Autonomie, sécurité et QoS de bout en bout dans un environnement de Cloud Computing

Mohamad Hamze

SPIM

Thèse de Doctorat

école doctorale **sciences pour l'ingénieur et microtechniques**

UNIVERSITÉ DE BOURGOGNE

Autonomie, sécurité et QoS de bout en bout dans un environnement de Cloud Computing

MOHAMAD HAMZE

Laboratoire Électronique Informatique et Image

SPIM

Thèse de Doctorat

école doctorale **sciences pour l'ingénieur et microtechniques**

UNIVERSITÉ DE BOURGOGNE

N° | X | X | X |

THÈSE présentée par

MOHAMAD HAMZE

pour obtenir le

Grade de Docteur de
l'Université de Bourgogne

Spécialité : **Informatique**

Autonomie, sécurité et QoS de bout en bout dans un environnement de Cloud Computing

Unité de Recherche :
Le2i - Laboratoire Electronique, Informatique et Image

Soutenue publiquement le 07 décembre 2015 devant le Jury composé de :

FRANCINE KRIEF	Rapporteur	Professeure à Bordeaux INP
DJAMAL BENSLIMANE	Rapporteur	Professeur à l'Université de Lyon
YEHIA TAHER	Examinateur	Maître de conférences à l'Université de Versailles
OLIVIER TOGNI	Directeur de thèse	Professeur à l'Université de Bourgogne
NADER MBAREK	Co-encadrant de thèse	Maître de conférences à l'Université de Bourgogne

RÉSUMÉ

De nos jours, le Cloud Networking est considéré comme étant l'un des domaines de recherche innovants au sein de la communauté de recherche du Cloud Computing. Les principaux défis dans un environnement de Cloud Networking concernent non seulement la garantie de qualité de service (QoS) et de sécurité mais aussi sa gestion en conformité avec un accord de niveau de service (SLA) correspondant. Dans cette thèse, nous proposons un Framework pour l'allocation des ressources conformément à un SLA établi de bout en bout entre un utilisateur de services Cloud (CSU) et plusieurs fournisseurs de services Cloud (CSP) dans un environnement de Cloud Networking (architectures d'inter-Cloud Broker et Fédération). Nos travaux se concentrent sur les services Cloud de types NaaS et IaaS. Ainsi, nous proposons l'auto-établissement de plusieurs types de SLA ainsi que la gestion autonome des ressources de Cloud correspondantes en conformité avec ces SLA en utilisant des gestionnaires autonomes spécifiques de Cloud. De plus, nous étendons les architectures et les SLA proposés pour offrir un niveau de service intégrant une garantie de sécurité. Ainsi, nous permettons aux gestionnaires autonomes de Cloud d'élargir leurs objectifs de gestion autonome aux fonctions de sécurité (auto-protection) tout en étudiant l'impact de la sécurité proposée sur la garantie de QoS. Enfin, nous validons notre architecture avec différents scénarios de simulation. Nous considérons dans le cadre de ces simulations des applications de vidéoconférence et de calcul intensif afin de leur fournir une garantie de QoS et de sécurité dans un environnement de gestion autonome des ressources du Cloud. Les résultats obtenus montrent que nos contributions permettent de bonnes performances pour ce type d'applications. En particulier, nous observons que l'architecture de type Broker est la plus économique, tout en assurant les exigences de QoS et de sécurité. De plus, nous observons que la gestion autonome des ressources du Cloud permet la réduction des violations, des pénalités et limite l'impact de la sécurité sur la garantie de la QoS.

Mots-clés : Cloud Computing, Cloud Networking, Inter-Cloud, Service Level Agreement, Qualité de Service, Sécurité, Gestion Autonome, Videoconférence, Calcul Intensif.

ABSTRACT

Today, Cloud Networking is one of the recent research areas within the Cloud Computing research communities. The main challenges of Cloud Networking concern Quality of Service (QoS) and security guarantee as well as its management in conformance with a corresponding Service Level Agreement (SLA). In this thesis, we propose a framework for resource allocation according to an end-to-end SLA established between a Cloud Service User (CSU) and several Cloud Service Providers (CSPs) within a Cloud Networking environment (Inter-Cloud Broker and Federation architectures). We focus on NaaS and IaaS Cloud services. Then, we propose the self-establishing of several kinds of SLAs and the self-management of the corresponding Cloud resources in conformance with these SLAs using specific autonomic cloud managers. In addition, we extend the proposed architectures and the corresponding SLAs in order to deliver a service level taking into account security guarantee. Moreover, we allow autonomic cloud managers to expand the self-management objectives to security functions (self-protection) while studying the impact of the proposed security on QoS guarantee. Finally, our proposed architecture is validated by different simulation scenarios. We consider, within these simulations, videoconferencing and intensive computing applications in order to provide them with QoS and security guarantee in a Cloud self-management environment. The obtained results show that our contributions enable good performances for these applications. In particular, we observe that the Broker architecture is the most economical while ensuring QoS and security requirements. In addition, we observe that Cloud self-management enables violations and penalties' reduction as well as limiting security impact on QoS guarantee.

Keyword : Cloud Computing, Cloud Networking, Inter-Cloud, Service Level Agreement, Quality of Service, Security, Self-management, Videoconferencing, Intensive Computing.

REMERCIEMENTS

En préambule à ce travail, je tiens tout d'abord à remercier les responsables de cette thèse, Pr. Olivier TOGNI et Dr. Nader MBAREK, pour m'avoir fait confiance malgré les connaissances plutôt légères que j'avais au début de ma thèse en octobre 2012 sur le Cloud, puis pour m'avoir guidé, encouragé, et conseillé pendant les trois ans tout en me laissant une grande liberté et en me faisant l'honneur de me déléguer plusieurs responsabilités dont j'espère avoir été à la hauteur. Leur disponibilité et leurs conseils ont été d'une grande qualité. Je ne sais pas comment exprimer ma gratitude à ces deux personnes autrement qu'en leur promettant d'agir comme eux avec des étudiants dans ma situation, si un jour l'occasion m'en est donnée.

Pr. Francine KRIEF et Pr. Djamal BENSLIMANE ont accepté d'être les rapporteurs de cette thèse, et je les en remercie, de même que pour leur participation au Jury.

Je remercie aussi bien mes collèges pour les discussions que j'ai eu la chance d'avoir avec eux.

Pour leurs encouragements et leur assistance aussi bien matérielle que morale qui m'ont permis de faire cette thèse dans de bonnes conditions, mes plus profonds remerciements vont à mes parents. Tout au long de mon cursus, ils m'ont toujours soutenu, encouragé et aidé. Qu'ils trouvent, dans la réalisation de ce travail, l'aboutissement de leurs efforts ainsi que l'expression de ma plus affectueuse gratitude. Je remercie aussi chaudement Ghenwa, ma femme qui m'a soutenu durant les trois ans de ma thèse. Elle a su me donner toutes les chances pour réussir. Je remercie DIEU qui m'a béni avec mon premier fils Ali qui est né à quelques jours de la finalisation de ma thèse. Je souhaiterais remercier également le reste de la famille, mes amis et mes proches pour leur encouragement tout au long de la réalisation de ce mémoire.

Je tiens aussi à mentionner le plaisir que j'ai eu à travailler au sein du laboratoire Le2i, l'école doctorale SPIM et l'université de Bourgogne, et je remercie ici tous les membres de ces institutions.

Merci à tous et à toutes.

SOMMAIRE

LISTE DES FIGURES

LISTE DES TABLES

LISTE DES ABRÉVIATIONS

AD	Autonomic cloud Domain	Domaine autonome de Cloud
AM	Autonomic cloud Manager	Gestionnaire autonome de Cloud
API	Application Programming Interface	Interface de programmation d'application
ASP	Application Service Provider	Fournisseur de services application
ATM	Asynchronous Transfer Mode	Mode de transfert asynchrone
B_iSLA	BoD inter-cloud Service Level Agreement	Accord de niveau de service pour la BoD
BoD	Bandwidth on Demand	bande passante à la demande
CPU	Central Processing Unit	Unité centrale de traitement
CSP	Cloud Service Provider	Fournisseur de services Cloud
CSU	Cloud Service User	Utilisateur de services Cloud
DC	Data Center	Centre de données
D_iSLA	DC inter-cloud Service Level Agreement	Accord de niveau de service pour le DC
DnAM	Datacenter network Autonomic Manager	Gestionnaire autonome de DC
GUI	Graphical User Interface	Interface d'utilisateur graphique
hAM	hypervisor Autonomic Manager	Gestionnaire autonome d'hyperviseur
IaaS	Infrastructure as a Service	Infrastructure en tant que service
iAM	inter-cloud Autonomic Manager	Gestionnaire autonome dl'Inter-Cloud
iSLA	inter-cloud Service Level Agreement	Accord de niveau de service d'Inter-Cloud
NaaS	Network as a Service	Réseau en tant que service
nAM	network Autonomic Manager	Gestionnaire autonome de réseau
PaaS	Platform as a Service	Plateforme en tant que service
QoS	Quality of Service	Qualité de service
RAM	Random Access Memory	Mémoire à accès direct
SaaS	Software as a Service	Logiciel en tant que service
SLA	Service Level Agreement	Accord de niveau de service
TTP	Trust Third Party	Tierce partie de confiance
VM	Virtual Machine	Machine virtuelle
VPN	Virtual Private Network	Réseau Privé Virtuel
XML	Extensible Markup Language	Langage à balise extensible

CHAPITRE 1
INTRODUCTION

1.1/ CONTEXTE

De nos jours, l'utilisation d'Internet et des nouvelles technologies, pour satisfaire l'évolution continue des besoins de différents types d'utilisateurs (affaire, particulier), fait partie de la vie quotidienne. Toute information est disponible partout dans le monde à tout moment. Cela n'était pas possible il y a quelques années. Récemment, un nombre important de possibilités d'accès à l'information publique et privée sont apparues. Ainsi, nous avons un accès généralisé à grand débit à Internet grâce au déploiement de dispositifs fixes, mobiles ou encore sans fil qui permettent la connexion à Internet sans presque se soucier de la limitation géographique.

Aujourd'hui, différents types d'utilisateurs consultent leurs courriers en ligne via des Webmail, rédigent des documents de collaboration en utilisant les navigateurs web, exécutent des applications et stockent des données dans des serveurs situés sur Internet et non dans leurs propres ordinateurs. De plus, ces services ainsi que d'autres sont utilisés d'une façon transparente pour l'utilisateur et sont donc perçus comme étant des services offerts par un nuage (Cloud) sans en connaître les détails. Cela signifie que de nombreux utilisateurs et organisations peuvent éviter l'installation de certaines applications sur leurs infrastructures ou peuvent avoir plus de puissance de calcul en utilisant les ressources de ce Cloud grâce à Internet. De plus, ces différents utilisateurs peuvent construire leurs propres Clouds privés et les administrer selon leurs propres politiques de gestion. Ainsi, la plupart des entreprises essayent de réduire leurs coûts d'exploitation et de traitement grâce à des techniques de virtualisation. Ces techniques et usages ont conduit à l'émergence d'un nouveau concept appelé Cloud Computing qui permet d'offrir plusieurs types de services avec une meilleure utilisation des ressources des infrastructures et une réduction de leurs coûts d'exploitation.

Le Cloud Computing est un terme utilisé pour décrire à la fois une plateforme et un type d'application. En tant que plateforme, le Cloud Computing fournit, configure et reconfigure les serveurs. Ces serveurs peuvent être peuvent être des machines physiques ou encore des machines virtuelles. D'autre part, le Cloud Computing permet à des applications d'être étendues pour devenir accessible à travers Internet. A cet effet, des grands centres de données et des serveurs puissants sont utilisés pour héberger ces applications qui peuvent être utilisées grâce à des services Web. Le Cloud Computing est devenu l'une des plus importantes technologies ces derniers temps et fait l'objet de plusieurs études dans différents domaines eu égard à la multiplicité des possibilités d'offre de services qu'il propose.

De plus, le système d'information a toujours été considéré comme un composant très important dans différents types d'organisations et ce du point de vue coût d'exploitation

(OPEX : Operational Expenditure) et celui de capital (CAPEX : Capital Expenditure). Le Cloud Computing offre à ces organisations plus de flexibilité en ce qui concerne le fonctionnement des infrastructures et la réduction de coûts. Il est devenu une partie intégrante des modèles technologiques et commerciaux, et a forcé les entreprises à s'adapter à des nouvelles stratégies. Ainsi, la demande accrue des services du Cloud Computing est à l'origine du développement de nouvelles offres sur le marché, représentant divers modèles de service et de livraison. Cependant, le Cloud Computing reste une technologie relativement nouvelle. Par conséquent, la plupart des entreprises ne sont pas très confiantes lors de son adoption à cause de plusieurs défis qui restent à relever. Ainsi, il subsiste beaucoup de doutes concernant la qualité des services offerts, la sécurité des données et la gestion efficace des ressources pour les entreprises.

D'une manière générale, il y a des paramètres critiques de qualité de service (QoS : Quality of Service) à prendre en compte lors du traitement d'une demande de service par le Cloud. Par exemple, il faut garantir un temps de réponse convenable des machines virtuelles (VM) afin de satisfaire les exigences de l'utilisateur de Cloud (CSU : Cloud Service User). De plus, le délai et la gigue sont des paramètres de qualité de services très importants pour les applications interactives temps réel telles que la vidéoconférence dans un environnement de Cloud Computing. Cependant, plusieurs fournisseurs de Cloud (CSP : Cloud Service Provider) proposent dans cet environnement de Cloud Computing des services similaires. Il devient alors difficile à l'utilisateur de choisir le fournisseur qui convient le mieux à ses besoins. Ainsi, un élément de différenciation entre les différents environnements de Cloud est le niveau de service qui sera garanti par cet environnement. Ce niveau de service peut être établi entre le fournisseur et le client du Cloud par le biais d'un contrat (SLA : Service Level Agreement). Par conséquent, il faut assurer la cohérence entre les exigences de qualité de service (QoS) demandée par le CSU et les SLA proposés par les différents CSP pour permettre à plusieurs CSP sélectionnés de collaborer ensemble afin de répondre aux exigences du CSU. Dans le cas où toutes les ressources d'un CSP sont allouées, ce dernier ne peut pas satisfaire de nouvelles demandes des CSU. De plus, il est difficile pour un CSP d'avoir des centres de données (DC) avec une répartition géographique mondiale. Par conséquent, le Cloud devrait être conçu comme un environnement multifournisseur, capable d'offrir la possibilité de faire migrer un service d'un CSP à un autre et de localiser la meilleure ressource, non seulement en termes de capacité de calcul ou de stockage, mais aussi de connectivité, de bande passante et de délai d'acheminement. Ainsi, la garantie du niveau de service doit être assurée de bout en bout à travers les différents Clouds concernés par l'offre y compris les réseaux qui relient le CSU et les CSP, les CSP entre eux, ou encore les ressources dans un centre de données (DC) d'un CSP.

Les offres de service doivent être assurées dans le cadre du Cloud avec un minimum d'intervention humaine en termes d'installation, configuration, maintenance et d'une façon générale en termes de fonctions de gestion. Ainsi, les pannes accidentelles causées par les défauts du logiciel, du matériel ou du réseau conduisent à des violations de SLA. Ce dernier n'est pas respecté lorsque le CSP fournit un service qui ne permet pas de satisfaire les exigences du CSU établies dans le SLA. Par conséquent, il est nécessaire d'assurer l'auto-établissement du SLA et la gestion autonome des ressources du Cloud pour réduire les violations et les pénalités qui en résultent.

La sécurité est l'un des défis à relever les plus importants pour l'adoption du Cloud. En effet, les données sont des éléments très précieux pour les utilisateurs et les entreprises veulent toujours s'assurer que ces données sont sécurisées. La confiance des utilisateurs

est naturellement plus importante lorsque les données sont traitées, stockées et contrôlées en interne. L'externalisation du traitement ou encore du stockage de ces données dans un environnement de Cloud Computing s'accompagne d'un risque de sécurité. En effet, les données peuvent être compromises à différentes étapes de leur cycle de vie lorsqu'elles sont stockées ou traitées dans le Cloud. Les risques de sécurité peuvent apparaître lors du transfert de ces données du réseau interne de l'entreprise vers le Cloud pour y être stockées ou traitées, ou encore lors du processus de restauration des données. Ainsi, la transition vers une utilisation massive des services du Cloud s'accompagne de plusieurs défis de sécurité et de confidentialité, principalement en raison de la nature dynamique du Cloud et le fait que dans cet environnement les composants logiciels et matériels qui permettent d'offrir un service appartiennent à de multiples domaines de confiance. Par conséquent, la garantie des services de sécurité dans un environnement de Cloud est beaucoup plus difficile.

1.2/ Objectifs de la thèse

Dans le cadre de cette thèse nous réalisons un état de l'art afin de relever les défis de recherches dans un environnement de Cloud Computing, Cloud Networking et Inter-Cloud. Cet état de l'art nous permettra de réaliser l'objectif de ce travail de recherche qui est de proposer un Framework pour la garantie de QoS de bout en bout dans un tel environnement grâce à l'établissement autonome d'un accord de niveau de service (SLA) qui répond aux exigences du CSU tout en assurant une optimisation des ressources lors de la sélection des offres de services proposées par les CSP. Ces ressources sont offertes comme un service de type IaaS (Infrastructure as a Service) ou de type NaaS (Network as a Service). De plus, nous envisageons de calculer les différents coûts relatifs à cette garantie de QoS. Ce Framework doit être par la suite géré d'une façon autonome grâce aux concepts innovants de la gestion autonome (Self-Management) qui découlent du paradigme appelé Autonomic Computing afin d'optimiser et automatiser les différentes fonctions de gestion dans un environnement de type Cloud. Ensuite, un autre défi de recherche que nous relevons dans le cadre de cette thèse est d'assurer la confiance nécessaire à la collaboration entre le CSU et les CSP et ce en améliorant le Framework qui sera proposé afin d'offrir un niveau de service intégrant une offre de sécurité grâce à des SLA comportant de nouveaux paramètres de sécurité. De plus, cette offre de sécurité sera gérée d'une façon autonome tout en étudiant l'impact de la sécurité sur la QoS dans un environnement de type Cloud.

1.3/ Plan de la thèse

La suite de cette thèse est structurée comme suit.

Tout d'abord, le chapitre 2 présente le contexte dans lequel s'inscrivent les travaux présentés dans cette thèse. Ce contexte concerne le Cloud Computing, le Cloud Networking et l'Inter-Cloud avec leurs définitions, leurs caractéristiques, leurs modèles de services et de déploiement. De plus, ce chapitre aborde la standardisation, les outils d'implémentation et de simulation ainsi que les produits commerciaux dans ces types d'environnements.

Le chapitre 3 présente l'état de l'art relatif à la QoS, la sécurité et la gestion autonome dans un environnement de Cloud Computing. Nous présentons les définitions et les caractéristiques ainsi que les défis les plus importants concernant ces concepts. Ces défis de recherche font l'objet de plusieurs efforts de normalisation au sein de différentes organisations de standardisation et de projets de recherches européens et internationaux.

Le chapitre 4 présente deux types d'architectures, que nous proposons pour un environnement de Cloud Networking, basés sur l'approche Inter-Cloud Broker et Fédération pour l'offre des services de type IaaS et NaaS avec une garantie d'un niveau de service décrivant la qualité de service requise par le CSU. Le CSU indique ses exigences de QoS par rapport aux services requis en utilisant une interface utilisateur graphique dédiée. De plus, chaque type d'architecture spécifie différents types de SLA ainsi que différentes interactions nécessaires pour leurs établissements, afin d'assurer le niveau de service demandé par le CSU.

Le chapitre 5 présente la méthodologie d'optimisation lors de la sélection des CSP offrant les meilleures ressources avec une garantie de QoS de bout en bout afin de répondre aux exigences du CSU. Ainsi, nous proposons des algorithmes en tant que solution pour l'optimisation de cette sélection. Ces algorithmes sont relatifs à l'offre du service NaaS, NaaS avec IaaS, IaaS de type VM, et IaaS de type stockage. De plus, nous spécifions les équations nécessaires pour calculer les différents coûts relatifs à cette garantie de QoS. Nous définissons dans ce chapitre l'environnement de simulation pour la validation de cette proposition de garantie de QoS pour deux types d'applications, à savoir la vidéoconférence et les calculs intensifs.

Le chapitre 6 présente notre architecture pour l'établissement autonome des SLA proposés ainsi que la gestion autonome des ressources correspondantes en utilisant des gestionnaires autonomes spécifiques de Cloud. Ainsi, nous décrivons les différents automates finis et algorithmes proposés pour l'établissement autonome de ces SLA et pour la gestion autonome de ces ressources pour garantir un niveau de service de bout en bout dans une architecture de type Broker et Fédération. De plus, nous spécifions les équations permettant la détection et le calcul des violations, des pénalités, et des réputations des différents CSP. Enfin, nous présentons l'environnement de simulation pour la validation de cette proposition de gestion autonome des SLA et des ressources du Cloud grâce à des scénarios de déploiement de deux types d'applications, à savoir la vidéoconférence et les calculs intensifs.

Le chapitre 7 présente une amélioration de nos architectures de Cloud Networking pour offrir un niveau de service intégrant une offre de sécurité grâce à des SLA comportant des paramètres de sécurité. Ainsi, nous présentons une architecture pour la distribution des certificats aux différentes entités de notre environnement de Cloud Networking afin de permettre aux gestionnaires autonomes d'étendre leurs fonctions de gestion aux aspects liés à la sécurité (Auto-protection) tout en étudiant l'impact de la sécurité sur la QoS. Dans ce contexte, nous spécifions l'environnement de simulation pour la validation de cette proposition d'extension du niveau de service à la sécurité tout en étudiant son impact sur la QoS grâce à un scénario de déploiement d'une application de vidéoconférence.

Enfin, le chapitre 8 conclut cette thèse et présente les perspectives des travaux de recherches réalisés.

CHAPITRE 2
CLOUD COMPUTING, CLOUD NETWORKING ET INTER-CLOUD

2.1/ INTRODUCTION

Au cours des dix dernières années, Internet s'est développé très rapidement. Les petites entreprises qui possèdent des ressources limitées ont besoin d'investir à grande échelle avec un budget convenable. Par contre, mettre en place une infrastructure traditionnelle est un processus long et coûteux qui nécessite un investissement initial conséquent, beaucoup de temps, et des personnes qualifiées pour mettre cette infrastructure en place et la gérer au quotidien. De plus, quand un problème se pose pour les services et les applications, ces entreprises ont besoin de beaucoup de temps et un coût élevé pour le régler. Ainsi, les entreprises qui possèdent leurs infrastructures doivent trouver continuellement les solutions adéquates face à ces problèmes. La question qui s'est alors posée : pourquoi mettre en place une infrastructure complète si nous pouvons louer des ressources à la place ? Cette réflexion a favorisé l'apparition du Cloud Computing.

Bien que dans sa forme actuelle, le Cloud Computing (ou simplement le Cloud) est un phénomène relativement récent, nous sommes probablement dans un monde où nous utilisons le Cloud sans s'en rendre compte. Si nous utilisons par exemple un fournisseur de messagerie Web comme «Gmail» ou «Hotmail», des appels vidéo avec «Skype» ou des interfaces vidéo comme «Vimeo» ou «YouTube», et si nous sauvegardons des données sur Internet plutôt que sur un appareil externe, alors nous utilisons le Cloud. Aujourd'hui, le Cloud est présent partout avec un intérêt croissant grâce à une offre de ressources à la demande.

Dans ce chapitre, nous présentons une description générale du Cloud Computing, du Cloud Networking et de l'Inter-Cloud, avec leurs définitions, leurs caractéristiques, leurs modèles de services et de déploiements. De plus, nous décrivons les propositions relatives à ces concepts émanant des organismes de standardisation ainsi que les outils d'implémentation et de simulation et les produits commerciaux utilisés dans ces environnements.

2.2/ CLOUD COMPUTING

2.2.1/ HISTORIQUE

Il n'y a pas de date à laquelle nous puissions dire que le Cloud est né. Le concept du Cloud remonte aux années 1950, quand les ordinateurs centraux à grande échelle sont

devenus disponibles dans les universités et les entreprises, et accessibles via des terminaux clients et/ou des ordinateurs [1]. John McCarthy a estimé dans les années 1960 que «le calcul peut un jour être organisé comme un service d'utilité public» [2]. De plus, les caractéristiques d'utilisation des secteurs publics et privés comme le secteur de l'électricité, le gouvernement et les formes communautaires, explorées dans le livre de Douglas Parkhill en 1966 [2] présentent beaucoup de similitudes avec les caractéristiques modernes du Cloud.

Depuis les années 70, la notion de «service bureau» est inventée pour qualifier une entreprise louant des lignes téléphoniques, répondeurs, services informatiques etc. Généralement, les clients des services bureau n'ont ni la capacité ni l'expertise pour intégrer en interne ces services, c'est pourquoi ils passent par un fournisseur. Les «Application Service Providers (ASP)» ont aussi leur part dans l'historique du Cloud. Une entité de type ASP propose une application fournie comme un service, c'est ce que nous désignons maintenant SaaS pour «Software as a Service» dans la terminologie actuelle du Cloud [3]. Ensuite, le terme Cloud a été déjà utilisé en 1990 pour faire référence aux grands réseaux ATM (Asynchronous Transfer Mode) qui s'appuient sur la notion de circuit virtuel et englobent des logiciels, des matériels et des médias de connexion, tout en supportant plusieurs niveaux de qualité de service.

Au 21e siècle, le terme de l'informatique en nuage «Cloud Computing» a commencé à apparaître. En 2002, Amazon, le leader du e-business inventa le concept de Cloud. En effet, ils investirent dans un groupement de machines immenses, dimensionnées pour absorber les charges importantes des commandes faites sur leur site au moment des fêtes de Noël, mais plutôt inutilisées le reste de l'année. Leur idée a donc été d'ouvrir toutes ces ressources inutilisées aux entreprises, pour qu'elles les louent à la demande. Après, de nombreuses entreprises entrent sur le marché du Cloud telles que IBM, Microsoft ou encore Google. L'année 2007 a vu une augmentation des activités avec ces entreprises (Google, IBM, etc). Ensuite, un certain nombre d'universités s'engageant à grande échelle sur des projets de recherche portant sur le Cloud, et avec le temps, le terme a commencé à gagner en popularité dans la presse [2].

2.2.2/ Définition

La notion de Cloud fait référence à un nuage, tel que nous avons l'habitude de l'utiliser dans des schémas techniques lorsque nous voulons représenter Internet. Un réseau comme Internet est constitué d'une multitude de systèmes fournissant des services. Le Cloud Computing est dans cette lignée : un ensemble de services et de données consommables fournis aux utilisateurs [3].

Pour une majorité d'utilisateurs, le Cloud est un nouveau modèle informatique avec une source d'économies. Il permet de répondre facilement aux besoins de développement et permet de se détacher des problèmes de gestion informatique. En effet, les utilisateurs appelés «Cloud Service Users (CSU)» ne sont pas généralement propriétaires de l'infrastructure informatique, mais ils possèdent la possibilité d'accéder ou d'allouer des services informatique du Cloud. Ces services sont fournis par des fournisseurs de services Cloud appelés «Cloud Service Providers (CSP)».

Dans l'industrie et le secteur académique, il existe de nombreuses définitions pour le Cloud. Vaquero et al. [4] listent 22 définitions du Cloud. Nous avons choisi la définition fournie par le «National Institute of Standards and Technology (NIST)» [5] couvrant tous

les aspects essentiels du Cloud : «Le Cloud Computing est un modèle informatique pratique pour établir un accès facile et à la demande, par le réseau, à un réservoir partagé de ressources informatiques configurables (réseau, serveurs, espaces de stockage, applications et services) qui peuvent être rapidement provisionnées et libérées avec un minimum d'efforts de gestion ou d'interaction avec le fournisseur du service».

2.2.3/ Caractéristiques

Le Cloud possède plusieurs caractéristiques intéressantes qui le rendent attirant pour les CSU et les CSP. Vaquero et al. [4], Buyya et al. [6] et Gong et al. [7] donnent une analyse complète des caractéristiques du Cloud. Nous présentons dans cette section les caractéristiques du Cloud les plus importantes :

— **Facilité d'accès :** les services hébergés dans le Cloud sont généralement basés sur le Web. Par conséquent, ils sont facilement accessibles grâce à une variété d'appareils connectés à Internet. En outre, pour atteindre une performance élevée en termes d'accessibilité, beaucoup de Clouds sont composés de centres de données «Data Center (DC)» situés à plusieurs endroits à travers le monde.
— **Faible investissement initial et tarification avantageuse :** le Cloud utilise un modèle de paiement de type «payez ce que vous utilisez» (voir figure 1). Les CSU sont facturés en se basant sur l'utilisation plutôt que sur un taux forfaitaire, cela permet des économies considérables pour ces derniers. D'autre part, un CSP n'a pas besoin d'un investissement trop important dans les infrastructures pour offrir ces services et commencer à faire des profits. Il loue simplement des ressources du Cloud en fonction des besoins de l'utilisateur.

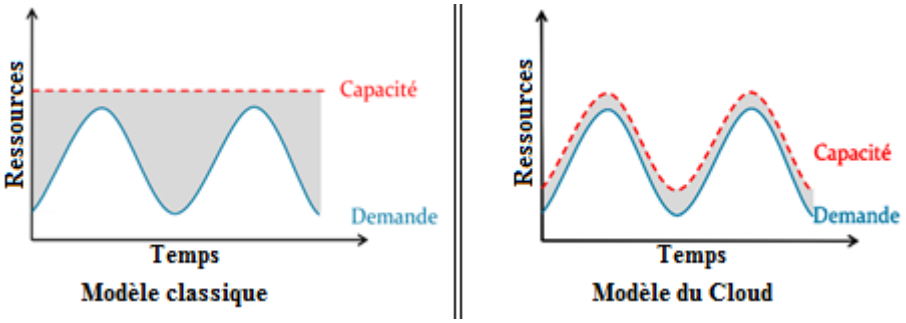

Figure 1 – Différence entre le modèle de paiement classique et celui du Cloud.

— **Groupement des ressources partagées et affectation dynamique et à la demande :** le CSP dispose d'un groupement de ressources informatiques qui peuvent être attribuées dynamiquement aux consommateurs de ressources (CSU). Une telle capacité d'affectation dynamique des ressources permet beaucoup de flexibilité aux CSP pour la gestion de leur propre utilisation des ressources et les coûts d'exploitation [5].
— **Auto-organisation et mise à l'échelle :** les ressources dans un environnement de Cloud peuvent être rapidement allouées et désallouées à la demande. Dans certain cas, elles doivent être faites automatiquement. Pour le CSU, les capacités doivent être disponibles et affectées à n'importe quel moment et avec n'importe quelle quantité [5]. Un CSP peut facilement étendre son service à grande échelle afin de gérer l'augmentation rapide de la demande de services.

— **Réduction des risques d'affaires et des charges de la maintenance :** en externalisant l'infrastructure des services vers les Clouds, le CSU déplace les risques d'affaires (tels que les pannes matérielles, etc.) et les charges de la maintenance vers les CSP, qui ont souvent une meilleure expertise et sont mieux équipés pour gérer ces risques.
— **Service mesuré :** le Cloud contrôle automatiquement et optimise l'utilisation des ressources grâce à une capacité de mesure avec un certain niveau d'abstraction approprié pour les types de service (par exemple, le stockage, le traitement, la bande passante, les comptes d'utilisateurs actifs, etc.). L'utilisation des ressources peut être quantifiée au moyen de mesures appropriées (les heures pour le CPU, la bande passante, etc.), surveillée, contrôlée et rapportée, en assurant la transparence pour le CSP et le CSU [5].
— **Ressources virtualisées :** la virtualisation est une technologie qui fait une abstraction des matériels physiques et fournit des ressources virtualisées pour les applications de haut niveau. Un serveur virtualisé est communément appelé une machine virtuelle (VM : Virtual Machine) et nous pouvons mettre plusieurs VM sur le même serveur physique. La virtualisation constitue la base du Cloud Computing, car elle offre la possibilité de mettre en commun les ressources informatiques de groupes de serveurs et d'affecter dynamiquement des ressources virtuelles à la demande aux applications. Nous pouvons virtualiser de nombreuses ressources telles que les ressources informatiques, les logiciels, les matériels, les systèmes d'exploitation et les espaces de stockage. De plus, nous pouvons gérer ces ressources virtualisées d'une manière isolée de l'infrastructure physique.

2.2.4/ Modèles de services

Le but principal du Cloud est d'offrir des services à des utilisateurs suivant différents modèles. Nous présentons dans cette section les modèles de services principaux du Cloud.

2.2.4.1/ Software as a service (SaaS)

Le modèle de services «logiciel en tant que service» (SaaS) permet de fournir des applications à la demande sur Internet. Ces services du Cloud sont fournis à des millions d'utilisateurs et sont accessibles à partir de différents dispositifs clients, par exemple un navigateur Web. Ceci permet de réduire un certain coût relatif aux logiciels et aux serveurs. Le CSU n'a pas à se soucier d'effectuer des mises à jour, et il ne gère ni contrôle l'infrastructure Cloud sous-jacente, y compris le réseau, les serveurs, les systèmes d'exploitation, le stockage, à l'exception possible des paramètres relatifs à la configuration de l'application. Gmail est un exemple de service de type SaaS qui offre au CSU un service de courrier électronique. D'autres exemples d'outils et de produits commerciaux qui fournissent des services du modèle SaaS seront présentés dans la section 2.2.8.

2.2.4.2/ Platform as a Service (PaaS)

Le modèle de services «plateforme en tant que service» (PaaS) est situé juste au-dessous du précédent (SaaS), puisqu'il permet de fournir aux développeurs (CSU) des

ressources pour les environnements d'exécution, les cadres de développement de logiciels et le soutien du système d'exploitation. Les développeurs (y compris les CSP de service SaaS) ont plusieurs avantages de développer leur application dans les Clouds en tant qu'environnement de programmation. Cet environnement permettra une mise à l'échelle automatique et l'équilibrage de charge, ainsi que l'intégration avec d'autres services (services d'authentification, services de courrier électronique, interface utilisateur, etc.). Le modèle de service PaaS offre aux développeurs un service qui fournit une gestion du cycle de vie complet du développement logiciel, depuis la planification jusqu'à la maintenance [8]. L'infrastructure informatique sous-jacente est abstraite du point de vue des développeurs, c.à.d. le développeur peut exploiter ce modèle de service pour construire ses applications sans avoir la moindre idée de ce qui se passe dans l'infrastructure sous-jacente à ce service. Le coût et la complexité du développement et de déploiement d'applications peuvent être largement réduits lorsque les développeurs utilisent ce service de Cloud. Heroku [9] est un service de Cloud de type PaaS créé en 2007 : il s'agit de l'un des tout premiers services Cloud de ce type. D'autres exemples d'outils et produits commerciaux qui fournissent des services du modèle PaaS seront présentés dans la section 2.2.8.

2.2.4.3/ Infrastructure as a Service (IaaS)

Le modèle de services «Infrastructure en tant que service» (IaaS) est situé juste audessous du précédent (PaaS), puisqu'il permet de créer un pool de ressources virtuelles en divisant les ressources physiques à l'aide des technologies de virtualisation. Ces ressources virtuelles d'infrastructure seront fournies à la demande, le plus souvent en termes de machines virtuelles, d'espaces de stockage et de ressources réseau. Les machines virtuelles sont utilisées pour la fourniture de ressources CPU pour les CSU (y compris les CSP de service PaaS ou SaaS). Le CSP d'IaaS contrôle l'initiation des VM selon la demande du CSU en lui donnant le contrôle total sur ses VM et il contrôle l'élimination des VM si le CSU n'en a plus besoin. De plus, les espaces de stockage de données permettent aux utilisateurs de stocker leurs données dans des disques distants et d'y accéder à tout moment à partir de n'importe quel endroit. Ce service est connu sous le nom de stockage de données en tant que service (DaaS : Data storage as a Service). VMware [10] offre un service de Cloud de type IaaS. D'autres exemples d'outils et produits commerciaux qui fournissent des services du modèle IaaS seront présentés dans la section 2.2.8.

2.2.4.4/ Network as a Service (NaaS)

Le modèle de services «réseau en tant que service» (NaaS) permet d'offrir l'ensemble des ressources réseaux en tant que service. Il existe deux types de service réseau dans le cadre de ce modèle. Le premier type est formé du réseau qui relie le CSU avec le CSP ou bien les CSP entre eux. Le deuxième type de réseau est celui du centre de données du CSP qui relie les différentes VM et les espaces de stockage. De plus, le service de type NaaS permet aux CSU de personnaliser les services qu'ils reçoivent à partir du réseau, c.à.d. de contrôler l'exploitation du réseau facilement et de manière efficace, tout en permettant au CSP de décider comment les ressources sont allouées et partagés entre les CSU [11]. Le service NaaS est offert dans un environnement de Cloud Networking (voir Section 2.3).

Les CSP peuvent encore optimiser leurs coûts en utilisant les services d'autres types de Cloud. D'une part, certains CSP de SaaS font appel à d'autres CSP de PaaS et IaaS. Ainsi, ils n'ont besoin de maintenir que l'application, et ils seront alors des CSU pour des services PaaS et IaaS. D'autre part, certains CSP offrent du PaaS et par la suite ils font appel à d'autres CSP d'IaaS, et ils seront alors des CSU pour des services IaaS.

2.2.5/ Modèles de déploiement

Il existe différents modèles de déploiement du Cloud, chacun avec ses avantages et ses inconvénients.

2.2.5.1/ Cloud public

Dans un Cloud public les CSP offrent leurs ressources comme services au grand public. Ce modèle peut être détenu, géré et exploité par une entreprise ou une organisation académique ou gouvernementale, ou une combinaison entre eux. Le Cloud public offre plusieurs avantages aux utilisateurs, y compris l'absence de coûts d'investissement élevés sur les infrastructures et le déplacement des risques vers les fournisseurs d'infrastructure. Mais, ces utilisateurs n'ont pas un contrôle fin sur les données, le réseau et les paramètres de sécurité, ce qui entrave l'efficacité de ce modèle de déploiement.

2.2.5.2/ Cloud privé

Le Cloud privé est conçu pour une utilisation exclusive par une seule organisation. Un Cloud privé peut être construit et géré par l'organisation, une tierce partie, ou une combinaison des deux. Il offre le plus haut degré de contrôle sur les performances, la fiabilité et la sécurité. Cependant, il est souvent critiqué car il est similaire aux serveurs propriétaires traditionnels qui ne fournissent pas les avantages du Cloud comme l'absence de coûts d'investissement élevés.

2.2.5.3/ Cloud communauté

C'est un Cloud qui partage des infrastructures entre plusieurs organismes d'une communauté spécifique avec des préoccupations communes (la mission, les exigences de sécurité, la politique, etc.). Ces infrastructures sont gérées par un ou plusieurs organismes de la communauté, une tierce partie, ou une combinaison de ces entités. Ce modèle de déploiement offre les avantages d'un Cloud public comme la structure de facturation de type «payez ce que vous utilisez», mais aussi les avantages d'un Cloud privé en termes de confidentialité et de sécurité d'une façon générale. Les coûts sont répartis sur moins d'utilisateurs qu'un Cloud public (mais plus qu'un Cloud privé), de sorte qu'une partie des économies potentielles de Cloud sont réalisées.

2.2.5.4/ Cloud hybride

Un Cloud hybride est une combinaison de déploiement des modèles de Cloud (public, privé et communauté) qui tente de remédier aux limitations de chaque approche. Dans un Cloud hybride, une partie du service de l'infrastructure s'exécute dans des Clouds privés tandis que la partie restante est dans des Clouds publics. Un Cloud hybride offre plus de flexibilité qu'un Cloud public ou privé, puisqu'il fournit un meilleur contrôle et une meilleure sécurité pour les données d'application des utilisateurs par rapport aux Clouds publics et une tarification avantageuse par rapport aux Clouds privés. Cependant, la conception d'un Cloud hybride nécessite une étude détaillée afin de déterminer la meilleure répartition entre les composantes de Cloud public et privé.

Pour la plupart des organisations, le choix du modèle de Cloud dépend du cas d'utilisation et des exigences du CSU (sécurité, QoS, etc.).

2.2.6/ Standardisation

Le Cloud implique un large éventail d'éléments techniques et commerciaux différents, et les infrastructures du Cloud ont commencé par des solutions propriétaires comme Amazon, Microsoft ou Google, etc. D'où le besoin de la standardisation pour définir la terminologie et les techniques à utiliser et aussi pour assurer l'émergence d'une infrastructure Cloud standard, interopérable et adoptée par la majorité des acteurs du marché.

De nombreuses organisations et des groupes informels travaillant sur les technologies du Cloud se sont spécialisés dans le traitement des problèmes de standardisation en ce qui concerne l'environnement de Cloud. Ces organisations de standardisation aident à maintenir les normes et les meilleures pratiques pour veiller à ce que les différents CSP et les équipements soient capables de coexister ensemble d'une manière interopérable. Nous présentons dans cette section des différents acteurs dans le domaine de standardisation du Cloud.

La figure 2 [12] montre une classification générale des différentes organisations de standardisations selon leurs rôles dans un environnement de Cloud. Par exemple, l'organisation «Distributed Management Task Force (DMTF)» [13] contient différents groupes de travail sur le Cloud. Nous pouvons citer à ce titre le groupe de travail «Open Virtualization Format (OVF)» qui travaille sur un format standard de la machine virtuelle, mais aussi «Open Cloud Standards Incubator (OCSI)» qui a été établi en Avril 2009 pour étudier les standards qui permettront l'interopérabilité entre les systèmes de Clouds. De plus, «Cloud Management Working Group (CMWG)» a été créé en Juin 2010 pour travailler sur les politiques, le SLA, la QoS, l'approvisionnement et la surveillance dans le Cloud. Le groupe de travail «Virtualization Management Initiative (VMAN)» fournit des standards d'interopérabilité et de portabilité, et enfin le groupe «Cloud Infrastructure Management Interface (CIMI)» vise la gestion des ressources au sein du domaine IaaS.

D'autre part, «IEEE Cloud Computing» [14] est une organisation de standardisation qui contient deux groupes de travail sur le Cloud, «Cloud Computing Standards Study Group (CCSSG)» et «Inter-Cloud Working Group (ICWG)». Cette organisation a annoncé en Avril 2011 le lancement de deux nouveaux projets d'élaboration de normes : P2301 [15], Guide pour la portabilité Cloud et profils d'interopérabilité (CPIP) et P2302 [16], standard pour l'interopérabilité et la Fédération Inter-Cloud (SIIF). De plus, «International Telecommunication Union - Telecommunication Standardization Sector (ITU-T)» [17] est une autre

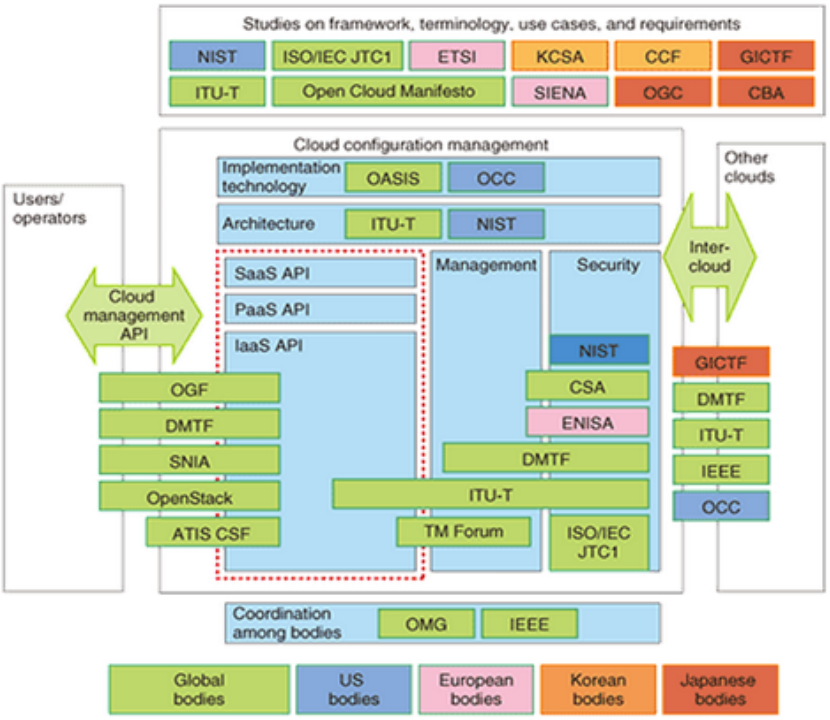

FIGURE 2 – Exemple d'organisations de standardisation du Cloud [12].

organisation de standardisation qui a terminé son étude préliminaire sur l'écosystème de standardisation de Cloud (Architecture de référence) et a sorti son rapport technique composé de 7 parties. Les travaux de standardisation de Cloud se déroulent sous la direction du Groupe d'étude 13 (Future Networks).

Un autre acteur dans la standardisation des environnements Cloud est le «National Institute of Standards and Technology (NIST)» [5]. Il s'agit d'une organisation qui comporte plusieurs groupes de travail ayant pour objectif de fournir une stratégie basée sur les standards pour guider les efforts de mise en œuvre des clouds tout en proposant une architecture de référence pour l'environnement Cloud (Cloud Computing Target Business Use Cases Working Group, Cloud Computing Reference Architecture and Taxonomy Working Group, Cloud Computing Standards Roadmap Working Group, Cloud Computing SAJACC Working Group, et Cloud Computing Security Working Group). Nous retrouvons aussi une organisation japonaise, «Global Inter-Cloud Technology Forum (GICTF)» [18], qui étudie les interfaces standards d'Inter-Cloud afin d'améliorer la fiabilité des Clouds. De son côté, «Internet Engineering Task Force (IETF)» [19] a publié des rapports (de type draft) sur le Cloud et le réseau du DC. «European Telecommunications Standards Institute (ETSI)» [20] a fait une description des tests d'interopérabilité, une étude sur le SLA, et une analyse initiale des exigences de standardisation pour les services Cloud. «Open Grid Forum (OGF)» [21] a créé un groupe de travail appelé «Open Cloud Computing Interface (OCCI) WG» en Avril 2009 qui a défini une spécification «Application Programming Interface (API) OCCI», pour la gestion du cycle de vie des VM. «Open Cloud Consortium (OCC)» [22] est une organisation créée en Janvier 2009 qui a pour objectif de réaliser l'interopérabilité entre les systèmes de Cloud. D'autres organisations comme «Object Management Group (OMG)» [23], «Cloud Security Alliance (CSA)» [24], «Cloud

Computing Interoperability Forum (CCIF)» [25], et «ISO/IEC JTC1» [26] contribuent aux efforts de standardisation dans le domaine du Cloud.

Les normes qui régissent le Cloud sont tout simplement en train de naître et cela peut prendre du temps pour se développer. De plus, les différentes organisations que nous venons de décrire ne cherchent pas à collaborer ensemble pour remédier aux problèmes posés par la normalisation. Ainsi, l'absence de standards ou encore la lenteur dans le processus de normalisation risquent de retarder l'entrée de certaines entreprises dans un environnement de type Cloud. Une coordination entre les différents acteurs du domaine permettrait aux différents CSP de proposer des offres standardisées avec une garantie d'interopérabilité des services émanant de différents fournisseurs de Cloud.

2.2.7/ Outils d'implémentation et de simulation

Nom	Modèle de service	Modèle de déploiement	Open source	But
OpenNebula [27]	IaaS	Cloud privé	Oui	Construction et gestion de façon centralisée d'une infrastructure IaaS virtuelle hétérogène.
Nimbus [28]	IaaS	Cloud public	Oui	Installation sur une grappe de serveurs et fourniture d'un service IaaS à son CSU.
Eucalyptus [29]	IaaS	Cloud public, privé ou hybride	Oui	Installation, déploiement et gestion de grappe de serveurs comme un seul Cloud.
OpenStack [30]	PaaS, IaaS, NaaS	Cloud public, privé ou hybride	Oui	Déploiement des infrastructures de Cloud et contrôle des différentes ressources de machines virtuelles, de stockage ou encore de réseau.
CloudStack [31]	IaaS, NaaS	Cloud public, privé ou hybride	Oui	Déploiement et gestion avec une interface Web de grands réseaux de machines virtuelles.
Snooze [32]	IaaS	Cloud public ou privé	Oui	Construction des infrastructures de ressources de calcul virtualisées, et gestion des machines virtuelles.

Table 1 – Exemples de plateformes d'implémentation du Cloud.

Dans le monde du Cloud, la plupart des outils d'implémentation et de simulation sont axés sur des expériences et une modélisation simplifiée du Cloud. Ces outils ouvrent la possibilité d'évaluer les hypothèses dans un environnement contrôlé où l'on peut facilement reproduire des résultats. De plus, ils permettent de tester les services Cloud dans un environnement répétitif et contrôlable. Ils peuvent permettre d'adapter le système avant de le déployer sur un vrai Cloud et de l'expérimenter avec différentes charges de travail et scénarios de performances.

Nous présentons dans la table 1 des exemples de plateformes d'implémentation et dans la table 2 des exemples de logiciels de simulation. Nous remarquons que la plupart des plateformes d'implémentation peuvent être installées sur des ressources physiques pour fournir principalement un service de type IaaS. En effet, quelques plateformes seulement permettent d'offrir des services de type PaaS ou encore NaaS. Cependant, les logiciels de simulation supportent la modélisation et la simulation de l'environnement de Cloud pour n'importe quel type de service sans besoin de ressources réelles.

Nom	Modèle de service	Modèle de déploiement	Open source	But
CloudSim [33]	SaaS, PaaS, IaaS, NaaS	Cloud public, privé ou hybride	Oui	Modélisation, simulation et expérimentation transparente du Cloud et de ses services.
Real-CloudSim [34]	IaaS, NaaS	Cloud public, privé ou hybride	Oui	Allocation de machines virtuelles basée sur CloudSim et fourniture d'une interface graphique pour lire les topologies de réseau basées sur le format BRITE [33].
MR-CloudSim [35]	IaaS	Cloud public, privé ou hybride	Oui	Outil de simulation basé sur CloudSim qui prend en charge le modèle MapReduce qui est le modèle de programmation le plus utilisé pour le traitement de grande quantité de données (Big data).
GroudSim [36]	SaaS, IaaS	Cloud public, privé ou hybride	Oui	Simulation des applications scientifiques sur un environnement de Cloud.
DCSim [37]	IaaS	Cloud public, privé ou hybride	Oui	Simulation d'un DC hébergeant un Cloud IaaS.
GreenCloud [38]	IaaS, NaaS	Cloud public, privé ou hybride	Oui	Simulation des centres de données du Cloud pour évaluer leurs coûts d'exploitation en termes d'énergie. Il s'agit une extension du simulateur réseau NS2 [39].
iCanCloud [40]	IaaS	Cloud public, privé ou hybride	Oui	Modélisation et simulation des systèmes de Cloud. Il peut fournir un hyperviseur global, flexible et entièrement personnalisable, et simuler des VM.

TABLE 2 – Exemples de logiciels de simulation du Cloud.

2.2.8/ PRODUITS COMMERCIAUX

De nombreuses entreprises sont spécialisées dans les produits commerciaux relatifs à l'environnement de Cloud. Nous présentons dans la table 3 quelques exemples de produits commerciaux de Cloud. De plus, il existe d'autres produits commerciaux comme SAP Business ByDesign [41], Zoho Office [42], GoGrid [43], Flexiscale [44] offerts dans un environnement de Cloud.

Généralement, les CSU utilisent des ressources du Cloud à la demande en fonction de leurs besoins sans la mise en place d'une infrastructure complète. Pour cela, comme nous remarquons dans la table 3, la plupart des produits commerciaux qui offrent ces services se basent sur un modèle de déploiement de type Cloud public.

Nom	Modèle de service	Modèle de déploiement	But
Amazon Web Services (AWS) [45]	IaaS, NaaS	Cloud public	Fourniture d'un ensemble de ressources de calcul, de stockage et de réseau, et d'autres fonctionnalités qui permettent aux CSU de déployer des applications et des services selon sa demande.
Google Cloud [46]]	PaaS, IaaS	Cloud public	Fourniture d'un environnement de développement pour les applications web traditionnelles dans les centres de données gérés par Google. Chaque environnement propose des protocoles standards et des technologies courantes en matière de développement d'applications Web.
Microsoft Windows Azure platform [47]	PaaS, IaaS	Cloud public	Fourniture d'un ensemble d'APIs permettant d'utiliser et d'accéder à cette plateforme et aux services associés.
SalesForce [48]	SaaS, PaaS	Cloud public	Distribution de logiciels de gestion basés sur Internet, hébergement des applications d'entreprises, et gestion de la relation client.
Rackspace [49]	PaaS, IaaS	Cloud public	Fourniture d'un ensemble de produits et services de Cloud contenant l'hébergement d'applications web, load balancers, bases de données, backup, et surveillance.

TABLE 3 – Exemples de produits commerciaux du Cloud.

2.3/ Cloud Networking

2.3.1/ Définition

Le Cloud Networking offre un nouveau modèle de services de Cloud Computing dans lequel nous utilisons l'ensemble des ressources de deux types de réseaux en tant que service [50]. Le premier service réseau (NaaS (DC)) offre les ressources réseau du centre de données (DC) du Cloud qui relient les différentes machines virtuelles (VM) et les espaces de stockage. Ce type de réseau est appelé Réseau Intra-Cloud. Le deuxième service réseau (NaaS (BoD)) offre une bande passante à la demande (BoD) grâce aux ressources réseau qui relient le CSU et les infrastructures des CSP (Réseau de transport de base (WAN/MAN)) ou encore les ressources réseau qui relient les infrastructures des CSP entre elles (Réseau Inter-Cloud). Ces ressources réseau peuvent être détenues par un même CSP ou par différents CSP.

2.3.2/ Caractéristiques

L'architecture de Cloud Networking s'appuie sur deux concepts principaux : d'une part, l'intégration de réseaux virtuels à travers les infrastructures d'opérateurs de réseau, et d'autre part, la connexion des ressources de calcul et de stockage à travers les équipements réseau de centres de données (DC). Ainsi, les CSU et les CSP ou bien les CSP entre eux, sont reliés par un réseau étendu, où les performances relatives au transfert de données sont difficiles à garantir. Une des solutions est d'utiliser une bande passante suffisante pour transporter les informations dans ce réseau. Cette bande passante est fournie à la demande (BoD), dans le cadre du Cloud Networking, par un fournisseur d'accès Internet ou n'importe quel acteur qui a la capacité d'offrir la connectivité et la réservation. La bande passante à la demande (BoD) offre la possibilité de réaliser d'une façon dynamique des modifications (augmenter, diminuer) concernant la fourniture de la bande passante sur des liens particuliers grâce à des interfaces standardisées. La BoD fournit une solution économique pour l'offre de QoS dans le Cloud [50]. De plus, le concept de Cloud Networking permet de fournir des services de réseau basés sur le Cloud comme le routage, les tunnels, etc., [51].

Par contre, il existe plusieurs exigences et défis de recherches à relever dans le cadre du Cloud Networking [50]. Ces exigences et défis concernent les spécifications de QoS (la bande passante, les volumes de trafic, la disponibilité, etc.), les performances (l'accélération des applications et l'optimisation des services à la demande), le support du SLA, la fourniture d'une redondance et d'un équilibrage de charge du trafic sur les liens, la sécurité (y compris les fonctions de sécurité à la demande au sein du réseau de transport ainsi que protéger et contrôler le trafic client (pare-feu, détection et prévention des intrusions, etc.)), et enfin la spécification des agents qui peuvent traiter les changements nécessaires d'une manière autonome. De plus, la topologie de communication doit être conçue de manière à supporter la migration rapide des VM et l'infrastructure du réseau doit être capable de s'adapter à un grand nombre de serveurs afin de permettre la mise à l'échelle progressive tout en étant tolérante à divers types de pannes.

2.3.3/ Outils d'implémentation et de simulation

Plusieurs outils d'implémentation et de simulation possèdent des fonctionnalités concernant le Cloud Networking. Des exemples de ces outils sont présentés dans la table 4.

Nom	Modèle de service	Modèle de déploiement	Open source	But
OpenStack :NaaS [51]	NaaS	Cloud public	Oui	Fourniture des ressources réseau nécessaires pour l'interconnexion des zones OpenStack [28] de calcul/stockage et les ressources réseau nécessaires pour soutenir de nouveaux services.
NetworkCloudSim [52]	SaaS, IaaS, NaaS	Cloud public, privé ou hybride	Oui	Modélisation des réseaux du DC du Cloud, des flux de travaux (workflow) et des applications génériques en se basant sur CloudSim.
CloudNaaS [53]	SaaS, NaaS	Cloud public	Oui	Une plateforme de Cloud Networking proposée par IBM pour les applications d'entreprises.
Python Open Cloud Networking Interface (pyOCNI) [54]	NaaS	Cloud privé	Oui	Une extension d'OCCI définie par l'Institut Télécom qui permet le contrôle externe des services réseau.
OpenNaaS [55]	NaaS	Cloud privé	Oui	Fourniture d'un ensemble d'outils souples pour le déploiement de services orientés NaaS. Chaque domaine du réseau sera en mesure d'utiliser sa propre instance d'OpenNaaS pour obtenir des ressources de l'infrastructure de réseau : routeurs, commutateurs, liens ou systèmes de provisionnement.

TABLE 4 – Exemples d'outils d'implémentation et de simulation du Cloud Networking.

2.4/ Inter-Cloud

2.4.1/ Définition

D'une manière générale, si toutes les ressources d'un CSP sont allouées, il ne peut pas satisfaire de nouvelles demandes des CSU. De plus, il est impossible pour un CSP d'avoir des centres de données (DC) dans tous les pays. Le Cloud devrait être conçu comme un environnement multifournisseur, capable d'offrir la possibilité de migrer un service d'un CSP à un autre et de localiser la meilleure ressource, non seulement en termes de capacité de calcul ou de stockage, mais aussi de connectivité, de bande passante et de délai d'acheminement. Dans ce contexte, un environnement de Clouds distribués (Inter-Cloud) offre une solution pour répondre aux besoins des CSU répartis géographiquement [56].

Ainsi, l'Inter-Cloud est une interconnexion globale de Clouds que nous pouvons désigner par l'appellation Cloud de Clouds. L'Inter-Cloud assure une affectation à la demande des ressources d'un Cloud vers les autres, y compris les capacités de calcul, de stockage et les ressources réseaux, ainsi que le transfert de la charge de travail, grâce à l'interfonctionnement des systèmes Cloud [50].

2.4.2/ Caractéristiques et Modèles d'Inter-Cloud

L'Inter-Cloud peut être mis en œuvre par différents modèles que nous décrivons dans les sections suivantes.

2.4.2.1/ Inter-Cloud de type Peering

Ce modèle d'Inter-Cloud (voir figure 3) est caractérisé par l'interconnexion directe entre deux CSP en utilisant des interfaces déjà établies. Les deux Clouds forment logiquement un même Cloud ; l'un d'entre eux peut externaliser dynamiquement les ressources à l'autre en réponse aux variations de la demande.

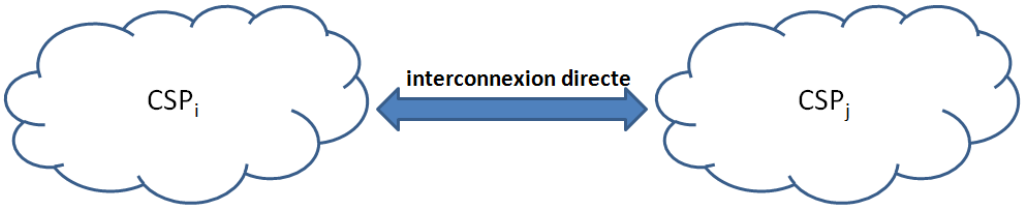

Figure 3 – Inter-Cloud de type Peering.

2.4.2.2/ Inter-Cloud de type Broker

Ce modèle d'Inter-Cloud (voir figure 4) est caractérisé par l'interconnexion indirecte entre deux ou plusieurs CSP grâce à une entité appelée Inter-Cloud Service Broker (ISB). Une entité de type ISB offre des fonctions de service d'interfonctionnement entre les CSP

interconnectés et fournit également des fonctions de service de courtage pour un ou plusieurs CSP interconnectés ainsi que pour le CSU. De plus, le Cloud Broker (responsable du service de courtage) est émergé comme une couche intermédiaire entre les CSP et les CSU pour aider les CSU à choisir les meilleurs services dans un environnement de type Cloud. D'autre part, le Cloud Broker peut agir en tant que négociateur pour les CSU avec de nombreux environnements de Cloud. Ainsi, il peut rechercher et réserver les ressources disponibles dans les autres CSP, en fonction de différents niveaux de service, afin d'éviter les violations de SLA (Service Level Agreement).

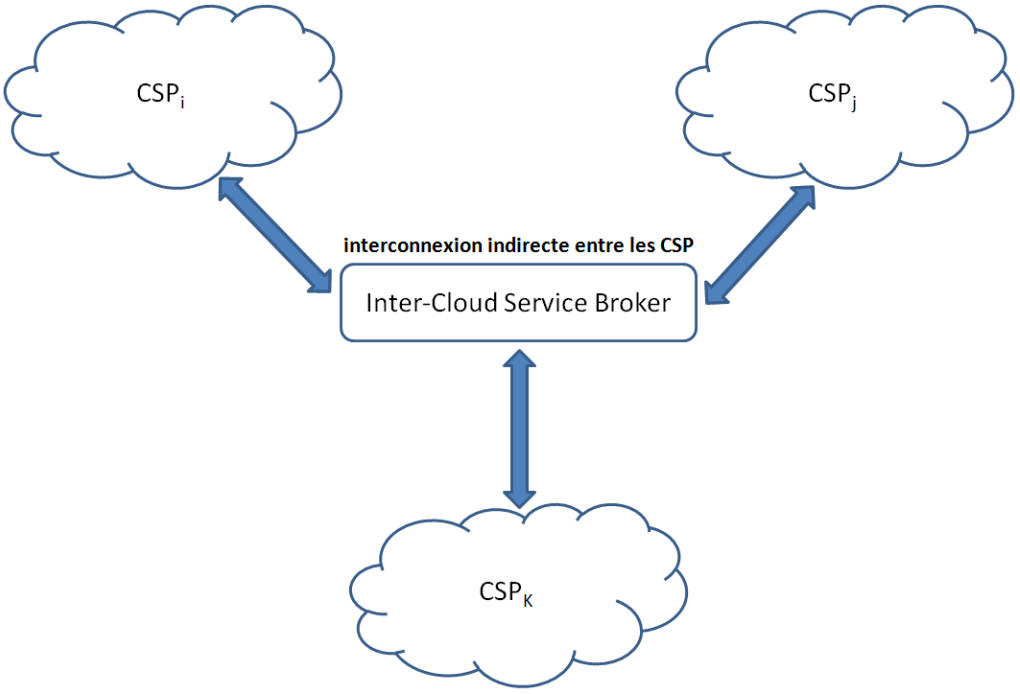

FIGURE 4 – Inter-Cloud de type Broker.

