

114-004 442-1  
République Algérienne Démocratique et Populaire  
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

**UNIVERSITE BLIDA 1-SAAD DAHLAB**



**Faculté des Sciences**

**Département Informatique**

**MEMOIRE**

Présenté par :

**BOUKRIF Feirouz & CHOUIT Asma**

**En vue d'obtenir le diplôme de master 2**

Spécialité : Ingénierie de logiciel

**Thème :**

**MODELE A BASE DE BROKER POUR  
L'INTEROPERABILITE DANS UN CLOUD HYBRIDE**

**Soutenu le : 19/09/2017.**

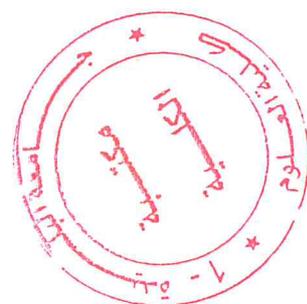
**Devant le Jury :**

ne BOUMAHDI Fatima Présidente  
ne ELGHERS Sabrina Examinatrice  
ne MANCER Yasmine Promotrice

**L'année universitaire  
2016 / 2017**

MA-004-442-1

*Nous dédions ce travail à :*



*Nos chers parents ...  
Nos frères et Nos sœurs ...  
Toutes nos familles ...  
Et tous nos amis ...*

*« Les grandes passions, comme les grands feux,  
sont agréables à voir de loin. »*

# Remerciement

Nous remerciant DIEU tout puissant pour la puissance, la volonté, le courage et la patience qu'il nous a donné durant toutes les années d'étude.

C'est avec un grand plaisir nous tenons à exprimer nos profondes reconnaissances à notre promotrice, Mme Manser Yasmine, qui n'a pas épargné le moindre effort dans l'encadrement de ce mémoire. Elle a été toujours disponible pour nous orienter à entreprendre les bonnes décisions. Nous avons beaucoup apprécié ses grandes qualités morales et son extrême modestie. Qu'elle trouve dans ce travail le fruit de son effort et l'expression de notre profonde gratitude. Nous exprimons notre grande gratitude et nos profonds respects aux membres du jury. Nous les remercions d'avoir accepté d'évaluer notre modeste travail.

Pour leurs encouragements, nos plus profonds remerciements vont à nos parents. Tout au long de notre cursus, ils nous ont toujours soutenus, encouragés et aidés. Qu'ils trouvent, dans la réalisation de ce travail, l'aboutissement de leurs efforts ainsi que l'expression de notre plus affectueuse gratitude.

Nous souhaiterions remercier également le reste de nos enseignants, nos amis, nos collègues et nos proches et familles pour leur encouragement tout au long de la réalisation de ce mémoire.

Nous remercions aussi tous le personnel du département informatique au sein de l'université Blida1 (Saad DAHLAB) pour leur sympathie.

A tous qui nous ont aidés de près ou de loin dans ce travail.

« La reconnaissance est la mémoire du coeur »  
Hans Christian Andersen

Feirouz & Assma.

## Résumé

Le Cloud Computing est apparu comme un nouveau paradigme informatique qui consiste à fournir des environnements informatiques fiables, particularisés et dynamiques, basés sur une meilleure qualité de service, meilleur temps de réponse et disponibilité de l'infrastructure.

Les leaders du marché de technologie de l'information, par exemple, Microsoft, Google et Amazone, changent largement vers des solutions à base de cloud. Cependant, il y a l'isolement dans les mises en œuvre de cloud fournies par les fournisseurs. L'interopérabilité limitée peut causer qu'un utilisateur adhère à un seul fournisseur de cloud. Ainsi, la migration exigée par une application/donné d'un fournisseur à un autre peut nécessiter un effort significatif pour adapter les standards du nouveau fournisseur et pour la mise en œuvre.

Le présent mémoire introduit une nouvelle solution d'interopérabilité à base de broker de nature web service avec le concept de fédération horizontale pour la couche SaaS (Software as a Service) dans le but de garantir l'automatisation des interactions dans un Cloud hybride.

Le modèle d'architecture de courtier de Cloud proposé dans ce principe a été mis en œuvre et testé dans un environnement Java en utilisant le simulateur CloudSim.

**Mots clés :** Cloud Computing, interopérabilité, Cloud hybride, Software-as-a-Service, Broker, web service, fédération horizontale, CloudSim.

## **Abstract**

Cloud Computing appeared as new IT paradigm which consists in supplying reliable, particularized and dynamic computing environments, based on a better quality of service, better time of answer and availability of the infrastructure.

Information technology market leaders, e.g., Microsoft, Google, and Amazon, are extensively shifting toward cloud-based solutions. However, there is isolation in the cloud implementations provided by the cloud vendors. Limited interoperability can cause one user to adhere to a single cloud provider; thus, a required migration of an application or data from one cloud provider to another may necessitate a significant effort and/or full-cycle redevelopment to fit the new provider's standards and implementation. The ability to move from one cloud vendor to another would be a step toward advancing cloud computing interoperability and increasing customer trust.

The present report introduces a new cloud broker solution to fill the interoperability gap between different software-as-a-service providers, in a horizontal federation layer

The proposed broker architecture was implemented and tested in Java environment by using the simulator CloudSim.

**Keywords:** Cloud Computing, interoperability, Cloud hybrid, Software-as-a-Service, Broker, web service, Horizontal federation, CloudSim.

## المخلص

برزت الحوسبة السحابية كنموذج جديد لتكنولوجيا المعلومات، تسمح بتوفير بيئات التكنولوجيا المعلومات الناجعة، المتخصصة والديناميكية على أساس أجود خدمة، أحسن وقت للاستجابة وأفضل إتاحة للبنية التحتية.

إن قادة سوق تكنولوجيا المعلومات، على سبيل المثال، مايكروسوفت وجوجل وأمازون، يتجهون في نطاق واسع إلى الحلول القائمة على السحابة إلا أن هناك عزلة في تنفيذها.

التوافقية المحدودة للحواسيب السحابية تدفع المستخدم للانضمام إلى مزود خدمة سحابة واحدة وأيضاً ترحيل التطبيقات والمعلومات من مورد خدمة للآخر يتطلب مجهود معتبر لتكييف معايير المزود الجديد ولتنفيذه.

تقدم هذه المذكرة حل جديد لتوافقية الحواسيب السحابية، يكون على أساس وسيط من طبيعة خدمة ويب. الحل مزود بمفهوم الاتحاد الأفقي لطبقة البرمجيات مع ضمان خدمة الية لتفاعلات السحابة الهجينة.

تم تنفيذ وتطبيق النموذج الهندسي المقترح للوسيط في هذا المجال ومن ثم اختباره في بيئة جافا باستخدام المتظاهر كلودسيم.

كلمات البحث: الحوسبة السحابية، التوافقية، السحابة الهجينة، البرمجيات كخدمة، الوسيط، خدمة ويب، الاتحاد الأفقي، كلودسيم.

<b>Introduction générale.....</b>	<b>2</b>
<b>CHAPITRE I : CLOUD COMPUTING.....</b>	<b>5</b>
I.1 Introduction.....	5
I.2 Cloud Computing.....	5
I.2.1 Définition.....	5
I.2.2 Les caractéristiques du Cloud Computing.....	6
I.2.3 Les services du Cloud.....	7
I.2.4 Modèles de déploiement de NIST.....	11
I.2.5 Les éléments constitutifs d'un Cloud Computing.....	13
I.2.6 Les Avantages du Cloud Computing.....	15
I.2.7 Les inconvénients du Cloud Computing.....	16
I.2.8 Les Défis du Cloud Computing.....	17
I.3 Evolution du Cloud Computing.....	19
I.3.1 Les étapes d'évolution du Cloud Computing :.....	19
I.3.2 Evolution vers le multi-Cloud.....	20
I.3.3 Inter-Cloud.....	21
I.4 Conclusion.....	22
<b>CHAPITRE II : INTEROPERABILITE DANS LES SI.....</b>	<b>23</b>
II.1 Introduction.....	23
II.2 La collaboration entre systèmes d'information.....	23
II.2.1 Définition.....	23
II.2.2 Les formes de collaboration entre systèmes d'information.....	24
II.2.3 Collaboration et le besoin de l'interopérabilité.....	25
II.3 L'interopérabilité.....	26
II.3.1 Définition.....	26
II.3.2 Les types de l'interopérabilité.....	26
II.3.3 Les niveaux d'abstraction de l'interopérabilité.....	27

II.3.4	Barrières de l'interopérabilité.....	29
II.3.5	Les exigences de l'interopérabilité en SI .....	30
II.3.6	Les Approches de l'interopérabilité.....	32
II.3.7	La médiation pour l'interopérabilité des SI.....	34
II.4	Conclusion.....	35
<b>CHAPITRE III : ETAT DE L'ART SUR LES SOLUTIONS DE L'INTEROPERABILITE DU CC.....</b>		<b>36</b>
III.1	Introduction.....	36
III.2	L'inter-cloud.....	37
III.2.1	Définition.....	38
III.2.2	Concepts liés à l'inter-cloud.....	39
III.3	L'inter-cloud et l'interopérabilité.....	40
III.3.1	Les dimensions conceptuelles de l'inter-cloud .....	41
III.4	Solutions de l'interopérabilité.....	41
III.4.1	Les standards.....	42
III.4.3	Discussion :.....	64
III.5	Conclusion :.....	65
<b>CHAPITRE IV : LA SOLUTION A BASE DE BROKER.....</b>		<b>66</b>
IV.1	Introduction.....	66
IV.1.1	Conception.....	66
IV.2	La solution proposée.....	67
IV.2.1	Architecture globale.....	67
IV.2.2	Description du composant médiateur.....	72
IV.2.3	Description des interactions.....	79
IV.2.4	Exemples des Scénario.....	83
IV.2.5	discussion :.....	84
IV.3	Conclusion :.....	85

CHAPITRE V : VALIDATION.....	86
V.1 Introduction.....	86
V.2 Langage et environnement de développent.....	86
V.2.1 Langage de programmation java.....	86
V.2.2 Netbeans environnement de développement.....	87
V.2.3 Java EE and Web Application.....	87
V.2.4 Glass fish.....	88
V.2.5 CloudSim.....	88
V.3 Description de l'application :.....	92
V.3.1 Interface principale.....	92
V.3.2 Simulation d'un seul cloud.....	93
V.3.3 Simulation de la solution Interopérabilité).....	98
V.4 Conclusion.....	108
Conclusion Générale.....	109
Bibliographie :.....	111

## Liste des acronymes

API = Application Programming Interface  
ATHENA = Advanced Technologies for Interoperability of Heterogeneous Enterprise Networks and their Applications.  
CC = Cloud Computing  
CRUD= Create Read Update Delete  
*CSP = Cloud Service Provider*  
CSU = Cloud Service User  
DC= Data Center  
DMTF = Distributed Management Task Force  
DSI = Direction des Systèmes d'Information  
EC2 = Elastic Compute Cloud  
EDA = Event-Driven Architecture  
HaaS = Hardware as a Service  
IaaS = Infrastructure as a Service  
ICSB= Inter-Cloud Service Broker  
ISO = International Organization for Standardization.  
IT = Information Technology  
NaaS = Network as a Service  
NIST = National Institute of Standards and Technology  
OASIS = Open Source API and platform for multiple clouds  
OGF = Open Grid Forum  
OMG = Object Management Group  
OSI = Open Systems Interconnection.  
PaaS = Platform as a Service  
PC = Personal Computer  
SaaS = Service as a Service  
SLA = Service-Level Agreement  
SNIA= Storage Networking Industry Association  
SOA = Service-oriented architecture  
SOAP = Simple Object Access Protocol.

UDDI = Universal Description, Discovery and Integration.

UEML: Extended Enterprise Modeling Language

VAM = Vulnerability Analysis and Mitigation

VM = Virtuel Machine.

WSDL = Web Services Description Language

XML = Extensible Markup Language.

XP = eXtreme Programming.

## Liste des figures

<i>Figure 1 : Structure du mémoire</i> .....	4
<i>Figure 2 : Les niveaux d'interaction entre l'utilisateur et le Cloud [9]</i> .....	9
<i>Figure 3 : Les modèles de déploiement Cloud [16]</i> .....	12
<i>Figure 4 : Évolution des modèles de déploiement [17]</i> .....	13
<i>Figure 5 : Les avantages du Cloud Computing [21]</i> .....	16
<i>Figure 6 : Evolution des concepts de déploiement [17]</i> .....	20
<i>Figure 7 : Evolution des concepts de déploiement vers le multi-Cloud [31]</i> .....	21
<i>Figure 8 : Exemple d'interopérabilité entre Clouds [31]</i> .....	22
<i>Figure 9 : Les différentes formes de collaboration [36]</i> .....	24
<i>Figure 10 : Dimensions et les niveaux d'abstraction de l'interopérabilité [47]</i> .....	27
<i>Figure 11 : Les Exigences de l'interopérabilité de SI [50]</i> .....	31
<i>Figure 12 : Un médiateur pour l'interopérabilité des SI [47]</i> .....	35
<i>Figure 13 : Broker de NIST [1]</i> .....	47
<i>Figure 14 : les services de CMBS [76]</i> .....	48
<i>Figure 15 : Composant de CMBS [76]</i> .....	49
<i>Figure 16 : Architecture du projet mOSAIC [77]</i> .....	50
<i>Figure 17 : Architecture du STRATOS [78]</i> .....	51
<i>Figure 18 : Architecture du CompatibleOne [79]</i> .....	53
<i>Figure 19 : Architecture de projet Inter-Cloud [56]</i> .....	54
<i>Figure 20 : Architecteur de Contrail [82]</i> .....	57
<i>Figure 21 : Architecture du Prologue [87]</i> .....	59
<i>Figure 22 : les rôles et les interactions au sein d'une architecture orientée service, (web service)</i> .....	69
<i>Figure 23 : Architecture de courtage</i> .....	71
<i>Figure 24 : Architecture De broker</i> .....	73
<i>Figure 25 : Ontologie de SLA</i> .....	74
<i>Figure 26 : description du web service en WSDL</i> .....	76
<i>Figure 27 : L'ontologie générale du cloud Computing</i> .....	78
<i>Figure 28 : structure d'une Tâche</i> .....	79
<i>Figure 29 : Automate des interactions dans le Cloud hybride</i> .....	80
<i>Figure 30 : Diagramme de classe de la solution proposée</i> .....	81
<i>Figure 31 : Diagramme de séquence de l'interopérabilité de cloud hybride à base d'un médiateur (web service)</i> .....	82
<i>Figure 32 : Le scénario d'interopérabilité de cloud hybride via un service web broker</i> .....	83
<i>Figure 33 : l'architecture du cloudSim [96]</i> .....	90
<i>Figure 34 : interface principale</i> .....	93
<i>Figure 35 : processus de simulation</i> .....	94
<i>Figure 36 : Configuration de Datacenter</i> .....	95
<i>Figure 37 : configuration des VM</i> .....	96
<i>Figure 38 : cofiguration du Cloudlet</i> .....	97
<i>Figure 39 : graphe résultat de la simulation d'un seul cloud</i> .....	98

# Introduction générale

Les utilisateurs du Cloud Computing gagnent en autonomie et en temps de déploiement. Ils bénéficient des innovations issues du web. De plus, la composition d'applications sur le Cloud leur ouvre des possibilités de collaboration inédites, entre les informations issues du web et celles issues de leurs applications métiers.

Des plateformes comme celles de Salesforce ou Amazon ont aujourd'hui atteint un niveau de maturité satisfaisant pour les entreprises : elles offrent des services de qualité et des garanties de disponibilité supérieures à celles de beaucoup de DSI. C'est pourquoi, dès qu'elles ont surmonté leurs craintes au sujet de la confidentialité, les entreprises externalisent leurs applications non stratégiques afin de recentrer leur direction des systèmes d'information sur leur cœur de métier.

Dans les modèles de déploiement de Cloud y a le modèle hybride. Le Cloud hybride est composé d'un Cloud privé et d'un Cloud public. Pour un consommateur, un Cloud privé offre des avantages en matière de sécurité par exemple par rapport au Cloud public, mais coute plus cher. Parmi les utilisations possibles du Cloud hybride, un utilisateur qui utilise déjà un Cloud public et veut pour une période précise avoir recours un Cloud privé pour sécuriser des données confidentielles par exemple. Cette collaboration entre le Cloud privé et le Cloud public nécessite un mécanisme d'interopérabilité.

Cependant, de nombreux fournisseurs de Cloud Computing proposent leurs Clouds selon leurs propres règles. Les descriptions des différentes ressources de Clouds diffèrent également pour chaque fournisseur. Ainsi, Il devient difficile de trouver des ressources nuageuses appropriées pour les caractéristiques d'une application.

## Problématique

Actuellement, il y a un manque d'interfaces communes et/ou standardisées (API) entre les différents Clouds freinant l'interopérabilité entre fournisseurs de Clouds, la migration de ressources (machines virtuelles, données et applications) d'un Cloud à un autre et l'interaction entre Clouds. De plus, la rude concurrence entre les grands

fournisseurs les pousse à utiliser des solutions propriétaires selon leurs règles.

Pour résoudre le problème d'interopérabilité, l'utilisation d'un médiateur (broker) peut être une solution prometteuse car elle peut réduire la nécessité de devoir modifier le Cloud.

### **Objectifs**

1. Etude comparative des solutions existantes pour le courtage dans les Clouds.
2. Définir un modèle à base de Broker pour l'interopérabilité dans un Cloud hybride. Cette définition doit prendre en compte les aspects suivants :
  - a. Description de l'architecture de courtage.
  - b. Description du composant médiateur.
  - c. Description de l'interaction.
3. Validation de la solution proposée par une expérimentation en utilisant le simulateur CloudSim.

### **Organisation du mémoire**

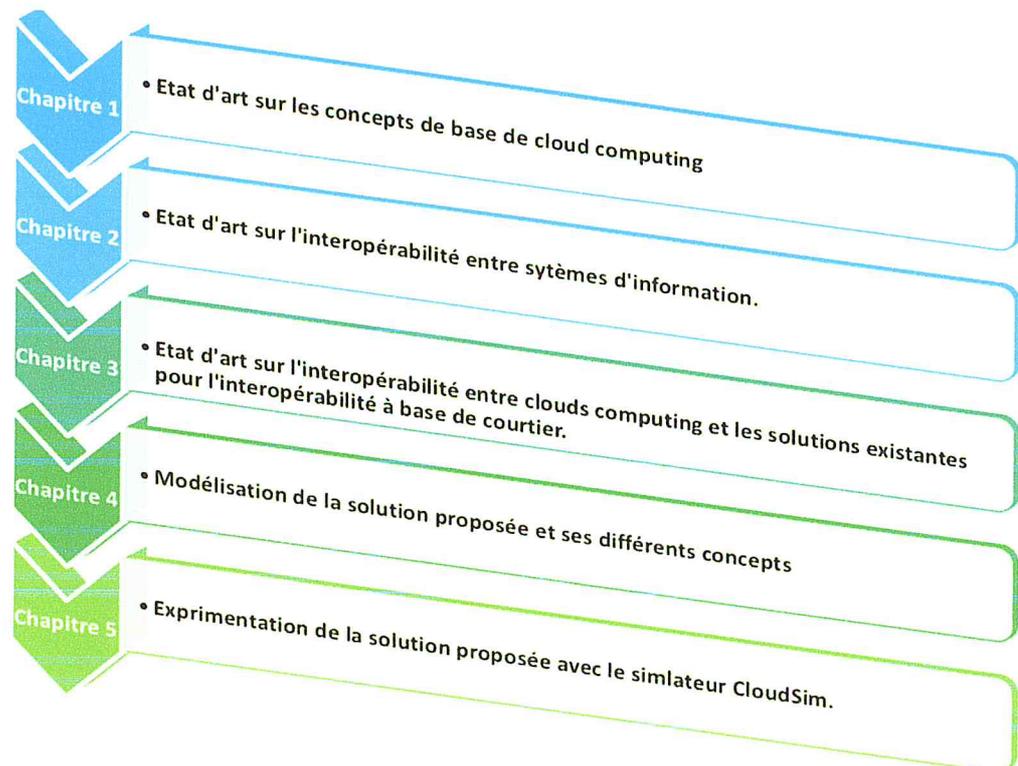
Comme illustré dans la figure 1, ce mémoire comporte cinq principaux chapitres qui sont :

**Chapitre 1 :** présente le concept du CC<sup>1</sup>, nous confrontons les points de vue des scientifiques et des industriels quant à la définition du CC. Ce chapitre introduit les caractéristiques, les services et les modèles de déploiement du CC. Il décrit aussi les défis, les étapes de l'émergence et de l'évolution de ce paradigme. Nous parlerons également de ces obstacles et ces avantages ainsi que son évolution vers l'inter-cloud.

**Chapitre 2 :** présente une vue globale sur les concepts de l'interopérabilité entre systèmes d'information : types, préoccupations, besoins, barrières et approches de l'interopérabilité.

---

<sup>1</sup> CC: Cloud Computing.



*Figure 1 : Structure du mémoire*

**Chapitre 3 :** dans ce chapitre, nous présentons un état de l'art sur l'interopérabilité entre Clouds et les solutions existantes à base de courtier. Suivi d'une étude comparative des solutions décrites déjà en prenant en compte plusieurs critères qui distinguent les solutions (architecture, couche, SLA, organisme, standard, etc.). Enfin, nous clôturons ce chapitre par une discussion sur les solutions présentées.

**Chapitre 4 :** illustre notre contribution à travers une architecture de courtage à base de broker pour l'interopérabilité dans un Cloud hybride. Dans ce chapitre, nous modélisons l'architecture globale de courtage qui sera suivi par une description détaillée de ses composants.

**Chapitre 5 :** présente la validation de l'architecture de courtage proposée dans le Cloud hybride. Nous présentons aussi le simulateur CloudSim et son architecture. Par la suite, nous présenterons l'ensemble des tests réalisés ainsi que leur interprétation.

# Chapitre I:

## *CLOUD COMPUTING*

## I.1 Introduction

Ce chapitre introduit, à travers une revue de la littérature, le concept du Cloud Computing, ses caractéristiques, ses services et ses modèles de déploiement. Par la suite, nous présentons ses avantages, ses inconvénients et les difficultés qui s'opposent au cloud. Dans la deuxième partie de ce chapitre, nous exposons l'évolution du Cloud Computing et le besoin de l'inter-cloud.

## I.2 Cloud Computing

### I.2.1 Définition

Le Cloud Computing, ou informatique en nuages, est un environnement de stockage et d'exécution flexible et dynamique qui offre à ses utilisateurs des ressources informatiques à la demande via internet. Plusieurs définitions du concept de Cloud Computing ont été proposées :

D'après le NIST<sup>2</sup> [1], le Cloud Computing (ou informatique dans les nuages) est défini comme

*« Un modèle informatique qui permet un accès facile et à la demande par le réseau à un ensemble partagé de ressources informatiques configurables (serveurs, stockage, applications et services) qui peuvent être rapidement provisionnées et libérées par un minimum d'efforts de gestion ou d'interaction avec le fournisseur du service. Ce modèle de nuage est composé de cinq caractéristiques essentielles, trois modèles de service et quatre modèles de déploiement ».*

Cependant, la définition suggérée par Marston et al [2] est considérée comme la définition la plus complète, basée sur les aspects commerciaux du phénomène de Cloud Computing :

*« Le Cloud Computing est un modèle de service technologique de l'information où les services informatiques (matériels et logiciels) sont livrés à la demande aux clients sur un réseau en mode libre-service, indépendamment du périphérique et de l'emplacement. Les ressources nécessaires pour fournir la qualité de service requise sont partagées, évolutives dynamiquement, rapidement provisionnées, virtualisées et libérées avec une interaction minimale avec les fournisseurs de services. Les utilisateurs paient le service comme une dépense d'exploitation sans encourir une dépense d'investissement initiale significative, les services en nuage utilisant un système de comptage qui divise la ressource informatique en blocs appropriés »*

Tout comme le SOA<sup>3</sup>, le Cloud Computing est en mal d'une définition unique. Cela est dû au fait que le Cloud Computing est avant tout un modèle englobant tant de principes métiers que technologiques. Chaque acteur du marché définit donc le

---

<sup>2</sup> National Institute of Standards and Technology.

