque propriété QoS, tel que :

$$\sum_{i=1}^{12} w_i = 1$$

4. L'algorithme génétique

Compte tenu de facteurs tels que les contraintes d'espace et de temps, l'efficacité énergétique et la configurabilité des services IoT, la sélection des services IoT pour la composition des services est réduite à un problème d'optimisation multi-objectifs et multi-contraintes.

L'optimisation consiste à utiliser la ressource de la manière la plus efficace. Cela signifie maximiser ou minimiser certains attributs via une fonction objective [6], des algorithmes heuristiques tels que l'algorithme génétique (GA), l'optimisation des colonies de fourmis (ACO) et l'optimisation des essaims de particules (PSO) sont adoptés pour trouver la composition optimale du service IoT.

Certains algorithmes d'optimisation globale sont couramment utilisés aujourd'hui, tels que l'algorithme d'énumération, l'algorithme glouton, la méthode Backtracking, la programmation dynamique et les algorithmes génétiques [124] etc…. Cependant, l'efficacité temporelle et spatiale de la méthode d'énumération est relativement faible, un algorithme glouton ne peut garantir l'acquisition d'une solution optimale, le coût en temps de la méthode Backtracking et de la programmation dynamique augmente de façon exponentielle en fonction de l'ampleur du problème.

Par rapport aux algorithmes ci-dessus, l'algorithme génétique (GA) peut aider à obtenir une solution optimale ou quasi optimale à un coût de calcul relativement faible. Tant que la fonction de fitness est modélisée mathématiquement, GA peut être utilisé pour résoudre un problème d'optimisation à grande échelle et il est donc préféré lorsque nous voulons rechercher dans un grand espace d'états.

Ces algorithmes d'optimisation sont basés sur les théories de l'informatique évolutive et génétique, et ils sont utilisés pour optimiser les configurations d'une application (ensemble de services qui constituent l'application). Les algorithmes effectuent l'optimisation sur la base des exigences de l'utilisateur en termes de qualité de service et les critères spécifiés pour l'optimisation. Par exemple, une configuration de l'application peut être optimisée afin de minimiser la consommation globale de l'application en termes de bande passante et de maximiser les propriétés de la QoS spécifiée par l'utilisateur.

Ces algorithmes sont génériques et supportent :

1. Les critères personnalisables qui peuvent inclure plusieurs objectifs d'optimisation simultanés.
2. Les différents types de propriétés de la QoS d'une application qui peuvent être ajoutées ou supprimées des descriptions des services au moment de l'exécution selon les besoins.

L'algorithme génétique (GA) est le choix relativement judicieux pour résoudre le problème d'optimisation globale de la composition du service. L'GA est un algorithme heuristique utilisé pour rechercher la solution optimale en simulant le processus naturel [14], dans l'GA classique [125], le critère d'arrêt de l'algorithme est pris comme étant N= Nmax, N étant le nombre d'itérations. Cependant, pour certains scénarios d'application en temps quasi réel dans le contexte de l'Internet des objets, il est nécessaire d'accélérer le processus de composition des services d'information. Pour gérer cela, la préservation de l'élitisme est utilisée pour améliorer l'efficacité de l'algorithme. Sinon, la similarité de population est utilisée comme conditions de terminaison supplémentaires de l'algorithme. Il est tout à fait possible de créer des individus au hasard. Et cette méthode apporte un concept très utile dans les algorithmes génétiques : la diversité. Plus les individus de la population initiale sont différents les uns des autres, plus nous aurons la chance d'y trouver, non pas la solution parfaite, mais de quoi fabriquer les meilleures solutions possibles.

4.1. L'algorithme génétique multi-population (MGA)

L'algorithme génétique multi-population (MGA) [126] divise les individus en plusieurs groupes ou sous-populations en fonction des valeurs de fitness. Les individus d'une même communauté ont la possibilité de se jumeler. Si une personne produit une forme physique très bien ajustée, la personne migre de son groupe d'origine vers le groupe approprié avec une valeur de forme physique plus élevée, et vice versa. Ainsi, tous les individus de la population bénéficient des mêmes chances, quelle que soit leur condition physique ou leur condition physique élevée. Cela permet à MGA de maintenir la diversité de la population. Aussi, MGA est facile à paralyser car toute la population est déjà divisée en plusieurs sous-populations (Figure 18).

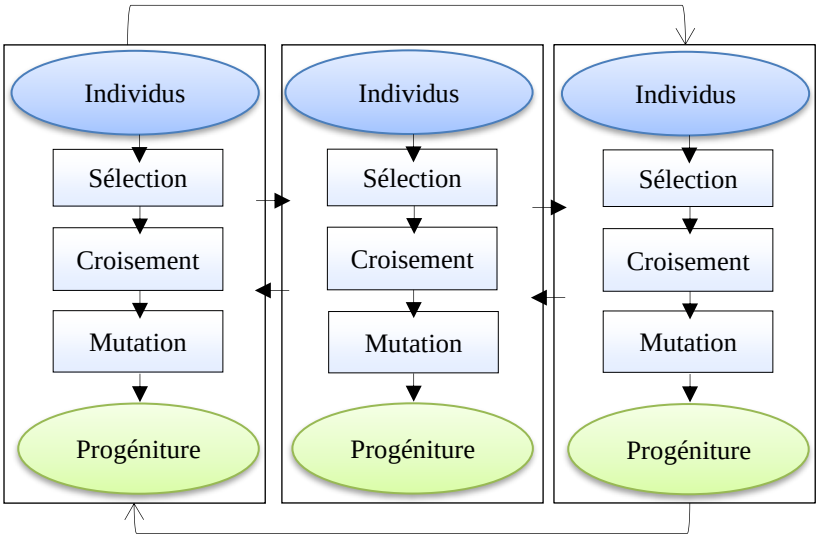


Figure 18 Le Framework de MGA

4.2. La Fonction de Fitness

Dans le modèle de Darwin, les individus présentant les meilleures caractéristiques ont de meilleures chances de survivre et de se reproduire. Pour déterminer cette capacité de survie, nous utiliserons une fonction mathématique appelée fonction de fitness ou fonction objectif (14).

$$f = w_1 f_1 + w_3 f_3 + w_4 f_4 + w_5 f_5 + w_6 f_6 + w_8 f_8 + w_9 f_9 + w_{10} f_{10} + w_{11} f_{11} + w_{12} f_{12} - w_2 f_2 - w_7 f_7) \quad (14)$$

Ici, w_i est le poids, et f_i est l'agrégation de chaque propriété QoS citée dans le (Tableau 11), telle que :

$$1 > w_i > 0 \; and \sum_{i=1}^{12} w_i = 1$$

4.3. Codage chromosomique et opérateurs génétiques

L'encodage définit la représentation des variables du problème à leur manipulation par l'algorithme évolutif, ainsi chaque chromosome représente une composition de service possible.

Nous supposons que tous les services ont été numérotés avec des nombres entiers, le chromosome est représenté par un tableau dont la taille est égale au nombre de tâches, et la valeur de chaque position dans le vecteur indique l'ordre du service candidat pour la tâche. Par exemple, dans notre cas d'étude ; le processus d'encodage de la solution est illustré à la (Figure

19). Ici (3,5,1,2,4,7,6) indique qu'il y a 7 tâches dans le service composite (basé sur notre service composite pour l'urgence hospitalière Figure 21).

Pour bien illustrer notre cas d'étude qui contient des services qui s'exécutent en parallèle, une deuxième colonne est donc ajoutée pour représenter l'ordre d'exécution en parallèle, par exemple dans la (figure 19) le service 5 et 1 s'exécute au même temps. Pareille pour les services 4,7 et 6.

Identifiant du service	**3**	**5**	**1**	**2**	**4**	**7**	**6**
L'ordre d'exécution en parallèle	*1*	*2*	*2*	*3*	*4*	*4*	*4*

Figure 19 Encodage des Chromosomes

L'opérateur de sélection est l'opérateur le plus important, son objectif est de trouver les meilleurs individus pour la sélection de la nouvelle population et la reproduction, c'est-à-dire la sélection des individus pour le croisement et la mutation. Nous avons utilisé la roue de loterie biaisée (roulette wheel) de Goldberg (1989). Selon cette méthode, chaque chromosome sera dupliqué dans une nouvelle population proportionnellement à sa valeur d'adaptation.

On effectue, en quelque sorte, autant de tirages avec remises qu'il y a d'éléments dans la population. Cette méthode consiste en la distribution de la conversation de sélection proportionnellement à la valeur de fitness.

La probabilité avec laquelle le chromosome sera réintroduit dans la nouvelle population de taille n est :

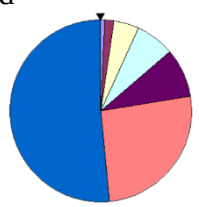
$$\boldsymbol{\varphi(x_i)} = \frac{\boldsymbol{f(x_i)}}{\sum_{i=1}^{n} \boldsymbol{f(x_i)}} \qquad (15)$$

$$\boldsymbol{Sum} = \sum_{i=1}^{n} \varphi_i \underline{\boldsymbol{= 1}} \qquad (16)$$

$\boldsymbol{f}$ Est la valeur de fitness pour chaque individu, et $\boldsymbol{n}$ est la taille de la population, $\boldsymbol{Sum}$ est la somme des $\boldsymbol{\varphi(x_i)}$.

Algorithme 1 Fonction (roulette wheel) pour le tirage proportionnel à la valeur fitness des chromosomes

Input	Population of Individual (x_i) of current generation
Output	2 Individuals (x_i) chosen for the next population
// n_select_{xi}	is the number of times that the individual has been selected
Step 1	i is integer = 1
Step 2	For y=1 to 2 (repeat the process 2 times)
Step 3	S is a real = 0 (Cumulative probability)
Step 4	r is a real = Random (0, 1) « 0 <r <1»
Step 5	For each individual (x_i)
Step 6	Calculate $\varphi(x_i)$



Step 7	If (S < r)
Step 8	S=S+ $\varphi(x_i)$
Step 9	else
Step 10	$n\_select_{x_i}$ =$n\_select_{x_i}$+1
Step 11	End
Step 12	i++
Step 13	End
Step 14	End
Step 15	Chose the most selected 2 Individuals (x_i)

Les 2 individus qui ont été choisi plusieurs fois à l'aide de l'algorithme 1, ont la possibilité de participer à la population suivante ou au croisement en un point, il a le même principe que la Roulette du casino. Un croisement en un point (Algorithme 5) est utilisé, en appliquant les deux règles de la fonction (17).

Offspring_1=begin_father+end_mother
Offspring_2=begin_mother+end_father (17)

Une mutation aléatoire (Algorithme 6) a été appliquée pour maintenir la diversité des individus, en changeant les valeurs du service sélectionné au hasard, le service sélectionné doit être dans la même classe de service que celui modifié.