L'Inter-Cloud Service Broker peut fournir trois types de services :

- **l'intermédiation de services :** avec ce type de service, l'ISB permet une amélioration du service en y ajoutant d'autres caractéristiques telles que la gestion de l'accès, la gestion des identités, les rapports de performance, des mesures de sécurité, etc.
- **l'agrégation de services :** en offrant un service de type agrégation, le Cloud Broker propose une combinaison et intégration de multiples services dans un ou plusieurs nouveaux services. Ainsi, il permet l'intégration des données et assure l'acheminement sécurisé des données entre les CSU et les CSP.
- **l'arbitrage de services :** avec ce type de service, le Cloud Broker a la possibilité de choisir les services à partir de multiples CSP. Ainsi, il permet des choix flexibles et opportunistes. Par exemple le Cloud Broker peut utiliser un service de crédit ou encore un score pour mesurer et sélectionner la source avec le meilleur score.

2.4.2.3/ INTER-CLOUD DE TYPE FÉDÉRATION

Dans ce modèle d'Inter-Cloud (voir figure 5), les Clouds forment logiquement un même Cloud en intégrant leurs ressources. Ils sont capables d'interagir de façon directe et transparente entre eux. Ce modèle permet à un CSP d'externaliser dynamiquement les ressources à d'autres CSP en réponse aux variations de la demande.

La fédération d'Inter-Cloud offre des avantages considérables pour les CSP [57]. Ainsi, ce modèle d'Inter-Cloud permet aux CSP de percevoir une rémunération à partir des ressources informatiques qui seraient autrement au repos ou sous-utilisés. De plus, ce modèle permet aux CSP d'absorber des augmentations brusques des demandes en recherchant et en réservant les ressources disponibles dans les autres CSP, en fonction de différents niveaux de service, afin d'éviter les violations de SLA.

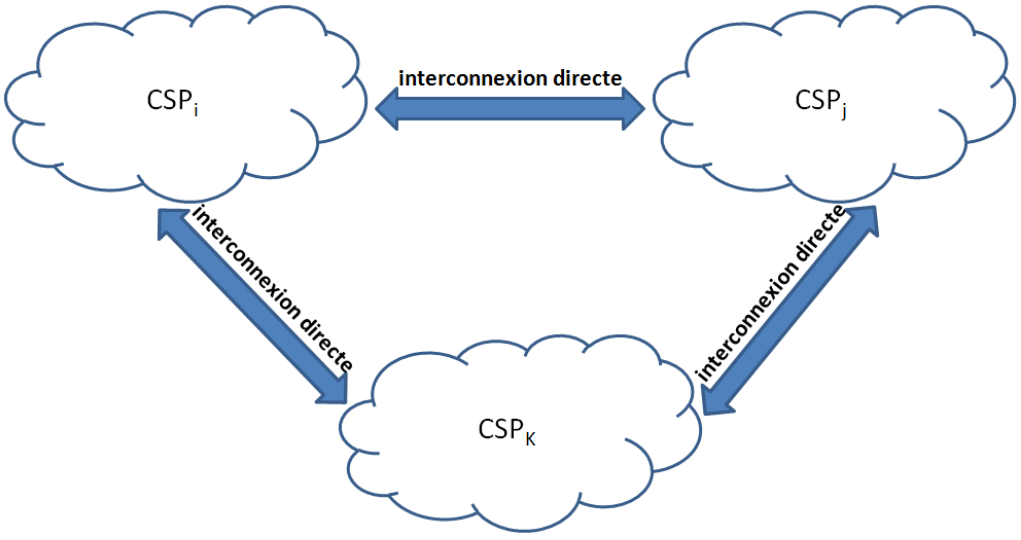

FIGURE 5 – Inter-Cloud de type Fédération.

La communication des différents Clouds est établie à travers un réseau qui peut être fourni en tant que service Cloud de type NaaS (BoD) et qui permet une solution économique pour l'interopérabilité des Cloud (migration de VM, transferts de données importants, etc.).

L'Inter-Cloud est l'un des domaines de recherche les plus récents dans les travaux portant sur le Cloud Computing. L'Inter-Cloud peut apporter de nouveaux créneaux commerciaux entre les CSP si ces Clouds peuvent inter-opérer les uns avec les autres pour le partage des ressources [58]. De plus, les centres de données (DC) répartis géographiquement dans l'Inter-Cloud offrent de meilleures performances de bout en bout entre les CSU et les CSP et améliorent la fiabilité lorsqu'une panne se produit mais aussi la QoS d'une façon générale.

Pour réaliser un modèle d'Inter-Cloud, il y a un certain nombre de défis de recherche ouverts, tels que : la spécification de méthodes standardisées pour les négociations entre les CSP afin d'établir un SLA, l'interopérabilité des formats de données et des interfaces (API) pour faciliter la communication entre différents Clouds, et la conception d'une couche de middleware pour la coordination des ressources [59].

2.4.3/ Outils d'implémentation et de simulation

Des exemples d'outils d'implémentation et de simulation concernant l'Inter-Cloud on été développés ou encore en cours de développement. Nous présentons ces outils dans la table 5. Nous remarquons que l'Inter-Cloud de type Broker est le plus utilisé dans un environnement d'Inter-Cloud.

Nom	Modèle de service	Modèle de déploiement	Open source	But
IEEE InterCloud Testbed [60]	SaaS, PaaS, IaaS	Cloud public ou privé	Oui	Mise en œuvre d'un outil pour les technologies d'Inter-Cloud.
mOSAIC [61]	SaaS, PaaS, IaaS	Cloud public ou privé	Oui	Un Cloud Broker qui fournit une plateforme multi-agents qui permet aux applications de négocier les services Cloud demandés par leurs CSU et chercher les services correspondants.
CompatibleOne [62]	SaaS, PaaS, IaaS	Cloud public, privé ou hybride	Oui	Un Cloud Broker qui fournit un modèle et une plateforme pour la description, l'intégration et l'agrégation des différentes ressources distribuées provisionnées par des CSP hétérogènes. La plateforme exploite pleinement l'Open Cloud Computing Interface (OCCI), et elle est alignée avec l'architecture de référence de NIST.

Table 5 – Exemples d'outils d'implémentation et de simulation de l'Inter-Cloud.

2.4.4/ Produits commerciaux

De nombreuses entreprises ont formé des partenariats avec des CSP pour offrir un service d'Inter-Cloud. Des exemples de produits commerciaux sont présentés dans la table 6. Nous remarquons qu'il faut avoir une interopérabilité entre les différentes entités dans un environnement d'Inter-Cloud pour que les CSU puissent utiliser les services offerts par les CSP.

Nom	Modèle de service	Modèle de déploiement	But
SpotCloud [63]	SaaS, PaaS, IaaS	Cloud public	Un Cloud broker qui permet, d'une part aux CSU de s'inscrire côté achat pour consulter le prix des services demandés et d'autre part aux CSP de s'inscrire côté vente pour fournir les ressources nécessaires. Par contre, une limitation importante de ce produit est que les CSP qui utilisent d'autres technologies que celles de SpotCloud ne peuvent pas s'inscrire.
Appirio [64]	SaaS, PaaS, IaaS	Cloud public	Un Cloud broker des CSP public comme Google, Amazon, Salesforce...
Cisco Intercloud Fabric [65]	SaaS, PaaS, IaaS, NaaS	Cloud hybride	Une société qui permet une connectivité à un large écosystème de CSP pour soutenir des déploiements de Clouds hybrides pour les entreprises.
Intercloud Systems [66]	SaaS, PaaS, IaaS	Cloud hybride	Une société qui offre des services d'Inter-Cloud et des services de maintenance.

TABLE 6 – Exemples de produits commerciaux de l'Inter-Cloud.

2.5/ CONCLUSION

Aujourd'hui, avec le Cloud, il est possible de déployer facilement une application à travers le monde. Cela permet aux entreprises de ne pas être obligées d'être implantées physiquement dans un pays pour aller à la rencontre de leurs clients locaux et de leur proposer leurs services. Dans cet environnement, les CSU peuvent accomplir différentes tâches indépendamment de leurs situations géographiques mais aussi de la complexité de la tâche à réaliser en utilisant des équipements d'extrémité (ordinateur, smartphone...) peu puissants.

Dans ce chapitre, nous avons réalisé une présentation générale du Cloud Computing, son histoire, ses caractéristiques, ses modèles de services et de déploiement, les travaux de standardisation, ainsi que les outils d'implémentation et de simulation et les produits commerciaux relatifs aux services du Cloud. De plus, nous avons présenté une évolution du Cloud Computing pour offrir des ressources réseaux à la demande à savoir le Cloud Networking avec une étude de ses caractéristiques, et les outils de simulation et d'implémentation relatifs à ce nouveau concept. Enfin, nous avons étudié l'approche d'Inter-Cloud en mettant l'accent sur ses caractéristiques et ses trois modèles de déploiement.

Dans ces environnements de Cloud, les défis demeurent en ce qui concerne la qualité de service, la sécurité et la confiance, la surveillance et la gestion autonome, la localisation des données, la récupération en cas de catastrophe, la facturation, etc. Ainsi, dans le chapitre suivant, nous présenterons un état de l'art sur la qualité de service, la sécurité, et la gestion autonome dans un environnement de Cloud Computing.

CHAPITRE 3
NIVEAU DE SERVICE ET GESTION AUTONOME DANS LE CLOUD

3.1/ INTRODUCTION

Un utilisateur du service d'un Cloud (CSU), qui peut être un particulier, une organisation, ou un fournisseur Cloud de type SaaS ou PaaS, a besoin de garanties pour ses services en termes de qualité de service (QoS : Quality of Service) de bout en bout, d'un haut niveau de fiabilité, et de disponibilité continue. Cependant, il n'est pas simple pour un utilisateur final ou une entreprise voulant faire appel à un service de Cloud de comparer les différentes offres de service avant d'élire l'offre la plus appropriée. Ainsi, la spécification d'un SLA (Service Level Agreement) intégrant les caractéristiques de QoS et de sécurité des offres de service des fournisseurs de Cloud (CSP) constitue un élément très important de différenciation entre les environnements de Cloud. De plus, un SLA peut être violé en raison d'une charge de travail imprévisible, l'insuffisance des ressources dans le centre de données [67] ou encore une attaque de sécurité sur les ressources de Cloud [68]. Par conséquent, le succès du Cloud exige qu'il y ait une gestion autonome des ressources en accord avec les niveaux de service établis afin de minimiser ces violations.

Le transfert de l'infrastructure de l'entreprise vers le Cloud est une tâche complexe qui comprend des défis techniques et organisationnels. Le Cloud est un nouveau paradigme qui peut présenter une opacité dans la définition de certains composants qui ne sont pas toujours maitrisés par l'utilisateur de ses services. Par conséquent, cette complexité couplée avec l'incertitude peut créer des barrières pour l'adoption du Cloud. Parmi les barrières les plus importantes, nous trouvons la QoS, la sécurité des données, l'incertitude sur les coûts, la perte de contrôle et de la gestion, la conformité réglementaire, les accords SLA, et la portabilité des données [69].

Dans ce chapitre, nous présentons une définition de la QoS, la sécurité et la gestion autonome ainsi que leurs caractéristiques dans un environnement de Cloud Computing. De plus, nous présentons les défis importants dans ces trois domaines en étudiant les activités des organisations de standardisation, ainsi que les projets et travaux de recherche portant sur ces thèmes.

3.2/ Qualité de service dans un environnement de Cloud

3.2.1/ Définition

La qualité de service est un concept de gestion qui a pour but d'optimiser les ressources et de garantir des performances convenables aux applications. Elle est aussi définie comme une capacité de pouvoir établir la priorité des charges spécifiques et d'allouer les ressources nécessaires pour répondre aux niveaux de service requis [50]. De plus, la qualité de service désigne la capacité d'un service à répondre par ses caractéristiques aux différentes exigences de ses CSU [70].

Dans les systèmes de Cloud existants, les CSU utilisent Internet pour accéder à leurs ressources. L'architecture Cloud peut bénéficier du support de réservation de ressources du réseau et la garantie des capacités de la qualité de service à travers le réseau sur lequel les services sont fournis.

3.2.2/ Caractéristiques

La qualité de service permet de véhiculer dans de bonnes conditions un type de trafic donné, en termes de plusieurs paramètres. Ces paramètres de QoS peuvent être associés à un service de type NaaS (par exemple : la latence, la gigue, le taux de perte de paquets, la bande passante, la disponibilité, etc.) ou un service de type IaaS (par exemple : le temps de réponse, la disponibilité, etc.).

La *latence* caractérise le retard entre l'émission et la réception d'un paquet dans le réseau. Certaines applications (temps réels par exemple) nécessitent que les données arrivent dans un temps minimal et borné. La *gigue* est définie comme la variation de ce délai, c.à.d. la valeur absolue de la différence entre les temps d'arrivés de deux paquets successifs et leurs temps de départ. Tout comme la latence, nous cherchons à minimiser la gigue pour certains types d'applications (temps réels ou streaming par exemple). La *bande passante* définit le volume d'information (bits) par unité de temps (seconde) que nous pouvons garantir pour une application. Le *taux de perte de paquets* correspond à la non-délivrance d'un paquet de données. Il est calculé comme étant le pourcentage des paquets non délivrés par rapport aux paquets émis. Le *temps de réponse* est l'intervalle de temps entre l'arrivée de la requête du CSU à la machine virtuelle dans le centre de données (DC) du CSP et l'envoi de la réponse à cette requête depuis cette machine virtuelle. La *disponibilité* décrit, dans une période définie, le temps où le service était disponible par rapport au temps total d'ouverture de ce service.

Chacun de ces paramètres peut être important pour certains types d'applications et très peu influent pour d'autres. Par exemple, pour les applications interactives temps réel la latence est importante. Cependant, ce paramètre n'est pas aussi important dans le cadre d'applications de type transfert de fichiers contrairement au paramètre taux de perte. Le modèle «payez ce que vous utilisez» permet aux CSP d'offrir leurs services à leurs CSU avec différents niveaux de QoS, et le CSU sera facturé selon son utilisation d'un de ces niveaux. Ainsi, pour assurer une garantie de cette QoS, un contrat doit être établi. Ce contrat est appelé SLA (Service Level Agreement).

3.2.3/ Service Level Agreement

Dans un environnement de Cloud, un accord de niveau de service (SLA : Service Level Agreement) est un contrat bilatéral entre un CSU et un CSP ou entre des CSP. Il énonce les conditions de service et caractérise l'accord sur un niveau de service. Il peut définir la qualité de service requise par un CSU et offerte par les CSP. Une bonne utilisation du SLA permet d'identifier et de définir les besoins du client, fournir un cadre pour la compréhension, simplifier les questions complexes, réduire les zones de conflit, et encourager le dialogue en cas de conflit [71].

La plupart des CSP actuels offrent un moyen de spécifier les accords de niveau de service (SLA). Cependant, ils sont limités exclusivement à la disponibilité des ressources et ne tiennent pas compte de nombreux autres paramètres importants tels que le temps de réponse et la bande passante, etc. [72]. Par exemple, Amazon EC2 [73] propose différentes instances de machine virtuelle en fonction de leur capacité de calcul, mais il fournit uniquement des garanties de disponibilité [74].

3.2.4/ Challenges

Le Cloud fournit des ressources informatiques, logicielles ou matérielles, accessibles à distance, en tant que service. L'adoption de ce modèle soulève un certain nombre de défis, notamment la qualité de service (QoS) des services fournis. Aujourd'hui, la prise en compte de la qualité de service dans le Cloud reste encore incomplète [70]. Certains CSP se sont intéressées à des critères de QoS comme l'indisponibilité du service due à une panne. Cependant, ces solutions ne s'intéressent qu'à un aspect unique de la QoS. Ainsi, les CSU sont facturés en fonction de la quantité de ressources utilisées ou réservées, mais une garantie limitée est assurée quant à la qualité de service des ressources fournies.

De plus, différents types de données et de trafics pour un seul CSU ont différents niveaux d'importance (données critiques, transfert de fichiers, communication temps réel, etc.). Donc pour garantir la QoS, il faut mettre en place des mécanismes permettant de traiter de manière différenciée ces différents types de flux, chacun avec le niveau de service correspondant. L'évaluation des valeurs des paramètres de qualité de service en accord avec les préférences des CSU est considérée comme un défi important. Le CSP doit garantir la délivrance d'un service avec une bonne QoS pour satisfaire les exigences des clients. Ainsi, il y a de grands enjeux concernant la façon d'établir un SLA dans un environnement de Cloud et la façon de sélectionner les meilleurs CSP pour l'allocation des ressources en fonction des exigences de QoS du CSU [75]. Ensuite, il faut assurer une continuité du niveau de service exigé par le CSU au cours de l'exécution de ses applications.

De plus, les ressources qui sont actuellement déployées dans les centres de données devront avoir des liens permettant des connexions rapides, flexibles, et sûres, non seulement entre le CSU et les CSP mais aussi les CSP entre eux. En effet, de nombreuses applications nécessitent des garanties de bande passante entre les sites des utilisateurs et les instances des serveurs qui les exécutent. Dans ce contexte, il faut assurer la standardisation effective des SLA et de la fourniture de la QoS dans le Cloud afin de permettre une communication claire et sans ambiguïté entre les CSP et les CSU.

3.2.5/ Travaux de standardisation portant sur la QoS dans le Cloud

Dans cette section, nous présentons quelques organismes de standardisation qui ont traité la QoS dans le Cloud (voir table 7). Nous remarquons que les travaux réalisés au sein de ces organismes de standardisation viennent de commencer et que cela peut prendre du temps pour développer les différentes propositions afin d'aboutir à des standards approuvés.

Nom	But
Distributed Management Task Force (DMTF) [13]	Travailler sur la QoS, le SLA, les politiques, l'approvisionnement et la surveillance dans le Cloud : activités réalisées dans le groupe de travail «Cloud Management Working Group (CMWG)» créé en Juin 2010.
Cloud Standards Customer Council (CSCC) [76]	Travailler sur la QoS et le SLA : publication du document «Practical Guide to Cloud Service Level Agreements Version 1.0».
European Telecommunications Standards Institute (ETSI) [20]	Travailler sur la QoS et le SLA : publication du document «SLA for Cloud services».
Organization for the Advancement of Structured Information Standards (OASIS) [77]	Travailler sur les moyens d'assurer que les CSP comprennent les exigences des CSU, comme la capacité et la QoS, lors de la conception et de la fourniture des services : activités réalisées par le comité technique «Symptoms Automation Framework Technical Committee».
SNIA Cloud Data Management Interface (CDMI) [78]	Travailler sur la standardisation du stockage et la QoS dans le Cloud.
Open Networking Foundation (ONF) [79]	Étudier des mécanismes de QoS dans la prochaine version de la spécification OpenFlow.

Table 7 – Exemple d'organismes de standardisation de la QoS dans le Cloud.

3.2.6/ Projets de recherche portant sur la QoS dans le Cloud

Dans cette section, nous présentons quelques projets de recherche qui portent sur la QoS dans le Cloud (voir table 8). Nous considérons que notre travail est très aligné avec les objectifs de ces projets. Mais, notre travail de recherche est innovant en considérant l'établissement d'un SLA et la garantie de la QoS pour les services non seulement de type IaaS mais aussi de type NaaS.

3.2.7/ Travaux de recherche portant sur la QoS dans le Cloud

Dans cette section, nous présentons une étude des travaux de recherche qui portent sur la QoS dans le Cloud. Goiri et al [74] proposent une métrique pour spécifier des garanties portant sur les performances du processeur (CPU). Cette métrique permet aux CSP d'allouer et de libérer dynamiquement les ressources de type VM pour un CSU selon leur

Nom	But
MyCloud [80]	Définir et mettre en œuvre un nouveau modèle de Cloud : SLAaaS (SLA aware a Service) qui contient des paramètres de QoS. Ces paramètres de QoS sont spécifiés pour le service de type IaaS seulement.
Reservoir [81]	Définir une architecture de référence pour une nouvelle génération d'IaaS capable de garantir les accords de niveau de service, d'assurer la qualité de service, et de soutenir une fédération de Clouds privés, publics et hybrides. Ce projet utilise des algorithmes pour l'allocation des ressources en conformité avec les SLA établis. Mais ce travail ne traite que les aspects de qualité de service relatifs au service IaaS.
Contrail [82]	Concevoir, mettre en œuvre, évaluer et améliorer un système Open-Source pour les fédérations Clouds. Le projet vise à relever les défis technologiques clés concernant les Clouds commerciaux et universitaires existants comme l'absence d'interfaces standardisés et la mauvaise garantie de la qualité de service en termes de performances et de disponibilité des ressources du Cloud.
Qu4DS [83] (Quality Assurance for Distributed Services)	Offrir un support pour assurer la QoS qui s'appuie sur les plates-formes distribuées comme le Cloud. Il traite le cycle de vie des SLA, tout en augmentant le profit du CSP.
ISONI [84] (Intelligent Service Oriented Network Infrastructure)	Fournir la QoS pour les ressources de type IaaS et les applications interactives en temps réel grâce à la découverte, la réservation et l'allocation des ressources.
EASI-CLOUDS [85]	Permettre une gestion avancée du SLA pour aider à garantir la qualité de service requise.
Broker@Cloud [86]	Assurer une optimisation et une QoS continue dans le Cloud en utilisant des brokers de services Cloud pour surveiller les SLA.
ETICS [87]	Définir un cycle de vie des SLA et de la QoS dans un Inter-Cloud.
MODAClouds [88]	Fournir un environnement d'exécution pour la conception de haut niveau des applications sur des multi-Clouds avec garantiede QoS.

TABLE 8 – Exemple de projets de recherche portant sur la QoS dans le Cloud.

utilisation par ce dernier. Kouki et al [70] présentent une solution pour intégrer la QoS et le SLA dans le Cloud. Ainsi, ils proposent un modèle de service de Cloud appelé SLAaaS (SLA aware as a Service) qui permet la prise en compte des SLA dans un environnement de Cloud. De plus, ces auteurs proposent dans [89] le langage Cloud Service Level Agreement (CSLA) qui permet la définition et l'établissement d'un contrat SLA avec garantie de QoS. Emeakaroha [90] propose et développe un Framework appelé LoM2HiS pour adapter les métriques relatives aux ressources de type IaaS et les métriques relatives aux ressources de type SaaS dans un SLA. Ye et al [91] proposent un modèle extensible basé sur les algorithmes génétiques pour calculer les valeurs de QoS des

services dans le Cloud. Linlin Wu et al [92] présentent une architecture globale pour la gestion de la QoS dans le Cloud et proposent des algorithmes d'allocation de ressources pour les CSP de SaaS qui veulent réduire les coûts d'infrastructure et les violations de SLA. Cependant, la QoS étudiée dans ce travail ne traite pas les services de type IaaS et NaaS. Cicotti et al [93] proposent QoSMONaaS, qui est un logiciel pour la surveillance de la QoS d'un service de type SaaS. Dans ce contexte, les utilisateurs ont le choix entre trois niveaux de services : Gold, Silver, Bronze. Cependant, le seul paramètre de QoS utilisé dans ce travail de recherche est la disponibilité. Park et al [94] proposent un modèle de QoS pour les applications permettant la planification des ressources d'entreprise (ERP : Entreprise Resource Planning) lorsqu'elles sont offertes en tant que service de type SaaS. Trajkovska et al [95] proposent de nouvelles fonctions de QoS grâce à des API au niveau du service de type PaaS dans un Cloud hybride. Ces fonctions sont utilisées par des CSU, qui sont des CSP de service SaaS, afin d'offrir des applications de streaming multimédia à des utilisateurs finaux. Ainsi, ces fonctions sont basées sur des paramètres de qualité de service tels que la gigue et la bande passante pour satisfaire les exigences des applications de streaming. Nathuji et al [96] ont développé un Framework appelé Q-Clouds pour le contrôle de la QoS à plusieurs niveaux et pour l'allocation des ressources afin d'assurer de bonnes performances pour les applications des CSU. Ferretti et al [97] décrivent une architecture pour répondre aux exigences de QoS des applications des clients de Cloud. L'architecture proposée intègre une entité responsable de l'équilibrage de charge qui permet de répartir la charge de calcul entre les différentes ressources. Edmondson et al [98] présentent un algorithme d'exclusion mutuelle distribuée appelé «Prioritizable Adaptive Distributed Mutual Exclusion (PADME)» conçu pour répondre aux besoins de différenciation entre les services et les applications pour les systèmes de fichiers et autres ressources partagées dans un Cloud public. Le paramètre de différenciation utilisé dans ce travail est la priorité. Stantchev et al [99] décrivent et appliquent une approche en trois étapes pour mapper les exigences de QoS des CSU et négocier le SLA correspondant afin d'allouer les ressources IaaS convenables. Ganghishetti et al [100] présentent «MC-QoSMS» qui est une composante du Cloud broker utilisée pour allouer les ressources selon les accords de niveau de service (SLA) entre les utilisateurs et les fournisseurs pour l'offre des services de type IaaS. Saurabh et al [101] proposent des politiques d'ordonnancement qui exploitent l'hétérogénéité entre plusieurs centres de données.

D'autres travaux de recherche ont été réalisés dans le cadre de la garantie de la qualité de service dans les environnements de Cloud [102][103], mais la qualité de service proposée dans ces travaux concerne les services de type SaaS, PaaS ou encore IaaS et ne considèrent pas les services de type NaaS. D'autre part, plusieurs travaux de recherche récents portant sur le service de type Cloud vidéoconférence [104][105][106][107][108] considèrent les paramètres de QoS pour ce type d'application sans étudier l'établissement d'un SLA correspondant. Dans le cadre de cette thèse, nous considérons les services Cloud de type IaaS mais aussi de type NaaS, les paramètres de QoS correspondants ainsi que l'établissement d'un SLA dans un environnement de Cloud Networking basé sur l'approche inter-Cloud Broker et Fédération. De plus, le niveau de service proposé dans le SLA permettra une garantie non seulement de la QoS mais aussi de la sécurité.

3.3/ Sécurité dans un environnement de Cloud

3.3.1/ Définition

La sécurité du Cloud peut être définie comme étant l'ensemble des moyens techniques, organisationnels, juridiques et humains nécessaires pour protéger les données, les applications et l'infrastructure associée au Cloud contre une faiblesse d'ordre logicielle ou matérielle qui peut être exploitée par une ou plusieurs menaces internes ou externes. La sécurité du Cloud implique des concepts tels que la sécurité des réseaux et du matériel ainsi que les stratégies de contrôle déployées afin de protéger les données, les applications et l'infrastructure d'un environnement de Cloud [109].

3.3.2/ Confiance

La notion de confiance dans le Cloud peut être définie comme étant la croyance du CSU dans le fait que le CSP est capable de fournir les services requis de façon précise et infaillible grâce à l'efficacité de ses mécanismes de sécurité et le respect de tous les règlements. De plus, la confiance dans un environnement de Cloud dépend fortement du modèle de déploiement choisi. Dans le cas d'un Cloud privé, le contrôle est réalisé par l'organisation propriétaire de l'infrastructure et la confiance reste alors à assurer à l'intérieur d'une même organisation. Cependant, lors du déploiement sur un Cloud public, le contrôle est délégué au CSP (propriétaire de l'infrastructure) pour appliquer une politique de sécurité adéquate afin de garantir que les mécanismes de sécurité appropriés soient mis en place. La confiance du CSU dépendra de l'efficacité de cette politique de sécurité qu'il ne contrôle pas dans un environnement de Cloud public.

Ainsi, la confiance est liée essentiellement aux processus de sécurité mis en place par le CSP et le modèle de sécurité du Cloud doit être basé sur l'hypothèse que le CSU devrait faire confiance au CSP. Cette sécurité relative à l'offre de service du Cloud peut faire l'objet d'un accord de niveau de service (SLA) qui définit les attentes des CSU et les obligations des CSP pour le provisionnement des services de Cloud [110]. De plus, une tierce partie de confiance peut être utilisée pour faciliter les interactions d'une façon sécurisée et garantir la confiance entre le CSU et les CSP ou entre les CSP qui font confiance à ce tiers.

3.3.3/ Services de sécurité dans un environnement de Cloud

La sécurité offre plusieurs services importants tels que l'authentification, le contrôle d'accès, la confidentialité, l'intégrité, la disponibilité et la non-répudiation. Par conséquent, ces services deviennent des composants importants que nous devons utiliser lors de la conception d'un système de Cloud sécurisé [111][112][113].

L'authentification est définie comme étant l'assurance que l'entité qui demande l'accès est celle qu'elle prétend être. L'authentification permet au CSP de faire confiance à l'identité du CSU suite à une vérification afin de lui permettre l'accès aux ressources déployées dans le Cloud. Une procédure d'authentification qui manque de robustesse peut conduire à un accès non autorisé aux comptes des utilisateurs dans un Cloud, et par conséquent à une violation de la confidentialité et de la vie privée. Le contrôle d'accès est défini

comme étant la capacité de décider, en fonction de l'identité authentifié d'une entité, qui est autorisé à accéder aux ressources du Cloud pour manipuler des données, exécuter des programmes et effectuer des actions.

La confidentialité se réfère à la capacité de permettre seulement aux parties autorisées d'avoir l'accès aux données et aux logiciels protégés dans le Cloud. Par conséquent, dans un environnement de Cloud l'utilisateur est tenu de déléguer la confiance au CSP qui doit assurer la confidentialité en utilisant par exemple le processus de chiffrement et de déchiffrement. La puissance de tout système de cryptographie, qui sera déployé dans un environnement de Cloud, repose sur la technique de distribution des clés d'une façon sécurisée afin de permettre à deux entités d'utiliser ces clés lors de l'échange de données. De plus, le CSP doit protéger la vie privée des CSU. La vie privée est le désir d'un CSU d'avoir une garantie de confidentialité relative à ses données personnelles. Dans ce contexte, l'environnement de Cloud présente un certain nombre de défis juridiques relatifs à la confidentialité.

L'intégrité se réfère à la protection des données et des logiciels dans le Cloud contre la suppression, la modification et l'ajout non autorisés qui peuvent être volontaires ou involontaires. Les CSP mettent en œuvre un ensemble d'interfaces logicielles ou des API que les CSU utilisent pour gérer et utiliser leurs services dans le Cloud. La sécurité de ces services dépend fortement de la sécurité de ces interfaces ou API car si un CSU non autorisé prend le contrôle de ces interfaces, il pourrait modifier, supprimer ou ajouter des données. Par conséquent, les CSP doivent sécuriser ces interfaces afin de maintenir l'intégrité des données et des services d'une façon générale.

La disponibilité se réfère à la propriété d'un système de Cloud d'être accessible et utilisable sur demande par une entité autorisée. Les services du Cloud présentent une forte dépendance vis à vis des ressources réseau et leur disponibilité. Le système de Cloud doit avoir la capacité de poursuivre ses activités même si une panne a eu lieu.

La non-répudiation est définie comme étant la capacité d'assurer qu'une partie d'un contrat ou d'une communication ne peut pas nier la réception d'un message ou d'être la source d'un message envoyé. Ainsi, dans un environnement de Cloud, le CSP doit assurer la traçabilité de tous les accès ou encore les modifications apportées aux ressources et aux données dans des registres vérifiables.

3.3.4/ Service Level Agreement de sécurité

La sécurité dans le Cloud peut être définie dans un accord de niveau de service (SLA : Service Level Agreement) afin de garantir un niveau de sécurité pour chaque service. Cet accord de niveau de service peut être négocié et établi entre le CSP et le CSU ou entre les CSP. Il définit la sécurité mise en place contre les attaques malveillantes et les pannes éventuelles entrainant des problèmes de sécurité.

De plus, la spécification d'un niveau de sécurité mesurable dans le SLA est utile pour améliorer la transparence et la confiance dans la relation entre le CSU et le CSP. Cette spécification d'un niveau de sécurité permet d'établir une sémantique commune afin de gérer la sécurité du Cloud à partir de deux perceptions différentes, à savoir le niveau de sécurité tel qu'il est offert par un CSP et le niveau de sécurité tel qu'il est demandé par un CSU. Cet accord sur le niveau de sécurité spécifié dans un SLA peut être violé en raison d'une attaque de sécurité visant les ressources du Cloud [68].

3.3.5/ CHALLENGES

Dans le Cloud, les ressources sont fournies comme un service accessible via Internet à des clients qui les utilisent à la demande selon leurs besoins. Ces ressources sont disponibles dans les centres de données et accessibles partout, de sorte que le Cloud est un point d'accès unique pour les clients. Par la suite, les données transférées, stockées, traitées, récupérées, ou bien détruites, ainsi que la puissance de calcul et de traitement, l'espace de stockage, et l'utilisation des données par une tierce partie, ont amené à des préoccupations portant sur la confiance et la sécurité de cet environnement de Cloud. Par conséquent, nombreuses organisations ont hésité à externaliser leur système d'informations vers le Cloud en raison de restrictions réglementaires et de ces préoccupations quant à la confiance et la garantie de sécurité.

La sécurité est l'un des défis à relever les plus importants pour l'adoption du Cloud. En effet, l'externalisation du traitement ou encore du stockage des données dans un environnement de Cloud Computing s'accompagne d'un risque de sécurité puisque les données peuvent être compromises à différentes étapes de leur cycle de vie lorsqu'elles sont stockées ou traitées dans le Cloud. De plus, les risques de sécurité peuvent apparaître lors du transfert de ces données du réseau interne de l'entreprise vers le Cloud pour y être stockées ou traitées, ou encore lors du processus de restauration des données. Ainsi, la transition vers une utilisation massive des services du Cloud s'accompagne de plusieurs défis de sécurité et de confidentialité, principalement en raison du fait que dans cet environnement les composants logiciels et matériels qui permettent d'offrir un service appartiennent à de multiples domaines de confiance. Par conséquent, la garantie des services de sécurité dans un environnement de Cloud est beaucoup plus difficile à assurer.

Le problème de la sécurité dans le Cloud soulève de nombreuses questions, en particulier pour les CSU, qui ont besoin de comprendre les risques associés lors de la migration de leurs services vers le Cloud, ainsi que de savoir quels sont les moyens disponibles pour s'assurer que la sécurité de ces données sera maintenue. Toutefois, le modèle de Cloud présente un certain nombre de menaces, y compris des attaques internes et externes sur des données et des logiciels. Cela est dû à la délégation du contrôle des données vers le Cloud et les données deviennent ainsi accessibles à un nombre important de partenaires dans un environnement de Cloud. D'autre part, les dernières études [114] montrent que des milliers nouveaux logiciels malveillants sont développés dans le monde chaque jour. Par conséquent, des mécanismes sécurisés sont nécessaires à la fois pour les CSU et les CSP afin de créer un environnement de Cloud sécurisé.

Dans le cadre d'une utilisation des services de type NaaS, plusieurs CSU peuvent transférer des données vers un CSP ou bien recevoir des données depuis ce dernier. De plus, dans un environnement d'Inter-Cloud, les données peuvent être transférées entre plusieurs CSP. Par conséquent, une mauvaise application des services de sécurité peut aboutir à un risque d'interception de ces données. D'autres menaces possibles peuvent affecter un environnement de Cloud tels que l'usurpation d'identité, ou encore les attaque de type man-in-the-middle, etc. [115]. Dans ce contexte, la Cloud Security Alliance (CSA) [24] a présenté quelques menaces liées aux services de type NaaS comme le phishing, l'analyse de ports, le déni de service (DoS) et le DoS Distribué (DDoS). De plus, Stallings et al. [113] ont énuméré quelques attaques de sécurité passives basées sur l'écoute et la surveillance des données transmises mais aussi des attaques de sécurité actives telles que la mascarade, le rejeu de trames et la modification des messages. De son côté,

l'ITU-T [112] a énuméré quelques menaces de sécurité dans un environnement de Cloud telles que le détournement de session, l'analyse des flux de trafic, l'altération des données, l'usurpation d'identité, et les communications perturbatrices. Ainsi, la sécurisation du service de type NaaS grâce à la protection du réseau qui permet d'accéder aux ressources du Cloud permet une offre de sécurité de bout en bout pour les services offerts dans un environnement de Cloud [112].

Dans le cadre de l'utilisation des services de type IaaS, la responsabilité principale de la sécurité de ce service revient au CSU plutôt qu'au CSP. En effet, un service de type IaaS offre des VM pour le CSU qui installe des systèmes d'exploitation et des applications sur ces VM et il est donc responsable de leur sécurité [116]. Cependant, cela n'empêche pas l'existence de menaces de sécurité dont l'origine incombe à des défaillances de la part du CSP. Ainsi, Studnia et al. [117] ont présenté des exemples de ce type de menaces dans un service de type IaaS tels que la mauvaise configuration de l'hyperviseur qui conduit à la violation de l'isolement des machines virtuelles, l'accès à la mémoire des autres machines virtuelles et le DoS qui empêche d'autres machines virtuelles de fonctionner correctement. De plus, un attaquant peut allouer des machines virtuelles pour produire une attaque en exploitant les vulnérabilités présentes dans le centre de données du CSP. Par exemple, une machine virtuelle d'un attaquant peut contrôler certaines activités d'une VM ciblée sur le même serveur physique à travers un canal caché. Par conséquent, le CSP doit assurer un isolement des machines virtuelles, une authentification forte, et une protection de l'hyperviseur pour remédier au problème des attaques internes grâce à une bonne configuration sécurisée des composants qui offrent le service de type IaaS. Par ailleurs, le CSP doit assurer le déploiement des systèmes de prévention et de détection d'intrusion sur les machines virtuelles et les hyperviseurs pour les protéger contre ces menaces [116]. De plus, il est nécessaire de définir les administrateurs habilités à intervenir sur les infrastructures du client afin de garantir la confidentialité des données des CSU. Un autre risque dans l'offre de service de type IaaS est la conservation et la suppression d'archives, y compris celles condamnés par une destruction. En effet, les données ne peuvent pas vraiment être effacées si le CSU ne sait pas où elles sont stockées [115]. D'autres menaces relatives au service de type IaaS ont été présentées par Khalil et al. [118] telles que le phishing, le DoS, DDoS, l'attaque par injection de logiciels malveillants, etc.

Dans un environnement de Clouds publics ou privés, les services de type IaaS et de type Naas doivent profiter du chiffrement de bout en bout. Le CSP peut utiliser le chiffrement du disque entier, ce qui garantit que toutes les données sur le disque sont chiffrées. Cela empêche également les attaques hors ligne. De plus, le CSP doit veiller à ce que toutes les communications vers les machines virtuelles soient chiffrées. Cela inclut non seulement les communications du CSU vers les machines virtuelles, mais aussi les communications entre les machines virtuelles elles-mêmes. Aussi, le CSP peut déployer des mécanismes tels que le chiffrement homomorphique pour sécuriser les données en cours de traitement. Cependant, il faut bien gérer la distribution des clés de chiffrement, par exemple par une infrastructure à clés publiques (PKI : Public-Key Infrastructure) qui permet d'obtenir des certificats de sécurité. De plus, la certification des clés par une tierce partie peut être utilisée dans le cadre de l'approche Inter-Cloud afin de permettre aux CSP de fournir des services sécurisés de Cloud qui prennent en considération les exigences de sécurité du CSU [111][112]. Ainsi, l'aspect réseau du Cloud est un facteur critique pour l'adoption de cette nouvelle technologie (Cloud). Dans ce contexte, de Cloud Networking est l'un des domaines de recherche récents dans les communautés de re-

cherche portant sur le Cloud. Ainsi, il y a des défis majeurs relatifs à la façon d'établir un SLA dans les réseaux de Cloud et à la sélection des meilleurs CSP pour l'allocation des ressources en fonction de ce SLA comportant des exigences de QoS et de sécurité [119].

D'autre part, si un CSP fait faillite, le client perd l'accès à ses applications et ses données car l'infrastructure du fournisseur n'est plus accessible. Ainsi, le CSU doit connaître le niveau de service en termes de restauration des données. Dans le cas où le CSP n'inclut pas dans son offre une réplication ou un backup des données, une défaillance de l'infrastructure utilisée par le CSU pourrait lui faire perdre une partie ou la totalité de ces données. Un CSP peut disposer de plusieurs Datacenter dans différents pays à travers le monde. Toutefois, il est nécessaire que les CSU prennent connaissance de la localisation de leurs données, car celles-ci peuvent être stockées dans des pays où les lois de confidentialité des données sont différentes. Ainsi, les données dans le Cloud peuvent être présentes sur plusieurs centres de données (DC) afin que le CSP puisse garantir un service de haute disponibilité à ses clients. Dans ce contexte, la sécurisation des données grâce au chiffrement est nécessaire pour leur stockage et transmission vers les différents centres de données.

La sécurité dans un environnement de Cloud peut présenter un enjeu économique important pour les CSU et les CSP. En effet, les attaques distribuées telles qu'un déni de service (DoS) peuvent engendrer des coûts supplémentaires pour le CSP s'il alloue des ressources supplémentaires pour assurer la disponibilité du service. Ainsi, le CSU risque d'avoir son infrastructure hors service avec des ressources allouées mais non consommées. De plus, l'application de la sécurité a un grand impact sur la qualité de service. Cet impact peut être exprimé en termes de surcharge réseau (Overhead). En fait, le stockage et la récupération des informations de sécurité ainsi que le chiffrement et le déchiffrement des données conduisent à une augmentation du trafic dans le réseau responsable d'une latence supplémentaire, et une monopolisation des capacités du processeur pour le traitement correspondant. Par conséquent, il existe une relation entre la qualité de service et la sécurité qui nécessite que ces deux paramètres soient soigneusement et étroitement gérés d'une façon globale et non pas abordés séparément.

Ainsi, le CSP doit être soumis à des audits et des certifications pour fournir les services et les outils nécessaires à la garantie de la sécurité des données du CSU. De plus, pour garantir un niveau de sécurité, le SLA doit définir les méthodes utilisées d'une manière standardisée surtout pour la conformité règlementaire tout en spécifiant l'emplacement des données et les mécanismes déployés pour maintenir et garantir la sécurité de l'infrastructure Cloud qui sera utilisée par le CSU. Par conséquent, les audits et l'adoption des standards sont parmi les moyens qui permettent au CSP de prouver d'une façon indépendante qu'un service de Cloud répond aux critères de sécurité exigés par le CSU visant à accroître la confiance.

3.3.6/ Travaux de standardisation portant sur la sécurité dans le Cloud

Dans cette section, nous présentons quelques organismes de standardisation qui étudient la sécurité dans le Cloud (cf. table 9). Nous remarquons que plusieurs organismes contribuent aux efforts de standardisation portant sur la sécurité de l'environnement de Cloud. Cependant, il n'y a pas actuellement une manière standardisée, qui fait l'unani-

mité, pour assurer et traduire parfaitement les exigences de sécurité à travers des offres de Cloud.

Nom	But
Cloud Standards Customer Council CSCC [120]	Accélérer l'adoption du Cloud en proposant des standards de sécurité et d'interopérabilité liés à la transition vers le Cloud.
GICTF (Global Inter-Cloud Technology Forum) [18]	Présenter des métriques pour le SLA, y compris les attributs de sécurité pour l'environnement Inter-Cloud.
ENISA (European Network and Information Security Agency) [115]	Présenter une liste hiérarchisée des risques organisationnels, techniques et juridiques, avec une comparaison des différents CSP selon leurs pratiques de sécurité. De plus, cet organisme a pour but de présenter les avantages et les inconvénients de sécurité des Clouds communautés, privés et publics.
CSA (Cloud Security Alliance) [24]	Proposer plusieurs documents de référence et des guides portant sur la sécurité du Cloud pour favoriser un niveau commun de compréhension entre le CSU et le CSP en ce qui concerne les exigences de sécurité. De plus, cet organisme a pour objectif de créer des listes de questions réponses avec des conseils pour la garantie de la sécurité du Cloud grâce à des programmes éducatifs portant sur les utilisations appropriées du Cloud et des solutions de sécurité.
ITU-T [17]	Publier des rapports et des guides portant sur la sécurité du Cloud Computing [112], tels que : Security guidelines for Cloud Computing in the telecommunication area (X.ccsec), Security requirements and framework of the Cloud-based telecommunication service environment (X.srfcts), Security functional requirements for SaaS application environment (X.sfcse), Requirement of IdM in Cloud Computing (X.idmcc), X.gpim, Guidelines on information security management for telecommunications (ITU-T X.1051, ITU-T X.1055), Cybersecurity Information Exchange (CYBEX), (ITU-T X.1500 (Overview), ITU-T X.1520 (CVE) et ITU-T X.1521 (CVSS)).
NIST [5]	Réaliser un projet d'évaluation et d'autorisation de sécurité dans le Cloud et étudier les exigences et les défis de sécurité du Cloud pour le gouvernement américain.
ISO/IEC JTC1/SC27 [26]	Étudier la sécurité et la vie privée dans le Cloud : ce travail est réalisé au sein des groupes de travail WG1/WG4/WG5.
OASIS [77]	Développer des guidelines pour l'atténuation des vulnérabilités et l'analyse des risques et des menaces concernant certains cas d'utilisation dans le Cloud.
SNIA Cloud Data Management Interface (CDMI) [78]	Développer une interface standardisée portant sur la sécurité dans le Cloud.

TABLE 9 – Exemple d'organisations de standardisation de la sécurité dans le Cloud.

3.3.7/ Projets de recherche portant sur la sécurité dans le Cloud

Dans cette section, nous présentons quelques projets de recherche qui portent sur la sécurité dans le Cloud (voir table 10). Nous considérons que notre travail est très aligné avec les objectifs de ces projets. Néanmoins, notre travail de recherche est innovant en considérant les paramètres de sécurité lors de l'établissement d'un SLA avec une garantie conjointe de la QoS et de la sécurité pour les services de type IaaS et NaaS.

3.3.8/ Travaux de recherche portant sur la sécurité dans le Cloud

Dans cette section, nous présentons une étude des travaux de recherche qui portent sur la sécurité dans le Cloud. Christmann et al. [130] décrivent une méthode pour identifier les exigences juridiques de sécurité concernant les services du Cloud. Studna al. [117] présentent les vulnérabilités de virtualisation et les contre-mesures pour faire face à ces vulnérabilités. Ali et al. [131] proposent un Framework pour assurer le partage sécurisé des données dans un Inter-Cloud, sans considérer la sécurité relative au service de type NaaS. Fan et al. [132] présentent un autre Framework pour la gestion de la confiance dans un environnement d'Inter-Cloud, mais sans assurer les services de sécurité. Cependant, d'autres travaux de recherche [133][134][135] présentent des architectures pour offrir ces services de sécurité pour le Cloud mais sans traiter les aspects relatifs à la confiance.

Plusieurs travaux de recherche [136][137][138][139][140][141][116] essayent d'offrir la sécurité pour les services de Cloud de type IaaS. Ainsi, Wei et al. [140] décrivent un Framework pour assurer la sécurité et la vie privée pour les ressources de stockage et de calcul. Chavan et al. [141] examinent les défis de sécurité qui sont introduits suite à l'évolution vers les services de type IaaS du Cloud. Karadsheh et al. [116] examinent les risques rencontrés par la mise en œuvre du modèle IaaS dans les organisations et décrivent le rôle des politiques de sécurité pour améliorer la sécurité du modèle de service IaaS.

D'autre part, pour les services de type NaaS, Moraes et al. [142] présentent un Framework permettant d'isoler les CSU qui utilisent des ressources réseaux de Cloud. He et al. [143] décrivent un autre Framework pour assurer la sécurité du réseau dans le Cloud en utilisant des arbres de règle de pare-feu. De plus, pour la gestion autonome de la sécurité, Ruj et al. [144] présentent un cadre de travail pour assurer l'authentification autonome et le contrôle d'accès décentralisée pour le stockage de données dans plusieurs Clouds. Cependant, Sah et al. [145] mettent l'accent sur l'intégration et la gestion des mécanismes d'authentification, d'autorisation et de comptabilité dans l'architecture du système de Cloud mais pas d'une manière autonome.

Plusieurs travaux de recherche [146][147][148][149][150][151] essayent d'inclure la sécurité dans le SLA établi entre le CSU et le CSP. Ainsi, Rak et al. [149] proposent de construire une application de Cloud orientée SLA qui permet la gestion des fonctions de sécurité liées à l'authentification de l'utilisateur et l'autorisation d'utilisation des services. Chaves et al. [151] donnent un aperçu sur un SLA intégrant des paramètres de sécurité. Na et al. [150] présentent une méthodologie d'évaluation de la sécurité basée sur un SLA et un Borker dans un environnement de Cloud.

De nombreuses enquêtes (Surveys) [152][153][154][155][156][157][158][159][160][161][162] que nous retrouvons dans la littérature présentent différentes menaces et défis de sé-

Nom	But
OpenNebula [27]	Utiliser des certificats x509 et permettre une authentification avec OpenNebula Sunstone : l'authentification est déléguée à Apache ou à SSL capable HTTP Proxy et configurée par l'administrateur.
SUPERCLOUD [121]	Développer une nouvelle plateforme de sécurité pour une infrastructure de Cloud. Cette approche de sécurité est axée sur l'utilisateur pour que les clients puissent définir leurs propres exigences en matière de protection.
Scalable & Adaptive Internet Solutions (SAIL) [122]	Décrire une architecture de Cloud Networking en se concentrant sur la sécurité.
SEED4C [123]	Faire un état de l'art afin de proposer des améliorations portant sur sécurité de dans le Cloud.
GEANT3 [124] et GEYSERS [125]	Définir une infrastructure de sécurité pour un Cloud offrant des services de type IaaS.
Open Web Application Security Project (OWASP) [126]	Travailler sur l'amélioration de la sécurité des logiciels dans le Cloud.
CloudSIRT [127]	Traiter les vulnérabilités, les menaces et les incidents afin de préserver la confiance dans le Cloud.
Sucre [128]	Offrir une série d'outils et d'instruments pour traiter la sécurité des données, la confiance, la fiabilité, la sûreté de fonctionnement du système, et la continuité du service.
CloudSpaces [129]	Etudier le traitement de la vie privée et de la sécurité dans un environnement de Cloud
RESERVOIR [81]	Définir une architecture de référence pour une nouvelle génération de services de type IaaS capables de répondre aux nouvelles exigences de sécurité. Il utilise des mécanismes de sécurité pour le déploiement sûr des machines virtuelles.
Contrail [82]	Concevoir, mettre en œuvre, évaluer et améliorer un système open source pour les fédérations de Clouds. Ce projet traite la sécurité et la confiance dans la fédération de Clouds et il a pour objectif de relever les défis technologiques clés relatifs à la sécurité et la confiance aux Clouds commerciaux et universitaires existants.
MODAClouds [88]	Fournir des techniques pour la cartographie des données et la synchronisation entre plusieurs Clouds.
Eucalyptos [29]	Assurer une communication sécurisée entre les processus interne via SOAP et WS-Security.

TABLE 10 – Exemple de projets de recherche portant sur la sécurité dans le Cloud.

curité qui ont été étudiés dans des travaux et des projets de recherche portant sur la sécurité dans un environnement de Cloud. Dans cette thèse, nous considérons la sécurité pour les services de type IaaS et de type NaaS avec la spécification d'un SLA pour assurer la sécurité et la confiance dans un environnement autonome de Cloud Networking en se basant sur l'approche d'Inter-Cloud Broker et Fédération.

Les mécanismes de sécurité utilisés lors de l'offre d'un service de Cloud, doivent être supervisés et gérés afin d'atténuer les menaces de sécurité. Dans ce contexte, la gestion autonome peut être utilisée comme une approche innovante pour améliorer l'efficacité de la garantie du niveau de service non seulement en termes de sécurité mais aussi de QoS grâce à une réduction de l'intervention humaine dans cet environnement de Cloud.

3.4/ Gestion Autonome

3.4.1/ Définition

L'augmentation de la complexité, l'hétérogénéité et la dynamicité de l'environnement de Cloud peuvent conduire à une infrastructure fragile, ingérable et non sécurisée. Ainsi, ces caractéristiques du Cloud exigent aujourd'hui le déploiement d'une stratégie intelligente de gestion pour offrir des services qui répondent aux exigences du CSU. D'autre part, faire une gestion d'une manière statique par des administrateurs, peut donner lieu à divers problèmes de sécurité, de mise à l'échelle, de temps et de coût. Par conséquent, une nouvelle méthode de gestion s'impose et le besoin de l'adoption d'un nouveau paradigme alternatif appelé gestion autonome devient primordial. La gestion autonome dans le cadre d'un environnement de Cloud a pour objectif le développement de systèmes autonomes qui peuvent assurer une auto-gestion des applications et des ressources du Cloud. Ce nouveau paradigme permet une meilleure gestion des attributs de qualité de service et de sécurité qui changent fréquemment en raison de la nature dynamique du Cloud, et qui doivent être bien surveillés d'une manière continue pour respecter les SLA, minimiser les violations, et réduire les pénalités.

3.4.2/ Caractéristiques

La gestion autonome apparaît comme une approche stratégique et holistique pour la conception de systèmes informatiques, des applications et des infrastructures distribuées complexes. Grâce à l'adoption de ce paradigme, ces systèmes peuvent s'auto-gérer conformément à des politiques de hauts niveaux spécifiées par l'administrateur. La gestion autonome permet au système d'adapter ses fonctionnalités aux changements de son environnement afin d'améliorer sa performance, sa sécurité, sa tolérance aux pannes, sa configuration, sa maintenance, etc. Le concept de gestion autonome qui est l'objectif de l'Autonomic Computing est inspiré du fonctionnement du système nerveux humain. Ce dernier décide d'une façon autonome des actions qu'il doit mener pour garder un état stable sans intervention externe. Ainsi, Il vérifie et optimise constamment son état, et s'adapte automatiquement aux conditions changeantes de son environnement.

La gestion autonome est caractérisée par la réalisation de quatre fonctions de gestion d'une façon autonome à savoir [163][164] : l'auto-restauration, l'auto-protection, l'auto-optimisation et l'auto-configuration.

(i) L'auto-restauration (Self-healing) désigne la capacité du système à examiner, rechercher, diagnostiquer et réagir d'une manière autonome aux dysfonctionnements du système. Les composants ou les applications permettant l'auto-restauration doivent être en mesure d'observer des défaillances du système, d'évaluer les contraintes imposées par

l'environnement extérieur, et d'appliquer les corrections appropriées. C'est à dire découvrir et corriger les défauts d'une manière autonome.

(ii) L'auto-protection (Self-protecting) est la capacité du système d'anticiper, identifier, détecter et se protéger de manière proactive et autonome contre les attaques malveillantes de n'importe où, ou des défaillances en cascade qui ne sont pas corrigées par des mesures d'auto-restauration.

(iii) L'auto-optimisation (Self-optimizing) désigne la capacité du système à surveiller et contrôler d'une manière autonome les ressources pour assurer une performance et une efficacité optimales par rapport aux exigences définies. De plus, cette fonction permet de maximiser l'allocation des ressources et leur utilisation pour satisfaire les demandes des utilisateurs.

(iv) L'auto-configuration (Self-configuring) est la capacité du système à adapter d'une manière autonome les paramètres des ressources logicielles ou matérielles dans le but d'atteindre un fonctionnement correct ou encore des meilleures performances. Ainsi, cette fonction permet d'effectuer des configurations selon les politiques prédéfinies de haut niveau et de s'adapter aux changements dynamiques de l'environnement.

3.4.3/ Boucle de contrôle (MAPE-K)

Afin d'atteindre les objectifs de la gestion autonome, les systèmes disposent d'une boucle de contrôle fermée appelée MAPE-K (Monitor, Analyse, Plan, Execute, Knowledge) qui permet d'éviter toute intervention externe. Cette boucle est contrôlée par un gestionnaire autonome (AM : Autonomic Manager) et l'ensemble constitue un système autonome (voir figure 6) [165]. Ainsi, un système autonome est composé d'un gestionnaire autonome qui respecte les politiques de haut niveau spécifiées par les administrateurs afin de gérer d'une façon autonome des ressources matérielles ou logicielles grâce à la boucle de contrôle fermée.

Comme le montre la figure 6, après l'intégration des différentes politiques dans sa base de connaissances, le gestionnaire autonome commence avec la phase de surveillance qui peut concerner les valeurs des paramètres de qualité de service ou de sécurité. Cette surveillance permet d'assurer la collection, l'agrégation, et le filtrage de données afin d'envoyer des rapports sur l'état des ressources gérées grâce aux interfaces de type capteurs. Ensuite, les données recueillies sont transmises à la phase d'analyse afin de corréler ces données conformément aux politiques de la base de connaissances. La décision relative à une demande de changement peut être prise en se basant sur un seuil défini par ces politiques. Ainsi, une demande de changement pourrait être envoyée à la phase de planification afin d'indiquer les actions nécessaires pour atteindre des objectifs spécifiques conformément aux politiques du plan de connaissances. Enfin, ces actions sont envoyées à la phase d'exécution afin de permettre aux modifications d'être effectuées au niveau des ressources et ce grâce aux interfaces de type effecteurs. De plus, les changements résultant de ces modifications seront vérifiés pour mettre à jour la base de connaissances grâce à la phase de surveillance. Ces phases constituent la boucle de contrôle fermée (MAPE-K) de la gestion autonome des ressources.

Un environnement de gestion autonome peut inclure de nombreuses ressources gérées et de nombreux gestionnaires autonomes. Ainsi, chaque ressource gérée peut disposer de son propre gestionnaire autonome ou encore une collection de ressources gérées

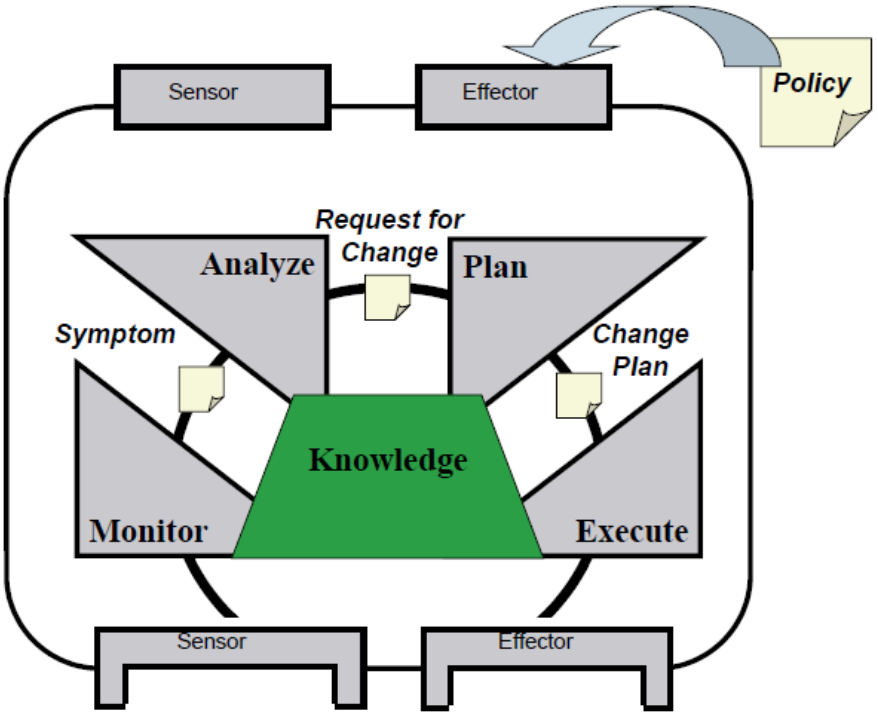

FIGURE 6 – Détails fonctionnels d'un gestionnaire autonome (AM) [165].

peut disposer d'un même gestionnaire autonome. De plus, un gestionnaire autonome peut jouer le rôle d'une ressource gérée auprès d'un gestionnaire autonome de plus haut niveau et disposera ainsi d'une interface de type effecteur et une autre de type capteur pour rendre compte de son état à ce gestionnaire de plus haut niveau. Nous obtenons ainsi plusieurs niveaux de gestionnaires autonomes. Les nombreux gestionnaires autonomes dans un système informatique complexe doivent collaborer ensemble dans le cadre d'une gestion autonome afin d'atteindre des objectifs communs en termes de garantie de la QoS, de sécurité, de réduction des coûts, des violations, et des pénalités, etc.

3.4.4/ CHALLENGES

Aujourd'hui, l'utilisation des services du Cloud par les différents CSU poursuit son augmentation, eu égard aux avantages offerts par cette nouvelle technologie. A ce titre, un grand nombre de vidéos transitant entre le Cloud et les CSU, demande une allocation de ressources réseau une bande passante convenable, plusieurs VM et des espaces de stockage importants. Ainsi, la gestion de ces ressources dans un environnement de Cloud constitue un challenge afin de satisfaire les exigences des CSU. En effet, la gestion d'un centre de données (DC) du Cloud implique une manipulation efficace des machines virtuelles, des espaces de stockages et du réseau dans ce DC. De plus, il faut gérer le réseau qui relie le CSU avec le CSP et celui qui relie les CSP entre eux. Cependant, un administrateur de Cloud ne peut pas gérer un grand nombre de ressources en assurant que ses actions sont compatibles avec la qualité du service attendue à un moment donné, et qu'elles seront aussi compatibles dans le futur. Donc, cette situation a conduit à la né-

cessité d'adopter de nouvelles approches pour la gestion autonome de l'infrastructure du Cloud afin de remplacer l'approche traditionnelle manuelle.

Ainsi, afin de gérer d'une façon autonome les ressources dans le Cloud, il faut assurer une auto-configuration de ces ressources ainsi qu'une auto-optimisation de leur utilisation tout en permettant une auto-restauration suite à des pannes éventuelles afin de fournir une meilleure garantie de la qualité de service associée aux services du Cloud. De plus, il faut assurer une auto-protection des données et une gestion sécurisée des identités pour prévenir des vulnérabilités d'authentification, d'autorisation et identifier ainsi les accès non sécurisés des CSU ainsi que les API d'administration non sécurisées, qui peuvent être à l'origine des attaques par phishing, fraude, ou bien par exploitation des failles de sécurité.

L'attribution des ressources à une application exige de suivre régulièrement la charge et la capacité des ressources allouées, la sécurité et la performance actuelle afin d'assurer les objectifs décrits dans le SLA tout en minimisant les coûts. A ce titre, un contrôle préalable doit être effectué pour examiner les objectifs du SLA afin d'éviter les violations et les pénalités qui en résultent. En effet, les pannes causées par les défauts du logiciel, du matériel ou du réseau conduisent à des violations de SLA. Un SLA peut être violé lorsque le CSP fournit un service qui ne correspond pas aux exigences spécifiées par le CSU dans le SLA. De plus, il faut trouver des solutions pour gérer la qualité de service et la sécurité des ressources de Cloud utilisées par les différents CSU. Ainsi, cette gestion des ressources doit être capable de garantir le niveau de service requis par le CSU en cours d'exécution de son application [166].

L'un des défis importants pour les Clouds répartis géographiquement est leurs capacités de collaboration grâce à des algorithmes efficaces pour la gestion des ressources distribuées dans ces différents Clouds et utilisées par les CSU [58]. Ainsi, les CSP doivent garantir la bonne exécution des services offerts par le Cloud, même dans le cas d'une augmentation inattendue de l'accès au service, et ce grâce aux ressources Cloud fournies temporairement par les autres CSP. Par conséquent, lorsqu'une surcharge entrainant la dégradation du niveau de service d'un CSU est détectée au niveau d'un CSP, les ressources disponibles dans les autres CSP doivent être découvertes et réservées d'une façon autonome par l'Inter-Cloud en se basant sur les informations de localisation du CSU et les connexions réseau doivent être instantanément mises en place ou reconfigurées.