<sup>3</sup> Service-oriented architecture

Cloud en fonction de sa compréhension du modèle et comment il s'adapte par rapport à son offre de marché. Mais nous devons nous attendre à ce que les organismes de définition de standards prennent le relais pour donner une base valide du modèle et acceptée par le plus grand nombre d'intervenants. C'est déjà le cas aux États-Unis où l'institut national des standards élabore sa propre définition du Cloud. De leurs côtés, l'OMG<sup>4</sup> et The Open Group mettent en place des groupes de travail pour définir des spécifications autour du Cloud Computing. Sans chercher à poser notre définition du Cloud, nous pouvons toutefois poser les principales idées : le Cloud Computing est un modèle :

- permettant d'offrir à travers le réseau un ensemble de ressources matérielles et logicielles configurables exposées sous forme de services ;
- garantissant une très grande disponibilité de ces services ;
- minimisant l'effort de gestion du côté de l'utilisateur pour accéder à ces services [3].

### **I.2.2 Les caractéristiques du Cloud Computing**

Un système informatique est considéré comme un service Cloud s'il respecte les cinq caractéristiques suivantes définies par le NIST [1] :

- **Ressources à la demande** : capacité à fournir une ressource informatique automatiquement, sans requérir d'interaction humaine côté fournisseur.
- **Large accès réseau** : (Internet) en utilisant des terminaux standards, ordinateurs, PC<sup>5</sup>, mobile, etc.
- **Elasticité rapide** : de la fourniture des capacités illimitées, qui peuvent être augmentées ou réduites selon l'usage pour suivre la demande et le passage à l'échelle de l'utilisateur.
- **Service faisant l'objet d'un SLA<sup>6</sup>** : un contrat entre l'utilisateur et le fournisseur.
- **Service mesuré** : car ce dernier doit respecter le SLA et peut être facturé selon l'utilisation « pay-as-you-use », évaluer et garantir un niveau de performance et de disponibilité adapté aux besoins spécifiques des clients.

---

<sup>4</sup> Object Management Group

<sup>5</sup> Personal Computer

<sup>6</sup>Service-Level Agreement

Il existe aussi d'autres caractéristiques de points de vue financier, organisationnel et technique [4] :

- Pas d'investissement initial : le Cloud Computing est construit pour satisfaire l'utilisation de ressources à la demande.
- Les couts fixes deviennent variable : plutôt que s'engager à utiliser un nombre donné de ressources pour la durée d'un contrat (habituellement un a trois ans), le cloud permet à la consommation de ressources de changer en temps réel.
- Le grain de l'allocation est fin : le Cloud permet une utilisation minimale en termes de temps et de ressources de changer (par exemple heures d'utilisation de serveurs, octets de stockage)
- Adaptation rapide du nouveau matériel : peut être mis en ligne en quelques minutes pour gérer des modifications imprévues dans la demande, en interne (grosses taches de traitement) ou en externe (trafic vers un site web).

### **I.2.3 Les services du Cloud**

Un service est l'entité centrale du Cloud Computing et fait l'objet de négociations entre le fournisseur (provider) et l'utilisateur (figure 1). Par conséquent, les modèles de services sont les éléments fondamentaux du Cloud Computing [5].

Selon le NIST [1], il existe trois catégories de services qui sont :

#### ***I.2.3.1 Infrastructure as a Service (IaaS)***

Est l'idée d'acheter des ressources (infrastructure) informatiques ou un Datacenter en entier qui peuvent être réapprovisionnées en-demande, sous la forme d'un service pour répondre aux besoins. De plus, en raison des progrès rapides de la virtualisation, l'automatisation, et des stratégies de paiement, l'utilisation du HaaS<sup>7</sup> se répande de plus en plus [6].

#### ***I.2.3.2 Platform as a Service (PaaS)***

Est une plateforme de développement qui supporte la gestion du cycle de vie du logiciel. La capacité fournie au consommateur est de déployer sur l'infrastructure

---

<sup>7</sup> Hardware as a Service.

Cloud des applications créées ou acquises pour un client à l'aide des langages de programmation, des bibliothèques, des services et des outils supportés par le fournisseur. L'idée derrière le PaaS est de fournir au développeur une plateforme qui contient tous les systèmes et environnements nécessaires au développement, au test, au déploiement et à l'hébergement d'applications complexes fournies comme un service [7].

#### *1.2.3.4 Software as a service (SaaS)*

Les consommateurs de ce service n'ont pas le contrôle sur l'infrastructure qui est souvent basée sur un système multi-tenant où les applications des différents consommateurs sont organisées en un seul environnement logique afin d'assurer une optimisation de la vitesse d'exécution, de la sécurité, de la disponibilité et de la reprise d'activités. Dans ce contexte, une application multi-tenante est une application qui permet aux utilisateurs de partager les mêmes ressources matérielles, tout en leur permettant de les configurer selon leurs besoins comme dans un environnement dédié [8].

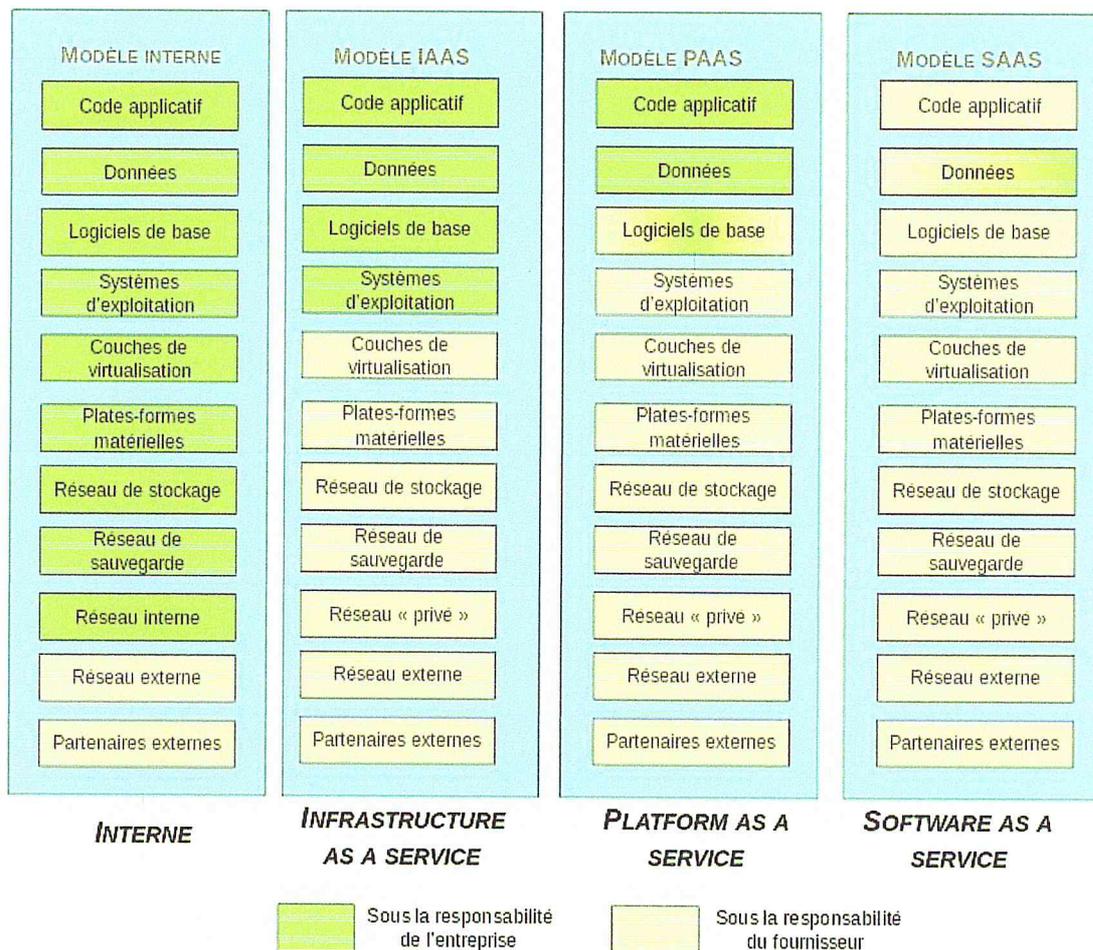


Figure 2 : Les niveaux d'interaction entre l'utilisateur et le Cloud [9].

### 1.2.3.5 X as a Service (XaaS)

A cause de la présence massive d'Internet, nous trouvons la notion du « Quoi que ce soit comme un Service ». Le modèle de XaaS supporte essentiellement l'idée que les fournisseurs offriront de plus en plus de services de dispositifs mobiles pour le développement de nouvelles générations en route [10].

Dans le tableau suivant, nous citons quelques services disponibles dans le Cloud Computing :

<b>X</b>	<b>Désignation</b>	<b>Services</b>
<b>A</b>	<b>accessoire</b>	Un service accessoire définit les fonctionnalités non stratégiques qui peuvent venir enrichir une application telles que : l'affichage d'une carte pour localiser des sites ou pour fournir le parcours d'un itinéraire ; l'ajout d'un moteur de recherche sur son site Web ; l'affichage des prévisions météo ; la collecte de flux d'information [3].
<b>CO</b>	<b>Commodité</b>	Un service est dit de commodité lorsqu'il offre à l'entreprise des fonctionnalités nécessaires pour son bon fonctionnement et la bonne communication entre collaborateurs : service de messagerie ; calendrier partagé ; partage de documents ; gestion des ressources humaines [3].
<b>M</b>	<b>Métier</b>	Un service métier constitue le cœur de fonctionnement de l'entreprise. Son fonctionnement est capital et les données traitées par de tels services peuvent requérir un niveau de sécurité et de confidentialité de sorte que l'externalisation de ce type de services métier peut s'avérer contraignante, voire impossible. Toutefois, l'externalisation de services métier est envisageable pour des entreprises qui ont pu rendre générique le comportement de leurs services ou pour lesquelles le niveau de sécurité du prestataire externe est considéré comme suffisant [3].
<b>C</b>	<b>Communication</b>	Est un service de communication audio et vidéo, services collaboratifs, communications unifiées, messagerie électronique, messagerie instantanée partage de donnée (web conférence) [11].
<b>M</b>	<b>Monitoring</b>	Est l'externalisation de services de sécurité à une équipe de sécurité tierce [11].
<b>D</b>	<b>Data</b>	Base de données en tant que service, offre des fonctions traditionnelles de bases de données. ça concerne la définition des données, le stockage et la récupération, sur une base d'abonnement sur le Web [12].
<b>N</b>	<b>Network</b>	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 client avec le fournisseur ou bien les fournisseurs entre eux. Le deuxième type de réseau est celui du centre de

		données qui relie les différentes VM et les espaces de stockage. De plus, le service de type NaaS permet aux clients 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 clients [13].
<b>Bp</b>	<b>Business Process</b>	Consiste à externaliser une procédure d'entreprise suffisamment industrialisée pour s'adresser directement aux managers d'une organisation, sans nécessiter l'aide de professionnels de l'informatique [14].

Tableau 1 : Exemples de services Cloud disponibles

#### **I.2.4 Modèles de déploiement de NIST**

Il existe quatre modèles de déploiement de CC : Privé, Communautaire, Public et Hybride. Les modèles de déploiement caractérisent globalement la gestion et la stratégie adoptées pour la livraison des services décrits précédemment [1,15].

##### ***I.2.4.1 Nuage privé***

L'infrastructure Cloud est fournie pour une utilisation exclusive par une seule organisation comprenant plusieurs consommateurs (par exemple : des unités commerciales). Il peut être détenu, géré et exploité par l'organisation, un tiers ou une combinaison d'entre eux.

##### ***I.2.4.2 Nuage communautaire***

L'infrastructure Cloud est fournie pour une utilisation exclusive par une communauté de consommateurs qui ont des préoccupations communes (mission, des exigences de sécurité, la politique et les considérations de conformité). Il peut être possédé, géré et exploité par une ou plusieurs des organisations de la collectivité, un tiers ou une combinaison d'entre eux.

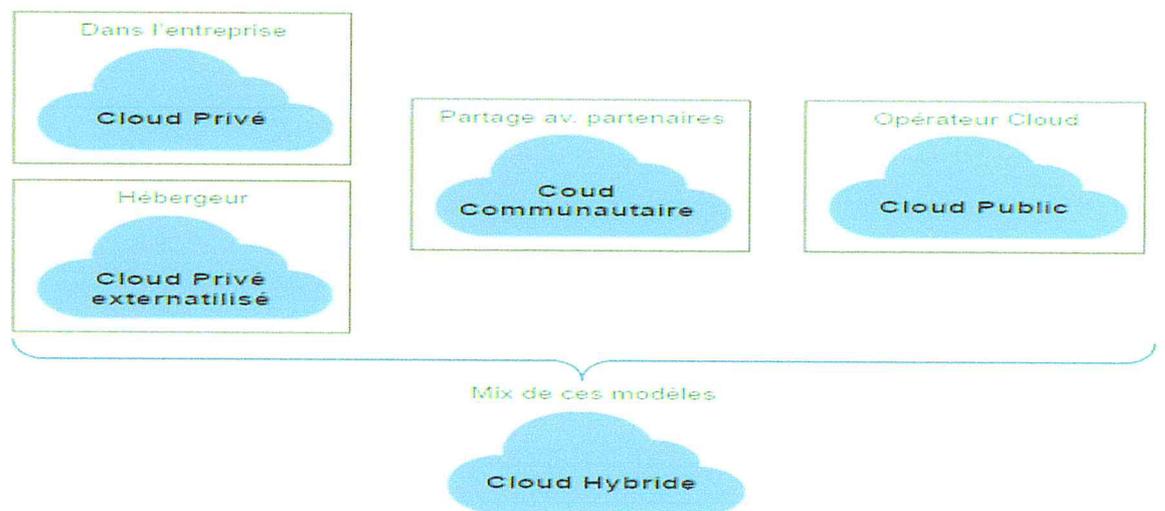
##### ***I.2.4.3 Nuage public***

L'infrastructure Cloud est fournie pour une utilisation ouverte par le grand public. Dans ce cas, il est possible qu'il soit détenu, géré et exploité par une

entreprise, une université, une organisation gouvernementale ou une combinaison d'entre eux. Il existe dans les locaux du fournisseur de Cloud.

#### ***1.2.4.4 Nuage hybride***

L'infrastructure du nuage est une composition de deux ou plusieurs nuages ayant des infrastructures distinctes (privées, communautaires ou publiques). Pour un consommateur, un Cloud privé offre des avantages en matière de sécurité par exemple par rapport au Cloud public, mais coute plus cher. Parmi les utilisations possibles du Cloud hybride, un utilisateur qui utilise déjà un Cloud public et veut pour une période précise avoir recours à un Cloud privé pour sécuriser des données confidentielles par exemple. Cette collaboration entre le cloud privé et le cloud public nécessite un mécanisme d'interopérabilité. La figure 3 suivante présente les divers modèles de déploiement :



*Figure 3 : Les modèles de déploiement Cloud [16]*

### I.2.4.5 Nouveaux modèles du Cloud

Côté fournisseurs, différents modèles coexistent, un ou plusieurs « **Cloud brokers** » (Courtiers, intermédiaires) ou « **Cloud integrators** » sont les interlocuteurs primaires du client, ils fédèrent ensuite un ensemble de « **Cloud providers** », eux-mêmes adossés à des partenaires et fournisseurs de technologie [17].

- Le **Cloud provider** de rang 1 peut être d'ailleurs le « broker », fédérant ensuite des fournisseurs de solutions Cloud de rang 2 ou N...
- Le **Cloud carrier** qui fournit l'accès réseau nécessaire à l'obtention des services, est un acteur évidemment indispensable, distinct ou confondu avec le broker et le provider.
  - Le **Cloud auditor** qui assure les missions d'Audit (performances, sécurité, ...) Outre ces acteurs natifs, SSII, éditeurs, fabricants de matériel, acteurs majeurs des réseaux sont tous potentiellement des fournisseurs d'une offre de Cloud. La figure 4 présente l'évolution de la chaîne de valeur.

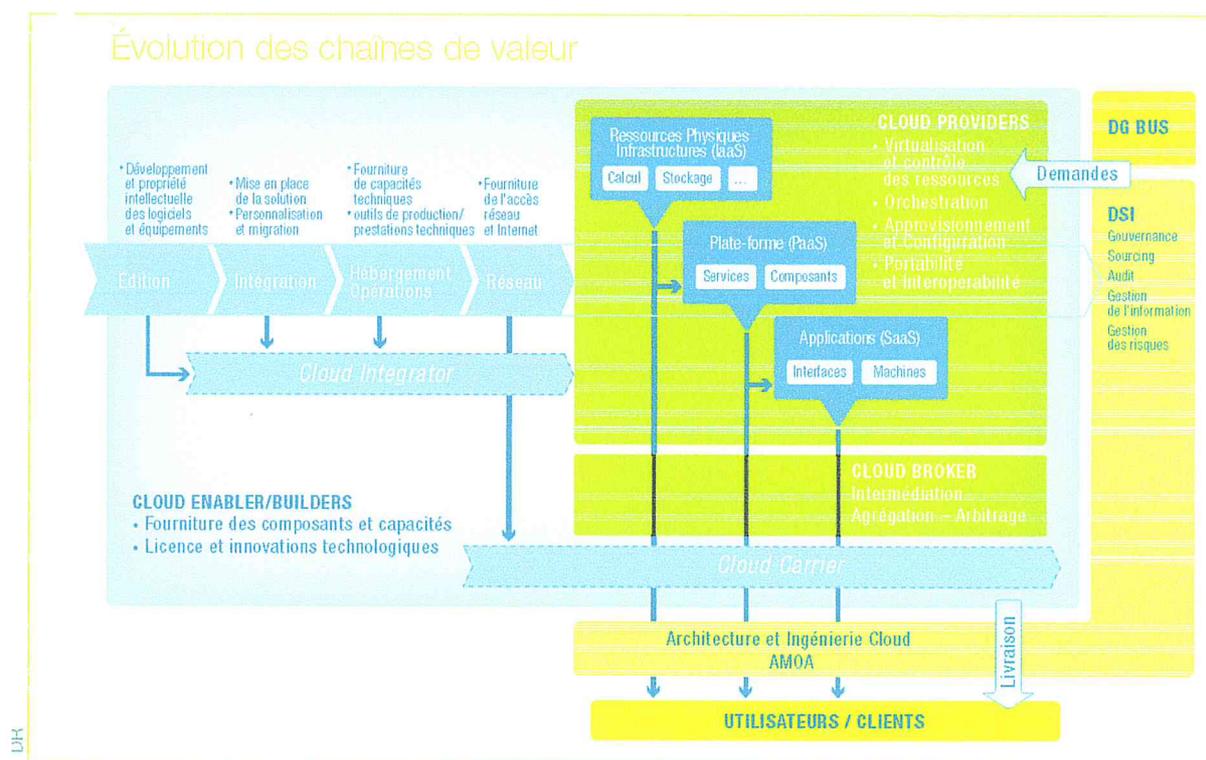


Figure 4 : Évolution des modèles de déploiement [17]

## I.2.5 Les éléments constitutifs d'un Cloud Computing

Selon [18], le CC est composé de :

- **Infrastructure** : L'infrastructure informatique du Cloud est un assemblage de serveurs, d'espaces de stockage et de composants réseau organisés de manière à permettre une croissance incrémentale supérieure à celle que l'on obtient avec les infrastructures classiques. Ces composants doivent être sélectionnés pour leur capacité à répondre aux exigences d'extensibilité, d'efficacité, de robustesse et de sécurité. Les serveurs d'entreprise classiques ne disposent pas des capacités réseau, de la fiabilité ni des autres qualités nécessaires pour satisfaire efficacement et de manière sécurisée les accords de niveau de service SLA.
- **Virtualisation** : est la principale technologie dans le Cloud. Elle consiste à partitionner une ressource physique en plusieurs ressources virtuelles. Par exemple : un serveur, un espace de stockage ou un réseau lors de la création des machines virtuelles. Elle permet d'intégrer les différents serveurs d'une manière plus flexible pour faciliter l'utilisation. Le but de la virtualisation est d'assurer la transparence d'utilisation et l'efficacité d'exploitation des ressources, ainsi que le fonctionnement des différents services avec la séparation entre de multiples locataires (utilisateurs) impliqués dans d'un matériel physique.
- **Interfaces de service** : placée entre le fournisseur et le client, est un élément de différenciation du Cloud. Elle représente un contrat qui fait respecter la proposition de valeur décrite par des SLA et des conditions tarifaires. Si le Cloud semble nouveau, c'est principalement en raison de cette interface. Elle représente la valeur d'un fournisseur et sert de base à la concurrence. Par l'ajout d'interfaces de libre-service, nous obtenons d'autres optimisations. Les clients du Cloud sont en mesure d'engager des ressources de manière automatisée sans que le service informatique soit un obstacle. L'espace de stockage et les ressources sont présentés au travers d'une interface graphique que l'utilisateur peut manipuler de manière à obtenir et à instancier une infrastructure informatique virtuelle.

- **Centre de données (Datacenter) :** Un centre de traitement de données, en anglais, « Datacenter » est un site physique sur lequel sont regroupés des équipements du système d'information de l'entreprise (mainframes, serveurs, baies de stockage, équipements réseaux et de télécommunications, etc.). Il peut être interne ou externe à l'entreprise, exploité ou non avec le soutien de prestataires. Il comprend en général un contrôle sur l'environnement (climatisation, système de prévention contre l'incendie, etc.), une alimentation d'urgence et redondante, ainsi qu'une sécurité physique élevée. Cette infrastructure peut être propre à une entreprise et utilisée par elle seule ou à des fins commerciales. Ainsi, des particuliers ou des entreprises peuvent venir y stocker leurs données suivant des modalités bien définies [19].

### **I.2.6 Les Avantages du Cloud Computing**

Le Cloud Computing offre plusieurs avantages par rapports aux approches traditionnelles. Nous reprenons quelques avantages cités dans [20, 21, 22] :

- **Baisse des charges d'exploitation.**
- **Evolutivité continue :** Les besoins des utilisateurs évoluent en permanence, de même que les exigences pour les interfaces et le stockage. Cela signifie qu'un nuage, surtout public, ne reste pas statique et évolue continuellement.
- **Accès très facile et rapide :** aux ressources informatiques.
- **Virtualisation :** Ensemble des techniques matérielles et/ou logiciels qui permettent de faire fonctionner simultanément sur une seule machine plusieurs systèmes d'exploitation (appelés machines virtuelles (VM)). Ex. : Xen, VMware, KVM, etc.
- **Élasticité rapide :** Faculté d'augmenter et diminuer dynamiquement les ressources allouées à une application.
- **Économies :** Les entreprises peuvent réduire leurs dépenses en immobilisations et utiliser les dépenses opérationnelles pour capacités informatiques. Il s'agit d'un obstacle mineur et nécessite également moins de ressources informatiques internes pour assurer le soutien du système.

informatiques. Il s'agit d'un obstacle mineur et nécessite également moins de ressources informatiques internes pour assurer le soutien du système.

- **Flexibilité** : la flexibilité du Cloud Computing permet aux entreprises d'utiliser des ressources supplémentaires aux heures de pointe pour satisfaire les demandes des consommateurs.
- **Fiabilité** : Les services utilisant plusieurs sites redondants peuvent prendre en charge la continuité des activités et la reprise après sinistre.
- **Maintenance** : Les fournisseurs de services Cloud effectuent la maintenance du système et l'accès se fait par le biais d'API<sup>8</sup> qui ne nécessite pas d'application d'installations sur PC, ce qui réduit encore les besoins de maintenance.
- **Accessible aux mobiles** : Les travailleurs mobiles ont une productivité meilleure avec le Cloud.

La figure 5 suivante résume les plus importants avantages du Cloud Computing.