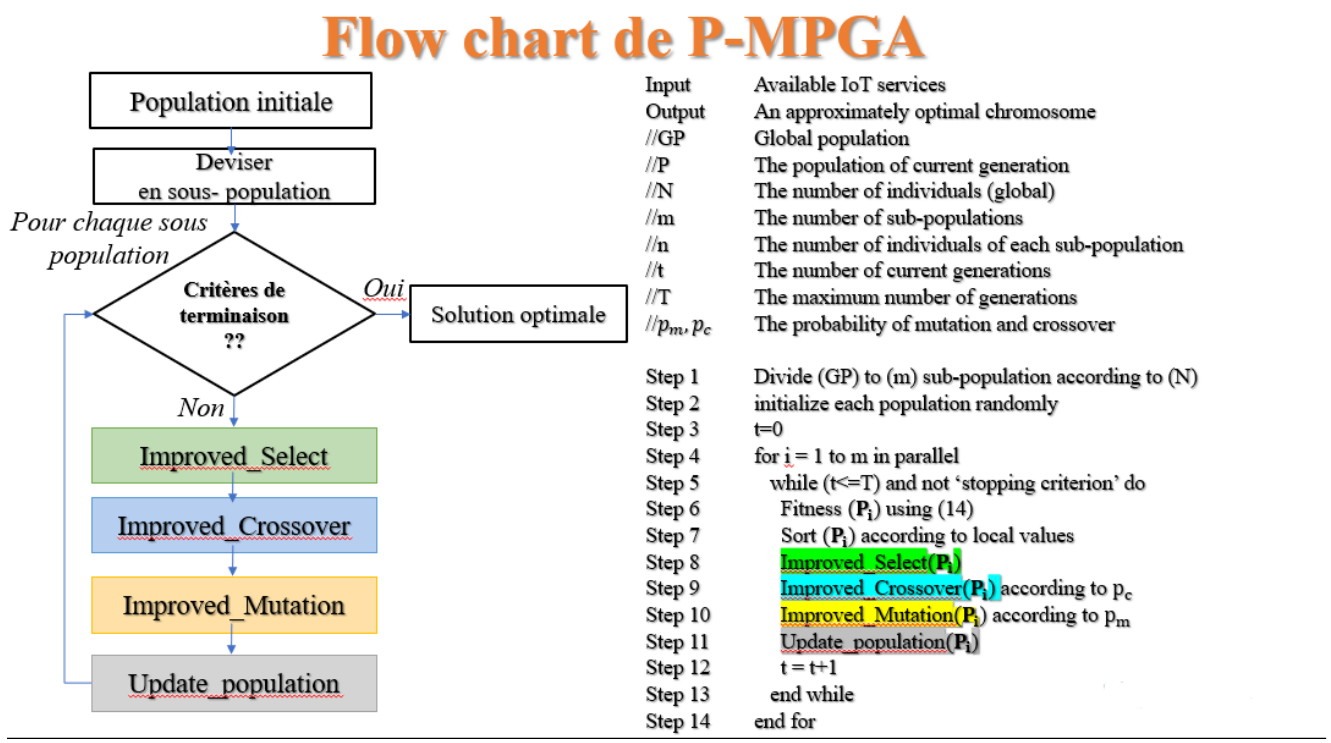
Aussi, une méthode de surveillance (Algorithme 3) est utilisée pour surveiller chaque point (gène) de chaque chromosome, chaque point (gène) représente un service donné, si un service devient indisponible ou lorsque des pannes/exceptions surviennent, ce service est instantanément remplacé par un autre de sa classe de service. Cela nous permettra de nous remettre de la situation inattendue afin que l'application continue son exécution sans interruption. Pour gagner du temps d'exécution, l'idée est d'exploiter la boucle de notre algorithme de sélection (Algorithme 4) lors du calcul de la valeur de fitness.

Pour résumer, notre algorithme génétique multi-population (P-MPGA) (Algorithme 2), consiste tout d'abord à diviser la population globale (GP) en (m) sous population (P_i) en fonction du nombre d'individus global (N). Pour chaque sous population (P_i) en parallèle, on calcule la valeur fitness de chaque individu de la population courante (P_i), tous les individus doivent être triés en fonction de leurs valeurs locales. Ensuite, notre algorithme de sélection (*improved_select*), notre algorithme de croissement (*improved_crossover*) et notre algorithme de mutation (*improved_mutation*) sont alors exécuter jusqu'à ce qu'on arrive au nombre maximum de génération (T) ou au critère d'arrêt tout simplement.

Algorithme 2 Notre algorithme génétique multi-population (P-MPGA)

Input	Available IoT services
Output	An approximately optimal chromosome
//GP	Global population
//P	The population of current generation
//N	The number of individuals (global)
//m	The number of sub-populations

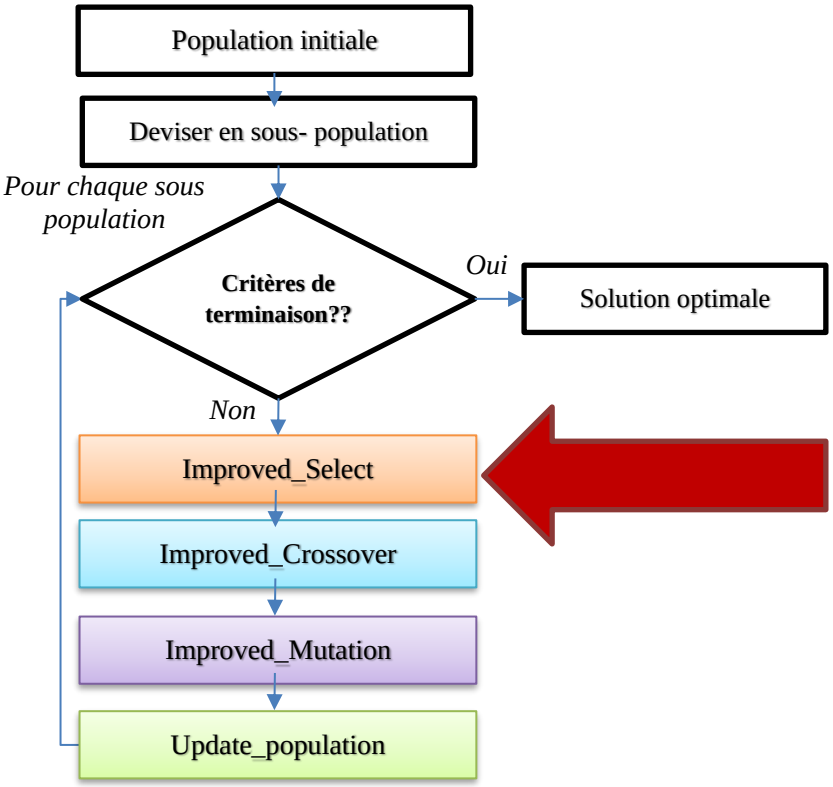
//n	The number of individuals of each sub-population
//t	The number of current generations
//T	The maximum number of generations
//p_m, p_c	The probability of mutation and crossover
Step 1	Divide (GP) to (m) sub-population according to (N)
Step 2	initialize each population randomly
Step 3	t=0
Step 4	for i = 1 to m in parallel
Step 5	while (t<=T) and not 'stopping criterion' do
Step 6	Fitness (P_i) using (14)
Step 7	Sort (P_i) according to local values
Step 8	Improved_Select(P_i)
Step 9	Improved_Crossover(P_i) according to p_c
Step 10	Improved_Mutation(P_i) according to p_m
Step 11	Update_population(P_i)
Step 12	t = t+1
Step 13	end while
Step 14	end for



Algorithme 3 Notre algorithme de surveillance (Monitoring)

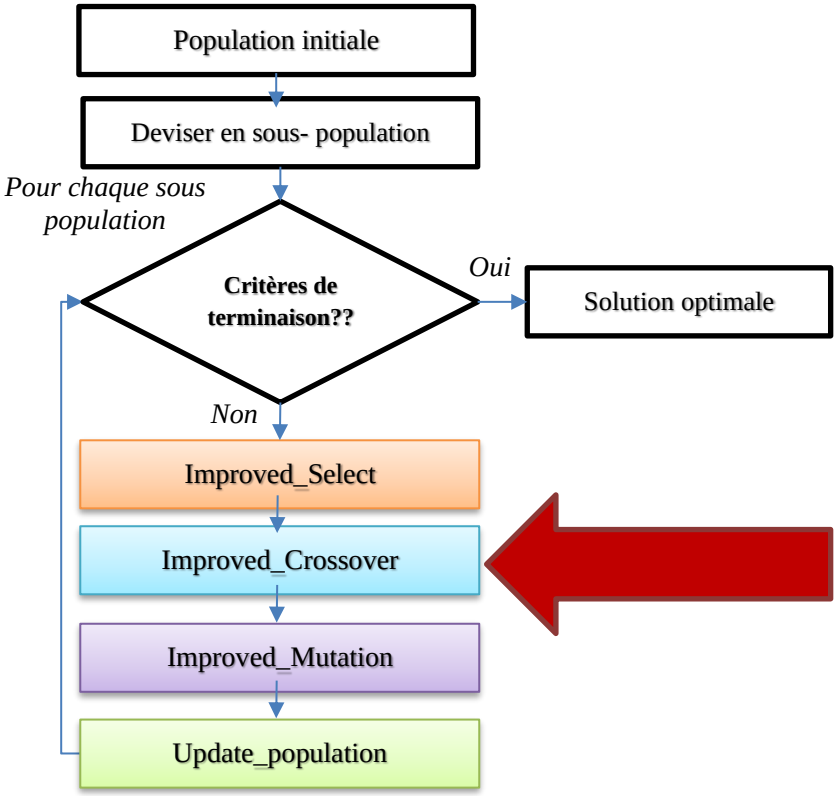
Input	Individual (x_i)
Output	A safe Individual (x_i)
//p	A point of (p_j) or the gene
//Sc	Service class
Step 1	for each (p_j)
Step 2	If (p_j) is not safe ()

Step 2	Replace (p_j) with another service picked up from same (Sc_i)
Step 3	End
Step 4	End



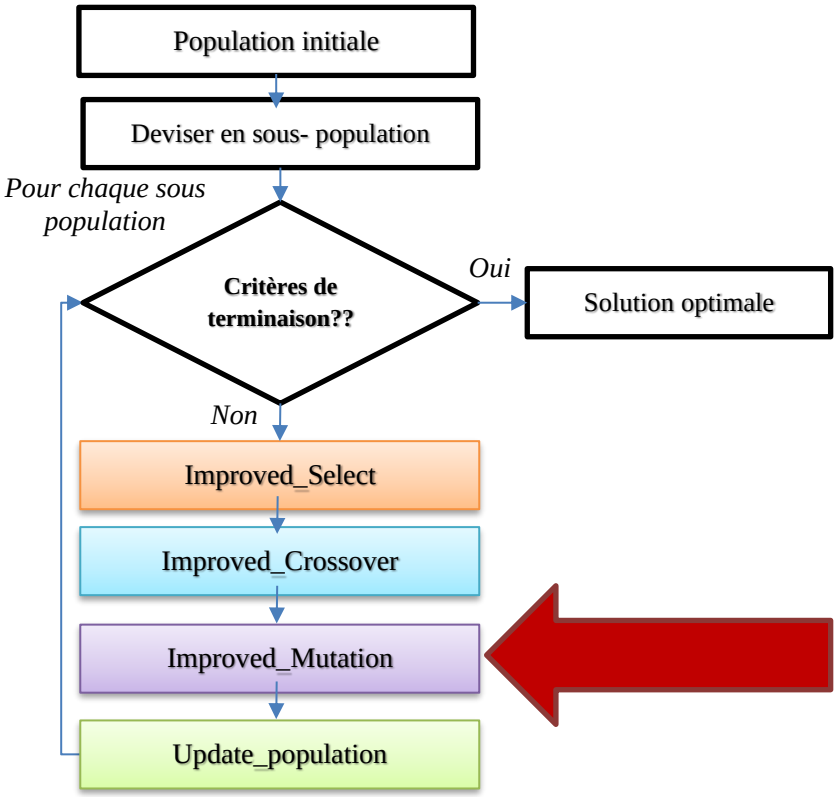
Algorithme 4 Notre algorithme de sélection amélioré (improved_select)