3.4.5/ Travaux de standardisation relatifs à la gestion autonome dans le Cloud

Dans cette section, nous présentons des organismes qui contribuent aux efforts de standardisation portant sur la gestion autonome dans le Cloud (voir table 11). Nous remarquons que ces organismes s'intéressent à la partie réseau pour la standardisation de la gestion autonome dans le Cloud.

Nom	But
Open Networking Foundation (ONF) [79]	Adopter la technologie Software-Defined Networking (SDN) à travers des standards ouverts de développement pour faire le pont entre la gestion autonome des ressources réseaux d'un côté et la demande de QoS des applications de type Cloud.
ITU-T [17]	Spécifier une plate-forme unifiée de gestion autonome des périphériques réseau physiques et virtuels, ainsi que la gestion du centre de données du Cloud. De plus, ITU-T a spécifié une méthode de mise en œuvre de routeurs et de commutateurs programmables pour répondre aux exigences de l'autonomie de réseau.

TABLE 11 – Exemple d'organismes de standardisation de la gestion autonome dans le Cloud.

3.4.6/ PROJETS DE RECHERCHE PORTANT SUR LA GESTION AUTONOME DU CLOUD

Dans cette section, nous présentons quelques projets de recherche qui portent sur la gestion autonome des ressources dans le Cloud (voir table 12). Nous considérons que notre travail est très aligné avec les objectifs de ces projets. Néanmoins, notre travail de recherche est innovant en considérant l'auto-établissement d'un SLA ainsi que la gestion autonome des ressources pour garantir le niveau de service en termes de QoS et de sécurité pour les services de type IaaS et NaaS. De plus, dans le cas de violation nous permettons au système autonome de Cloud de calculer les pénalités et de mettre à jour les réputations des CSP. La gestion autonome que nous déployons dans notre environnement de Cloud permet de réduire l'impact de l'offre de sécurité sur la garantie de QoS.

Nom	But
OpenStack :NaaS [167]	Fournir des services d'une manière dynamique dans un environnement Intra/Inter-Cloud en se basant sur une topologie formée par des commutateurs logiciels et matériels.
BtrScript [168]	Concevoir un système autonome pour la gestion des machines virtuelles en se basant sur des actions et des règles de placement.
Entropy [169]	Implémenter une boucle MAPE-K classique. Ce projet se concentre sur la phase de la planification de cette boucle. L'objectif est d'assurer que les règles de placement des VM soient constamment satisfaites.
SAFDIS [170]	Développer des systèmes d'adaptation autonome pour les différents modèles de service de Cloud : IaaS, PaaS et SaaS.
SUPERCLOUD [121]	Développer une plateforme de gestion autonome de l'infrastructure Cloud pour l'auto-protection des services de Cloud afin de réduire la complexité d'administration.

MyCloud [80]	Proposer une gestion autonome des SLA pour contrôler les problèmes de performance, de disponibilité, de l'énergie et des coûts dans le Cloud.
Reservoir [81]	Définir une architecture de référence pour une nouvelle génération d'Infrastructure IaaS avec une gestion autonome du cycle de vie pour la fourniture de services et la mise à l'échelle.
Contrail [82]	Intégrer un système d'exploitation distribué open source pour la gestion autonome des ressources dans des environnements IaaS et PaaS et surveiller les SLA à tous les niveaux d'une fédération de Cloud.
Qu4DS [83]	Permettre une gestion autonome des services des infrastructures distribuées.
EASI-CLOUDS [85]	Permettre une gestion avancée du SLA dans un environnement de Cloud pour améliorer la garantie de la QoS.
SLA@SOI [171]	Mettre en œuvre un Framework pour la gestion des SLA qui peut être facilement intégré dans une infrastructure orientée services (SOI).
FoSII (Foundations of Self-governing Infrastructures) [172]	Développer des modèles et des concepts pour la gestion autonome et l'application des SLA dans les Clouds, la surveillance des ressources lors de l'exécution des services, et la prévention des violations de SLA tout en réduisant la consommation d'énergie.
ADAMANT [173]	Configurer de façon autonome des ressources disponibles dans les Clouds distribués avec une gestion efficace en cas de catastrophes.
CELAR [174]	Permettre une allocation automatique des ressources pour les applications de Cloud.
Cloud-TM [175]	Minimiser les coûts de fonctionnement des applications de Cloud grâce à une fourniture autonome et une auto-optimisation des ressources correspondantes.
StratusLab [176]	Fournir des fonctionnalités pour la gestion dynamique des ressources IaaS dans le cadre d'une fédération de Cloud.

TABLE 12 – Exemples de projets de recherche portant sur la gestion autonome dans le Cloud.

3.4.7/ TRAVAUX DE RECHERCHE PORTANT SUR LA GESTION AUTONOME DU CLOUD

Dans cette section, nous présentons des travaux de recherche qui portent sur la gestion autonome des ressources dans le Cloud. Kouki et al [177] proposent une approche de planification permettant à un CSP offrant des services de type SaaS de trouver, pour une charge de travail donnée, une configuration optimale pour l'application Cloud, afin d'assurer le SLA et minimiser le coût. Patel et al [178] proposent un mécanisme de gestion des SLA dans un environnement de Cloud en utilisant WSLA (Web Services SLA), élaboré pour l'application et la surveillance du SLA dans une architecture orientée services (SOA : Service Oriented Architecture). Faniyi et al [179] décrivent la conception d'un middleware pour une fédération de Cloud capable de répondre aux demandes des CSU

avec les offres des CSP sans violer le SLA. Demchenko et al. [110] présentent une base pour définir une infrastructure de sécurité proposée pour un Cloud offrant des services de type IaaS avec une configuration dynamique des services d'infrastructure de Cloud et une gestion des SLA. Ardagna et al [180] proposent une approche d'allocation autonome des ressources ainsi qu'une gestion des SLA de plusieurs applications en cours d'exécution sur un seul Cloud. Nguyen Van et al. [181][182] spécifient un gestionnaire autonome de ressources pour contrôler l'environnement virtualisé lors de la fourniture des ressources et du placement dynamique des machines virtuelles. Ce gestionnaire vise à optimiser une fonction d'utilité globale qui intègre à la fois le degré d'accomplissement des SLA et les coûts d'exploitation. Itani et al [183] décrivent la conception et la mise en œuvre de ServBGP, un protocole de routage autonome qui vise à trouver le meilleur chemin en prenant en considération la performance, la sécurité et les exigences de coûts pour la gestion des services et la collaboration entre les CSP dans un environnement de Cloud. Dans les travaux de recherche de cette thèse, nous considérons la gestion autonome des services de type IaaS et de type NaaS avec l'auto-établissement d'un SLA afin d'assurer la QoS et la sécurité dans un environnement de Cloud Networking en se basant sur l'approche d'Inter-Cloud Broker et Fédération. Ainsi, nous assurons la détection et le calcul des violations, des pénalités, et des réputations des différents CSP.

3.5/ Conclusion

Le Cloud doit être considéré comme une opportunité pour améliorer les performances des applications utilisées par les CSU en assurant un niveau acceptable de QoS et de sécurité, et pour réduire la complexité de la gestion de ces applications. Les paramètres de QoS et de sécurité relatifs à l'environnement de Cloud peuvent être spécifiés dans le cadre d'un SLA. Ces paramètres doivent être surveillés d'une manière continue pour respecter les garanties du SLA, en raison de la nature dynamique du Cloud.

Dans ce chapitre, nous avons présenté un état de l'art portant sur trois domaines qui constituent des défis importants dans un environnement de Cloud Computing à savoir la QoS, la sécurité et la gestion autonome. Nous avons définit ces concepts et décrit leurs caractéristiques, ainsi que les challenges relatifs à ces domaines dans un environnement de Cloud. De plus, nous avons présenté les efforts de standardisation mais aussi les projets et les travaux de recherche effectués pour relever ces défis.

Dans les chapitres suivants, nous proposons un Framework pour la garantie d'un niveau de service demandé par le CSU. Ce niveau de service intègre des paramètres de QoS et de sécurité en utilisant différents types de SLA proposés dans une architecture de Cloud Networking. Cette architecture sera ensuite gérée d'une manière autonome grâce à des gestionnaires autonomes spécifiques proposés dans le cadre d'un environnement de Cloud.

CHAPITRE 4
PROPOSITION D'UNE ARCHITECTURE POUR L'OFFRE DE QOS DANS LE CLOUD NETWORKING

4.1/ INTRODUCTION

Afin de fournir le niveau de service attendu par le CSU, la conception et l'élaboration d'une architecture performante, fiable et efficace de Cloud Networking sont très critiques. Nous devons assurer la cohérence entre les exigences de la qualité de service demandée par le CSU et les SLA proposés par les CSP pour permettre à plusieurs CSP de collaborer ensemble afin de répondre aux exigences du CSU.

Dans ce chapitre, nous proposons une architecture de Cloud Networking pour l'offre des services de type IaaS et NaaS avec une garantie d'un niveau de service décrivant la qualité de service requise par le CSU. Le CSU indique ses exigences de QoS par rapport aux services requis en utilisant une interface utilisateur graphique. Dans ce contexte, nous proposons deux types d'architecture de Cloud Networking, la première est basée sur l'approche Inter-Cloud de type Broker (voir figure 19) et la deuxième est basée sur l'approche Inter-Cloud de type Fédération (voir figure 21). De plus, nous proposons dans chaque type d'architecture différents types de SLA et nous décrivons les interactions nécessaires pour leurs établissements, afin d'assurer le niveau de service demandé par le CSU.

4.2/ DESCRIPTION DE L'ARCHITECTURE

4.2.1/ TYPES DE CSP ET DES SERVICES OFFERTS

Au sein de notre architecture de Cloud Networking, nous proposons un environnement avec plusieurs CSP interconnectés ensemble. Nous spécifions deux types de CSP. Le premier type correspond au CSP (BoD) qui fournit un service de type NaaS (BoD), par exemple un operateur réseau. Le CSP (BoD) offre ses ressources réseau avec quatre niveaux de service (voir figure 7) (Platinum, Gold, Silver, ou Bronze), et chaque niveau de service est caractérisé par :

— **Service Level ID :** un identifiant unique du niveau de service.
— **NaaS QoS Parameters :** les paramètres de qualité de service qui peuvent être offerts d'une manière quantitative comme la latence, la gigue, le taux de perte des

paquets, la bande passante et la disponibilité.

— **BW Cost :** le coût unitaire d'un niveau de service qui est exprimé par un coût unitaire de la bande passante consommée par le CSU. Cette bande passante sera facturée sur la base de son utilisation. Les équations qui permettent de calculer ce coût seront définies dans le chapitre 5.

— **Monitoring Interval Time :** l'intervalle du temps de surveillance pour calculer les violations et les pénalités durant la gestion autonome des ressources. Ce paramètre est utilisé dans le chapitre 6.

— **Validity Period :** une période de validité (voir figure 11) qui peut être exprimée d'une manière périodique avec un temps de début de service, une périodicité, et un temps de fin de service. De plus, elle peut être exprimée d'une manière continue entre un temps de début et de fin de service.

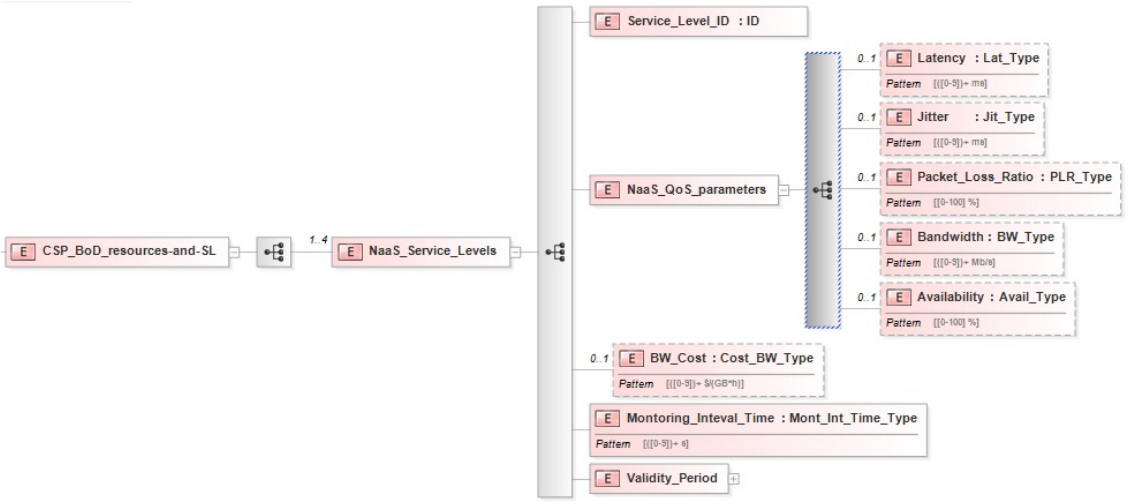

FIGURE 7 – Représentation du schéma XML des niveaux de service associés aux ressources offertes par le CSP (BoD).

Le deuxième type de CSP proposé est le CSP (DC) qui fournit un service de type IaaS (des machines virtuelles et des capacités de stockage) et un service de type NaaS (DC) (ressources réseau dans le Data Center qui relie les ressources locales de type IaaS). Le CSP (DC) peut offrir des ressources émanant d'un ou de plusieurs DC. Le CSP (DC) (voir figure 8) offre des ressources réseau (NaaS) du DC avec quatre niveaux de service (Platinum, Gold, Silver, ou Bronze), et chaque niveau de service est décrit par les mêmes caractéristiques utilisées dans l'offre du CSP (BoD). Cependant, Le CSP (DC) peut aussi offrir, en plus du service NaaS (DC), des ressources IaaS avec une combinaison de niveaux de service de VM (voir figure 9) et d'espaces de stockage (voir figure 10).

Le CSP (DC) offre quatre niveaux de service (Platinum, Gold, Silver, ou Bronze) pour une VM (voir figure 9), et chaque niveau de service est caractérisé par :

— **VM Service Level ID :** un identifiant unique du niveau de service d'une VM.

— **VM Type Characteristics :** les caractéristiques d'une VM pour ce niveau de service. Elles sont définies par le type de VM qui peut être Platinum, Gold, Silver, ou Bronze, le type d'architecture du processeur, le type de l'hyperviseur, la capacité du CPU, le nombre de CPU, la capacité de RAM, la capacité de la mémoire d'une VM, et le nombre de VM disponibles dans le DC.

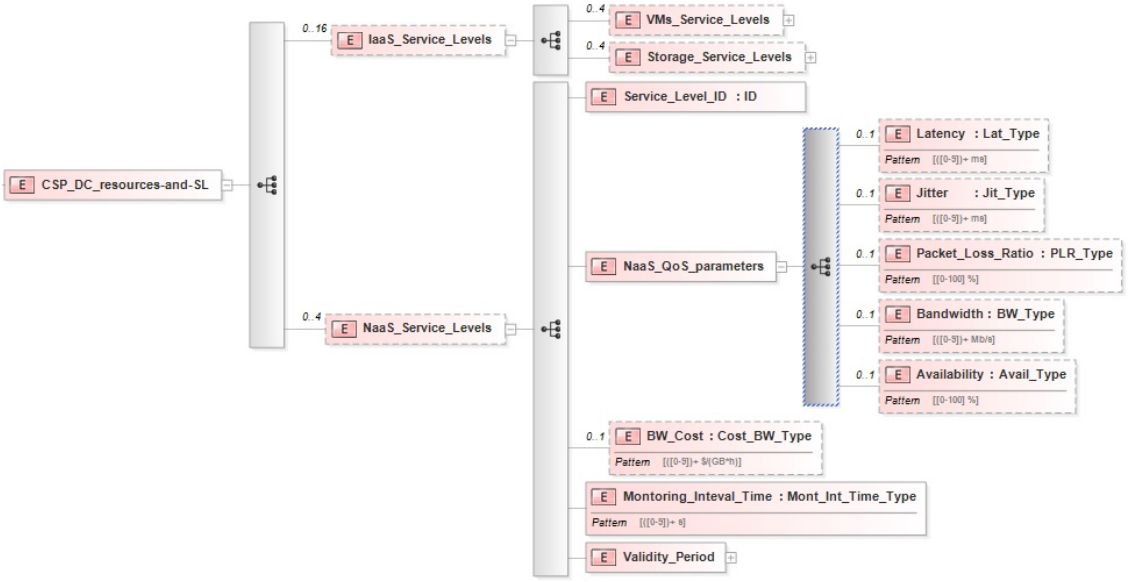

FIGURE 8 – Représentation du schéma XML des niveaux de service associés aux ressources offertes par le CSP (DC).

— **Availability :** un paramètre de QoS qui désigne la disponibilité pour chaque type de VM.
— **VM Type Cost :** le coût unitaire de chaque type de VM. Le CSU sera facturé sur la base de son utilisation. Les équations qui calculent ce coût seront définies dans le chapitre 5.
— **Monitoring Interval Time :** l'intervalle du temps de surveillance
— **Validity Period :** une période de validité du niveau de service

De plus, le CSP (DC) propose quatre niveaux de service (Platinum, Gold, Silver, ou Bronze) pour l'offre d'un espace de stockage (voir figure 10), et chaque niveau de service est caractérisé par :

— **Storage Service Level ID :** un identifiant unique du niveau de service de l'offre de stockage.
— **Storage Characteristics :** les caractéristiques de l'espace de stockage pour ce niveau de service. Elles sont définies par la capacité de stockage et la capacité de lecture/écriture séquentielle dans l'espace de stockage.
— **Availability :** un paramètre de QoS désignant la disponibilité d'un espace de stockage avec quatre niveaux (Platinum, Gold, Silver, ou Bronze).
— **Storage Cost :** le coût unitaire de chaque niveau de stockage. Le CSU sera facturé sur la base de son utilisation. Les équations qui calculent ce coût seront définies dans le chapitre 5.
— **Monitoring Interval Time :** l'intervalle du temps de surveillance
— **Validity Period :** lne période de validité du niveau de service.

Il est à noter que toutes les unités utilisées dans les représentations des schémas XML sont présentées dans l'annexe A.

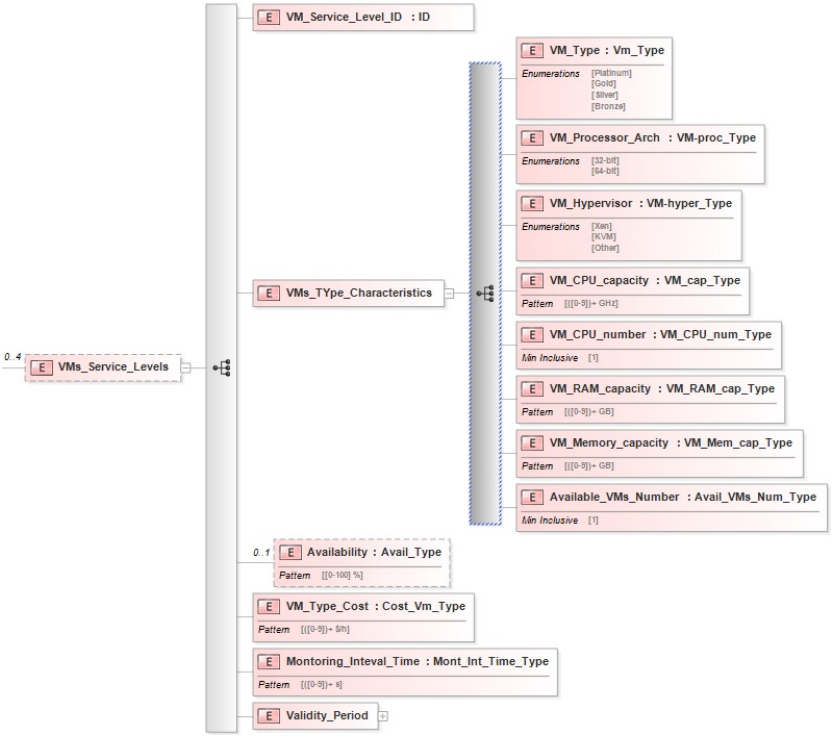

FIGURE 9 – Représentation du schéma XML des niveaux de service de VM.

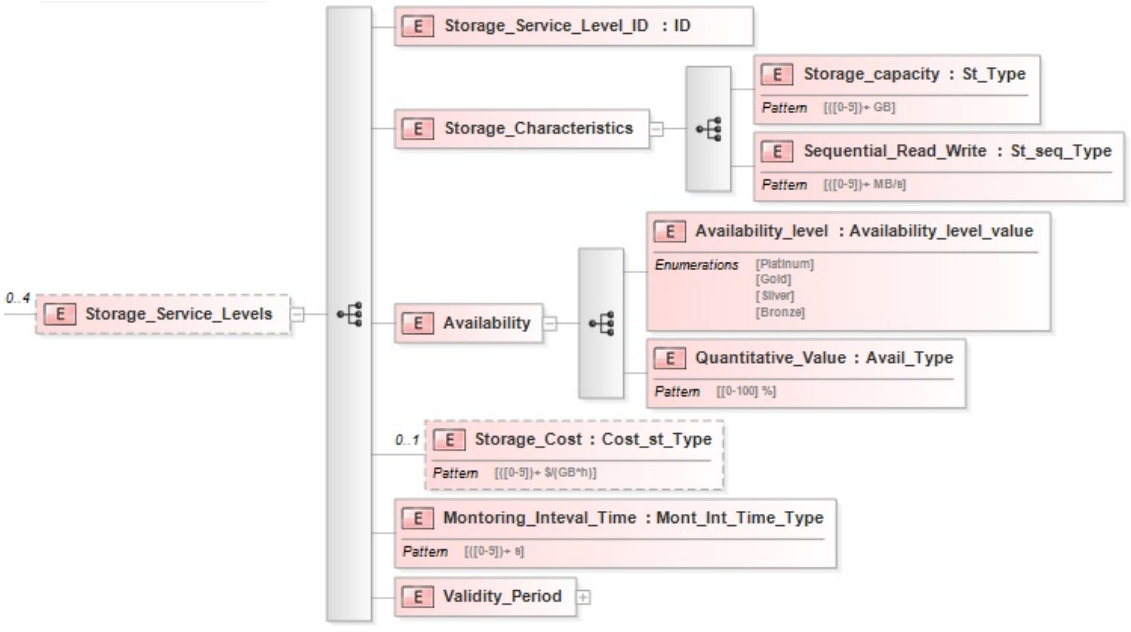

FIGURE 10 – Représentation du schéma XML des niveaux de service de stockage.

4.2.2/ TYPES DE SLA

Un accord de niveau de service (SLA) est un contrat entre deux parties qui peuvent être un CSU, un Cloud Broker, un CSP (DC) ou un CSP (BoD). Il contient différentes

garanties de qualité de service offertes par les CSP et requises par le CSU. Le SLA peut être structuré en utilisant plusieurs attributs :

— **SLA_ID :** un identifiant unique du SLA.
— **Validity Period :** une période de validité (voir figure 11). Cette période peut être définie d'une manière périodique avec un temps de début de service, une périodicité, et un temps de fin de service. Elle peut être aussi définie d'une manière continue entre un temps de début et de fin de service.

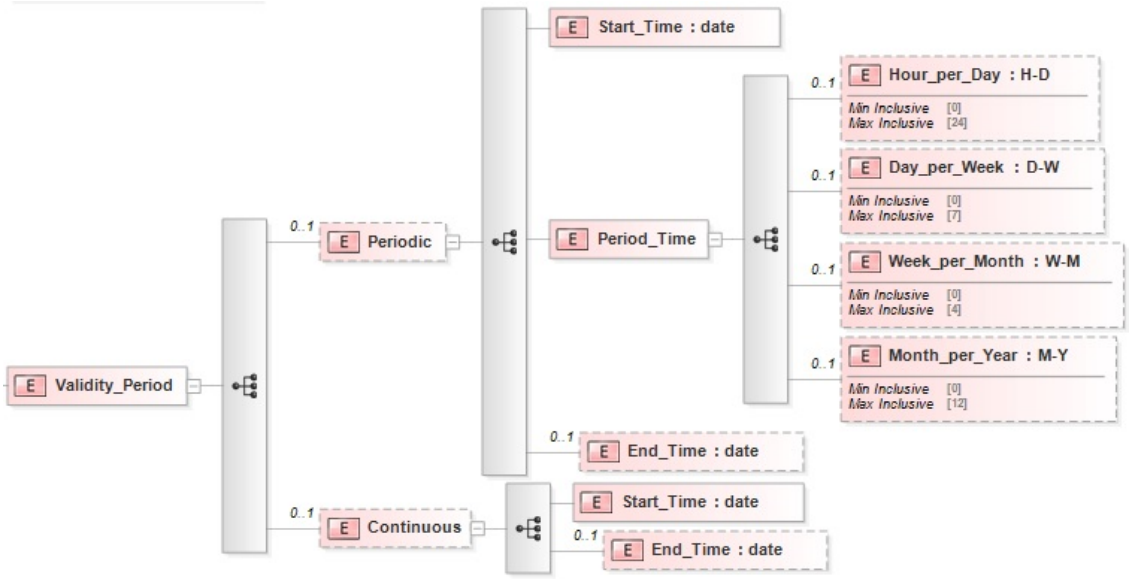

FIGURE 11 – Représentation du schéma XML de l'attribut période de validité d'un SLA.

— **Obliged Parties :** les entités concernées par ce SLA (CSU, Broker, CSP, etc.).
— **Service Type :** le type de service contient un attribut «Service Identification» qui décrit une méthode d'identification de services (voir figure 12) ainsi que les services à garantir tels que les services de type IaaS et de type NaaS (voir figure 13) :
  — L'identification d'un service peut être faite à partir de l'indentification de la source (un site de CSU par exemple) qui veut consommer ce service et de la destination qui peut être une ressource ou un site de CSU par exemple. De plus, le service peut être identifié par le type de l'application (temps réel, streaming, critique, etc.).
  — Un service NaaS peut être de type BoD délivré par un CSP (BoD) qui offre des services de bande passante à la demande avec une garantie de la QoS. Par contre, si c'est un CSP (DC) qui offre ce service, alors il délivre des ressources réseau et de la bande passante avec une garantie de la QoS dans son DC.
  — Un service de type IaaS peut être un service d'offre de VM ou de capacités de stockage. Un service de stockage peut être caractérisé par la capacité de stockage et la capacité de lecture/écriture séquentielle dans l'espace de stockage. Cependant, un service de VM est caractérisé par le type de VM qui peut être Platinum, Gold, Silver, ou Bronze. Il est aussi caractérisé par le type d'architecture du processeur, le type de l'hyperviseur, la capacité de CPU, le nombre de CPU, la capacité de RAM, la capacité de la mémoire de VM, et le nombre de VM requises.
— **Service Performance Guarantees :** l'attribut garanties de performance (voir fi-

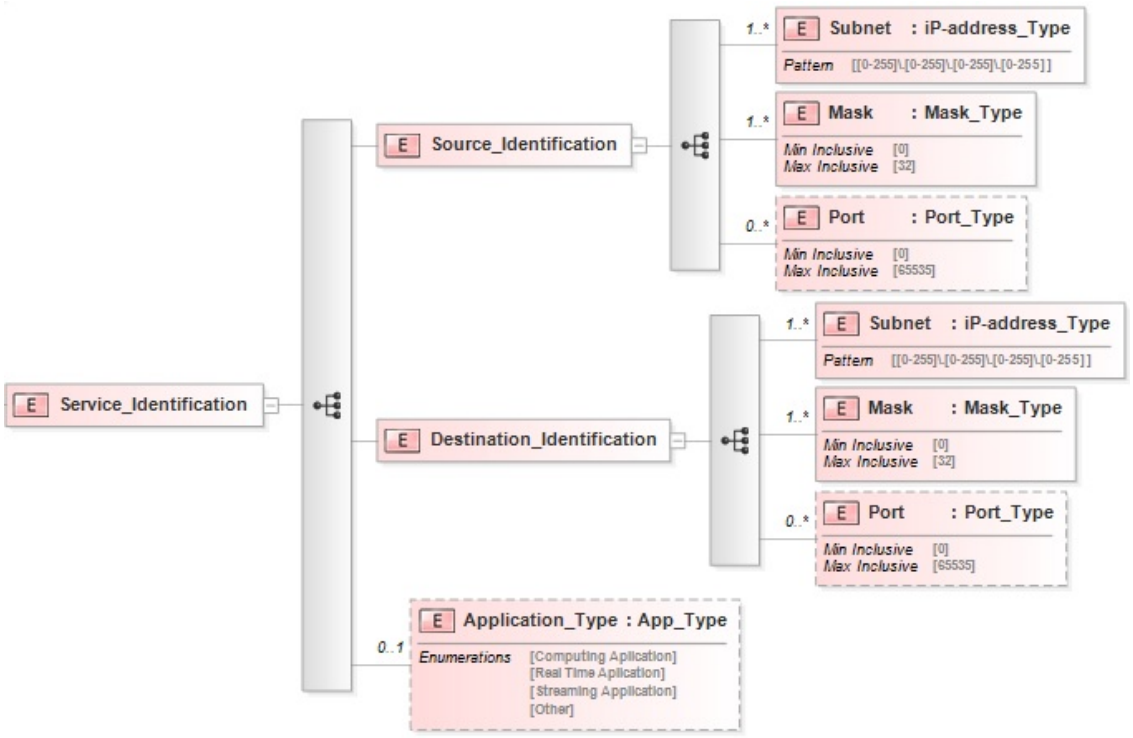

FIGURE 12 – Représentation du schéma XML de l'attribut identification de service.

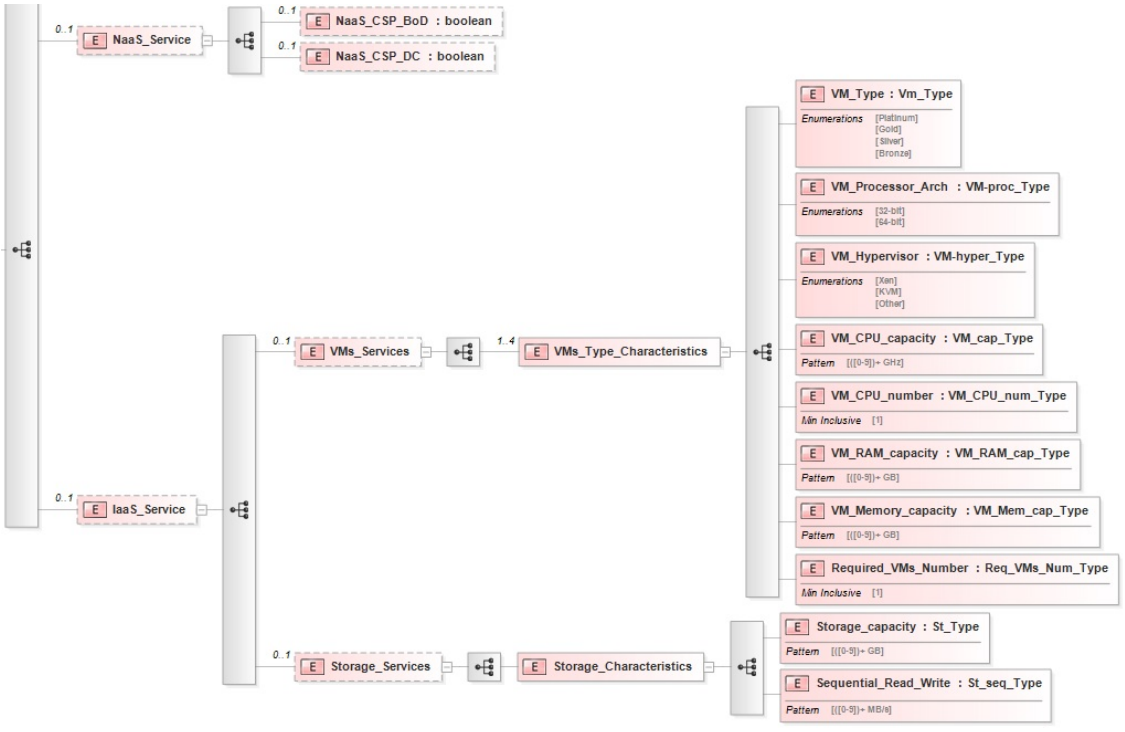

FIGURE 13 – Représentation du schéma XML des services de type NaaS et IaaS.

gure 14) comporte les paramètres de QoS qui seront garantis pour les services de type NaaS et de type IaaS. D'une part, nous définissons les paramètres de QoS suivants pour l'offre de service de type NaaS : la latence, la gigue, le taux de perte

des paquets, la bande passante et la disponibilité. D'autre part, les paramètres de QoS pour les services de type IaaS sont : le temps de réponse et la disponibilité. Ces paramètres peuvent être quantitatifs ou qualitatifs.

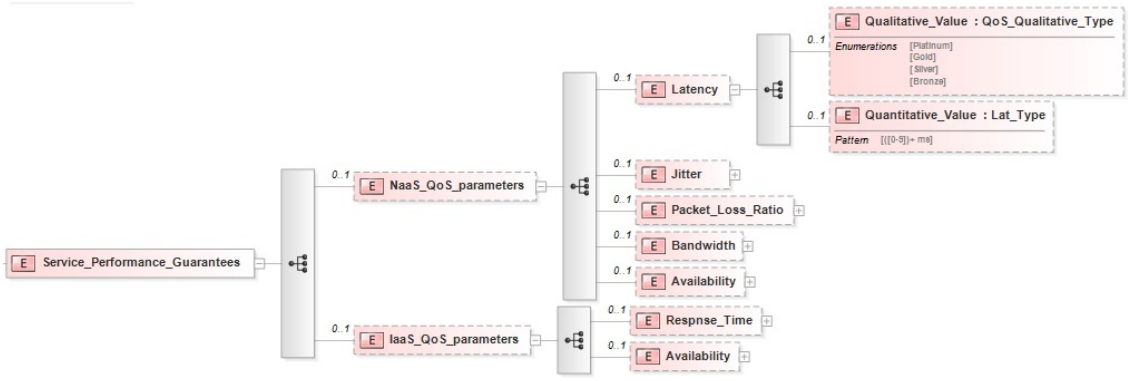

FIGURE 14 – Représentation du schéma XML de l'attribut garanties de performance.

— **Business Attributes :** cet attribut (voir figure 15) contient le coût unitaire de chaque service offert (VM, espace de stockage, bande passante), et l'intervalle du temps de surveillance pour calculer les violations et les pénalités.

De plus, en se basant sur une caractéristique importante du Cloud à savoir «payez ce que vous utilisez», le SLA proposé doit garantir le niveau de service qui permet de satisfaire toutes les exigences du CSU en termes de QoS afin de ne payer que pour les ressources utilisées dans un environnement de Cloud Networking.

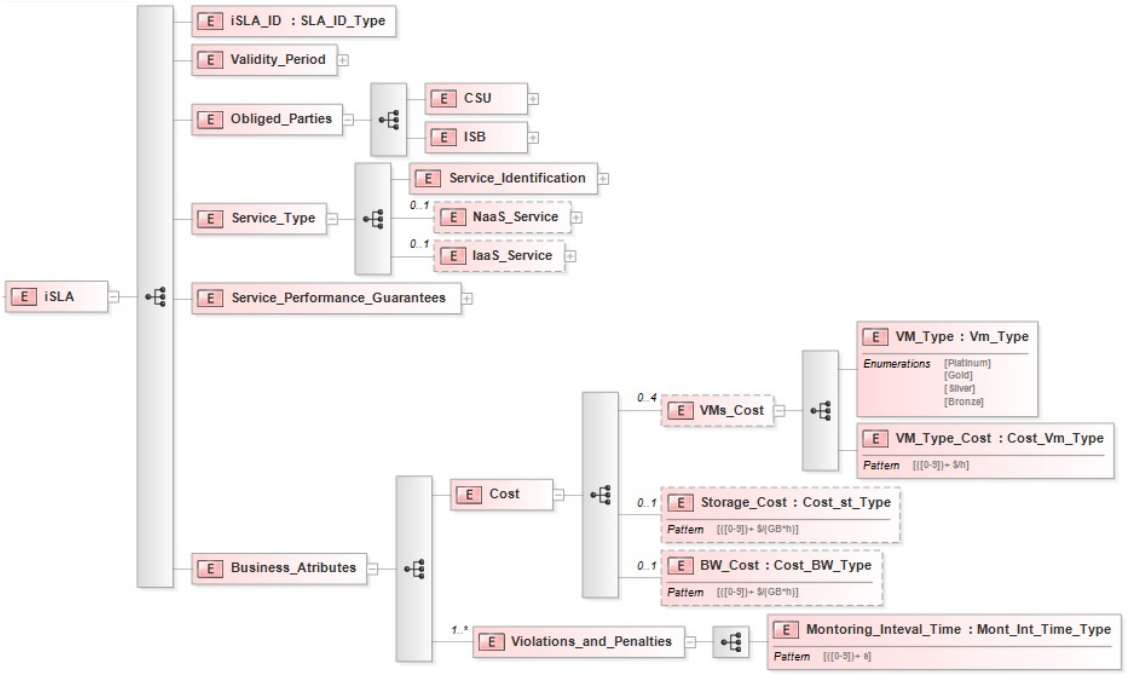

FIGURE 15 – Représentation du schéma XML de l'iSLA.

Ainsi, nous proposons, dans notre architecture de Cloud Networking, trois types de SLA qui sont spécifiés en utilisant le langage XML pour l'interopérabilité et la portabilité entre

les entités :

— **inter-cloud Service Level Agreement (iSLA) :** c'est un contrat entre un CSU et une autre entité qui peut être de type Cloud Broker tel que nous l'utilisons dans l'architecture de type Broker que nous définissons dans la section 4.3 ou encore un CSP tel est le cas dans l'architecture de type Fédération que nous décrivons dans las section 4.4. Ce type de SLA garantit la qualité de service pour les services de type NaaS (BoD et/ou DC) et/ou la qualité de service pour les services de type IaaS (machines virtuelles et/ou espaces de stockage). Les caractéristiques les plus importantes d'un iSLA sont présentées dans la figure 15 pour une architecture de type Broker (c.à.d. un contrat entre un CSU et un Broker). La partie «Bussiness Attributes» du iSLA définit le coût unitaire des ressources de type machines virtuelles, des espaces de stockage et de la bande passante. Ainsi, le CSU sera facturé sur la base de son utilisation des ressources. L'intervalle de temps de surveillance est utilisé pour calculer les pénalités en cas de violation. De plus, le contrat de type iSLA offre des garanties pour les paramètres de QoS relatifs aux services de type IaaS et/ou NaaS, grâce à l'attribut «Service Performance Guarantees». Chaque paramètre de QoS défini dans un iSLA peut être quantitatif ou qualitatif.

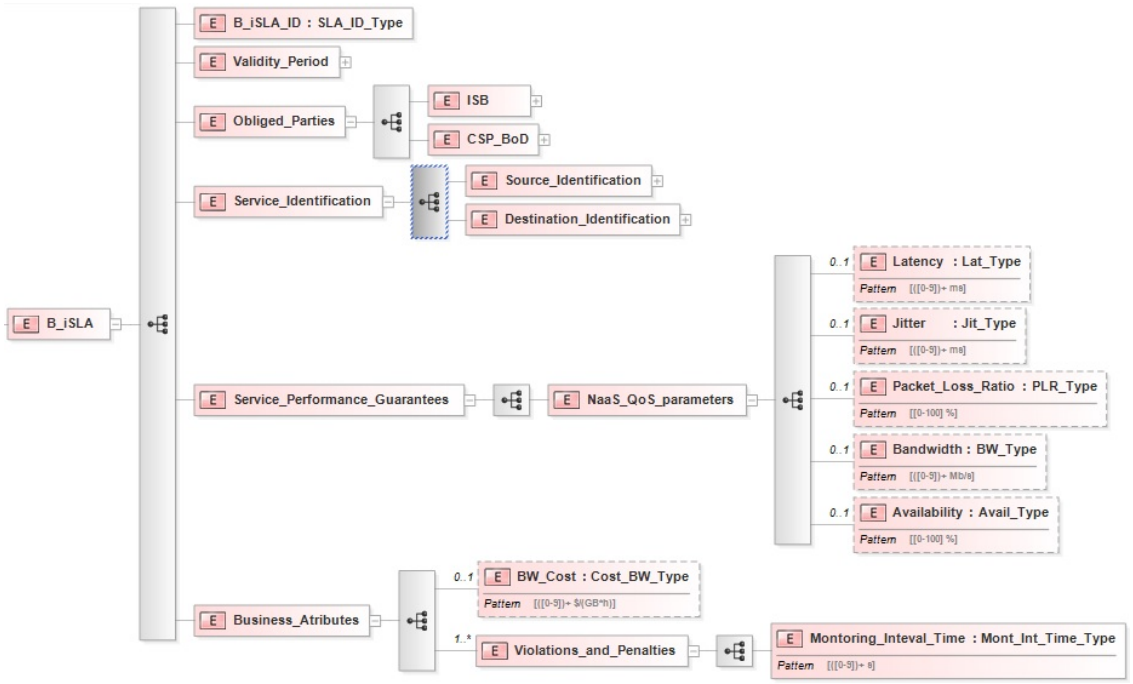

FIGURE 16 – Représentation du schéma XML du B_iSLA.

— **BoD inter-cloud Service Level Agreement (B_iSLA) :** c'est un contrat entre un Cloud Broker et un CSP (BoD) dans le cas d'une architecture de type Broker (cf. section 4.3) ou entre un CSP (DC) et un CSP (BoD) dans le cas d'une architecture de type Fédération (cf. section 4.4). Sa structure ressemble dans la définition de certains attributs à la structure d'un iSLA. Les caractéristiques les plus importantes d'un B_iSLA sont présentées dans la figure 16 pour une architecture de type Broker (c.à.d. un contrat entre un Broker et un CSP (BoD)). Cependant, le B_iSLA ne garantit que la qualité de service pour les services de type NaaS (BoD). De plus, l'attribut «Service Performance Guarantees» contient seulement les paramètres

quantitatifs de QoS pour un service de type NaaS et le «Bussiness Attributes» ne contient que le coût unitaire de la bande passante.

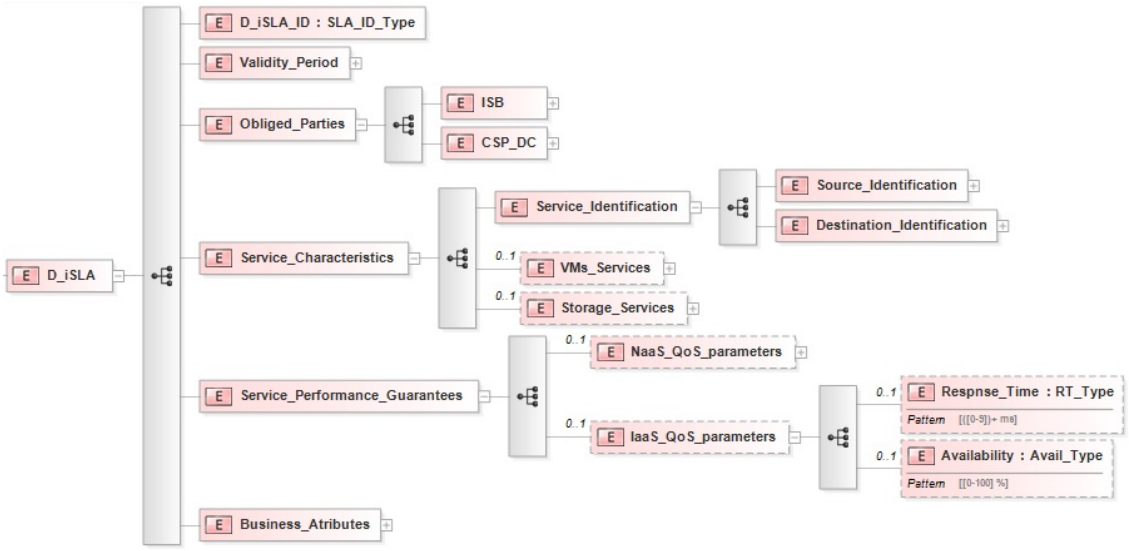

FIGURE 17 – Représentation du schéma XML du D_iSLA.

— **Datacenter inter-cloud Service Level Agreement (D_iSLA) :** c'est un contrat entre un Cloud Broker et un CSP (DC) dans le cas d'une architecture de type Broker (cf. section 4.3) ou entre un CSP (DC) et un autre CSP (DC) dans le cas d'une architecture de type Fédération (cf. section 4.4). Ce type de contrat garantit la qualité de service pour les services de type NaaS (DC) et/ou les services de type IaaS. Sa structure ressemble dans la définition de certains attributs à la structure d'un contrat de type iSLA. Les caractéristiques les plus importantes d'un D_iSLA sont présentées dans la figure 17 pour une architecture de type Broker (c.à.d. un contrat entre un Broker et un CSP (DC)). Les paramètres de QoS pour l'offre de services de type NaaS et IaaS dans l'attribut «Service Performance Guarantees» sont quantitatifs.

4.2.3/ GUI

Le CSU utilise une interface d'utilisateur graphique («Graphical User Interface (GUI)») proposée pour spécifier les services demandés et les exigences de la qualité de service associée (voir figure 18). Le CSU spécifie le nombre de ses sites qui veulent consommer des ressources Cloud à la demande, et pour chaque site il spécifie sa localisation en utilisant des adresses IP. Dans le cas où un CSU communique avec d'autres sites en utilisant les services Cloud alors ce CSU a la possibilité via cette interface de spécifier les sites de destination concernés. De plus, le CSU spécifie, en utilisant la GUI, le type d'application souhaitée (applications temps réel, critiques, interactives, streaming, de calcul), le type de service requis (NaaS et/ou IaaS) ainsi que les paramètres de QoS correspondants (quantitatifs ou qualitatifs) et les poids de ces paramètres qui peuvent être normalisés lors de la sélection des ressources. Ainsi, pour les services de type IaaS, le CSU peut spécifier les différentes longueurs de ses Jobs, le nombre de VM souhaité et le débit d'envoie de ses Jobs. Il peut encore spécifier la capacité de stockage désirée s'il demande des services de stockage.

Ces informations sont envoyées vers l'entité responsable de la garantie des exigences du CSU selon l'architecture de Cloud Networking utilisée. En effet, cette entité responsable peut être un Broker dans le premier type d'architecture que nous proposons ou alors un CSP Leader dans le cas de la deuxième architecture proposée. Ces architectures sont décrites dans la section suivante.

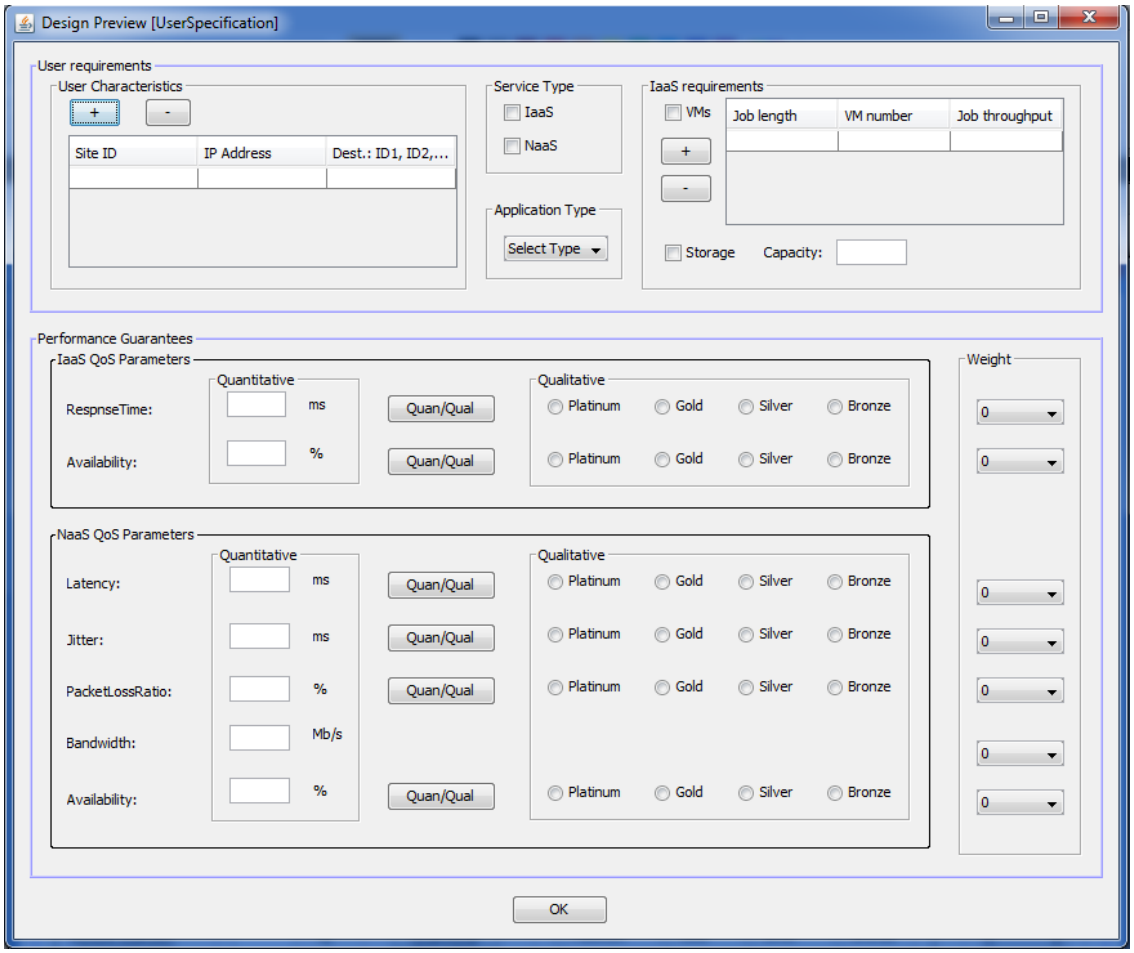

FIGURE 18 – GUI pour les Préférences du CSU.

4.3/ ARCHITECTURE DE TYPE BROKER

4.3.1/ DESCRIPTION DE L'ARCHITECTURE

Dans la première architecture proposée (voir figure 19), le Cloud Broker est émergé comme une entité intermédiaire entre le CSU et les CSP pour aider à l'établissement d'un niveau de service qui répond aux exigences du CSU. De plus le Broker permet de simplifier l'intégration sécurisée des services et d'assurer des coûts moindres pour le CSU. Ainsi, il permet aux entreprises de consommer un portefeuille de services de Cloud qui sont bien intégrés et offerts avec des garanties spécifiées dans un seul SLA global et avec une facturation unifiée. Les CSP (DC/BoD) fournissent des services de type IaaS et/ou NaaS avec différents niveaux de service. Les services de types NaaS concernent

les ressources du réseau de centre de données (DC) qui relient les ressources locales relatives au service IaaS ou concernent les ressources réseaux de type BoD qui relient le CSU aux CSP ou encore les CSP entre eux. Les services de type IaaS concernent les ressources de stockage et de VM.

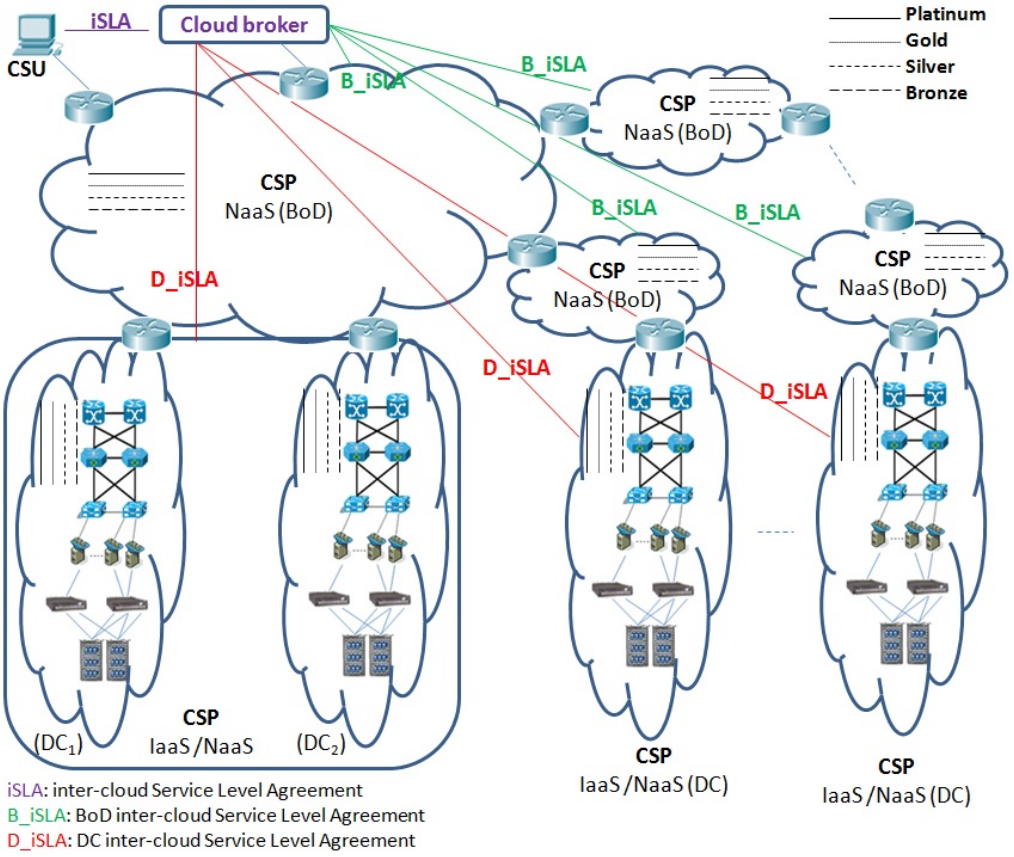

FIGURE 19 – Architecture Cloud Networking de type Inter-Cloud Broker.

4.3.2/ SPÉCIFICATION ET DESCRIPTION DES SLA

Dans notre architecture Cloud Networking de type Broker, nous proposons les trois types de SLA avec la même représentation des schémas XML que nous avons spécifiés dans la section 4.2.2. Cependant, le contrat de type iSLA (voir figure 15) spécifié dans l'architecture de type Broker est un contrat entre le CSU et le Cloud Broker. D'autre part, le contrat de type B_iSLA (voir figure 16) que nous spécifions dans notre première architecture est un contrat entre le Cloud Broker et un CSP (BoD) permettant aux sites du CSU de s'interconnecter ou d'atteindre un CSP (DC) concerné par l'offre de service de type IaaS, et enfin le D_iSLA (voir figure 17) considéré dans cette même architecture est un contrat entre le Cloud Broker et un CSP (DC).

4.3.3/ Interactions entre les entités de l'architecture Broker

Pour établir un SLA de bout-en-bout et assurer une garantie du niveau de service correspondant aux exigences du CSU, plusieurs interactions existent entre le CSU, le Cloud Broker et les CSP (voir figure 20). Au début, les CSP (DC/BoD) décrivent les services de type IaaS et/ou NaaS qu'ils peuvent offrir en se basant sur leurs ressources disponibles avec différents niveaux de service (cf. section 4.2.1). Ensuite, les CSP envoient périodiquement ces informations au Cloud Broker tout en respectant les schémas XML correspondants (cf. section 4.2.1). Cette communication peut se faire en utilisant une interface de Service Web basée sur le langage XML afin d'améliorer l'interopérabilité et la portabilité. Cependant, si des changements importants se produisent dans ces CSP, ces derniers doivent envoyer ces changements immédiatement au Cloud Broker afin de l'informer des nouvelles offres disponibles.

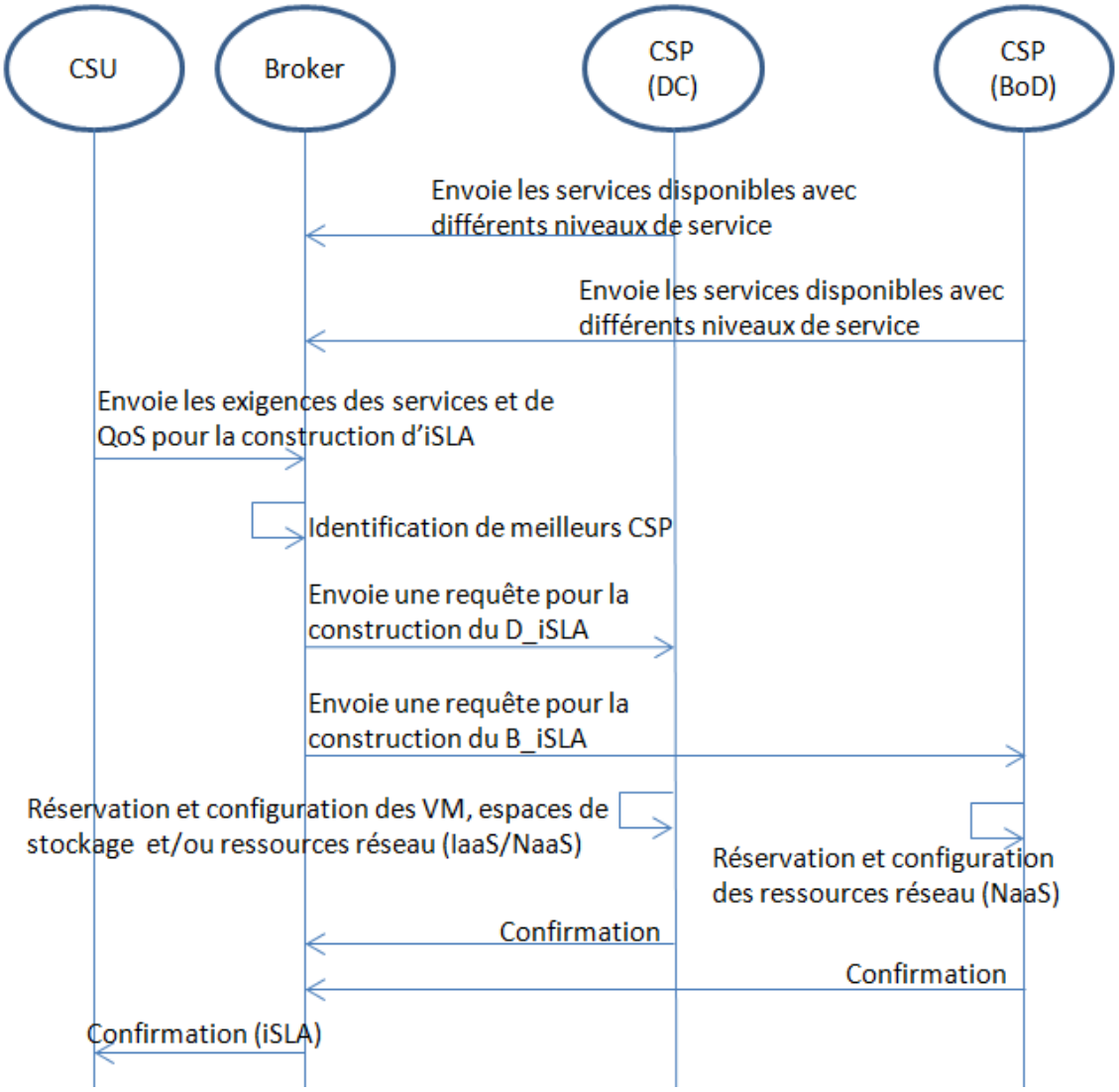

Figure 20 – Interactions entre les entités de l'architecture de type Broker.

De plus, chaque entité (Cloud Broker ou CSP) possède un dépôt (Repository) qui contient des informations sur les ressources disponibles et les niveaux de service correspondants.

Les services de type IaaS reposent sur l'offre de machines virtuelles (VM) et les capacités de stockage avec garantie de QoS. Cependant, les services de type NaaS reposent sur l'offre de bande passante à la demande (BoD) avec garantie de QoS. Dans l'offre de type IaaS, chaque VM est décrite par un type spécifique (Platinum, Gold, Silver, ou Bronze) qui définit ses caractéristiques. Ainsi cette caractérisation peut concerner le type de l'hyperviseur, la capacité de calcul du processeur, le nombre de CPU, la capacité de RAM, et la capacité de mémoire.

Une fois les offres des CSP renseignées au Broker, le CSU utilise l'interface de type GUI proposée (voir figure 18) pour spécifier les services requis et les exigences de QoS. Ces informations sont envoyées vers le Cloud Broker pour construire un iSLA. Le Cloud Broker consulte son dépôt et compare les exigences du CSU avec les niveaux de service offerts par les CSP pour sélectionner les offres appropriées, c'est à dire celles qui répondent aux exigences de qualité de service et associée aux différents services demandés par le CSU et ce en utilisant des algorithmes spécifiques. Le processus de sélection des meilleurs CSP est décrit dans le chapitre 5 (cf. section 5.3). A l'issu de ce processus de sélection, le Cloud Broker envoie une demande pour établir un D_iSLA avec les CSP (DC) sélectionnés et un B_iSLA avec les CSP (BoD) sélectionnés.

Après l'établissement des D_iSLA et B_iSLA, les CSP (DC) concernés réservent et configurent les ressources de type VM, les capacités de stockage et les ressources réseau pour fournir des services de type IaaS et/ou de type NaaS (DC). De plus, les CSP (BoD) concernés réservent et configurent les ressources réseau pour offrir un service de type NaaS (BoD). Une fois les ressources réservées, les CSP mettent à jour leurs dépôts avec ces changements et envoient la confirmation au Cloud Broker. Finalement, le Cloud Broker met à jour son dépôt avec ces changements et établit le contrat de type iSLA avec le CSU.

4.4/ Architecture de type Fédération

4.4.1/ Description de l'architecture

Dans la deuxième architecture proposée (voir figure 21), la fédération offre une alliance entre plusieurs CSP (DC/BoD) qui collaborent ensemble pour aider à l'établissement d'un niveau de service qui répond aux exigences du CSU. Les CSP (DC/BoD) fournissent des services de type IaaS et/ou NaaS avec différents niveaux de service. Les services de types NaaS concernent les ressources du réseau de centre de données (DC) qui relient les ressources locales nécessaires pour l'offre de service de type IaaS ou concernent des ressources réseau de type BoD. Les services de type IaaS concernent les ressources de stockage et les VM. De plus, nous considérons dans l'architecture proposée de type Fédération que le CSU est associé à un CSP (DC) leader (CSP_L).