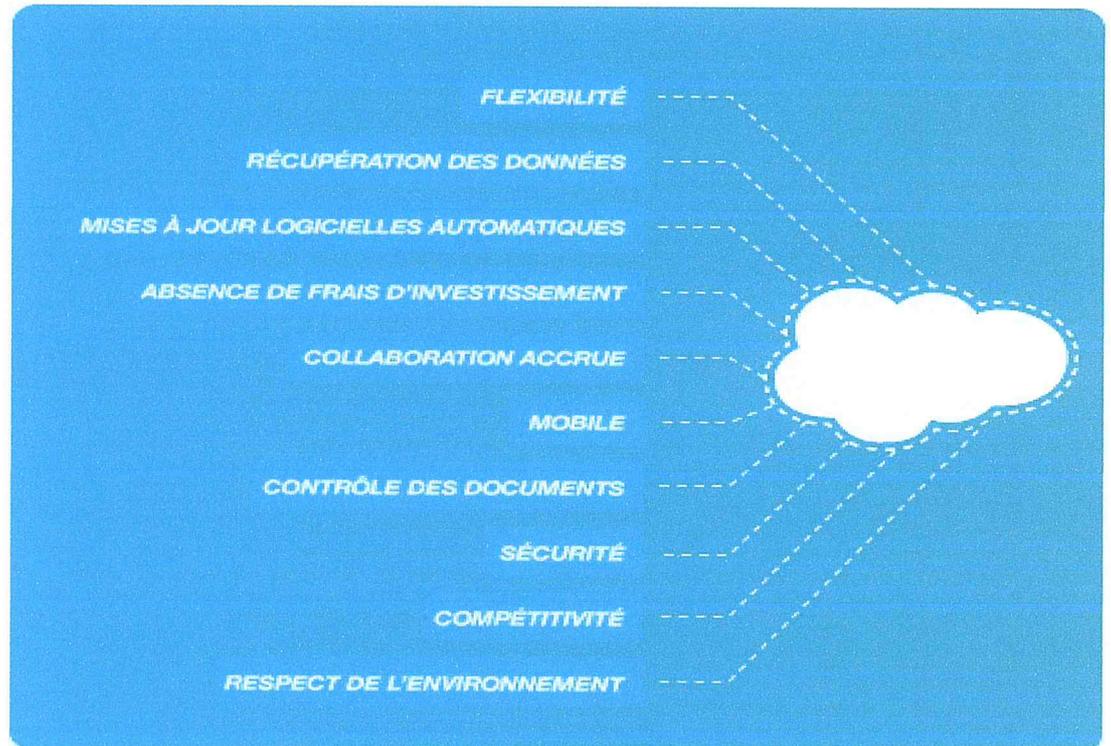


Figure 5 : Les avantages du Cloud Computing [21].

<sup>8</sup> Application Programming Interface.

### I.2.7 Les inconvénients du Cloud Computing

Le Cloud Computing présente de nombreux avantages. Cependant, certaines entreprises n'ont pas intérêt à passer à l'informatique dans les nuages, pour des raisons légales et techniques. Parmi les inconvénients que présente le Cloud Computing [23, 24] :

- Le propriétaire perd le contrôle total sur ses données : il y a lieu d'instaurer un mécanisme de confiance entre les parties intervenantes ;
- Le manque d'interopérabilité entre les différents fournisseurs de Cloud.
- Déploiement des applications vers le Cloud : comporte un coût additionnel puisque les architectures sous-jacentes peuvent différer. Les mêmes préoccupations sont soulevées lors de la migration entre différents fournisseurs ;
- Débits réseau importants : Le Cloud nécessite des débits réseau importants pour ne pas décourager les utilisateurs par des temps de réponse trop lent ;
- Incertitudes des avantages du Cloud : à long terme.

### I.2.8 Les Défis du Cloud Computing

Le CC a offert un certain nombre d'avantages aux utilisateurs et organisations qui l'ont adopté. Cependant, il y a encore un nombre de défis, qui sont actuellement relevés par les chercheurs du domaine [25, 26, 27, 28] :

- **Performance** : l'enjeu majeur de la performance arrive dans le cas de transactions et applications gérant un volume de données important. Aussi, les utilisateurs qui se trouvent géographiquement à une grande distance du fournisseur peuvent rencontrer des problèmes de latence et de retard.
- **Sécurité et confidentialité** : les entreprises sont toujours préoccupées par la sécurité de leurs données quand elles utilisent le CC. En effet, les utilisateurs sont inquiets face aux attaques et vulnérabilités des systèmes quand leurs ressources et données critiques sont à l'extérieur de leurs locaux. Ainsi les fournisseurs de CC doivent suivre les pratiques standards de sécurité comme

le paramétrage des pare-feux, délimitation de zones, segmentation du réseau, la détection d'intrusion, etc.

- **Contrôle** : certains départements informatiques sont préoccupés par le fait que le fournisseur de service Cloud a le contrôle total des plates-formes et généralement ne conçoivent pas des plates-formes spécifiques pour les entreprises et leurs pratiques commerciales.
- **Coût de la bande passante** : avec les CC, les entreprises économisent de l'argent et réduisent les coûts sur l'infrastructure matérielle et logicielle mais peuvent rencontrer une augmentation des frais de bande passante. Le coût de la bande passante peut être faible pour les petites applications orientées web, qui ne nécessitent pas le traitement d'un volume important de données, en revanche, il peut augmenter de manière significative quand le volume de données à gérer devient important.
- **Fiabilité** : le CC n'assure toujours pas une fiabilité et un fonctionnement sans interruption. Il y a eu des cas où le service a été interrompu quelques heures. Néanmoins, des travaux sont en cours pour établir des normes et définir de meilleures pratiques pour assurer la fiabilité du service.
- **Data-lockin** : une des préoccupations majeures des utilisateurs de Cloud est d'avoir des données bloquées chez un certain fournisseur. Pour [29] le lockin existe lorsque le coût du changement de la plate-forme technologique d'un vendeur vers un autre est à ce point onéreux que le client est incapable de quitter les offres du vendeur. En effet, les utilisateurs peuvent vouloir déplacer leurs données et applications et ainsi quitter un fournisseur qui ne répond pas à leurs besoins et exigences [30]. Toutefois, pour résoudre ce problème il faut d'abord assurer l'interopérabilité entre les Clouds.

### I.3 Evolution du Cloud Computing

Le Cloud va être de plus en plus intelligent. Autrement dit, il nécessitera de moins en moins une intervention humaine. De ce fait, le Cloud pourra gérer lui-même la complexité des systèmes d'exploitation que nous utilisons au quotidien.

Le Cloud Computing est une évolution logique des technologies qui ont jalonné l'histoire de l'informatique. Les systèmes ont convergé vers un modèle basé sur la consolidation et le partage des ressources [26].

Pendant que les acteurs traditionnels du marché IT<sup>9</sup> débattaient de l'ineptie de tel ou tel mouvement ou critiquaient les modèles business de ces apprentis entrepreneurs, des acteurs comme Google et Amazon n'ont cessé d'innover et de préparer leur nuage. Les techniques de virtualisation, la maturité grandissante du SOA et les technologies et standards issus du Web 2.0, ont favorisé leur croissance.

Pour ces acteurs visionnaires, le terrain était prêt pour apporter aux entreprises et aux particuliers des solutions souples, intégrables et financièrement accessibles [3].

Les premiers nuages pouvaient apparaître, en 2006, avec le lancement du service de nuage extensible pour le calcul « Amazon EC2<sup>10</sup> » apportant une réponse aux attentes d'agilité et offrant un potentiel mondial aux acteurs performants [30].

#### I.3.1 Les étapes d'évolution du Cloud Computing :

Le Cloud apporte des modifications profondes du positionnement des acteurs de l'IT sur deux chaînes de valeur distinctes [17] :

- **Cloud Enablement (1)** : Création, construction et management de services Cloud.
- **Cloud Delivery (2)** : Packaging, vente et fourniture de services Cloud

La première chaîne de valeur (1) est « classique » dans l'IT. Elle repose à la fois sur la valorisation des licences et des innovations ainsi que sur les capacités de mise en œuvre et d'infogérance. Elle est aussi novatrice de par les nouveaux modèles induits de fourniture de ressources. Différents modèles apparaissent avec le Cloud,

---

<sup>9</sup> Information Technology (Technologies de l'information).

<sup>10</sup> Elastic Compute Cloud.

essentiellement pour compenser le manque de maturité des infrastructures ou compétences clés : Cloud integrator, Cloud carrier. Leurs clients sont principalement les opérateurs de Cloud public ou les DSI<sup>11</sup> mettant en œuvre un Cloud privé. La fourniture de services Cloud (2) constitue la seconde chaîne de valeur, délivrant la fourniture d'un service final d'infrastructures (IaaS), de plates-formes (PaaS), d'applications (SaaS) ou de processus métiers. Enfin, la DSI amenée à renforcer ses rôles de gouvernance, audit et architecture dans le cadre de la définition et le déploiement de services Cloud, pourra se faire assister par des sociétés spécialisées à même d'appréhender les challenges technologiques et opérationnels de ces nouvelles offres.

Positionnement « complémentaire » : le Cloud et les nouveaux modèles économiques associés opèrent sur une transformation des acteurs, qui choisissent un positionnement multiple et parfois en complément ou compétition.

La figure 6 présente l'évolution des concepts Cloud Computing cités précédemment.

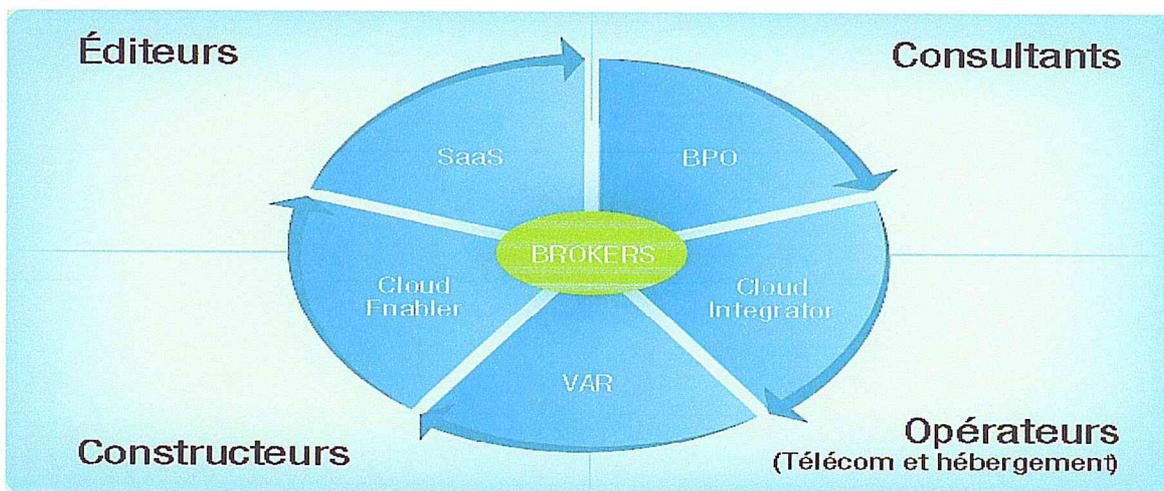


Figure 6 : Evolution des concepts de déploiement [17]

### I.3.2 Evolution vers le multi-Cloud

Beaucoup d'entreprises utilisent déjà du cloud hybride : des services dans leurs Datacenter et des services en externe, mais sans véritable gouvernance. Tant que l'utilisation est faible cela peut sembler gérable. Sachant qu'il faut appliquer à ces

<sup>11</sup> Direction des Systèmes d'Information.

de coûts, la multiplicité des réponses externes à des besoins internes. Cela risque de devenir vite ingouvernable. Un autre challenge du multi-Cloud est la compatibilité des architectures. Ce n'est pas parce que nous aurons deux API compatibles qui transmettront des fonctionnalités et des paramètres, que ces fonctionnalités seront gérées de la même façon, avec les mêmes paramètres dans chaque Cloud [31].

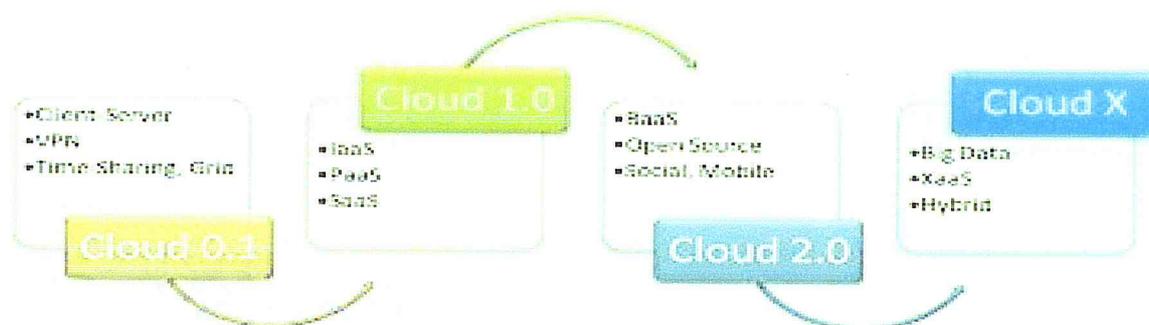


Figure 7 : Evolution des concepts de déploiement vers le multi-Cloud [31]

### I.3.3 Inter-Cloud

Le manque d'interopérabilité entre les différents fournisseurs de nuages est un problème majeur qui s'est posé en raison de la divergence des approches et des structures.

#### I.3.3.1 Besoin d'Inter-Cloud

L'Inter-nuage remédie à de telles situations où chaque Cloud utiliserait l'information, le stockage, ou n'importe quelle sorte de ressource des infrastructures d'autre Cloud. L'environnement d'Inter-Cloud fournit des avantages comme des emplacements géographiques divers, meilleure résistance d'application et surtout éviter le problème du Vendor Lock-In (situation d'enfermement propriétaire). Les avantages pour le fournisseur de nuage sont le fait d'établir des accords de niveau de service à la demande avec le client avec de meilleurs SLA [15,31]. Mais, pour pouvoir atteindre ce but, il faut d'abord résoudre les problèmes d'interopérabilité.

#### I.3.3.2 Vers l'inter-Cloud

Aujourd'hui, l'interopérabilité entre différents Cloud au niveau le plus basique, la communication entre machines est raisonnablement possible. Cela nécessite évidemment des travaux et l'établissement de standards appropriés pour l'interconnexion des réseaux, la fédération des identités, la gestion des accès et de la

sécurité. La portabilité et le management unifié (notamment par exemple le transfert automatisé de charges de travail ou de données) entre Clouds sont deux sujets plus délicats, pour lesquels des travaux de normalisation sont engagés. Ce problème devrait évoluer raisonnablement rapidement, sous l'impulsion de très grands donneurs d'ordres (dont le gouvernement américain ou l'union européenne) et de certains éditeurs de briques Cloud, qui ont tout intérêt à fournir ce type de portabilité [15,31].

La figure 8 donne un exemple de schéma d'interopérabilité entre Clouds.

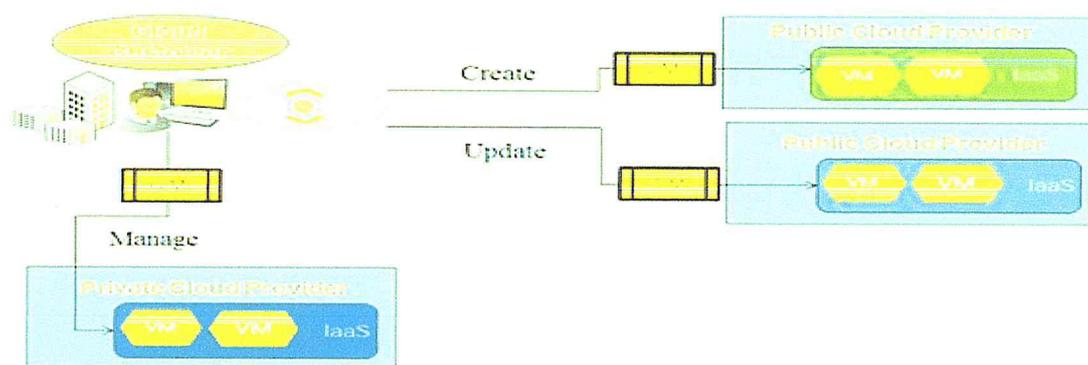


Figure 8 : Exemple d'interopérabilité entre Clouds [31]

#### I.4 Conclusion

Que ce soit pour des raisons économiques, stratégiques, ou des besoins de croissance, la portabilité des composants, et donc leur interchangeabilité, est essentielle. La négliger, et s'exposer à ne pas pouvoir bénéficier pleinement des avantages du Cloud (Élasticité, flexibilité et économies).

Dans ce chapitre, nous avons présenté les concepts de base du Cloud Computing (caractéristiques, types de déploiement, types de services, etc.), ainsi que l'évolution du Cloud Computing vers l'Inter-Cloud.

L'interopérabilité demeure un réel problème, il n'y a aucune garantie que l'on puisse migrer d'un Cloud A vers un Cloud B ou C en quelques minutes.

Dans le chapitre suivant, nous allons présenter et discuter le problème d'interopérabilité entre les systèmes d'informations.

# Chapitre II :

*L'interopérabilité entre  
systèmes d'information*

## II.1 Introduction

Pour mieux comprendre l'interopérabilité dans le Cloud Computing, il faut bien comprendre cette notion entre les systèmes d'information.

Dans ce chapitre, nous nous intéressons tout d'abord à définir la notion de collaboration entre systèmes d'information : comment les SI<sup>12</sup> établissent-ils des collaborations ? Quelles sont les formes de collaboration ? Par la suite, nous présentons les concepts de base liés à notre travail. Après, nous nous intéressons aux types qui permettent de caractériser l'interopérabilité des SI ; les niveaux d'abstraction, ses obstacles et ses exigences.

## II.2 La collaboration entre systèmes d'information

Face à l'évolution des caractéristiques du marché et de leur environnement, le challenge pour les entreprises est de collaborer, avec pour objectif de faire en sorte que leurs produits et/ou services se vendent au mieux. [32] signale que les motivations pour mettre en place un réseau d'entreprises collaboratives sont pour la plupart des motivations à court terme visant l'optimisation des coûts, des délais et de la qualité. En conséquence, la valeur ajoutée d'une collaboration doit être mesurable pour pouvoir évaluer son intérêt pour une entreprise donnée.

### II.2.1 Définition

Plusieurs chercheurs ont essayé de donner une définition avec leurs propres caractéristiques, on trouve que la collaboration est définie comme un partage mutuel et un travail commun pour atteindre un but commun de telle façon qu'une contribution de chaque personne est reconnue et la croissance professionnelle est encouragée [33].

D'après [34], « *La collaboration est définie comme le processus de communication avec le but de partager les informations qui aboutiront à une décision mutuellement convenue.* »,

---

<sup>12</sup> Système d'information.

## II.2.2 Les formes de collaboration entre systèmes d'information

Nous reprenons les niveaux de collaboration cités dans [35]. Chaque niveau englobe le niveau précédent comme illustre la figure 9.

- **Networking (ou communication):** implique la communication et l'échange d'informations pour réaliser un avantage commun.
- **Coordination :** en plus du networking, elle implique l'alignement et la synchronisation des activités mises en commun pour rendre plus efficaces les résultats attendus.
- **Coopération :** elle implique non seulement la coordination, mais aussi le partage des ressources pour réaliser des objectifs compatibles. La coopération est réalisée à travers la répartition des tâches à réaliser entre les participants.
- **Collaboration (l'intégration):** c'est un processus dans lequel les entités partagent des informations, des ressources et des responsabilités pour planifier conjointement, mettre en œuvre et évaluer un programme d'activités leur permettant de réaliser les missions communes. La collaboration implique l'engagement mutuel des participants pour résoudre ensemble un problème. Ceci implique une confiance mutuelle

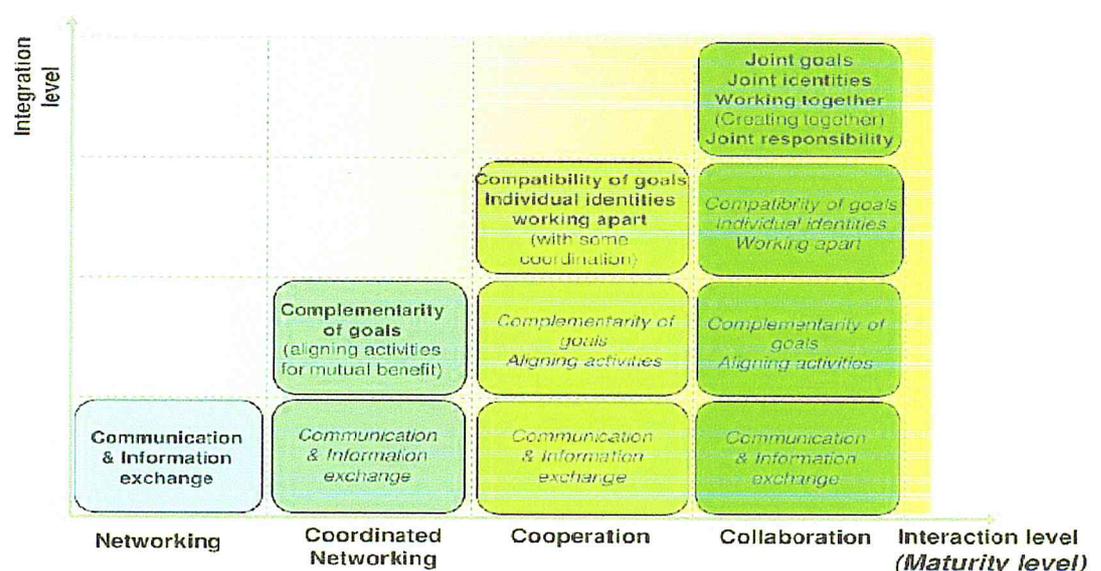


Figure N° 9 : Les différentes formes de collaboration [36]

### II.2.3 Collaboration et le besoin de l'interopérabilité

Pour rendre la collaboration possible, il est important que les entreprises aient intégré les Technologies de l'Information et de la Communication et les outils supportant la prise de décision. D'ailleurs, la coordination des opérations au niveau technique est au cœur de la notion de collaboration.

Les TIC<sup>13</sup> sont ainsi indispensables pour assurer la fluidité des échanges d'information. Il en découle que les SI jouent un rôle majeur dans la chaîne de création de la valeur des entreprises collaboratives. Pour réaliser un objectif commun, les Systèmes d'Information d'Entreprise en réseau collaborent et développent différentes interactions.

Dans ce cadre, ces systèmes sont complètement **distribués** au niveau de l'ensemble du réseau des entreprises. Ils sont **autonomes** en regard de leur propre mission. Ils sont **hétérogènes** de par leur architecture, le contenu des fonctions qu'ils assurent, leur conception (modèles, architectures...), leur langage de modélisation ainsi que leurs fonctionnalités. L'information est ainsi complètement distribuée tout au long de la chaîne de valeur et utilisée par plusieurs applications. Ceci implique que l'échange des flux informationnels devient de plus en plus complexes (quantité, diversité, hétérogénéité) et peut engendrer des problèmes d'interopérabilité. Les entreprises doivent s'assurer du niveau d'échange d'informations et précisément de la quantité, la qualité et la fréquence des informations échangées. Ceci met en évidence des questions concernant la mesure de performance de l'interopérabilité des Systèmes d'Informations ainsi que l'identification de métriques quantifiables. [37,38]

Dans un contexte de collaboration de systèmes d'information, les entreprises doivent gérer le degré d'ouverture de leurs SI vis-à-vis des autres partenaires.

---

<sup>13</sup>Technologies de l'Information et de la Communication.

## II.3 L'interopérabilité

### II.3.1 Définition

La compatibilité est la possibilité pour deux systèmes de types différents de fonctionner ensemble, notamment en utilisant les mêmes fichiers de données. Il y a souvent de multiples interprétations du terme interopérabilité dans le discours courant [39, 40]. L'interopérabilité y est décrite parfois comme un état final, d'autres fois comme une condition ou encore comme une propriété du système. La plupart des positions adoptées par les groupes et les réseaux compétents (comme l'IEEE, par exemple), « *L'interopérabilité désigne une capacité de systèmes, nativement étrangers les uns par rapport aux autres, à interagir afin d'établir des comportements collectifs harmonieux et finalisés, sans avoir à modifier en profondeur leur structure ou leur comportement individuel.* »

Selon [41] L'interopérabilité est « *La capacité que possèdent deux ou plusieurs systèmes ou composants à échanger des informations puis à exploiter les informations venant d'être échangées* ». Carney et al dans [42] étendent cette définition en ajoutant la notion d'objectif lié à l'interopération ainsi que la notion de contexte qui définit l'environnement dans lequel évolue l'ensemble des informations échangées, ils définissent l'interopérabilité comme « *la capacité d'un ensemble d'entités communicantes à échanger de l'information spécifique ; à opérer à partir de cette information, selon une sémantique commune ; dans le but d'accomplir une mission spécifiée dans un contexte donné* ».

### II.3.2 Les types de l'interopérabilité

Il existe plusieurs types d'interopérabilité ; les principaux sont décrits par [43, 44,45] :

- **L'interopérabilité technique** désigne le recours à la définition et l'utilisation d'interfaces technologiques, des normes et protocoles, en vue de créer des systèmes d'information collaboratifs fiables, efficaces et performants capables d'échanger l'information.
- **L'interopérabilité sémantique** consiste à faire en sorte que la signification des informations échangées c'est-à-dire leur sémantique ne soit pas perdue dans la procédure d'interopération et qu'elle soit préservée et comprise par les personnes, les applications et les institutions concernées (les formats des messages, la structuration, le sémantique des constituants). Elle nécessite que l'interopérabilité technique soit effective.