Input	Population of Individual (x_i) of current generation
Output	2 Individuals (x_i) chosen for the next population
// n_select_{xi}	is the number of times that the individual has been selected
Step 1	i is integer = 1
Step 2	For y=1 to 2 (repeat the process 2 times)
Step 3	S is a real = 0
Step 4	r is a real = Random (0, 1) « 0 <r <1»
Step 5	For each individual (x_i)
Step 6	Monitoring (x_i)
Step 7	Calculate $\varphi(x_i)$ (Cumulative probability)
Step 8	If (S < r)
Step 9	S=S+ $\varphi(x_i)$
Step 10	else
Step 11	$n\_select_{x_i}$ =$n\_select_{x_i}$+1
Step 12	End
Step 13	i++
Step 14	End
Step 15	End
Step 16	Chose the most selected 2 Individuals (x_i)



Algorithme 5 Notre algorithme de croisement amélioré (improved_crossover)

Input	Population ($\mathbf{P_i}$)
Output	2 new individuals
//P	The population of current generation
// x	Individual of ($\mathbf{P_i}$)
//po	The one point
// x_father	Individual father for the crossover
// x_mother	Individual mother for the crossover
//b_father	The begin of the father according to po
//e_father	The end of the father according to po
//b_mother	The begin of the mother according to po
//e_mother	The end of the mother according to po
//os_1	Offspring 1
//os_2	Offspring 2
Step 1	Choose the two-best induvial for x_father & x_mother respectively
Step 2	Choose a random one point
Step 3	os_1= b_father + e_mother
Step 4	os_2=b_mother + e_father



Algorithme 6 Notre algorithme de mutation amélioré (improved_mutation)

Input	Individual (x_i)
Output	Individual (x_i) with a mutation for the new generation
//p	The random point (gene) for the mutation
//Sc	Service class of (x_i)
// x_service	New service from (Sc_i)
Step 1	po = random (1,7)
Step 2	x_service= service picked up from (Sc_i)
Step 3	Replace p with x_service

5. Conclusion

Dans ce chapitre, Notre étude nous a montré que l'architecture Fog-IoT en cinq couches peut répondre aux exigences de l'IoT et satisfaire au maximum les critères cités dans le chapitre 3.

Aussi L'utilisation du Fog computing et la défragmentation de la couche Transport en quatre sous-couches (sécurité, stockage, pré-traitement & surveillance) comme le montre la (Figure 16) nous permet d'avoir plusieurs avantages essentiels à l'environnement IoT.

Compte tenu de facteurs tels que les contraintes d'espace et de temps, l'efficacité énergétique et la configurabilité des services IoT, la sélection des services IoT pour la composition des services est réduite à un problème d'optimisation multi-objectifs et multi-contraintes. Un algorithme génétique multi-population sensible aux QoS P-MPGA est proposé

qui pour but d’utiliser les ressources de la manière la plus efficace pour l'optimisation. Cela signifie maximiser ou minimiser certains attributs QoS.

Dans le prochain chapitre, nous évaluerons notre algorithme génétique multi-populations (P-MPGA) en le comparant aux algorithmes génétiques traditionnels (GA) [14] et à l'algorithme génétique (MGA) proposés dans [15].

CHAPITRE 5

Configuration expérimentale et évaluation

Sommaire

Dans ce chapitre, nous évaluons notre algorithme génétique multi-populations (P-MPGA) (algorithme 2) en le comparant à l'algorithme génétique traditionnel (GA) [14] et à l'algorithme génétique (MGA) proposés dans [15].

La comparaison entre notre algorithme génétique multi-populations (P-MPGA) qui fait des traitements en parallèles avec un algorithme génétique traditionnel, a pour but de montrer l'efficacité et la performance en termes de temps d'exécution entre un GA traditionnel et un GA parallèle. Ensuite, c'est judicieux de comparer notre algorithme P-MPGA avec l'algorithme génétique multi-population (MGA) proposés dans [15] qui est lui-même un algorithme multi-objectifs QoS pour la composition de services dans l'IoT.

Nous exécutons P-MPGA, GA et MGA 10 fois et utilisons les valeurs moyennes pour l'évaluation.

1. Configuration utilisée

Les algorithmes ont été implémentés avec Java à l'aide de PC-Soft Windev23 IDE et fonctionnaient sur un ordinateur portable Intel i7-6820HQ 2.71Ghz, SDD, 16 Go de RAM, ordinateur portable avec Windows 10.

Pour illustrer l'idée de notre composition de service, un exemple de l'urgence hospitalière est utilisé, qui consiste aux processus de secourir une personne au cas de besoin en passant du trajet de l'ambulance jusqu'à l'hôpital x.

Nous supposons que notre système est équipé de différents dispositifs, dont des capteurs et une caméra, toutes les expériences sont basées sur le processus présenté dans la (Figure 21), qui représentent un exemple de service composite pour notre cas d'étude. Il contient environ sept services et chacun peut être lui-même un service composite, comme la sélection de la route ou le contrôle automatique des feux de circulation sur le parcours de l'ambulance, qui consiste à laisser passer l'ambulance en donnant le feu vert. Le tableau 12 énumère la signification de chacun d'eux.

Dans notre approche, une classe de service (Sc) pour un service IoT est un tuple (nm, dsc, op) où :

- nm est le nom du service
- dsc est la description textuelle du service
- op est une opération du service. Ici, nm, dsc et op sont les mêmes que ceux spécifiés pour les services.

Nos expériences sont basées sur une base de données de service disponible publiquement QWS Dataset (2.0) [127], nous étendons la taille du jeu de données QWS de manière aléatoire. Le nombre de services est finalement étendu à 10 000 et 12 attributs QoS sont sélectionnés pour nos expérimentations, à savoir Disponibilité (A), Coût (C), Documentation (D), Emplacement (L), Ressources mémoire (M), Précision (P), Fiabilité (R), Temps de réponse (Rt), Réputation (Rp), Sécurité (S), Classification de service (Sc), Taux de réussite (Sr), Débit (T), ils sont liés au 2e, 8e, 9e, 6e, 5e, 12e, 1er, 7e, 10e, 11e, 4e, 3e, champs de la base de données QWS respectivement.

Tableau 12 Service classes (Sc).

Sc Id	Nom et description du Sc
Sc1	Choisir un hôpital : comme le nom l'indice, ces services (ou services composés) ont pour but de trouver l'hôpital le plus optimal selon certains critères précis (distance, disponibilité, etc..)
Sc2	Ramasser une ambulance : l'ensemble de services et services composés qui consiste à trouver l'ambulance la plus adéquate pour ramasser le patient et l'emmener à l'hôpital.
Sc3	Envoyer les informations du patient à l'hôpital
Sc4	Choisir le meilleur chemin de l'ambulance vers l'hôpital
Sc5	Feux de circulation : l'ensemble de services et services composés qui consiste à faciliter la circulation routière en utilisant des feux de circulation, ces services composés se basent sur plusieurs capteurs et caméras.
Sc6	Notifier les voitures publiques : l'ensemble de services et services composés qui consiste à faciliter la circulation routière en notifiant les voitures publiques par exemple.
Sc7	Avertir les autorités : l'ensemble de services et services composés qui consiste à faciliter la circulation routière en avertissant les autorités comme la police par exemple.
Initialisation et fin du processus	
Init	Urgence signalée : grâces soit ; à des capteurs, des caméras, des objets intelligents (comme les smartphones ou smart-Watch ou tout simplement l'intervention humaine.)
End	L'arrivée du patient à l'hôpital choisi

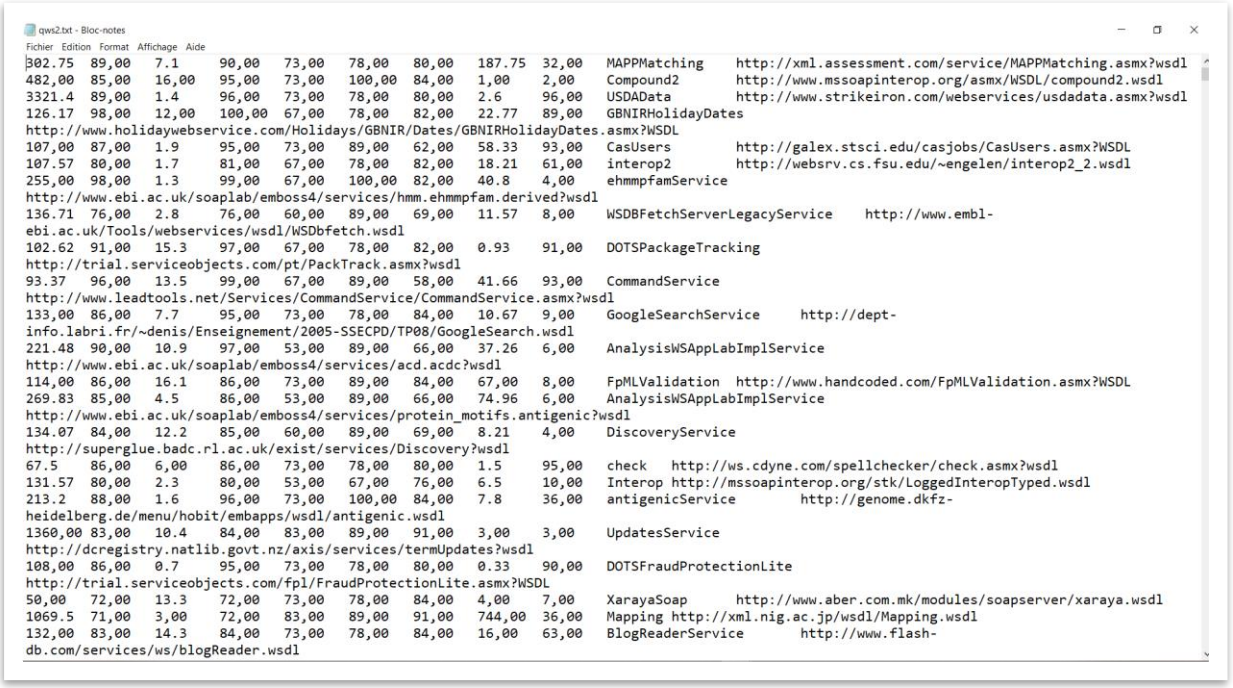


Figure 20 QWS Dataset (2.0) [127]

Dans la (Figure 20), chaque ligne représente un service IoT et chaque colonne c'est une valeur QoS du même service.

Dans notre expérimentation, nous supposons que le schéma de composition est déjà défini (figure 21). Chaque lieu Sci comme dans Sc1, est une représentation de la sélection entre les services « Ramasser une ambulance » qui sont disponible dans le service classe (Sc2).