4.4.2/ Spécification et description des SLA

Dans notre architecture de Cloud Networking de type Fédération, nous proposons les trois types de SLA avec la même représentation des schémas XML que nous avons spécifiés dans la section 4.2.2. Cependant, le contrat de type iSLA que nous spécifions pour l'architecture de type Fédération est un contrat entre le CSU et le CSP_L. De plus,

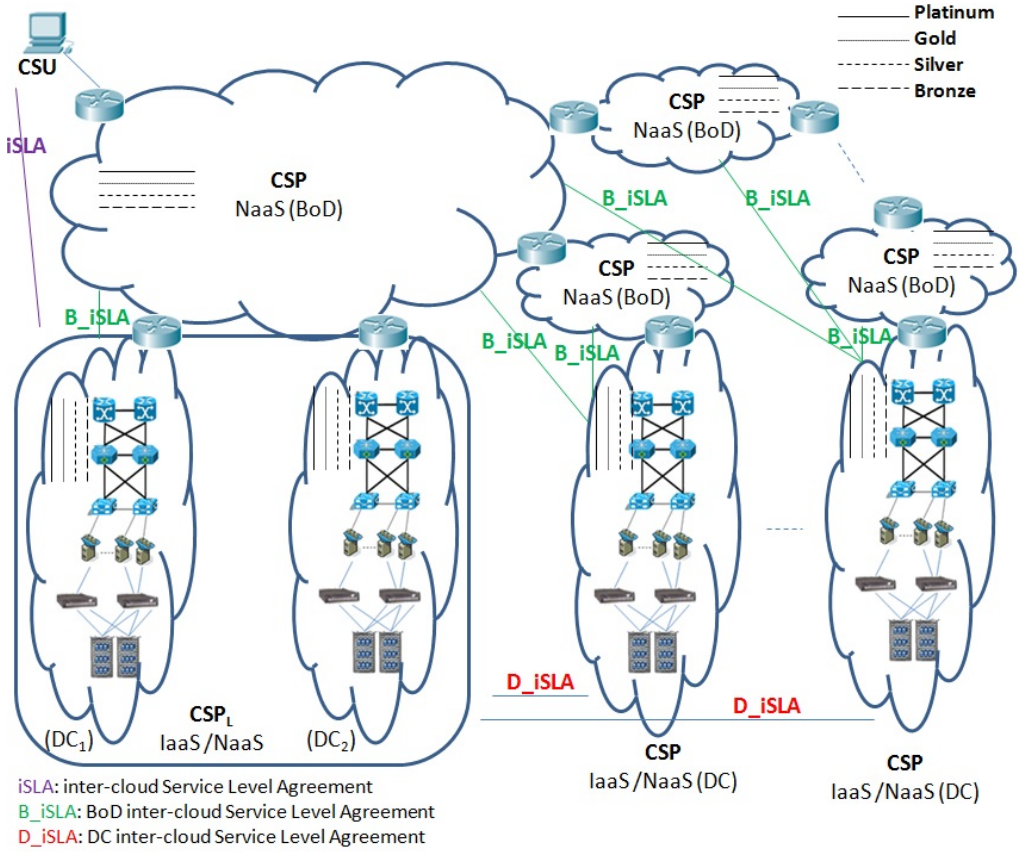

FIGURE 21 – Architecture Cloud Networking de type Fédération.

le D_iSLA considéré est un contrat entre le CSP_L et un CSP (DC). Enfin, le B_iSLA est un contrat entre un CSP (DC) et un CSP (BoD) permettant aux sites du CSU de s'interconnecter ou d'atteindre le CSP (DC) concerné par l'offre de service.

4.4.3/ INTERACTIONS ENTRE LES ENTITÉS DE L'ARCHITECTURE DE TYPE FÉDÉRATION

Les interactions entre le CSU, le CSP_L et les CSP sont décrites selon trois scénarios. Dans le scénario 1 (voir figure 22), le CSU demande un seul service de type NaaS (BoD) sans solliciter le service de type IaaS. Cependant, dans les scénarios 2 et 3 (voir figure 23), le CSU demande un service de type IaaS avec ou sans le service de type NaaS.

Au début, le CSU envoie ses exigences de service au CSP_L afin de commencer l'établissement d'un contrat de type iSLA tout en spécifiant un niveau de service qui contient les paramètres de QoS pour les services de type IaaS et/ou NaaS en utilisant l'interface graphique proposée (voir figure 18).

a) offre de service de type NaaS (BoD) (scénario1) (voir figure 22) : le CSU demande un service de type NaaS (BoD) seulement. Ainsi, le CSP_L va commencer par solliciter les CSP (BoD), qui permettent aux sites du CSU de communiquer,

afin de recevoir leurs ressources disponibles avec les différents niveaux de services correspondants. Ensuite, le CSP_L sélectionne les meilleurs CSP (BoD) tout en considérant ses propres ressources réseau disponibles s'il est un CSP_L (BoD). Cette sélection se base sur un algorithme que nous spécifions dans le chapitre 5 (cf. section 5.3) pour répondre aux exigences du CSU. Après cette sélection, le CSP_L envoie des demandes pour établir des contrats de type B_iSLA avec les CSP (BoD) sélectionnés. Ensuite, si le CSP_L est un CSP (BoD), alors ce dernier va réserver et configurer les ressources réseau pour offrir un service de type NaaS (BoD) avec garantie de QoS. De plus, Les CSP (BoD) concernés réservent et configurent les ressources réseau pour offrir un service de type NaaS (BoD) avec garantie de QoS. Finalement, le CSP_L établit le contrat de type iSLA avec le CSU.

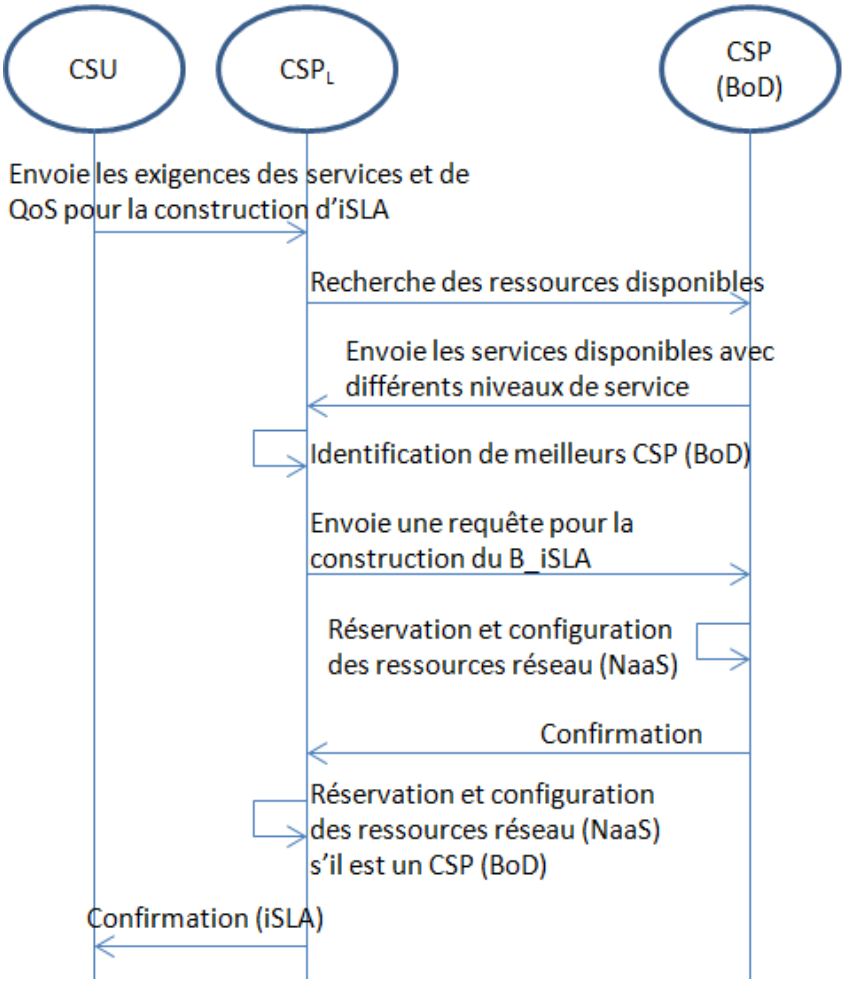

FIGURE 22 – Interactions entre des entités de l'architecture de type Fédération (scénario 1).

b) offre de service de type IaaS avec/sans NaaS (voir figure 23) : Le CSU demande à un CSP_L (DC) un service de type IaaS avec ou sans le service de type NaaS. Ainsi, le CSP_L va commencer par solliciter les CSP (BoD) qui permettent aux sites du CSU de l'atteindre, afin de recevoir leurs ressources disponibles pour le service de type NaaS (BoD) avec leurs différents niveaux de service. Ensuite, le CSP_L sélectionne les meilleurs CSP (BoD) en utilisant un algorithme spécifique.

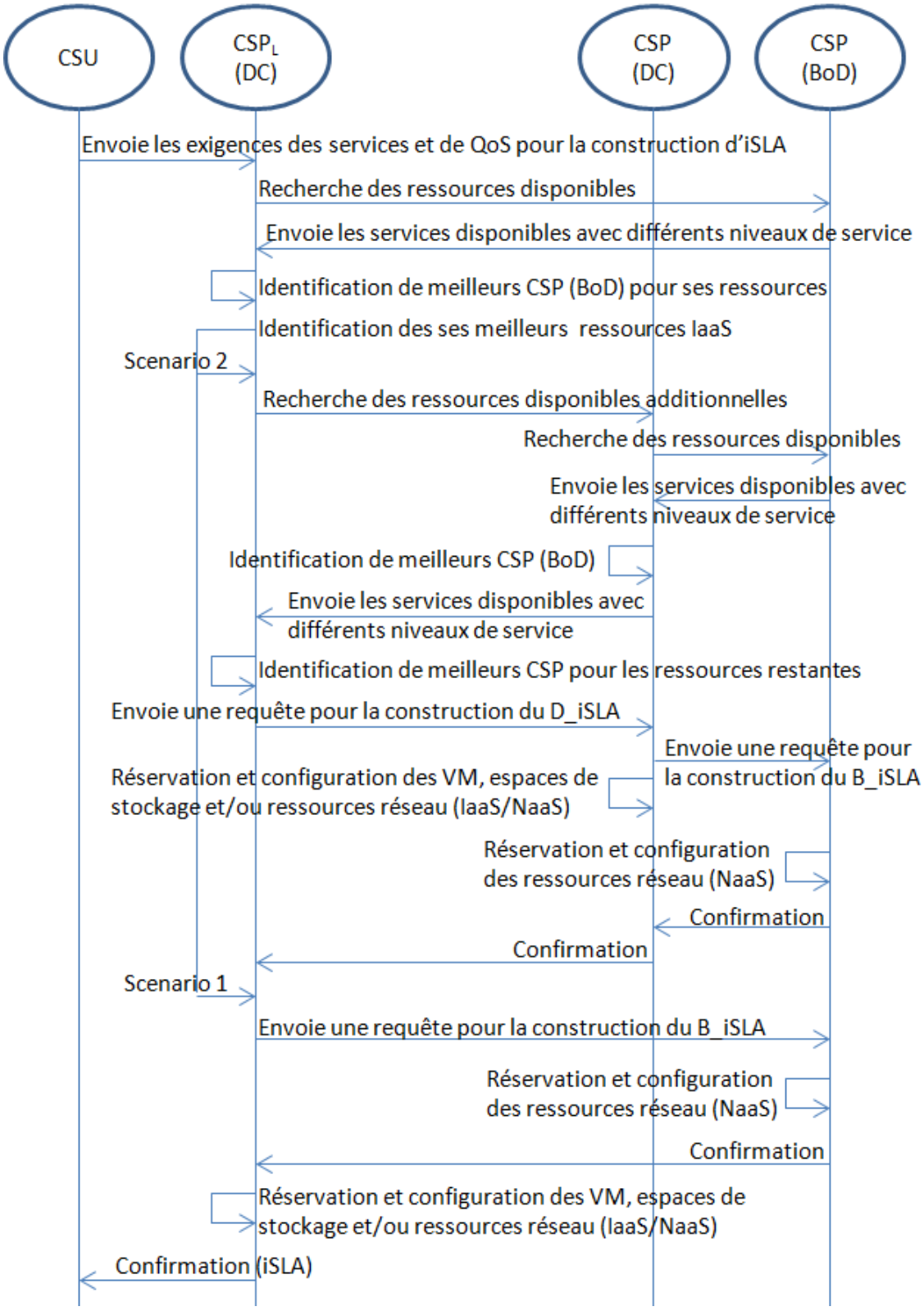

FIGURE 23 – Interactions entre les entités de l'architecture de type Fédération (scénarios 2 et 3).

Le processus de sélection est décrit dans le chapitre 5 (cf. section 5.3). De plus, le CSP_L consulte son dépôt pour identifier ses propres ressources disponibles avec leurs différents niveaux de service afin de voir s'il peut satisfaire lui même toutes les exigences du CSU ou non. Par conséquent, nous pouvons avoir les scénarios suivants :

i) Scénario 2 : dans ce scénario, le CSP_L possède des ressources disponibles qui répondent aux exigences du CSU. Ainsi, le CSP_L sélectionne ses meilleures ressources pour offrir le service de type IaaS en utilisant un algorithme spécifique (cf. Chapitre 5 section 5.3) et envoie des demandes pour établir des contrats de type B_iSLA avec les CSP (BoD) qu'il a déjà sélectionnés.

ii) Scénario 3 : dans ce scénario, le CSP_L n'a pas de ressources suffisantes pour répondre aux exigences du CSU. Par conséquent, il alloue ses ressources disponibles et contacte les CSP (BoD) qui permettent aux sites du CSU de l'atteindre, afin de recevoir leurs ressources disponibles pour le service de type NaaS (BoD) avec leurs différents niveaux de service. Ainsi, le CSP_L sélectionne les meilleurs CSP (BoD). Ensuite, le CSP_L calcule les ressources manquantes et contacte les autres CSP (DC) dans la fédération pour obtenir leurs ressources disponibles. Et par la suite, les CSP (DC) contactent les CSP (BoD) qui permettent aux sites du CSU de les atteindre, afin de demander les ressources disponibles de type NaaS (BoD) avec leurs différents niveaux de service. Ainsi, les CSP (DC) concernés sélectionnent les meilleurs CSP (BoD) en utilisant un algorithme spécifique (cf. Chapitre 5 section 5.3) et envoient les ressources disponibles des services de type IaaS avec/sans NaaS avec leurs différents niveaux de service au CSP_L. Ce dernier sélectionne les meilleurs CSP qui répondent aux exigences de QoS du CSU pour les services de type IaaS avec/sans NaaS. Après cette sélection, le CSP_L envoie des demandes pour établir des contrats de type D_iSLA avec chaque CSP (DC) sélectionné. De plus, le CSP_L et chaque CSP (DC) sélectionné établissent un B_iSLA avec les meilleurs CSP (BoD) qui permettent au CSU de les atteindre.

Ensuite, pour ces deux scénarios correspondant à l'offre de service de type IaaS avec/sans NaaS (scénario 2 et 3), le CSP_L (DC) va réserver et configurer ses ressources de type VM, de stockage et de réseau pour fournir les services de type IaaS avec/sans NaaS (DC) selon les exigences du CSU. De plus, les CSP (DC) concernés réservent et configurent les ressources de VM, de stockage et de réseau pour fournir les services de type IaaS avec/sans NaaS (DC). Les CSP (BoD) concernés réservent et configurent les ressources de réseau pour offrir un service de type NaaS (BoD) avec garantie de QoS si c'est demandé par le CSU. Finalement, le CSP_L établit le contrat de type iSLA avec le CSU.

4.5/ Conclusion

Nous avons proposé dans ce chapitre deux architectures de Cloud Networking. Notre première architecture est basée sur l'approche Inter-Cloud Broker et la deuxième est basée sur l'approche Inter-Cloud de type Fédération. L'objectif des architectures proposées est de permettre aux fournisseurs de Cloud (CSP) d'offrir des services de type IaaS et

NaaS avec une garantie d'un niveau de service décrivant la QoS demandée par le CSU. De plus, nous avons proposé une interface d'utilisateur graphique pour permettre au CSU de renseigner ses exigences de QoS et de spécifier les types de service qu'il veut utiliser à la demande. Afin de décrire la garantie de QoS relative aux services de type IaaS et NaaS, nous avons proposé différents types de SLA. Ainsi, nous avons spécifié trois contrats de niveau de service à savoir iSLA, B_iSLA et D_iSLA. Nous avons utilisé ces contrats dans les deux types d'architecture afin de garantir une QoS de bout en bout lors de l'utilisation par les CSU des services offerts par les CSP. Nous avons aussi montré que l'établissement de ces contrats fait appel à des interactions entre les différentes entités de nos deux architectures de Cloud Networking et à des algorithmes de sélection spécifiques.

Dans le chapitre suivant, nous allons détailler ces algorithmes qui permettent de sélectionner les meilleurs CSP capables de satisfaire les exigences du CSU. De plus, les deux types d'architectures proposés dans ce chapitre seront validés pour deux applications différentes : Cloud vidéoconférence et calculs intensifs.

CHAPITRE 5
GARANTIE DE QOS DANS UN ENVIRONNEMENT DE CLOUD NETWORKING

5.1/ INTRODUCTION

Dans un environnement de Cloud Networking, un CSU peut demander des services de type IaaS et/ou NaaS pour exécuter des applications spécifiques avec des exigences de qualité de service pour ses différents sites. Ces derniers peuvent consommer ces services en utilisant les différents CSP que nous avons défini dans notre architecture de type Broker ou Fédération. Les CSP (DC) offrent des services de type NaaS (DC) et de type IaaS (VM et/ou stockage) tandis que les CSP (BoD) offrent des services de type NaaS (BoD).

Dans ce chapitre, nous présentons le challenge relatif à la sélection des CSP qui offrent les meilleures ressources avec une garantie de bout en bout de la QoS qui répond aux exigences du CSU. Ensuite, nous décrivons les algorithmes que nous proposons en tant que solution pour ce challenge de sélection. Ces algorithmes sont relatifs à l'offre du service NaaS, NaaS avec IaaS, IaaS de type VM, et IaaS de type stockage. De plus, nous détaillons les équations nécessaires pour calculer les différents coûts relatifs à cette garantie de QoS. Enfin, nous spécifions l'environnement de simulation pour la validation de notre proposition de garantie de QoS pour deux types d'applications, à savoir la vidéo-conférence et les calculs intensifs.

5.2/ PROBLÉMATIQUE

La sélection des meilleurs CSP pour offrir les ressources demandées par le CSU tout en assurant ses exigences pour un iSLA donné est un problème d'optimisation multi-objectifs avec contraintes (QoS offerte, coût total, etc.). Ce problème devient important et difficile lorsque le nombre de services fonctionnellement équivalents offerts par les CSP à différents niveaux de service augmente.

Dans ce chapitre, notre objectif est de minimiser le coût total des ressources offertes qui sont soumises à des contraintes de qualité de service de type NaaS et/ou IaaS (contraintes définies dans les équations de (2) à (20)). Nous devons assurer une sélection faisable et optimale des CSP. D'une part, une sélection faisable signifie que les valeurs de QoS agrégées et fournies par les CSP sélectionnés répondent aux exigences de QoS du CSU (voir les équations de (2) à (20)). D'autre part, nous considérons comme

une sélection optimale la sélection faisable qui minimise la valeur globale des coûts (voir les algorithmes de 1 à 4).

Cependant, si nous avons le même coût minimum pour les différentes offres, notre objectif devient alors de maximiser une fonction d'utilité (voir l'équation (1)) pour ces offres afin de sélectionner la solution optimale en utilisant les différents poids normalisés «Wi» attribués dans l'interface graphique GUI par le CSU pour chaque *i-ème* paramètre de QoS. Cette spécification du poids pour le paramètre de QoS est basée sur son importance et le type d'application utilisé par le CSU.

$$f(sl) = \sum_{i=1}^{q}(w_i \times \frac{|Ur_i - Uo_i|}{Ur_i}),\ avec\ \sum_{i=1}^{q} w_i = 1 \tag{1}$$

Ur_i est la valeur du *i-ème* paramètre de QoS demandé par le CSU et Uo_i est la valeur agrégée du *i-ème* paramètre de QoS pour une offre de niveau de service (sl) proposée par le CSP, tandis que q est le nombre de paramètres de QoS spécifiés. Par conséquent, vu que toutes les sélections sont faisables et vu l'utilisation de la valeur absolue, la valeur la plus élevée parmi les valeurs de la fonction d'utilité indique la meilleure offre. Dans cet environnement, une valeur estimée de bout en bout d'un paramètre de QoS peut être calculée en agrégeant les valeurs correspondantes des paramètres de QoS offerts par les CSP. Dans notre modèle, nous considérons trois types de fonctions d'agrégation de la QoS : la relation de sommation ($\sum$), de produit ($\prod$), et de minimisation ($\min$).

5.3/ Algorithmes et contraintes proposés

En se basant sur l'objectif d'optimisation, nous pouvons diviser les paramètres de qualité de service pour assurer un niveau de service requis par le CSU (voir les équations de (2) à (20)). Ainsi, nous choisissons les meilleurs CSP avec un coût optimal (voir les algorithmes de 1 à 4) pour une demande de CSU relative à l'un des quatre types de services suivants : services de type NaaS seulement, services de type IaaS avec les services de type NaaS, services IaaS de VM, et services IaaS de stockage. La notation relative aux variables utilisées dans les algorithmes de sélection et les équations des contraintes est définie dans la table 13.

Symbole	Signification
$Q, Q_1, Q_2,$ T, T_1, T_2	Ensembles temporaires
$T_3[]$	Table des ensembles temporaires
$min_c, flag$	Variables locales
L	Latence (Latency (L)), $L(y)$ est la valeur de la latence dans y exprimée en ms
J	Gigue (Jitter (J)), $J(y)$ est la valeur de la gigue dans y exprimée en ms
P	Taux de perte des paquets (Packet Loss Ratio (P)), $P(y)$ est la valeur du taux de perte des paquets dans y exprimée en $\%$
BW	Bande passante (Bandwidth (BW)), $BW(y)$ est la valeur de la Bande passante dans y exprimée en Mb/s

A	Disponibilité (Availability (A)), $A(y)$ est la valeur de la disponibilité dans y exprimée en $\%$
RT	Temps de réponse (Response Time (RT)), $RT(y)$ est la valeur du temps de réponse dans y exprimée en ms
C	$= \{c_1, \ldots, c_n\}$, ensemble des CSP (DC), n est le nombre de CSP (DC)
c_j	*j-ème* CSP (DC)
S	$= \{s_1, \ldots, s_m\}$, ensemble des sites du CSU, m est le nombre de sites
s_i	*i-ème* site
s_{sr}, s_d	Site source, site destination
R_{ij}	$= \{r_1, \ldots, r_K\}$, ensemble des routes entre s_i et s_j, K est le nombre de routes
r_k	*k-ème* route $= [b_1, \ldots, b_{zk}]$, ensemble des CSP (BoD)
zk	Nombre de CSP (BoD) qui forment la route r_k
b_i	*i-ème* CSP (BoD)
SL_k	$= \{sl_{1k}, \ldots, sl_{Xk}\}$, ensemble des combinaisons des niveaux de services du trafic offert par r_k, Xk nombre de combinaisons
sl_{xk}	*x-ème* combinaison des niveaux de services dans SL_K
$cost_{xk}$	Coût unitaire de sl_{xk} $(\$ \, par \, GB)$
$Cost_t$	Coût total des routes sélectionnées pour tous les sites
IVM	Hash table $= \{[Jlu_1, nbVu_1], \ldots, [Jlu_D, nbVu_D]\}$, exigences du CSU en terme de VM, D est le nombre de différents Jlu requis par le CSU
Jlu_d	*d-ème* longueur d'un job du CSU $(Instructions)$ et la Clé du IVM
$nbVu_d$	Nombre de VM requis par le CSU, et la valeur de IVM, $= IVM[Jlud]$
VT_j	$= \{vt_{1j}, \ldots, vt_{Nj}\}$, ensemble des types de VM dans c_j, N est le nombre de types de VM
vt_{ij}	*i-ème* type de VM dans VT_j
VCa_{ij}	Capacité de traitement de vt_{ij} (GHz)
$nbVc_{ij}$	Nombre disponible de vtij dans c_j
RT_{ij}	Temps de Réponse (Response Time) de vt_{ij} qui correspond à un Jlu_d particulier
$A_{ij}(VM)$	Disponibilité (Availability) de vt_{ij}
VM_cost_{ij}	Coût unitaire de vt_{ij} $(\$ \, par \, heure)$
SCa_u	Capacité de stockage demandée par le CSU (GB)
ST_j	$= \{st_{1j}, \ldots, st_{Mj}\}$, ensemble des types de stockage dans c_j, M est le nombre de types de stockage
st_{ij}	*i-ème* type de stockage dans ST_j
SCa_{ij}	Capacité de traitement de st_{ij} (GB)
$A_{ij}(ST)$	Disponibilité (Availability) de st_{ij} dans c_j
st_cost_{ij}	Coût unitaire de st_{ij} $(\$ \, par \, GB \, par \, heure)$

TABLE 13 – Notation des variables.

5.3.1/ ALGORITHME ET CONTRAINTES RELATIFS À L'OFFRE DU SERVICE NAAS SANS IAAS

Dans cette section, nous supposons que le CSU demande des services de type NaaS seulement, sans les services de type IaaS, pour avoir une connexion réseau entre ses différents sites repartis géographiquement. Donc, le CSU contacte le Cloud Broker (architecture Broker) ou le CSP_L (architecture de type Fédération) en précisant ses besoins

grâce à l'interface graphique proposée.

5.3.1.1/ Algorithme et contraintes pour la sélection des ressources réseau

Dans le cas d'une demande de service NaaS seulement, le CSU spécifie les paramètres de QoS de type NaaS (Latence, Gigue, etc.) nécessaires pour ses applications. Ensuite, le Cloud Broker dans l'architecture Broker ou le CSP_L (scénario 1 décrit dans section 4.4.3) dans l'architecture Fédération sélectionnent les meilleurs CSP (BoD) en termes de coût minimal et de garantie de QoS en utilisant un algorithme que nous proposons pour l'optimisation et la sélection des meilleurs routes entre les sites du CSU (voir algorithme 1). La sélection des ressources est soumise à des contraintes de qualité de service de type NaaS (voir les équations de (2) à (6)).

$$L(iS\,LA) \geq \sum_{i=1}^{zk} L(b_i) \tag{2}$$

$$J(iS\,LA) \geq \sum_{i=1}^{zk} J(b_i) \tag{3}$$

$$P(iS\,LA) \geq 1 - \prod_{i=1}^{zk} (1 - P(b_i)) \tag{4}$$

$$BW(iS\,LA) \leq \min_{i=1}^{zk} (BW(b_i)) \tag{5}$$

$$A_{NaaS}(iS\,LA) \leq \min_{i=1}^{zk} (A(b_i)) \tag{6}$$

Tel que $b_i \in r_k$ et r_k est une route entre s_{sr} et s_d.

Ainsi, comme nous le décrivons dans la figure 24, au début le Cloud Broker ou le CSP_L calcule et trie dans l'ordre croissant les coûts des différentes combinaisons de niveaux de service offerts par les CSP (BoD) pour chaque route entre chaque site source s_{sr} et site destination s_d qui répond aux exigences de la QoS du CSU (voir les contraintes de (2) à (6)). Ensuite, la route avec le coût minimal est choisie. Cependant, le Broker ou le CSP_L calcule et trie dans l'ordre décroissant les valeurs de la fonction d'utilité des différentes offres qui ont le même coût minimal. Par la suite, il choisit la route qui correspond au coût minimal avec la valeur maximale de la fonction d'utilité. L'algorithme proposé utilise la fonction d'utilité avec seulement les paramètres de QoS de type NaaS. Enfin, lorsque le Cloud Broker ou le CSP_L ne trouve pas de routes qui répondent aux exigences du CSU, il libère les ressources allouées et n'accepte pas la demande du CSU. Cependant, si tous les sites ont des routes, il attribue les meilleures routes entre les sites et il accepte la demande du CSU en établissant un contrat de type iSLA avec ce dernier et un contrat de type B_iSLA avec les CSP (BoD) sélectionnés. Il est à noter que la communication entre le site source et le site destination est bidirectionnelle.

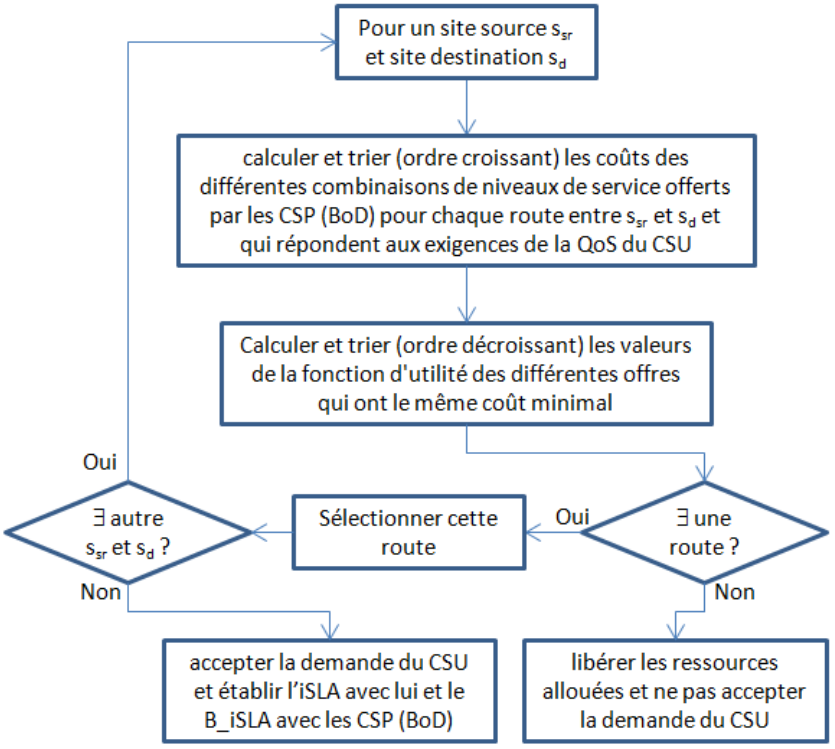

FIGURE 24 – Représentation schématisée de l'algorithme 1.

5.3.2/ ALGORITHME ET CONTRAINTES RELATIFS À L'OFFRE DU SERVICE IAAS AVEC/SANS NAAS

Dans cette section, nous supposons que le CSU demande des services de type IaaS avec ou sans demande de service de type NaaS pour exécuter des applications selon deux manières :

(i) La communication commence à partir du site du CSU vers le DC puis du DC vers le même site utilisateur ou un autre site de CSU (Site/Site). Nous pouvons citer à titre d'exemples pour ce type de communication, le service de type vidéoconférence entre les sites de CSU en utilisant des ressources IaaS dans le DC, ou encore des Jobs de calcul intensif envoyés par un site de CSU et exécutés par des ressources IaaS dans le DC qui va retourner les résultats vers le site demandeur de service.

(ii) La communication commence d'un site de CSU vers un DC ou bien d'un DC vers un site de CSU (Site/DC). Nous pouvons citer à titre d'exemple, le stockage des données dans le DC du Cloud ou encore l'utilisation d'un service de vidéo à la demande (VoD) avec une vidéo stockée dans le DC du Cloud.

Dans les deux cas de communication (Site/Site et Site/DC) le CSU commence par interagir avec le Cloud Broker (architecture Broker) ou le CSP_L (DC) (architecture de Fédération) en précisant ses besoins en utilisant l'interface graphique proposée. Ainsi, il peut spécifier le service demandé de type IaaS avec ou sans services NaaS, ainsi que les VM ou des espaces de stockage correspondants.

Algorithm 1 Sélection des meilleures routes entre les sites (S) du CSU

for each s_{sr} and $s_d \in S$ **do**
  $R_{sc} \leftarrow Get\_routes\_between(s_{sr}, s_d)$
  for each $r_k \in R_{sc}$ **do**
    for each $sl_{xk} \in SL_k$ **do**
      if Constraints (2) to (6) are met with r_k, sl_{xk} **then**
        $Q.Enqueue(r_k, sl_{xk}, cost_{xk})$
      end if
    end for
  end for
  $Q.sorting\_ascending\_order(cost)$
  $\{r_k, sl_{xk}, cost_{xk}\} \leftarrow Q.Dequeue()$ ▷ c.à.d. l'offre avec le coût minimal
  $T.Enqueue(r_k, sl_{xk}, cost_{xk}, f(sl_{xk}))$ ▷ calcul de la fonction d'utilité
  $flag = 0$
  while $Q \neq \emptyset$ && $flag == 0$ **do**
    $\{r_j, sl_{xj}, cost_{xj}\} \leftarrow Q.Dequeue()$
    if $cost_{xk} == cost_{xj}$ **then**
      $T.Enqueue(r_j, sl_{xj}, cost_{xj}, f(sl_{xj}))$
    else
      $flag = 1$
    end if
  end while
  $T.sorting\_descending\_order(f(sl))$
  if $T \neq \emptyset$ **then**
    $\{r_k, sl_{xk}, cost_{xk}, f(sl)\} \leftarrow T.Dequeue()$
    allocate r_k with sl_{xk} and $cost_{xk}$ between s_{sr} and s_d
  else
    release resources and reject CSU request
  end if
end for

5.3.2.1/ Algorithme et contraintes pour la sélection des ressources réseau

Dans le cas d'une demande de service de type IaaS avec/sans NaaS, le CSU peut spécifier les paramètres de QoS de type NaaS (Latence, Gigue, etc.) nécessaires pour ses applications. Ensuite, le Cloud Broker dans l'architecture du même nom ou le CSP_L seulement (scénario 2 défini dans la section 4.4.3 avec un CSP_L qui possède des ressources disponibles afin de répondre aux exigences du CSU) ou avec d'autres CSP (DC) (scénario 3 défini dans la section 4.4.3 avec un CSP_L qui n'a pas de ressources suffisantes pour répondre aux exigences du CSU) dans l'architecture Fédération sélectionnent les meilleurs CSP (BoD) en termes de coût minimal et de garantie de QoS en utilisant l'algorithme 2 que nous proposons pour l'optimisation et la sélection des meilleurs routes entre les sites du CSU el les DC des CSP (DC). La sélection des ressources est faite pour un CSP (DC) spécifique (c_j). De plus, cette sélection est soumise à des contraintes de qualité de service de type NaaS (voir les contraintes de (7) à (11) pour la communication Site/Site et les contraintes de (12) à (16) pour la communication site/DC) si le CSU demande un service de type IaaS avec NaaS. Cependant, si le CSU demande un service de type IaaS seulement sans le service NaaS, l'algorithme nous permet de choisir

seulement les routes avec le coût minimal sans contraintes.

$$L(iSLA) \geq \sum_{i1=1}^{zk_1} L(b_{i1}) + 2 \times L(c_j) + \sum_{i2=1}^{zk_2} L(b_{i2}) \tag{7}$$

$$J(iSLA) \geq \sum_{i1=1}^{zk_1} J(b_{i1}) + 2 \times J(c_j) + \sum_{i2=1}^{zk_2} J(b_{i2}) \tag{8}$$

$$P(iSLA) \geq 1 - \prod_{i1=1}^{zk_1}(1 - P(b_{i1})) \times (1 - P(c_j))^2 \times \prod_{i2=1}^{zk_2}(1 - P(b_{i2})) \tag{9}$$

$$BW(iSLA) \leq \min(\min_{i1=1}^{zk_1}(BW(b_{i1})), BW(c_j), \min_{i2=1}^{zk_2}(BW(b_{i2}))) \tag{10}$$

$$A_{NaaS}(iSLA) \leq \min(\min_{i1=1}^{zk_1}(A(b_{i1})), A(c_j), \min_{i2=1}^{zk_2}(A(b_{i2}))) \tag{11}$$

Tel que $b_{i1} \in r_{k1}$, $b_{i2} \in r_{k2}$ et r_{k1} est la route entre s_{sr} et c_j tandis que r_{k2} est la route entre c_j et s_d.

$$L(iSLA) \geq \sum_{i=1}^{zk} L(b_i) + L(c_j) \tag{12}$$

$$J(iSLA) \geq \sum_{i=1}^{zk} J(b_i) + J(c_j) \tag{13}$$

$$P(iSLA) \geq 1 - \prod_{i=1}^{zk}(1 - P(b_i)) \times (1 - P(c_j)) \tag{14}$$

$$BW(iSLA) \leq \min(\min_{i=1}^{zk}(BW(b_i)), BW(c_j)) \tag{15}$$

$$A_{NaaS}(iSLA) \leq \min(\min_{i=1}^{zk}(A(b_i)), A(c_j)) \tag{16}$$

Tel que $b_i \in r_k$ et r_k est la route entre s et c_j.

Ainsi, comme le montre la figure 25 pour un CSP (DC) spécifique c_j, le Cloud Broker ou le CSP_L avec/sans les autres CSP (DC) calculent et trient dans l'ordre croissant les coûts des différentes combinaisons de niveaux de service offerts par les CSP (BoD) pour chaque route entre un site s et c_j. Ensuite, pour chaque site source s_{sr} avec/sans un site destination s_d, si le CSU demande un service de type IaaS sans NaaS, le Cloud Broker ou le CSP_L avec/sans les autres CSP (DC) choisissent la route avec le coût minimal entre s_{sr} et c_j (Site/DC) ou entre s_{sr} et s_d en traversant c_j (Site/Site). Cependant, si le CSU demande un service de type IaaS avec NaaS, le Cloud Broker ou le CSP_L avec/sans les autres CSP (DC) calculent et trient dans l'ordre décroissant les valeurs de la fonction d'utilité des différentes offres qui ont le même coût minimal des routes entre s et c_j (Site/DC) ou entre s_{sr} et s_d en traversant c_j (Site/Site), et qui répondent aux exigences de QoS du CSU (voir les contraintes de (7) à (11) pour le Site/Site et les contraintes de (12) à (16) pour le Site/DC). De plus, la route qui correspond au coût minimal avec la valeur maximale de la fonction d'utilité est choisie. L'algorithme 2 que nous proposons,

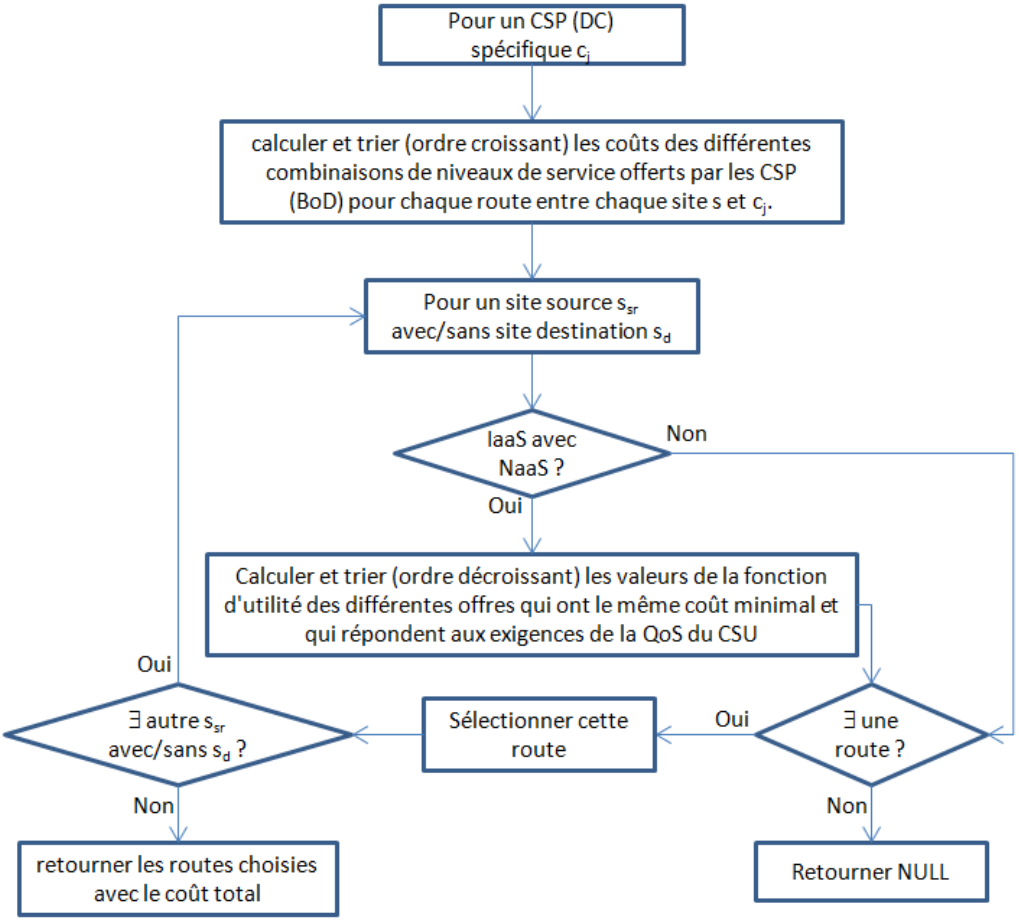

FIGURE 25 – Représentation schématisée de l'algorithme 2.

utilise alors la fonction d'utilité avec seulement les paramètres de QoS de type NaaS. Enfin, lorsque le Cloud Broker ou le CSP_L avec/sans les autres CSP (DC) ne trouvent pas de routes qui répondent aux exigences du CSU, ils renvoient une valeur vide ($Null$). Cependant, si tous les sites ont des routes, ils renvoient les routes choisies avec le coût total. Il est à noter que l'algorithme 2 est utilisé par la suite par les algorithmes 3 et 4 (cf. section 5.3.2.2 et section 5.3.2.3). De plus, la communication entre le site source et le site destination est bidirectionnelle, et le site source peut être lui-même le site destination ($s_{sr} \equiv s_d$).

5.3.2.2/ ALGORITHME ET CONTRAINTES POUR LA SÉLECTION DES RESSOURCES DE TYPE VM

Dans le cas d'une demande de service de type IaaS correspondant à des ressources de type VM, le CSU spécifie les paramètres de QoS de type IaaS (temps de réponse et/ou disponibilité) à garantir pour ses applications. De plus, il spécifie les longueurs des différentes Jobs et le nombre de VM requis. Ensuite, le Cloud Broker dans l'architecture Broker ou le CSP_L seulement (scénarios 1 ou 2 spécifiés dans la section 4.4.3) ou avec d'autres CSP (DC) (scénario 3 spécifié dans la section 4.4.3) dans l'architecture Fédération sélectionnent les meilleurs CSP (DC et BoD) en termes de coût minimal et de

Algorithm 2 Sélection des meilleures routes entre les sites (S) du CSU et un CSP (DC) spécifique (c_j) [$Get\_suitable\_Sites\_routes(S, c_j)$]

1: **for** each $s \in S$ **do**
2: $R_{sc} \leftarrow Get\_routes\_between(s, c_j)$
3: **for** each $r_k \in R_{sc}$ **do**
4: **for** each $sl_{xk} \in SL_k$ **do**
5: $T_3[s].Enqueue(r_k, sl_{xk}, cost_{xk})$
6: **end for**
7: **end for**
8: $T_3[s].sorting\_ascending\_order(cost)$
9: **end for**
10: **for** each s_{sr} and/or $s_d \in S$ **do**
11: **if** $IaaS\ without\ NaaS$ **then**
12: **if** $Site/DC$ **then**
13: $T.Enqueue(T_3[s_{sr}].Dequeue(), 0)$ ▷ l'offre avec le coût minimal, 0 : (valeur réservée pour la fonction d'utilité) signifie qu'il n'y a pas de garantie de QoS de type NaaS
14: **else if** $Site/Site$ **then**
15: $\{r_{k1}, sl_{xk1}, cost_{xk1}\} \leftarrow T_3[s_{sr}].Dequeue()$ ▷ c.à.d. l'offre avec le coût minimal
16: $\{r_{k2}, sl_{xk2}, cost_{xk2}\} \leftarrow T_3[s_d].Dequeue()$
17: $T.Enqueue(r_{k1} \cdot r_{k2}, sl_{xk1} \cdot sl_{xk2}, cost_{xk1} + cost_{xk2}, 0)$ ▷ nous utilisons le symbole . comme une concaténation
18: **end if**
19: **else if** $IaaS\ with\ NaaS$ **then**
20: $flag_1 = 0,\ flag_2 = 0$
21: **if** $Site/DC$ **then**
22: **while** $T_3[s_{sr}] \neq \emptyset$ && $flag_1 == 0$ **do**
23: $\{r_k, sl_{xk}, cost_{xk}\} \leftarrow T_3[s_{sr}].Dequeue()$
24: **if** Constraints (12) to (16) are met with r_k, sl_{xk} **then**
25: $flag_1 = 1$
26: $T.Enqueue(r_k, sl_{xk}, cost_{xk}, f(sl_{xk}))$ ▷ calcul de la fonction d'utilité
27: **while** $T_3[s_{sr}] \neq \emptyset$ && $flag_2 == 0$ **do**
28: $\{r_j, sl_{xj}, cost_{xj}\} \leftarrow T_3[s_{sr}].Dequeue()$
29: **if** $cost_{xk} == cost_{xj}$ **then**
30: **if** Constraints (12) to (16) are met with r_j, sl_{xj} **then**
31: $T.Enqueue(r_j, sl_{xj}, cost_{xj}, f(sl_{xj}))$
32: **end if**
33: **else**
34: $flag_2 = 1$
35: **end if**
36: **end while**
37: $T.sorting\_descending\_order(f(sl))$
38: **end if**
39: **end while**
40: **else if** $Site/Site$ **then**
41: $min_c = T_3[s_{sr}].get\_max\_cost() + T_3[s_d].get\_max\_cost()$ ▷ initialisation de la variable min_c

```
42:             while T_3[s_sr] ≠ ∅ do
43:                 T_1 ← T_3[s_d]
44:                 {r_k1, sl_xk1, cost_xk1} ← T_3[s_sr].Dequeue()
45:                 while T_1 ≠ ∅ && flag_1 == 0 do
46:                     {r_k2, sl_xk2, cost_xk2} ← T_1.Dequeue()
47:                     if Constraints (7) to (11) are met with r_k1 and r_k2 then
48:                         flag_1 = 1
49:                         if min_c == cost_xk1 + cost_xk2 then
50:                             T.Enqueue(r_k1 · r_k2, sl_xk1 · sl_xk2, min_c, f(sl_xk1 · sl_xk2))
51:                             while T_1 ≠ ∅ && flag_2 == 0 do
52:                                 {r_j, sl_xj, cost_xj} ← T_1.Dequeue()
53:                                 if cost_xk1 == cost_xj then
54:                                     if Constraints (7) to (11) are met with r_k1 and r_j then
55:                                         T.Enqueue(r_k1 · r_j, sl_xk1 · sl_xj, min_c, f(sl_xk1 · sl_xj))
56:                                     end if
57:                                 else
58:                                     flag_2 = 1
59:                                 end if
60:                             end while
61:                         else if min_c > cost_xk1 + cost_xk2 then
62:                             T ← ∅
63:                             min_c = cost_xk1 + cost_xk2          ▷ mise à jour du coût minimal
64:                             T.Enqueue(r_k1 · r_k2, sl_xk1 · sl_xk2, min_c, f(sl_xk1 · sl_xk2))
65:                             while T_1 ≠ ∅ && flag_2 == 0 do
66:                                 {r_j, sl_xj, cost_xj} ← T_1.Dequeue()
67:                                 if cost_xk1 == cost_xj then
68:                                     if Constraints (7) to (11) are met with r_k1 and r_j then
69:                                         T.Enqueue(r_k1 · r_j, sl_xk1 · sl_xj, min_c, f(sl_xk1 · sl_xj))
70:                                     end if
71:                                 else
72:                                     flag_2 = 1
73:                                 end if
74:                             end while
75:                         end if
76:                     end if
77:                 end while
78:             end while
79:             T.sorting_descending_order(f(sl))
80:         end if
81:     end if
82:     if T ≠ ∅ then
83:         {r_k, sl_xk, cost_xk, f(sl)} ← T.Dequeue()
84:         Cost_t = Cost_t + cost_xk
85:         Q.Enqueue(s_sr, s_d, r_k, sl_xk, cost_xk)
86:     else
87:         return ∅
88:     end if
89: end for
90: Q.Enqueue(Cost_t)                                  ▷ ajout du coût total
91: return Q
```

garantie de QoS en utilisant l'algorithme 3 que nous proposons pour l'optimisation et la sélection des meilleurs ressources de type VM. La sélection des ressources est soumise à des contraintes de qualité de service de type IaaS (voir les contraintes (17) et (18)).

$$RT(iSLA) \geq RT_{ij} \tag{17}$$

$$A_{IaaS}(iSLA) \leq A_{ij}(VM) \tag{18}$$

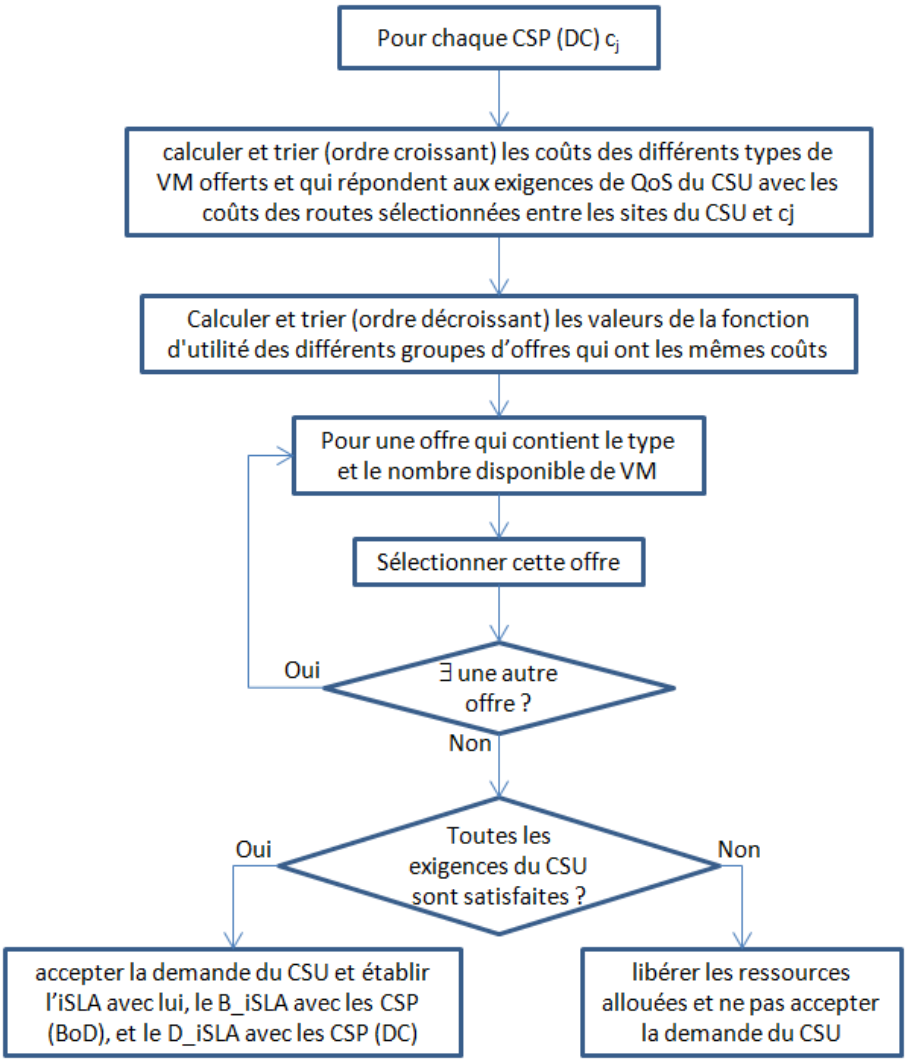

FIGURE 26 – Représentation schématisée de l'algorithme 3.

Ainsi, comme décrit dans la figure 26, l'algorithme 2 est utilisé par le Cloud Broker dans l'architecture Broker afin d'obtenir les meilleures routes entre les sites du CSU et chaque CSP (DC) c_j. Ce même algorithme 2 peut être aussi utilisé par le CSP_L seulement (scénario 2 spécifié dans la section 4.4.3) ou avec d'autres CSP (DC) (scénario 3 spécifié dans la section 4.4.3) dans l'architecture Fédération afin d'obtenir les meilleures routes entre les sites du CSU et eux-mêmes.

Ensuite, le Cloud Broker ou le CSP_L calcule et trie dans l'ordre croissant les coûts des différents types de VM qui répondent aux exigences du CSU (voir les contraintes (17)

et (18)) en tenant compte du coût des meilleures routes (voir algorithme 2). Puis, ils calculent et trient dans l'ordre décroissant les valeurs de la fonction d'utilité correspondantes aux différents groupes d'offres qui ont le même coût. Ainsi, le Broker ou le CSP_L choisit l'offre qui correspond au coût minimal avec la valeur maximale de la fonction d'utilité. L'algorithme 3 que nous proposons utilise la fonction d'utilité avec seulement les paramètres de QoS de type IaaS.

Enfin, si le Cloud Broker ou le CSP_L ne trouve pas des CSP qui répondent aux exigences du CSU, il libère les ressources allouées et n'accepte pas la demande du CSU. Cependant, si toutes les exigences du CSU sont satisfaites, le Cloud Broker ou le CSP_L attribue les meilleures ressources de type VM et les meilleures routes entre les sites et les VM (en utilisant l'algorithme 2). Ainsi, il accepte la demande du CSU et établit un contrat de type iSLA avec ce dernier, un contrat de type B_iSLA avec les CSP (BoD) sélectionnés, et un contrat de type D_iSLA avec les CSP (DC) sélectionnés.

Il est à noter que dans l'architecture Fédération, lorsque le CSP_L (DC) calcule ses ressources disponibles qui répondent aux exigences du CSU, l'ensemble C (l'ensemble des centres de données (DC) des CSP) utilisé dans l'algorithme 3 proposé ne contient que les centres de données du CSP_L (DC). Cependant, lorsque le CSP_L (DC) calcule les ressources dans les autres CSP (DC), C contient tous les centres de données de ces CSP (DC) sans les centres de données du CSP_L (DC).

Algorithm 3 Sélection des CSP et allocation des ressources de type VM

for each $c_j \in C$ **do**
  $T \leftarrow Get\_suitable\_Sites\_routes(S, c_j)$ ▷ algorithme 2
  if $T \neq \emptyset$ **then**
    for each $Jlu_d \in IVM$ **do**
      for each $vt_{ij} \in VT_j$ **do**
        $RT_{ij} = Get\_appropriate\_RT(Jlu_d, VCa_{ij})$
        if Constraints (17) and (18) are met **then**
          $Q.Enqueue(c_j, T, Jlu_d, vt_{ij}, VM\_cost_{ij})$
        end if
      end for
    end for
  end if
end for
$Q.sorting\_ascending\_order(VM\_cost + Cost_t)$
$\{c_j, T, Jlu_d, vt_{ij}, VM\_cost_{ij}\} \leftarrow Q.Dequeue()$ ▷ l'offre avec le coût minimal
$T_1.Enqueue(c_j, T, Jlu_d, vt_{ij}, VM\_cost_{ij}, f(sl))$ ▷ calcul de la fonction d'utilité
while $Q \neq \emptyset$ **do**
  $flag = 0$
  while $Q \neq \emptyset \;\&\&\; flag == 0$ **do** ▷ groupes d'offres qui ont les mêmes coûts
    $\{c'_j, T', Jlu'_d, vt'_{ij}, VM\_cost'_{ij}\} \leftarrow Q.Dequeue()$ ▷ l'offre suivante
    if $VM\_cost_{ij} + Cost_t == VM\_cost'_{ij} + Cost'_t$ **then**
      $T_1.Enqueue(c'_j, T', Jlu'_d, vt'_{ij}, VM\_cost'_{ij}, f(sl))$
    else
      $flag = 1$
    end if
  end while

$T_1.sorting\_descending\_order(f(sl))$
$Q_1.Enqueue(T_1)$
if $Q \neq \emptyset$ **then**
$T_1 \leftarrow \emptyset$
$T_1.Enqueue(c'_j, T', Jlu'_d, vt'_{ij}, VM\_cost'_{ij}, f(sl))$
$VM\_cost_{ij} = VM\_cost'_{ij}$
$Cost_t = Cost'_t$
end if
end while ▷ Réserver les VM et les routes qui répondent aux exigences du CSU
while $Q_1 \neq \emptyset$ and $\exists IVM[Jlu_x] \neq 0, \forall 1 \leq x \leq D$ **do**
$\{c_j, T, Jlu_d, vt_{ij}, VM\_cost_{ij}, f(sl)\} \leftarrow Q_1.Dequeue()$
while $nbVc_{ij} \neq 0$ and $IVM[Jlu_d] \neq 0$ **do**
allocate vt_{ij} in c_j with VM_cost_{ij}
$nbVc_{ij}--$, $IVM[Jlu_d]--$
if $c_j \notin T_2$ **then**
$T_2.Enqueue(c_j)$
while $T \neq \emptyset$ **do**
$\{s_{sr}, s_d, r_k, sl_{xk}, cost_{xk}\} \leftarrow T.Dequeue()$
allocate r_k with sl_{xk} and $cost_{xk}$ between s_{sr}, c_j and s_d
end while
end if
end while
end while
if $\exists IVM[Jlu_x] \neq 0, \forall 1 \leq x \leq D$ **then** ▷ il existe des exigences du CSU non satisfaites
release resources and reject CSU request
else
accept CSU request
end if

5.3.2.3/ ALGORITHME ET CONTRAINTES POUR LA SÉLECTION DES RESSOURCES DE STOCKAGE

Dans le cas d'une demande de service de type IaaS correspondant à des ressources de stockage, le CSU spécifie le paramètre de QoS de type IaaS à savoir Disponibilité nécessaire pour ses applications. De plus, il spécifie la capacité de stockage à utiliser. Ensuite, le Cloud Broker dans l'architecture Broker ou le CSP_L seulement (scénarios 2 spécifié dans la section 4.4.3) ou avec d'autres CSP (DC) (scénario 3 spécifié dans la section 4.4.3) dans l'architecture Fédération sélectionnent les meilleurs CSP (DC et BoD) en termes de coût minimal et de garantie de QoS en utilisant l'algorithme 4 que nous proposons pour l'optimisation et la sélection des meilleurs ressources de stockage. La sélection de ces ressources est soumise à des contraintes de qualité de service de type IaaS (voir les contraintes (19) et (20)).

$$SCa_u(iSLA) \leq SCa_{ij} \tag{19}$$

$$A_{IaaS}(iSLA) \leq A_{ij}(ST) \tag{20}$$

Ainsi, comme nous le représentons dans la figure 27, l'algorithme 2 est utilisé au début par le Cloud Broker dans l'architecture Broker afin d'obtenir les meilleures routes entre

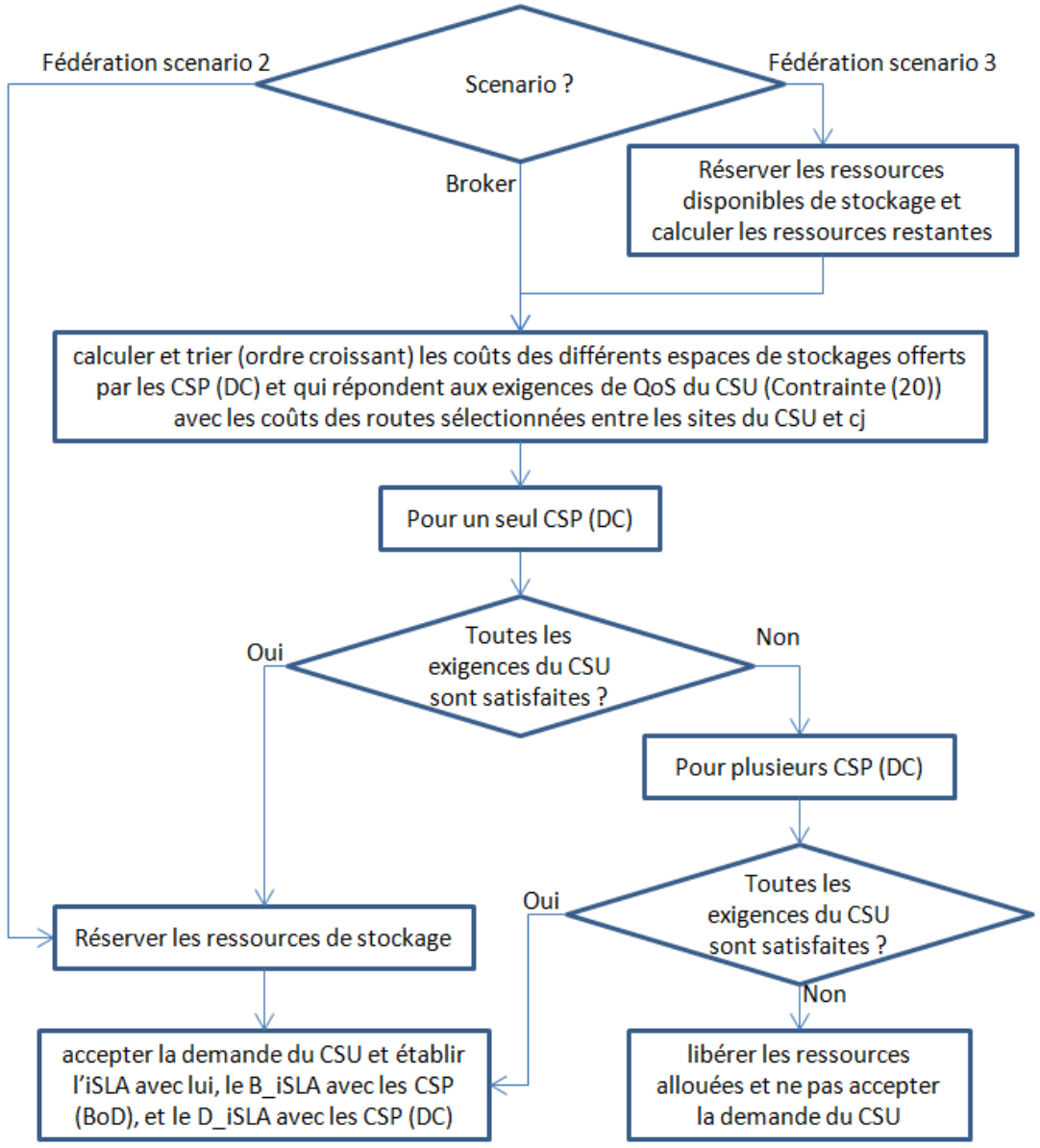

FIGURE 27 – Représentation schématisée de l'algorithme 4.

les sites du CSU et chaque CSP (DC) c_j. Cet algorithme 2 est aussi utilisé par le CSP_L seulement (voir scénario 2 décrit dans la section 4.4.3) ou avec d'autres CSP (DC) (voir scénario 3 décrit dans la section 4.4.3) dans l'architecture Fédération afin d'obtenir les meilleures routes entre les sites du CSU et eux-mêmes.