- **L'interopérabilité organisationnelle** est la capacité d'identifier les acteurs et les procédures organisationnelles intervenant dans la fourniture d'un service spécifique et de parvenir à un accord entre ces acteurs et à des procédures sur la manière de structurer leur interaction. En d'autres termes, il s'agit de définir les «interfaces d'entreprise» et s'intéresser aux rôles des entités et des acteurs en interaction avec les systèmes d'information. Elle nécessite que l'interopérabilité sémantique soit effective.

### II.3.3 Les niveaux d'abstraction de l'interopérabilité

Dans l'étude du concept de collaboration, il est indiqués que dans les collaborations, les acteurs échangent ou partagent différentes entités telles que des données ou des informations à l'aide de différents outils de communication. Par exemple, un échange ou partage peut inclure un transfert manuel d'un document d'un acteur à un autre ou il pourrait impliquer le partage d'un service via une application. Selon Chen la deuxième dimension de l'interopérabilité, les préoccupations de l'interopérabilité, considère les entités échangées entre les acteurs. [46]

*« Dans une entreprise, les données sont utilisées par des services (ou fonctionnent pour fournir un service). Les services (des fonctions/des activités) sont employés par des processus pour réaliser l'affaire de l'entreprise »*

Donc il existe quatre niveaux sont : données, services, processus et business.

Par contre [47], comme présenté la figure 10 classifie les entités échangées en 6 niveaux : Donnée/Information, Composante, Application, Système, Entreprise, Communauté.

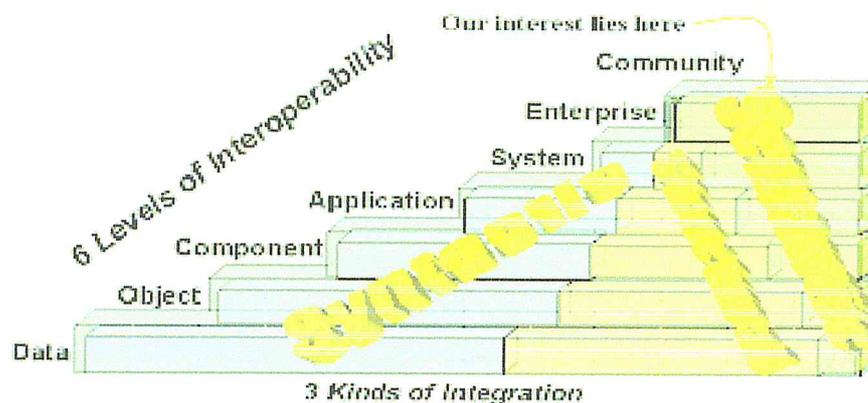


Figure 10 : Dimensions et les niveaux d'abstraction de l'interopérabilité [47]

Concernant un modèle pour l'interopérabilité des systèmes d'information (LISI<sup>14</sup>) proposé par (C4ISR) [48] les préoccupations de l'interopérabilité sont classées en quatre catégories : procédure, application, infrastructure et donnée (PAID) :

- **Procédure** : ce niveau comprend les pratiques, les architectures et les normes pour l'échange d'informations
- **Application** : ce niveau comprend les logiciels pour l'échange, le traitement et la manipulation de l'information
- **Infrastructure (ou les architectures)** : ce niveau représente l'environnement technique (matériel, réseau, système) qui permet les interactions entre les applications
- **Donnée** : ce niveau représente les formats et les protocoles d'échanges d'informations à travers une sémantique commune.

Les préoccupations d'interopérabilité, ces travaux se focalisent sur le niveau données/information. En effet, dans ces travaux, il est considéré que les collaborations entre les concepteurs sont réalisées par les échanges des fichiers de données numériques. Le niveau de données/informations est énoncé dans plusieurs références car tous les échanges comprennent inévitablement des échanges de données. Par exemple, dans le cas du partage d'un service, les données décrivant son fonctionnement sont également échangées. L'OTAN [49] propose quatre degrés pour les données et les informations échangées :

- **Degré 1** : échange de données non structurées et interprétables par l'Homme telles qu'un texte libre trouvé dans les résultats analytiques et documents textuels.
- **Degré 2** : échange de données structurées et interprétables par l'Homme, destiné au traitement qui nécessite la réception et l'envoi de message et une compilation manuelle.
- **Degré 3** : partage transparent de données qui implique le partage automatique des données entre les systèmes basés sur un modèle d'échange commun.

---

<sup>14</sup>Levels of Information Systems Interoperability.

- **Degré 4** : partage transparent d'information, une extension du degré 3, pour l'interprétation universelle de l'information via le traitement de données basé sur les applications collaboratives.

### II.3.4 Barrières de l'interopérabilité

La barrière de l'interopérabilité est un concept fondamental dans le domaine d'interopérabilité. De nombreux problèmes d'interopérabilité sont Spécifiques à des domaines d'application particuliers. Ceux-ci peuvent être des choses comme un support pour des attributs particuliers, ou un accès particulier aux régimes de contrôle. Le terme «barrière», ou «incompatibilité» entrave le partage et l'échange d'informations. Trois catégories d'obstacles étaient identifiés conceptuels, technologiques et organisationnel. [50]

- **Obstacles conceptuels (syntaxiques)** : Concernent les aspects syntaxiques et l'incompatibilité sémantique de l'information à échanger. Ces problèmes concernent la modélisation au niveau élevé Abstraction (modèles d'entreprise) ainsi que le niveau de programmation (capacité faible de la représentation sémantique de XML<sup>15</sup>). L'incompatibilité syntaxique peut être trouvée sa solution en l'initiative (UEML<sup>16</sup>, 2002) qui vise à faire un modèle neutre pour permettre la cartographie entre les divers modèles d'entreprise. La sémantique des logiciels n'est pas clairement définie afin de permettre une compréhension d'information sans ambiguïté. À l'heure actuelle, les techniques les plus connus pour résoudre ce problème est la sémantique d'annotation et la réconciliation en utilisant l'ontologie [51].
- **Obstacles technologiques** : Concernent l'utilisation d'informatique ou les TIC pour échanger des informations. Les barrières technologiques sont l'incompatibilité d'architecture, des protocoles, de plates-formes informatiques, infrastructure, système d'exploitation etc. l'absence d'un ensemble de normes compatibles permettant le partage et d'échange des informations.

---

<sup>15</sup> XML: *Extensible\_Markup\_Language*.

<sup>16</sup>UEML: *Extended\_Enterprise\_Modeling\_Language..*

L'incompatibilité des outils utilisés pour coder / décoder les Informations échangées.

- **Obstacles organisationnels** : la Structure organisationnelle est liée à la manière d'assigner la responsabilité et l'autorité. Ceux-ci peuvent être considérés comme des «Technologies» ou «facteurs humains» qui peuvent créer des obstacles à l'interopérabilité.

- La responsabilité doit être définie pour permettre à deux parties sachant qui est responsable de quoi (processus, Données, logiciels, ordinateurs, ...).

- Si la responsabilité d'entreprise n'est pas clairement et explicitement définie l'interopérabilité entre deux systèmes est obstruée.

- L'autorité est un concept organisationnel, il définit les rôles et les autorisations pour créer, modifier, maintenir des données, des processus, des services, etc.

- La structure organisationnelle se réfère au style de responsabilité, d'autorité et de la prise de décision selon l'une des organisations (Décentralisées, hiérarchiques ou réseau).

Les obstacles organisationnels constituent des obstacles qui proviennent des êtres humains comparé aux barrières conceptuelles (centrées sur les problèmes d'information) et Technologiques (concernant les problèmes de machines).

### **II.3.5 Les exigences de l'interopérabilité en SI**

La communication est une condition fondamentale permettant l'interopérabilité. Elle concerne principalement L'interconnexion de systèmes et d'équipements, ainsi les moyens de communication, elle concerne des protocoles de communication des couches 1 à 4 du modèle OSI<sup>17</sup> et des interfaces des couches 5 à 7 du modèle OSI.

---

<sup>17</sup>Open Systems Interconnection.

Dans le domaine de l'interopérabilité des entreprises, il y'a quatre exigences d'interopérabilité identifiées comme illustré la figure 11 Adapté par ATHENA<sup>18</sup> (2005) : Données, services, processus et affaires.

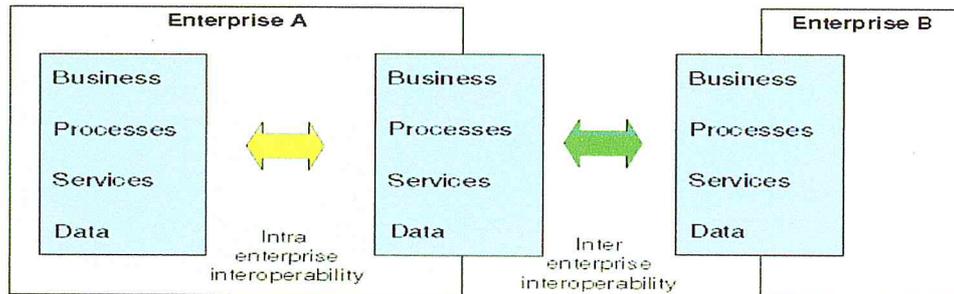


Figure 11 : Les Exigences de l'interopérabilité de SI [50].

- **L'interopérabilité des données** : L'interopérabilité des données concerne la recherche et le partage d'informations provenant de bases de données hétérogènes, et qui peuvent, en outre, se connecter avec différentes machines et différents systèmes d'exploitation et différents systèmes de gestion de bases de données. L'interopérabilité des données joue un rôle essentiel dans l'interopérabilité SI. Elle s'intéresse à la capacité d'échanger à la fois des données non électroniques (documents) et des données transportables (fichiers de données, données stockées dans une base de données) en utilisant des données / informations échangées.
- **L'interopérabilité des services** : Identifier, composer et exploiter ensemble divers applications en résolvant les différences syntaxiques, sémantiques ainsi que trouver les connexions aux différentes bases de données hétérogènes. Le terme «service» n'est pas limité aux applications mais aussi aux fonctions de la société ou au réseau entreprises. L'interopérabilité des services porte sur la capacité d'échanger des Services (travaux) entre partenaires. Elle connaît deux problèmes principaux :
  - L'échange de service entre un service demandeur et un prestataire de services.
  - Interconnexion entre différents services pour former un service complexe.

<sup>18</sup>ATHENA : Advanced Technologies for Interoperability of Heterogeneous Enterprise Networks and their Applications

- **Interopérabilité du processus** : Plusieurs processus s'exécutent en Interactions (série ou en parallèle). Et en réseau entreprise, il est également nécessaire d'étudier comment connecter processus de deux entreprises pour créer un processus commun. L'interopérabilité de processus doit se faire en liant des descriptions des processus différents pour former des processus collaboratifs fait la vérification, la simulation et l'exécution. Différents langages de description de processus sont utilisés pour définir des modèles de processus différents pour des buts différents.
- **Interopérabilité d'affaire (activité)** : Il s'agit du travaillé en harmonisation au niveau d'organisation d'entreprise malgré les différents modes de prise de décision, les méthodes de travail, les législations, la culture de l'entreprise et approches commerciales afin que les entreprises puissent être développées et partagées entre elles sans ambiguïté. Donc il convient à trouver des moyens de correspondances nécessaires pour les solutions de la connexion et d'intégration d'entreprises.

### II.3.6 Les Approches de l'interopérabilité

Nous reprenons les approches citées dans [50,52] :

- **Approche intégrée** : Développer l'interopérabilité grâce à une approche intégrée signifie qu'il existe un format commun pour tous les modèles. Le format commun n'est pas nécessairement un Internationale, mais doit être approuvé par toutes les parties pour élaborer des modèles et construire des systèmes. Exemple de développement l'interopérabilité à l'aide d'une approche intégrée est ebXML<sup>19</sup>.

L'approche intégrée est plus orientée vers l'intégration complète plutôt que l'interopérabilité totale. Cette approche convient à concevoir et mettre en œuvre de nouveaux systèmes plutôt que réingénierie des systèmes existants d'interopérabilité. L'approche de la réingénierie est plus adaptée au développement de l'interopérabilité intra-entreprise plutôt Entreprise. La normalisation au niveau du système (pas au niveau méta) est nécessaire pour développer l'interopérabilité par une approche intégrée.

---

<sup>19</sup>Electronic Business using eXtensible Markup Language.

Les normes matures sont toujours manquantes. La nouvelle norme EN / [ISO<sup>20</sup> 19440, 2004] qui est à l'origine définie pour une approche intégrée n'a pas obtenu une acceptation générale pour construire un modèle d'entreprise, il est plutôt considéré comme référence au niveau de la méta-modélisation pour cartographier modèles l'un à l'autre. L'approche intégrée assure la cohérence du système. Divers composants du système sont conçus et mis en œuvre selon un format commun (standard) afin que l'interopérabilité soit perçue comme étant de qualité (Des composants du système). Interopérabilité entre divers pièces peuvent être obtenues «a priori» sans aucun effort d'interfaçage.

• **Approche unifiée** : L'interopérabilité peut également être établie en utilisant une approche unifiée. Cela signifie qu'il existe un format commun, mais il existe seulement au niveau métadonnées. Il fournit un moyen d'équivalence sémantique pour permettre la cartographie entre modèles et applications. En utilisant le méta modèle, une traduction entre des modèles constitutifs est possible même si rencontrer la perte de certaines sémantiques ou des informations.

La plupart des résultats de recherche développés dans l'interopérabilité ont adopté l'approche unifiée. Par exemple UEML vise à définir un format neutre au niveau de la méta-modélisation pour Cartographie entre les modèles d'entreprise et les outils. Le pas initiative élaborée dans l'ISO TC184 SC4 <sup>21</sup> a également défini un format de données de produit au niveau de la méta-modélisation pour des modèles de données de produits échangeant des informations sur les produits.

L'approche unifiée est particulièrement adaptée à l'interopérabilité pour les entreprises en collaboration ou en réseau. Interopérable avec des partenaires en réseau doit mapper son propre modèle / système au méta-format neutre sans la nécessité de faire des changements sur ses propres Modèle / système. Cette approche présente l'avantage intégrée en raison de la réduction des efforts, du

---

<sup>20</sup> ISO: International Organization for Standardization.

<sup>21</sup> *Industrial data* is Subcommittee 4 of ISO/TC 184, *Automation systems and integration*, which is Technical Committee 184 of the International Organization for Standardization (ISO).

temps et des coûts en œuvre. Il est également adapté à la situation avec différentes entreprises, l'approche unifiée semble être une Solution appropriée.

• **Approche fédérée :** Dans le cas d'une approche fédérée, il n'existe pas de format pour établir l'interopérabilité, les parties doivent accueillir et ajuster «à la volée». Utilisation de la fonction approche fédérée implique qu'aucun partenaire n'impose leurs modèles, les langues et les méthodes de travail. Cela signifie qu'ils doivent partager une ontologie.

L'approche fédérée peut également faire appel à des méta-modèles, la différence entre approche unifiée est que ce méta-modèle n'est pas prédéfini mais établie «dynamiquement» avec la négociation. Elle est particulièrement adaptée aux entreprises virtuelles où diverses sociétés associent leurs ressources et leurs connaissances pour fabriquer un produit avec une durée limitée. Utiliser l'approche fédérée pour Interopérabilité d'entreprises est le plus difficile et peu d'activité a été menée dans ce sens.

Il est à noter qu'un soutien spécifique à l'approche fédérée est vu dans l'entité profils, qui identifient les caractéristiques d'une entité particulière et les propriétés pertinentes pour l'interopérabilité (ISO 15745<sup>22</sup> et ISO 16100<sup>23</sup>). La fédération est considérée l'approche la plus intéressante pour développer l'interopérabilité. Cependant, le choix dépend du contexte d'exigences. Si le besoin d'interopérabilité provient d'une entreprise, l'approche intégrée semble être la plus adaptée. Dans ce cas, Si le besoin d'interopérabilité concerne une collaboration à long terme, l'approche unifiée semble une solution possible. Pour cela, un méta-modèle commun aux modèles de partenaires fournit un moyen d'établir des équivalences permettant la cartographie entre divers modèles.

---

<sup>22</sup> Industrial automation systems and integration — Open systems application integration frameworks.

<sup>23</sup> Industrial automation systems and integration — Manufacturing software capability profiling for Interoperability

### II.3.7 La médiation pour l'interopérabilité des SI

Le concept de médiateur ou d'un système intermédiaire, supporte et gère l'hétérogénéité des SI des partenaires différents et sur tas niveaux : organisationnel, sémantique, syntaxique et technique.

La médiation doit réussir à faire inter opérer les SI hétérogènes en suivant un processus avec des étapes et des règles de fonctionnement bien définies. La reconnaissance des spécificités de chaque système d'information font objet de la naissance de la solution bénéfique de médiateur des SI.

La figure N° 12 présentée ci-dessous montre un modèle d'un médiateur pour l'interopérabilité d'un SI.

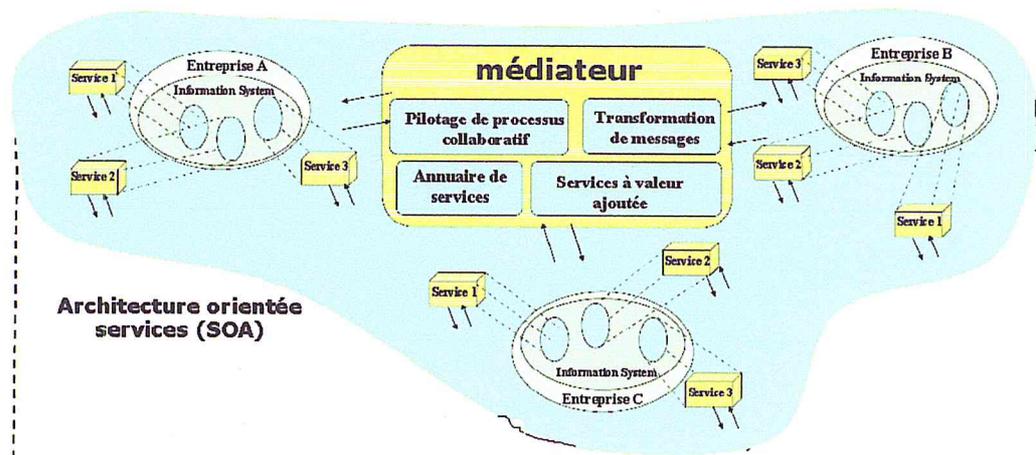


Figure 12 : Un médiateur pour l'interopérabilité des SI [47]

### II.4 Conclusion

Dans ce chapitre, nous avons décrit l'interopérabilité des systèmes d'information en se basant sur une caractérisation des niveaux d'interopérabilité (métier, sémantique et technique) et identification des dimensions liées à l'hétérogénéité dans les systèmes d'information (processus, données et applications).

Le prochain chapitre sera consacré à l'étude de l'interopérabilité dans le domaine du Cloud Computing.

# Chapitre III :

*état de l'art sur les  
solutions de*

*l'interopérabilité du CC*

### III.1 Introduction

Dans ce chapitre, nous exposons une étude détaillée sur l'interopérabilité dans le Cloud Computing. Nous commençons d'abord par l'évolution du Cloud Computing vers l'inter-Cloud et la relation entre les deux concepts. Ensuite, nous présentons les deux dimensions conceptuelles de l'interopérabilité.

Après, nous présentons les solutions et les standards à base de brokers pour la résolution du problème de l'interopérabilité dans le cloud Computing. Enfin, nous finissons ce chapitre par une étude comparative des solutions déjà présentées.

### III.2 L'inter-cloud

L'Inter-cloud a été décrit comme un cloud de cloud [55], Il tient compte de la coordination dynamique et de la distribution de charge parmi un ensemble de Datacenter de cloud.

Les avantages d'un environnement d'Inter-cloud pour des clients de cloud sont nombreux et peuvent être largement récapitulés comme suit :

- **Emplacements géographiques divers :**

Les fournisseurs principaux de Cloud ont établi des Datacenter dans le monde entier. Cependant, il est peu probable que n'importe quel fournisseur pourra établir des Datacenter dans tous les pays et les régions administratives [56]. Seulement en utilisant de multiples Cloud on peut obtenir l'accès à des ressources largement distribuées et fournir des services bien performants aux clients.

- **Meilleure résistance (élasticité) d'application**, durant les dernières années, il y a eu plusieurs cas de coupures de courant de service de nuage, y compris ceux des vendeurs leaders [57-58]. Par exemple, les implications de l'échec de Datacenter d'Amazon étaient très sérieuses pour les clients qui ont compté sur cet emplacement seulement.

- **Évitement de l'enfermement propriétaire du fournisseur (*vendor lock-in*)**, en utilisant de multiples Cloud et étant capable de librement transiter la charge de travail parmi eux, un client de Cloud peut facilement éviter l'enfermement propriétaire. Dans le cas où un fournisseur change une politique ou une tarification (évaluation) qui a un impact négative sur ses clients, ces derniers pourraient facilement migrer ailleurs.

Une idée primordiale de Cloud Computing est qu'un service de nuage devrait livrer la disponibilité constante, l'élasticité et l'adaptabilité pour respecter les exigences des clients [54]. Le fournisseur du Cloud devrait assurer assez de ressources à tout moment. Mais combien sont assez ? Une autre question est la quantité énorme de consommation électrique de Datacenter [53, 60]. Les avantages des fournisseurs de Cloud peuvent être récapitulés comme suit :

- **S'étendre sur demande**, un Cloud devrait maintenir des ressources pour respecter sa charge attendue. Si la charge de travail augmente au-delà de ses limites, les ressources d'autres Cloud peuvent être louées [56].
- **Améliorer le niveau du contrat de services (SLA) aux clients**, sachant que même dans le pire des cas où un Datacenter en panne ou bien l'insuffisance de ressource, la charge de travail entrante peut être déplacée vers un autre Cloud.

### III.2.1 Définition

Le Cloud Computing est un nouveau domaine de recherche et fait toujours face à certaines ambiguïtés terminologiques. L'Inter-Cloud est encore plus récent et les travaux de recherches dans ce domaine utilisent plusieurs termes de façon interchangeable. L'inter-Cloud a été formellement définie comme [61] :

*« Un modèle de Cloud qui a pour but de garantir la qualité de service, les performances et la disponibilité de chaque service, à la demande. Il permet la réaffectation des ressources et le transfert de charge de travail. L'interfonctionnement des systèmes Cloud de différents fournisseurs est basé sur la coordination des exigences des consommateurs pour chaque qualité de service avec chaque SLA fournisseurs et l'utilisation des interfaces standard ».*

### III.2.2 Concepts liés à l'inter-cloud

L'Inter-Cloud peut être mis en œuvre par différents modèles que nous décrivons dans les sections suivantes [63].

- **Inter-cloud de type PEERING** : Ce modèle d'Inter-Cloud est caractérisé par l'interconnexion directe entre deux fournisseurs 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.
- **Inter-cloud de type BROKER** : Ce modèle d'Inter-Cloud est caractérisé par l'interconnexion indirecte entre deux ou plusieurs fournisseurs grâce à une entité appelée Inter-Cloud Service Broker (ISB<sup>24</sup>). Une entité de type ISB offre des fonctions de service d'interfonctionnement entre les fournisseurs

---

<sup>24</sup> Inter-Cloud Service Broker.

interconnectés et fournit également des fonctions de service de courtage pour un ou plusieurs CSP<sup>25</sup> (fournisseur) interconnectés ainsi que pour le client.

- **Inter-cloud de type FÉDÉRATION** : Dans ce modèle d'Inter-Cloud, 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 [64]. Interconnectés et fournit également des fonctions de service de courtage pour un ou plusieurs CSP<sup>26</sup> (fournisseur) interconnectés ainsi que pour le client.

La fédération d'Inter-Cloud permet aussi 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.

- **Multi-Cloud** : ou bien « Independent » Cloud indique l'utilisation de plusieurs Cloud indépendants, par un client ou un service. Contrairement à une fédération, un environnement Multi-Cloud n'implique pas l'interconnexion des volontaires et le partage des infrastructures des fournisseurs. Les clients ou leurs représentants sont directement responsables de la gestion des ressources humaines et la planification d'approvisionnement [65]. Un autre terme utilisé dans la littérature connexe est *Cloud hybride*. Il a été défini comme une composition de deux ou plusieurs infrastructures de Cloud Computing - par exemple, un privé et un Cloud public [1]. Ainsi, un Cloud hybride est un type d'un Multi-Cloud qui relie des Cloud divers en termes de modèles de déploiement, des Clouds hybrides sont utilisés pour l'éclatement des nuages - l'utilisation des ressources de Cloud externe lorsque les ressources locales sont insuffisantes.

---

<sup>25</sup> Cloud Service Provider.