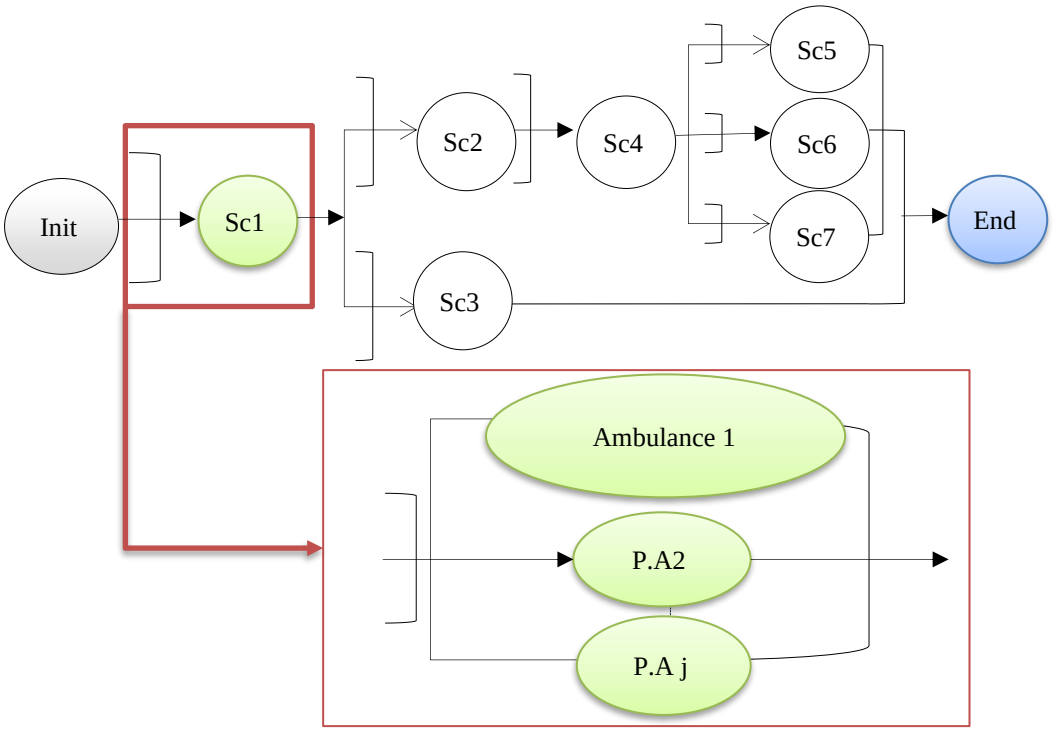


Figure 21 Le schéma de la composition de services pour l'urgence hospitalière Fog-IoT

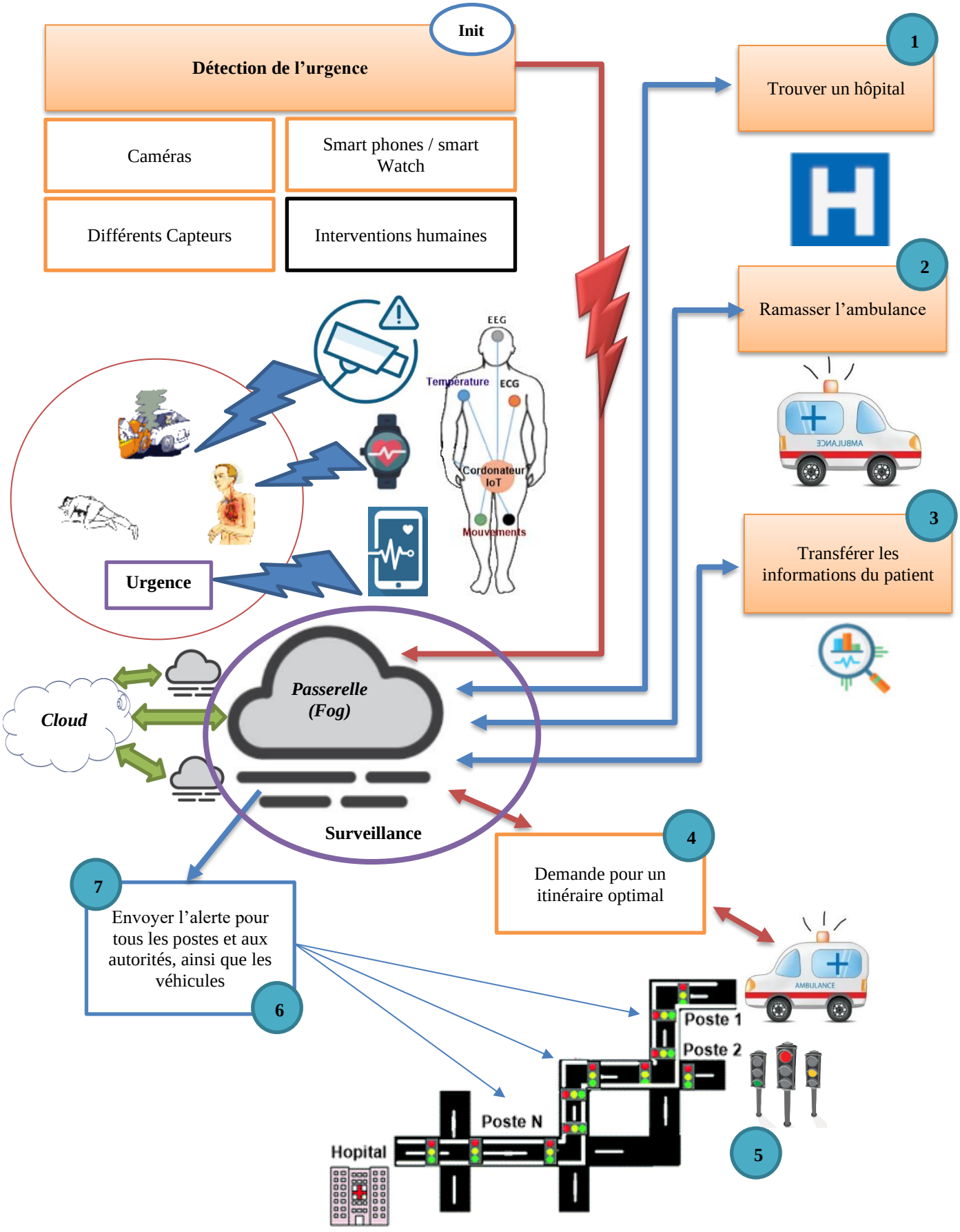


Figure 22 Urgence hospitalière Fog-IoT

La (Figure 22) est une représentation de l'ensemble des processus de composition de services IoT pour notre cas d'étude "l'urgence hospitalière", pour commencer l'urgence est détectée soit par des capteurs qui se situent sur le corps de la personne (par exemple des capteurs intégrés dans une Smart-Watch ou un smartphone), ou bien par le biais des caméras de surveillance, une intervention humaine peut être aussi un déclencheur de se processus (par un appel téléphonique par exemple). Dés que l'urgence est signalée, une notification est alors envoyée vers notre Fog-IoT (passerelle de la zone ou se situe l'incident). Apres l'analyse de la situation, plusieurs services alors sont déclenchés soit en séquentielle ou bien en parallèle tout en gardant contacte avec notre centre de calcule (Fog-IoT) qui a pour rôle de gérer l'ensemble des opérations, le processus de composition se poursuit jusqu'a l'arriver du patient à l'hôpital choisi. Chaque Fog-IoT dispose d'un système de surveillance qui a pour but de surveiller tous les services autour de lui.

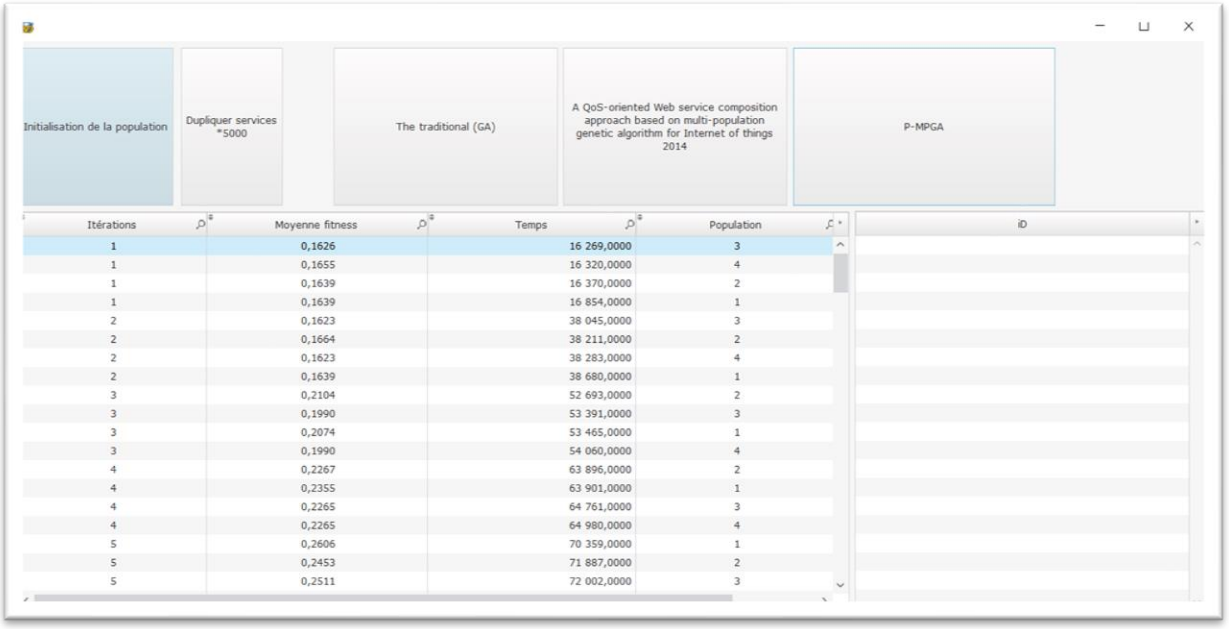


Figure 23 Notre environnement de simulation développé

Tableau 13 Paramètre P-MPGA

Nombre de services disponibles	**10 000**
Taille de la population	5 000
Nombre de générations	2 000
Méthode de sélection	Roulette wheel
Type de croisement	Un point
Probabilité de croisement	90%
Type de mutation	Choix aléatoire parmi un Sc spécifique
Probabilité de mutation	10%
Nombre d'itérations	40

2. Résultat de l'expérimentation et discussion

Dans cette section, nous présentons les résultats obtenus par notre P-MPGA par rapport à GA et MGA, nous avons utilisé les mêmes opérateurs génétiques, paramètres (présentés dans le Tableau 13) et la même fonction de fitness pour tous les algorithmes, nous avons également utilisé quatre sous-populations pour P- MPGA.

La (Figure 24) montre qu'avec P-MPGA, le temps d'exécution est considérablement réduit par rapport à GA & MGA. La (Figure 25) montre que le P-MPGA fonctionne mieux que GA et MGA, le P-MPGA converge mieux vers la valeur de fitness maximale.