Ensuite, dans l'architecture Fédération, si le CSP_L (DC) seulement (scénario 2) peut répondre à toutes les exigences de stockage du CSU, alors il réserve ses ressources. Cependant, dans le scénario 3 de l'architecture Fédération, le CSP_L (DC) réserve ses ressources disponibles et demande des ressources supplémentaires à d'autres CSP (DC) dans la fédération. De même, dans l'architecture Broker, le Cloud Broker demande les ressources disponibles dans les autres CSP (DC). Ainsi, le Cloud Broker ou le CSP_L (DC) calcule et trie dans l'ordre croissant les coûts des capacités de stockage et les routes qui répondent aux exigences de QoS du CSU correspondant à la contrainte (20). Dans ce

cas, si le Cloud Broker ou le CSP_L (DC) trouve des CSP (DC) tel que chacun seul peut répondre à toutes les exigences de stockage du CSU pour la contrainte (19), alors il sélectionne celui qui possède le meilleur coût et la meilleure valeur de la fonction d'utilité s'il y a plusieurs offres de même coût minimal. Cependant, si la contrainte (19) n'est pas satisfaite pour un seul CSP (DC), alors le Cloud Broker ou le CSP_L (DC) cherche les CSP (DC) qui peuvent ensemble répondre à toutes les exigences de stockage du CSU pour la contrainte (19) tout en proposant le meilleur coût.

Enfin, si le Cloud Broker ou le CSP_L ne trouve pas les CSP qui répondent aux exigences du CSU, il libère les ressources allouées et n'accepte pas la demande du CSU. Cependant, si toutes les exigences du CSU sont satisfaites, le Cloud Broker ou le CSP_L attribue les meilleures ressources de stockage et les meilleures routes entre les sites du CSU et ces ressources (en utilisant l'algorithme 2). Par conséquent, il accepte la demande du CSU et établit un contrat de type iSLA avec lui, un contrat de type B_iSLA avec les CSP (BoD) sélectionnés, et un contrat de type D_iSLA avec les CSP (DC) sélectionnés.

Il est à noter que dans l'architecture Fédération, lorsque le CSP_L (DC) calcule ses ressources disponibles qui répondent aux exigences du CSU, l'ensemble C (l'ensemble des centres de données (DC) des CSP) utilisé dans l'algorithme 4 proposé ne contient que les centres de données du CSP_L (DC). Cependant, lorsque le CSP_L (DC) calcule les ressources dans les autres CSP (DC), C contient tous les centres de données de ces CSP (DC) sans les centres de données du CSP_L (DC).

Algorithm 4 Sélection des CSP et allocation des ressources de stockage

for each $c_j \in C$ **do**
  $T \leftarrow Get\_suitable\_Sites\_routes(S, c_j)$ ▷ algorithme 2
  if $T \neq \emptyset$ **then**
    for each $st_{ij} \in ST_j$ **do**
      if Constraints (20) is met **then**
        if Constraints (19) is met **then**
          $Q.Enqueue(c_j, T, st_{ij}, st\_cost_{ij})$
        else
          $Q_1.Enqueue(c_j, T, st_{ij}, st\_cost_{ij})$
        end if
      end if
    end for
  end if
end for
if $Q \neq \emptyset$ **then**
  $Q_2 \leftarrow Q$
else
  $Q_2 \leftarrow Q_1$
end if
$Q_2.sorting\_descending\_order(st\_cost + route\_Cost_t)$
$\{c_j, T, st_{ij}, st\_cost_{ij}\} \leftarrow Q_2.Dequeue()$ ▷ l'offre avec le coût minimal
$T_1.Enqueue(c_j, T, st_{ij}, SCa_{ij}, st\_cost_{ij}, f(sl))$ ▷ calcul de la fonction d'utilité du niveau de service (sl) selon la disponibilité
$flag_1 = 0$

```
24: while Q2 ≠ ∅ && flag1 == 0 do
25:     flag = 0
26:     while Q2 ≠ ∅ && flag == 0 do
27:         {c'j, T', st'ij, st_cost'ij} ← Q2.Dequeue()          ▷ l'offre suivante
28:         if st_costij + route_Costt == st_cost'ij + route_Cost't then
29:             T1.Enqueue(c'j, T', st'ij, SCa'ij, st_cost'ij, f(sl))
30:         else
31:             flag = 1
32:         end if
33:     end while
34:     T1.sorting_descending_order(f(sl))
35:     while SCau ≠ 0 or T1 ≠ ∅ do
36:         {cj, T, stij, SCaij, st_costij, f(sl)} ← T1.Dequeue()
37:         if cj ∉ T2 then
38:             T2.Enqueue(cj)
39:             while T ≠ ∅ do
40:                 {ssr, sd, rk, slxk, costxk} ← T.Dequeue()
41:                 allocate rk with slxk and costxk between ssr, cj and sd
42:             end while
43:         end if
44:         if SCau ≤ SCaij then
45:             allocate SCau in cj with st_costij
46:             SCau = 0
47:         else
48:             SCau = SCau − SCaij
49:             allocate SCaij in cj with st_costij
50:         end if
51:     end while
52:     if SCau == 0 then
53:         flag1 = 1
54:     end if
55: end while
56: if SCau ≠ 0 then
57:     release resources and reject CSU request
58: else
59:     accept CSU request
60: end if
```

5.4/ Évaluation du coût relatif à l'offre de QoS

En se basant sur la caractéristique du Cloud «payez ce que vous utilisez», le CSU remplace le coût de l'utilisation de sa propre infrastructure par un coût faible et variable selon l'utilisation des services Cloud car il ne paie que pour ce qu'il utilise. Dans cette section, nous présentons une méthodologie générale pour calculer les coûts de l'utilisation des services Cloud de type IaaS et NaaS. Elle est similaire à la méthode utilisée par Amazon [45] pour faire payer ses clients. Cependant, Amazon ne considère pas séparément les ressources réseau ni tous les paramètres de qualité de service que nous spécifions dans nos contrat de niveau de service. Seule la Disponibilité est considérée dans leur

évaluation du coût. Par conséquent, nous proposons de calculer le coût en se basant sur l'équation (21) :

$$Cost_{total} = Cost_{BW} + Cost_{VM} + Cost_{ST} \tag{21}$$

Tel que, $Cost_{total}$ est le coût total de la consommation des différentes ressources sélectionnées par le Cloud Broker ou le CSP_L et utilisées par le CSU. Le CSU est facturée en \$. De plus, $Cost_{BW}$ est le coût total des ressources réseau calculé en se basant sur l'équation (22), $Cost_{VM}$ est le coût total des ressources de type VM calculé en se basant sur l'équation (23), et $Cost_{ST}$ est le coût total des ressources de stockage calculé en se basant sur l'équation (24).

Il est à noter que chaque CSP possède des coûts différents pour les différents niveaux de services de ressources, et que ces coûts sont envoyés d'abord au Cloud Broker ou le CSP_L sous la forme d'un coût unitaire. Les détails des ces coûts sont décrits dans les sections suivantes.

5.4.1/ Coût relatif au service de type NaaS

Pour les ressources réseau, le CSU paie sur une base de quantité de données transférées ($\$ \, par \, gigaoctet$). L'équation (22) spécifie la manière de calculer le coût total de l'utilisation des ressources réseau par le CSU.

$$Cost_{BW} = \sum_{j} (BW\_cost_j \times BW_j) \tag{22}$$

Tel que, BW_cost_j est le coût unitaire ($\$ \, par \, GB$) du trafic qui traverse un CSP_j (BoD ou DC) sélectionné avec un garantie d'un niveau de QoS, et BW_j (GB) est la bande passante consommée par le trafic du CSU qui traverse CSP_j.

5.4.2/ Coût relatif au service IaaS de type VM

Pour les ressources de type VM, le CSU paie sur une base horaire d'utilisation ($\$ \, par \, heure$) à partir du moment où il lance l'utilisation d'une ressource jusqu'au moment où il termine cette utilisation. Quand il n'a pas besoin de ses ressources, le CSU peut les désactiver et arrêter de payer. L'équation (23) spécifie la manière de calculer le coût total de l'utilisation des machines virtuelles par la CSU.

$$Cost_{VM} = \sum_{j} \sum_{i} (VM\_cost_{ij} \times t_{ij}) \tag{23}$$

Tel que, VM_cost_{ij} est le coût unitaire d'utilisation ($\$ \, par \, heure$) d'un type de VM sélectionné (vt_{ij}) dans un CSP_j (DC) sélectionné, et t_{ij} ($heure$) est le nombre d'heures de consommation de vt_{ij}.

Il est à noter que nous considérons que le coût de la capacité de mémoire propre pour chaque VM est inclus dans VM_cost_{ij}. Cependant, si le CSU veut allouer des ressources de stockage, il est facturé séparément comme décrit dans la section suivante.

5.4.3/ Coût relatif au service IaaS de type Stockage

Pour le stockage des données, le CSU paie sur une base de durée et de quantité de données stockées ($\$ \, par\, GB\, par\, heure$). L'équation (24) spécifie la manière de calculer le coût total de l'utilisation des espaces de stockage par la CSU.

$$Cost_{ST} = \sum_{j} \sum_{i} (st\_cost_{ij} \times SCau_{ij} \times t_{ij}) \tag{24}$$

Tel que, st_cost_{ij} est le coût unitaire ($\$ \, par\, GB\, par\, heure$) d'un type d'espace de stockage sélectionné (st_{ij}) dans un CSP_j (DC) sélectionné, $SCau_{ij}$ est la capacité de stockage utilisée par le CSU pour le type st_{ij} (GB), et t_{ij} ($heure$) est le nombre d'heures de consommation de st_{ij}.

5.5/ Validation de notre proposition de garantie de QoS dans un environnement de Cloud Networking

Afin de bénéficier des garanties assurées par notre proposition d'architecture de Cloud Networking et démontrer les avantages de ce type d'architecture, nous présentons dans cette section deux cas d'utilisation pour l'établissement efficace des applications de Cloud vidéoconférence et des applications de calcul intensif. Nous testons les scénarios correspondants comme une preuve de concept qui nous permet de valider nos architectures de type Broker et Fédération. Ainsi, nous évaluons les performances de notre proposition en effectuant un ensemble de simulations correspondant à des scénarios différents grâce à la boîte à outils CloudSim [33].

En effet, dans un système distribué comme le Cloud, il est difficile d'effectuer différents scénarios avec un nombre de ressources et d'utilisateurs variables afin d'évaluer les performances des architectures et des algorithmes proposés. Lorsque nous voulons évaluer des scénarios d'utilisation de manière répétitive et contrôlable, cela peut parfois induire des coûts énormes et une gestion complexe pour prendre en considérations toutes nos propositions en termes d'ajout d'entités et exécution d'algorithmes. Afin de remédier à ces difficultés, nous utilisons le simulateur CloudSim pour évaluer des scénarios d'utilisation avant de les déployer au sein d'un système distribué réel.

5.5.1/ Environnement de simulation

CloudSim supporte la modélisation et la simulation de l'environnement de centre de données (DC) du Cloud, tels que des interfaces de gestion dédiées aux VM, la mémoire, le stockage, la bande passante, les politiques de provisionnement des ressources etc. Ces interfaces peuvent être étendues. La couche basse de CloudSim gère l'instanciation et l'exécution des entités de base (VM, hôtes, applications, etc.) au cours de la période de simulation. Dans la couche plus haute, nous retrouvons le code de l'utilisateur qui expose la configuration des fonctionnalités liées aux hôtes (nombre de machines, leurs spécifications, etc.), les politiques d'ordonnancement des applications (nombre de tâches et leurs besoins, etc.), nombre d'utilisateurs, etc. Actuellement, CloudSim supporte la simulation

d'environnements de Cloud qui sont constitués d'un seul Cloud ou plusieurs Clouds interconnectés. De plus, il supporte la simulation des connexions réseau entre les éléments du système de simulation, et la simulation de l'environnement de fédération du Cloud.

Ainsi, nous étendons CloudSim pour prendre en charge trois nouvelles entités que nous avons spécifiées dans nos propositions d'architecture pour la garantie de QoS dans un environnement de Cloud Networking. La première est une entité de type CSU et la deuxième est une entité de type Cloud Broker, qui remplace l'entité «CloudSim DatacenterBroker», permettant au CSU de lui envoyer ses exigences pour la sélection et l'allocation des meilleures ressources et l'établissement des SLA. Enfin, la troisième entité que nous proposons pour étendre CloudSim est une entité de type CSP (BoD) qui interconnecte le Cloud Broker, les CSP (DC) et les sites du CSU en utilisant une topologie de type BRITE [33] pour la modélisation de la bande passante des liens et des latences associées.

Il est à noter que le CSP_L dans l'architecture de type Fédération est simulé comme un CSP normal avec des fonctionnalités en plus comme la sélection des meilleurs CSP. De plus, nous simulons pour chaque entité (Cloud Broker, CSP_L, CSP (DC/BoD)) les algorithmes proposés pour la sélection des meilleures ressources comme décrit dans la section 5.3.

Le modèle simulé de notre architecture est composé d'un Cloud Broker, quatre CSP (BoD) et quatre CSP (DC). Chaque CSP (DC) contient 10 hôtes et chaque hôte dispose d'un processeur Quad-core ($4 \times 1,2GHz$), 16 Go de RAM, et 5 TB d'espace de stockage. Toutes les entités sont initiées au début de la simulation. Dans notre modèle de simulation, un CSP (DC) peut offrir les niveaux de services présentés dans la table 14 pour un service de type IaaS. Il offre ces niveaux de service pour les VM et les espaces de stockages.

	Capacité	A_{IaaS}	Coût
VM	1000 MHz	99.999%	0.1 $ par heure
	750 MHz	99.999%	0.2 $ par heure
	500 MHz	99.999%	0.3 $ par heure
	250 MHz	99.999%	0.4 $ par heure
Stockage	10 TB	99.999%	10^{-4} $ par GB par heure
	9 TB	97.999%	0.7×10^{-4} $ par GB par heure
	9 TB	95.999%	0.5×10^{-4} $ par GB par heure
	8 TB	93.999%	0.3×10^{-4} $ par GB par heure

TABLE 14 – Niveaux de service pour un service de type IaaS offerts par un CSP (DC).

De plus, dans la table 15, nous présentons les niveaux de services offerts par un CSP (BoD) ou un CSP (DC) pour un service de type NaaS. Chaque ligne de chaque table correspond à un niveau de service (Platinum, Gold, Silver, Bronze).

Il est à noter que les définitions des variables utilisées dans les tables 14 et 15 sont spécifiées dans la table 13.

La CSU spécifie le type de service demandé et les exigences correspondantes, en termes de qualité de service, au Cloud Broker ou au CSP_L en utilisant l'interface graphique proposée. Ensuite, après la sélection des meilleurs CSP, le Cloud Broker ou le CSP_L alloue les ressources nécessaires aux sites du CSU et établit les différents SLA. Les entités

	L	J	P	BW	A_{NaaS}	Coût
CSP (DC)	3 ms	0.6 ms	10^{-3}	15 Mb/s	99.999%	0.11 $ par GB
	7 ms	1 ms	3.5×10^{-3}	10 Mb/s	99.999%	0.05 $ par GB
	13 ms	1.5 ms	8.5×10^{-3}	5 Mb/s	99.999%	0.03 $ par GB
	22 ms	2 ms	10^{-2}	1 Mb/s	99.999%	0.01 $ par GB
CSP (BoD)	6 ms	1 ms	0.5×10^{-3}	15 Mb/s	99.999%	0.3 $ par GB
	12 ms	2 ms	2.5×10^{-3}	10 Mb/s	99.999%	0.2 $ par GB
	25 ms	4 ms	7.5×10^{-3}	5 Mb/s	99.999%	0.15 $ par GB
	50 ms	5 ms	10^{-2}	1 Mb/s	99.999%	0.1 $ par GB

TABLE 15 – Niveaux de service pour un service de type NaaS offerts par un CSP (DC/BoD).

participantes lancent les applications et quand elles arrêtent l'exécution de ces applications, le Cloud Broker ou le CSP_L libère les ressources allouées et le CSU paie pour ce qu'il a utilisé comme ressources.

5.5.2/ CLOUD VIDÉOCONFÉRENCE

La vidéoconférence multi-partie peut être l'une des applications multimédia les plus exigeantes en termes de bande passante, de gigue et de délai de bout en bout [106]. Un système de vidéoconférence permet un échange de données multimédia en temps réel entre plusieurs parties, ce qui dépasse les limites de la communication humaine en raison de l'emplacement géographique des participants. De plus, le transcodage des flux de données est nécessaire pour encoder la vidéo et la voix avec le codec correspondant.

5.5.2.1/ SCÉNARIO DE SIMULATION

Pour évaluer la performance de notre architecture de Cloud Networking tout en considérant le service de vidéoconférence, nous implémentons nos algorithmes de sélection des CSP pour valider le scénario de la vidéoconférence dans les architectures de type Broker et Fédération. Ces algorithmes déterminent les meilleures routes de distribution des flux de vidéoconférence en passant par des machines virtuelles allouées pour implémenter les serveurs de vidéoconférence. Il est possible que les machines virtuelles soient distribuées dans différents centres de données. L'objectif du Cloud Broker ou du CSP_L est de minimiser le coût total du service sans violer les contraintes de bout-en-bout de QoS.

Ainsi, le CSU choisit un service de type IaaS (VM sans stockage) et de type NaaS. Nous évaluons trois scénarios de simulation de vidéoconférence. Le premier correspond à une sélection statique des ressources sans garantie de QoS. Le deuxième et le troisième scénario correspondent respectivement à une sélection basée sur l'architecture Broker et l'architecture Fédération avec garantie de QoS. Dans chaque scénario, plusieurs sites du CSU sont connectés à différentes CSP (BoD) et chaque site peut avoir de multiples destinations. De plus, pour chaque site source le CSU demande une VM pour le transcodage. Ensuite, à chaque simulation, le CSP_L dans le troisième scénario (Fédération) et les sites destinations dans tous les scénarios sont sélectionnés d'une façon aléatoire.

Nous évaluons pour ces scénarios, le délai moyen global et la gigue moyenne globale depuis la source jusqu'à la destination. D'une manière formelle, le délai est défini comme la période de temps nécessaire à un paquet de données pour se déplacer d'un point initial à un autre final. Cependant, la gigue est définie comme la variation de ce délai, c.à.d. la valeur absolue de la différence entre les temps d'arrivées de deux paquets successifs et leurs temps de départ. Dans ces simulations, pour un site source (s_{sr}) et un site destination (s_d), un type de VM (vt_{ij}), et un CSP (DC) (c_j), le délai (D) est calculé en utilisant l'équation (25) et la gigue (G) est calculée en utilisant l'équation (26).

$$D = \sum_{i1=1}^{zk_1} L(b_{i1}) + L(c_j) + RT_{ij} + L(c_j) + \sum_{i2=1}^{zk_2} L(b_{i2}) \tag{25}$$

Tel que $b_{i1} \in r_{k1}$, $b_{i2} \in r_{k2}$ et r_{k1} est la route entre s_{sr} et c_j tandis que r_{k2} est la route entre c_j et s_d, et RT_{ij} est le temps de réponse du vt_{ij}.

$$D = |D_{i+1} - D_i| \tag{26}$$

Tel que D_{i+1} et D_i sont les délais de deux paquets consécutifs.

Le délai et la gigue sont des paramètres de qualité de services très importants pour les applications interactives temps réel telles que la vidéoconférence. Un délai de bout en bout supérieur à 200 ms ou une gigue supérieure à 30 ms peut provoquer une dégradation du service de vidéoconférence. Donc, dans les scénarios de Broker et de Fédération, le CSU spécifie une latence réseau maximale de 180 ms, une gigue réseau maximale de 30 ms et un temps de réponse maximal de transcodage de 20 ms. Pour les autres paramètres de qualité de service, le CSU demande un niveau de service Gold et tous les poids sont égaux à 1.

Nous simulons quatre types de vidéos envoyées par le CSU avec une longueur d'une heure, une taille de 1, 2, 3 et 4 Go, respectivement, et une bande passante demandée égale à 2.2Mb/s, 4.5Mb/s, 6.8Mb/s et 9.1Mb/s, respectivement.

Après plusieurs simulations pour chaque type de vidéo avec une distribution géographique différente des sites du CSU, nous calculons au début le délai moyen global incluant le temps de réponse et la latence du réseau. Ensuite, nous calculons la gigue moyenne globale. Enfin, nous calculons les coûts globaux de la bande passante et des VM.

5.5.2.2/ RÉSULTATS

Les résultats illustrés dans les figures 28 et 29, montrent un bon délai global obtenu grâce à une sélection de CSP basée sur notre proposition d'architecture de type Broker ou Fédération, par rapport à une sélection statique (voir figure 30). Ainsi, pour tous les types de vidéos dans l'architecture Broker et l'architecture Fédération, les délais globaux de bout-en-bout sont bien contrôlés grâce à une minimisation des délais de mise en file d'attente lors de la traversée des différents CSP qui garantissent leurs parts dans le niveau de service global à assurer pour les flux de vidéoconférence.

De plus, la gigue globale (voir figure 31) est bien contrôlée dans les architectures avec garantie de QoS que nous proposons et ne dépasse pas le seuil maximal toléré pour ce

type d'applications qui est de 30 ms.

D'autre part, comme le montre la figure 32, le coût global de la bande passante augmente lorsque la taille de la vidéo et la bande passante correspondante augmente en raison des contraintes de QoS de type NaaS. De même, comme le montre la figure 33, le coût global de VM augmente lorsque la taille de la vidéo augmente en raison des contraintes de QoS de type IaaS pour nos architectures de type Broker et Fédération.

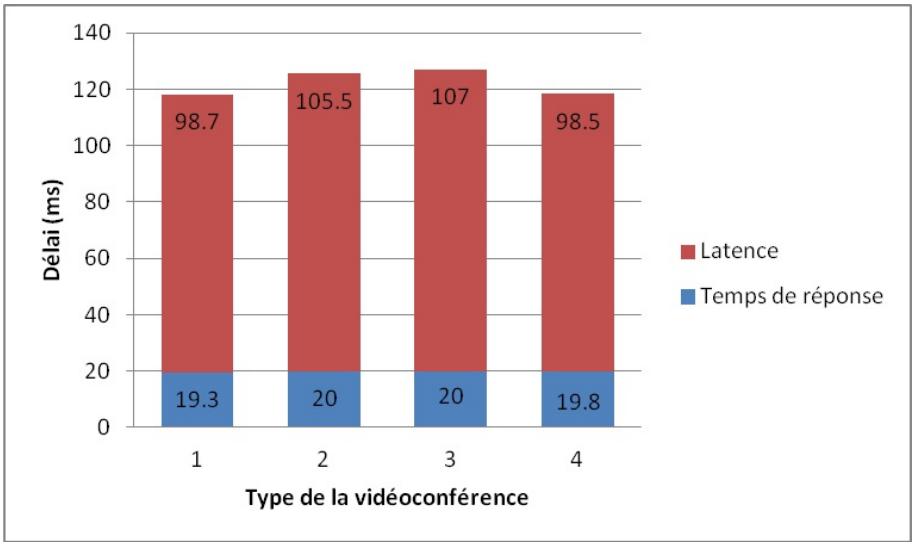

FIGURE 28 – Délai moyen global de bout en bout dans une architecture de type Broker.

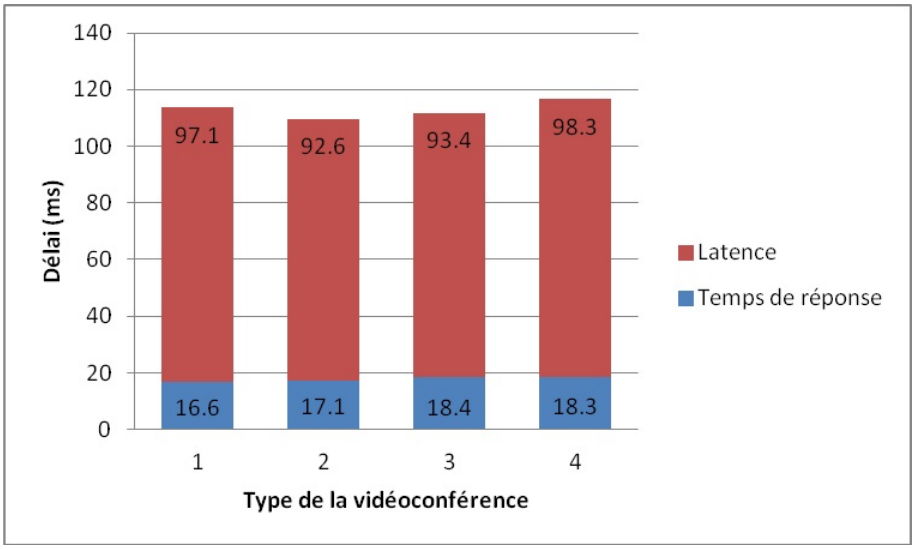

FIGURE 29 – Délai moyen global de bout en bout dans une architecture de type Fédération.

De plus, dans une sélection statique, le coût global de la bande passante est inférieur à ceux de la sélection basée sur les algorithmes que nous proposons dans l'architecture Broker et l'architecture Fédération (voir figure 32). Ceci s'explique par la garantie de QoS de type NaaS qui doit être assurée par le Cloud Broker ou le CSP_L. De même, dans une

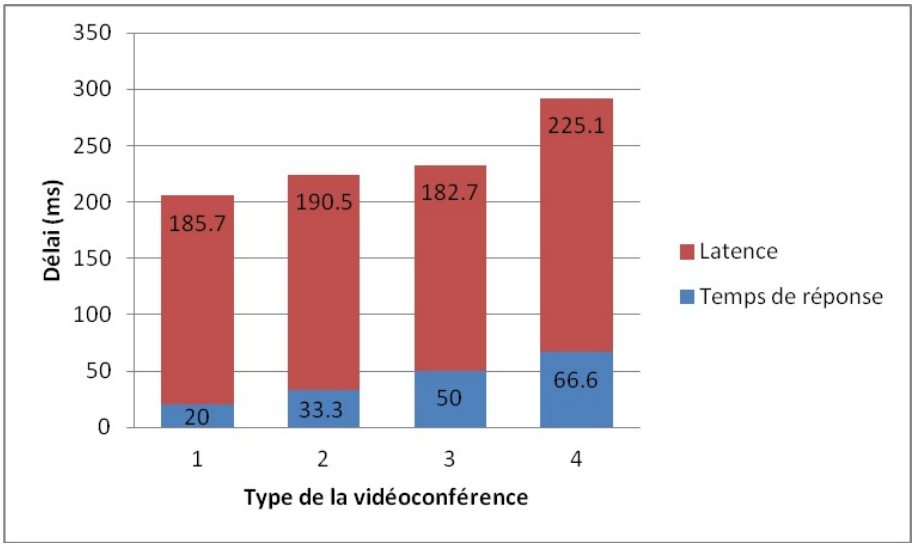

FIGURE 30 – Délai moyen global de bout en bout pour une sélection statique.

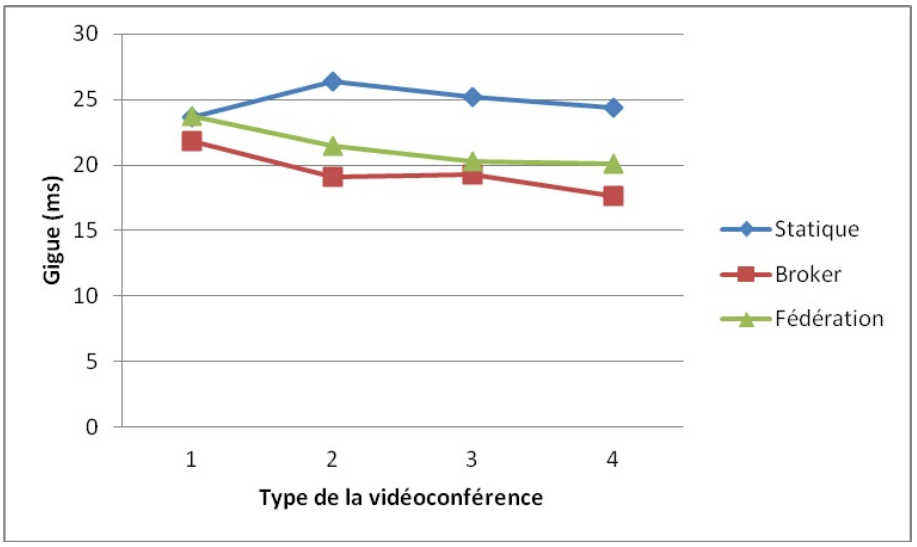

FIGURE 31 – Comparaison de la gigue entre les trois scénarios.

sélection statique, le coût global de VM est inférieur à ceux de la sélection proposée dans l'architecture de type Broker et Fédération (voir figure 33) en raison de la garantie de QoS de type IaaS qui doit être assurée par le Cloud Broker ou le CSP_L.

Enfin, nous pouvons remarquer que le coût global de la bande passante dans le scénario de vidéoconférence dans l'architecture Broker est moins important que le coût de la bande passante dans le scénario de l'architecture Fédération (voir figure 32). De même, le coût global de VM dans le scénario de l'architecture Broker est moins important que le coût de VM dans le scénario de l'architecture Fédération (voir figure 33). Ces résultats d'expliquent par le fait qu'un CSP_L commence d'abord par sélectionner ses propres ressources disponibles, avant de faire appel au besoin à des ressources au sein d'autres CSP. En effet, les ressources du CSP_L ne sont pas forcément les meilleures en termes

de coût. Ainsi, nous pouvons conclure que l'architecture de type Broker est la plus économique, tout en assurant les exigences de qualité de service du CSU relatives à son application temps réel.

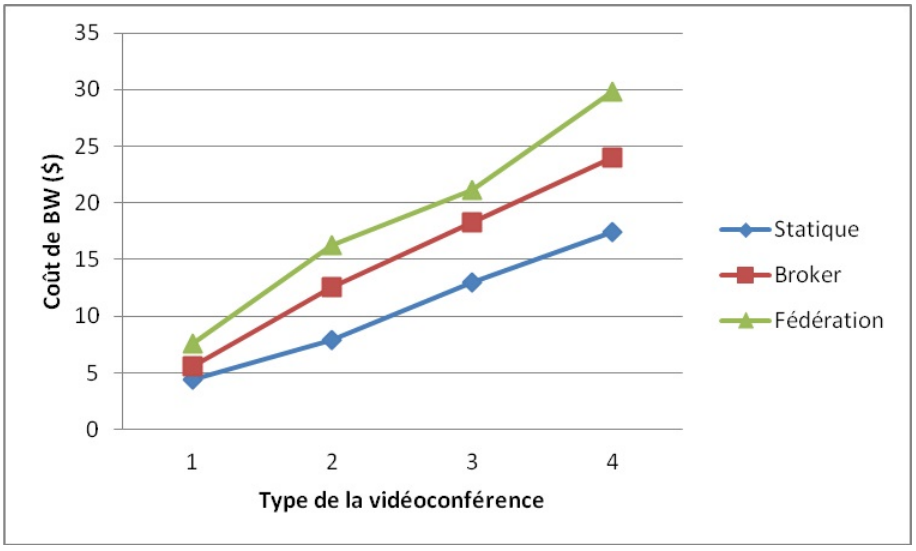

FIGURE 32 – Comparaison du coût global de la bande passante entre les trois scénarios.

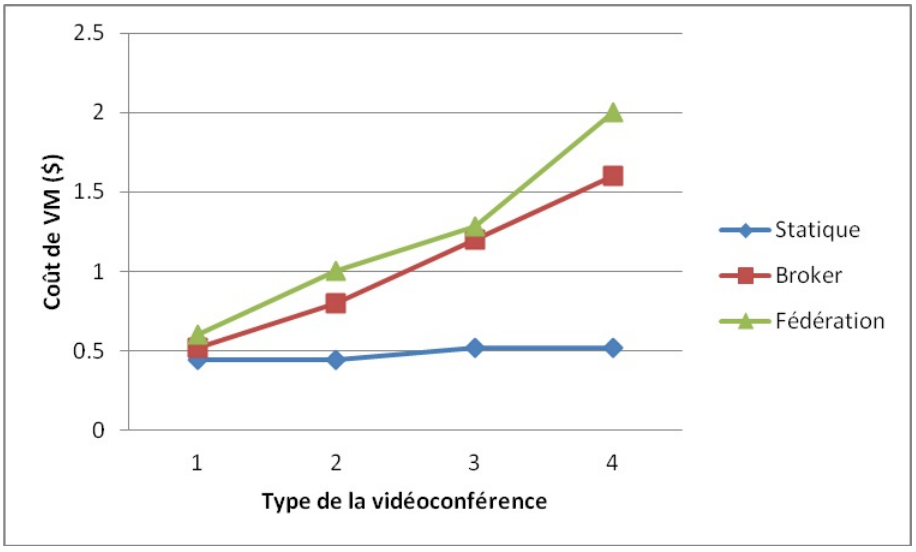

FIGURE 33 – Comparaison du coût global de VM entre les trois scénarios.

5.5.3/ CALCULS INTENSIFS

Les services du Cloud peuvent être utilisés pour le traitement d'énormes quantités d'informations telles que les applications de calcul intensif qui sont basées sur la demande d'exécution de tâches de calculs scientifiques par exemple [184]. En effet, la vision de l'e-science est que les scientifiques répartis puissent collaborer ensemble pour la réalisation des d'expériences scientifiques à grande échelle et d'applications de découverte

de connaissances à l'aide de systèmes distribués de ressources informatiques (le Cloud).

5.5.3.1/ Scénario de simulation

Dans les scénarios de simulation de ce type d'applications, nous implémentons dans CloudSim nos algorithmes de sélection des meilleurs CSP pour valider ce cas d'utilisation dans les architectures de Cloud Networking de type Broker et Fédération. Le but de ces algorithmes proposés est de déterminer les meilleures machines virtuelles à allouer pour les Jobs demandés, tout en minimisant le coût total de l'application sans violer les contraintes de QoS correspondantes. Les exigences de service du CSU sont simulées comme une série de demandes de Jobs à exécuter par les machines virtuelles allouées.

Nous évaluons trois scénarios de simulation de calculs intensifs. Dans le premier scénario, le CSU choisit un service de type IaaS sans NaaS et un temps de réponse maximal de 300 ms. Dans le deuxième scénario, le CSU choisit un service de type IaaS avec NaaS, un temps de réponse maximal de 300 ms, une latence maximale de 150 ms, et une bande passante de 1Mb/s. Pour les autres paramètres de qualité de service, le CSU demande un niveau de service Silver et tous les poids sont égaux à 1. Dans le troisième scénario, le CSU choisit une sélection statique sans garantie de QoS.

Après plusieurs exécutions des scénarios de simulation avec quatre sites CSU où chacun envoie 100 Jobs à quatre VM, nous calculons pour les différentes longueurs de Jobs (200, 250, 300 et 350 instructions) le délai moyen global (temps de réponse moyen et latence moyenne), ainsi que les coûts globaux de la bande passante et des VM.

5.5.3.2/ Résultats

Comme illustré dans les figures 34 et 35, en fournissant seulement une garantie de QoS de type IaaS lors de la sélection de CSP grâce aux algorithmes correspondants que nous déployons dans l'architecture Broker et l'architecture Fédération, le temps de réponse global est bien contrôlé par rapport à une sélection statique (voir figure 38), mais la latence n'est pas contrôlée. Cependant, comme le montre les figures 36 et 37, en fournissant une garantie de QoS de type IaaS et NaaS lors de la sélection des CSP basée sur les algorithmes correspondants dans l'architecture Broker et l'architecture Fédération, le temps de réponse et la latence sont bien contrôlés contrairement aux résultats obtenus dans le scénario de sélection statique (voir figure 38).

D'autre part, en termes de coûts relatifs au service demandé, les résultats reportés dans la figure 39 montrent que le coût global de la bande passante augmente lorsque la longueur du Job augmente. De plus, le coût global de la bande passante sans le service de type NaaS (c.à.d. sélection statique, IaaS sans NaaS pour l'architecture Broker (B-IaaS seulement) et l'architecture Fédération (F-IaaS seulement)) est inférieur au coût global de la bande passante dans l'architecture Broker (B-IaaS avec NaaS) et l'architecture Fédération (F-IaaS avec NaaS) lorsque le service de type NaaS est sollicité, et ce en raison de la garantie de QoS de type NaaS que le Cloud Broker ou encore le CSP_L assurent dans nos propositions d'architectures de Cloud Networking. De plus, nous observons grâce aux résultats obtenus (voir figure 39) que le coût global de la bande passante dans le cas d'une sélection des routes avec un service de type IaaS sans NaaS est inférieur au coût global de la bande passante dans la sélection statique. En effet, l'algorithme de

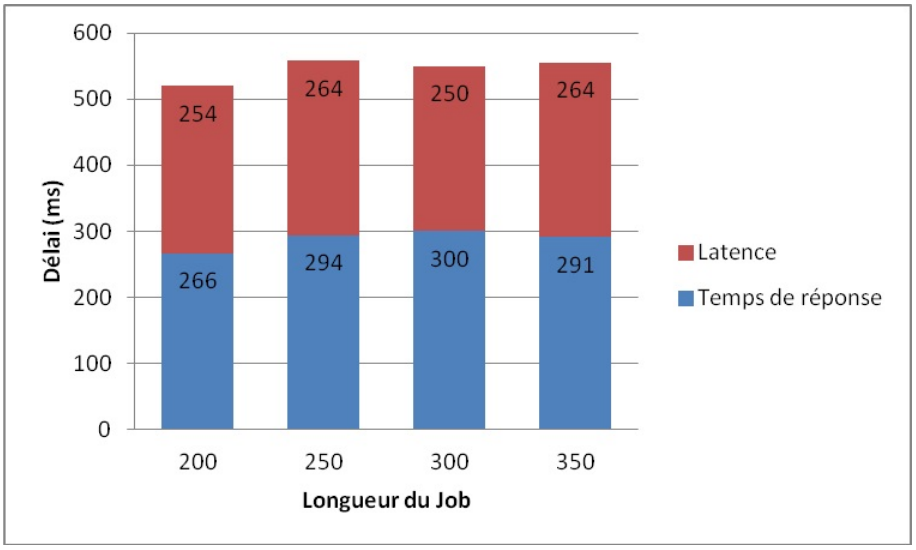

FIGURE 34 – Délai moyen global pour un service IaaS sans NaaS dans l'architecture Broker.

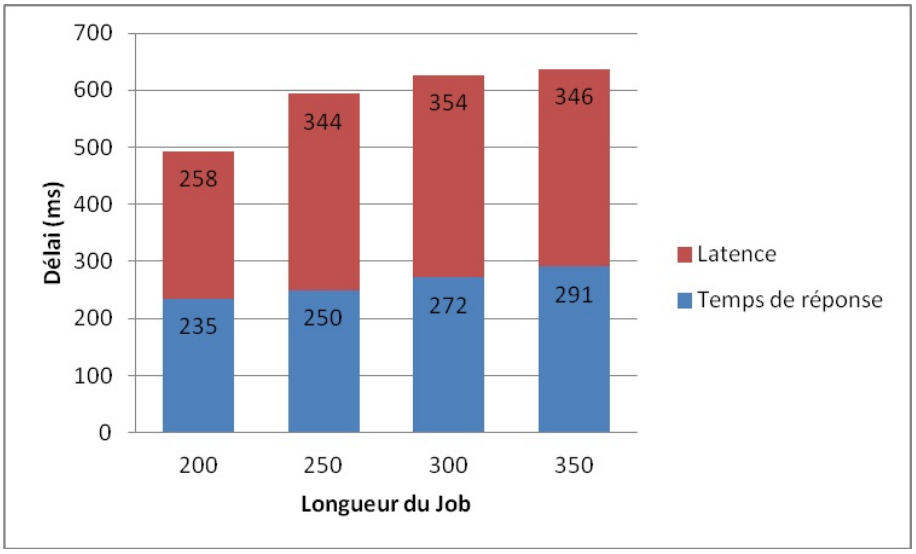

FIGURE 35 – Délai moyen global pour un service IaaS sans NaaS dans l'architecture Fédération.

sélection que nous proposons permet de choisir la route qui a le coût minimal dans l'architecture Broker et l'architecture Fédération si le CSU demande un service de type IaaS sans NaaS.

En ce qui concerne les ressources de type VM, la figure 40 montre que le coût global des VM augmente lorsque la longueur des Jobs augmente étant donné la contrainte du temps de réponse spécifié par le CSU. De plus, nous remarquons que le coût global des VM dans la sélection statique est inférieur au coût des VM choisies avec notre algorithme de sélection dans le cas d'un service de type IaaS avec/sans NaaS pour les deux architectures Broker et Fédération. En effet, contrairement à une sélection statique, nous

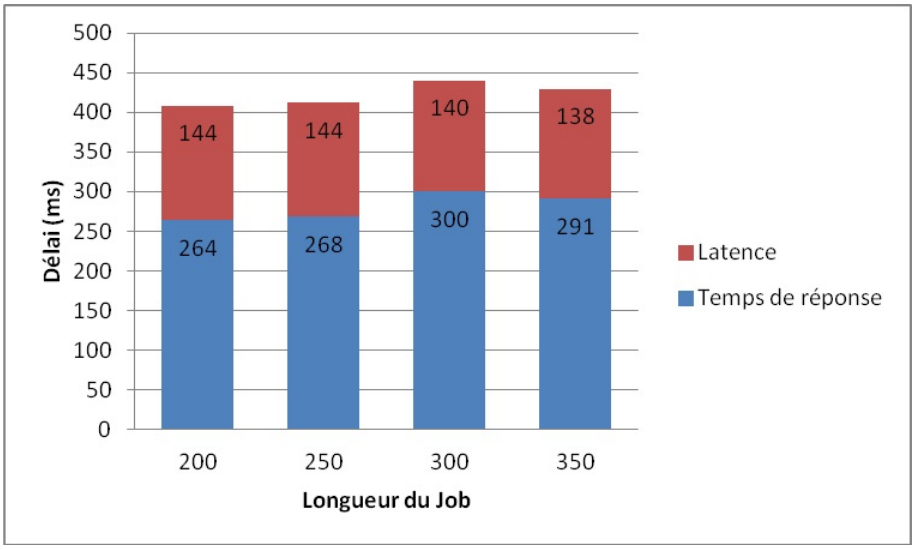

FIGURE 36 – Délai moyen global pour un service IaaS avec NaaS dans l'architecture Broker.

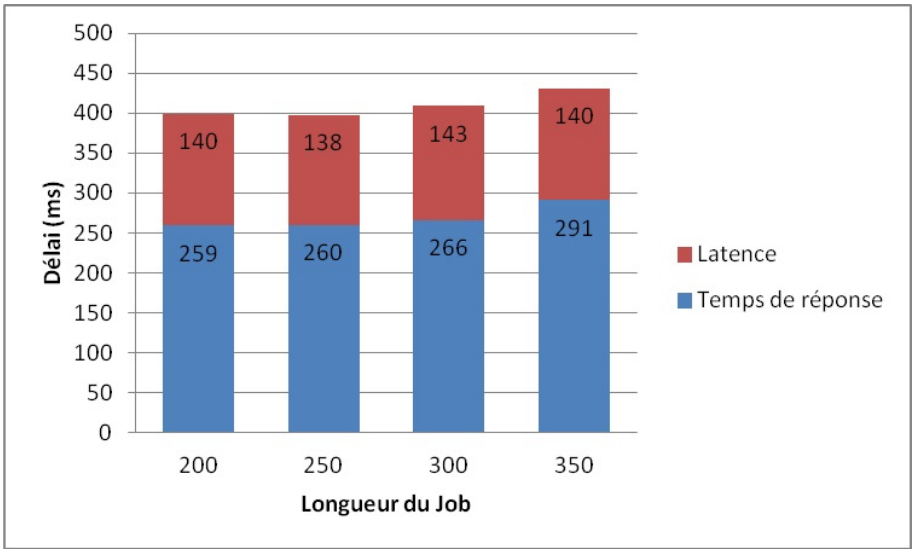

FIGURE 37 – Délai moyen global pour un service IaaS avec NaaS dans l'architecture Fédération.

permettons grâce notre algorithme de sélection de prendre en considération la garantie de QoS de type IaaS par le Cloud Broker ou encore le CSP_L afin de respecter les exigences du CSU. Ceci s'explique par choix de VM plus performantes et par conséquent un coût global plus important.

Enfin, les résultats obtenus dans les figures 39 et 40 montrent que le coût global de la bande passante dans le scénario de l'architecture Broker est moins important que le coût de la bande passante dans le scénario de l'architecture Fédération. De même, le coût global des VM dans le scénario de l'architecture Broker est moins important que le coût des VM dans le scénario de l'architecture Fédération. Ces résultats s'expliquent par le fait

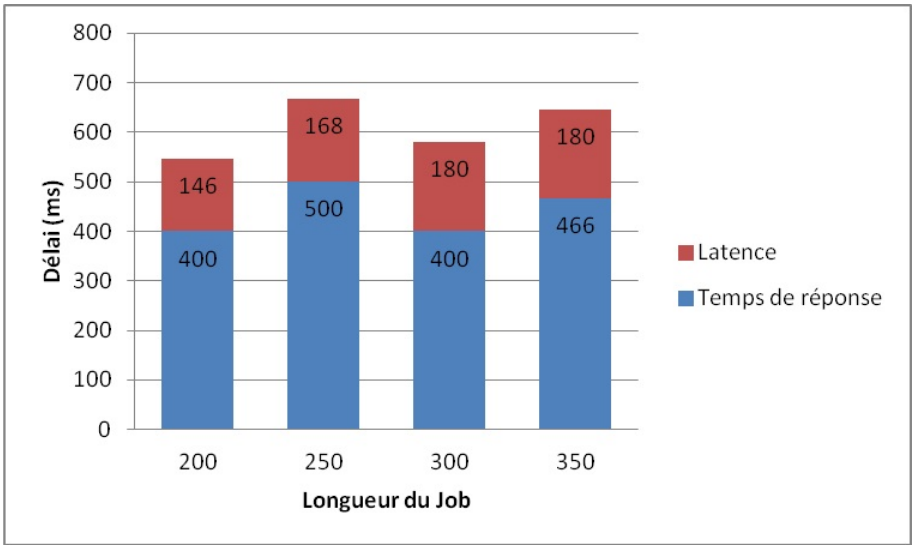

FIGURE 38 – Délai moyen global pour une sélection statique sans garantie de QoS.

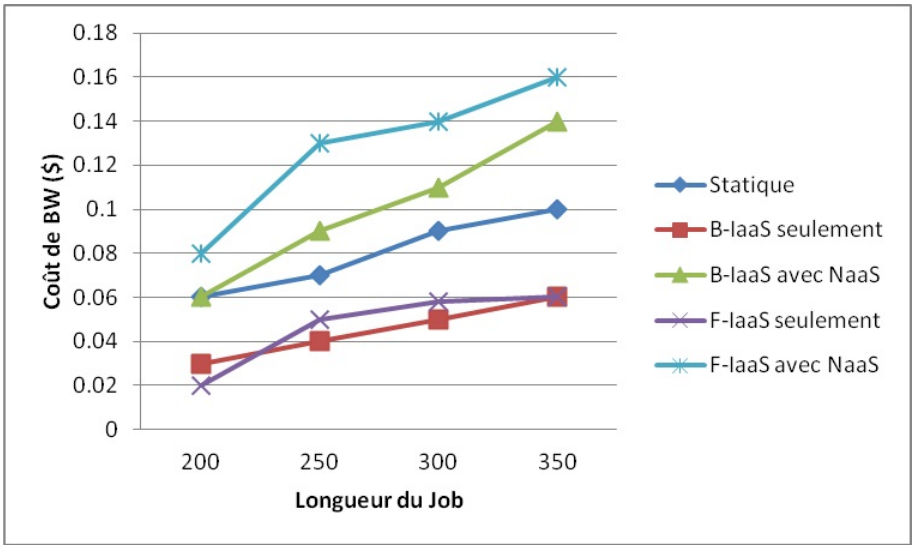

FIGURE 39 – Comparaison du coût global de la bande passante.

que la sélection des ressources disponibles au sein du CSP_L est prioritaire par rapport à la sélection des ressources au sein des autres CSP dans une architecture de type Fédération. Par conséquent, nous pouvons affirmer que l'architecture Broker est la plus économique des architectures de Cloud Networking proposées, lors de l'offre de service de type IaaS avec ou sans service de type NaaS. Cette offre de service se fait avec une garantie de QoS qui permet de satisfaire les exigences spécifiées par le CSU.

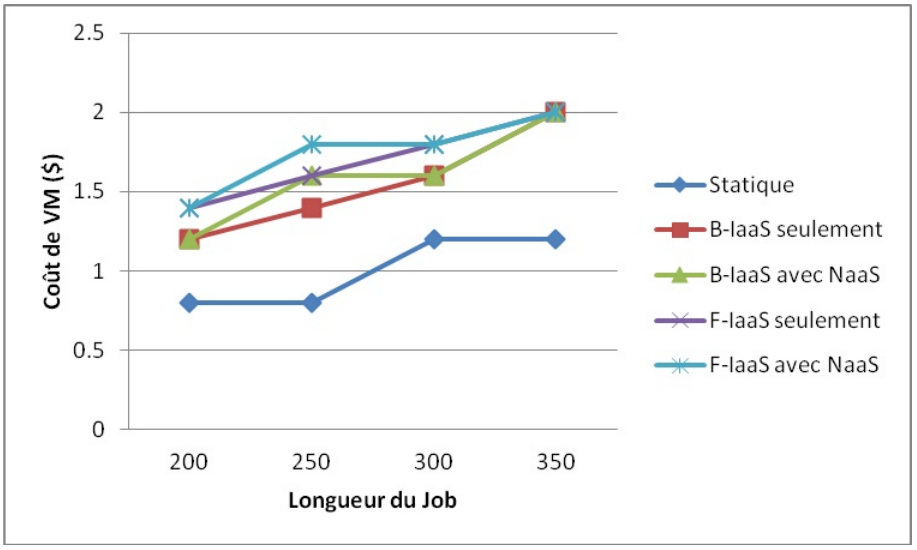

FIGURE 40 – Comparaison du coût global des VM.

5.6/ CONCLUSION

Dans ce chapitre, nous avons spécifié des algorithmes de sélection pour déterminer les meilleurs CSP permettant de répondre aux exigences du CSU. Ainsi, nous avons proposé un premier algorithme pour sélectionner les ressources réseau relatives à un service de type NaaS uniquement. De plus, nous avons proposé trois autres algorithmes pour la sélection des ressources réseau, de VM, et de stockage, pour un service de type IaaS avec/sans NaaS. Le but était d'assurer un service de type IaaS et/ou NaaS avec un coût minimal tout en garantissant la qualité de service demandée par le CSU et en respectant les contraintes relatives à cette QoS. Ainsi, si nous avions plusieurs offres équivalentes, nous avons proposé l'utilisation d'une fonction d'utilité en se basant sur la QoS offerte par les CSP et demandée par le CSU afin sélectionner la meilleure offre. De plus, nous avons proposé une méthodologie pour calculer le coût pour chaque type de service (NaaS et IaaS (VM et stockage)). Enfin, nous avons simulé nos architectures de type Broker et Fédération, et implémenté les algorithmes de sélection proposés afin d'assurer les services de types NaaS et IaaS pour deux types d'applications, à savoir la vidéoconférence et le calcul intensif. Nous avons obtenu de bons résultats de performances qui valident notre proposition et montrent que l'architecture de Broker est la plus économique par rapport à l'architecture Fédération lors de l'offre des services NaaS et/ou IaaS tout en garantissant les exigences de qualité de service du CSU. Cependant, l'augmentation du nombre de CSP (BoD) et des sites du CSU dans l'architecture de Cloud Networking pourrait produire des problèmes de passage à l'échelle. Ainsi, nous pouvons avoir un temps de calcul important pour la sélection des routes et des combinaisons de niveaux de services dans l'algorithme 1 et l'algorithme 2. Ces traitements demandent une capacité de calcul importante qui est heureusement disponible dans un environnement de Cloud Computing.

Dans le chapitre suivant, nous présenterons un Framework pour la gestion autonome de ces types d'architectures de Cloud Networking. Ce Framework permettra l'établissement autonome des SLA proposés dans le chapitre 4 ainsi que la détection des violations et le calcul des pénalités correspondantes.

CHAPITRE 6
GESTION AUTONOME DES ARCHITECTURES DE CLOUD NETWORKING

6.1/ INTRODUCTION

Aujourd'hui, il est nécessaire qu'une infrastructure de Cloud supporte l'établissement autonome d'un SLA ainsi que la gestion autonome des ressources de ce Cloud, en raison des caractéristiques du Cloud concernant l'auto-organisation et l'affectation dynamique à la demande de ses ressources. De plus, la disponibilité d'un système fiable et autonome réduit la complexité de gestion des ressources et minimise les interactions avec l'utilisateur.

Dans ce chapitre, nous présentons une architecture que nous proposons pour l'établissement autonome des SLA spécifiés dans le chapitre 4 ainsi que la gestion autonome des ressources correspondantes en utilisant des gestionnaires autonomes spécifiques de Cloud. Ensuite, nous décrivons les différents automates et algorithmes proposés pour l'établissement autonome de ces SLA et pour la gestion autonome de ces ressources pour garantir un niveau de service de bout en bout dans l'architecture Broker et Fédération. Ainsi, nous spécifions par la suite les équations permettant la détection et le calcul des violations, des pénalités, et des réputations des différents CSP. Enfin, nous présentons l'environnement de simulation pour la validation de notre proposition de gestion autonome des SLA et des ressources du Cloud grâce à des scénarios de déploiement de deux types d'applications, à savoir la vidéoconférence et les calculs intensifs.

6.2/ PROBLÉMATIQUE

Dans le chapitre précédant, nous avons présenté la sélection des meilleurs CSP comme étant un problème d'optimisation avec contraintes multi-objectifs (QoS offerte, coût total, etc.). Dans ce chapitre, notre objectif est d'auto-établir les SLA et d'auto-gérer les ressources pour assurer le niveau de service demandé par le CSU. Dans ce contexte, face à des problèmes de congestion ou de charge excessive dans environnement de Coud, le système doit réagir pour éviter les violations ou alors pour calculer les pénalités en cas de violations. Par conséquent, dans le cas de pénalités, la réputation du CSP concerné est affectée.

Ainsi, dans le processus de sélection des meilleurs CSP, nous devons assurer une sélection faisable et optimale de ces CSP. D'une part, une sélection faisable signifie que les

valeurs de la QoS agrégées et assurées par les CSP sélectionnés répondent aux exigences de QoS du CSU. D'autre part, nous considérons comme sélection optimale une sélection faisable qui minimise la valeur globale des coûts. Cependant, si nous avons le même coût minimum pour des offres différentes, notre objectif devient de maximiser une nouvelle fonction d'utilité (voir l'équation (27)) calculée à partir de ces offres pour sélectionner la solution optimale. Cette fonction utilise la réputation des CSP et les différents poids normalisés «W_i» attribués dans l'interface graphique (GUI) par la CSU pour chaque *i-ème* paramètre de QoS. Cette attribution des poids est basée sur l'importance du paramètre de QoS pour un type d'application.

$$f(sl) = \sum_{j=1}^{nb} \frac{\psi_j}{nb} \times \sum_{i=1}^{q} (w_i \times \frac{|Ur_i - Uo_i|}{Ur_i}), \; avec \; \sum_{i=1}^{q} w_i = 1 \tag{27}$$

Ur_i est la valeur du *i-ème* paramètre de QoS demandé par le CSU et Uo_i est la valeur agrégée du *i-ème* paramètre de QoS pour une offre de niveau de service (sl) assurée par le CSP, tandis que q est le nombre de paramètres de QoS spécifiés. ψ_j est la valeur de la réputation du CSP$_j$ (voir section 6.5.3) qui offre un service dans ce sl, et nb est le nombre de CSP qui offrent des services dans ce sl. Par conséquent, étant donné que toutes les sélections sont faisables et vu l'utilisation de la valeur absolue, la valeur la plus élevée de la fonction d'utilité indique la meilleure offre. Ainsi, pour la sélection des meilleures offres, nous améliorons les algorithmes 1 à 4 proposés dans le chapitre 5 en intégrant l'utilisation de cette fonction d'utilité.

6.3/ Architecture autonome proposée

Dans cette section, nous proposons une architecture autonome de Cloud Networking afin d'auto-gérer les ressources Cloud et d'auto-établir les trois types de SLA proposés (iSLA, B_iSLA et D_iSLA) et ce grâce à l'utilisation de gestionnaires autonomes de Cloud (Autonomic cloud Manager (AM)).

6.3.1/ Présentation de l'architecture

En général, un domaine se réfère à une collection de ressources gérées par une seule entité. Dans un environnement de Cloud, un domaine peut être un Centre de Données (DC) ou encore un réseau de communication d'un Cloud. Ainsi, lorsque nous proposons la gestion de ce domaine d'une manière autonome, nous l'appelons domaine autonome de Cloud (Autonomic cloud Domain (AD)). Dans ce contexte, nous présentons dans la figure 41 l'architecture proposée de Cloud Networking avec plusieurs domaines autonomes de Cloud (Cloud Broker, CSP (BoD) et CSP (DC)). Dans cette architecture, le Cloud Broker est présent uniquement dans le cas de la gestion autonome d'une architecture de type Broker.

Chaque domaine autonome de Cloud (AD) est sous l'autorité d'un gestionnaire Autonome d'inter-cloud que nous appelons inter-cloud Autonomic Manager (iAM). Un iAM offre des possibilités de communication de haut niveau avec d'autres iAM pour conclure un accord de niveau de service correspondant à des ressources sélectionnées en fonction du coût

optimal et de la garantie de qualité de service (auto-optimisation). De plus, le gestionnaire iAM contrôle un ou plusieurs gestionnaires autonomes (AM : Autonomic Manager) de bas niveau pour configurer les ressources allouées (auto-configuration) et pour éviter les violations (auto-restauration) en conformité avec le niveau de service convenu. Ces gestionnaires autonomes (AM) de bas niveau jouent des rôles différents au sein de notre architecture autonome de Cloud Networking. Ainsi, nous spécifions trois types de gestionnaires autonomes de bas niveau (voir figure 41) :

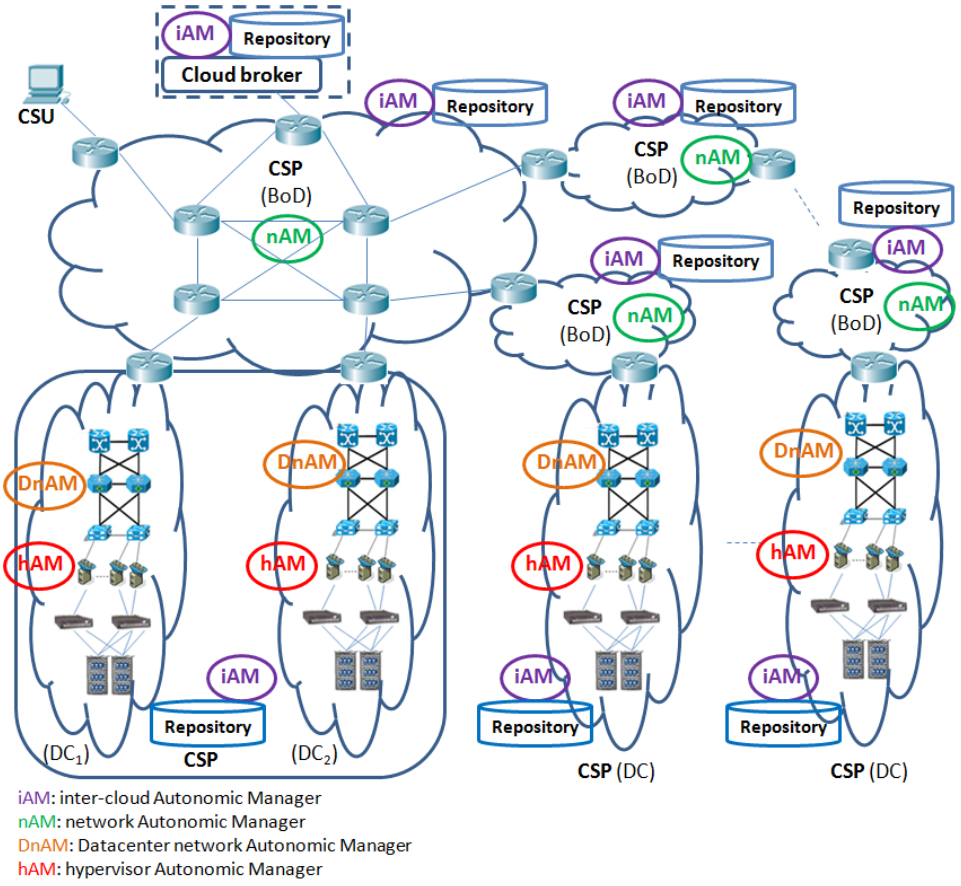

FIGURE 41 – Architecture Autonome de Cloud Networking.

— **network Autonomic Manager (nAM) :** ce gestionnaire autonome de réseau est responsable de la création et de la gestion des réseaux virtuels du CSU au sein du CSP (BoD) et de la surveillance des flux et de leurs performances en conformité avec le B_iSLA établi.
— **Datacenter network Autonomic Manager (DnAM) :** ce gestionnaire autonome de centre de données est responsable de la création et de la gestion des réseaux virtuels du CSU au sein du DC du CSP (DC), et de la surveillance des flux et de leurs performances en conformité avec la partie NaaS du D_iSLA.
— **hypervisor Autonomic Manager (hAM) :** ce gestionnaire autonome d'hyperviseur est responsable de la création et de la gestion des machines virtuelles et des ressources de stockage en conformité avec la partie IaaS du D_iSLA.

Dans notre architecture de gestion autonome, les CSP peuvent décider de l'attribution ou alors la libération des ressources pour maintenir un niveau de performance et d'utilisation

acceptable. De plus, nous spécifions pour les gestionnaires autonomes de haut niveau (iAM) la capacité de réaliser un accord entre les domaines autonomes de Cloud (AD). Cet accord porte sur la QoS à garantir pour les différents modèles de services Cloud tels que IaaS et NaaS. Ainsi, chaque iAM utilise un dépôt (Repository) pour stocker des informations relatives à la gestion des ressources et pour faciliter l'interaction avec d'autres gestionnaires autonomes (AM). Ce dépôt peut contenir des enregistrements de réputation pour les différents CSP dans l'environnement de Cloud Networking. Les enregistrements de réputation sont constitués d'un ensemble de scores de réputation générés par le Cloud Broker (dans le cas d'une architecture de type Broker) ou le CSP_L (dans le cas d'une architecture Fédération) en se basant sur les violations de QoS détectées grâce à une surveillance continue des CSP.

6.3.2/ Description du gestionnaire autonome proposé

Un Gestionnaire Autonome de Cloud (AM) doit connaître son environnement et tout mettre en œuvre afin de le conserver dans des conditions optimales d'utilisation, sans la nécessité d'une intervention externe. Dans ce contexte, les AM proposés (iAM, nAM, DnAM, ou hAM) peuvent gérer une ressource unique ou un ensemble de ressources (AM de bas niveau, machines virtuelles, espaces de stockage, ressources réseau, etc.) grâce à des interfaces de type capteurs et de type effecteurs (voir figure 42).