<sup>26</sup> Cloud Service Provider

### III.3 L'inter-Cloud et l'interopérabilité

La communication des différents Clouds est établie à travers un réseau qui peut être fourni en tant que service Cloud de type NaaS<sup>27</sup> et qui permet une solution économique pour l'interopérabilité des Cloud (migration de VM<sup>28</sup>, 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<sup>29</sup> si ces Clouds peuvent inter-opérer les uns avec les autres pour le partage des ressources [68]. De plus, les centres de données (DC<sup>30</sup>) répartis géographiquement dans l'Inter-Cloud offrent de meilleures performances de bout en bout entre les CSU<sup>31</sup> 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.

#### III.3.1 Les dimensions conceptuelles de l'inter-Cloud

Il y a deux dimensions principales de l'interopérabilité du Cloud Computing, verticale et horizontale [69] :

- **La dimension verticale :** concerne l'interopérabilité en termes des potentialités du Cloud Computing d'un seul fournisseur, généralement d'un utilisateur final ou une perspective grand public.
- **La dimension horizontale :** concerne l'interopérabilité pour des fournisseurs de Cloud différents. Cette dimension implique comment facilement une activité qui fait accueillir son logiciel dans le Cloud peut se déplacer vers un fournisseur de Cloud rivalisant qui pourrait offrir des tarifs plus favorables ou le service plus fiable.

---

<sup>27</sup> Network AS a Service

<sup>28</sup> Virtuelle machine

<sup>29</sup> Cloud service provide

<sup>30</sup> Datacenter

<sup>31</sup> Cloud service user

## 1.2 III.4 Solutions de l'interopérabilité

La concurrence croissante entre les fournisseurs du marché de nuages, comme Amazon, Microsoft, Google et Salesforce où chacun défend ses propres standards et formats incompatibles accroît le nombre de solutions propriétaires. Parmi les solutions existantes nous trouvons :

### 1.2.1 III.4.1 Les standards

Il existe quelques efforts de standardisation dans le domaine de CC, nous allons regrouper ces efforts en fonction des trois principaux modèles de services (SaaS, PaaS et IaaS) présentés dans les 3 tableaux suivants :

Couche	Standard	organisme	but
SaaS	Cloud Portability and Interoperability Profiles CPIP	IEEE	un méta standard avec des profils pour les clouds existants pour faciliter la portabilité et la gestion des applications. Le but de ce guide dans le développement des applications en nuage est d'assurer des solutions logicielles uniformes, cohérentes et de haute qualité. [70]

Tableau 2 : Les standards de la couche SaaS

Couche	Standard	organisme	but
PaaS	Cloud Application Management for Platforms CAMP	OASIS <sup>32</sup>	définir une API standard RESTful simple, avec un protocole JSON-BASÉ, avec un cadre d'extensibilité qui permet l'interopérabilité à travers

<sup>32</sup> Open Source API and platform for multiple clouds

			à travers les offres des vendeurs multiples [71]
	Topology and Orchestration Specification for Cloud Applications <b>TOSCA</b>	<b>OASIS</b>	un langage XML pour décrire des applications en nuages et leur régime de fonctionnement et de gestion La portabilité des plans de TOSCA est hérité du langage de workflow et les moteurs utilisés. [72]

Tableau 3 : les standards de la couche PaaS

Couche	Standard	organisme	but
IaaS	Open Virtualization Format <b>OVF</b>	<b>DMTF</b> <sup>33</sup>	standard pour les formats d'images qui vont être gérant le système [73]
	Virtualization Management <b>VMAN</b>	<b>DMTF</b>	la gestion de ressources réelles pour couvrir la gestion de ressources virtuelles. Ils comprennent <b>OVF</b> qui représente la clé du standard de la portabilité. [73]
	Cloud Infrastructure Management Interface <b>CIMI</b>	<b>DMTF</b>	gérer des ressources IaaS. Il implémente une interface REST et définit des APIs rendus aux formats XML et JSON. CIMI vise à fournir un support au standard OVF. [73]

<sup>33</sup> Distributed Management Task Force

	Open Cloud Computing Interface <b>OCCI</b>	<b>OGF</b> <sup>34</sup>	la création de nuage hybride exploitant les environnements de nuages indépendants des fournisseurs et des intergiciels de plateformes. Le principal formalisme utilisé pour définir les modèles cloud est UML, le travail est encore à un stade préliminaire [74]
	Cloud Data Management Interface <b>CDMI</b>	<b>SNIA</b> <sup>35</sup>	définit une API REST pour effectuer des opérations CRUD <sup>36</sup> sur des données à partir d'un environnement de stockage en nuage [75]
	Standard for Inter-cloud Interoperability and Federation <b>SIIF</b>	<b>IEEE</b>	visé à définir une topologie (se concentre sur la description du NIST), un ensemble de fonctionnalités et un modèle de gouvernance pour l'interopérabilité de cloud et la fédération. La norme actuelle, toujours dans le développement.

*Tableau 4 : les standards de la couche IaaS*

<sup>34</sup> Open Grid Forum

<sup>35</sup> Storage Networking Industry Association

<sup>36</sup> Create Read Update Delete

Dans le cas idéal, lorsqu'on développe une application en nuage, la meilleure façon pour lutter contre le coût élevé et la dépendance vis-à-vis d'un seul fournisseur est d'utiliser les standards ouverts. La plupart des efforts de standardisation effectués dans le domaine de l'informatique en nuage sont destinés aux niveaux IaaS et PaaS. En résumé, un tel idéal n'est pas encore adopté dans le jeune marché de Cloud. Le problème est que la plupart des vendeurs rendent le processus d'adoption de standard difficile, frustrant et tellement long que certaines organisations abandonnent.

### III.4.2 Les solutions à base d'intermédiaire « Broker »

**Broker** : est un courtier en Cloud fonctionne généralement selon les principes typiques du processus de courtage. Ils aident les acheteurs en nuage à prendre des décisions en l'aidant à évaluer, à sélectionner un fournisseur de cloud ou une solution basée sur des exigences spécifiques. En règle générale, les courtiers en nuage collaborent et ont des accords mutuels avec différents fournisseurs de cloud. Il négocie également des termes et conditions des prix de la livraison, du déploiement et d'autres détails avec un fournisseur de nuages pour le compte d'un acheteur. Bien que principalement considéré comme un fournisseur de services axés sur les ventes et le marketing, un courtier en nuage peut également fournir des services de consultation, de déploiement, d'intégration et de surveillance de la migration. En principe, le broker ne peut ni modifier le processus, ni implémenter des contrôles de sécurité, vu qu'il ne contrôle pas les offres directement. Cependant, dans certaine configuration de déploiement, il peut modifier certains risques [63].

Dans le cas de l'utilisation d'une offre **IaaS** il peut modifier les vulnérabilités émanant du *moteur d'exécution de processus*. Par contre, modifier les risques de la couche *infrastructure* est du ressort du fournisseur.

- dans le cas de l'utilisation d'une offre **PaaS** il peut contrer les vulnérabilités émanant de la couche *application*. Par contre, celles venant des couches *infrastructure* et du *moteur d'exécution* doivent être gérées par le fournisseur.
- dans le cas de l'utilisation d'une offre **SaaS**, le broker ne peut pas contrer les vulnérabilités, vu qu'il n'a accès à rien.

#### **Les services du Broker :**

Les services fournis par un courtier en nuage, tels qu'ils sont définis par NIST, peuvent être divisés en trois catégories mentionnées au-dessous [1] voir figure 13 :

- **Intermédiation de service** : le courtier ajoute de la valeur à un service donné en fournissant certaines améliorations fonctionnelles utiles aux consommateurs, telles que la gestion de l'identité, la gestion d'accès, rapports de performance, sécurité améliorée, etc.
- **Agrégation de services** : le courtier combine et intègre plusieurs services existants dans le but d'offrir de nouveaux services ou fonctionnalités.

Intégration de données et sécurité de cheminement de ces données durant les transferts via plusieurs fournisseurs sont assurés par le courtier lui-même.

- **Service Arbitrage** : le courtier agrège les services de plusieurs fournisseurs, en sélectionnant l'agence de livraison selon un score attribué précédemment. Cependant, les services à sélectionnés qui sont présentés aux consommateurs ne sont pas fixés (choix flexible et opportuniste).

À cet égard, cela diffère de l'agrégation des services dans laquelle Broker offre également une intégration de services.

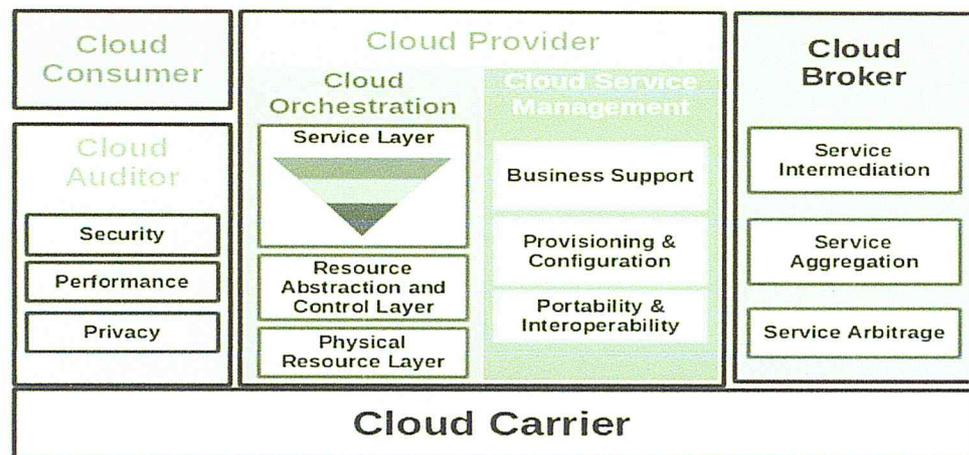


Figure 1.13 : Broker de NIST [1].

## Les solutions de Recherches

**1. Architecture orientée événements** : l'architecture orientée événements (en anglais Event Driven Architecture, ou EDA) est une forme d'architecture de médiation qui est un modèle d'interaction applicative mettant en œuvre des services (composants logiciels) répondant à des sollicitations externes : avec une forte cohérence interne (par l'utilisation d'un format d'échange pivot, le plus souvent XML), et des couplages externes lâches (par l'utilisation d'événements).

### SOLUTION

**Cloud Message Brokering Service – CMBS** : CMBS [76] est un service d'échange et de courtage Cloud à base de messages entre plusieurs domaines, fournisseurs ou entités dans le Cloud. Le modèle d'échange de message qui convient le mieux aux

caractéristiques du Cloud est l'architecture orientée événements (EDA<sup>37</sup>) est présenté dans la figure suivante (figure 14).

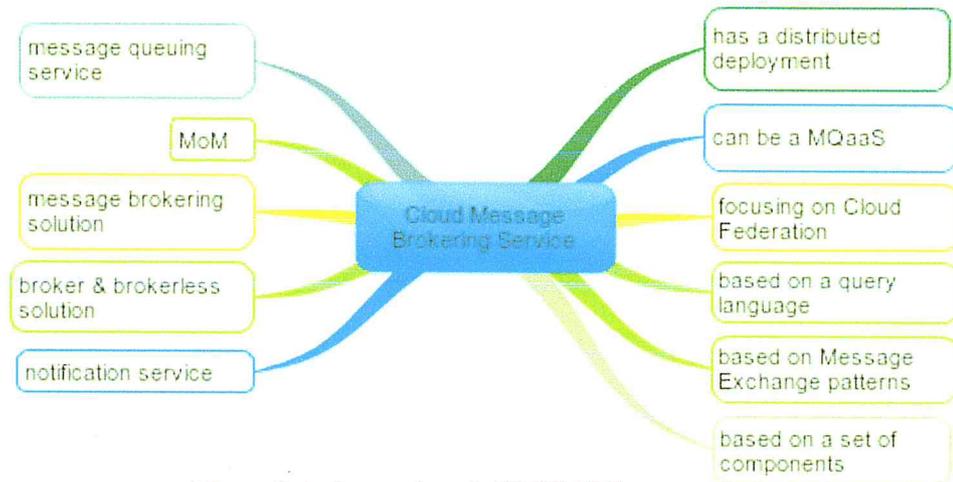


Figure 1.4 : les services de CMBS [76]

**Les composants de CMBS :** Pour mettre en place une architecture de fédération à base d'échange de messages, quatre composants ont été définis dans CMBS. Ces composants, illustrés ci-dessous dans la figure 15 sont :

- **MQ-S (Message Queuing Service) :** est le composant périphérique d'une architecture CMBS. MQ-S est le plus proche des applications et des utilisateurs, car il initie l'envoi des messages. De plus, il est le destinataire final des messages et celui qui fait leurs traitements.
- **MQ-LB (Message Queuing Load Balancer) :** Pour assurer la scalabilité et disponibilité du Framework CMBS, les composants MQ-S peuvent se dupliquer. Le rôle du MQ-LB est de répartir la charge entre les différents MQ-S. La répartition de la charge peut être définie soit selon une approche classique à la Round Robin soit par l'utilisateur.
- **MQ-B (Message Queuing Broker) :** MQ-B est l'élément essentiel dans la mise en place d'une architecture de fédération Cloud à base de CMBS. MQ-B est mis comme front-end pour chaque Provider. Il s'occupe de transférer et recevoir les messages à la liste des Providers concernés à travers leurs composants MQ-B respectifs. De même, il s'occupe de la réception des messages des autres Providers.

<sup>37</sup> Event-Driven Architecture.

- **Links** : Les Links ont le rôle d'établir des liens entre les composants.

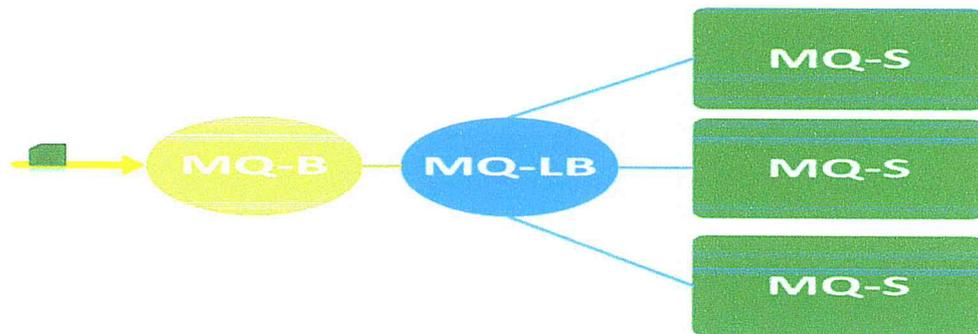


Figure 1.5 : Composant de CMBS [76]

**2. Architecture orientée service** : Le paradigme des Architectures Orientées Services consiste à orchestrer convenablement des services ou des composants pour exécuter une tâche, une activité, un processus SOA est utilisé pour interconnecter différents composants du système d'information, tels que les ERP, CRM, SCM.... La notion d'annuaire est centrale : elle permet aux consommateurs de trouver rapidement et efficacement le service dont ils ont besoin dans un système qui va croître en richesse mais aussi en complexité avec le temps. Un des avantages majeurs de cette architecture réside dans l'utilisation d'interfaces standards qui permettent l'agrégation de composants distribués autonomes.

## SOLUTIONS

**mOSAIC** : mOSAIC [77] propose une API et une plateforme PaaS libre permettant de développer et déployer des applications pour un environnement multi-nuages. L'application déployée avec mOSAIC exige que cette dernière soit constituée de composants avec des dépendances explicites pour communiquer et échanger des données entre eux , voir le figure 16.

En outre, l'architecture de l'application doit être de type SOA et doit utiliser uniquement l'API mOSAIC pour la communication mOSAIC utilise uniquement la communication asynchrone. mOSAIC déploie automatiquement les applications à travers les nuages sans l'intervention directe du développeur d'applications. Pour chaque composant de l'application, le développeur doit préciser les exigences en matière de ressources pour le traitement, le stockage et la communication. On peut spécifier certaines ressources pour l'ensemble de l'application. Ainsi, la seule façon

de contrôler l'application est d'utiliser les accords au niveau service (SLA) prédéfinis sur les indicateurs de performance au niveau composant ou application.

L'approvisionnement des ressources dans la plateforme mOSAIC est fait par un courtier appelé courtier de ressources (Resource Broker). Il est chargé de la médiation entre les fournisseurs de nuage et le client. Le composant "Resource broker" est en outre composé d'une agence de nuage (Cloud Agency) qui est responsable de la découverte et de la négociation avec les fournisseurs de nuage. Le courtier de ressources est aussi composé d'une interface client (Client Interface) qui est responsable de demander des ressources supplémentaires en utilisant le composant exécuteur d'applications Application Executor. Le composant exécuteur d'applications gère le déploiement, l'exécution et la surveillance des applications. La figure N° 16 présente l'architecture de la plateforme mOSAIC.

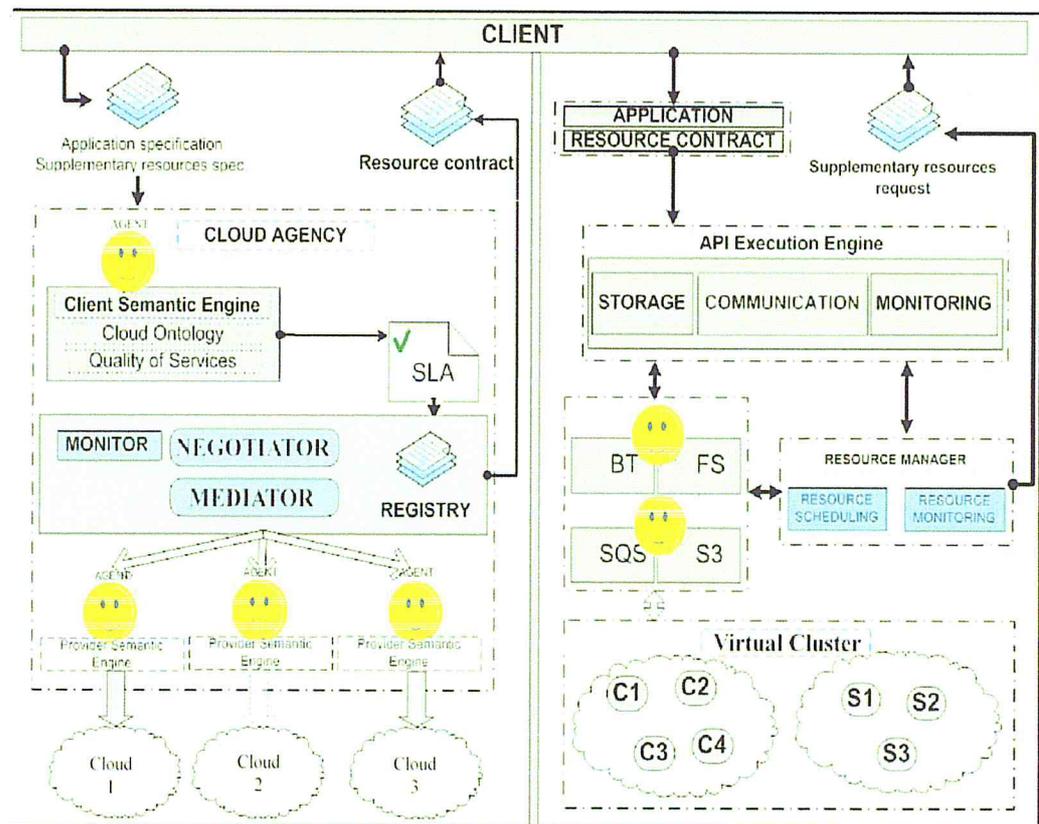


Figure 1 6 : Architecture du projet mOSAIC [77].

**STRATOS** : STRATOS [78] est un projet de recherche supporté par l'Engineering Research Council of Canada (NSERC), Ontario Centre of Excellence (OCE) et AmazonWeb Service. Le but de ce projet est de fournir un courtier de nuage (Cloud Broker) au niveau IaaS permettant d'approvisionner des ressources provenant de plusieurs nuages. Il supporte le déploiement et la gestion des applications à l'exécution. STRATOS permet de déployer des applications, tout en définissant sur ces dernières des exigences qui sont importantes en termes d'indicateurs de performance (Key Performance Indicators) ou KPI. Ces exigences sont transformées en un ensemble de requêtes de demande de ressources qui sont évaluées en fonction de l'offre des fournisseurs de nuage, afin de sélectionner la meilleure offre.

La figure N° 17 présente l'architecture de STRATOS. Dans cette architecture, le document de topologie nommé Topology Descriptor File (TDF) est utilisé pour définir la topologie de l'application qui doit être déployée dans le nuage. Le module de déploiement spécifie tous les détails sur l'environnement de l'application dans ce document. Lors de la réception du document de la topologie, le gestionnaire de nuage (Cloud Manager) contacte le courtier (Broker) pour l'instanciation de la topologie. Le courtier effectue des calculs préliminaires afin de déterminer la façon la plus efficace pour allouer des ressources à travers plusieurs fournisseurs de nuage.

Le gestionnaire de nuage et le courtier utilisent les informations de surveillances pour prendre la décision d'élasticité.

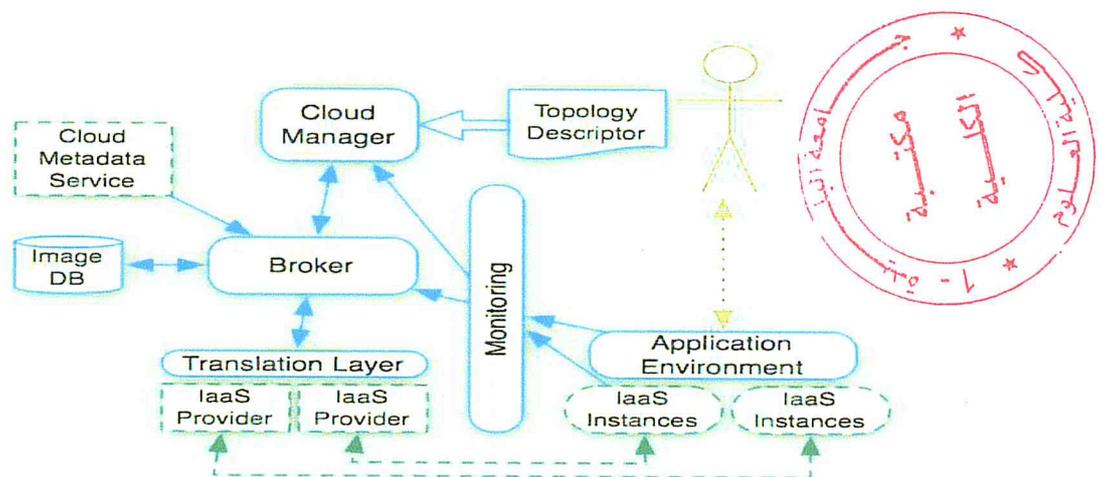


Figure 17 : Architecture du STRATOS [78].

**CompatibleOne** : CompatibleOne [79] est un projet libre initié entre l'industrie et la recherche. Il est un courtier de nuage (Cloud Broker) qui fournit un intergiciel interopérable permettant de décrire et de fédérer des ressources provenant de nuages hétérogènes. Il permet aux développeurs de combiner différents services provenant de plusieurs fournisseurs. Il supporte différents types de ressources qui peuvent être fournies aux niveaux IaaS et PaaS. CompatibleOne est une plateforme d'exécution qui fournit un modèle permettant de décrire les ressources. Ce modèle qui est nommé CompatibleOne Resource Description System (CORDS), permet de décrire une application à base d'objets. La plateforme d'exécution nommée Advanced Capabilities for CORDS (ACCORDS) est un système pour approvisionner et déployer des applications en nuage. La plateforme CompatibleOne est basée sur deux Standards CDMI et OCCI. Elle utilise OCCI comme modèle de communication via REST et CDMI pour l'accès aux données de stockage.

La figure n 18 illustre l'architecture de la plateforme CompatibleOne qui est scindée selon les quatre étapes suivantes :

- **Gestion des besoins utilisateurs.** A cette étape, l'utilisateur spécifie ses besoins en ressources au niveau IaaS ou/et PaaS. Les besoins exprimés sont utilisés pour créer des manifests. Le gestionnaire des accords au niveau service SLAM<sup>38</sup> utilise le modèle OCCI pour gérer la négociation des accords.
- **Validation et plan d'approvisionnement.** A la deuxième étape, le manifest créé est transféré au parseur de l'ACCORDS qui analyse et valide le document. Le parseur construit un plan d'approvisionnement qui correspond aux besoins exprimés. Le parseur interagit aussi avec le publisher de l'ACCORDS, afin de déterminer s'il y a un fournisseur de nuage qui satisfait les exigences requises.
- **Exécution et approvisionnement de plan :** Le courtier fournit les services nécessaires pour sélectionner les fournisseurs de nuage appropriés dans le but de satisfaire les contraintes du plan d'approvisionnement.
- **Fourniture de services de nuage :** A la quatrième étape, le courtier ACCORDS interagit avec les proxies afin d'obtenir les contrats d'approvisionnement. Le modèle CORDS est utilisé pour décrire des ressources

---

<sup>38</sup> Service Level Agreement Manager.

aux niveaux IaaS et PaaS. CORDS est un modèle à base d'objets qui utilise le standard OCCI.