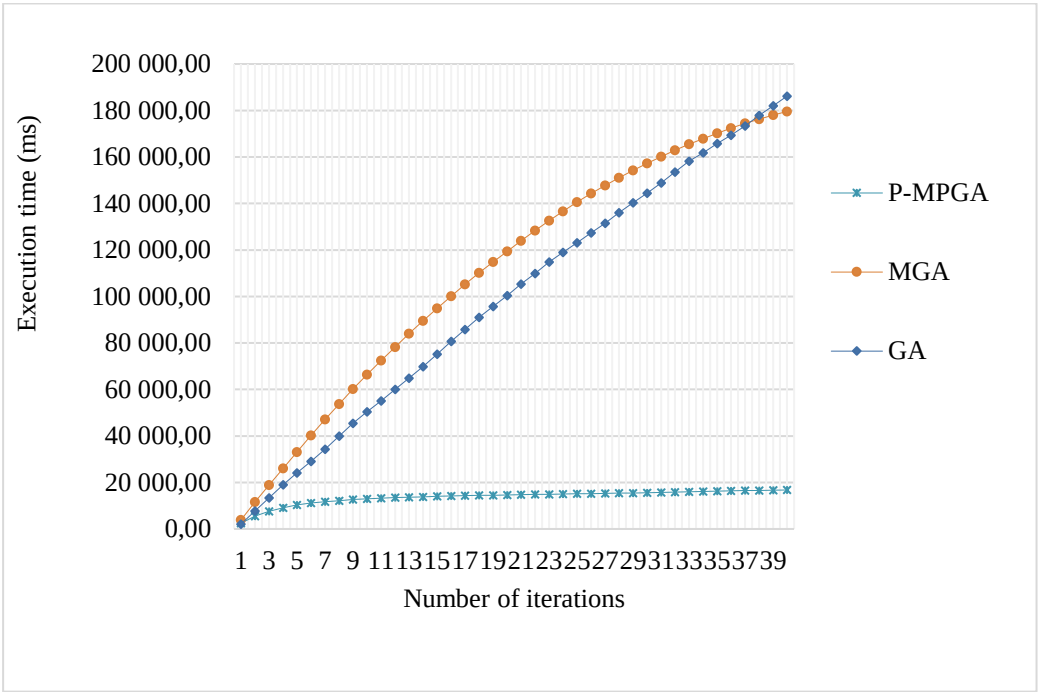


Figure 24 Comparaison entre-MPGA, MGA & GA (temps d'exécution)

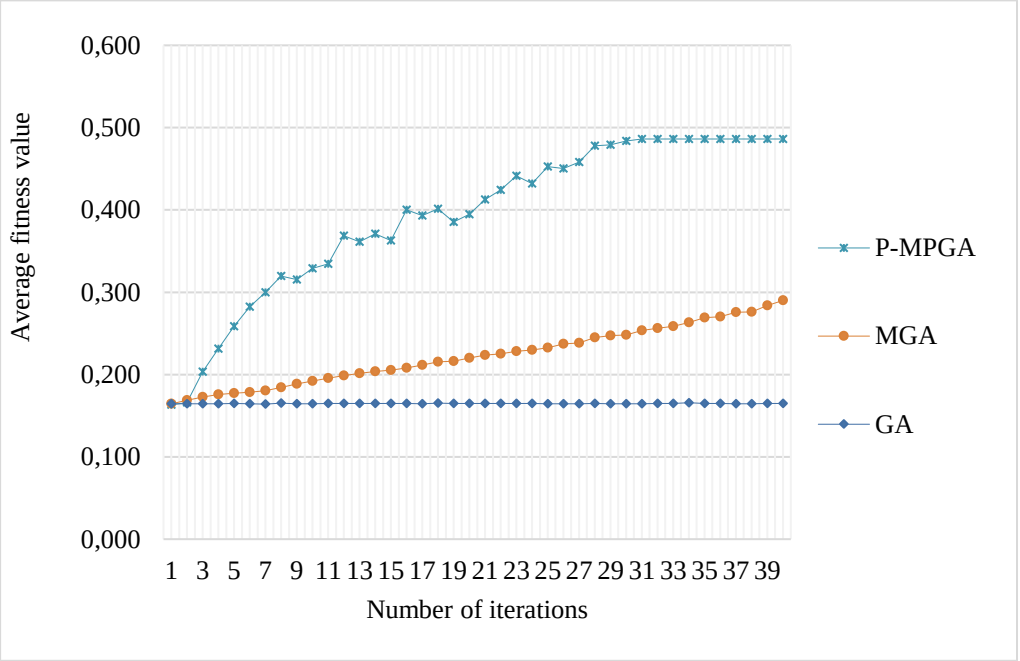


Figure 25 Comparaison des valeurs de fitness obtenues par P-MPGA, MGA et GA

Tableau 14 Ratio (Durée d'exécution / Meilleure valeur de fitness) après 40 itérations.

Méthode	Meilleur valeur fitness	Temps d'exécution (ms)	Ratio
Avec 5 000 Services			
MGA	0,5555995702744	179590,0000	323236,3911
GA	0,5517482757568	172156,0000	312019,0992
P-MPGA	0,4922352731228	16775,7500	34080,7555
Avec 10 000 Services			
MGA	0,5178905129433	191 823,0000	370392,9599
GA	0,5178905129433	173699,0000	335397,1460
P-MPGA	0,5178905129433	16 752,2500	32347,0880
Avec 15 000 Services			
MGA	0,4989299476147	219 689,0000	440320,3316
GA	0,4989299476147	195192,0000	391221,2545
P-MPGA	0,4989299476147	18 655,7500	37391,5218
Avec 20 000 Services			
MGA	0,6055026650429	234 021,0000	386490,4541
GA	0,5641102790833	201543,0000	357275,8864
P-MPGA	0,5641102790833	18 946,2500	33586,0747

Le (tableau 14) représente la valeur de fitness optimale obtenue après 40 itérations avec le temps d'exécution, malgré l'augmentation du nombre de services (5000,10000,15000 & 20000), le P-MPGA montre un meilleur ratio (temps d'exécution/valeur de fitness optimale).

Sur la (Figure 26), malgré l'augmentation de la population initiale, le P-MPGA obtient toujours une bonne valeur de fitness optimale, ce qui rend notre algorithme évolutif.

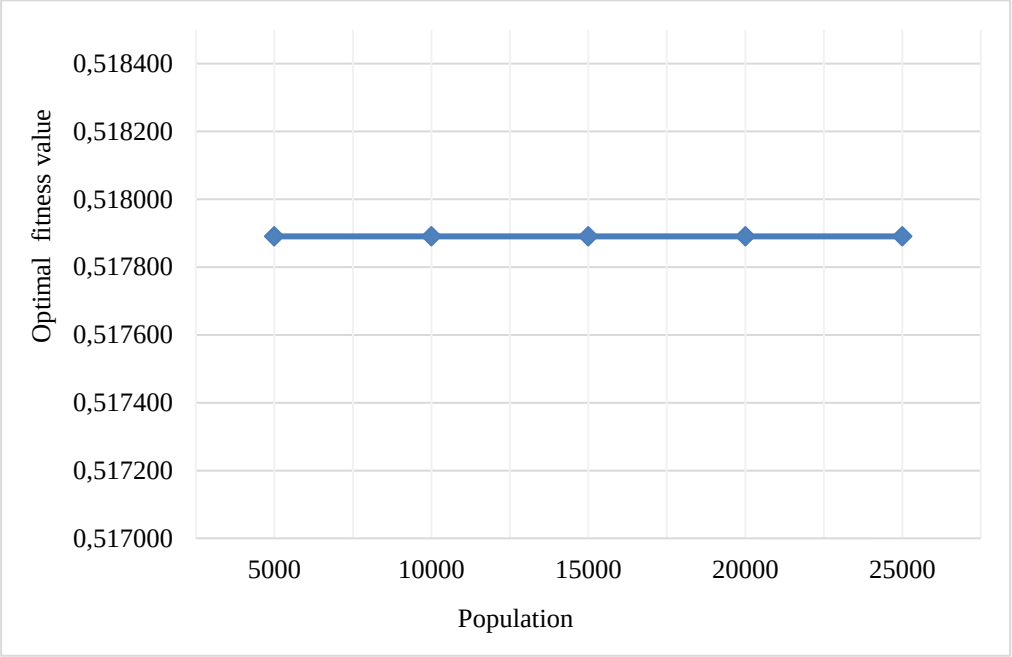


Figure 26 Valeurs de fitness optimales obtenues par P-MPGA en fonction de la population

3. Conclusion

Dans ce chapitre, nous avons évalué notre algorithme génétique multi-populations (P-MPGA) en le comparant aux algorithmes génétiques traditionnels (GA) [14] et à l'algorithme génétique (MGA) proposés dans [15]. Les résultats expérimentaux montrent les excellents résultats de P-MPGA en termes de temps d'exécution, la moyennes de la valeurs fitness et un rapport temps d'exécution / meilleure valeur de fitness malgré l'augmentation de la population initiale.

Dans la section prochaine, nous conclurons et discuterons les perspectives de ce travail.

Conclusion et perspective

Sommaire

1. Récapitulatif du travail réalisé et des contributions

L'internet des objets (IoT) est une technologie émergente qui devient de plus en plus populaire. L'IoT est une fusion entre l'espace d'information et l'espace physique, connue sous le nom d'interconnexion d'ordinateurs embarqués, de capteurs, d'appareils mobiles ou d'autres objets identifiables de manière unique, qui sont capables d'interagir les uns avec les autres et de coopérer avec l'environnement pour atteindre des objectifs communs, en exploitant l'infrastructure Internet existante.

L'IoT consiste à connecter un grand nombre d'objets quotidiens à Internet, en leur donnant leur propre identité, leur permettant ainsi d'offrir des fonctionnalités et de collecter des informations sous la forme d'un service. Afin de répondre à la demande compliquée de l'utilisateur, la plupart des périphériques IoT ne fonctionnent pas seuls, une composition de service multiple doit être faite, et elle est définie comme la composition des services.

La composition des services consiste à combiner la fonctionnalité de plusieurs services au sein d'un seul processus afin de répondre à des demandes complexes qu'un seul service ne peut satisfaire [1]. La composition de service offre la possibilité d'effectuer des activités complexes à partir de services existants, et les services composés forment un nouveau service, qui peut être réutilisé dans une autre composition [2].

La croissance de l'Internet des Objets (IoT) implique la disponibilité d'un très grand nombre de services qui peuvent être similaires ou identiques, la gestion de la Qualité de Service (QoS) permet de différencier un service d'un autre. Dans cette thèse, nous avons présenté une approche de composition de service adaptative basée sur la QoS en utilisant un algorithme génétique multi-population dans un environnement Fog-IoT-Healthcare.

Cette Thèse est organisée en deux partie soit six chapitres. La première partie est consacrée à l'introduction et l'analyse de l'état de l'art. Dans la deuxième nous présenterons notre Contribution et sa validation.

Le premier chapitre présente le cadre général et la problématique de la recherche. Ce chapitre identifie également les objectifs de la thèse et présente brièvement les principales contributions.

Le deuxième chapitre est consacré à la présentation de concepts de base et fondamentaux qui serviront à clarifier les termes utilisés tout au long du document. Aussi, une partie de ce

chapitre parle des défis de l'IoT avec une comparaison entre la composition de services Web et celle de l'IoT.

Le troisièmes chapitre traite certaines approches proposées populaires qui peuvent être considéré pour la composition de services dans l'IoT. Puis dans sa deuxième partie, nous avons présenté une étude comparative entre les approches déjà proposées. Nous avons commencé notre enquête en introduisant quelques critères spécifiques, qui sont basés sur les défis fondamentaux de la composition des services cités précédemment dans le chapitre 2.

Dans le quatrième chapitre, nous avons présenté notre architecture à cinq couches basées sur le concept Fog-IoT, le modèle QoS utilisé et notre algorithme génétique multi-population sensible aux QoS P-MPGA.

Dans le cinquième chapitre, nous avons évalué notre algorithme génétique multi-populations (P-MPGA) (algorithme 1) en le comparant aux algorithmes génétiques traditionnels (GA) [14] et à l'algorithme génétique (MGA) proposés dans [15]. Nous exécutons P-MPGA, GA et MGA 10 fois et utilisons les valeurs moyennes pour l'évaluation. Enfin dans ce dernier chapitre qui est le sixième, nous terminons cette thèse par une conclusion générale et des quelques perspectives sur des travaux futurs.

Pour conclure, cette thèse nous a permis de comprendre ce qui a été fait (protocoles, algorithmes, solutions proposées) et ce qui reste à traiter. Les problèmes d'architecture IoT-cloud nous ont amenés à utiliser l'architecture à 5 couches implémentée sur un système informatique Fog-IoT, particulièrement la couche de traitement. Notre travail s'est concentré sur cette couche traitement (Processing) où nous l'avons divisée en quatre sous-couches (sécurité, stockage, pré-traitement et surveillance), cela nous permet d'avoir des avantages prometteurs, et en se basant sur cela, nous avons mis en œuvre un algorithme génétique multi-population sensible à la QoS (P-MPGA), et nous avons considéré 12 attributs QoS, c'est-à-dire la disponibilité (A), le coût (C), la documentation (D), l'emplacement (L), les ressources mémoire (M), Précision (P), Fiabilité (R), Temps de réponse (Rt), Réputation (Rp), Sécurité (S), Classification de service (Sc), Taux de réussite (Sr), Débit (T).