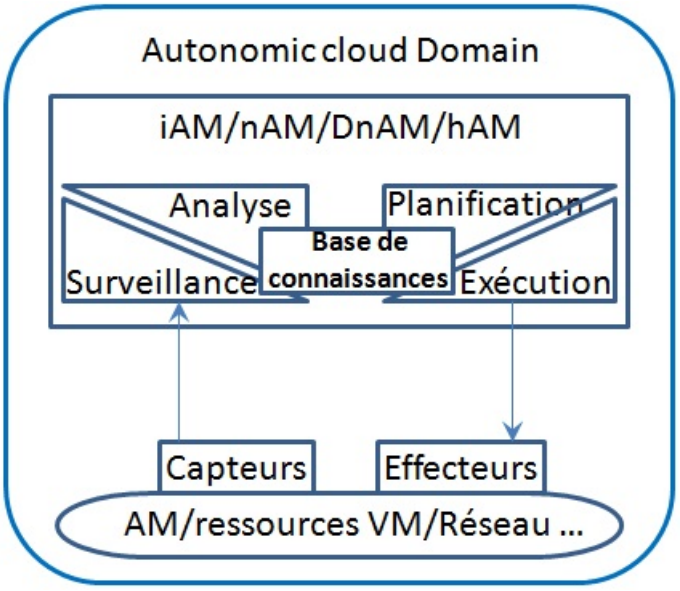

Figure 42 – Gestionnaire Autonome de Cloud (AM).

Après l'intégration des différentes politiques dans sa base de connaissances (seuils des paramètres de qualité de service, les algorithmes de sélection des meilleures ressources du Cloud, etc.), le gestionnaire autonome de Cloud (AM) commence avec la phase de surveillance qui peut concerner les valeurs des paramètres de qualité de service. Cette surveillance permet d'assurer la collection, l'agrégation, et le filtrage de données afin d'envoyer des rapports sur l'état des ressources gérées grâce aux interfaces de type capteurs. Ensuite, les données recueillies sont transmises pour la phase d'analyse afin de corréler ces données conformément aux politiques de la base de connaissances (vio-

lation des paramètres de QoS, la défaillance, la congestion, etc.). Ainsi, une demande de changement pourrait être envoyée pour la phase de planification afin d'indiquer les actions nécessaires pour atteindre des objectifs spécifiques conformément aux politiques spécifiées (allocation ou libération des ressources du Cloud, migration des machines virtuelles, etc.). Enfin, ces actions sont envoyées pour la phase d'exécution afin de permettre aux modifications d'être effectuées au niveau des ressources du Cloud et ce grâce aux interfaces de type effecteurs (configuration des ressources, migration des VM, etc.). De plus, les changements seront vérifiés pour mettre à jour la base de connaissances grâce à la phase de surveillance.

Ces phases constituent la boucle de contrôle fermée (MAPE-K) de la gestion autonome des ressources du Cloud telle que nous la spécifions dans les gestionnaires autonomes (AM) de notre architecture. Cette boucle de contrôle est conforme à celle proposée dans l'architecture IBM [185].

6.3.3/ Interactions entres les gestionnaires autonomes

Dans le cadre de notre architecture de Cloud Networking avec prise en compte de la gestion autonome des ressources tout en offrant une garantie de QoS de bout-en-bout, deux types d'interactions pourraient avoir lieu entre les gestionnaires autonomes de cloud (voir figure 43 pour le scénario Broker et figure 44 pour le scénario Fédération).

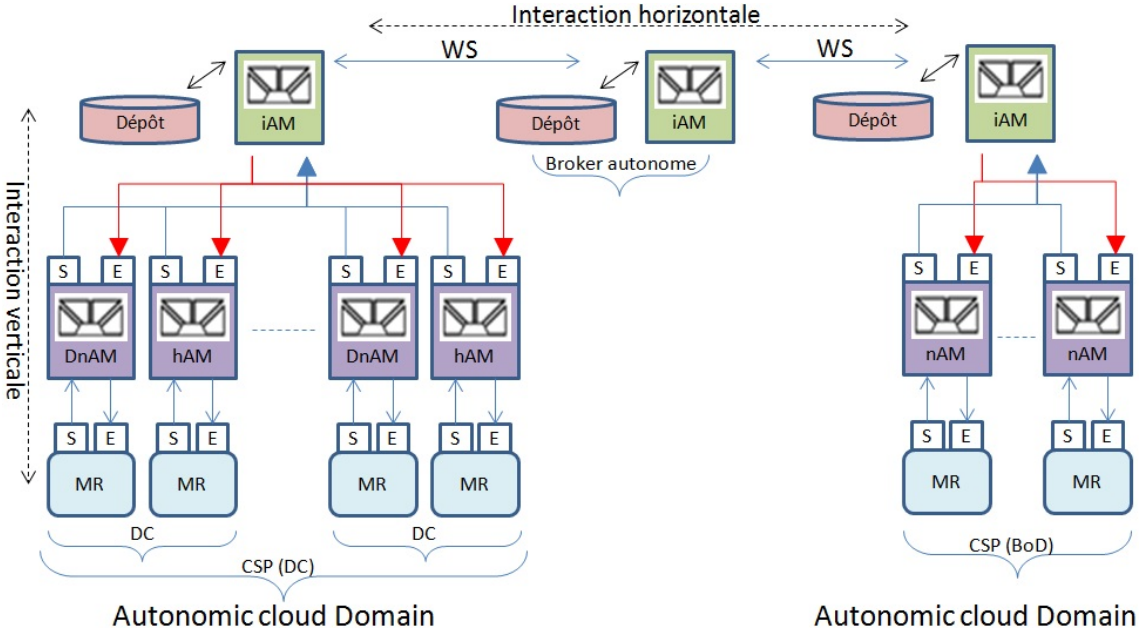

Figure 43 – Interactions des AM pour le scénario Broker.

Grâce à la première interaction, un gestionnaire autonome de haut niveau (iAM) initie un processus de communication pair à pair avec les autres iAM correspondants à l'aide d'une interaction horizontale en utilisant des technologies de Services Web (WS : Web Services) pour atteindre un accord sur un niveau de service. De plus, chaque iAM est responsable de la garantie du niveau de service au sein du domaine autonome de Cloud (AD) correspondant. Cette garantie est rendue possible grâce à un second type d'interaction. En effet, un iAM contrôle un ou plusieurs gestionnaires autonomes de Cloud de bas

niveau (nAM, DnAM ou hAM) grâce aux interfaces de gestion (effecteurs (E) et capteurs (S)) en utilisant les technologies de WS pour obtenir cette garantie de niveau de service. Ainsi, un iAM fournit aux gestionnaires autonomes de bas niveau (AM) le niveau de service correspondant (B_iSLA ou D_iSLA) dans une interaction verticale. Ces derniers utilisent une interaction verticale similaire pour allouer, libérer, ou modifier la configuration de leurs ressources gérées (Managed Resources (MR) : routeur, VM, espace stockage, etc.) en fonction du niveau de service reçu.

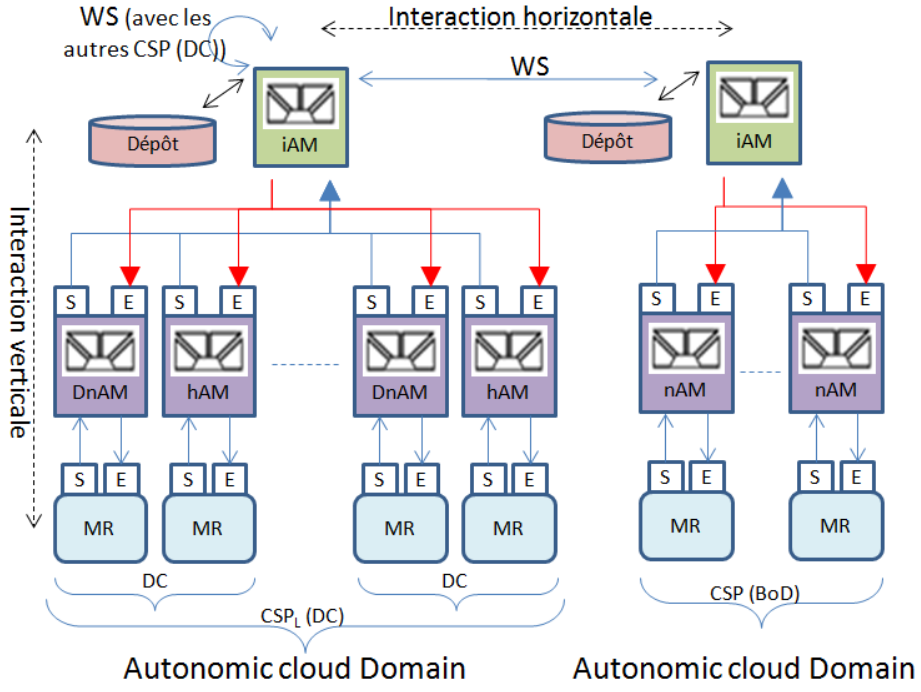

FIGURE 44 – Interactions des AM pour le scénario Fédération.

6.4/ GESTION AUTONOME DE L'OFFRE DE QOS DANS LE CLOUD

Dans cette section, nous décrivons le cycle de vie de chaque gestionnaire autonome de Cloud en se basant sur leurs interactions pour assurer un établissement autonome des SLA proposés ainsi que la gestion autonome des ressources correspondantes.

6.4.1/ GESTION AUTONOME DU SLA DANS L'ARCHITECTURE BROKER

Nous spécifions un automate fini (FSM : Finite State Machine) pour décrire le cycle de vie de l'établissement d'un SLA par un gestionnaire autonome de haut niveau (iAM Broker) dans notre architecture autonome de type Broker (voir figure 45). L'automate proposé comprend trois états. Dans le premier état S0, le gestionnaire de type iAM Broker reçoit périodiquement, de ses homologues iAM des CSP (DC/BoD) dans l'alliance, leurs services disponibles avec différents niveaux de service ou encore les changements de niveaux de service. À la réception de ces informations, l'iAM Broker met à jour son dépôt

et reste dans le même état S0. Par la suite, après la réception des exigences de service du CSU pour construire un iSLA, l'iAM Broker passe à l'état S1.

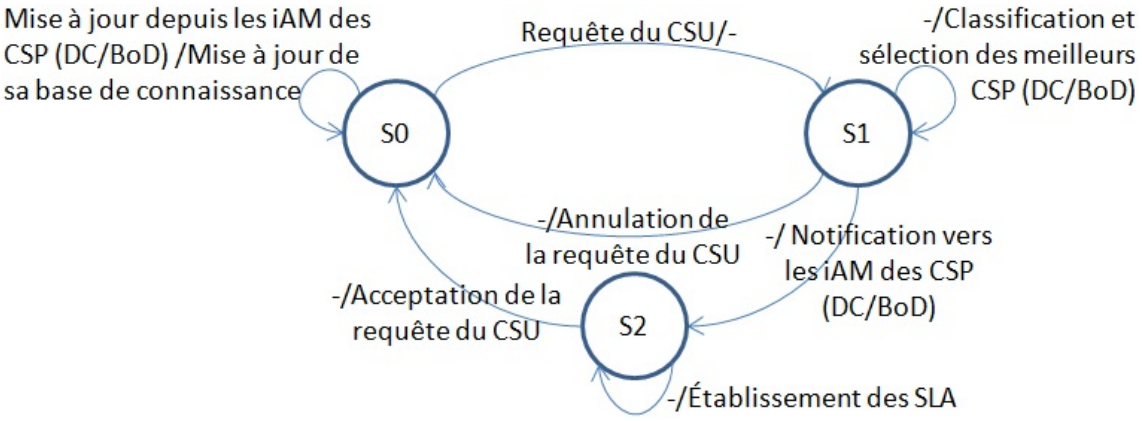

FIGURE 45 – Automate du cycle de vie d'établissement autonome du SLA pour l'iAM Broker.

Dans l'état S1, l'iAM Broker consulte son dépôt et compare les exigences du CSU avec les différents services et leurs niveaux de services proposés par les iAM des CSP partenaires, pour sélectionner les CSP appropriés c.à.d. ceux qui répondent aux exigences de QoS du CSU selon les algorithmes proposés (voir les algorithmes 1 à 4 qui intègrent l'amélioration évoquée au début de ce chapitre concernant la nouvelle fonction d'utilité). Si l'iAM Broker ne trouve pas des CSP qui répondent aux exigences du CSU, il rejette la demande du CSU et passe à l'état initial S0. Sinon, l'iAM Broker choisit les CSP appropriés, notifie chaque iAM des CSP sélectionnés pour réserver les ressources nécessaires, et passe à l'état S2. Dans l'état S2, l'iAM Broker établit un D_iSLA avec chaque iAM CSP (DC) et un B_iSLA avec chaque iAM CSP (BoD). Ensuite, les iAM CSP (DC) et les iAM CSP (BoD) informent leurs AM de bas niveau du niveau de service qu'ils doivent assurer afin d'allouer les ressources et offrir des services IaaS et/ou NaaS avec garantie de QoS selon ce niveau de service reçu (D_iSLA ou B_iSLA). Enfin, l'iAM Broker accepte la demande du CSU, établit le contrat de type iSLA, et passe à l'état initial S0.

6.4.2/ GESTION AUTONOME DU SLA DANS L'ARCHITECTURE FÉDÉRATION

Pour l'architecture Fédération, l'automate que nous spécifions pour l'iAM CSP_L comprend cinq états (voir figure 46). Dans l'état S0, lorsque l'iAM CSP_L reçoit une demande du CSU, l'état S1 est atteint pour calculer les besoins en ressources. Dans cet état, si le CSP_L peut répondre à toutes les exigences du CSU (scénario 1 ou 2 spécifiés dans la section 4.4.3), l'iAM CSP_L sélectionne les meilleures ressources de son domaine autonome (AD) et passe à l'état S2 pour réserver les ressources. Cependant, si le CSP_L ne peut pas offrir suffisamment de ressources (scénario 3 spécifié dans la section 4.4.3), il calcule les besoins en ressources restantes et passe à l'état S3.

Dans l'état S3, l'iAM CSP_L contacte d'autres iAM de CSP (DC) afin de répondre aux besoins du CSU pour les ressources restantes. Ensuite, chacun de ces iAM CSP (DC) sélectionne les meilleurs CSP (BoD) en fonction des algorithmes proposés (voir algorithme 1 ou algorithme 2 spécifiés dans la section 5.3). De plus, chaque iAM de CSP

décrit les services IaaS et/ou NaaS pour les ressources disponibles avec les différents niveaux de service afin de les envoyer à l'iAM CSP_L. Ainsi, si l'iAM CSP_L ne trouve pas des CSP qui répondent aux exigences du CSU, il rejette la demande du CSU et passe à l'état initial S0. Sinon, il sélectionne les meilleurs CSP qui répondent aux exigences de qualité de service du CSU pour les services de type IaaS et/ou NaaS selon les algorithmes de sélection proposés (voir les algorithmes 1 à 4 spécifiés dans la section 5.3) et passe à l'état S2.

Dans l'état S2, l'iAM CSP_L classifie et sélectionne les meilleurs CSP (BoD) qui permettent aux sites du CSU de l'atteindre selon l'algorithme proposé de sélection des routes (voir algorithme 1 ou algorithme 2 spécifiés dans la section 5.3). Ensuite, l'iAM CSP_L seulement (scénario 1 ou 2) ou alors avec les iAM des CSP (DC) sélectionnés (scénario 3) notifient les iAM des CSP (BoD) sélectionnés afin d'allouer les ressources réseau de type BoD et d'établir un B_iSLA avec chacun d'eux. Ainsi, chaque iAM des CSP (BoD) informe son nAM du niveau de service correspondant afin d'offrir des services de type NaaS avec garantie de QoS. De plus, dans le scénario 3, l'iAM CSP_L envoie une demande pour établir un D_iSLA avec chacun des iAM des CSP (DC) sélectionnés. Par conséquent, chaque iAM d'un CSP (DC) concerné fournit à son DnAM et à son hAM le niveau de service correspondant afin d'offrir des services de type IaaS et/ou NaaS avec garantie de QoS. Les différentes notifications du CSP_L vers les CSP (DC/BoD) font passer le CSP_L à l'état S4.

Dans l'état S4, le CSP_L envoie une notification à son hAM et à son DnAM afin de réserver et de configurer les machines virtuelles, les espaces de stockage et les ressources réseau dans le cadre d'une offre de services de type IaaS et/ou NaaS avec garantie de QoS. Enfin, l'iAM CSP_L accepte la demande du CSU, établit le contrat de type iSLA, et passe à l'état initial S0.

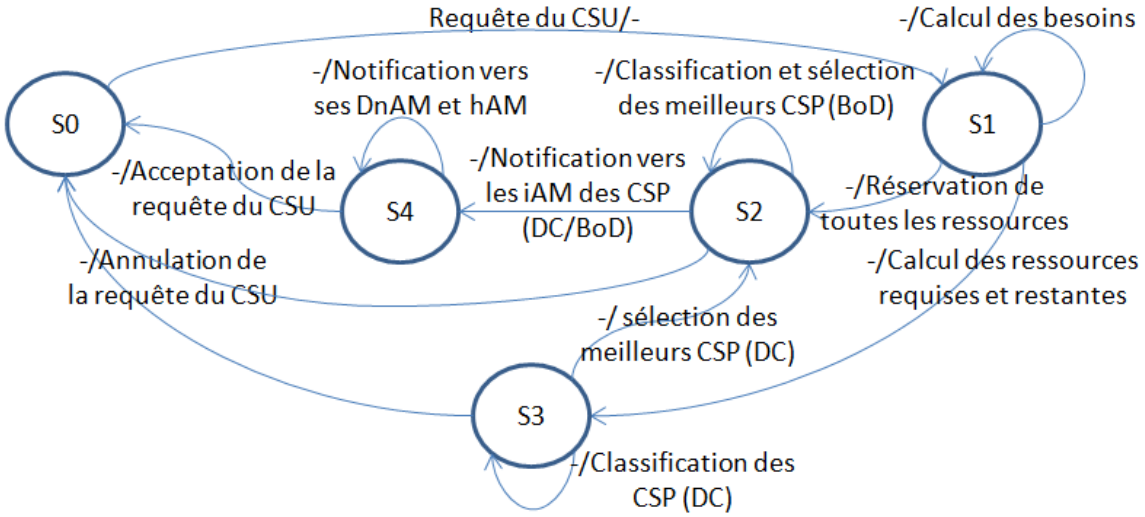

FIGURE 46 – Automate du cycle de vie d'établissement autonome du SLA pour l'iAM CSP_L.

6.4.3/ GESTIONNAIRES AUTONOMES DE BAS NIVEAU (AM)

Le cycle de vie d'un gestionnaire autonome (AM) de bas niveau (nAM, DnAM ou hAM) comportant les différentes phases de la boucle de contrôle fermée (voir figure 42) est décrit grâce à l'automate fini présenté dans la figure 47. Cet automate comprend trois états. Dans le premier état S0, le gestionnaire autonome (AM) surveille les ressources gérées et analyse les résultats remontés via l'interface capteur. Pour un gestionnaire autonome de type nAM dans un domaine autonome (AD) d'un CSP (BoD), les ressources gérées sont des ressources réseau (routeurs, commutateurs, liens de communication, etc.). D'autre part, pour un gestionnaire autonome de type DnAM dans un domaine autonome (AD) d'un CSP (DC), les ressources gérées sont des ressources réseau du DC. De plus, pour un gestionnaire autonome de type hAM dans un domaine autonome (AD) d'un CSP (DC), les ressources gérées sont des ressources de calcul de type machine virtuelle (VM) et des ressources de stockage. Dans ce contexte, chaque AM surveille l'état de ses ressources gérées et des paramètres de qualité de service en conformité avec le niveau de service fourni par l'iAM de son AD. En effet, le gestionnaire de type hAM surveille les paramètres de QoS de type IaaS tels que le temps de réponse et la disponibilité. Cependant, le nAM et le DnAM surveillent les paramètres de QoS de type NaaS tels que la latence, la gigue, le taux de perte de paquets, la bande passante et la disponibilité.

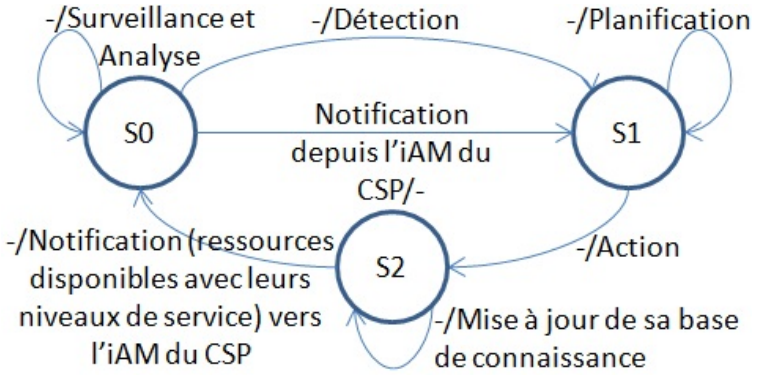

FIGURE 47 – Automate du cycle de vie d'AM de bat niveau (nAM, DnAM et hAM).

Le gestionnaire autonome de bas niveau (AM) passe à l'état S1 quand il reçoit une notification émanant de l'iAM de son CSP ou quand il détecte une violation d'un paramètre de QoS ou un dépassement du débit maximal du trafic du CSU défini dans le contrat de type iSLA en se basant sur des seuils prédéfinis (seuil de violation, seuil de risque de violation et seuil de débit maximal du trafic du CSU spécifiés dans la section 6.5). La notification de l'iAM d'un CSP peut être une requête pour obtenir l'ensemble des ressources disponibles avec leurs niveaux de service ($ASL = \{r_i, sl_i\}$ tel que r_i est la *i-ème* ressource avec son niveau de service sl_i), ou bien allouer, libérer, ou migrer un ensemble de ressources avec leurs niveaux de service ($RSL = \{r_i, sl_i\}$). Ces ressources peuvent être un ensemble de routes dans le réseau ($RT = \{rt_i, sl_i\}$ tel que rt_i est la *i-ème* route avec son niveau de service sl_i), un ensemble de machines virtuelles ($V = \{vm_i, sl_i\}$ tel que vm_i est la *i-ème* VM avec son niveau de service sl_i), ou un ensemble de capacités de stockage ($ST = \{st_i, sl_i\}$ tel que st_i est la *i-ème* ressource de stockage avec son niveau de service sl_i). Les différentes notations décrites dans cette partie sont utilisées dans l'algorithme 5 que nous spécifions dans ce qui suit.

Dans l'état S1, le gestionnaire autonome de bas niveau (AM) planifie la réalisation des actions appropriées en se basant sur l'algorithme 5 que nous proposons pour détailler la phase d'analyse et de planification. Ensuite, le gestionnaire AM exécute l'action planifiée et passe de l'état S1 à l'état S2. Dans cet état, le gestionnaire AM met à jour sa base de connaissances, notifie son iAM avec les ressources disponibles et leurs niveaux de service et passe à l'état initial S0 afin de surveiller et d'analyser le nouvel état des ressources. Enfin, s'il y a un dépassement du débit maximal du trafic du CSU par une requête rq_i, une violation ou un risque de violation, le gestionnaire autonome de bas niveau envoie à l'iAM du CSP une notification de dépassement, de violation ou alors de risque de violation avec le seuil du paramètre de qualité de service affecté ainsi que la ressource r_i concernée.

Algorithm 5 Phases d'analyse et de planification d'un gestionnaire autonome de bas niveau

```
1: if notification from CSP iAM then
2:     if allocate (RSL) then
3:         Allocate_resources(RSL)
4:         Configure_resources(RSL)
5:     else if release (RSL) then
6:         Release_resources(RSL)
7:     else if migrate (RSL) then
8:         Migrate_resources(RSL)
9:         Reconfigure_resources(RSL)
10:    else if get (ASL) then
11:        Notify_CSP_iAM(ASL)
12:    end if
13:    Notify_CSP_iAM(Confirmation Notification)
14: end if
15: if resource r_i is finished then
16:    Notify_CSP_iAM(Termination Notification, r_i)
17: end if
18: for each vm_i ∈ V or st_i ∈ ST or rt_i ∈ RT do
19:    if rq_i exceeds the throughput_threshold then
20:        Notify_CSP_iAM(exceeding throughput_threshold, rq_i)
21:    else if exceeding QoS_i_violation_threshold then
22:        Notify_CSP_iAM(violation, QoS_i_violationthreshold, vm_i or st_i or rt_i)
23:    else if exceeding QoS_i_riskthreshold then
24:        Notify_CSP_iAM(risk_violation, QoS_i_riskthreshold, vm_i or st_i or rt_i)
25:    end if
26: end for
27: Notify_CSP_iAM(ASL)
28: Update_knowledge_base()
```

6.4.4/ Gestionnaires autonomes de haut niveau : iAM d'un CSP (DC/BoD)

Le cycle de vie d'une entité de type iAM d'un CSP (DC/BoD) est décrit grâce à l'automate fini présenté dans la figure 48. L'automate que nous spécifions comprend quatre états.

Le premier état S0 est l'état initial où l'iAM du CSP attend de recevoir une notification de l'AM de bas niveau qui est sous son contrôle (nAM, DnAM ou hAM) ou encore une notification émanant de l'iAM du Broker ou du CSP_L. À la réception de cette notification, l'iAM du CSP passe au deuxième état S1. Le premier type de notification émanant de l'AM de bas niveau peut concerner les ressources disponibles avec leurs niveaux de service (ASL), une notification de confirmation (CN), une notification de terminaison (TN) d'une ressource r_i, une violation ou un risque de violation, ou une notification relative à certaines requêtes du CSU ayant dépassé le seuil de débit correspondant. Le deuxième type de notification émanant de l'iAM du Broker ou du CSP_L peut concerner une requête pour obtenir les ressources disponibles avec leurs niveaux de service (ASL), une demande pour obtenir les offres (O) des ressources réseau des CSP (BoD) entre ce CSP et le CSU, et enfin une requête pour allouer, libérer, ou migrer un ensemble de ressources avec leurs niveaux de service (RSL).

Par conséquent, dans l'état S1, l'iAM du CSP analyse ces notifications et planifie la réalisation des actions appropriées en se basant sur l'algorithme 6 que nous proposons pour spécifier cette phase d'analyse et de planification au sein de ce gestionnaire autonome . Par conséquent, en fonction du type de notification reçue, l'iAM du CSP peut notifier l'iAM du Broker ou du CSP_L et passe à l'état S3, ou bien il peut notifier les AM de bas niveau et passe à l'état S2.

Dans l'état S2, l'iAM du CSP attend de recevoir la notification de confirmation des gestionnaires autonomes de bas niveau pour passer à l'état S1 afin d'analyser le nouvel état des ressources. Cependant, dans l'état S3, l'iAM du CSP met à jour son dépôt, notifie l'iAM du Broker ou du CSP_L avec les ressources disponibles et leurs niveaux de service, et passe à l'état initial S0 pour surveiller le nouvel état des ressources.

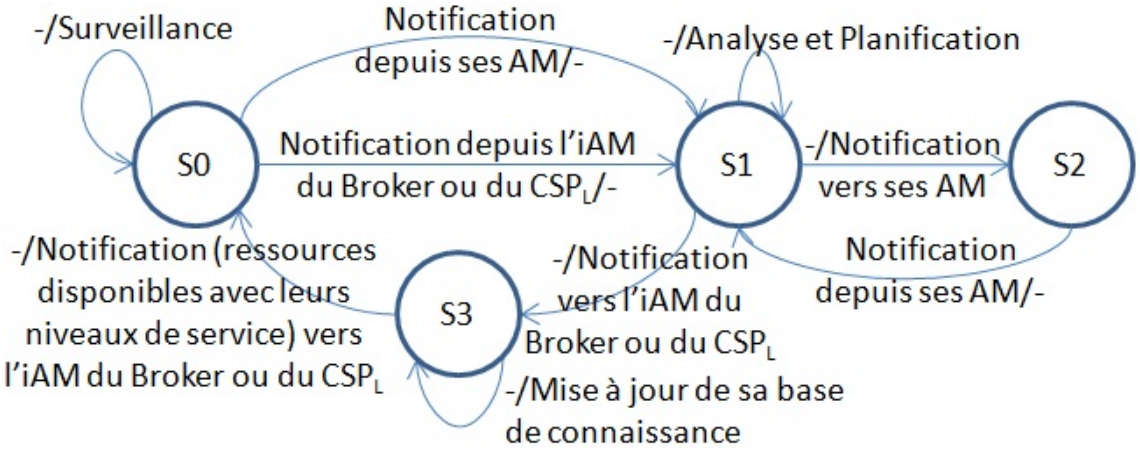

FIGURE 48 – Automate du cycle de vie de l'iAM d'un CSP (DC/BoD).

Dans ce contexte, l'iAM du CSP spécifie un seuil pour le débit maximal du CSU basé sur le D_iSLA ou le B_iSLA convenu. Par conséquent, si certaines requêtes rq_i du CSU dépassent le seuil de débit correspondant, l'iAM du CSP peut arrêter ou retarder ces requêtes comme il peut allouer de nouvelles ressources payées par le CSU en se basant sur le contrat de type iSLA établi. Ensuite, il envoie une notification de dépassement du seuil de débit pour l'iAM du Broker ou du CSP_L.

De plus, s'il y a une violation ou un risque de violation, l'iAM du CSP calcule ses res-

Algorithm 6 Phases d'analyse et de planification de l'iAM d'un CSP (DC/BoD)

if notification from Broker iAM or CSP$_L$ iAM **then**
  if get (ASL) **then**
    Notify_AM(get (ASL))
  else if allocate (RSL) **then**
    Notify_AM(allocate (RSL))
  else if release (RSL) **then**
    Notify_AM(release (RSL))
  else if migrate (RSL) **then**
    Notify_AM(migrate (RSL))
  end if
  if CSP$_L$ iAM and get_CSP(BoD) resources **then**
    $O \leftarrow$ get_CSP(BoD)_resources(CSU) ▹ algoritme 1 ou 2
    Notify_CSP$_L$_iAM(O)
  end if
  Notify_CSP$_L$_or_Broker_iAM(CN)
else if notification from its AMs **then**
  if ASL **then**
    Update_repository()
    Notify_Broker/CSP$_L$_iAM(ASL)
  else if CN **then**
    Update_repository()
  else if resource r_i TN **then**
    Compute_violation_and_penalty() ▹ voir section 6.5
    Notify_Broker/CSP$_L$_iAM(TN, r_i, cv_i)
    Notify_AM(release (r_i))
  else if rq_i exceeds the throughput threshold **then**
    Delay_or_Reject(rq_i) or Notify_AM(allocate (RSL))
  else if violation risk or violation notification **then**
    $AfR \leftarrow$ get_affected_resources()
    $AvR \leftarrow$ get_available_resources()
    if $AvR \neq \emptyset$ **then**
      Notify_AM(allocate (AvR))
      Notify_AM(release (AfR))
    else if $AvR == \emptyset$ or insufficient_resources **then**
      Notify_Broker_or_CSP$_L$_iAM(risk_violation or Violation, QoS$_i$_violation-threshold, r_i)
    end if
    if violation **then**
      Compute_violation_and_penalty()
    end if
  end if
end if
Notify_Broker/CSP$_L$_iAM(ASL)
Update_knowledge_base()

sources disponibles (AvR) et essaie d'allouer de nouvelles ressources gratuitement dans son domaine autonome (AD) mais aussi de libérer les ressources affectées (AfR) pour

éviter ou encore réduire la violation. Cependant, si l'iAM du CSP ne trouve pas suffisamment de ressources dans son domaine autonome, il notifie l'iAM du Broker ou du CSP_L afin d'allouer de nouvelles ressources dans d'autres CSP. Ces nouvelles ressources seront payées par le gestionnaire iAM CSP lui-même. Ensuite, en cas de violation, l'iAM du CSP calcule la violation et la pénalité correspondante (cf. section 6.5). Ainsi, quand l'utilisation d'une ressource r_i est terminée, l'iAM du CSP calcule les violations pour r_i et envoie une notification à l'AM de bas niveau sous son contrôle pour libérer la ressource r_i, puis il envoie une notification de terminaison à l'iAM du Broker ou du CSP_L avec les violations calculées pour r_i (cv_i) selon le B_iSLA ou le D_iSLA établi. Le fonctionnement que nous venons de décrire est illustré par l'algorithme 6.

6.4.5/ Garantie de QoS avec l'iAM du Broker ou du CSP_L

Après l'établissement autonome des SLA, l'objectif est de garantir les performances des services Cloud en surveillant régulièrement le iSLA contracté avec le CSU et les B_iSLA/D_iSLA spécifiés entre les CSP et ce en utilisant les capteurs des AM ainsi que les politiques définies par chaque domaine autonome (AD) afin d'éviter ou encore réduire les violations des SLA. En effet, lorsqu'un pic inattendu d'utilisation des services de Cloud se produit, il y a un risque de dégradation des paramètres de QoS dans un ou plusieurs CSP (DC/BoD). Par conséquent, quand un risque de violation concernant un paramètre de QoS est détecté au niveau d'un CSP, les ressources disponibles dans ce CSP ou dans d'autres CSP doivent être découvertes et réservées d'une manière autonome (c.à.d. assurer l'auto-restauration, l'auto-optimisation et l'auto-reconfiguration) grâce à la boucle de contrôle fermée et aux interactions entre les AM. Pour ce faire, nous décrivons le processus de gestion autonome des ressources du Cloud grâce à un automate fini qui contient quatre états (voir figure 49).

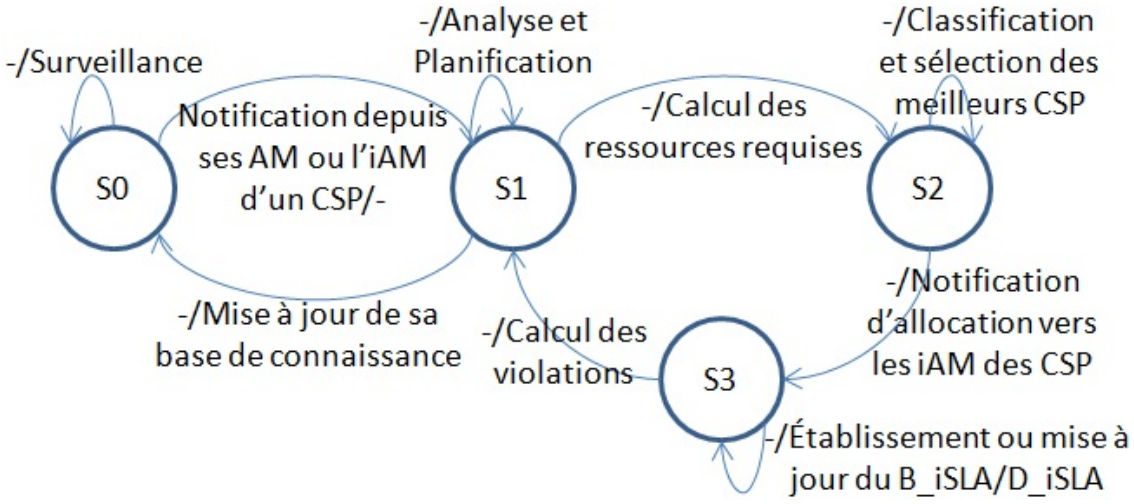

Figure 49 – Automate du cycle de vie de l'iAM du Broker ou du CSP_L pour la gestion autonome des ressources.

Dans l'état S0, l'iAM du Broker dans l'architecture Broker attend la réception d'une notification de l'iAM d'un CSP partenaire pour passer au deuxième état S1. Cependant, dans l'architecture de type Fédération, l'iAM du CSP_L attend la réception d'une notification

de l'iAM d'un CSP ou des gestionnaires autonomes (AM) de bas niveau qui sont sous son contrôle (nAM, DnAM ou hAM) pour passer au deuxième état S1. Cette notification collectée par l'iAM grâce à l'interface de type capteurs peut concerner les ressources disponibles avec leurs niveaux de service (ASL) ou encore une notification de confirmation (CN), une notification de terminaison (TN) d'une ressource r_i avec la violation cv_i calculée et associée à r_i, mais aussi une violation ou un risque de violation avec le seuil de QoS et la ressource r_i affectée. Cette notification peut aussi concerner les offres (O) des ressources des CSP (BoD) entre un CSP (DC) et le CSU, ou une notification que certaines requêtes du CSU ont dépassé le seuil de débit correspondant.

Dans l'état S1, l'iAM du Broker ou du CSP_L analyse la notification reçue et planifie sa réaction avec les mesures appropriées en se basant sur l'algorithme 7 que nous proposons pour spécifier la phase d'analyse et de planification de l'iAM du Broker ou du CSP_L dans le cadre de la gestion autonome des ressources. Ainsi, si la notification concerne les ressources disponibles avec leurs niveaux de service (ASL), ou s'il s'agit d'une notification de confirmation (CN) ou encore les offres (O) des CSP (BoD), l'iAM du Broker ou du CSP_L met à jour son dépôt et passe à l'état S0. Cependant, si la notification est une notification de terminaison (TN) d'une ressource r_i avec la violation cv_i calculée et associée à r_i, l'iAM du Broker ou du CSP_L calcule cette violation pour calculer la pénalité, il met à jour son dépôt et passe à l'état S0. De plus, si la notification de terminaison (TN) est envoyée à l'iAM du CSP_L depuis ses gestionnaires autonomes (AM) de bas niveau, ce dernier envoie en plus une notification de libération de r_i à ces AM. D'autre part, si la notification est une violation ou un risque de violation émanant de l'iAM d'un CSP vers l'iAM du Broker, ce dernier essaie d'allouer de nouvelles ressources dans d'autres CSP. Ces nouvelles ressources, payées par le CSP à l'origine de la notification, permettent, d'éviter ou de réduire les violations relatives aux contrats de type iSLA, D_iSLA ou le B_-iSLA déjà établis. Ainsi, le gestionnaire iAM du Broker calcule les ressources disponibles (AvR) et passe à l'état S2. Enfin, si la notification est une violation ou un risque de violation émanant de l'iAM d'un CSP ou d'un gestionnaire autonome (AM) de bas niveau vers l'iAM du CSP_L, ce dernier essaie d'allouer de nouvelles ressources dans son domaine autonome (AD) seulement (scénario 1 ou 2) ou dans son AD et d'autres AD de CSP (scénario 3), payées par la source de notification, pour éviter ou réduire les violations relatives aux contrats établis de type iSLA, D_iSLA ou B_iSLA. Ainsi, l'iAM du CSP_L calcule les ressources disponibles (AvR) et alloue l'intégralité (scénario 1 ou 2) ou une partie (scénario 3) de ces ressources dans son AD. Puis, dans le scénario 1 ou le scénario 2, l'iAM du CSP_L met à jour son dépôt et passe à l'état S0. Par contre, dans le scénario 3, il passe à l'état S2 pour calculer les ressources restantes.

Dans l'état S2, l'iAM du Broker ou du CSP_L sélectionne et classe l'ensemble des meilleures ressources des CSP en se basant sur les algorithmes d'optimisation et de sélection des ressources (voir les algorithmes 1 à 4) en accord avec le contrat iSLA établi. Ensuite, l'iAM du Broker ou du CSP_L envoie une notification d'allocation à chaque iAM des CSP sélectionnés pour allouer des ressources et passe à l'état S3. Dans cet état, l'iAM du Broker ou du CSP_L établit ou met à jour le B_iSLA ou le D_iSLA. De plus, lorsque l'iAM du Broker ou du CSP_L ne peut pas éviter la violation, il calcule la pénalité correspondante lorsque le service est terminé et passe à l'état S1. D'autre part, si certaines requêtes du CSU (rq_i) dépassent le seuil de débit correspondant dans l'AD du CSP_L, l'iAM du CSP_L peut arrêter ou retarder ces requêtes comme il peut allouer de nouvelles ressources (r) payées par le CSU en se basant sur le contrat iSLA établi. Enfin, l'iAM du Broker ou du CSP_L qui se retrouve dans l'état S1, analyse le nouvel état des ressources.

Si un problème est détecté, l'iAM du Broker ou du CSP_L essaye de planifier et de résoudre ce problème. Sinon, il met à jour son dépôt (base de connaissances) avec les changements et passe à l'état initial S0. Le fonctionnement que nous venons de décrire dans cette partie est illustré par l'algorithme 7.

Algorithm 7 Phases d'analyse et de planification de l'iAM du Broker ou du CSP_L pour la gestion autonome des ressources

```
1: if (ASL) or (CN) or (O) then
2:     Update_repository()
3: else if resource r_i TN with cv_i then
4:     Compute_violation_and_penalty(cv_i)          ▹ voir section 6.5
5:     Update_repository()
6:     if TN is from AM to CSP_L iAM then
7:         Notify_AM(release (r_i))
8:     end if
9: else if violation risk or violation notification then
10:    AvR ← get_available_resources()
11:    BR ← classify_best_resources()               ▹ algorithmes 1 à 4
12:    Notify_selected_CSPs(allocate(BR))
13:    Notify_CSP_iAM(release (affected resources))
14:    Update_or_establish_D_iSLA/B_iSLA()
15:    if violation then
16:        Compute_violation_and_penalty()          ▹ voir section 6.5
17:    end if
18: else if rq_i exceeds the throughput threshold then
19:    Delay_or_Reject(rq_i) or Notify_AM(allocate (r))
20: end if
21: Update_knowledge_base()
```

6.5/ Gestion autonome des violations

Afin de bien auto-gérer les ressources du Cloud, la méthode du calcul des coûts, de la détection des violations futures du SLA, et du calcul des pénalités et des réputations doit être spécifiée dans notre architecture autonome de Cloud Networking.

6.5.1/ Détection de violation

La stratégie que nous proposons pour de détection des violations du SLA est basée sur l'utilisation de seuils prédéfinis de violations spécifiés par chaque AD. D'une part, un seuil de violation est une valeur qui indique le niveau de performance minimal acceptable pour une application en accord avec le SLA convenu. Par conséquent, le dépassement de la valeur du seuil de violation pour un paramètre de QoS, indique notamment l'occurrence de la violation du SLA et par la suite le système enregistre les informations nécessaires pour calculer les pénalités appropriées. D'autre part, un seuil de risque de violation est une valeur qui indique le risque d'une violation pour un niveau de performance. Par conséquent, avec cette information, le système peut réagir rapidement pour

éviter la violation du SLA et éviter au CSP des pénalités coûteuses.

Actuellement, la différence entre le seuil de violation et le seuil de risque de violation est de 3%. Par exemple, si la latence définie par le CSU dans l'iSLA est inférieure à 100 ms, donc, 100 ms est le seuil de violation et le seuil de risque de violation est de 97 ms. De plus, chaque iAM du CSP spécifie un seuil, surveillé par ses AM de bas niveau, qui indique si le CSU a dépassé le débit maximal du trafic défini dans le contrat de type iSLA.

De plus, le CSP surveille les performances de chaque application en utilisant les différentes interfaces capteurs pour la détection des violations du SLA et pour déterminer l'intervalle optimal de mesure pour la surveillance. Par exemple Amazon CloudWatch fournit un service gratuit de surveillance des ressources pour les utilisateurs avec un intervalle de 5 minutes [45]. Dans chaque CSP_j (DC/BoD), nous considérons que le nombre d'intervalles de temps de surveillance au cours de l'exécution de l'application du CSU est I_j et le nombre de violation qui a eu lieu dans le CSP_j pour le *i-ème* paramètre de QoS est V_{ij}. Ainsi le nombre maximal de violations est atteint lorsque le SLA est violé dans chaque intervalle, c.à.d. $V_{ij} = I_j$. Cependant, si le CSU n'a pas respecté le débit maximal du trafic et une violation a eu lieu, le système ne tient pas compte de cette violation.

6.5.2/ Calcul des pénalités

La pénalité (Pn_j) dans un CSP_j est calculée comme une partie (X_j) des revenus du CSP_j (Rev_j) quand il n'y a pas des violations de SLA dans ce CSP_j. Cette partie X_j présentée par l'équation (28) dépend du nombre de violations (V_{ij}) qui ont eu lieu dans le CSP_j pour le *i-ème* paramètre de QoS par rapport au nombre maximum de violations I_j dans ce CSP_j.

De plus, pour un paramètre de QoS particulier, l'influence de la violation sur l'application du CSU n'est pas la même comparé à d'autres paramètres. Ainsi, nous ne pouvons pas avoir la même pénalité pour chaque violation. Donc, face à cette différence de signification, nous utilisons des poids normalisés w_i affectés par la CSU pour chaque *i-ème* paramètre de QoS en fonction de son importance et du type d'application.

D'autre part, pour que X_j soit plus réaliste, nous utilisons une fonction de Gompertz présentée par l'équation (29). Cette fonction est un type de fonction sigmoïde, et elle est défini par $y(t) = ae^{be^{ct}}$, tel que a est l'asymptote supérieure, b est le déplacement le long de l'axe x, c est le taux de croissance, et enfin b et c sont des nombres négatifs. Dans notre cas, nous considérons $a = 1, b = -10, c = -7$. Cependant, nous considérons qu'au-delà de 10% du seuil de violations, le CSP_j commence à payer les pénalités (voir figure 50).

$$X_j = \sum_{i=1}^{q} \frac{(w_i \times V_{ij})}{I_j} \tag{28}$$

$$\alpha(X_j) = \begin{cases} 0 & \text{si } X_j \leq 0.1 \\ 1 & \text{si } X_j \geq 1 \\ e^{-10e^{-7X_j}} & \text{si } 0.1 < X_j < 1 \end{cases} \tag{29}$$

Par conséquent, la pénalité (Pn_j) dans un CSP_j est calculée en se basant sur l'équation (30), le profit d'un CSP_j ($Prof_j$) est calculé en se basant sur l'équation (31) et le coût payé par le CSU ($Cost_{CSU}$) est calculé en se basant sur l'équation (32).

$$Pn_j = \alpha(X_j) \times Rev_j \tag{30}$$

$$Prof_j = Rev_j - Pn_j \tag{31}$$

$$Cost_{CSU} = Cost_{total} - \sum_j Pn_j \tag{32}$$

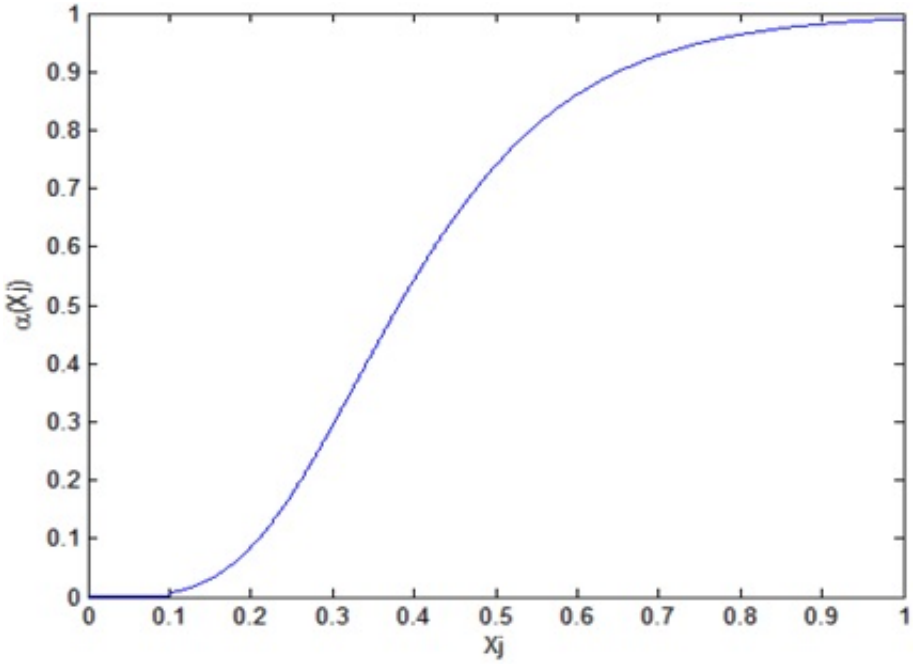

FIGURE 50 – Fonction proposée pour les pénalités.

6.5.3/ CALCUL DE LA RÉPUTATION DES CSP

Chaque CSP dans l'alliance a un score de réputation basé sur le nombre de violations de QoS qui ont eu lieu dans ce CSP. La valeur de la réputation appartient à l'intervalle]0, 1] et la valeur initiale est égale à 1. Par conséquent, lorsque des violations se produisent dans un CSP_j, l'iAM du Broker ou du CSP_L recalcule, après la terminaison de l'application du CSU, la réputation (ψ_j) du CSP_j en se basant sur l'équation (33).

$$\psi_j = \frac{\psi_j + (1 - \alpha(X_j))}{2} \tag{33}$$

6.6/ Validation de notre architecture de gestion autonome

Pour valider notre architecture autonome de Cloud Networking, nous présentons dans cette section deux cas d'utilisation pour l'établissement efficace des applications de Cloud vidéoconférence et des applications de calcul intensif. Nous testons les scénarios correspondants comme une preuve de concept et une validation de nos architectures autonomes de type Broker et Fédération. De plus, nous évaluons les performances avec les pénalités en cas de violation grâce à différentes simulations en utilisant la boîte à outils CloudSim [33].

Ainsi, nous utilisons notre architecture autonome pour établir d'une façon autonome les SLA et auto-gérer les applications correspondantes tout en réduisant le coût total de l'application sans violer les paramètres de QoS de bout-en-bout.

6.6.1/ Environnement de simulation

Nous utilisons le même environnement de simulation décrit dans le chapitre 5 pour la garantie de QoS. Afin de simuler notre architecture autonome, nous étendons CloudSim en proposant la classe «Sensor» pour la surveillance des ressources et pour préciser les seuils de violation et de risque de violation. De plus, nous étendons la classe «SimEven» pour spécifier les notifications. Ainsi, nous implémentons pour chaque entité autonome les algorithmes proposés et qui constituent la boucle de contrôle fermée des gestionnaires autonomes de notre architecture (iAM, nAM, DnAM et hAM).

Dans ce contexte, le CSU spécifie son type de service demandé et les exigences correspondantes de QoS au Cloud Broker ou au CSP_L en utilisant l'interface graphique proposée. Ensuite, après la sélection des meilleurs CSP, le Cloud Broker ou le CSP_L alloue les ressources nécessaires aux sites du CSU et établit d'une façon autonome les différents SLA. Ainsi, les entités participantes lancent les applications, puis l'iAM du Broker ou du CSP_L commence la gestion autonome des ressources en se basant sur l'architecture autonome proposée. Enfin, quand les applications sont terminées, le Cloud Broker ou le CSP_L libère les ressources allouées et calcule les réputations des CSP ainsi que les pénalités en se basant sur les violations, tout en permettant au CSU de payer en fonction de ressources utilisées.

6.6.2/ Cloud vidéoconférence

6.6.2.1/ Scénario de simulation

Pour évaluer les performances de notre architecture autonome tout en considérant le service de vidéoconférence, nous utilisons les gestionnaires autonomes de Cloud spécifiés avec les algorithmes proposés de sélection des CSP et de la gestion autonome des ressources pour valider le scénario de vidéoconférence dans les architectures de Cloud Networking (Broker et Fédération). L'objectif du Cloud Broker ou du CSP_L est de minimiser le coût total du service de vidéoconférence sans violer les contraintes de bout-en-bout de QoS en accord avec les SLA établis, tout en réduisant les pénalités des CSP.

Dans ce cas d'utilisation avec l'application de Cloud vidéoconférence, nous évaluons quatre scénarios de simulation :

1- Le premier correspond à une sélection statique (S) des ressources sans établissement autonome des SLA, ni sélection des meilleures ressources, ni garantie de QoS.

2- Le deuxième scénario correspond à une sélection des meilleures ressources en se basant sur l'architecture Broker (B-1) et sur l'architecture Fédération (F-1) avec établissement autonome des SLA et avec garantie de QoS mais sans présence de violations.

3- Le troisième scénario correspond à une sélection des meilleures ressources en se basant sur l'architecture Broker (B-2) et sur l'architecture Fédération (F-2) avec établissement autonome des SLA, une garantie de QoS, et présence de violations mais sans gestion autonome des ressources. Ainsi, le système ne réagit pas en cas de violation.

4- Le quatrième scénario est le même que le troisième, mais avec gestion autonome des ressources (B-3/F-3).

Dans chaque scénario, nous avons plusieurs sites de CSU connectés à différents CSP (BoD) qui utilisent des services de type IaaS et/ou NaaS dans l'architecture Broker et l'architecture Fédération. Le but est de sélectionner les meilleures routes de distribution des flux de vidéoconférence en passant par des machines virtuelles allouées pour les serveurs de vidéoconférence. Ensuite, le système autogère les ressources pour éviter ou réduire les violations. De plus, la violation est simulée grâce à un pic de trafic envoyé dans le Cloud afin d'affecter les paramètres de qualité de service des ressources du CSU.

Le délai et la gigue sont des paramètres de qualité de service très importants pour les applications interactives temps réel telles que la vidéoconférence. Un délai de bout en bout supérieur à 200 ms et une gigue supérieure à 30 ms peuvent provoquer une dégradation du service de vidéoconférence. Ainsi, dans les scénarios de Broker et de Fédération, le CSU spécifie une latence réseau maximale de 180 ms, une gigue de réseau maximale de 30 ms et un temps de réponse maximal de transcodage de 20 ms. Pour les autres paramètres de qualité de service, le CSU demande un niveau de service Gold et tous les poids sont égaux à 1.

Nous simulons quatre types de vidéos envoyées par le CSU avec une longueur d'une heure, une taille de 1, 2, 3 et 4 Go, respectivement, et une bande passante demandée égale à 2.2Mb/s, 4.5Mb/s, 6.8Mb/s et 9.1Mb/s, respectivement. Après plusieurs simulations pour chaque type de vidéo, avec à chaque fois une distribution géographique différente des sites du CSU, nous calculons au début le coût global de la bande passante (voir figure 51) de chaque type de vidéo tout en comparant le premier scénario (sélection statique (S)) avec le second scénario (sélection en se basant sur les architectures Broker et Fédération sans violation (B-1 / F-1)). Ensuite, nous calculons la latence moyenne globale du réseau (voir figure 52) des types de vidéo simulées, avec une période d'intervalle d'échantillonnage d'une minute, pour faire une comparaison entre le premier scénario (S), le troisième scénario (B-2/F-2) et le quatrième scénario (B-3/F-3). De plus, nous calculons le coût total des différentes consommations de ressources du CSU ($Cost_{total}$). Ce coût total est composé du coût payé par le CSU ($Cost_{CSU}$) et la pénalité globale pour chaque type de vidéo (voir figure 53). Enfin, nous calculons la réputation du CSP_1 (BoD) (voir figure 54) et faire une comparaison entre le troisième scénario (SC3) et le quatrième scénario (SC4).

6.6.2.2/ RÉSULTATS

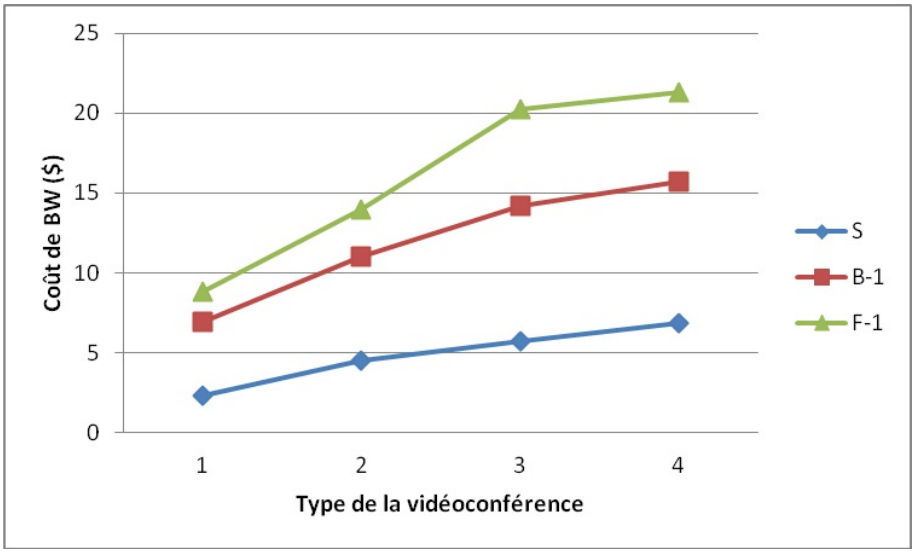

FIGURE 51 – Évaluation du coût global de la bande passante.

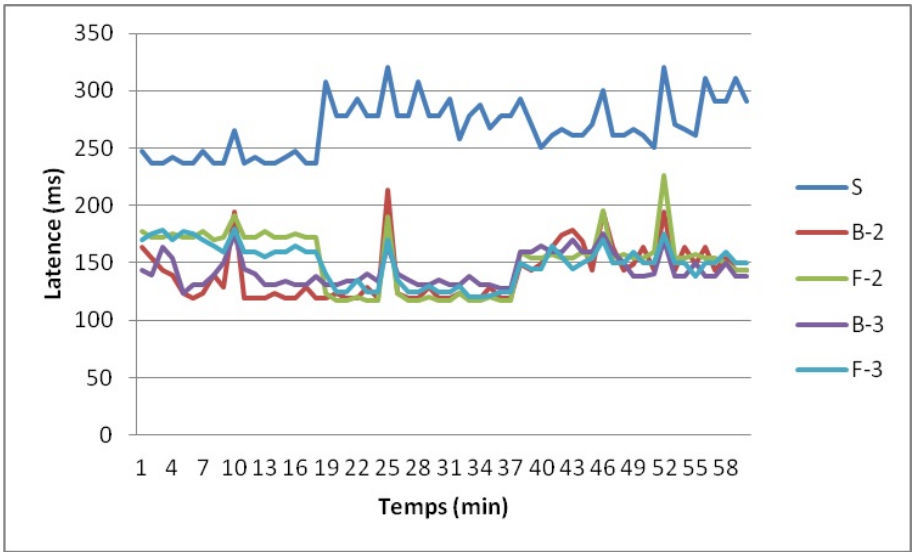

FIGURE 52 – Évaluation de la latence globale de bout en bout.

Comme le montre la figure 51, le coût global de la bande passante augmente lorsque la taille de la vidéo et la bande passante correspondante augmente. De plus, dans une sélection statique (S), le coût global de la bande passante est inférieur à ceux de la sélection basée sur l'architecture Broker (B-1) et l'architecture Fédération (F-1) en raison de la garantie de la QoS de type NaaS. Cependant, le coût global de la bande passante dans le scénario de l'architecture Broker est moins important que le coût de la bande passante dans le scénario de l'architecture Fédération. Ces résultats sont obtenus en raison de la sélection des ressources disponible au sein du CSP_L d'abord, puis la sélection des ressources restantes au sein des autres CSP. En effet, les ressources du CSP_L ne sont

pas obligatoirement les meilleures en termes de coût. Ainsi, nous remarquons que l'architecture Broker est la plus économique, tout en assurant les exigences de qualité de service du CSU.

D'autre part, comme illustré dans la figure 52, les résultats obtenus montrent une meilleure latence pour notre architecture Broker (B-2/3) et Fédération (F-2/3), par rapport à une sélection statique (S) qui ne respecte pas la limite de 180 ms. Cependant, en cas de violation, le Cloud Broker (B-3) et le CSP_L (F-3) peuvent réagir afin d'éviter ou réduire les violations (ex. 10/25/46/52 minutes) contrairement aux scénarios Broker et Fédération sans gestion autonome des ressources (B-2/F-2).

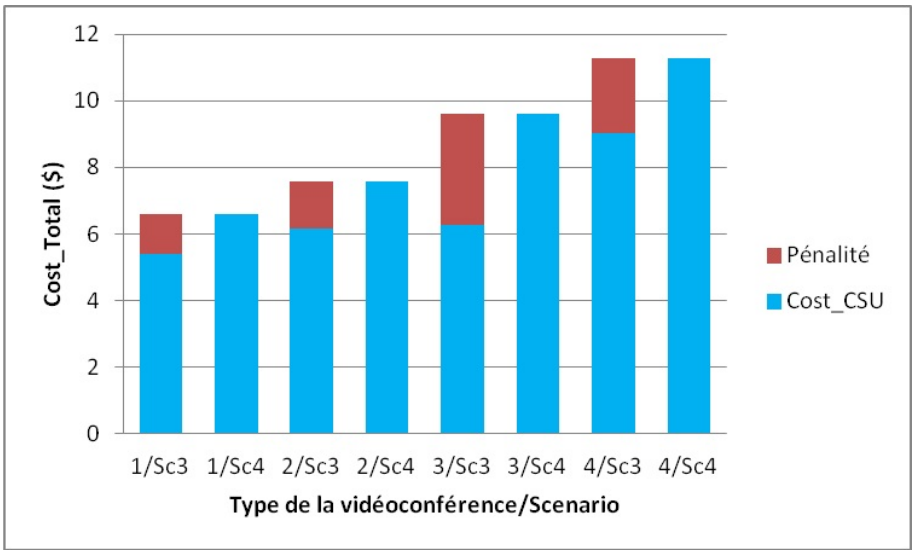

FIGURE 53 – Évaluation du coût global du CSU et de la pénalité.

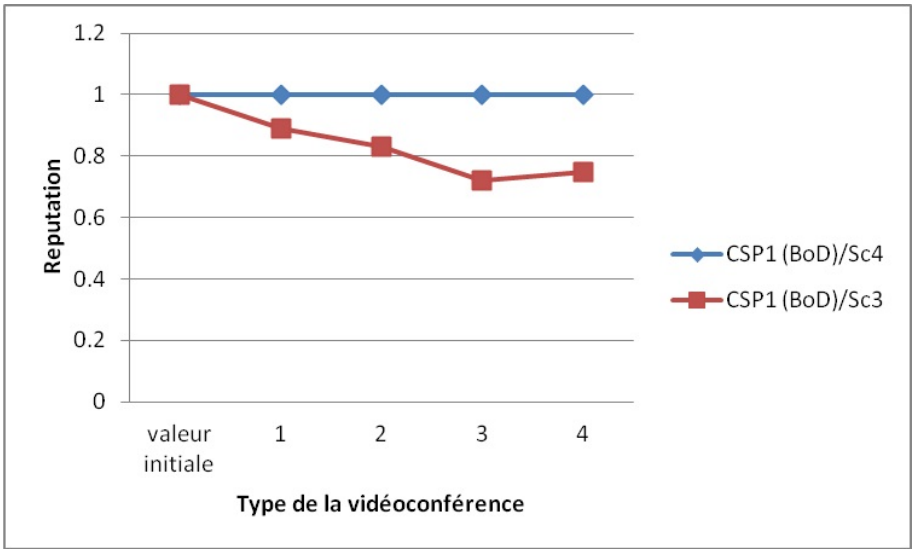

FIGURE 54 – Évaluation de la réputation pour le CSP_1 (BoD).

De plus, comme l'illustre la figure 53, en utilisant notre architecture autonome de Cloud Networking (quatrième scénario (B-3/F-3)), la pénalité globale est réduite ou évitée par

rapport à l'architecture sans gestion autonome des ressources (troisième scénario (B-2/F-2)).

Enfin comme le montre la figure 54, la réputation du CSP_1 (BoD) dans le quatrième scénario reste meilleure que celle obtenue avec le troisième scénario grâce à la réduction des violations et des pénalités.