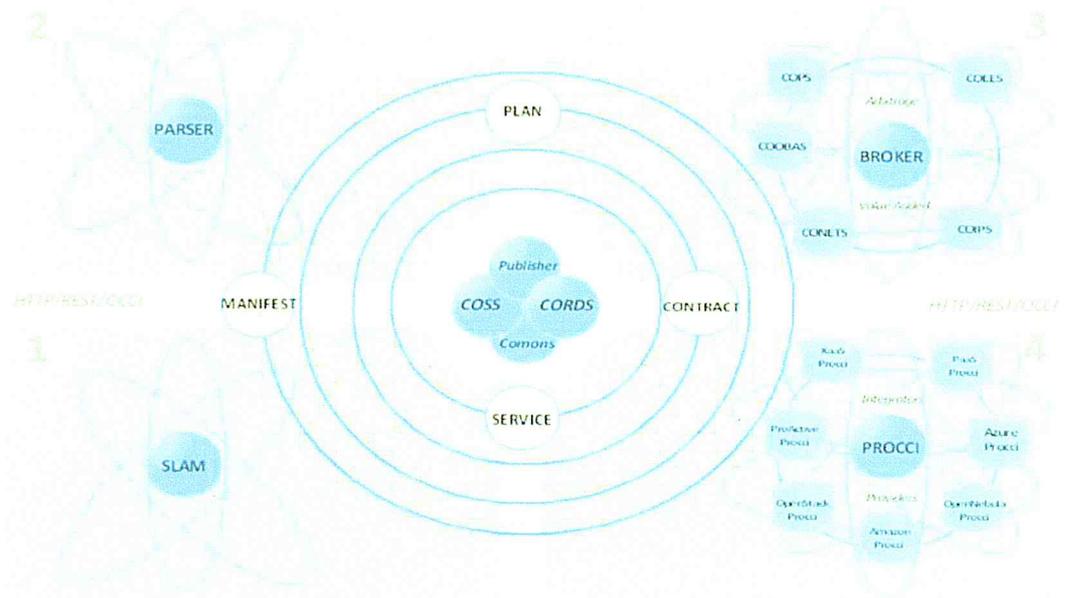


Figure 1.8 : Architecture du CompatibleOne [79].

**3. Architecture à base d'agent :** "On appelle agent une entité physique ou virtuelle, qui est capable d'agir dans un environnement, qui peut communiquer directement avec d'autres agents, qui est mue par un ensemble de tendances (sous la forme d'objectifs individuels ou d'une fonction de satisfaction. Voire de survie, qu'elle cherche à optimiser), qui possède des ressources propres, qui est capable de percevoir (mais de manière limitée) son environnement, qui ne dispose que d'une représentation partielle de cet environnement (et éventuellement aucune), qui possède des compétences et offre des services, qui peut éventuellement se reproduire, dont le comportement tend à satisfaire ses objectifs, en tenant compte des ressources et des compétences dont elle dispose, et en fonction de sa perception, de ses représentations et des communications qu'elle reçoit".

## SOLUTION

**Inter-Cloud Project :** Le terme "Inter-cloud" est utilisé pour se référer à un environnement CC fédéré sous lequel le ravitaillement évolutif peut être exécuté dans le changement de conditions [56]. Ce cadre fédéré ou architecture sont consistés en trois sections majeures ; inclure le client négocie, des services de coordinateur et l'échange de nuage. La responsabilité d'un courtier de nuage est de recevoir la

demande d'un client et accomplir leurs exigences. La tâche des coordinateurs de nuage est d'émettre et distribuer des services à la fédération. En attendant, l'échange de nuage essaye de créer la connexion plus proche entre les prestataires de services et les clients.

Le cloud broker (CB) identifie les fournisseurs de services Cloud adaptés via le Cloud Exchange et il négocie avec Cloud Coordinateurs pour une allocation de ressources en répondant aux besoins QoS des utilisateurs. L'architecture de Cloud Broker est illustrée à la figure N°19 et ses composants sont décrits ci-dessous :

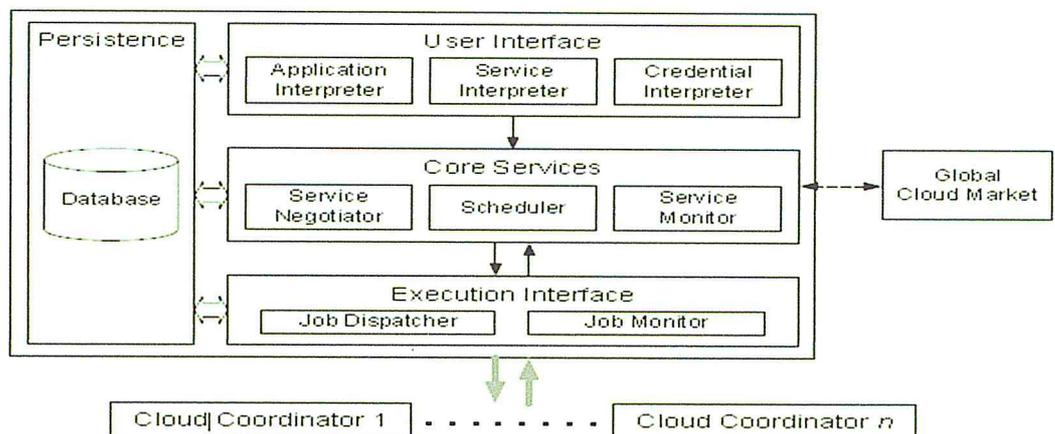


Figure 1°9 : Architecture de projet Inter-Cloud [56]

- **Interface utilisateur :** c'est la liaison d'accès entre une interface d'application d'utilisateur et le courtier.
- **L'interprète d'application** traduit les exigences d'exécution d'une application utilisateur qui comprend ce qui doit être exécuté, la description des entrées de tâches, y compris des fichiers de données distants (si nécessaire), les informations sur les sorties de tâches (si présent) et la QoS souhaitée.
- **L'interprète de service :** comprend les exigences de service nécessaires à l'exécution qui comprennent l'emplacement du service, le type de service et les détails spécifiques tels que les systèmes de soumission de tâches par lots à distance pour les services informatiques.
- **L'interprète d'accréditation :** lit les lettres de créance pour avoir accès aux services nécessaires.

- **Services de base** : c'est la fonctionnalité principale du courtier. Le service négociait avec les services Cloud depuis Cloud Exchange. Le Planificateur détermine les services Cloud les plus appropriés pour l'application utilisateur basée sur ses exigences en matière d'application et de service. Le moniteur de service maintient le statut des services en nuage en vérifiant périodiquement la disponibilité de services Cloud connus et fait découvrir de nouveaux services disponibles. Si le nuage local est incapable de satisfaire les exigences de l'application, Cloud Broker établie une demande de recherche qui encapsule le paramètre QoS de l'utilisateur est soumis à Cloud Exchange en respectant les offres disponibles.
- **Interface d'exécution** : ceci fournit un support d'exécution pour l'application utilisateur. Le Job Dispatcher crée l'agent de courtier nécessaire et demande des fichiers de données (si existe) pour être envoyé avec l'application utilisateur aux ressources Cloud distantes pour exécution. Le Job Monitor observe l'état d'exécution du travail afin que les résultats du travail soient retournés à l'utilisateur lors de l'achèvement du travail.
- **Persistence** : ceci maintient l'état de l'interface utilisateur, les services de base et de l'interface d'exécution dans une base de données. Cela facilite la récupération lorsque le courtier échoue et aide à la comptabilité au niveau de l'utilisateur.

**Financial brokerage model** : Dans [80], les auteurs proposent une mise en œuvre de courtier avec la simulation à base d'agent. Ce courtier traite une forme de contrat dérivé financier livré par des fournisseurs de nuage a appelé une option. Il utilise l'assimilation des contrats d'option pour se décider s'il doit investir en achetant l'accès de ressources, pendant la détermination de périodes, de ce fournisseur ou non. On peut alors par la suite fournir les ressources aux clients qui exigent d'eux. Ce système ne négocie pas vraiment de demandes aux fournisseurs de nuage mais il investit dans des ressources de bail de fournir aux consommateurs la meilleure obéissance d'option à leurs critères. Ce travail est basé sur un WZH négociant le modèle présenté dans [81]. Le modèle de WZH utilise un intermédiaire tiers, appelé le coordinateur, qui utilise une variété

d'actifs(atouts) de nuage pour livrer des ressources aux utilisateurs finaux à un prix réduit, en faisant un bénéfice et aidant des fournisseurs dans la prédiction de ressource. Le coordinateur agit alors comme un courtier

**OPTIMIS** : [82] est une boîte à outils (toolkit) exige que des agents logiciels OPTIMIS soient déployés dans des fournisseurs de nuage et des courtiers d'application. Ces agents communiquent avec l'un l'autre pour mettre en œuvre ravitaillement d'inter-nuage. Un des composants d'agent est le Moteur de Déploiement (*Deployment Engine DE*). C'est le responsable de la découverte et de la négociation avec des nuages appropriés pour accueillir un service particulier (broker). D'abord, un ensemble de cloud appropriés respectant quelques conditions prédéfinies est identifié et le manifeste de service (SLA) est envoyé à chacun d'entre eux. Les fournisseurs de nuage répondent avec un rejet pour accepter le service ou une offre de déploiement. DE choisit la meilleure offre sur la base du quantitatif (par exemple le coût) et qualitatif (par exemple un peu de mesure de confiance (trust)) les aspects de l'offre. DE est aussi responsable d'introduire le déploiement du service dans le nuage choisi en fournissant les images de VM appropriées qui constituent le service nécessaire. Après le déploiement du service, le module d'Optimiseur de Service est responsable de continuellement contrôler les paramètres de service pour des violations SLA. Sur le côté de l'entrée en contact par le nuage, Contrôleur d'Admission (*the Admission Controller*) est responsable de la fabrication d'une offre ou d'un rejet d'une requête. La décision est basée sur la charge de travail actuelle du nuage, les ressources demandées et une évaluation du bénéfice potentiel. L'affectation des ressources pour VMS fourni est faite par le composant Optimiseur de Nuage (*Cloud Optimizer*).

**4. Architecture à base de composants** : est une nouvelle discipline du génie logiciel émerge : le Component-Based Software Engineering (CBSE). Elle est portée notamment par un centre de recherche créé par le gouvernement fédéral américain et exclusivement dédié au génie logiciel : le Software Engineering Institute. Les bénéfices attendus de l'utilisation des composants sont a priori clairs : il s'agit essentiellement de réduire les coûts de production des logiciels, mais aussi d'en réduire les coûts de maintenance et d'en favoriser la réutilisation, De plus, les

composants doivent permettre une meilleure structuration des systèmes facilitant ainsi les aspects de vérification et de validation.

## SOLUTION

**Contrail** : l'architecture de l'Européen Projet Contrail [82] est construite autour d'une entité composite centralisée qui agit comme un point d'entrée seul à une fédération de fournisseurs de nuage. Il est responsable des observations périodiques des états des fournisseurs de nuage, Il fournit aussi un seul signe, pour que les utilisateurs doivent authentifier une seule fois pour travailler avec la fédération entière. Un composant architectural spécial appelé le Directeur de Temps d'exécution de Fédération (*Federation Runtime Manager FRM*) est consacré à dresser la carte des demandes des utilisateurs aux ressources de nuage. Il met en œuvre le coût et l'heuristique d'optimisation de performance et devrait idéalement avoir l'accès à l'emplacement géographique et d'autres méta informations sur chaque fournisseur du cloud, Le courtage d'application est réalisé en spécifiant une application détaillée SLA. C'est la responsabilité du composant *Federation Runtime Manager* à la disposition conformément au SLA et minimiser les coûts. Cependant, la documentation n'est pas spécifique si les utilisateurs sont autorisés à spécifier géographique, législatif et des contraintes d'emplacement de données dans le SLA.

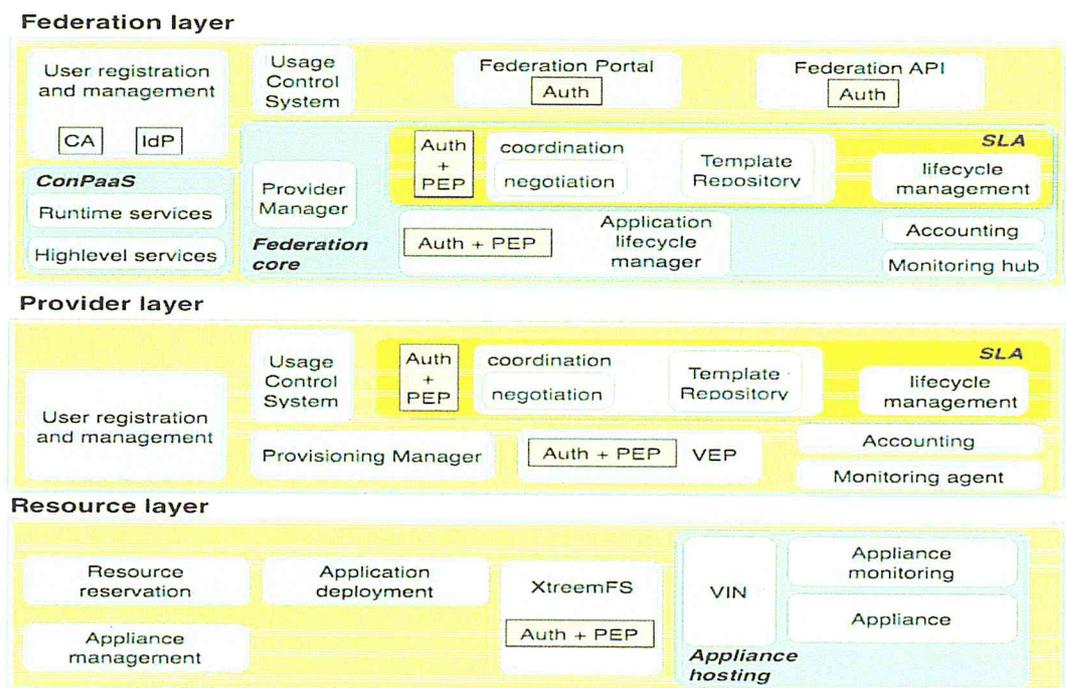


Figure 20 : Architecteur de Contrail [82].

## Les solutions dans le monde commercial :

1. **SpotCloud** Un Cloud broker (utilisé dans SaaS, PaaS, IaaS au niveau public) 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. [83].

2. **Gravitant** : La solution Gravitant permet aux entreprises de planifier, choisir puis gérer divers services en ligne au sein de leur cloud hybride. Les salariés peuvent ainsi utiliser différents services cloud depuis le même écran, qu'ils soient hébergés en interne, sur un cloud privé administré par IBM ou sur un cloud public type AWS et consorts. IBM compte intégrer la technologie Gravitant à son unité Global Technology Services ainsi qu'à ses offres SaaS. [84].

3. **HPE** : Offrir le moyen aux administrateurs de choisir, de prévoir et de consolider des accès à des services cloud en fonction des besoins Disponible à partir de 2016, ce service est proposé comme un outil de service à la demande et permettra d'optimiser les abonnements qu'il s'agisse de performances, de sécurité ou de fournitures de moyens de contrôle. L'outil comportera des systèmes de mesure de la qualité offerte ; en se présentant comme un arbitre au sein de l'entreprise, HP vise surtout les grandes entreprises [85].

4. **Appirio** : Un Cloud broker des CSP public comme Google, Amazon, Salesforce. [86]

5. **Prologue** : Le groupe Prologue a dévoilé officiellement en mai dernier son offre « Use It Cloud Brooker », une solution pour aider les entreprises à déployer sur le cloud en prenant le projet du cahier des charges au déploiement. Les entreprises ont aujourd'hui beaucoup de difficultés à choisir le fournisseur et savoir comment déployer des applications SaaS, des services cloud. Prologue met en place (si l'entreprise le veut) une approche multi-cloud (privé, public, hybride). Use it Cloud Brooker génère les besoins exprimés par l'entreprise en un plan cloud précis (computer, ressources, réseau, etc.), et les aide à choisir les bons outils et applications. Use It Cloud Brooker prend en compte les standards et les spécifications actuels du cloud. [87].

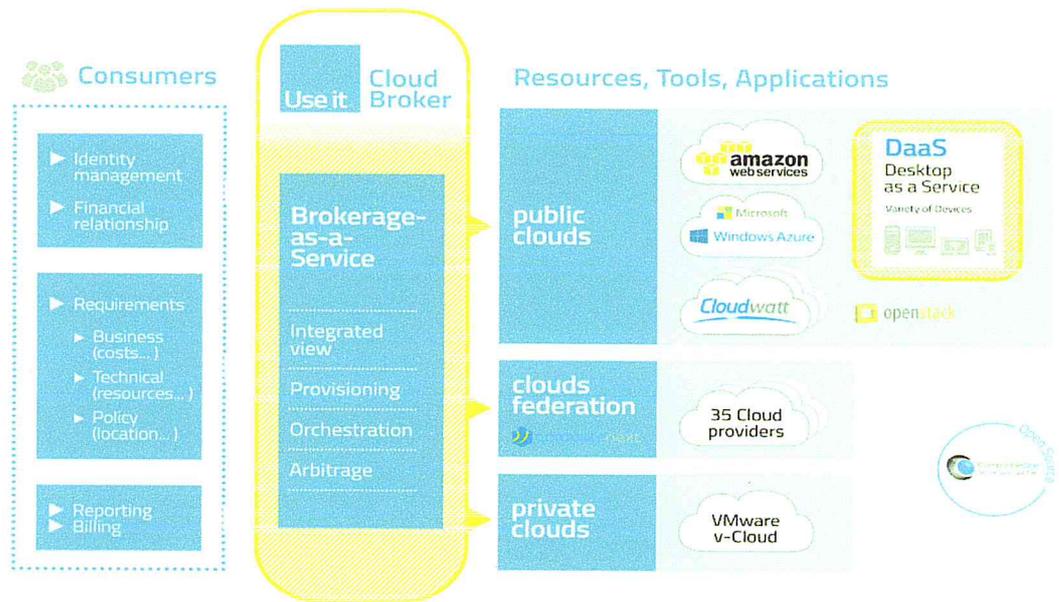


Figure 21 : Architecture du Prologue [87].

Les deux tableaux suivants (5 et 6) représentent une étude comparative des solutions existant dans les deux domaines (commercial et recherche)

Solutions	Couche	Architecture	Principe	organisme	SLA	standard	Architecture inter-cloud	avantages	Inconvénients
<b>mOSAIC [77]</b>	Paas	A base de service	développer et déployer des applications pour un environnement multi-nuages.	European research organisations	oui	/	Multi-cloud	API + Plateforme libre ; Composant indépendant ; déploie automatiquement les applications sans intervention des développeurs.	-Uniquement utilisation des API <b>mOSAIC</b> communication en mode asynchrone.
<b>STRATOS [78]</b>	IaaS	A base de service	Approvisionnement des ressources provenant de plusieurs nuages	Engineering Research Council of Canada, Ontario Centre of Excellence, Amazon.	oui	/	Multi-cloud	-Définir indicateur de performance pour sélectionner meilleur offre de ressources des fournisseurs. -Utilisation des informations de surveillances pour prendre la décision d'élasticité.	Calculs préliminaire pour déterminer efficacement l'allocation des ressources à travers plusieurs fournisseurs des nuages.

Solutions	Couche	Architecture	Principe	organisme	SLA	standard	Architecture inter-cloud	avantages	Inconvénients
CompatibleOne [79]			Fournit un intergiciel interopérable permettant de décrire et de fédérer des ressources provenant de nuages hétérogènes.	European research organisations	oui	CDMI OCCI	Fédération	Projet libre ; Support différent type de ressources de niveau IaaS/PaaS ; Modèle permettant décrire les ressources ; Utilisation deux standard CDMI et OCCI comme modèle de communication pour accès aux données de stockage.	permet de décrire application à base d'objets seulement
<b>RESERVOIR</b>	IaaS	A base de service	Développer une infrastructure orientée service	European research organisations	oui	/	Fédération Peer-to-peer	-le fournisseur de service interagit avec les utilisateurs finaux	Séparation le rôle des fournisseurs service, infrastructure
<b>Inter-Cloud Project [56]</b>	IaaS	A base d'agent	architecture permettre le partage de ressources sur les fournisseurs de cloud	Université de Melbourne	oui	/	Fédération	L'architecture est clairement orientée vers le marché et facilite le courrage de candidature sur le prix.	Travaillé uniquement dans la couche infrastructure

Tableau 5 : Solutions de recherche de l'Inter-Cloud Broker

solution	Couche	Architecture	standard	principe	organisme	SLA	Architecture inter-cloud	Avantages	Inconvénients
SpotCloud [83]	IaaS	SOA	/	Permet CSU de s'inscrire côté achat pour consulter le prix des services demandés CSP de s'inscrire côté vente pour fournir les ressources nécessaires	SpotCloud	/	fédération	/	les CSP qui utilisent d'autres technologies que celles de SpotCloud ne peuvent pas s'inscrire. Fonctionne uniquement avec le cloud public.
Appirio [86]	SaaS, PaaS, IaaS	SOA	/	Un Cloud broker des CSP public comme Google, Amazon, Salesforce...	Appirio Entreprise USA	oui		Appirio un cloud broker au même temps un provider	Fonctionnel uniquement pour le cloud public.
Prologue [87]	IaaS PaaS	SOA	OCCI	génère les besoins exprimés par l'entreprise en un plan cloud précis et les aide à choisir les bons outils et applications	Le groupe prologue de France	oui	Multi-cloud	pouvoir passer d'un cloud à un cloud en toute transparence.	externaliser à outrance et perdre des compétences internes.

Tableau 6 : Solutions Commerciales

Solution	Couche	Architecture	Standard	principe	Organisme	SLA	Architecture inter-cloud	Avantages	Inconvénients
Gravitant [84]	SaaS	SOA	/	Un cloud broker qui permet de planifier, choisir puis gérer divers services en ligne au sein de leur cloud hybride.	IBM	/	Multi-cloud (cloud hybride)	La possibilité d'utiliser différents services cloud depuis le même écran	/
HPE [85]	SaaS	SOA	/	Offrir le moyen aux administrateurs de choisir, de prévoir et de consolider des accès à des services cloud en fonction des besoins	HPE Helion	oui	Fédération	optimiser les abonnements qu'il s'agisse de performances, de sécurité ou de fournisseurs de moyens de contrôle.	Le problème reste qu'avec les nouvelles générations d'informaticiens passionnés par une niche de marché,

Tableau 6 : Solutions Commerciales

### III.4.3 Discussion :

Dans cette section, nous analysons les travaux réalisés sur l'interopérabilité à base de Broker en tenant compte uniquement des critères à savoir : la portabilité, la couche, l'approvisionnement et le Service Level Agreements (SLAs).

#### *Couche :*

La plupart des efforts ont été déployés dans l'Infrastructure-as-a-Service, (RÉSERVOIR, STRATOS, Inter-Cloud Project) et pour la plate-forme en tant que service (mOSAIC, OPTIMIS, 4Caast). La recherche d'interopérabilité dans le logiciel en tant que service est toujours très immature.

#### *Portabilité multi-nuages*

L'approche de la portabilité peut être classée en trois catégories [99] : portabilité (fonctionnelle, données et d'applications).

mOSAIC [100] fournit une solution pour la portabilité d'applications et de données au niveau IaaS et PaaS.

CompatibleOne [101] assure la portabilité des données en utilisant le standard CDMI. Comparées à la solution mOSAIC, les autres solutions assurent uniquement la portabilité des applications. CompatibleOne utilisent des standards pour assurer la portabilité respectivement des applications et des données.

L'utilisation du standard est la meilleure façon d'adresser le problème de la portabilité. Toutefois, l'adoption de standards est quasi inexistante et rendue très difficile voire même contraignante par les fournisseurs de nuages.

#### *Service Level Agreements (SLAs) :*

La majorité des solutions Broker Cloud dispose de SLA (C.-à-d. l'architecture à base de SLA qui se compose de plusieurs règles), ce contrat mutuel entre le fournisseur et les utilisateurs qui prédéfinit les services à fournir.