P-MPGA met en œuvre une méthode de sélection intelligente qui nous permet de toujours sélectionner le bon service. En outre, une fonction est utilisée pour surveiller les services afin de gérer le changement dynamique des environnements IoT. Les résultats expérimentaux montrent les excellents résultats de P-MPGA en termes de temps d'exécution, la moyennes de la valeurs fitness et un rapport temps d'exécution / meilleure valeur de fitness malgré l'augmentation de la population initiale. P-MPGA peut rapidement atteindre un service composite satisfaisant les besoins de QoS de l'utilisateur, ce qui le rend approprié pour un environnement IoT à grande échelle.

2. Perspectives

À la suite des réalisations que nous avons réussi à mener à bien, dans le cadre de cette thèse, plusieurs perspectives peuvent être planifiées afin d'optimiser et améliorer la démarche proposée :

- Se concentrer davantage sur le système de surveillance, la consommation d'énergie du Framework et évaluer notre modèle dans le système de service du monde réel (urgence d'ambulance), qui est un cas d'étude important pour la vie humaine.
- Implémenter l’approche proposée en utilisant éventuellement des simulations dans le monde réel et l’appliquer sur un véritable ensemble de données

Pour finir, l’Interne des objets (IoT) est un domaine récent et émergeant. Ainsi, plusieurs axes de recherche restent à étudier.

Bibliographie

[1] D. Berardi, D. Calvanese, G. D. Giacomo, M. Lenzerini, and M. Mecella, “A foundational vision of e-services,” *C. Bussler, D. Fensel, M. E. Orlowska, J. Yang, Ed. Vol. 3095 Lect. Notes Comput. Sci.*, pp. 28–40, 2003.

[2] I. Aoudia, S. Benharzallah, L. Kahloul, and O. Kazar, “Service composition approaches for internet of things : a review,” *Int. J. Commun. Networks Distrib. Syst.*, vol. 23, no. 2, pp. 194–230, 2019, doi: 10.1109/TASE.2016.2539240.

[3] I. Aoudia, S. Benharzallah, L. Kahloul, and O. Kazar, “A comparative analysis of IoT service composition approaches,” *Int. Arab Conf. Inf. Technol. Yassmine Hammamet, Tunis.*, pp. 1–7, 2017.

[4] D. E. D. Abou-Tair, S. Büchsenstein, and A. Khalifeh, “A fog computing-based framework for privacy preserving IoT environments,” *Int. Arab J. Inf. Technol.*, vol. 17, no. 3, pp. 306–314, 2020, doi: 10.34028/iajit/17/3/4.

[5] A. Sarhan, “Fog Computing as Solution for IoT-Based Agricultural Applications,” in *Smart Agricultural Services Using Deep Learning, Big Data, and IoT*, no. January 2021, IGI Global, Ed. 2021, pp. 46–68.

[6] N. Kashyap, A. C. Kumari, and R. Chhikara, “Service Composition in IoT using Genetic algorithm and Particle swarm optimization,” *Open Comput. Sci.*, vol. 10, no. 1, pp. 56–64, 2020, doi: 10.1515/comp-2020-0011.

[7] I. Mashal, O. Alsaryrah, T. Y. Chung, C. Z. Yang, W. H. Kuo, and D. P. Agrawal, “Choices for interaction with things on Internet and underlying issues,” *Ad Hoc Networks*, vol. 28, pp. 68–90, 2015, doi: 10.1016/j.adhoc.2014.12.006.

[8] O. Said and M. Masud, “Towards internet of things: Survey and future vision,” *Int. J. Comput. Networks*, vol. 5, no. 1, pp. 1–17, 2013, [Online]. Available: http://www.cscjournals.org/csc/manuscript/Journals/IJCN/volume5/Issue1/IJCN-265.pdf.

[9] M. Yun and B. Yuxin, “Research on the architecture and key technology of Internet of Things (IoT) applied on smart grid,” 2010, doi: 10.1109/ICAEE.2010.5557611.

[10] D. G. Darwish and E. Square, “Improved Layered Architecture for Internet of Things,” *Int. J. Comput. Acad. Res.*, vol. 4, no. 4, pp. 214–223, 2015, [Online]. Available: http://www.meacse.org/ijcar.

[11] S. Madakam, R. Ramaswamy, and S. Tripathi, “Internet of Things (IoT): A Literature Review,” *J. Comput. Commun.*, vol. 03, no. 05, pp. 164–173, 2015, doi: 10.4236/jcc.2015.35021.

[12] P. Sethi and S. R. Sarangi, “Internet of Things: Architectures, Protocols, and Applications,” *J. Electr. Comput. Eng.*, vol. 2017, 2017, doi: 10.1155/2017/9324035.

[13] S. M. A. Oteafy and H. S. Hassanein, “IoT in the Fog: A Roadmap for Data-Centric IoT Development,” *IEEE Commun. Mag.*, 2018, doi: 10.1109/MCOM.2018.1700299.

[14] M. Sun, Z. Shi, S. Chen, Z. Zhou, and Y. Duan, “Energy-Efficient Composition of Configurable Internet of Things Services,” *IEEE Access*, vol. 5, pp. 25609–25622, 2017, doi: 10.1109/ACCESS.2017.2768544.

[15] Q. Li, R. Dou, F. Chen, and G. Nan, “A QoS-oriented Web service composition approach based on multi-population genetic algorithm for Internet of things,” *Int. J. Comput. Intell. Syst.*, vol. 7, no. SUPPL.2, pp. 26–34, 2014, doi: 10.1080/18756891.2014.947090.

[16] I. Aoudia, S. Benharzallah, L. Kahloul, and O. Kazar, “A Multi-Population Genetic Algorithm for Adaptive Qos-Aware Service Composition in Fog-Iot Healthcare Environment,” *Int. Arab J. Inf. Technol.*, vol. 18, no. 3, pp. 464–475, 2021, doi: 10.34028/iajit/18/3a/10.

[17] G. Hubert, “L’internet des objets ‘web,’” 2015. http://blogs.lesechos.fr/internetactu-net/l-internet-des-objets-web-a13507.html.

[18] R. Saad, “Modèle collaboratif pour l ’ Internet of Things (IoT),” 2016.

[19] G. Babin and M. Leblanc, “Impact Sur Le Commerce B2B,” *Distrib. Comput.*, no. Août, 2003.

[20] Mohammed Faical Abouzaid, “ANALYSE FORMELLE D’ORCHESTRATIONS DE SERVICES WEB MOHAMMED,” 2010.

[21] H. Liu, “Study and Application of Urban Flood Risk Map Information Management System Based on SOA,” *J. Softw.*, vol. 10, no. 2, pp. 180–189, Feb. 2015, doi: 10.17706/jsw.10.2.180-189.

[22] B. Soukkarieh, “Technique de l’internet et ses langages: vers un système d’information web restituant des services web sensibles au contexte,” 2010.

[23] D. Zeng, S. Guo, and Z. Cheng, “The web of things: A survey,” *J. Commun.*, vol. 6, no. 6, pp. 424–438, 2011, doi: 10.4304/jcm.6.6.424-438.

[24] J. M. Alliot, “Qu’est ce que le ‘Middleware,’” pp. 1–17, 2003, [Online]. Available: http://www.recherche.enac.fr/~alliot/middle.pdf.

[25] Oracle, “Fusion Middleware Concepts Guide,” p. 4, 2015, [Online]. Available: http://docs.oracle.com/cd/E21764_01/core.1111/e10103/intro.htm#ASCON109.

[26] T. I. Crew, “Conseil en Systèmes d’Information, Urbanisation, Architectures et Expertise JEE The Best Way to Predict The Future Is To Invent IT™,” 2015. .

[27] M. Weiser, “The computer for the 21st Century,” *Sci. Am.*, vol. 256, no. 3, pp. 94–105, 1991.

[28] A. Kevin, “That ’ Internet of Things ’ Thing- In the real world, things matter more than ideas,” *RFiD J.*, p. 4986, 2010, [Online]. Available: http://www.itrco.jp/libraries/RFIDjournal-That Internet of Things Thing.pdf.

[29] C. C. Aggarwal, N. Ashish, and A. Sheth, “The Internet of Things: A Survey from the Data-Centric Perspective,” in *Managing and Mining Sensor Data*, vol. 9781461463, no. February 2014, Boston, MA: Springer US, 2013, pp. 383–428.

[30] D. Miorandi, S. Sicari, F. De Pellegrini, and I. Chlamtac, “Internet of things: Vision, applications and research challenges,” *Ad Hoc Networks*, vol. 10, no. 7, pp. 1497–1516, 2012, doi: 10.1016/j.adhoc.2012.02.016.

[31] L. Atzori, A. Iera, and G. Morabito, “The Internet of Things: A survey.,” *Comput. Networks*, vol. 54, no. 15, pp. 2787–2805, 2010.

[32] H. Mei and Zhanghang, “Business intelligence architecture based on internet of things,” *J. Theor. Appl. Inf. Technol.*, vol. 50, no. 1, pp. 90–95, 2013.

[33] P.-J. Benghozi, S. Bureau, F. Massit-Folléa, C. Waroquiers, and S. Davidson, *L’internet*

des objets: quels enjeux pour l'Europe, Éd. de la. 2009.

[34] M. Weill and M. Souissi, "L'Internet des objets : concept ou réalité ?," in *Annales des Mines - Réalités industrielles*, vol. Novembre 2, no. 4, 2010, p. 90.

[35] "INFSO D.4 Networked Enterprise & RFID INFSO G.2 Micro & Nanosystems, in: Co-operation with the Working Group RFID of the ETP EPOSS, Internet of Things in 2020, Roadmap for the Future, Version 1.1, 27 May 2008," *Internet Things 2020, Roadmap Futur.*, 2008.

[36] Casagras, "CASAGRAS Final Report: RFID and the Inclusive Model for the Internet of Things," *Sci. Am.*, vol. 291 (4), pp. 10–12, 2009, [Online]. Available: http://www.weave.de/linklisten/internetofthings0611.

[37] T. Teixeira, S. Hachem, V. Issarny, and N. Georgantas, "Service oriented middleware for the internet of things: A perspective (invited paper)," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 6994 LNCS, no. January, pp. 220–229, 2011, doi: 10.1007/978-3-642-24755-2_21.

[38] les experts Ooreka, "Système RFID : définition et fonctionnement d'un système RFID," 2015. https://rfid.ooreka.fr/comprendre/systeme-rfid.

[39] J. A. Stankovic, "Wireless sensor networks," *IEEE Comput. Soc.*, vol. 41, no. 10, pp. 92–95, 2008.

[40] N. Daniel, R. Marcel, and K. Daniel, "Livre blanc Machine To Machine enjeux et perspectives: Orange Business Services," in *Syntec informatique*, Fing, Ed. 2006, p. 40.

[41] J. Gubbi, R. Buyya, S. Marusic, and M. Palaniswami, "Internet of Things (IoT): A vision, architectural elements, and future directions," *Futur. Gener. Comput. Syst.*, vol. 29, no. 7, pp. 1645–1660, 2013, doi: 10.1016/j.future.2013.01.010.