6.6.3/ Calculs intensifs

6.6.3.1/ Scénario de simulation

Dans ce type d'applications, nous utilisons les AM et les algorithmes proposés pour l'établissement autonome des SLA et la sélection des meilleurs CSP pour valider ce cas d'utilisation dans les architectures Broker et Fédération. Le but est de déterminer les meilleures machines virtuelles à allouer pour les Jobs demandés, tout en minimisant le coût total de l'application et éviter ou réduire les violations des contraintes de QoS de bout en bout grâce à une gestion autonome des ressources allouées. Les demandes de service du CSU sont simulées comme une série de Jobs exécutés par les machines virtuelles allouées.

Nous simulons cette application de calculs intensifs avec quatre sites CSU où chacun envoie 100 Jobs à quatre VM, et avec des longueurs différentes de Jobs (200, 250, 300 et 350 instructions). Le CSU choisit un service de type IaaS avec NaaS et un temps de réponse maximal de 300 ms, une latence maximale de 150 ms, et une bande passante de 1Mb/s. Pour les autres paramètres de qualité de service, le CSU demande un niveau de service Silver et tous les poids sont égaux à 1. Dans ce cas d'utilisation avec une application de calcul intensif, nous évaluons les mêmes quatre scénarios de simulation que ceux du premier cas d'utilisation relatif à la vidéoconférence. Ainsi, nous évaluons les scénarios relatifs à une sélection statique (S), une sélection sans présence de violation (B-1/F-1), une sélection sans gestion autonome des ressources (B-2/F-2), et une sélection avec gestion autonome des ressources (B-3/F-3). Dans chaque scénario, nous avons plusieurs sites du CSU connectés à différents CSP (BoD) qui utilisent des services de type IaaS afin de solliciter des VM pour l'application de calcul intensif avec ou sans services de type NaaS dans l'architecture Broker et Fédération. De plus, la violation est simulée grâce à un pic de trafic envoyé dans le Cloud afin d'affecter les paramètres de qualité de service des ressources du CSU.

Après plusieurs exécutions des scénarios de simulation pour chaque longueur de Job pendant une heure d'utilisation des VM, nous calculons au début le coût global de VM (voir figure 55) tout en comparant le premier scénario (sélection statique (S)) au second scénario (sélection en se basant sur les architectures Broker et Fédération sans violation (B-1/F-1)). Ensuite, nous calculons le temps de réponse moyen global des Jobs (voir figure 56) pour faire une comparaison entre le premier scénario (S), le troisième scénario (B-2/F-2) et le quatrième scénario (B-3/F-3). De plus, nous calculons le coût total des différentes consommations de ressources du CSU ($Cost_{total}$). Ce coût est composé du coût payé par le CSU ($Cost_{CSU}$) et la pénalité globale pour chaque longueur de Job (voir figure 57). Enfin, nous calculons la réputation du CSP_1 (DC) (voir figure 58) pour faire une comparaison entre le troisième scénario (SC3) et le quatrième scénario (SC4).

6.6.3.2/ RÉSULTATS

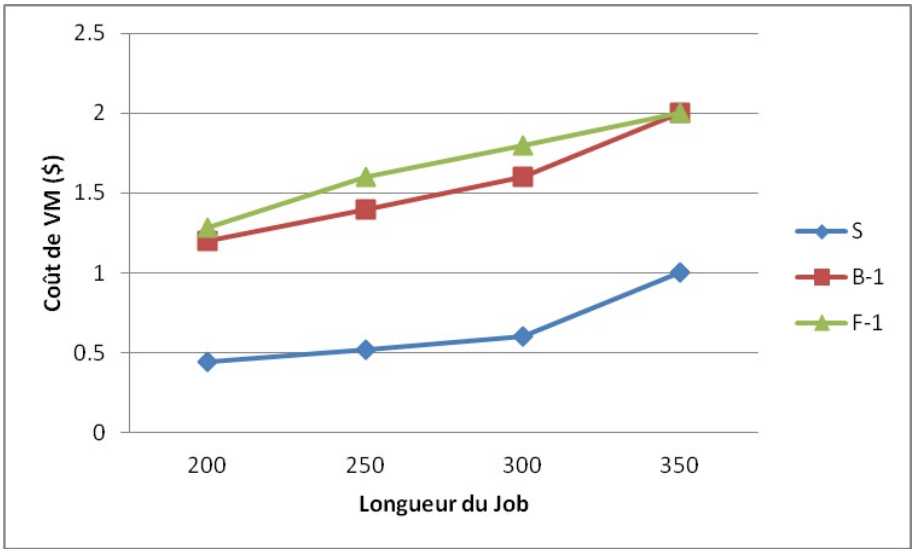

FIGURE 55 – Évaluation du coût global de VM.

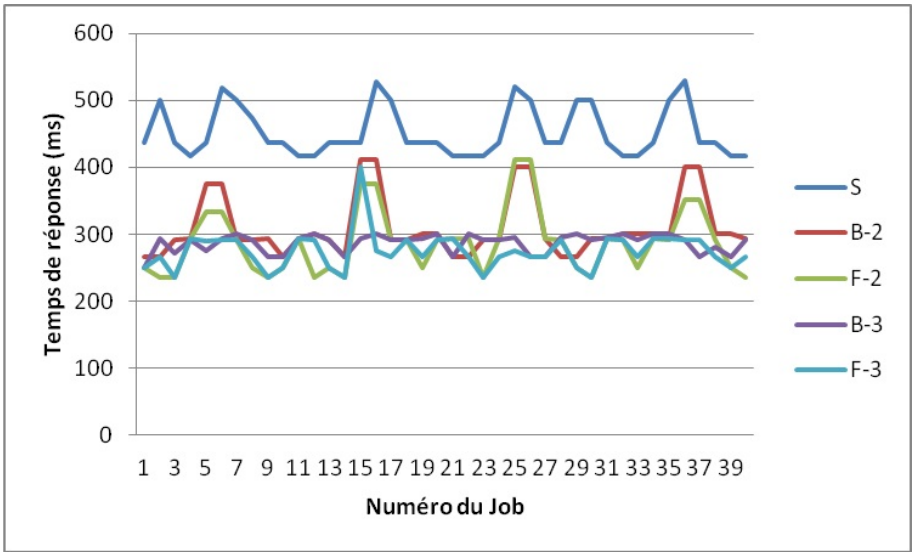

FIGURE 56 – Évaluation du temps de réponse global de bout en bout.

La figure 55 montre que le coût global des VM augmente lorsque la longueur des Jobs augmente. Nous pouvons constater que d'une part, le coût global des VM dans la sélection statique (S) est inférieur au coût des VM pour la sélection basée sur l'architecture Broker (B-1) et l'architecture Fédération (F-1), en raison de la garantie de QoS de type IaaS qui est assurée dans les deux derniers scénarios. D'autre part, le coût global de VM dans le scénario de l'architecture Broker est moins important que le coût de VM dans le scénario de l'architecture Fédération. Ces résultats sont obtenus suite à une sélection prioritaire des ressources disponibles au sein du CSP_L, suivie d'une éventuelle sélection des ressources restantes au sein des autres CSP. Par conséquent, selon les

mêmes raisons décrites dans le premier cas d'utilisation avec une application de type vidéoconférence, l'architecture Broker est la plus économique des architectures de Cloud Networking proposées, tout en assurant les exigences de qualité de service du CSU.

Les résultats obtenus et reportés dans la figure 56 montrent que le temps de réponse est bien contrôlé dans les scénarios (B-2/F-2 et B-3/F-3) contrairement au scénario de sélection statique (S) et ce grâce à la garantie de qualité de service de type IaaS offerte par ces scénarios. Cependant, en cas de violation, les résultats obtenus montrent que le Broker et le CSP_L (B-3/F-3) peuvent réagir et éviter ou réduire les violations (ex. Job 5/16/26/37) contrairement aux scénarios sans gestion autonome de ressources (B-2/F-2).

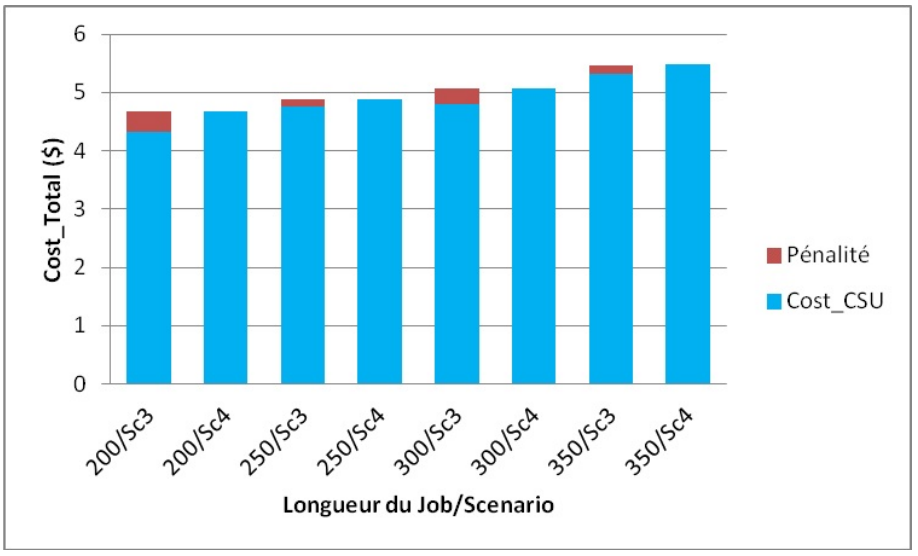

FIGURE 57 – Évaluation du coût global du CSU et de la pénalité.

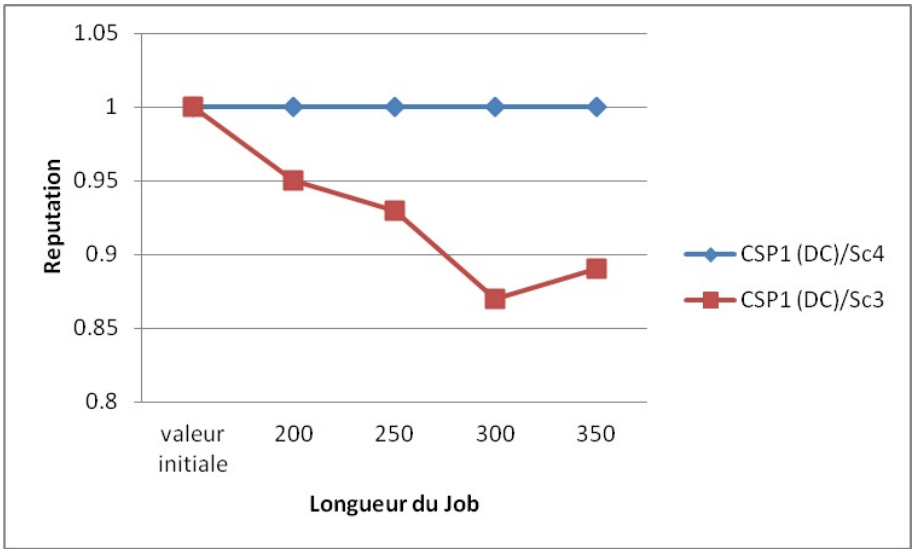

FIGURE 58 – Évaluation de la réputation pour le CSP_1 (DC).

Ainsi, comme l'illustre la figure 57, en utilisant notre architecture autonome de Cloud

Networking (quatrième scénario (B-3/F-3)), la pénalité globale est réduite ou évitée par rapport à l'architecture sans la gestion autonome des ressources (troisième scénario (B-2/F-2)). De plus, comme le montre la figure 58, la réputation du CSP_1 (DC) dans le quatrième scénario reste meilleure que celle obtenue dans le troisième scénario grâce à la réduction des violations et des pénalités.

6.7/ Conclusion

Dans un environnement de Cloud Networking, l'utilisation des concepts de l'Autonomic Computing pour la gestion des ressources et l'établissement autonome des SLA réduit les violations et minimise les interactions avec l'utilisateur. Dans ce chapitre, nous avons proposé une architecture pour l'établissement autonome des SLA et la gestion autonome des ressources du Cloud en utilisant des gestionnaires autonomes spécifiques de Cloud. Nous avons spécifiés nos gestionnaires autonomes (AM) ainsi que les interactions entre eux pour assurer un niveau de service pour le CSU tout en minimisant les violations. Ensuite, nous avons présenté les différents automates (FSM) et algorithmes proposés pour assurer un niveau de service de QoS dans l'architecture Broker ainsi que l'architecture Fédération. Dans cet environnement de gestion autonome, nous avons décrit les équations pour la détection et le calcul des violations, des pénalités, et des réputations.

Après l'étape de spécification de conception, nous avons présenté l'environnement de simulation pour la validation de notre proposition de gestion autonome des SLA et des ressources avec l'utilisation de deux types d'applications, la vidéoconférence et les calculs intensifs. Nous avons obtenu de bons résultats de performances et de coût qui valident notre proposition et montrent que l'architecture Broker est la plus économique par rapport à l'architecture Fédération tout en garantissant les exigences de qualité de service du CSU. De plus, nous avons obtenu de bons résultats pour les scénarios avec gestion autonome des ressources qui montrent l'intérêt de ce type de gestion. Ainsi, les violations sont évitées, les pénalités sont réduites, et les réputations sont meilleures qu'une architecture sans gestion autonome.

Dans le chapitre suivant, nous étendrons notre architecture de Cloud Networking pour offrir un niveau de service intégrant une offre de sécurité grâce à des SLA comportant des paramètres de sécurité. De plus, nous présenterons une architecture pour la distribution des certificats aux différentes entités et nous permettrons aux gestionnaires autonomes (AM) d'étendre leur gestion autonome aux aspects liés à la sécurité (Auto-protection) tout en étudiant l'impact de la sécurité sur la QoS.

CHAPITRE 7
SÉCURITÉ DES ARCHITECTURES DE CLOUD NETWORKING

7.1/ INTRODUCTION

L'utilisation fréquente du Cloud fait apparaître de nouveaux risques de sécurité tout en exposant les données des utilisateurs à ces risques et en suscitant l'intérêt des attaqueurs pour trouver de nouvelles vulnérabilités [186]. Par conséquent, pour atténuer ces vulnérabilités, les intervenants du Cloud devraient investir dans l'évaluation des risques de sécurité afin de s'assurer que les données sont bien protégées et d'améliorer la confiance dans cette nouvelle technologie [187]. De plus, un problème de sécurité dans le Cloud peut engendrer une perte économique, avoir un impact sur la QoS, et induire une mauvaise réputation pour le CSP. Ainsi, le CSU, en externalisant son infrastructure chez un CSP, a besoin de garanties de sécurité pour ses applications et ses données. Les problèmes de sécurité du Cloud peuvent être à l'origine d'un retard dans l'adoption massive de cette nouvelle solution [188].

Dans ce chapitre, nous étendons notre architecture de Cloud Networking pour offrir un niveau de service intégrant une offre de sécurité. Ce niveau est assuré grâce à des SLA comportant des paramètres de sécurité. De plus, nous présentons une architecture pour la distribution des certificats aux différentes entités pour assurer un niveau de confiance entre le CSU et les CSP. Enfin, nous permettons aux gestionnaires autonomes (AM) d'étendre leur gestion autonome aux aspects liés à la sécurité pour assurer l'autoprotection des données tout en étudiant l'impact de la sécurité sur la QoS.

7.2/ AMÉLIORATION DE L'ARCHITECTURE PROPOSÉE AVEC LA SÉCURITÉ

Afin de fournir le niveau de service attendu par le CSU, la bonne conception et la construction d'une architecture de Cloud Networking sont des défis très critiques. Dans cette section, nous proposons d'améliorer notre architecture de Cloud Networking (cf. section 4.2.1) et d'étendre les SLA (cf. section 4.2.2) proposés dans le cadre de cette architecture pour assurer la cohérence entre les exigences de qualité de service et de sécurité demandées par le CSU et celles offertes par les CSP et pour permettre à plusieurs CSP de collaborer ensemble afin de répondre aux exigences du CSU qui demande des services de types IaaS et/ou NaaS.

7.2.1/ Amélioration du niveau de service avec la sécurité

Dans le chapitre 5, nous avons proposé la sélection des ressources de type IaaS et/ou NaaS avec une garantie de QoS dans un environnement de Cloud Networking. Cependant, nous proposons dans ce chapitre d'étendre la garantie à la sécurité lors de la sélection des ressources. Ainsi, nous permettons au Cloud Broker dans l'architecture de type Broker (cf. section 4.3) et au CSP_L dans l'architecture de type Fédération (cf. section 4.4), d'assurer la sélection des ressources de Cloud de type IaaS et/ou NaaS avec une garantie de la sécurité. En effet, les CSP envoient leurs offres de services de type IaaS et/ou NaaS avec garantie de QoS et de sécurité. Différents niveaux de service (Platinum, Gold, Silver, ou Bronze) chacun avec différents paramètres de QoS et de sécurité sont associés aux offres de service des CSP.

Les niveaux de service intégrant uniquement une garantie de QoS sont présentés dans le chapitre 4 (cf. section 4.2.1). Cependant, nous spécifions dans ce qui suit pour les services de type NaaS et IaaS un niveau de service intégrant la sécurité.

Ainsi, nous caractérisons les services NaaS avec un niveau de service intégrant la sécurité par :

- **Service Level ID :** un identifiant unique du niveau de service.
- **NaaS Security Parameters :** les paramètres de sécurité (voir figure 59) qui peuvent être offerts d'une manière qualitative tels que l'authentification, le contrôle d'accès, la confidentialité, l'intégrité, la non-répudiation, et la mise à jour des systèmes de détection et de prévention des intrusions (IDPS : Intrusion Detection and Prevention Systems).
- **BW Cost :** le coût de la bande passante pour un service avec offre de sécurité. Il est spécifié par un coût unitaire de la bande passante consommée par le CSU. Cette bande passante sera facturée sur la base de son utilisation d'une manière sécurisée grâce aux services de sécurité.
- **Monitoring Interval Time :** l'intervalle du temps de surveillance pour calculer les violations et les pénalités durant la gestion autonome des ressources.
- **Validity Period :** une période de validité du niveau de service intégrant la sécurité.

```
<xs:element name="NaaS_Security_parameters" minOccurs="0" maxOccurs="1">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="Authentication" minOccurs="0" maxOccurs="1" type="Security_Qualitative_value"/>
            <xs:element name="Access_Control" minOccurs="0" maxOccurs="1" type="Security_Qualitative_value"/>
            <xs:element name="Confidentiality" minOccurs="0" maxOccurs="1" type="Security_Qualitative_value"/>
            <xs:element name="Integrity" minOccurs="0" maxOccurs="1" type="Security_Qualitative_value"/>
            <xs:element name="Non-repudiation" minOccurs="0" maxOccurs="1" type="Security_Qualitative_value"/>
            <xs:element name="IDPS_Update" minOccurs="0" maxOccurs="1" type="Security_Qualitative_value"/>
        </xs:sequence>
    </xs:complexType>
</xs:element>
```

FIGURE 59 – Représentation XML des paramètres de sécurité associés aux services NaaS.

De plus, pour les services de type IaaS (VM ou stockage) chaque niveau de service intégrant la sécurité est caractérisé par :

- **Service Level ID :** un identifiant unique du niveau de service.
- **IaaS Security Parameters :** les paramètres de sécurité (voir figure 60) qui

peuvent être offerts d'une manière qualitative tels que l'authentification, le contrôle d'accès, la confidentialité, l'intégrité, la non-répudiation, la mise à jour des systèmes de détection et de prévention des intrusions (IDPS), et la sauvegarde (Backup).

— **VM Cost/Storage Cost :** le coût unitaire d'utilisation des ressources de type VM ou stockage d'une manière sécurisée en accord avec l'attribut IaaS Security Parameters.
— **Monitoring Interval Time :** l'intervalle du temps de surveillance pour calculer les violations et les pénalités durant la gestion autonome des ressources.
— **Validity Period :** une période de validité du niveau de service intégrant la sécurité.

```
<xs:element name="IaaS_Security_parameters" minOccurs="0" maxOccurs="1">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="Authentication" minOccurs="0" maxOccurs="1" type="Security_Qualitative_value"/>
            <xs:element name="Access_Control" minOccurs="0" maxOccurs="1" type="Security_Qualitative_value"/>
            <xs:element name="Confidentiality" minOccurs="0" maxOccurs="1" type="Security_Qualitative_value"/>
            <xs:element name="Integrity" minOccurs="0" maxOccurs="1" type="Security_Qualitative_value"/>
            <xs:element name="Non-repudiation" minOccurs="0" maxOccurs="1" type="Security_Qualitative_value"/>
            <xs:element name="IDPS_Update" minOccurs="0" maxOccurs="1" type="Security_Qualitative_value"/>
            <xs:element name="Backup" minOccurs="0" maxOccurs="1" type="Security_Qualitative_value"/>
        </xs:sequence>
    </xs:complexType>
</xs:element>
```

FIGURE 60 – Représentation XML des paramètres de sécurité associés aux services IaaS.

Il est à noter que toutes les unités utilisées dans les représentations des schémas XML sont présentées dans l'annexe A.

De plus, les paramètres de sécurité spécifiés dans un niveau de service intégrant ce type de garantie sont définis comme suit :

— **Authentification :** ce paramètre décrit un niveau de garantie de bon fonctionnement relatif au mécanisme utilisé pour authentifier un utilisateur lors de l'accès à une ressource. Le niveau 1 est le niveau le plus bas de garantie et le niveau 4 est le plus haut niveau de garantie. Ce niveau de garantie peut être basé sur des standards pertinents tels que «NIST SP 800-63 (Electronic Authentication Guidelines)» [189] et «ISO/IEC 29115 (Entity Authentication Assurance Framework)» [190].
— **Contrôle d'accès :** ce paramètre décrit le degré de protection d'un service contre les accès non autorisés.
— **Confidentialité :** ce paramètre décrit la force d'une protection cryptographique appliquée à une ressource en fonction de sa longueur de la clé. Il existe de nombreuses méthodes de chiffrement qui varient selon leur force et leur coût.
— **Intégrité :** ce paramètre exprime la force d'une fonction de hachage ou encore de la fonction de type MAC (Message Authentication Code).
— **Non-répudiation :** ce paramètre exprime la force d'une signature et le niveau de traçabilité des données (données stockées, traitées, transférées, etc.).
— **La mise à jour des systèmes de détection et de prévention des intrusions (IDPS) :** ce paramètre décrit la période de temps entre deux mises à jour complètes pour ces systèmes (antivirus, etc.).
— **Sauvegarde :** ce paramètre décrit la période de temps entre les sauvegardes

complètes de données des CSU.

7.2.2/ Amélioration de l'interface utilisateur graphique avec la sécurité

Nous étendons l'interface utilisateur graphique (GUI) proposée dans le chapitre 4 (cf. section 4.2.3) afin de permettre au CSU de spécifier ces exigences en termes de sécurité. La figure 61 présente l'amélioration de la partie exigences du CSU en permettant au CSU de choisir le type de garantie requis (QoS et/ou sécurité). Ainsi, nous présentons dans la figure 62 l'ajout de la partie garantie de sécurité qui contient les paramètres de sécurité relatifs aux niveaux de service intégrant ce type de garantie pour les services de type IaaS et/ou de type NaaS.

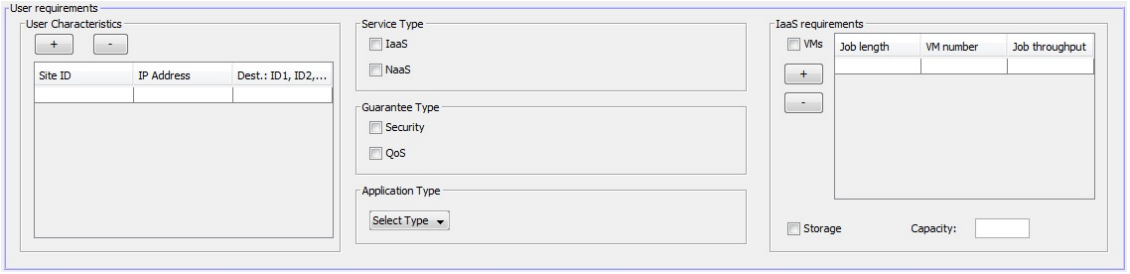

FIGURE 61 – Amélioration de l'interface GUI par la sécurité pour les exigences du CSU.

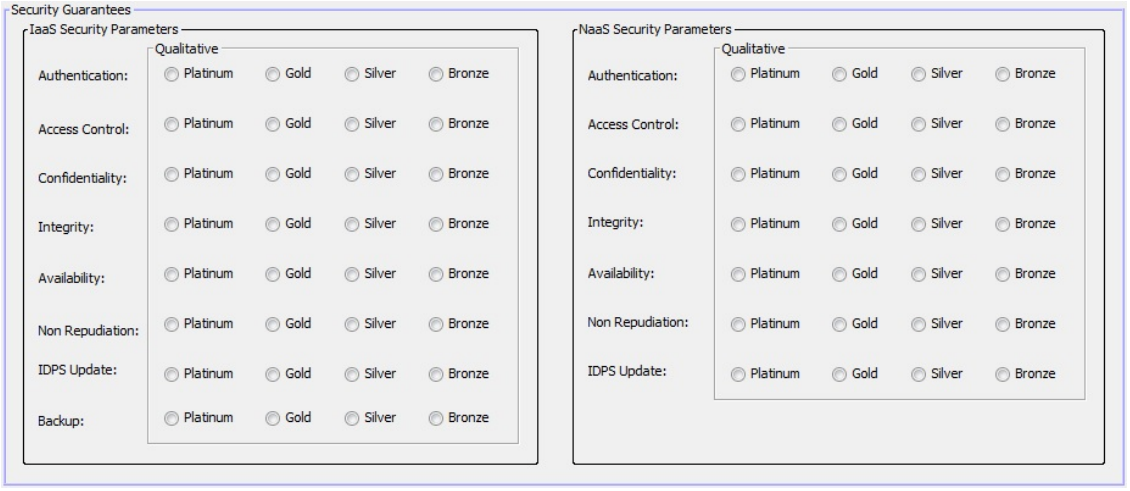

FIGURE 62 – Partie garantie de sécurité ajoutée dans l'interface GUI.

7.2.3/ Amélioration des SLA avec la garantie de sécurité

Dans notre architecture de Cloud Networking (cf. chapitre 4), l'accord de niveau de service (SLA) proposé (cf. section 4.2.2) ne comprend que les paramètres de qualité de service. Nous proposons dans ce chapitre l'extension de ce SLA par l'ajout des paramètres de sécurité pour garantir un niveau de service intégrant la sécurité offerte par les

CSP et demandée par le CSU. De plus, en se basant sur une caractéristique importante du Cloud à savoir «payez ce que vous utilisez», le SLA proposé doit garantir un niveau de service qui permet de satisfaire toutes les exigences du CSU en termes de QoS et de sécurité afin de ne payer que pour les ressources utilisées dans un environnement de Cloud Networking. Ainsi, nous reprenons les mêmes trois types de SLA (c.à.d. iSLA, B_iSLA et D_iSLA) et nous les étendons avec des paramètres de sécurité spécifiés en utilisant le langage XML pour l'interopérabilité et la portabilité entre les entités :

— **iSLA :** la nouvelle structure proposée du iSLA et présentée dans la figure 63 contient, en plus des attributs proposés dans la section 4.2.2, l'attribut «Service Security Guarantees» pour les garanties de sécurité offertes lors de l'utilisation du service correspondant. Ainsi, Comme le montre la figure 64, cet attribut contient les paramètres de sécurité qui seront garantis pour les services de type NaaS et de type IaaS. D'une part, nous définissons les paramètres de sécurité suivants pour l'offre de service de type NaaS : l'authentification, le contrôle d'accès, la confidentialité, l'intégrité, la non-répudiation, et la mise à jour des systèmes de détection et de prévention des intrusions. D'autre part, les paramètres de sécurité pour les services de type IaaS sont : l'authentification, le contrôle d'accès, la confidentialité, l'intégrité, la non-répudiation, la mise à jour des systèmes de détection et de prévention des intrusions, et la sauvegarde (Backup). De plus, nous proposons que le coût unitaire des ressources de type machines virtuelles, des espaces de stockage et de la bande passante prenne en considération le coût relatif non seulement à la garantie de QoS mais aussi de sécurité grâce au déploiement des mécanismes de QoS et de sécurité correspondants à ce niveau de service. Par la suite le CSU sera facturé sur la base de son utilisation des ressources (cf. section 7.2.4).

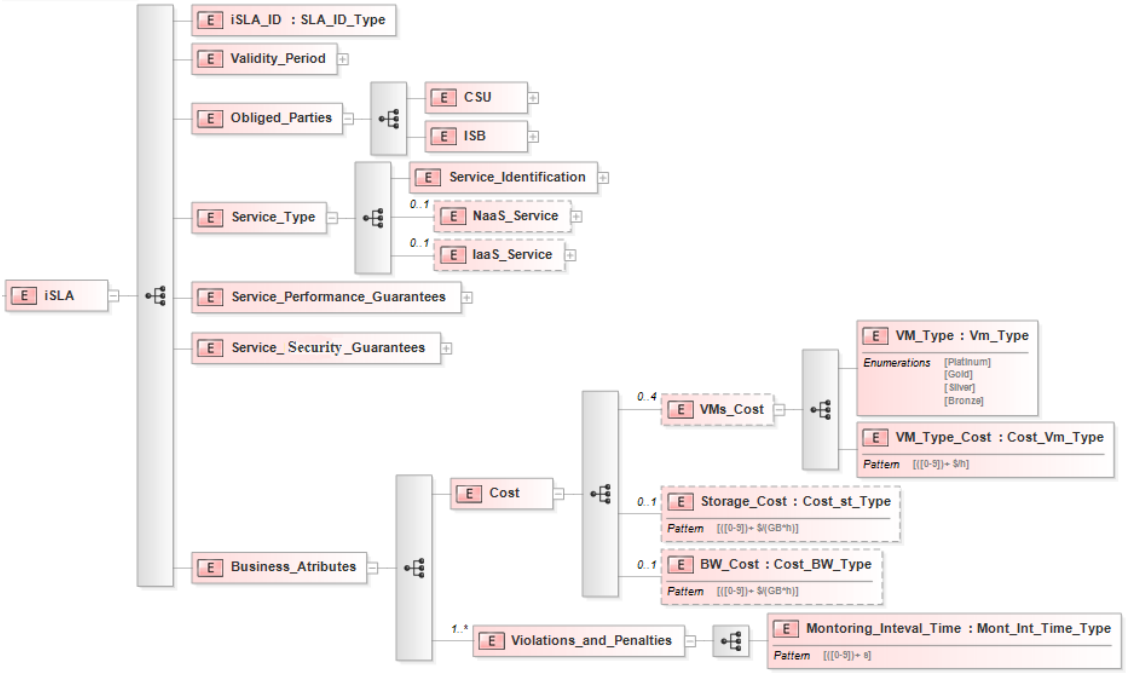

FIGURE 63 – Représentation du schéma XML de l'iSLA.

— **B_iSLA :** la nouvelle structure proposée du B_iSLA contient, en plus des attributs proposés dans la section 4.2.2 (voir figure 16), l'attribut «Service Security Gua-

```
<xs:element name="Service_Security_Guarantees">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="NaaS_Security_parameters" minOccurs="0" maxOccurs="1">
                <xs:complexType>
                    <xs:sequence>
                        <xs:element name="Authentication" minOccurs="0" maxOccurs="1" type="Security_Qualitative_value"/>
                        <xs:element name="Access_Control" minOccurs="0" maxOccurs="1" type="Security_Qualitative_value"/>
                        <xs:element name="Confidentiality" minOccurs="0" maxOccurs="1" type="Security_Qualitative_value"/>
                        <xs:element name="Integrity" minOccurs="0" maxOccurs="1" type="Security_Qualitative_value"/>
                        <xs:element name="Non-repudiation" minOccurs="0" maxOccurs="1" type="Security_Qualitative_value"/>
                        <xs:element name="IDPS_Update" minOccurs="0" maxOccurs="1" type="Security_Qualitative_value"/>
                    </xs:sequence>
                </xs:complexType>
            </xs:element>
            <xs:element name="IaaS_Security_parameters" minOccurs="0" maxOccurs="1">
                <xs:complexType>
                    <xs:sequence>
                        <xs:element name="Authentication" minOccurs="0" maxOccurs="1" type="Security_Qualitative_value"/>
                        <xs:element name="Access_Control" minOccurs="0" maxOccurs="1" type="Security_Qualitative_value"/>
                        <xs:element name="Confidentiality" minOccurs="0" maxOccurs="1" type="Security_Qualitative_value"/>
                        <xs:element name="Integrity" minOccurs="0" maxOccurs="1" type="Security_Qualitative_value"/>
                        <xs:element name="Non-repudiation" minOccurs="0" maxOccurs="1" type="Security_Qualitative_value"/>
                        <xs:element name="IDPS_Update" minOccurs="0" maxOccurs="1" type="Security_Qualitative_value"/>
                        <xs:element name="Backup" minOccurs="0" maxOccurs="1" type="Security_Qualitative_value"/>
                    </xs:sequence>
                </xs:complexType>
            </xs:element>
        </xs:sequence>
    </xs:complexType>
</xs:element>
```

FIGURE 64 – Représentation XML de l'attribut Service Security Guarantees.

rantees» pour les garanties de sécurité offertes lors de l'utilisation des services correspondants. Cependant, le B_iSLA ne garantit que la sécurité pour les services de type NaaS (BoD). Ainsi, l'attribut «Service Security Guarantees» contient seulement les paramètres de sécurité pour un service de type NaaS.

— **D_iSLA :** la nouvelle structure proposée du D_iSLA contient, en plus des attributs proposés dans la section 4.2.2 (voir figure 17), l'attribut «Service Security Guarantees» pour les garanties de sécurité offertes lors de l'utilisation des services correspondants. Cependant, ce type de contrat garantit la sécurité pour les services de type NaaS (DC) et/ou les services de type IaaS.

7.2.4/ AMÉLIORATION DES COÛTS INTÉGRANT LA QOS ET LA SÉCURITÉ

Dans cette section, nous étendons les coûts proposés dans le chapitre 5 (cf. section 5.4) afin de permettre le calcul du coût d'un service offert avec garantie non seulement de QoS mais aussi de sécurité. En effet, nous modifions les équations utilisées pour calculer les coûts en prenant en considération cette extension de sécurité. Ainsi, BW_cost_j sera le coût unitaire ($\$\, par\, GB$) du trafic qui traverse un CSP$_j$ (BoD ou DC) sélectionné avec une garantie d'un niveau de QoS et/ou de sécurité (voir équation (34)), VM_cost_{ij} sera le coût unitaire d'utilisation ($\$\, par\, heure$) d'un type de VM sélectionné (vt_{ij}) dans un CSP$_j$ (DC) avec un garantie d'un niveau de QoS et/ou de sécurité (voir équation (35)), st_cost_{ij} est le coût unitaire ($\$\, par\, GB\, par\, heure$) d'un type d'espace de stockage sélectionné (st_{ij}) dans un CSP$_j$ (DC) avec un garantie d'un niveau de QoS et/ou de sécurité (voir équation (36)).

$$BW\_cost_j = BW\_QoS_j + BW\_Sec_j \tag{34}$$

$$VM\_cost_{ij} = VM\_QoS_{ij} + VM\_Sec_{ij} \tag{35}$$

$$st\_cost_{ij} = st\_QoS_{ij} + st\_Sec_{ij} \tag{36}$$

Tel que BW_QoS_j, VM_QoS_{ij}, et st_QoS_{ij} sont les coûts unitaires avec une garantie de QoS pour l'utilisation respectivement de la bande passante, des machines virtuelles, et du stockage. De plus BW_Sec_j, VM_Sec_{ij}, et st_Sec_{ij} sont les coûts unitaires avec une garantie de sécurité pour l'utilisation respectivement de la bande passante, des machines virtuelles, et du stockage. Ainsi, si le CSU ne spécifie pas de garantie de QoS ou de sécurité, le coût unitaire relatif à cette garantie sera égal à 0.

7.3/ Amélioration des algorithmes de sélection proposés

7.3.1/ Contrainte et violation de sécurité

Dans le cadre de l'intégration de la garantie de sécurité dans le niveau de service que nous spécifions, nous proposons une nouvelle contrainte de sécurité (voir équation (37)) pour la sélection des meilleures ressources avec garantie de sécurité. Cette contrainte est définie comme suit : «le niveau de service (SL_i) du *i-ème* paramètre de sécurité requis par le CSU dans le iSLA, doit être inférieur ou égal ($\leq$) au niveau de service du même paramètre de sécurité (SL_i) offert par le CSP choisi (c_j)».

$$SL_i(iSLA) \leq SL_i(c_j) \tag{37}$$

De plus, la violation d'un paramètre de sécurité (c.à.d. non respect de la contrainte (37)) entraine l'occurrence d'une violation du SLA et par la suite le système enregistre les informations nécessaires pour calculer les pénalités appropriées. Pour ce faire, dans chaque CSP_j (DC/BoD), nous considérons le nombre d'intervalles (I_sec_j) de temps de surveillance au cours de l'exécution de l'application du CSU et le nombre de violations de sécurité (V_sec_j) qui ont eu lieu dans le CSP_j. Ainsi, dans le cadre de la garantie de sécurité, nous proposons une équation pour le calcul d'une partie des revenus du CSP (X_sec_j) (voir équation (38)) tout en considérant le cas de violation de sécurité dans l'évaluation de cette partie des revenus du CSP. Cette nouvelle équation est similaire à l'équation (28) du chapitre 6, qui permet de calculer une partie (X_j) des revenus du CSP_j pour la garantie de la QoS.

$$X\_sec_j = \frac{V\_sec_j}{I\_sec_j} \tag{38}$$

X_sec_j est la partie des revenus du CSP qui est considérée en cas de violation des paramètres de sécurité, V_sec_j le nombre de violation de sécurité qui a eu lieu dans un CSP_j, et I_sec_j est le nombre d'intervalles de temps de surveillance au cours de l'exécution de l'application du CSU.

Et par la suite, nous adaptons le calcul de la pénalité, du profit d'un CSP, du coût payé par le CSU, et de la réputation dans le cas de garantie de sécurité en utilisant les équations correspondantes du chapitre 6 (cf. équations de (29) à (33)) mais avec X_sec_j.

7.3.2/ Garantie de QoS et de sécurité

Dans le cas où le CSU demande une garantie de QoS et de sécurité pour les services de type IaaS et/ou NaaS, nous améliorons les algorithmes de 1 à 4 proposés dans le chapitre 5 (cf. section 5.3) par la prise en compte de la contrainte de sécurité (voir équation (37)) en plus des contraintes de QoS. Les changements qui en résultent concernent la ligne 5 dans l'algorithme 1, les lignes 27, 33, 51, 59 et 75 dans l'algorithme 2, la ligne 7 dans l'algorithme 3, et la ligne 5 dans l'algorithme 4. Ainsi, la condition suivante doit être vérifiée :

«if Constraints of QoS and Constraint (37) are met then»

De plus, nous améliorons la fonction d'utilité (voir équation (27)) pour les offres avec garantie de sécurité et de QoS comme présenté dans l'équation (39).

$$\psi_j = \frac{\psi\_QoS_j + \psi\_Sec_j}{2} \tag{39}$$

ψ_QoS_j est la valeur de la réputation du CSP$_j$ pour la garantie de QoS, et ψ_Sec_j est la valeur de la réputation du CSP$_j$ pour la garantie de sécurité.

7.3.3/ Garantie de sécurité seulement (c.à.d. sans QoS)

Dans le cas où le CSU demande une garantie de sécurité seulement (c.à.d. sans QoS) pour des services de type NaaS seulement (c.à.d. sans IaaS), nous améliorons l'algorithme 1 proposé dans le chapitre 5 (cf. section 5.3.1) grâce à la prise en compte de la contrainte de sécurité (voir équation (37)) mais sans prendre en considération les contraintes de QoS. De plus, dans le cas où le CSU demande une garantie de sécurité seulement (c.à.d. sans QoS) pour des services de type IaaS avec NaaS, nous améliorons l'algorithme 2 proposé dans le chapitre 5 (cf. section 5.3.2) grâce à la prise en compte de la contrainte de sécurité (voir équation (37)) mais sans prendre en considération les contraintes de QoS. Par conséquent, Les changements qui en résultent concernent la ligne 5 dans l'algorithme 1, et les lignes 27, 33, 51, 59 et 75 dans l'algorithme 2. Ainsi, la condition suivante doit être vérifiée :

«if Constraint (37) is met then»

De plus, nous proposons une fonction d'utilité pour les offres de sécurité seulement (c.à.d. sans QoS) qui est égale à la moyenne des réputations des CSP qui offrent la combinaison de niveau de service (sl) (voir équation (40)).

$$f(sl) = \sum_{j=1}^{nb} \frac{\psi_j}{nb} \tag{40}$$

ψ_j est la valeur de la réputation du CSP$_j$ qui offre un service avec un niveau de service (sl) intégrant une garantie de sécurité, et nb est le nombre de CSP qui offrent des services avec ce niveau de service sl.

Il est à noter que dans ce travail, nous ne considérons pas le cas où le CSU demande des services IaaS (ressources de type VM ou stockage) avec une garantie de sécurité seulement (c.à.d. sans QoS).

7.4/ Architecture pour la distribution des certificats de sécurité

En général, les CSU ne possèdent pas les capacités d'évaluer le niveau de sécurité mis en œuvre par les CSP. De plus, afin d'assurer le niveau de service de sécurité requis par le CSU et de fournir des contrôles de sécurité appropriés pour les environnements de Cloud, les exigences de sécurité du CSU doivent être clarifiées et précisées par une tierce partie qui a la capacité d'évaluer les niveaux de sécurité des CSP.

Dans cette section, nous proposons d'inclure une tierce partie de confiance (TTP : Trusted Third Party) avec une infrastructure à clé publique (PKI : Public Key Infrastructure) au sein de notre architecture de Cloud Networking afin d'assurer les services de sécurité et un niveau de confiance entre les CSU et les CSP.

7.4.1/ Tierce partie de confiance (TTP)

Une TTP est une entité qui facilite les interactions sécurisées entre deux ou plusieurs parties qui font confiance à cette tierce partie. L'utilisation d'une TTP peut contribuer à l'élimination des frontières de sécurité traditionnelles dans un environnement d'Inter-Cloud. En effet, une TTP permet non seulement de produire des domaines de sécurité et de confiance, mais aussi une coopération entre ces domaines, qui peuvent être par exemples des CSP, tout en établissant des interactions sécurisées. Par conséquent, nous proposons l'ajout d'une TTP au sein de notre architecture de Cloud Networking afin de contribuer à la création d'un niveau de confiance et à la fourniture des services de sécurité de bout en bout pour les données des CSU. De plus, nous proposons que la TTP utilise un système de distribution de certificat numérique de sécurité en incluant une infrastructure à clé publique (PKI). Cette PKI contient une Autorité de Certification (AC) qui est chargée de générer les certificats requis tout en les enregistrant dans un annuaire. Une entité peut communiquer avec l'AC d'une manière sûre pour obtenir un certificat. Par conséquent, l'infrastructure de type PKI aide à la construction d'un domaine de sécurité pour les entités de Cloud. Ainsi, la PKI fournit des moyens techniquement et légalement acceptables pour mettre en œuvre une authentification forte, une confidentialité et une intégrité des données, et une non-répudiation [113]. Dans ce contexte, La norme X.509 [191] est devenue universellement acceptée pour le formatage des certificats dans les infrastructures de type PKI. X.509 définit des protocoles d'authentification basés sur l'utilisation de certificats à clés publiques. De plus, X.509 est basée sur l'utilisation de la cryptographie à clé publique et les signatures numériques et n'impose pas un algorithme de chiffrement ou de hachage spécifique. Ainsi, chaque certificat possède un bloc qui contient la clé publique de l'entité et son ID, une période de validité, l'algorithme de chiffrement et la fonction de hachage utilisés pour signer le certificat. L'ensemble de ce bloc est signé par l'AC. Les certificats X.509 sont utilisés dans la plupart des applications et des protocoles de sécurité, y compris le protocole IP security (IPsec), Secure Sockets Layer (SSL), etc.

Par conséquent, nous proposons que la tierce partie soit une autorité de certification (AC) approuvée par la communauté. Toute entité dans notre environnement de Cloud Networking qui a besoin de vérifier une clé publique d'une autre entité peut obtenir son certificat et vérifier qu'il est valide grâce à la signature faite par l'autorité de confiance. De plus, nous proposons que la TTP soit associée au Cloud Broker dans l'architecture

de type Broker. Cependant, dans l'architecture de type Fédération, la TTP est une entité distincte dans l'alliance et que nous ajoutons à ce type d'architecture.

7.4.2/ Architecture de distribution des certificats

Dans cette section, nous proposons que les CSP (DC / BoD) soient connectés opérationnellement à travers des chaînes de confiance, habituellement appelés chemins de certificat, formant la notion d'une infrastructure à clé publique (PKI) (voir figure 65).

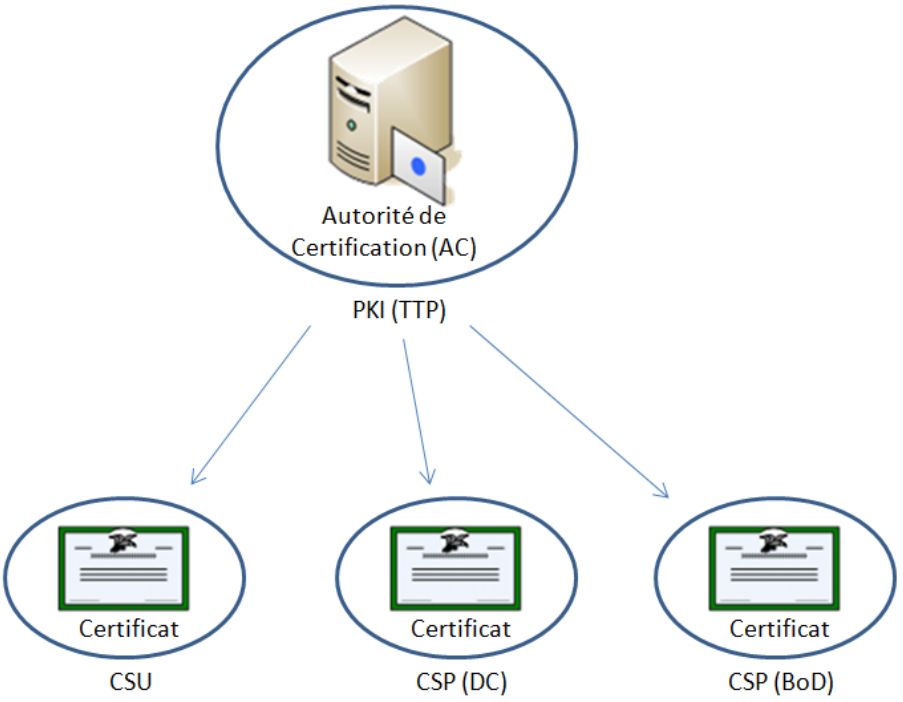

Figure 65 – Architecture de distribution des certificats.

Nous présentons dans la figure 65 une architecture pour la distribution des certificats pour les CSU et les CSP dans un environnement de Cloud Networking. En effet, un CSU a besoin d'un certificat numérique pour s'authentifier et valider ses droits d'accès afin d'utiliser les ressources d'un service de Cloud avec une garantie de sécurité pour les services de type NaaS ou IaaS. Ainsi, le CSU avec son certificat (clé publique) peut chiffrer ces données transmises dans les réseaux, stockées dans les centres de données des CSP, ou traitées dans les machines virtuelles (en utilisant le chiffrement homomorphique). De plus, le CSU avec la partie privée correspondante à son certificat (clé privée) peut signer ces données pour un niveau de non-répudiation et d'intégrité.

D'autre part, un CSP (BoD) a besoin d'un certificat numérique afin d'offrir une garantie de sécurité pour les services de type NaaS. Ce certificat peut être utilisé pour offrir une communication sécurisée entre les sites du CSU, entre le CSU et les CSP (DC) ou entre les CSP (DC). Ainsi, ce certificat peut être utilisé pour offrir le chiffrement et la signature des données échangées du CSU afin de garantir leur sécurité. Par exemple, les réseaux privés virtuels (VPN : Virtual Private Network) réalisés en utilisant IPSec au niveau de la couche réseau ou encore en utilisant SSL au niveau de la couche transport présentent une bonne solution pour assurer les services de sécurité en utilisant ces certificats [112].

De plus, les VPN peuvent être utilisés dans un environnement de Cloud Networking pour transporter un trafic de CSU authentifié selon un niveau de sécurité donné (Gold par exemple). Par conséquent, les VPN permettent de simplifier l'application de la sécurité, la QoS et le SLA d'une façon générale dans notre architecture de Cloud Networking.

Ainsi, le trafic du CSU peut être sécurisé à différents niveaux. Au niveau du transport, le protocole SRTP (Secure Real-time Transport Protocol) [192] et/ou le protocole DTLS (Datagram Transport Layer Security) [193] peuvent être utilisés. Au niveau du réseau le protocole IPsec (IP security) [194] peut être utilisé. Il permet de sécuriser les communications grâce à l'authentification et le chiffrement des paquets de données. A ce titre, IPsec offre des services de sécurité au niveau de la couche réseau en permettant à un système de sélectionner les protocoles de sécurité nécessaires (AH :Authentication Header, ESP : Encapsulating Security Payload), mais aussi de déterminer les algorithmes (de chiffrement par exemple) à utiliser, et de mettre en place toutes les clés de chiffrement nécessaires pour fournir les services de sécurité demandés avec le niveau de service correspondant. Les utilisateurs IPsec sont capables de s'authentifier en utilisant des certificats délivrés par une PKI. IPsec englobe plusieurs domaines fonctionnels [113] : l'authentification, l'intégrité, la confidentialité et la gestion des clés. Le mécanisme d'authentification utilise le HMAC (keyed Hash Message Authentication Code) et s'assure qu'un paquet reçu a été transmis par la partie identifiée comme source dans l'entête du paquet. L'authentification peut être appliquée sur tout le paquet IP d'origine (mode tunnel) ou l'ensemble du paquet à l'exception de l'entête IP (mode transport). Ainsi, ce mécanisme (HMAC) s'assure en plus que le paquet n'a pas été modifié en transit (intégrité). La confidentialité est assurée grâce à la capacité donnée aux entités de l'environnement de Cloud de chiffrer les messages afin d'éviter les écoutes. La gestion des clés concerne leur échange sécurisé en utilisant le protocole IKE (Internet Key Exchange) qui est une collection de documents décrivant les systèmes de gestion des clés pour une utilisation avec IPsec. À tout moment de l'échange IKE, l'expéditeur peut demander le certificat de l'autre entité réceptrice. De plus, pour assurer l'interopérabilité, les implémentations de la suite de protocoles IPsec partagent un ou plusieurs algorithmes de sécurité [195].

Contrairement à un CSP (BoD), un CSP (DC) a besoin d'un certificat numérique pour offrir une garantie de sécurité, non seulement, pour les services de type NaaS mais aussi pour les services de type IaaS. Ce certificat peut être utilisé pour offrir une communication sécurisée entre les différentes ressources dans un centre de données ou enore un accès sécurisé aux ressources du CSU dans le centre de données du CSP. De plus, le CSP (DC) peut chiffrer les données du CSU (chiffrement homomorphique par exemple) mais aussi signer ces données stockées dans son centre de données. Enfin, les CSP (DC/BoD) peuvent déployer des systèmes de détection et de prévention des intrusions (IDPS) tels qu'un pare-feu, un anti-virus, un système de détection et de prévention des attaques malveillantes. Ainsi, selon les exigences du CSU en termes de services de sécurité, les CSP choisissent les protocoles de sécurité et les algorithmes convenables pour fournir ces services de sécurité.

7.5/ Intégration de la sécurité dans l'architecture autonome de Cloud

Actuellement, Dans la majorité des architectures déployées, la sécurité et la qualité de service ne sont pas gérées ensemble, mais sont plutôt mises en œuvre et gérées séparément. Cependant, ces deux mécanismes sont étroitement liés et par conséquent leur gestion doit être faite conjointement car la sécurité peut avoir un impact sur la garantie de la QoS.

7.5.1/ Impact de la sécurité sur la qualité de service

La sécurité peut avoir un grand impact sur les paramètres de qualité de service. Cet impact peut s'exprimer en termes de retard et de surcharge du réseau (Overhead). Ainsi, les opérations cryptographiques pour le chiffrement et le déchiffrement des données du CSU, la signature de ces données, et le stockage et la récupération des informations de sécurité conduisent à un délai supplémentaire. En effet, ces opérations entrainent une augmentation du trafic dans le réseau, responsable d'une latence supplémentaire et une monopolisation des capacités du processeur pour le traitement correspondant. De plus, elles conduisent à une bande passante supplémentaire nécessaire pour les entêtes ajoutés aux paquets.

L'impact de la sécurité sur la qualité de service est plus ou moins important en fonction de l'ensemble des services de sécurité sélectionnés et des algorithmes utilisés pour fournir ces services. Ainsi, l'intégrité a un impact sur l'utilisation de la bande passante, car les données d'authentification sont ajoutées aux paquets originaux. De même, la confidentialité nécessite des opérations cryptographiques telles que le chiffrement et le déchiffrement. Par conséquent, cet impact de la sécurité sur la QoS nécessite que ces deux attributs de niveau de service soient soigneusement et étroitement gérés d'une façon globale et non pas abordés séparément dans notre architecture autonome de Cloud Networking.

7.5.2/ Intégration de la sécurité dans les gestionnaires autonomes

Dans cette section, nous proposons d'adapter notre architecture autonome de Cloud Networking, spécifiée dans le chapitre 6 (cf. section 6.3), afin d'auto-établir les SLA et d'auto-gérer les ressources de Cloud de type IaaS et NaaS avec garantie de qualité de service et de sécurité grâce à l'utilisation de gestionnaires autonomes de Cloud (AM). Par conséquent, un iAM offre des possibilités de communication de haut niveau avec d'autres iAM pour conclure un accord de niveau de service correspondant à des ressources sélectionnées en fonction du coût optimal et de la garantie de sécurité (auto-protection) et de qualité de service (auto-optimisation). De plus, le gestionnaire iAM contrôle un ou plusieurs gestionnaires autonomes de bas niveau (nAM, DnAM et hAM) pour configurer les ressources allouées (auto-configuration) et pour éviter les violations (auto-restauration) en conformité avec le niveau de service convenu. Ces gestionnaires autonomes de bas niveau sont responsables de la surveillance des ressources et de leurs performances et sécurité en conformité avec les SLA établis (B_iSLA et D_iSLA).

Chaque iAM peut faire une évaluation des mécanismes de sécurité dans son domaine

autonome (AD : Autonomic Domain) à l'aide des gestionnaires autonomes de bas niveau suivant le deuxième type d'interactions (verticale) décrit dans le chapitre 6 (cf. section 6.3.3). L'évaluation permet de vérifier que les mécanismes de sécurité sont correctement mis en œuvre, qu'ils fonctionnent comme prévu, et qu'ils produisent le résultat souhaité pour satisfaire les exigences de sécurité pour le CSU et le respect de toutes les réglementations. De plus, la tierce partie de confiance (TTP) peut faire une évaluation périodique des CSP et mettre à jour leurs réputations.

Les automates finis (FSM : Finite State Machine) pour l'auto-établissement des SLA dans l'architecture de type Broker (cf. section 6.4.1) et de type Fédération (cf. section 6.4.2) pour les services de types IaaS et NaaS peuvent être réutilisés pour une garantie d'un niveau de service intégrant la QoS et/ou la sécurité en utilisant les algorithmes qui prennent en considération l'amélioration évoquée au début de ce chapitre (algorithmes 1 à 4 modifiés pour une garantie de QoS et de sécurité, et algorithmes 1 et 2 modifiés pour une garantie de sécurité seulement). Dans ce contexte, nous proposons d'améliorer les gestionnaires autonomes de Cloud pour la gestion de l'impact de la sécurité sur la QoS afin de réduire cet impact, dans le cas où le CSU demande une garantie de QoS et de sécurité. Par conséquent, à la réception d'une notification de violation ou de risque de violation d'un paramètre de QoS depuis un AM de bas niveau sous son contrôle, le iAM d'un CSP essaye dans un premier temps de diminuer la longueur de la clé utilisée par le chiffrement et la signature pour assurer la confidentialité et l'intégrité. Le iAM du CSP envoie une notification de diminution de la longueur de la clé vers le AM de bas niveau. Cependant, cette diminution de la longueur de la clé ne doit pas entrainer un niveau de sécurité moins important que celui qui est demandé par le CSU, c.à.d. il faut toujours respecter la contrainte (37). Par contre, si le problème n'est pas résolu grâce à cette diminution de la longueur de la clé, le iAM du CSP va suivre les mêmes étapes décrites dans le chapitre 6 (cf. section 6.4.4). Enfin, dans tous les cas, le iAM du CSP met à jour les paramètres de sécurité et de QoS dans le niveau de service offert pour que la sécurité n'impacte pas la QoS. Ainsi, nous pouvons avoir une diminution de la longueur de la clé ou bien une augmentation de la valeur de la latence, de la gigue, ou de la bande passante.

7.6/ Validation de l'offre de sécurité dans l'architecture Cloud Networking

Pour valider notre architecture améliorée de Cloud Networking, nous présentons dans cette section un cas d'utilisation pour l'établissement efficace et sécurisé des applications de Cloud vidéoconférence. Nous évaluons l'impact de la sécurité sur la QoS grâce à différentes simulations en utilisant la boîte à outils CloudSim [33].

7.6.1/ Environnement de simulation

Nous utilisons l'environnement de simulation décrit dans le chapitre 5 et le chapitre 6. Ainsi, nous implémentons les différentes modifications et ajouts des algorithmes proposés dans ce chapitre dans cet environnement de simulation. Dans ce contexte, le CSU spécifie son type de service demandé et les exigences correspondantes en termes de QoS et de sécurité au Cloud Broker ou au CSP_L en utilisant l'interface graphique proposée. Ensuite, après la sélection des meilleurs CSP, le Cloud Broker ou le CSP_L alloue

les ressources nécessaires aux sites du CSU et établit d'une façon autonome les différents SLA proposés. Ainsi, les entités participantes lancent les applications, puis nous évaluons l'impact de la sécurité sur la QoS en fonction de la prise en compte ou non des concepts de la gestion autonome adoptés dans notre architecture de Cloud Networking.

Nous présentons dans la table 16 les niveaux de services de sécurité offerts par un CSP (DC/BoD) et les coûts unitaires correspondants. Chaque colonne de la table décrit les coûts unitaires relatifs à un niveau de service de sécurité (Platinum, Gold, Silver, Bronze) pour les services de types NaaS (réseau) et IaaS (stockage et VM).

	Bronze	Silver	Gold	Platinum
Réseau	0.05 $ par GB	0.11 $ par GB	0.15 $ par GB	0.19 $ par GB
VM	0.1 $ par heure	0.2 $ par heure	0.3 $ par heure	0.4 $ par heure
Stockage	0.5×10^{-4} $ par GB par heure	10^{-4} $ par GB par heure	1.6×10^{-4} $ par GB par heure	2×10^{-4} $ par GB par heure

TABLE 16 – Coût des niveaux de service de sécurité offerts par un CSP (DC/BoD).

7.6.2/ CLOUD VIDÉOCONFÉRENCE

7.6.2.1/ SCÉNARIO DE SIMULATION

Dans ce cas d'utilisation avec l'application de Cloud vidéoconférence, nous évaluons trois scénarios de simulation :

1- Le premier scénario correspond à une sélection des meilleures ressources en se basant sur l'architecture Broker (Bs-1) et sur l'architecture Fédération (Fs-1) avec établissement autonome des SLA et avec garantie de QoS seulement (c.à.d. sans sécurité).

2- Le deuxième scénario correspond à une sélection des meilleures ressources en se basant sur l'architecture Broker (Bs-2) et sur l'architecture Fédération (Fs-2) avec établissement autonome des SLA et avec garantie de QoS et de sécurité mais sans gestion autonome des ressources. Ainsi, le système ne réagit pas en cas de violation.

3- Le troisième scénario est le même que le deuxième, mais avec gestion autonome des ressources (Bs-3/Fs-3).

Dans chaque scénario, nous avons plusieurs sites de CSU connectés à différents CSP (BoD) qui utilisent des services de type IaaS et/ou NaaS dans l'architecture Broker et l'architecture Fédération. Le but est de sélectionner les meilleures routes de distribution des flux de vidéoconférence en passant par des machines virtuelles allouées pour les serveurs de vidéoconférence. Ensuite, le système gère d'une façon autonome les ressources pour éviter ou réduire les violations de QoS qui peuvent être causées par la garantie de la sécurité.

Le CSU spécifie une latence réseau maximale de 180 ms, une gigue réseau maximale de 30 ms et un temps de réponse maximal de transcodage de 20 ms. Pour les autres paramètres de qualité de service, le CSU demande un niveau de service Gold et tous les poids sont égaux à 1. De plus, le CSU demande pour tous les paramètres de sécurité un niveau Gold. Nous simulons quatre types de vidéos envoyées par le CSU avec une longueur d'une heure, une taille de respectivement 1, 2, 3 et 4 Go, et une bande

passante demandée égale à respectivement à 2.2Mb/s, 4.5Mb/s, 6.8Mb/s et 9.1Mb/s. Après plusieurs simulations pour chaque type de vidéo, avec à chaque fois une distribution géographique différente des sites du CSU, nous calculons au début le coût global de la bande passante (voir figure 66) et le coût global de VM (voir figure 67) de chaque type de vidéo tout en comparant le premier scénario (Bs-1/Fs-1) avec le deuxième scénario (Bs-2/Fs-2). Ensuite, nous calculons la latence moyenne de bout en bout sans le temps de réponse (voir figure 68) pour les différents types de vidéos simulées, avec une période d'intervalle d'échantillonnage d'une minute, pour faire une comparaison entre le premier scénario (Bs-1/Fs-1), le deuxième scénario (Bs-2/Fs-2) et le troisième scénario (Bs-3/Fs-3).

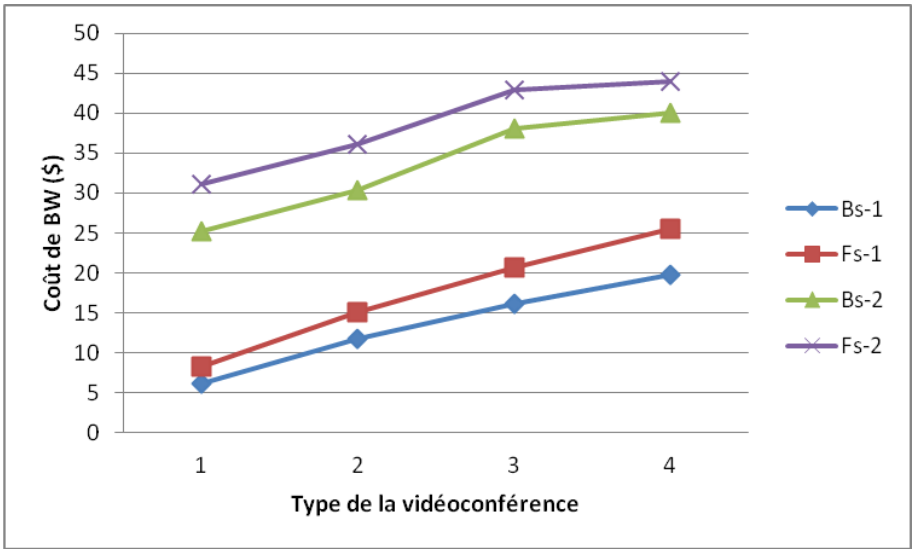

FIGURE 66 – Évaluation du coût global de la bande passante.