## *Approvisionnement multi-nuages*

Il existe un certain nombre de travaux sur l'approvisionnement dynamique de ressources matérielles (machines physiques ou machines virtuelles) et les grappes (cluster) de machines [102].

Le travail sur l'approvisionnement dynamique de ressources dans l'informatique en nuage peut être classifié en deux catégories, la granularité des machines virtuelles et la granularité plus fine (cpu, mémoire, disque, réseau).

RESERVOIR [103] supporte deux modes d'approvisionnement : explicite et implicite.

Le mode explicite exige qu'au moment du déploiement, le fournisseur de service spécifie précisément les besoins en capacité de l'application dans des conditions de charge de travail spécifiques. Le mode implicite couvre les besoins en capacité pour les applications de services non calibrés.

CompatibleOne [101] fournit deux modèles différents pour approvisionner des ressources aux niveaux IaaS et PaaS. Ainsi, il offre [101] la possibilité de déployer une application sur un IaaS et PaaS. Toutefois, CompatibleOne n'a pas un modèle uniforme pour approvisionner des ressources à la fois au niveau IaaS et PaaS.

CompatibleOne implante le standard (OCCI) [104]. En outre, les ressources approvisionnées doivent être gérées et consolidées comme un ensemble cohérent et logique afin de faciliter leurs utilisations et permettre une gestion efficace et optimale de celles-ci.

### III.5 Conclusion :

Dans ce chapitre, nous avons présenté deux parties sur l'interopérabilité dans le Cloud Computing.

La première partie se focalise sur les différents concepts de l'interopérabilité Cloud Computing, Inter-Cloud, Multi-Cloud ainsi que les deux dimensions conceptuelles (horizontale, verticale) de l'interopérabilité en CC.

Quant à la deuxième partie, nous avons recensé les solutions de recherches et les solutions commerciales les plus importantes pour le courtage dans le Cloud.

Les solutions d'intermédiaire prennent en considération l'agrégation des offres, la capacité à conseiller judicieusement leurs clients en fonction de leurs moyens et de leurs besoins, de négocier les contrats avec les prestataires et souvent d'ajouter une couche de service supplémentaire (Sauvegarde, personnalisation, intégration, visualisation, sécurité...). Cela s'adresse à des sociétés gérant des données très hétérogènes.

# Chapitre IV :

## *La Solution à base de Broker*

## **IV.1 Introduction**

Le Cloud Computing est un paradigme de calcul distribué à grande échelle qui est un reconditionnement de divers concepts existants de technologies, telles que l'informatique utilitaire, Grid Computing, informatique autonome, la virtualisation et de l'Internet.

Le Cloud est désormais rentré dans une phase d'hybridation mêlant Cloud public et Cloud privé. La question de l'interopérabilité des applications devient un enjeu majeur pour garantir la qualité des services dans le Cloud. Les applications fonctionnant sur des plates-formes différentes n'ont qu'une faible capacité de partage de données. La prise de conscience de ces limites a entraîné un gros effort de standardisation des formats et d'échange de données.

Dans ce chapitre, nous présentons le mécanisme de courtage en Cloud hybride qui permet de faire l'interaction entre les différents fournisseurs. Il possède un catalogue global unifié de service. L'objectif de ce chapitre est de montrer la possibilité d'intégrer les exigences des clients déclarées dans les contrats SLA et l'interopérabilité des fournisseurs de services dans un Cloud hybride.

### **IV.1.1 Conception**

La conception regroupe les activités d'étude qui suivent la spécification, elle englobe : la modélisation, l'architecture ou la conception préliminaire et la conception détaillée.

Nous avons choisi la démarche XP pour le développement et la réalisation de notre solution informatique. Nous débutons par décrire ci-dessous la méthode XP.

#### **La méthode agile : XP-eXtreme Programming**

La méthode eXtreme Programming (XP) [88], se situe dans la lignée des méthodes de développement rapide. Proposé par Kent Beck et Ron Jeffries pendant le projet "C3" (Chrysler Comprehensive Compensation System).

Les méthodes agiles sont basées sur l'expérience et le bon sens. Elles visent avant tout à obtenir un résultat et non à appliquer un formalisme rigide. La méthode est centrée sur le produit, elle n'est qu'un outil au service de la réalisation.

Le développement XP obéit à un cycle en V, l'intérêt réside dans la définition progressive et rigoureuse de l'objectif avec la capacité à s'adapter aux changements.

eXtreme Programming repose sur un certain nombre de pratiques canoniques :

- ▶ Le client pilote lui-même le projet, et ce de très près grâce à des cycles itératifs extrêmement courts (1 ou 2 semaines).
- ▶ L'équipe livre très tôt dans le projet une première version du logiciel, et les livraisons de nouvelles versions s'enchaînent en suite à un rythme soutenu pour obtenir un feedback maximal sur l'avancement des développements.
- ▶ L'équipe s'organise elle-même pour atteindre ses objectifs, en favorisant une collaboration maximale entre ses membres.
- ▶ L'équipe met en place des tests automatiques pour toutes les fonctionnalités qu'elle développe, ce qui garantit au produit un niveau de robustesse très élevé.
- ▶ Les développeurs améliorent sans cesse la structure interne du logiciel pour que les évolutions y restent faciles et rapides.

Pour notre cas, notre promotrice de projet de recherche en master 2 informatique « ingénierie logiciel », madame Mancier joue le rôle de client.

## **IV.2 La solution proposée**

Dans un premier temps, nous allons détailler l'architecture globale de notre solution. Après, nous présenterons l'architecture de courtage.

### **IV.2.1 Architecture globale**

Notre solution est classée dans la famille des architectures distribuées, elle repose sur le développement d'un mécanisme d'interopérabilité à base d'un médiateur dans un Cloud hybride.

Les architectures orientées services et le Cloud hybride sont étudiés en termes d'infrastructure pour développer, déployer, exploiter et maintenir des services de communication et d'échange à l'heure où nous devenons à la fois consommateurs et producteurs de ressources.

#### **IV.2.1.1 Choix logique de conception « Architecture orientée service »**

De nos jours, la plupart des systèmes d'information sont devenus de plus en plus distribués, complexes et coûteux en termes de gestion. Cette évolution du monde informatique a entraîné le développement de nouvelles architectures et technologies qui facilitent la création et le déploiement des applications. En effet, l'apparition de l'approche orientée objet [89] a donné lieu à de nouvelles technologies de distribution des applications [90] (i.e., middleware orienté objet) telles que RMI (Remote Method Invocation) et CORBA (Common Object Request Broker Architecture).

Les principes du modèle orienté objet ont été étendus et améliorés par une autre approche conçue aussi pour la conceptualisation des systèmes distribués, appelée approche orientée composant [91] (i.e., middleware orienté composant). Un composant est une boîte noire regroupant un ensemble d'objets interdépendants (i.e., homogènes en terme de fonctionnalité), et interagissant avec l'extérieur à travers une interface dédiée. Cependant, l'approche orientée composant est freinée par plusieurs obstacles, à savoir l'hétérogénéité technique des composants (i.e., hétérogénéité des langages de programmation : JEE, .NET ; hétérogénéité des protocoles de communication : IIOP, RMI ; etc.) ce qui empêche les différents types d'applications telles que les applications web (PHP, ASP, JSP, etc.), et les applications mobiles (Android, IOS, etc.) d'interagir avec ces composants logiciels. Ainsi, la forte dépendance (moins d'autonomie) qui existe entre les composants au-sein d'une application empêche leur évolution et leur maintenabilité. Ces lacunes font naître le besoin d'une architecture plus flexible et plus agile où les applications sont totalement indépendantes autonomes et facilement intégrables. D'où l'apparition de l'architecture orientée service (en anglais, Service-Oriented Architecture (SOA)).

#### IV.2.1.2 choix technique de conception « Services Web »

Les services Web sont certainement la technologie la plus populaire dans le monde industriel et académique pour migrer vers l'architecture à services. Les principaux mécanismes de fonctionnalités requises pour la réalisation d'une SOA sont assurés par une pile de standard de protocoles construite autour de la définition des services web. En fait, il existe deux types de services web, les services web SOAP<sup>39</sup>, et les services web REST. Ces deux types de services web proposent différents standards et langages afin de mettre en place les principes de la SOA. Dans notre cas on va utiliser le SW-SOAP. Ci-dessous dans la figure N° 22 nous décrivons les différents rôles et interactions de concept web service.

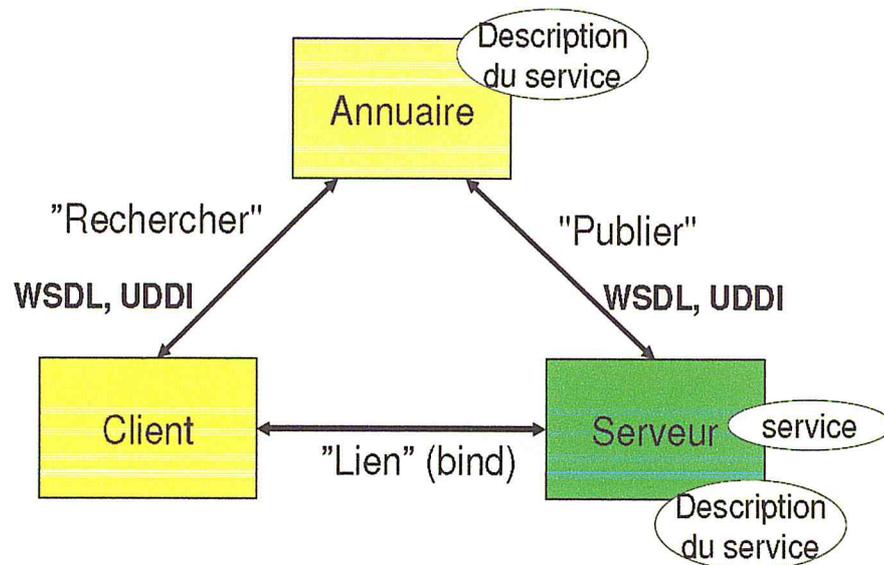


Figure 21 : les rôles et les interactions au sein d'une architecture orientée service, (web service). [98]

#### Service web SOAP

Les services web SOAP sont la première technologie qui a été proposée dans le but d'implémenter l'architecture SOA. Cette réalisation de la SOA est assurée par une panoplie de standards qui sont tous basés sur le modèle XML. Ce dernier est un format universel compréhensible par toutes les plateformes de

<sup>39</sup> Simple Object Access Protocol

qui respecte les principes de la SOA, tels que : le couplage faible et l'interopérabilité. Le W3C définit les services web SOAP comme étant [92] :

*« Un service web est une entité logicielle capable d'être interopérable avec d'autres systèmes distribués. Il a une interface décrite dans une description facilement exploitable, appelée WSDL. Les autres systèmes interagissent avec le service web en se basant sur des messages SOAP, généralement transmis via le protocole HTTP avec une sérialisation XML ».*

Autour de cette définition, les services web SOAP reposent sur un ensemble de standards décrits comme suit :

- **SOAP** : Simple Object Access Protocol (SOAP) est un protocole dont la syntaxe est basée sur XML. Il permet des échanges standardisés de données, tout en résolvant les conflits syntaxiques et techniques. Il permet d'invoquer des services indépendamment de la plate-forme d'exécution. En outre, il s'appuie sur d'autres protocoles de la couche d'application, notamment HTTP, et cela pour la négociation et la transmission des messages.
- **WSDL** : Web Services Description Language, est un langage de description à base d'XML, utilisé pour décrire les fonctionnalités offertes par un service web (i.e. ; l'interface du service). Le fichier WSDL d'un service définit la façon dont le service peut être appelé, avec quels paramètres, et la structure de données qu'il retourne.
- **UDDI** : Universal Description, Discovery and Integration (UDDI) est une plateforme indépendante, à base d'XML, qui assure le regroupement, le stockage et la diffusion des descriptions de services Web.

## **L'architecture proposée :**

Notre architecture de courtage est illustrée dans la figure N°23, par la suite nous donnons sa description.

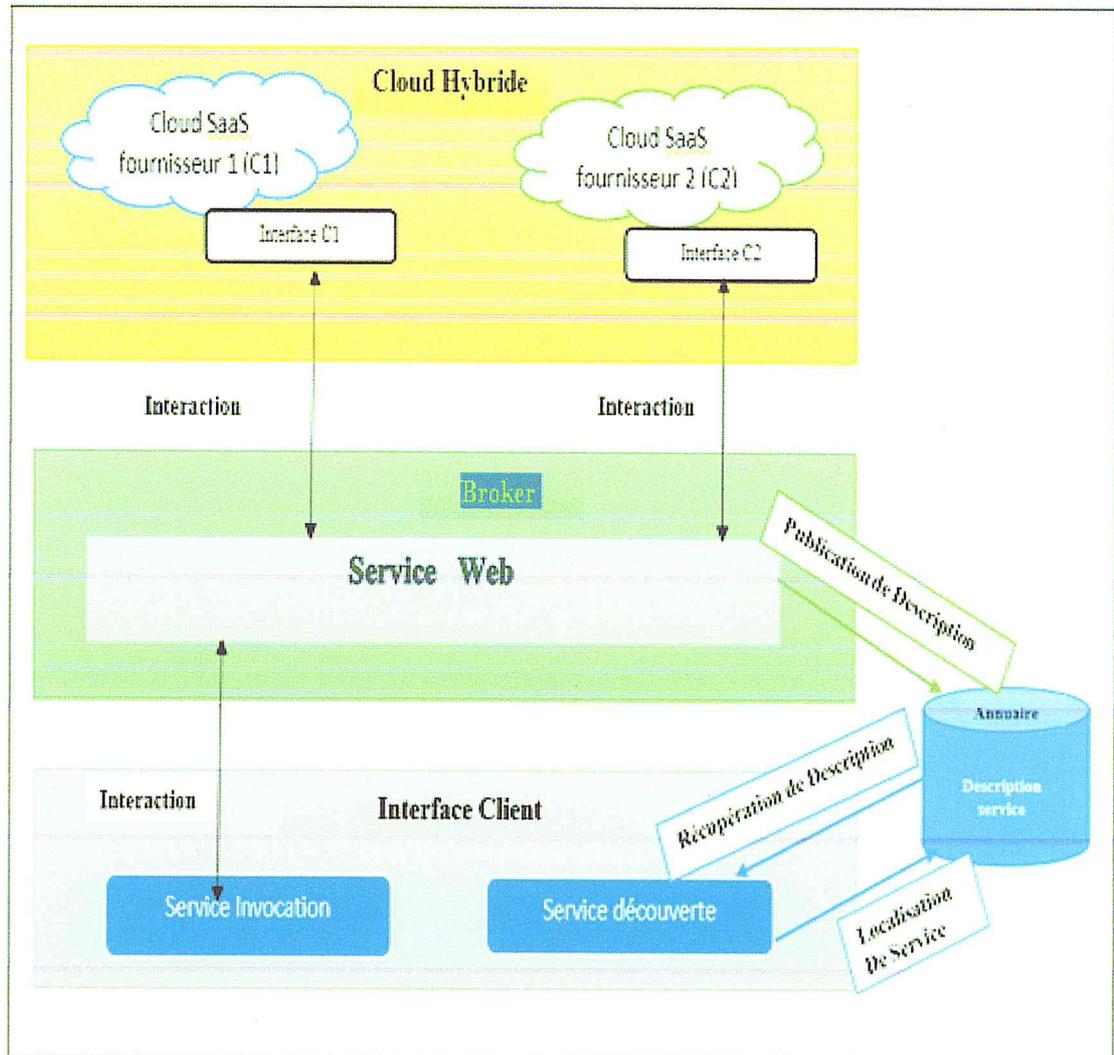


Figure N°23 : Architecture de courtage.

## Description de l'architecture proposée

L'architecture proposée est basée sur l'échange et la communication dans le Cloud hybride via le broker de nature web service. Elle repose sur les principes suivants :

1. **Modéliser une architecture de l'interopérabilité dans le Cloud hybride à base d'un Broker :** Le but est définir une architecture qui rapproche le Cloud public et le Cloud privé. Cette architecture comporte la conception d'un Broker à travers un service web qui résout les problèmes d'interopérabilité des Clouds.

2. **Définir une interface broker** : Le but est de spécifier une interface qui permet d'étendre les requêtes des clients avec tout type de service de Cloud hybride. Cette interface est l'un des composants qui assurent la communication avec le médiateur en direct et les Clouds indirectement.
3. **Définir le composant médiateur (Web Service) à base des messages** : Le but est de définir le broker (web Service) pour assurer l'échange, la communication et la coopération dans le Cloud hybride d'un côté et de l'autre côté assurer l'échange entre le broker et le client.
4. **Définir les interactions entre les différents acteurs de l'architecture de courtage** : Le but est de définir les interactions entre le client et le broker et entre le broker et le Cloud hybride. L'échange s'organise selon un algorithme d'ordonnancement.

#### **IV.2.2 Description du composant médiateur**

Afin de remédier aux problématiques d'interopérabilité des Cloud, nous avons proposé une architecture de Broker (web service). Ce broker joue le rôle d'intermédiaire entre les fournisseurs (Cloud) et les consommateurs du Cloud (clients).

Cette architecture est illustrée ci-dessous par la figure N°24 où nous schématisons ses différents modules.

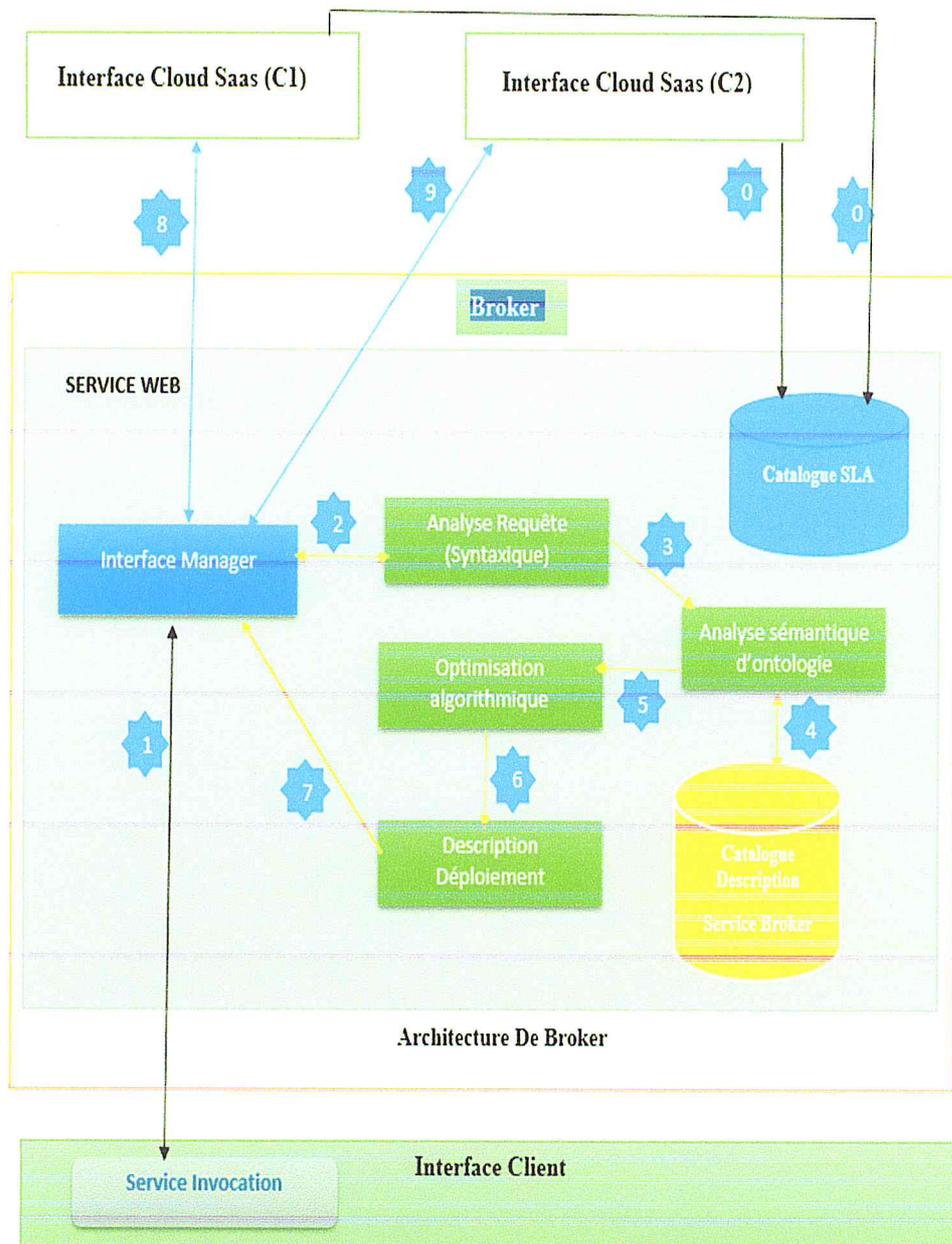


Figure N°24 : Architecture de broker.

**Description des composants de l'architecture proposée :**

- **Catalogue Description Service Broker :** Les services des fournisseurs sont unifiés et stockés au niveau d'un module " **Description Service Broker** ". Ces services sont prédéfinis par les fournisseurs du broker, en adéquation avec leurs publications exprimées au niveau de l'annuaire.
- **Catalogue SLA :** décrit les structures des contrats SLA établis entre le client, le broker et les fournisseurs (Cloud hybride). Il possède les règles utilisées

pour détecter les violations dans le contrat SLA et pour déterminer aussi les pénalités à imposer au fournisseur de service en cas de la violation. Au final, le catalogue assure le respect des contraintes signalées et dégagées préalablement entre les différents acteurs de notre solution (utilisateur, broker, Cloud hybride). La figure N° 25 ci-dessous, présente l'ontologie générale de SLA.

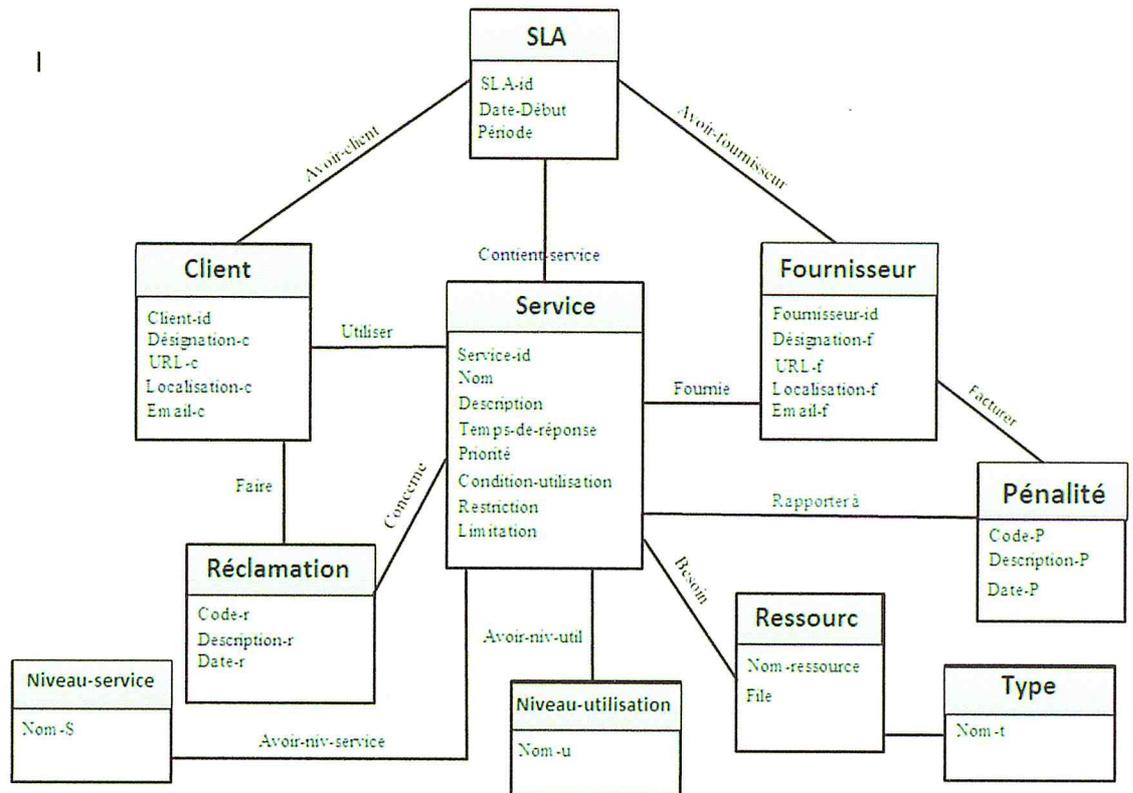


Figure N°25 : Ontologie de SLA

Afin d'être compris par tous les acteurs et par tous les composants de notre architecture, le contrat SLA est standardisé en format XML.