[42] D. Niyato, E. Hossain, and S. Camorlinga, "Remote patient monitoring service using heterogeneous wireless access networks: Architecture and optimization," *IEEE J. Sel. Areas Commun.*, vol. 27, no. 4, pp. 412–423, 2009, doi: 10.1109/JSAC.2009.090506.

[43] A. Zanella, N. Bui, A. Castellani, L. Vangelista, and M. Zorzi, "Internet of things for smart cities," *IEEE Internet Things J.*, vol. 1, no. 1, pp. 22–32, 2014, doi: 10.1109/JIOT.2014.2306328.

[44] W. Z. Khan, Y. Xiang, M. Y. Aalsalem, and Q. Arshad, "Mobile phone sensing systems: A survey," *IEEE Commun. Surv. Tutorials*, vol. 15, no. 1, pp. 402–427, 2013, doi: 10.1109/SURV.2012.031412.00077.

[45] A. Kansal, S. Nath, J. Liu, and F. Zhao, "SenseWeb: An infrastructure for shared sensing," *IEEE Multimed.*, vol. 14, no. 4, pp. 8–13, 2007, doi: 10.1109/MMUL.2007.82.

[46] J. Cao, M. Li, S. Zhang, and Q. Den, "Composing web services based on agent and workflow," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 3032, pp. 948–955, 2004, doi: 10.1007/978-3-540-24679-4_157.

[47] I. Müller, R. Kowalczyk, and P. Braun, "Towards agent-based coalition formation for service composition," *Proc. - 2006 IEEE/WIC/ACM Int. Conf. Intell. Agent Technol. (IAT 2006 Main Conf. Proceedings), IAT'06*, pp. 73–80, 2006, doi: 10.1109/IAT.2006.122.

[48] A. Yachir, "Composition dynamique de services sensibles au contexte dans les systèmes intelligents ambiants," 2015.

[49] Z. Yang and D. Li, "IoT information service composition driven by user requirement,"

Proc. - 17th IEEE Int. Conf. Comput. Sci. Eng. CSE 2014, Jointly with 13th IEEE Int. Conf. Ubiquitous Comput. Commun. IUCC 2014, 13th Int. Symp. Pervasive Syst., pp. 1509–1513, 2015, doi: 10.1109/CSE.2014.280.

[50] R. Yang, B. Li, and C. Cheng, "A petri net-based approach to service composition and monitoring in the IOT," in *Proceedings - 2014 Asia-Pacific Services Computing Conference, APSCC 2014*, 2014, pp. 16–22, doi: 10.1109/APSCC.2014.11.

[51] I. R. Chen, J. Guo, and F. Bao, "Trust management for service composition in SOA-based IoT systems," in *IEEE Wireless Communications and Networking Conference, WCNC*, 2014, vol. 4, no. 4, pp. 3444–3449, doi: 10.1109/WCNC.2014.6953138.

[52] Y. BENAZZOUZ, "Context discovery for the automatic adaptation of services in ambient intelligence," 2011.

[53] S. Cherrier, "Architecture and application protocols for services choreography in the Internet of Things," École doctorale MSTIC, Université Paris-Est, 2013.

[54] R. T. Fielding and R. N. Taylor, "Principled design of the modern Web architecture," *ACM Trans. Internet Technol.*, vol. 2, no. 2, pp. 115–150, 2002, doi: 10.1145/514183.514185.

[55] M. zur Muehlen, J. V. Nickerson, and K.D, "Swenson, Developing web services choreography standards-the case of rest vs. soap," *Decis. Support Syst. 40*, vol. 1, pp. 9–29, 2005.

[56] K. Li and L. Jiang, "The research of web services composition based on context in Internet of Things," *2012 IEEE Int. Conf. Comput. Sci. Autom. Eng.*, vol. 1, pp. 160–163, 2012, doi: 10.1109/CSAE.2012.6272570.

[57] G. Chen, J. Huang, B. Cheng, and J. Chen, "A Social Network Based Approach for IoT Device Management and Service Composition," *2015 IEEE World Congr. Serv.*, pp. 1–8, 2015, doi: 10.1109/SERVICES.2015.9.

[58] K. Hwang, C., Yoon, "Multiple Attribute Decision Making: Theory and Applications," *Tsinghua Univ. Press*, p. 1, 1981.

[59] I. R. Chen, J. Guo, and F. Bao, "Trust Management for SOA-Based IoT and Its Application to Service Composition," *IEEE Trans. Serv. Comput.*, vol. 9, no. 3, pp. 482–495, 2016, doi: 10.1109/TSC.2014.2365797.

[60] D. Guinard, V. Trifa, S. Karnouskos, P. Spiess, and D. Savio, "Interacting with the SOA-based internet of things: Discovery, query, selection, and on-demand provisioning of web services," *IEEE Trans. Serv. Comput.*, vol. 3, no. 3, pp. 223–235, 2010, doi: 10.1109/TSC.2010.3.

[61] I. R. Chen, F. Bao, and J. Guo, "Trust-Based Service Management for Social Internet of Things Systems," *IEEE Trans. Dependable Secur. Comput.*, vol. 13, no. 6, pp. 684–696, 2016, doi: 10.1109/TDSC.2015.2420552.

[62] R. Yang, B. Li, and C. Cheng, "Adaptable service composition for intelligent logistics: A middleware approach," in *Proceedings - 2014 International Conference on Cloud Computing and Big Data, CCBD 2014*, 2014, pp. 75–82, doi: 10.1109/CCBD.2014.10.

[63] B. Li, R. Yang, and Y. Hu, "An Experimental Study for Intelligent Logistics: A Middleware Approach," *Chinese J. Electron.*, vol. 25, no. 3, pp. 561–569, 2016, doi: 10.1049/cje.2016.05.024.

[64] L. Li, Z. Jin, G. Li, L. Zheng, and Q. Wei, "Modeling and analyzing the reliability and

cost of service composition in the IoT: A probabilistic approach," *Proc. - 2012 IEEE 19th Int. Conf. Web Serv. ICWS 2012*, pp. 584–591, 2012, doi: 10.1109/ICWS.2012.25.

[65] A. Kouicem, A. Chibani, A. Tari, Y. Amirat, and Z. Tari, "Dynamic services selection approach for the composition of complex services in the web of objects," in *2014 IEEE World Forum on Internet of Things, WF-IoT 2014*, 2014, pp. 298–303, doi: 10.1109/WF-IoT.2014.6803176.

[66] L. Zhang, S. Yu, X. Ding, and X. Wang, "Research on IOT RESTful web service asynchronous composition based on BPEL," *Proc. - 2014 6th Int. Conf. Intell. Human-Machine Syst. Cybern. IHMSC 2014*, vol. 1, pp. 62–65, 2014, doi: 10.1109/IHMSC.2014.23.

[67] L. Li, T. Tang, and W. Chou, "An XML Based Monadic Framework for REST Service Compositions," in *Proceedings - 2015 IEEE International Conference on Web Services, ICWS 2015*, 2015, pp. 487–494, doi: 10.1109/ICWS.2015.71.

[68] H. Abelson, G. J. Sussman, and J. Sussman, *Structure and Interpretation of Computer Programs, second edition*, 2nd ed. Cambridge, Massachusetts London, England: MIT Press, 1996.

[69] L. Bossi, S. Braghin, and A. Trombetta, "Multidimensional reputation network for service composition in the internet of things," *Proc. - 2014 IEEE Int. Conf. Serv. Comput. SCC 2014*, pp. 685–692, 2014, doi: 10.1109/SCC.2014.95.

[70] K. Dar, A. Taherkordi, R. Vitenberg, R. Rouvoy, and F. Eliassen, "Adaptable service composition for very-large-scale Internet of Things systems," *Proc. Work. Posters Demos Track - PDT '11*, no. DECEMBER, pp. 1–2, 2011, doi: 10.1145/2088960.2088971.

[71] S. S. Ara, Z. U. Shamszaman, and I. Chong, "Web-of-objects based user-centric semantic service composition methodology in the internet of things," *Int. J. Distrib. Sens. Networks*, vol. 2014, 2014, doi: 10.1155/2014/482873.

[72] A. Ciortea, O. Boissier, A. Zimmermann, and A. M. Florea, "Responsive Decentralized Composition of Service Mashups for the Internet of Things," *Proc. 6th Int. Conf. Internet Things - IoT'16*, pp. 53–61, 2016, doi: 10.1145/2991561.2991573.

[73] L. Chen, L. Kuang, and J. Wu, "MapReduce based skyline services selection for QoS-aware composition," in *Proceedings of the 2012 IEEE 26th International Parallel and Distributed Processing Symposium Workshops, IPDPSW 2012*, 2012, pp. 2035–2042, doi: 10.1109/IPDPSW.2012.253.

[74] M. Tao, J. Zuo, Z. Liu, A. Castiglione, and F. Palmieri, "Multi-layer cloud architectural model and ontology-based security service framework for IoT-based smart homes," *Futur. Gener. Comput. Syst.*, vol. 78, pp. 1040–1051, 2018, doi: 10.1016/j.future.2016.11.011.

[75] L. Barakat, S. Miles, and M. Luck, "Adaptive composition in dynamic service environments," *Futur. Gener. Comput. Syst.*, vol. 80, pp. 215–228, 2018, doi: 10.1016/j.future.2016.12.003.

[76] N. Temglit, A. Chibani, K. Djouani, and M. A. Nacer, "A Distributed Agent-Based Approach for Optimal QoS Selection in Web of Object Choreography," *IEEE Syst. J.*, pp. 1–12, 2017, doi: 10.1109/JSYST.2016.2647281.

[77] A. Urbieta, A. González-Beltrán, S. Ben Mokhtar, M. Anwar Hossain, and L. Capra, "Adaptive and context-aware service composition for IoT-based smart cities," *Futur. Gener. Comput. Syst.*, vol. 76, pp. 262–274, Nov. 2017, doi:

10.1016/j.future.2016.12.038.

[78] A. Urbieta *et al.*, “Hybrid service matchmaking in ambient assisted living environments based on context-aware service modeling,” *Clust. Comput*, vol. 18, no. 3, pp. 1171–1188, 2015, doi: 10.1007/s10586-015-0469-1.

[79] T. Baker, M. Asim, H. Tawfik, B. Aldawsari, and R. Buyya, “An energy-aware service composition algorithm for multiple cloud-based IoT applications,” *J. Netw. Comput. Appl.*, vol. 89, pp. 96–108, 2017, doi: 10.1016/j.jnca.2017.03.008.

[80] M. E. Khanouche, Y. Amirat, A. Chibani, M. Kerkar, and A. Yachir, “Energy-Centered and QoS-Aware Services Selection for Internet of Things,” *IEEE Trans. Autom. Sci. Eng.*, vol. 13, no. 3, pp. 1256–1269, 2016.

[81] Y. Zhang, J. L. Chen, and B. Cheng, “Integrating Events into SOA for IoT Services,” *IEEE Commun. Mag.*, vol. 55, no. 9, pp. 180–186, 2017, doi: 10.1109/MCOM.2017.1600359.

[82] E. Fki, S. Tazi, and K. Drira, “Automated and flexible composition based on abstract services for a better adaptation to user intentions,” *Futur. Gener. Comput. Syst.*, vol. 68, no. March, pp. 376–390, 2017, doi: 10.1016/j.future.2016.07.008.

[83] N. Temglit, A. Chibani, K. Djouani, and M. A. Nacer, “Distributed Approach for QoS Service Selection in Web of Objects,” *Procedia Comput. Sci.*, vol. 83, pp. 1170–1175, 2016, doi: 10.1016/j.procs.2016.04.240.