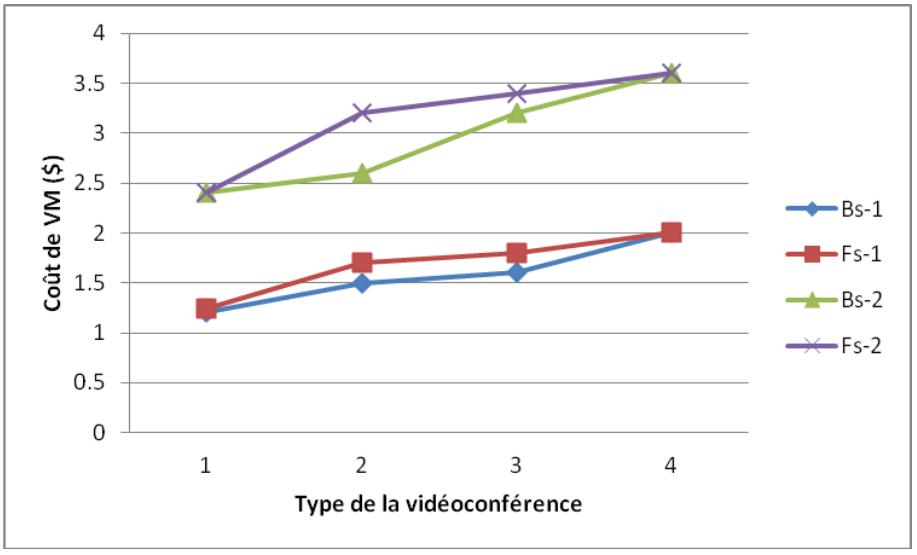

FIGURE 67 – Évaluation du coût global de VM.

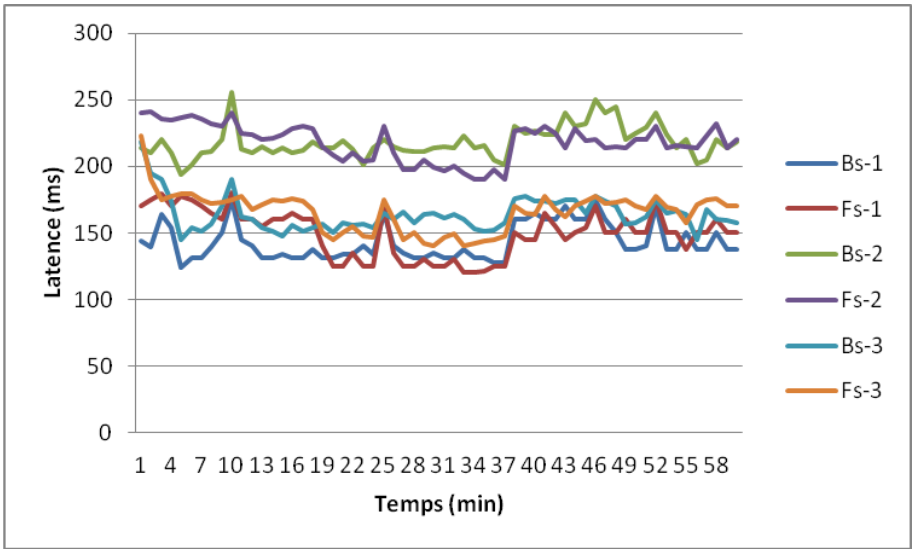

FIGURE 68 – Évaluation de la latence moyenne de bout en bout.

7.6.2.2/ RÉSULTATS

Les figures 7.8 et 7.9 montrent que le coût global de la bande passante et des VM augmente lorsque la taille de la vidéo et la bande passante correspondante augmentent. De plus, le coût global de la bande passante et des VM correspondant à une sélection basée sur l'architecture Broker (Bs-1) et l'architecture Fédération (Fs-1) avec une garantie de QoS seulement est inférieur à celui de la sélection basée sur l'architecture Broker (Bs-2) et l'architecture Fédération (Fs-2) avec une garantie de QoS et de sécurité. Cependant, le coût global de la bande passante et des VM dans le scénario de l'architecture Broker est moins important que le coût de la bande passante et des VM dans le scénario de l'architecture Fédération. Ces résultats sont obtenus en raison de la sélection des ressources disponibles au sein du CSP_L d'abord, puis la sélection des ressources restantes au sein des autres CSP. En effet, les ressources du CSP_L ne sont pas obligatoirement les meilleures en termes de coût. Ainsi, nous remarquons que l'architecture Broker est la plus économique, tout en assurant les exigences de qualité de service et de sécurité du CSU.

D'autre part, comme illustré dans la figure 7.10, dans le premier scénario (Bs-1/Fs-1) nous remarquons que les valeurs de la latence sont inferieures ou égales à la valeur maximale de latence exigée par le CSU (180 ms). Cependant, dans le deuxième scénario (Bs-2/Fs-2) nous remarquons que lorsque le CSU demande en plus une garantie de sécurité sur ces données, les valeurs de la latence augmentent considérablement (plus que 180 ms) (ex. aux instants 10, 16, 40 et 55 minutes) à cause des opérations cryptographiques pour le chiffrement et le déchiffrement des données du CSU. Par conséquent, ces résultats obtenus montrent qu'en cas de garantie de QoS et de sécurité, un impact des mécanismes de sécurité sur les paramètres de QoS est observé. De plus, en cas de violation de la valeur maximale de latence exigée par le CSU (180 ms) (ex. aux instants 1, 2 et 10 minutes), le Cloud Broker (Bs-3) et le CSP_L (Fs-3) peuvent réagir afin d'éviter ou réduire les violations causées par l'impact de la sécurité sur la QoS (ex. aux instants 3 et 11 minutes) contrairement aux scénarios Broker et Fédération sans gestion autonome

des ressources (Bs-2/Fs-2).

7.7/ Conclusion

Dans ce chapitre, nous avons étendu notre architecture de Cloud Networking pour offrir des services de types IaaS et NaaS avec une garantie d'un niveau de service intégrant non seulement une garantie de QoS mais aussi une garantie de sécurité demandée par le CSU grâce à des SLA comportant des paramètres de sécurité. Ainsi, nous avons amélioré l'interface d'utilisateur graphique pour permettre au CSU de renseigner ses exigences de sécurité. De plus, nous avons proposé une amélioration et une extension des algorithmes pour la sélection des ressources avec une garantie de QoS et/ou de sécurité. Ensuite, nous avons présenté une architecture pour la distribution des certificats aux différentes entités de notre environnement de Cloud Networking pour assurer un niveau de confiance entre le CSU et les CSP. De plus, nous avons étendu les gestionnaires autonomes de Cloud pour assurer une gestion autonome des aspects liés à la sécurité (Auto-protection). Ainsi, nous avons étudié l'impact de la sécurité sur la QoS dans le cas d'utilisation d'une application de Cloud vidéoconférence. Nous avons obtenu de bons résultats pour les scénarios avec gestion autonome de la sécurité qui montre que l'impact de la sécurité sur la QoS est réduit par rapport à une architecture sans gestion autonome.

Dans le chapitre suivant, nous présenterons la conclusion générale de cette thèse ainsi que les perspectives liées aux travaux proposés.

CHAPITRE 8
CONCLUSION GÉNÉRALE

8.1/ BILAN

Dans cette thèse, nos travaux de recherche ont porté sur la conception et l'implémentation d'un Framework pour la garantie de bout en bout de la QoS ainsi que la sécurité en se basant sur un accord de niveau de service (SLA) dans le cadre d'un environnement de Cloud Computing, de Cloud Networking et d'Inter-Cloud. De plus, les architectures de type Broker ou encore Fédération qui valident ce Framework sont gérées d'une façon autonome grâce aux concepts de l'Autonomic Computing.

Dans un premier temps, nous avons réalisé une présentation générale du paradigme de Cloud Computing, ses caractéristiques, ses modèles de services et de déploiement, les travaux de standardisation, ainsi que les outils d'implémentation et de simulation et les produits commerciaux relatifs aux services du Cloud. De plus, nous avons présenté le nouveau paradigme de Cloud Networking qui est défini comme une évolution du Cloud Computing pour offrir des ressources réseaux à la demande. Nous avons décrit ses caractéristiques, et les outils de simulation et d'implémentation relatifs à ce nouveau concept. Dans ce contexte, nous avons étudié l'approche Inter-Cloud avec ses caractéristiques et ses modèles de déploiement.

Ensuite, nous avons réalisé un état de l'art sur trois concepts importants qui constituent des défis de recherche dans un environnement de Cloud Computing à savoir la QoS, la sécurité et la gestion autonome. Ces défis ont été étudiés dans un contexte de Cloud et nous avons présenté les travaux de standardisation en cours de finalisation, ainsi que les projets et les travaux de recherche effectués dont l'objectif est de proposer des solutions face à ces défis.

Dans le cadre de cette thèse, nous avons proposé une première contribution dont l'objectif est d'apporter des solutions face à ces défis afin de permettre aux fournisseurs de Cloud (CSP) d'offrir des services de types IaaS et NaaS avec une garantie d'un niveau de service décrivant la QoS demandée par leurs utilisateurs (CSU). Ainsi, nous avons proposé deux architectures de Cloud Networking. La première architecture est basée sur l'approche Inter-Cloud de type Broker et la deuxième est basée sur l'approche Inter-Cloud de type Fédération. De plus, nous avons proposé différents types de contrat de niveau de service (iSLA, B_iSLA et D_iSLA) pour décrire la garantie de QoS relative aux services de type IaaS et NaaS. Nous avons utilisé ces contrats dans les deux types d'architecture afin de garantir une QoS de bout en bout lors de l'utilisation par les CSU des services offerts par les CSP. De plus, nous avons proposé une interface d'utilisateur graphique pour permettre au CSU de renseigner ses exigences de QoS et de spécifier les types de service qu'il veut utiliser à la demande.

Une deuxième contribution concerne la spécification des algorithmes de sélection pour

déterminer les meilleurs CSP permettant de répondre aux exigences du CSU en termes de QoS relative à l'offre de service de type IaaS et/ou NaaS. Ainsi, nous avons proposé un premier algorithme pour sélectionner les ressources réseau relatives à un service de type NaaS uniquement. Par la suite, nous avons proposé trois autres algorithmes pour la sélection des ressources réseau, de machines virtuelles (VM), et de stockage, pour un service de type IaaS avec/sans NaaS. Ces algorithmes permettent de déterminer la meilleure offre avec le coût le plus faible. Dans le cas où plusieurs offres équivalentes (QoS satisfaite et même coût minimal) sont retenues, nous avons proposé l'utilisation d'une fonction d'utilité en se basant sur la QoS offerte par les CSP et celle demandée par le CSU afin de sélectionner l'offre qui maximise cette fonction d'utilité. De plus, nous avons proposé une méthodologie pour calculer le coût pour chaque type de service (NaaS et IaaS (VM et stockage)).

Ensuite, nous avons simulé nos architectures de type Broker et Fédération, et implémenté les algorithmes de sélection proposés afin d'assurer la QoS de bout en bout pour les services de types NaaS et IaaS pour deux types d'applications, à savoir la vidéoconférence et le calcul intensif. Nous avons obtenu de bons résultats de performances qui valident nos deux premières contributions et montrent que l'architecture de type Broker est la plus économique par rapport à l'architecture Fédération pour l'offre des services NaaS et/ou IaaS tout en garantissant les exigences de qualité de service du CSU.

Une troisième contribution concerne la proposition d'une architecture pour l'établissement autonome des SLA et la gestion autonome des ressources du Cloud en utilisant des gestionnaires autonomes spécifiques à cet environnement. Nous avons spécifié nos gestionnaires autonomes ainsi que les interactions entre eux pour assurer un niveau de service de bout en bout pour le CSU tout en minimisant les violations. Ensuite, nous avons présenté les différents automates et algorithmes proposés pour garantir la QoS dans l'architecture de type Broker ainsi que l'architecture de type Fédération. Dans cet environnement de gestion autonome, nous avons décrit les équations pour la détection et le calcul des violations, des pénalités, et des réputations.

Ensuite, nous avons présenté l'environnement de simulation pour la validation de cette troisième contribution concernant la gestion autonome des SLA et des ressources en prenant les cas d'utilisation de deux types d'applications, la vidéoconférence et les calculs intensifs. Nous avons obtenu de bons résultats de performances et de coût qui valident notre proposition et montrent que l'architecture Broker est la plus économique par rapport à l'architecture Fédération tout en garantissant les exigences de qualité de service du CSU. De plus, nous avons obtenu de bons résultats pour les scénarios avec gestion autonome des ressources qui montrent l'intérêt de ce type de gestion. Ainsi, les violations sont évitées, les pénalités sont réduites, et les réputations sont meilleures que celles d'une architecture sans gestion autonome.

Une quatrième contribution concerne l'extension de notre architecture de Cloud Networking pour offrir des services de types IaaS et NaaS avec une garantie d'un niveau de service décrivant la sécurité demandée par le CSU grâce à des SLA comportant des paramètres de sécurité. Ainsi, nous avons présenté une architecture pour la distribution des certificats aux différentes entités de l'environnement de Cloud pour assurer un niveau de confiance entre les CSU et les CSP. Nous avons amélioré l'interface d'utilisateur graphique pour permettre au CSU de renseigner ses exigences de sécurité. De plus, nous avons étendu les gestionnaires autonomes de Cloud pour assurer une gestion autonome qui prend en considération la fonction relative à la sécurité (Auto-protection). Dans ce

contexte, nous avons étudié l'impact de l'offre de sécurité sur la garantie des paramètres de QoS.

Enfin, nous avons présenté l'environnement de simulation pour la validation de notre dernière contribution la garantie d'un niveau de service intégrant non seulement la QoS mais aussi la sécurité dans le cas d'utilisation d'une application de vidéoconférence. Nous avons obtenu de bons résultats pour les scénarios avec gestion autonome de la sécurité qui montrent que l'impact de la sécurité sur la QoS est réduit par rapport à une architecture sans gestion autonome.

8.2/ PERSPECTIVES

En se basant sur les travaux et résultats obtenus dans le cadre de cette thèse, nous envisageons de définir des contraintes plus spécifiques concernant chaque paramètre de sécurité. La spécification de ces contraintes nous permettra par la suite d'adapter notre architecture autonome pour réduire les violations et les pénalités. De plus, nous visons à étendre notre architecture de Cloud Networking afin de prendre en considération en plus des services de type IaaS et NaaS les services de type PaaS et SaaS tout en leur garantissant un niveau de service intégrant une QoS et une sécurité adaptées grâce à de nouveaux types de SLA qui seront gérés d'une façon autonome. Une autre perspective concerne l'étude de la mise à l'échelle de notre architecture et plus particulièrement la scalabilité des algorithmes proposés pour la sélection et la gestion autonome des ressources. L'implémentation de notre architecture dans un Cloud réel nous permettra de tester ces algorithmes pour la sélection et la gestion des ressources avec une garantie d'un niveau de service contenant la QoS et la sécurité. Enfin, une architecture de Cloud Networking hybride qui permet de profiter des concepts de l'architecture de type Broker et l'architecture de type Fédération peut être étudiée afin d'évaluer les éventuelles améliorations en termes de performances.

8.3/ PUBLICATIONS

Dans cette thèse, une partie de notre travail a été publiée dans un journal international (chapitres de 2 à 5) et l'autre partie est en cours de révision ou encore de préparation dans deux journaux internationaux (chapitres 6 et 7). De plus, nos travaux ont été publiés dans plusieurs conférences internationales.

- Journaux internationaux :

1- **Mohamad Hamze**, Nader Mbarek and Olivier Togni : "SLA-based Resource Allocation within Cloud Networking Environment". International Journal of New Computer Architectures and their Applications (IJNCAA), Vol. 5, No. 2, pp. 61–78, August 2015.

2- **Mohamad Hamze**, Nader Mbarek and Olivier Togni : "End-to-End SLA-based Cloud Networking Resource Self-management". (Under review)

3- **Mohamad Hamze**, Nader Mbarek and Olivier Togni : "SLA-based Security Guarantee

within Autonomic Cloud Networking Environment". (In preparation)

- Conférences internationales :

4- **Mohamad Hamze**, Nader Mbarek and Olivier Togni : "Broker and Federation Based Cloud Networking Architecture for IaaS and NaaS QoS Guarantee". 13th Annual IEEE Consumer Communications & Networking Conference, IEEE CCNC 2016. Las Vegas, USA. 9–12 January 2016.

5- **Mohamad Hamze**, Nader Mbarek and Olivier Togni : "Resource Self-management under an SLA within a Cloud Networking Environment". Sixth International Conference on Cloud Computing, GRIDs, and Virtualization, CLOUD COMPUTING 2015. pp. 131–137. Nice, France. 22–27 March 2015.

6- **Mohamad Hamze**, Nader Mbarek and Olivier Togni : "Self-establishing a Service Level Agreement within Autonomic Cloud Networking Environment". IEEE/IFIP Network Operations and Management Symposium, IEEE/IFIP NOMS 2014. Poland. 5–9 May 2014.

7- **Mohamad Hamze**, Nader Mbarek and Olivier Togni : "Autonomic Brokerage Service for an End-to-End Cloud Networking Service Level Agreement". IEEE 3rd Symposium on Network Cloud Computing and Applications, IEEE NCCA 2014. pp. 54–61. Rome, Italy. 5–7 February 2014.

BIBLIOGRAPHIE

[1] Strachey, Christopher, «Time Sharing in Large Fast Computers,» Proceedings of the International Conference on Information processing, UNESCO. Paper B.2.19, pp. 336–341, June 1959.

[2] «Cloud computing - History,» En ligne : https ://en.wikipedia.org/wiki/Cloud_computing [accédé en 10/2015].

[3] Wygwam, «Le Cloud Computing : Réelle révolution ou simple évolution ?,» En ligne : http ://www.wygwam.com/documents/cloud-computing.pdf [accédé en 10/2015].

[4] L. Vaquero, L. Merino, J. Caceres, M. Lind, «A break in the clouds : towards a cloud definition,» ACM SIGCOMM Computer Communication Review, vol. 39 n.1, pp. 50–55, 2009.

[5] «NIST Definition of Cloud Computing v15,» En ligne : http ://www.nist.gov/itl/cloud/upload/cloud-def-v15.pdf [accédé en 10/2015].

[6] R. Buyya, C. Yeo, S. Venugopal, J. Broberg, I. Brandic, «Cloud computing and emerging IT platforms : Vision, hype, and reality for delivering computing as the 5th utility,» Future Generation Computer Systems, vol. 25, n. 16, pp. 599–616, 2009.

[7] C. Gong, J. Liu, Q. Zhang, H. Chen, Z. Gong, «The Characteristics of Cloud Computing» 39th International Conference on Parallel Processing Workshops (ICPPW), pp. 275–279, 2010.

[8] S. Subashini, V. Kavitha, «A survey on security issues in service delivery models of cloud computing,» Journal o Network Comput Appl, pp. 1–11, 2010.

[9] «Heroku,» En ligne : https ://www.heroku.com/ [accédé en 10/2015]

[10] «VMware,» En ligne : http ://www.vmware.com/ [accédé en 10/2015]

[11] P. Costa, «Bridging the Gap Between Applications and Networks in Data Centers,» LADIS Workshop 12 Madeira, Portugal, 2012.

[12] H. Sakai, «Standardization Activities for Cloud Computing,» Global Standardization Activities, NTT Information Sharing Platform Laboratories Musashino-shi, vol. 9 No. 6, June 2011.

[13] «DMTF Standard Cloud,» En ligne : http ://dmtf.org/standards/cloud [accédé en 10/2015].

[14] «IEEE Cloud Computing,» En ligne : http ://cloudcomputing.ieee.org [accédé en 10/2015].

[15] «CPIP IEEE,» En ligne : http ://standards.ieee.org/develop/project/2301.html [accédé en 10/2015].

[16] «SIIF IEEE,» En ligne : http ://standards.ieee.org/develop/project/2302.html [accédé en 10/2015].

[17] «International Telecommunication Union - Telecommunication Standardization Sector (ITU-T),» En ligne : http ://www.itu.int/en/ITU-T/focusgroups/cloud/Pages/default.aspx [accédé en 10/2015].

[18] «Global Inter-Cloud Technology Forum (GICTF),» En ligne : http ://cloud-standards.org [accédé en 10/2015].

[19] «Internet Engineering Task Force (IETF),» En ligne : http ://www.googlesyndicatedsearch.com/u/ietf ?q=cloud&Search=Search&safe=active [accédé en 10/2015].

[20] «European Telecommunications Standards Institute (ETSI),» En ligne : http ://www.etsi.org/index.php/technologies-clusters/technologies/grid-and-cloud-computing [accédé en 10/2015].

[21] «Open Grid Forum (OGF),» En ligne : http ://www.gridforum.org/standards/ [accédé en 10/2015].

[22] «open cloud consortium (OCC),» En ligne : http ://opencloudconsortium.org/working-groups/ [accédé en 10/2015].

[23] «Object Management Group (OMG),» En ligne : http ://www.omg.org/ [accédé en 10/2015].

[24] «Cloud Security Alliance (CSA)» En ligne : https ://cloudsecurityalliance.org/ [accédé en 10/2015].

[25] «Cloud Computing Interoperability Forum (CCIF),» En ligne : http ://www.omg.org/news/meetings/tc/dc/special-events/Cloud_Computing/Cloud_Computing_Interoperability_Forum.pdf [accédé en 10/2015].

[26] «ISO/IEC JTC 1,» En ligne : http ://www.iso.org/iso/jtc1_home.html [accédé en 10/2015].

[27] «Opennebula,» En ligne : http ://opennebula.org/ [accédé en 10/2015].

[28] «Nimbus,» En ligne : http ://www.nimbusproject.org/ [accédé en 10/2015].

[29] «Eucalyptus,» En ligne : http ://www.eucalyptus.com/ [accédé en 10/2015].

[30] «OpenStack,» En ligne : http ://www.openstack.org/ [accédé en 10/2015].

[31] «Cloudstack,» En ligne : http ://cloudstack.apache.org/ [accédé en 10/2015].

[32] «Snooze,» En ligne : http ://snooze.inria.fr/ [accédé en 10/2015].

[33] R. N. Calheiros, R. Ranjan, A. Beloglazov, C. A. F. De Rose, R. Buyya, «CloudSim : a toolkit for modelsing and simulation of cloud computing environments and evaluation of resource provisioning algorithms,» Software : Practice and Experience, Vol.41, No.1, pp.23–50, 2011.

[34] «ReaCloudSim,» En ligne : http ://sourceforge.net/projects/realcloudsim/ [accédé en 10/2015].

[35] J. Jung, H. Kim, «MR-CloudSim : Designing and implementing MapReduce computing model on CloudSim,» ICTC 2012 International Conference on ICT Convergence, pp. 504–509, 2012.

[36] S. Ostermann, K. Plankensteiner, R. Prodan, T. Fahringer, «GroudSim : an event-based simulation framework for computational grids and clouds,» CoreGRID/ERCIM Workshop on Grids and Clouds. Springer Computer Science Editorial, Ischia, pp. 305–313, 2010.

[37] M. Tighe, G. Keller, M. Bauer, H. Lutfiyya, «DCSim : a data centre simulation tool for evaluating dynamic virtualized resource management,» The 6th International DMTF Academic Alliance Workshop on Systems and Virtualization Management : Standard and the Cloud, pp. 385–392, 2012.

[38] D. Kliazovich, P. Bouvry, S. U. Khan, «GreenCloud : a packet-level simulator of energy-aware cloud computing cata centers,» IEEE Global Telecommunications Conference, pp. 1–5, 2010.

[39] «The Network Simulator Ns2,» En ligne : http ://www.isi.edu/nsnam/ns/ [accédé en 10/2015].

[40] A. Nunez, J. L. Vazquez-Poletti, A. C. Caminero, G. G. Castane et al, «iCanCloud : a flexible and scalable cloud infrastructure simulator,» Journal of Grid Computing, Vols. 1, No.1, pp. 185–209, 2012.

[41] «SAP Business ByDesign,» En ligne : www.sap.com/sme/solutions/business management/businessbydesign/index.epx [accédé en 10/2015].

[42] L. Youseff, M. Butrico, D. Da Silva, «Toward a Unified Ontology of Cloud computing,» IEEE Grid Computing Environments Workshop, pp. 1–10, 2008.

[43] «Cloud Hosting, CLoud Computing and Hybrid Infrastructure from GoGrid,» En ligne : http ://www.gogrid.com [accédé en 10/2015].

[44] «FlexiScale Cloud Comp and Hosting,» En ligne : www.flexiscale.com [accédé en 10/2015].

[45] «Amazon Web Service,» En ligne : http ://aws.amazon.com/ [accédé en 10/2015].

[46] «Google Cloud,» En ligne : https ://cloud.google.com/ [accédé en 10/2015].

[47] «Windows Azure,» En ligne : www.microsoft.com/azure accédé en 10/2015].

[48] «Salesforce CRM,» En ligne : http ://www.salesforce.com/platform [accédé en 10/2015].

[49] «Dedicated Server, Managed Hosting, Web Hosting by RackspaceHosting„» En ligne : http ://www.rackspace.com [accédé en 10/2015].

[50] ITU-T, «Focus Group on Cloud Computing, Technical Report,» Parts 1 to 7, 2012.

[51] «OpenStack : NaaS,» En ligne : https ://wiki.openstack.org/wiki/NaaS [accédé en 10/2015].

[52] S. K. Garg, R. Buyya, «NetworkCloudSim : modeling parallel applications in cloud simulations,» 4th IEEE International Conference on Utility and Cloud Computing, pp. 105–113, 2011.

[53] T. Benson, A. Akella, A. Shaikh, S. Sahu, «CloudNaaS : a cloud networking platform for enterprise applications.,» In Proceedings of the 2nd ACM Symposium on Cloud Computing (SOCC'11). ACM, New York, NY, USA, , Article 8 , pp. 1–13, 2011.

[54] «Institut Telecom, pyOCNI,» En ligne : http ://occi-wg.org/2012/02/20/occi-pyocni/ [accédé en 10/2015].

[55] «OpenNaaS,» En ligne : http ://www.opennaas.org/overview/ [accédé en 10/2015].

[56] R. Buyya, R. Ranjan, R. Calheiros, «Intercloud : Utility-oriented federation of cloud computing environments for scaling of application services,» Algorithms and Architectures for Parallel Processing, Springer Berlin / Heidelberg, pp. 13–31, 2010.

[57] «Cloud federation,» En ligne : http ://searchcloudprovider.techtarget.com/definition/What-is-cloud-federation [accédé en 10/2015].

[58] C.J. Wu, J.M. Ho,M.S. Chen, «Time-Critical Event Dissemination in Geographically Distributed Clouds,» IEEE INFOCOM Workshop on Cloud Computing, pp. 654–659, 2011.

[59] F. Faniyi, R. Bahsoon, G. Theodoropoulos, «A Dynamic Data-Driven Simulation Approach for Preventing Service Level Agreement Violations in Cloud Federation,» International Conference on Computational Science, ICCS 2012, Procedia Computer Science 9 , pp. 1167–1176, 2012.

[60] «IEEE InterCloud Testbed,» En ligne : http ://www.intercloudtestbed.org/ [accédé en 10/2015].

[61] «Mosaic,» En ligne : http ://www.mosaic-cloud.eu [accédé en 10/2015].

[62] «CompatibleOne,» En ligne : http ://www.compatibleone.org/bin/view/Main/ [accédé en 10/2015].

[63] «Spotcloud,» En ligne : http ://spotcloud.com/ [accédé en 10/2015].

[64] «Appirio,» En ligne : http ://www.appirio.com/ [accédé en 10/2015].

[65] «Cisco intercloud fabric,» En ligne : http ://www.cisco.com/c/en/us/products/cloud-systems-management/intercloud-fabric/index.html [accédé en 10/2015].

[66] «Intercloudsys,» En ligne : http ://www.intercloudsys.com/ [accédé en 10/2015].

[67] H. S. Gunawi, T. Do, J. M. Hellerstein, I. Stoica, D. Borthakur, J. Robbins, “Failure as a service (faas) : A cloud service for large-scale, online failure drills,” Tech. Rep. UCB/EECS-2011-87, Electrical Engineering and Computer Sciences, University of California, Berkeley, July 2011.

[68] S. Paquette, P. T. Jaeger, S. C. Wilson, “Identifying the security risks associated with governmental use of cloud computing,” Government Information Quaterly 27 (3), pp. 245–253, 2010.

[69] “Cloud Computing Takes Off”, Morgan Stanley, May 23, 2011

[70] Y. Kouki, T. Ledoux, D. Serrano, S. Bouchenak, J. Lejeune, J. Sopena, L. Arantes et P. Sens, «SLA et qualité de service pour le Cloud Computing,» Conférence d’informatique en Parallélisme, Architecture et Système, ComPAS 2013, 2013.

[71] M. Kandukuri Balachandra Reddy, R. Paturi V et A. Rakshit, «Cloud Security Issues,» IEEE International Conference on Services Computing, IEEE, pp. 517–520, 2009.

[72] I. Brandic et E. Feller, «Energy and QoS-aware workload management in clouds,» The final proposal of the focus group, 2011.

[73] «Amazon Elastic Computing Cloud,» En ligne : aws.amazon.com/ec2 [accédé en 10/2015]

[74] I. Goiri, F. Julia, J. O. Fito, M. Macıas et J. Guitart, «Resource-Level QoS Metric for CPU-Based Guarantees in Cloud Providers,» GECON 2010, LNCS 6296, pp. 34–47, 2010.

[75] Yanzhi, W., Shuang, C., Pedram, M. : Service Level Agreement-Based Joint Application Environment Assignment and Resource Allocation in Cloud Computing Systems. In : Proc. IEEE Green Technologies Conference, pp. 167–174, 2013.

[76] «Practical Guide to Cloud Service Level Agreements Version 1.0,» Cloud Standards Customer Council Steering Committee , 10 April 2012. En ligne : http ://www.cloud-council.org/PGCloudSLA040512MGreer.pdf. [accédé en 10/2015]

[77] «Oasis,» En ligne : https ://www.oasis-open.org/committees/ [accédé en 08/2015]

[78] «SNIA Cloud Data Management Interface (CDMI),» http ://www.snia.org/cdmi [accédé en 10/2015]

[79] «ONF,» https ://www.opennetworking.org/ [accédé en 10/2015]

[80] «mycloud,» En ligne : http ://mycloud.inrialpes.fr/ [accédé en 10/2015]

[81] «reservoir,» En ligne : http ://www.reservoir-fp7.eu/ [accédé en 10/2015]

[82] «contrail,» En ligne : http ://contrail-project.eu/ [accédé en 10/2015]

[83] «quads,» En ligne : http ://www.irisa.fr/myriads/software/quads/ [accédé en 10/2015]

[84] «Isoni,» En ligne : http ://www.irmosproject.eu/Isoni.aspx. [accédé en 10/2015]

[85] «EASI-CLOUDS,» En ligne : http ://easi-clouds.eu/2012/02/03/project-description/ [accédé en 10/2015]

[86] «broker@cloud,» En ligne : http ://www.broker-cloud.eu/ [accédé en 10/2015]

[87] «ETICS,» En ligne : http ://ec.europa.eu/information_society/apps/projects/logos/7/248567/080/deliverables/001_ETICSD43v10.pdf. [accédé en 10/2015]

[88] «modaclouds,» En ligne : http ://www.modaclouds.eu. [accédé en 10/2015]

[89] Y. Kouki et T. Ledoux, «CSLA : a Language for improving Cloud SLA Management,» In International Conference on Cloud Computing and Services Science, CLOSER 2012, Porto, Portugal, April 2012.

[90] V. C. Emeakaroha, «Managing Cloud Service Provisioning and SLA Enforcement via Holistic Monitoring Techniques,» these, 2012.

[91] Z. Ye, X. Zhou et A. Bouguettaya, «Genetic Algorithm Based QoS-Aware Service Compositions in Cloud Computing,» DASFAA 2011, Part II, LNCS 6588, pp. 321–334, 2011.

[92] L. Wu, S. Garg et R. Buyya, «SLA-based resource allocation for a software as a service provider in cloud computing environments.,» In : Proceedings of the 11th IEEE/ACM international symposium on cluster computing and the grid (CCGrid 2011), Los Angeles, USA, May, pp. 23–26, 2011.

[93] G. Cicotti, L. Coppolino, R. Cristaldi, S. D'Antonio et L. Romano, «QoS Monitoring in a Cloud Services Environment : the SRT-15 Approach,» Epsilon srl, Naples - Italy, 2011.

[94] H. Park et H.-Y. Jeong, «The QoS-based MCDM system for SaaS ERP applications with Social Network,» Springer Science+Business Media New York, 2012.

[95] I. Trajkovska, J. Salvachúa et A. M. Velasco, «A Novel P2P and Cloud Computing Hybrid Architecture for Multimedia Streaming with QoS Cost Functions,» MM'10, Firenze, Italy, 2010.

[96] R. Nathuji, A. Kansal et A. Ghaffarkhah, «Q-Clouds : Managing Performance Interference Effects for QoS-Aware Clouds,» EuroSys'10, Paris, France, 2010.

[97] S. Ferretti, V. Ghini, F. Panzieri, M. Pellegrini et E. Turrini, «QoS-aware clouds.,» In : The 3rd IEEE international conference on cloud computing, pp. 321–328, 2010.

[98] J. Edmondson, D. Schmidt et A. Gokhale, «QoS-Enabled Distributed Mutual Exclusion in Public Clouds,» OTM'11 Proceedings of the 2011th Confederated international conference on On the move to meaningful internet systems, vol. 2, pp. 542–559, 2011.

[99] V. Stantchev et C. Schröpfer, «Negotiating and Enforcing QoS and SLA in Grid and Cloud Computing,» in Proc. of the 4th International Conference on Advances in Grid and Pervasive Computing (GPC), Geneva, Switzerland, pp. 25–35, 2009.

[100] P. Ganghishetti, R. Wankar, R. M. Almuttairi et C. R. Rao, «Rough Set Based Quality of Service Design for Service Provisioning in Clouds,» RSKT 2011, pp. 268–273, 2011.

[101] S. Garg, C. Yeo, A. Anandasivam et R. Buyya, «Environment-conscious scheduling of HPC applications on distributed cloud-oriented data centers,» Journal of Parallel and Distributed Computing, 2011.

[102] R. Karim, D. Chen, A. Miri "An End-to-End QoS Mapping Approach for Cloud Service Selection," IEEE Ninth World Congress on Services, pp. 341–348, 2013.

[103] W. Yanzhi, L. Xue, M. Pedram "A Game Theoretic Framework of SLA-based Resource Allocation for Competitive Cloud Service Providers," Sixth Annual IEEE Green Technologies Conference, pp. 37–43, 2014.

[104] H. Feng and W. Wu, "Framework and User Migration Strategy of Cloud-Based Video Conference Multi-Gateway System," 19th International Conference on High Performance Computing (HiPC), pp. 1–8, Dec 2012.

[105] Y. Wu, C. Wu, B. Li and F. Lau, "vSkyConf : Cloud-assisted Multi-party Mobile Video Conferencing," Second ACM SIGCOMM workshop on Mobile cloud computing, 2013

[106] Y. Feng, Ba. Li and Bo. Li, "Airlift : Video Conferencing as a Cloud Service using Inter-Datacenter Networks," 20th IEEE International Conference on Network Protocols, 2012.

[107] J. Cerviño, P. Rodríguez, I. Trajkovska, F. Escribano, J. Salvachúa, "A Cost-Effective Methodology Applied to Videoconference Services over Hybrid Clouds," Mobile Netw Appl, Volume 18, pp. 103–109, 2013.

[108] J. Cervino, F. Escribano, P. Rodríguez, I. Trajkovska, J. Salvachúa, "Videoconference Capacity Leasing on Hybrid Clouds," IEEE International Conference on Cloud Computing (CLOUD), pp. 340–347, July 2011.

[109] «Cloud,» En ligne : https ://fr.wikipedia.org/wiki/S%C3%A9curit%C3%A9_du_-cloud#whatnewaboutcloudcomputing [accédé en 10/2015]

[110] Y. Demchenko et al., Security Infrastructure for On-demand Provisioned Cloud Infrastructure Services. Third IEEE International Conference on Coud Computing Technology and Science. pp. 255–263, 2011.

[111] Dimitrios Zissis, Dimitrios Lekkas. Addressing cloud computing security issues. Future Generation Computer Systems 28, 583–592, 2012.

[112] ITU-T, "Focus Group on Cloud Computing, TR part 5," 2012.

[113] W. Stallings, "Network security essentials : applications and standards," fourth edition, 2005.

[114] Kaspersky, Global Corporate IT Security Risks : 2013. Technical Report, May 2013.

[115] «ENISA,» En ligne : https ://www.enisa.europa.eu/ [accédé en 10/2015]

[116] L. Karadsheh, "Applying security policies and service level agreement to IaaS service model to enhance security and transition". computers & security 31, 315–326, 2012.

[117] Studnia, I. et al. : Survey of security problems in cloud computing virtual machines. Computer and Electronics Security Applications. 2012.

[118] Khalil, I.M. ; Khreishah, A. ; Azeem, M. : Cloud Computing Security : A Survey. Computers, vol. 3, pp. 11-35, 2014.

[119] W. Yanzhi, C. Shuang, M. Pedram “Service Level Agreement-Based Joint Application Environment Assignment and Resource Allocation in Cloud Computing Systems,” IEEE Green Technologies Conference, pp. 167–174, 2013.

[120] «Cloud Standards Customer Council CSCC,» En ligne : http ://www.cloud-council.org/ [accédé en 10/2015]

[121] «supercloud,» En ligne : http ://www.supercloud-project.eu/ [accédé en 10/2015]

[122] «SAIL Project,» En ligne : http ://www.sail-project.eu/ [accédé en 10/2015]

[123] «Seed4c,» En ligne : http ://www.celticplus-seed4c.org/index.php[accédé en 10/2015]

[124] «GEANT Project,» En ligne : http ://www.geant.net/pages/home.aspx [accédé en 10/2015]

[125] «Generalised Architecture for Dynamic Infrastructure Services (GEYSERS),» En ligne : http ://www.geysers.eu/ [accédé en 10/2015]

[126] «OWASP,» En ligne : https ://www.owasp.org/index.php/Main_Page [accédé en 10/2015]

[127] «CloudSIRT,» En ligne : https ://cloudsecurityalliance.org/pr20110215.html [accédé en 10/2015]

[128] «Sucre Project,» En ligne : http ://www.sucreproject.eu/ [accédé en 10/2015]

[129] «Cloudspaces,» En ligne : http ://Cloudspaces.eu/ [accédé en 10/2015]

[130] C. Christmann, J. Falkner, A. Horch, H. Kett. Identification of IT Security and legal requirements regarding Cloud services. Sixth International Conference on Cloud Computing, GRIDs, and Virtualization, CLOUD COMPUTING 2015. pp. 1–7, 2015.

[131] M. Ali, R. Dhamotharan, E. Khan, S.U. Khan, A.V. Vasilakos, K. Li, A.Y. Zomaya, SeDaSC : secure data sharing in clouds, IEEE Systems Journal, pp. 1–10, 2015.

[132] Wenjuan Fan, Harry Perros. A novel trust management framework for multi-cloud environments based on trust service providers. Knowledge-Based Systems 70, 392–406, 2014.

[133] K. Sankar, S. Kannan, P. Jennifer, On-demand security architecture for cloud computing, Middle-East J. Sci. Res. 20 (2), 241–246, 2014.

[134] Q. Liu, G. Wang, J. Wu, Time-based proxy re-encryption scheme for secure data sharing in a cloud environment, Inform. Sci. 258, 355–370, 2014.

[135] M. Ficco, M. Rak, Stealthy denial of service strategy in cloud computing, IEEE Trans. Cloud Comput, 2014.

[136] V. Varadharajan, U. Tupakula, Counteracting security attacks in virtual machines in the cloud using property based attestation, J. Network Comput. Appl. 40, 31–45, 2014.

[137] M. Kazim, R. Masood, M.A. Shibli, Securing the virtual machine images in cloud computing, in : Proceedings of the ACM 6th International Conference on Security of Info and Networks, pp. 425–428, 2013.

[138] C. Li, A. Raghunathan, N.K. Jha, A trusted virtual machine in an untrusted management environment, IEEE Trans. Serv. Comput. 5 (4), 472–483, 2012.

[139] M. Aslam, C. Gehrmann, M. Bjorkman, Security and trust preserving VM migrations in public clouds, in : IEEE 11th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom), pp. 869–876, 2012.

[140] L. Wei, H. Zhu, Z. Cao, X. Dong, W. Jia, Y. Chen, A.V. Vasilakos, Security and privacy for storage and computation in cloud computing, Inform. Sci. 258, 371–386, 2014.

[141] P. Chavan et al. IaaS Cloud Security. International Conference on Machine Intelligence Research and Advancement, 2013.

[142] H. Moraes, R. Nunes, D. Guedes, DCPortalsNg : efficient isolation of tenant networks in virtualized datacenters, in : Thirteenth International Conference on Networks, pp. 230–235, 2014.

[143] X. He, T. Chomsiri, P. Nanda, Z. Tan, Improving cloud network security using the tree-rule firewall, Future Gener. Comput. Syst. 30, 116–126, 2014.

[144] S. Ruj, M. Stojmenovic, A. Nayak, Decentralized access control with anonymous authentication of data stored in clouds, IEEE Trans. Parallel Distrib. Syst. 25 (2), 384–394, 2014.

[145] S.K. Sah, S. Shakya, H. Dhungana, A security management for cloud based applications and services with diameter-AAA, in : IEEE International Conference on Issues and Challenges in Intelligent Computing Techniques (ICICT), pp. 6–11, 2014.

[146] M.L. Hale, R. Gamble, Building a compliance vocabulary to embed security controls in cloud SLA, in : IEEE Ninth World Congress on Services (SERVICES), pp. 118–125, 2013.

[147] D. Serrano et al. SLA guarantees for cloud services. Future Generation Computer Systems, 2015.

[148] M.L. Hale, R. Gamble, Risk propagation of security SLA in the cloud, in : IEEE Globecom Workshops (GC Wkshps), pp. 730–735, 2012.

[149] M. Rak, L. Liccardo, R. Aversa. A SLA-based interface for security management in Cloud and GRID Integrations. 7th International Conference on Information Assurance and Security (IAS), pp. 378–383, 2011.

[150] S.H. Na, E.N. Huh, A broker-based cooperative security-SLA evaluation methodology for personal cloud computing, Sec. Commun. Netw, 2014.

[151] Shirlei Aparecida de Chaves, Carlos Becker Westphall and Flavio Rodrigo Lamin. SLA Perspective in Security Management for Cloud Computing. Sixth International Conference on Networking and Services, 2010.

[152] Y. Neng-Hai, Z. Hao, J. Xu, W. Zhang, C. Zhang, Review of cloud computing security, Acta Electron. Sinica 41 (2), 371–381, 2013.

[153] C. Rong, S.T. Nguyen, M.G. Jaatun, Beyond lightning : a survey on security challenges in cloud computing, Comput. Electr. Eng. 39 (1), 47–54, 2013.

[154] M.D. Ryan, Cloud computing security : the scientific challenge, and a survey of solutions, J. Syst. Softw. 86 (09), 2263–2268, 2013.

[155] Z. Xiao, Y. Xiao, Security and privacy in cloud computing, IEEE Commun. Surveys Tutorials 15 (2), 843–859, 2013.

[156] Z. Tari, Security and privacy in cloud computing, IEEE Cloud Comput. 1 (1), 54–57, 2014.

[157] L. FB Soares, D. AB Fernandes, J.V. Gomes, M.M. Freire, P. RM Inácio, Cloud security : state of the art, in : Security, Privacy and Trust in Cloud Systems, Springer, Berlin, Heidelberg, pp. 3–44, 2014.

[158] C. Modi, D. Patel, B. Borisaniya, A. Patel, M. Rajarajan, A survey on security issues and solutions at different layers of Cloud computing, J. Supercomput. 63 (2), 561–592, 2013.

[159] R. Latif, H. Abbas, S. Assar, Q. Ali, Cloud computing risk assessment : a systematic literature review, in : Future Information Technology, Springer, Berlin, Heidelberg, pp. 285–295, 2014.

[160] D. AB. Fernandes, L. FB. Soares, J.V. Gomes, M.M. Freire, P. RM Inácio, Security issues in cloud environments : a survey, Int. J. Inform. Sec. 13 (2), 113–170, 2014.

[161] R. Chandramouli, M. Iorga, S. Chokhani, Cryptographic key management issues and challenges in cloud services, in : Secure Cloud Computing, Springer, New York, pp. 1–30, 2014.

[162] M. Ali, S. Khan, A. Vasilakos. Security in cloud computing : Opportunities and challenges, Information Sciences 305, 357–383, 2015.

[163] P. Horn, «Autonomic computing : Ibms perspective on the state of information technology,» En ligne : http ://www.research.ibm.com/autonomic/manifesto/autonomic computing.pdf. [accédé en 10/2015]

[164] IBM, «Autonomic computing : The solution,» En ligne : http ://www.research.ibm.com/autonomic/overview/solution.html. [accédé en 10/2015]

[165] IBM, «An architectural blueprint for autonomic computing,» Tech. rep., IBM, 2006.

[166] R. Jia, «Autonomic management of virtualized resources in cloud computing,» Wayne State University Dissertations. Paper 358., 2011.

[167] «Dynamic Cloud Network Service,» En ligne : https ://wiki.openstack.org/wiki/DynamicCloudNetworkService. [accédé en 10/2015]

[168] R. Pottier et J.-M. Jean-Marc Menaud, «BtrScript : A Safe Management System for Virtualized Data Center,» ICAS 2012, The Eighth International Conference on Autonomic and Autonomous Systems, pp. 49–56, 2012.

[169] F. Hermenier, A. Lèbre et J.-M. Menaud, «Cluster-wide context switch of virtualized jobs,» in Proceedings of the 19th ACM International Symposium on High Performance Distributed Computing, ser. HPDC '10. New York, NY, USA : ACM, pp. 658–666, 2010.

[170] E. Daubert, F. André et O. Barais, «Adaptation multi-niveaux : l'infrastructure au service des applications,» RenPar'20 / SympA'14 / CFSE 8, Saint-Malo, France, 2011.

[171] «sla-at-soi,» En ligne : http ://sla-at-soi.eu/ [accédé en 10/2015]

[172] Project, «FoSII (Foundations of Self-governing Infrastructures),» En ligne : http ://www.infosys.tuwien.ac.at/linksites/FOSII/index.html [accédé en 10/2015]

[173] I. Hoffert, D. Schmidt et A. Gokhale, «Adapting distributed realtime and embedded publish/subscribe middleware for cloud-computing environments,» Proc. of ACMIIFIPIUSENIX Middleware, 2010.

[174] «celarCloud,» En ligne : http ://www.celarCloud.eu/ [accédé en 10/2015]

[175] «CloudTM,» En ligne : http ://www.cloudtm.eu/ [accédé en 10/2015]

[176] «stratuslab,» En ligne En ligne : http ://stratuslab.eu/index.html. [accédé en 10/2015]

[177] Y. Kouki et T. Ledoux, «SLA-driven Capacity Planning for Cloud applications,» In Proc of the IEEE International Conference on Cloud Computing Technology and Science (CloudCom), Dec 2012.

[178] P. Patel, A. Ranabahu et A. Sheth, «Service Level Agreement in Cloud Computing,» Kno.e.sis Center, Wright State University, DA-IICT, OOPSLA 2009 - Workshop, 2009.

[179] F. Faniyi, R. Bahsoon et G. Theodoropoulos, «A Dynamic Data-Driven Simulation Approach for Preventing Service Level Agreement Violations in Cloud Federation,» International Conference on Computational Science, ICCS 2012, Procedia Computer Science 9 , pp. 1167–1176, 2012.

[180] D. Ardagna, M. Trubian et L. Zhang, «Sla based resource allocation policies in autonomic environments,» J. Parallel Distrib. Comput. 67, pp. 259–270, 2007.

[181] H. Van, F. Tran et J.-M. Menaud, «SLA-aware virtual resource management for cloud infrastructures,» The 9th IEEE international conference on computer and information technology, vol. 1, pp. 357–362, 2009.

[182] H. Van, F. Tran et J.-M. Menaud, «Performance and power management for cloud infrastructures,» The 3rd IEEE international conference on cloud computing, pp. 329–336, 2010.

[183] W. Itani, C. Ghali, R. Bassil, A. Kayssi et A. Chehab, «ServBGP : BGP-Inspired Autonomic Service Routing for Multi-Provider Collaborative Architectures in the Cloud,» Elsevier Future Generation Computer Systems (Impact Factor : 2.365), in press, May 2012.

[184] Alwesabi, A., Okba, K. : Security Method : Cloud Computing Approach Based on Mobile Agents. International Journal of New Computer Architectures and their Applications (IJNCAA), Vol. 4, No. 1, pp. 17–29, 2014.

[185] M. Huebscher, J. Mccann. A survey of Autonomic Computing—degrees, models and applications. ACM Computing Surveys, vol 40, Issue 3, pp. 1–28, 2008.

[186] Kresimir Popovic et Zeljko Hocenski, «Cloud computing security issues and challenges,» Conference Publications, pp. 1–8, 29 juillet 2010.

[187] Yanpei Chen, Vern Paxson et Randy H. Katz,«What's New About Cloud Computing Security ?,» EECS Department, University of California, Berkeley, pp. 1–8, 20 janvier 2010.

[188] Damien Riquet, Gilles Grimaud et Michaël Hauspie, «Étude de l'impact des attaques distribuées et multi-chemins sur les solutions de sécurité réseaux ,» 9ème Conférence Internationale Jeunes Chercheurs, Lille, France, pp. 1–8 Octobre 2012.

[189] «NIST SP 800-63,» En ligne : http ://csrc.nist.gov/publications/nistpubs/800-63/SP800-63V1_0_2.pdf [accédé en 10/2015]

[190] «ISO/IEC 29115,» En ligne : https ://www.oasis-open.org/committees/download.php/44751/285-17Attach1.pdf [accédé en 10/2015]

[191] International Telecommunication Union, X-509 | ISO/IEC 9594-8, The directory : Public-key and attribute certificate frameworks, ITU, X-Series, 2001.

[192] M. Baugher, D. McGrew, M. Naslund, E. Carrara, K. Norrman, “RFC 3711 : The Secure Real-time Transport Protocol (SRTP)”, Request for Comments, IETF, March 2004.

[193] E. Rescola and N. Modadugu, “RFC 4347 : Datagram Transport Layer Security (DTLS)”, Request for Comments, IETF, April 2006.

[194] S. Kent and K. Seo, “RFC4301 : Security Architecture for Internet Protocol (IPSec),” Request for Comments, IETF, December 2005.

[195] J. Schiller “Cryptographic Algorithm Implementation Requirements for Encapsulating Security Payload and Authentication Header,” RFC 4305, Dec. 2005.

ANNEXE A
UNITÉS UTILISÉES DANS LES REPRÉSENTATIONS DES SCHÉMAS XML

Dans cette annexe, nous présentons les unités utilisées dans les représentations des schémas XML.

```
<xs :simpleType name="App_Type">
        <xs :restriction base="xs :string">
                <xs :enumeration value="Computing Aplication" />
                <xs :enumeration value="Real Time Aplication" />
                <xs :enumeration value="Streaming Application" />
                <xs :enumeration value="Other" />
        </xs :restriction>
</xs :simpleType>
<xs :simpleType name="Availability_level_value">
        <xs :restriction base="xs :string">
                <xs :enumeration value="Platinum" />
                <xs :enumeration value="Gold" />
                <xs :enumeration value="Silver" />
                <xs :enumeration value="Bronze" />
        </xs :restriction>
</xs :simpleType>
<xs :simpleType name="Avail_Type">
        <xs :restriction base="xs :string">
                <xs :pattern value="[0-100] %"/>
        </xs :restriction>
</xs :simpleType>
<xs :simpleType name="Avail_VM_Num_Type">
        <xs :restriction base="xs :integer">
                <xs :minInclusive value="1" />
        </xs :restriction>
</xs :simpleType>
<xs :simpleType name="BW_Type">
        <xs :restriction base="xs :string">
                <xs :pattern value="([0-9])+ Mb/s"/>
        </xs :restriction>
</xs :simpleType>
<xs :simpleType name="Cost_BW_Type">
```

```
        <xs :restriction base="xs :string">
                <xs :pattern value="([0-9])+ $/(GB*h)"/>
        </xs :restriction>
</xs :simpleType>
<xs :simpleType name="Cost_st_Type">
        <xs :restriction base="xs :string">
                <xs :pattern value="([0-9])+ $/(GB*h)"/>
        </xs :restriction>
</xs :simpleType>
<xs :simpleType name="Cost_Vm_Type">
        <xs :restriction base="xs :string">
                <xs :pattern value="([0-9])+ $/h"/>
        </xs :restriction>
</xs :simpleType>
<xs :simpleType name="D-W">
        <xs :restriction base="xs :integer">
                <xs :minInclusive value="0" />
                <xs :maxInclusive value="7" />
        </xs :restriction>
</xs :simpleType>
<xs :simpleType name="H-D">
        <xs :restriction base="xs :integer">
                <xs :minInclusive value="0" />
                <xs :maxInclusive value="24" />
        </xs :restriction>
</xs :simpleType>
<xs :simpleType name="iP-address_Type">
        <xs :restriction base="xs :string">
                <xs :pattern value="[0-255]̇[0-255]̇[0-255]̇[0-255]" />
        </xs :restriction>
</xs :simpleType>
<xs :simpleType name="Jit_Type">
        <xs :restriction base="xs :string">
                <xs :pattern value="([0-9])+ ms"/>
        </xs :restriction>
</xs :simpleType>
<xs :simpleType name="Lat_Type">
        <xs :restriction base="xs :string">
                <xs :pattern value="([0-9])+ ms"/>
        </xs :restriction>
</xs :simpleType>
<xs :simpleType name="mail_Type">
        <xs :restriction base="xs :string">
                <xs :pattern value=".+@.+" />
        </xs :restriction>
</xs :simpleType>
<xs :simpleType name="Mask_Type">
        <xs :restriction base="xs :integer">
                <xs :minInclusive value="0" />
```

```
                <xs :maxInclusive value="32" />
        </xs :restriction>
</xs :simpleType>
<xs :simpleType name="Mont_Int_Time_Type">
        <xs :restriction base="xs :string">
                <xs :pattern value="([0-9])+ s"/>
        </xs :restriction>
</xs :simpleType>
<xs :simpleType name="M-Y">
        <xs :restriction base="xs :integer">
                <xs :minInclusive value="0" />
                <xs :maxInclusive value="12" />
        </xs :restriction>
</xs :simpleType>
<xs :simpleType name="PLR_Type">
        <xs :restriction base="xs :string">
                <xs :pattern value="[0-100] %"/>
        </xs :restriction>
</xs :simpleType>
<xs :simpleType name="Port_Type">
        <xs :restriction base="xs :integer">
                <xs :minInclusive value="0" />
                <xs :maxInclusive value="65535" />
        </xs :restriction>
</xs :simpleType>
<xs :simpleType name="QoS_Qualitative_Type">
        <xs :restriction base="xs :string">
                <xs :enumeration value="Platinum" />
                <xs :enumeration value="Gold" />
                <xs :enumeration value="Silver" />
                <xs :enumeration value="Bronze" />
        </xs :restriction>
</xs :simpleType>
<xs :simpleType name="Req_VM_Num_Type">
        <xs :restriction base="xs :integer">
                <xs :minInclusive value="1" />
        </xs :restriction>
</xs :simpleType>
<xs :simpleType name="RT_Type">
        <xs :restriction base="xs :string">
                <xs :pattern value="([0-9])+ ms"/>
        </xs :restriction>
</xs :simpleType>
<xs :simpleType name="Security_Qualitative_Type">
        <xs :restriction base="xs :string">
                <xs :enumeration value="Platinum" />
                <xs :enumeration value="Gold" />
                <xs :enumeration value="Silver" />
                <xs :enumeration value="Bronze" />
```

```
        </xs :restriction>
</xs :simpleType>
<xs :simpleType name="SLA_ID_Type">
        <xs :restriction base="xs :ID"/>
</xs :simpleType>
<xs :simpleType name="St_seq_Type">
        <xs :restriction base="xs :string">
                <xs :pattern value="([0-9])+ MB/s"/>
        </xs :restriction>
</xs :simpleType>
<xs :simpleType name="St_Type">
        <xs :restriction base="xs :string">
                <xs :pattern value="([0-9])+ GB"/>
        </xs :restriction>
</xs :simpleType>
<xs :simpleType name="VM_cap_Type">
        <xs :restriction base="xs :string">
                <xs :pattern value="([0-9])+ GHz"/>
        </xs :restriction>
</xs :simpleType>
<xs :simpleType name="VM_CPU_num_Type">
        <xs :restriction base="xs :integer">
                <xs :minInclusive value="1" />
        </xs :restriction>
</xs :simpleType>
<xs :simpleType name="VM-hyper_Type">
        <xs :restriction base="xs :string">
                <xs :enumeration value="Xen" />
                <xs :enumeration value="KVM" />
                <xs :enumeration value="Other" />
        </xs :restriction>
</xs :simpleType>
<xs :simpleType name="VM_Mem_cap_Type">
        <xs :restriction base="xs :string">
                <xs :pattern value="([0-9])+ GB"/>
        </xs :restriction>
</xs :simpleType>
<xs :simpleType name="VM-proc_Type">
        <xs :restriction base="xs :string">
                <xs :enumeration value="32-bit" />
                <xs :enumeration value="64-bit" />
        </xs :restriction>
</xs :simpleType>
<xs :simpleType name="VM_RAM_cap_Type">
        <xs :restriction base="xs :string">
                <xs :pattern value="([0-9])+ GB"/>
        </xs :restriction>
</xs :simpleType>
<xs :simpleType name="Vm_Type">
```

```
        <xs :restriction base="xs :string">
                <xs :enumeration value="Platinum" />
                <xs :enumeration value="Gold" />
                <xs :enumeration value="Silver" />
                <xs :enumeration value="Bronze" />
        </xs :restriction>
</xs :simpleType>
<xs :simpleType name="W-M">
        <xs :restriction base="xs :integer">
                <xs :minInclusive value="0" />
                <xs :maxInclusive value="4" />
        </xs :restriction>
</xs :simpleType>
```

Document réalisé avec LaTeX et :
le style LaTeX pour Thèse de Doctorat créé par S. Galland — http://www.multiagent.fr/ThesisStyle
la collection de paquets `tex-upmethodology` — http://www.arakhne.org/tex-upmethodology/

Résumé :

De nos jours, le Cloud Networking est considéré comme étant l'un des domaines de recherche innovants au sein de la communauté de recherche du Cloud Computing. Les principaux défis dans un environnement de Cloud Networking concernent non seulement la garantie de qualité de service (QoS) et de sécurité mais aussi sa gestion en conformité avec un accord de niveau de service (SLA) correspondant. Dans cette thèse, nous proposons un Framework pour l'allocation des ressources conformément à un SLA établi de bout en bout entre un utilisateur de services Cloud (CSU) et plusieurs fournisseurs de services Cloud (CSP) dans un environnement de Cloud Networking (architectures d'inter-Cloud Broker et Fédération). Nos travaux se concentrent sur les services Cloud de types NaaS et IaaS. Ainsi, nous proposons l'auto-établissement de plusieurs types de SLA ainsi que la gestion autonome des ressources de Cloud correspondantes en conformité avec ces SLA en utilisant des gestionnaires autonomes spécifiques de Cloud. De plus, nous étendons les architectures et les SLA proposés pour offrir un niveau de service intégrant une garantie de sécurité. Ainsi, nous permettons aux gestionnaires autonomes de Cloud d'élargir leurs objectifs de gestion autonome aux fonctions de sécurité (auto-protection) tout en étudiant l'impact de la sécurité proposée sur la garantie de QoS. Enfin, nous validons notre architecture avec différents scénarios de simulation. Nous considérons dans le cadre de ces simulations des applications de vidéoconférence et de calcul intensif afin de leur fournir une garantie de QoS et de sécurité dans un environnement de gestion autonome des ressources du Cloud. Les résultats obtenus montrent que nos contributions permettent de bonnes performances pour ce type d'applications. En particulier, nous observons que l'architecture de type Broker est la plus économique, tout en assurant les exigences de QoS et de sécurité. De plus, nous observons que la gestion autonome des ressources du Cloud permet la réduction des violations, des pénalités et limite l'impact de la sécurité sur la garantie de la QoS.

Mots-clés : Cloud Computing, Cloud Networking, Inter-Cloud, Service Level Agreement, Qualité de Service, Sécurité, Gestion Autonome, Videoconférence, Calcul Intensif.

Abstract:

Today, Cloud Networking is one of the recent research areas within the Cloud Computing research communities. The main challenges of Cloud Networking concern Quality of Service (QoS) and security guarantee as well as its management in conformance with a corresponding Service Level Agreement (SLA). In this thesis, we propose a framework for resource allocation according to an end-to-end SLA established between a Cloud Service User (CSU) and several Cloud Service Providers (CSPs) within a Cloud Networking environment (Inter-Cloud Broker and Federation architectures). We focus on NaaS and IaaS Cloud services. Then, we propose the self-establishing of several kinds of SLAs and the self-management of the corresponding Cloud resources in conformance with these SLAs using specific autonomic cloud managers. In addition, we extend the proposed architectures and the corresponding SLAs in order to deliver a service level taking into account security guarantee. Moreover, we allow autonomic cloud managers to expand the self-management objectives to security functions (self-protection) while studying the impact of the proposed security on QoS guarantee. Finally, our proposed architecture is validated by different simulation scenarios. We consider, within these simulations, videoconferencing and intensive computing applications in order to provide them with QoS and security guarantee in a Cloud self-management environment. The obtained results show that our contributions enable good performances for these applications. In particular, we observe that the Broker architecture is the most economical while ensuring QoS and security requirements. In addition, we observe that Cloud self-management enables violations and penalties' reduction as well as limiting security impact on QoS guarantee.

Keywords: Cloud Computing, Cloud Networking, Inter-Cloud, Service Level Agreement, Quality of Service, Security, Self-management, Videoconferencing, Intensive Computing.

■ École doctorale SPIM - Université de Bourgogne/UFR ST BP 47870 F - 21078 Dijon cedex
■ tél. +33 (0)3 80 39 59 10 ■ ed-spim@univ-fcomte.fr ■ www.ed-spim.univ-fcomte.fr