[84] E. Rapti, A. Karageorgos, and V. C. Gerogiannis, “Decentralised service composition using potential fields in internet of things applications,” *Procedia Comput. Sci.*, vol. 52, no. 1, pp. 700–706, 2015, doi: 10.1016/j.procs.2015.05.079.

[85] E. Rapti, A. Karageorgos, C. Houstis, and E. Houstis, “Decentralized service discovery and selection in Internet of Things applications based on artificial potential fields,” *Serv. Oriented Comput. Appl.*, vol. 11, no. 1, pp. 75–86, 2017, doi: 10.1007/s11761-016-0198-1.

[86] A. Yachir, Y. Amirat, A. Chibani, and N. Badache, “Event-Aware Framework for Dynamic Services Discovery and Selection in the Context of Ambient Intelligence and Internet of Things,” *IEEE Trans. Autom. Sci. Eng.*, vol. 13, no. 1, pp. 85–102, 2015, doi: 10.1109/TASE.2015.2499792.

[87] H. Kil, W. Nam, and R. Cha, “Optimal algorithm for Internet-of-Things service composition based on response time,” *Int. J. Web Grid Serv.*, vol. 12, no. 4, p. 388, 2016, doi: 10.1504/IJWGS.2016.10001004.

[88] L. Huo and Z. Wang, “Service composition instantiation based on cross-modified artificial Bee Colony algorithm,” *China Commun.*, vol. 13, no. 10, pp. 233–244, 2016, doi: 10.1109/CC.2016.7733047.

[89] Jue Wang, Chaocan Xiang, Meng Wang, Chang Tian, Wendong Zhao, Deng-po Dai, “A Survey on Semantic Web Services Discovery,” *Appl. Res. Comput.*, vol. 30, no. 1, pp. 7–12, 2013.

[90] K. Dar, A. Taherkordi, H. Baraki, F. Eliassen, and K. Geihs, “A resource oriented integration architecture for the Internet of Things: A business process perspective,” *Pervasive Mob. Comput.*, vol. 20, no. November 2014, pp. 145–159, 2015, doi: 10.1016/j.pmcj.2014.11.005.

[91] J. Wu, L. Chen, Q. Yu, L. Kuang, Y. Wang, and Z. Wu, “Selecting skyline services for

QoS-aware composition by upgrading MapReduce paradigm," *Cluster Comput.*, vol. 16, no. 4, pp. 693–706, Dec. 2013, doi: 10.1007/s10586-012-0240-9.

[92] S. N. Han, S. Park, G. M. Lee, and N. Crespi, "Extending the devices profile for web services standard using a REST proxy," *IEEE Internet Comput.*, vol. 19, no. 1, pp. 10–17, 2015, doi: 10.1109/MIC.2014.44.

[93] J. Guo, I. R. Chen, and J. J. P. Tsai, "A survey of trust computation models for service management in internet of things systems," *Comput. Commun.*, vol. 97, pp. 1–14, 2017, doi: 10.1016/j.comcom.2016.10.012.

[94] S. N. Han, I. Khan, G. M. Lee, N. Crespi, and R. H. Glitho, "Service composition for IP smart object using realtime Web protocols: Concept and research challenges," *Comput. Stand. Interfaces*, vol. 43, pp. 79–90, 2016, doi: 10.1016/j.csi.2015.08.006.

[95] A. B. M. Zorzi, A. Gluhak, S. Lange, M. Zorzi, A. Gluhak, S. Lange, and A. Bassi, "From today's INTRAnet of things to a future INTERnet of things: a wireless- and mobility-related view," *IEEE Wirel. Commun*, vol. 17 (6), no. 6, pp. 44–51, 2010, doi: 10.1109/MWC.2010.5675777.

[96] S. E. Lee, M. Choi, and S. Kim, "How and what to study about IoT: Research trends and future directions from the perspective of social science," *Telecomm. Policy*, vol. 41, no. 10, pp. 1056–1067, 2017, doi: 10.1016/j.telpol.2017.09.007.

[97] L. Atzori, A. Iera, and G. Morabito, "Understanding the Internet of Things: definition, potentials, and societal role of a fast evolving paradigm," *Ad Hoc Networks*, vol. 56, pp. 122–140, 2017, doi: 10.1016/j.adhoc.2016.12.004.

[98] I. Khajenasiri, A. Estebsari, M. Verhelst, and G. Gielen, "A Review on Internet of Things Solutions for Intelligent Energy Control in Buildings for Smart City Applications," *Energy Procedia*, vol. 111, no. September 2016, pp. 770–779, 2017, doi: 10.1016/j.egypro.2017.03.239.

[99] P. P. Ray, "A survey on Internet of Things architectures," *J. King Saud Univ. - Comput. Inf. Sci.*, no. October, 2016, doi: 10.1016/j.jksuci.2016.10.003.

[100] A. Mukherjee, "Physical-Layer Security in the Internet of Things: Sensing and Communication Confidentiality Under Resource Constraints," *Proc. IEEE*, vol. 103, no. 10, pp. 1747–1761, 2015, doi: 10.1109/JPROC.2015.2466548.

[101] N. Hesami Rostami, E. Kheirkhah, and M. Jalali, "Web Services Composition Methods and Techniques: A Review," *Int. J. Comput. Sci. Eng. Inf. Technol.*, vol. 3, no. 6, pp. 15–29, 2013, doi: 10.5121/ijcseit.2013.3603.

[102] S. G. H. Tabatabaei and S. Ibrahim, "A Review of Web Service Composition Approaches," 2011.

[103] K. N. Shah, S. Santoki, H. Ghetia, and K. Ra, "A Review on Web Service Composition Techniques," *Int. J. Eng. Res. Appl.*, vol. 3, no. 3, pp. 929–934, 2013.

[104] M. Abbas, E. Sabeil, and A. Abdul Manaf, *Comparative Evaluation of Semantic Web Service Composition Approaches*, vol. 180, no. PART 2. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011.

[105] S. Pakari, E. Kheirkhah, and M. Jalali, "Web Service Discovery Methods and Techniques : a Review," *Int. J. Comput. Sci. Eng. Inf. Technol.*, vol. 4, no. 1, pp. 1–14, 2014, doi: 10.5121/ijcseit.2014.4101.

[106] A. L. Lemos, F. Daniel, and B. Benatallah, "Web Service Composition: A Survey of

Techniques and Tools," *ACM Comput. Surv.*, vol. 48, no. 3, pp. 1–41, 2015, doi: 10.1145/2831270.

[107] Y. Syu, Y. Y. Fanjiang, J. Y. Kuo, and S. P. Ma, "A review of the automatic web service composition surveys," *Proc. - 2014 IEEE Int. Conf. Semant. Comput. ICSC 2014*, pp. 199–202, 2014, doi: 10.1109/ICSC.2014.41.

[108] A. Jula, E. Sundararajan, and Z. Othman, "Cloud computing service composition: A systematic literature review," *Expert Syst. Appl.*, vol. 41, no. 8, pp. 3809–3824, 2014, doi: 10.1016/j.eswa.2013.12.017.

[109] *NEXOF-RA. Deliverable D10.1: Requirements Report IST- FP7-216446*, NESSI Open. 2009.

[110] M. Singh and G. Baranwal, "Quality of Service (QoS) in Internet of Things," *Proc. - 2018 3rd Int. Conf. Internet Things Smart Innov. Usages, IoT-SIU 2018*, 2018, doi: 10.1109/IoT-SIU.2018.8519862.

[111] Y. Huo, P. Qiu, J. Zhai, D. Fan, and H. Peng, "Multi-objective service composition model based on cost-effective optimization," *Appl. Intell.*, vol. 48, no. 3, pp. 651–669, 2018, doi: 10.1007/s10489-017-0996-y.

[112] X. Zhang, J. Geng, J. Ma, H. Liu, S. Niu, and W. Mao, "A hybrid service selection optimization algorithm in internet of things," *Eurasip J. Wirel. Commun. Netw.*, vol. 4, no. 1, 2021, doi: 10.1186/s13638-020-01883-2.

[113] X. Zhang, J. Geng, J. Ma, H. Liu, and S. Niu, "A QoS-driven Service Selection Optimization Algorithm for Internet of Things," *Researchsquare*, 2020.

[114] M. Burhan, R. A. Rehman, B. Khan, and B. S. Kim, "IoT elements, layered architectures and security issues: A comprehensive survey," *Sensors (Switzerland)*, vol. 18, no. 9, pp. 1–37, 2018, doi: 10.3390/s18092796.

[115] C. Puliafito, E. Mingozzi, F. Longo, A. Puliafito, and O. Rana, "Fog computing for the Internet of Things: A survey," *ACM Trans. Internet Technol.*, vol. 19, no. 2, 2019, doi: 10.1145/3301443.

[116] M. Haghi Kashani, A. M. Rahmani, and N. Jafari Navimipour, "Quality of service-aware approaches in fog computing," *Int. J. Commun. Syst.*, vol. 33, no. 8, pp. 1–34, 2020, doi: 10.1002/dac.4340.

[117] M. Chalmers, D., & Sloman, "A survey of quality of service in mobile computing environments," *IEEE Commun. Surv. Tutorials*, vol. 2, pp. 2–10, 1999, doi: 10.1109/comst.1999.5340514.

[118] T. Buchholz, A. Küpper, and M. Schiffers, "Quality of Context: What it is and why we need it," *Proc. 10th Work. OpenView Univ. Assoc. OVUA'03*, no. January 2003, pp. 1–14, 2003, doi: 10.1.1.147.565.

[119] A. Manzoor, H.-L. Truong, and S. Dustdar, "Quality of Context: Models and Applications for Context-aware Systems in Pervasive Environments," *Knowl. Eng. Rev.*, vol. 20, no. 2, pp. 117–125, 2004, doi: 10.1017/S000000000000000.

[120] Z. Houhamdi and B. Athamena, "Identity identification and management in the internet of things," *Int. Arab J. Inf. Technol.*, vol. 17, no. 4 Special Issue, pp. 645–654, 2020, doi: 10.34028/iajit/17/4A/9.

[121] N. Kashyap and C. A. Kumari, "Hyper-heuristic approach for service composition in internet of things," *Electron. Gov.*, vol. 14, no. 4, pp. 321–339, 2018, doi:

10.1504/EG.2018.095546.

[122] X. Zhao, B. Song, P. Huang, Z. Wen, J. Weng, and Y. Fan, “An improved discrete immune optimization algorithm based on PSO for QoS-driven web service composition,” *Appl. Soft Comput. J.*, vol. 12, no. 8, pp. 2208–2216, 2012, doi: 10.1016/j.asoc.2012.03.040.

[123] “W3C.” https://www.w3.org/.

[124] W. Qiufen and D. Liang, “A HEURISTIC GENETIC ALGORITHM FOR SOLVING 0-1 KNAPSACK PROBLEM,” *Comput. Appl. Softw.*, vol. 30, no. 2, pp. 33–37, 2013, [Online]. Available: http://en.cnki.com.cn/Article_en/CJFDTOTAL-JYRJ201302009.htm.

[125] D. E. Goldberg, “Genetic algorithms in search, optimization, and machine learning,” *Choice Rev. Online*, vol. 27, no. 02, pp. 27-0936-27–0936, 1989, doi: 10.5860/choice.27-0936.

[126] S. Siva Sathya and M. V. Radhika, “Convergence of nomadic genetic algorithm on benchmark mathematical functions,” *Appl. Soft Comput. J.*, vol. 13, no. 5, pp. 2759–2766, 2013, doi: 10.1016/j.asoc.2012.11.011.

[127] E. Al-Masri and Q. H. Mahmoud, “Investigating web services on the world wide web,” 2008, doi: 10.1145/1367497.1367605.