La figure N°26, représente le fichier SLA.xml

```
<? Xml version="1.0" encoding="UTF-8"?>
<!--
DO NOT ALTER OR REMOVE COPYRIGHT NOTICES OR THIS
HEADER.
Copyright 2009-2017 Oracle and/or its affiliates. All rights reserved.
-->

<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
targetNamespace="http://xml.netbeans.org/schema/newXmlSchema"
```

```

<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
targetNamespace="http://xml.netbeans.org/schema/newXmlSchema"
xmlns:tns="http://xml.netbeans.org/schema/newXmlSchema"
elementFormDefault="qualified">
  <xsd:complexType name="SLA">
    <xsd:sequence>
      <xsd:element name="Service">
        <xsd:complexType>
          <xsd:sequence>
            <xsd:sequence/>
            <xsd:element name="reclamation">
              <xsd:complexType>
                <xsd:sequence/>
                <xsd:attribute name="nom" type="xsd:string"/>
                <xsd:attribute
type="xsd:string"/>
                    name="description"
                    <xsd:attribute name="Date-r" type="xsd:string"/>
              </xsd:complexType>
            </xsd:element>
            <xsd:element name="resource">
              <xsd:complexType>
                <xsd:sequence>
                  <xsd:element name="entree">
                    <xsd:complexType>
                      <xsd:sequence/>
                      <xsd:attribute name="type-e"
type="xsd:anyType"/>
                    </xsd:complexType>
                  </xsd:element>
                  <xsd:element name="sortie">
                    <xsd:complexType>
                      <xsd:sequence/>
                      <xsd:attribute name="type-s"
type="xsd:anyType"/>
                    </xsd:complexType>
                  </xsd:element>
                </xsd:sequence>
                <xsd:attribute name="nom" type="xsd:string"/>
              </xsd:complexType>
            </xsd:element>
            <xsd:element name="penalite">
              <xsd:complexType>
                <xsd:sequence/>
                <xsd:attribute name="nom" type="xsd:string"/>
                <xsd:attribute
type="xsd:string"/>
                    name="Destription"
                    <xsd:attribute name="Date-p" type="xsd:string"/>
              </xsd:complexType>
            </xsd:element>
          </xsd:sequence>
        </xsd:complexType>
      </xsd:element>
    </xsd:sequence>
  </xsd:complexType>

```

```

        </xsd:complexType>
        </xsd:element>
        </xsd:sequence>
        <xsd:attribute name="nom" type="xsd:string"/>
        <xsd:attribute name="Description" type="xsd:string"/>
        <xsd:attribute
            name="Temps-de-reponse"
type="xsd:string"/>
        <xsd:attribute name="restriction" type="xsd:string"/>
        <xsd:attribute
            name="conditionUtilisation"
type="xsd:string"/>
        <xsd:attribute name="limitation" type="xsd:string"/>
        <xsd:attribute name="niveauUtilisation" type="xsd:int"/>
        <xsd:attribute name="niveauService" type="xsd:int"/>
    </xsd:complexType>
</xsd:element>
<xsd:element name="Client">
    <xsd:complexType>
        <xsd:sequence/>
        <xsd:attribute name="telephone" type="xsd:string"/>
        <xsd:attribute name="URL" type="xsd:string"/>
        <xsd:attribute name="designation" type="xsd:string"/>
        <xsd:attribute name="mail" type="xsd:string"/>
        <xsd:attribute name="localisation" type="xsd:string"/>
    </xsd:complexType>
</xsd:element>
<xsd:element name="fournisseur">
    <xsd:complexType>
        <xsd:sequence></xsd:sequence>
        <xsd:attribute name="localisation" type="xsd:string"/>
        <xsd:attribute name="mail" type="xsd:string"/>
        <xsd:attribute name="telephone" type="xsd:string"/>
        <xsd:attribute name="Designation" type="xsd:string"/>
        <xsd:attribute name="URL" type="xsd:string"/>
    </xsd:complexType>
</xsd:element>
</xsd:sequence>
<xsd:attribute name="Name-SLA" type="xsd:string"/>
<xsd:attribute name="Periode" type="xsd:duration"/>
<xsd:attribute name="dateDebut" type="xsd:date"/>
</xsd:complexType>
</xsd:schema>

```

Figure 26 : description du web service en WSDL

A partir de notre fichier SLA.xml, nous avons établi le tableau N°7, ce dernier englobe toutes les relations engendrées par les différentes interactions des acteurs de

notre architecture. Le tableau n°7 décrit ci-dessous, présente les diverses relations de l'ontologie de SLA.

<b>Object Propreties</b>	<b>Domain</b>	<b>Rang</b>
Avoir-Fournisseur	SLA	Fournisseur
Avoir-client	SLA	Client
Contient-service	SLA	Service
Faire-une-req	Client	Requête
fournie	Fournisseur	Offre
utiliser	Client	Service
Contrat	Client, Fournisseur	SLA
Rapporter- a	Pénalité	Service
Concerne-req	Requête	Fournisseur
Facturer	fournisseur	pénalité
Besoin	Service	Ressource
Avoir-type	Ressource	Type
réclamer	Client	réclamation
Concerne-ser	Réclamation	Service

*Tableau 7 : Les relations de l'ontologie du SLA*

- **Interface Manager** : c'est le module principal du broker, il assure la manipulation des différentes opérations. Il englobe la demande « Request » et la réponse. Cette unique interface permet de traiter les différentes requêtes des utilisateurs.
- **Analyse Requête (Syntaxique)** : ce module contient des règles utilisées pour assurer la vérification de la cohérence de la requête de l'utilisateur. Dans le cas d'incohérence, le broker alerte l'utilisateur pour qu'il reformule sa demande.
- **Analyse sémantique d'ontologie** : après une analyse ontologique profonde, on construit ce module qui va vérifier la cohérence sémantique de la requête de l'utilisateur.

La figure N° 27 présentée ci-dessous illustre les divers concepts de la taxonomie de Cloud Computing. Elle représente un méta-modèle pour notre solution.

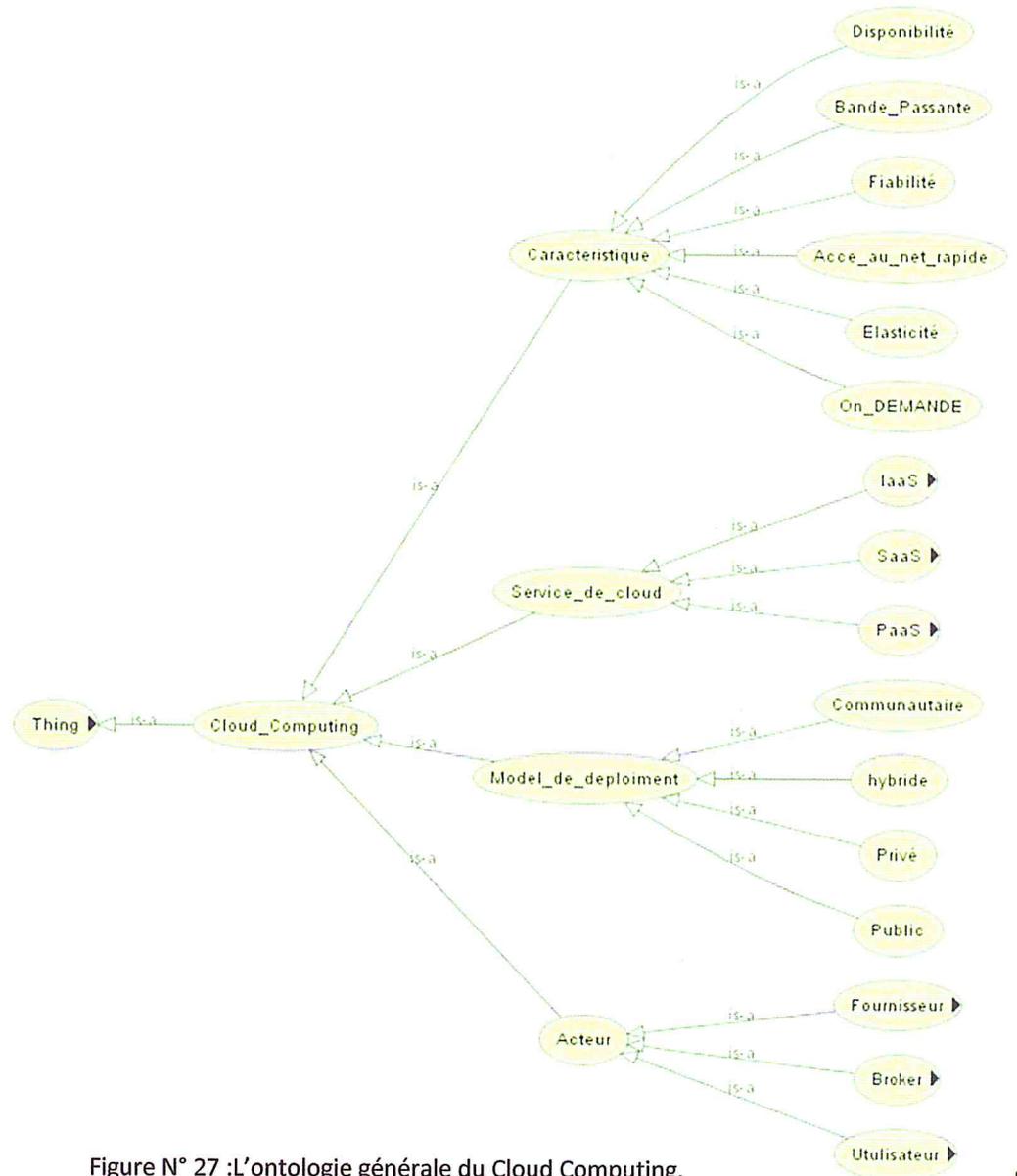


Figure N° 27 :L'ontologie générale du Cloud Computing.

- **Optimisation algorithmique** : le broker applique un algorithme de système distribué, il y'aura un partitionnement et un ordonnancement des tâches. Les tâches sont classées selon un paramètre donné (ex : période, priorité, charge Cloud, ...). On traite les tâches séquentiellement, selon le classement.

Nous présentons ci-dessous en figure N°27 la structure d'une tâche de requête après le partitionnement en n tâches unitaire,

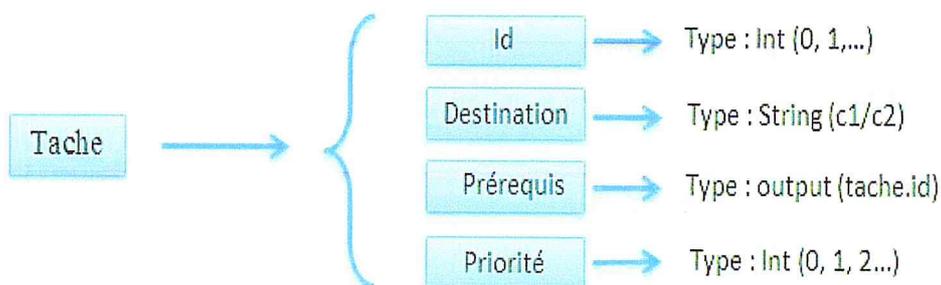


Figure N° 28 : La structure d'une Tâche

- **Description Déploiement :** La phase de déploiement est la période pendant laquelle le broker déploie le message soap et s'assure que la requête est bien stable. Les points d'interaction (les messages) sont décrits dans le fichier WSDL pour qu'ils deviennent accessibles sous forme de web-services. Les processus peuvent même être déployés à distance et de façon automatique. Un point intéressant avec ces processus décomposés est que chaque Cloud ne "voit" que ses sous processus, et n'a donc pas de vue complète du processus global.

### IV.2.3 Description des interactions

Les interactions des acteurs de Cloud hybride (fournisseur SaaS-1 et SaaS-2) et le web service broker et l'échange des entrées/sorties sont présentés dans la figure N°28 ci-dessous. L'interopérabilité effectuée dans le Cloud hybride est détaillé dans ce schéma d'automate.

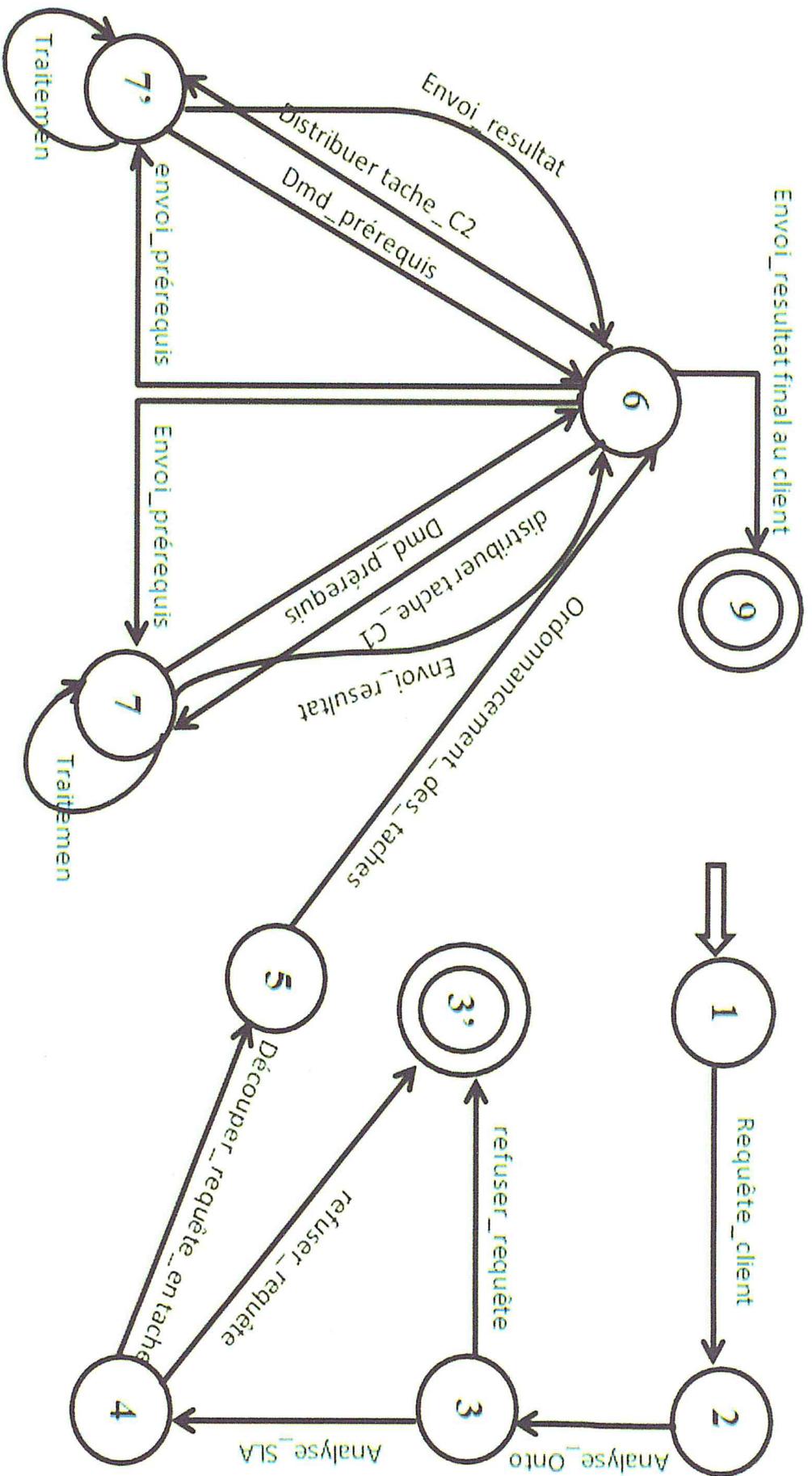


Figure 29 : Automate des interactions dans le Cloud hybride

La figure N°29, résume le déroulement du processus d'interopérabilité effectuée dans un Cloud hybride via un broker de type service web. Les opérations sont numérotées selon leurs ordres de déclenchement. La figure N°30, présentée ci-dessous, illustre le diagramme de classes de notre solution.

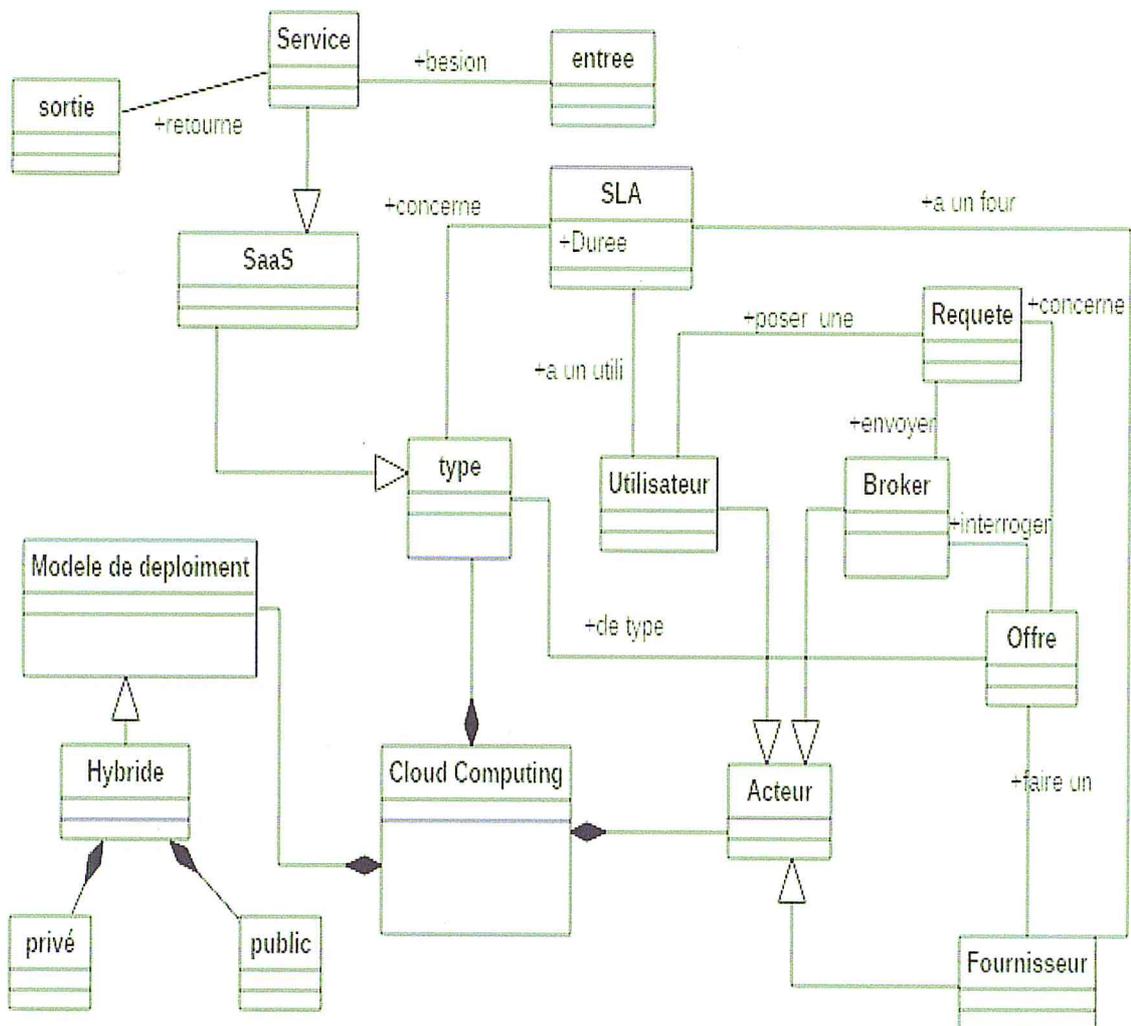


Figure N°30 : Diagramme de classe de la solution proposée.

Le diagramme de classe décrit toutes les classes nécessaires pour réussir les opérations d'échanges et de communications.

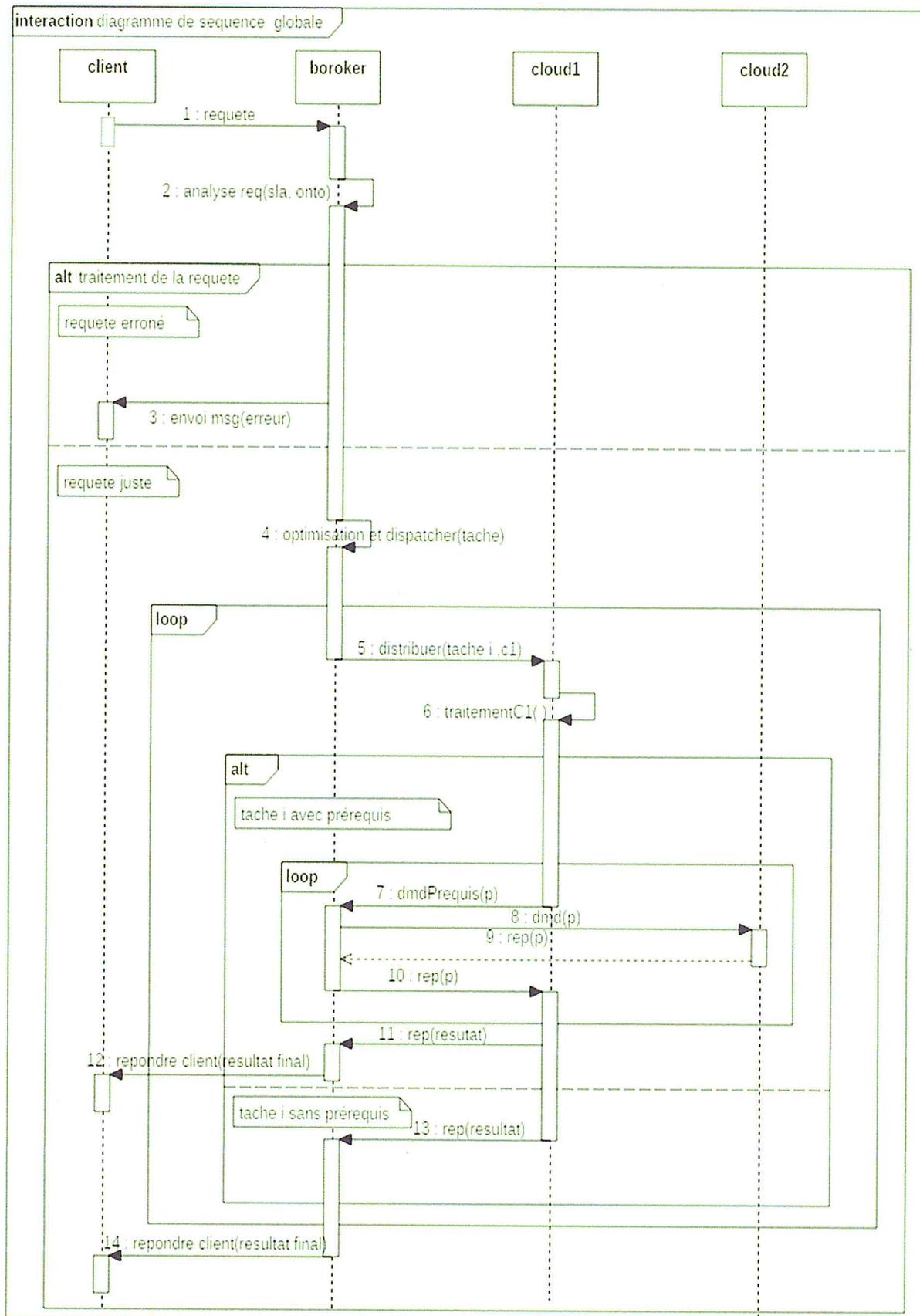


Figure N°31 : Diagramme de séquence de l'interopérabilité de cloud hybride à base d'un médiateur (web service).

La figure ci-dessus présente le diagramme de séquence qui montre l'interopérabilité dans un Cloud hybride (cloud1, cloud2), et toutes les interactions possibles. Le broker joue le rôle d'intermédiaire entre eux. Notre exemple fait apparaître la distribution de tâche1 au cloud1. Le broker pourra attribuer la 1ère tâche au Cloud 2 et donc pour séquences entre 5 et 12, les opérations de cloud1 se dérouleront au niveau de cloud2 et vice versa.

Il pourra avoir un autre scénario de celui d'attribué des tâches au cloud1 et cloud2 de manière parallèle, donc ce cas ce même diagramme se répète au niveau de chacun.

#### IV.2.4 Exemples de scénario

**Scénario 1 :** la figure N°31 illustre le déroulement d'un processus d'interopérabilité mené par un service web broker au sein d'un Cloud hybride :

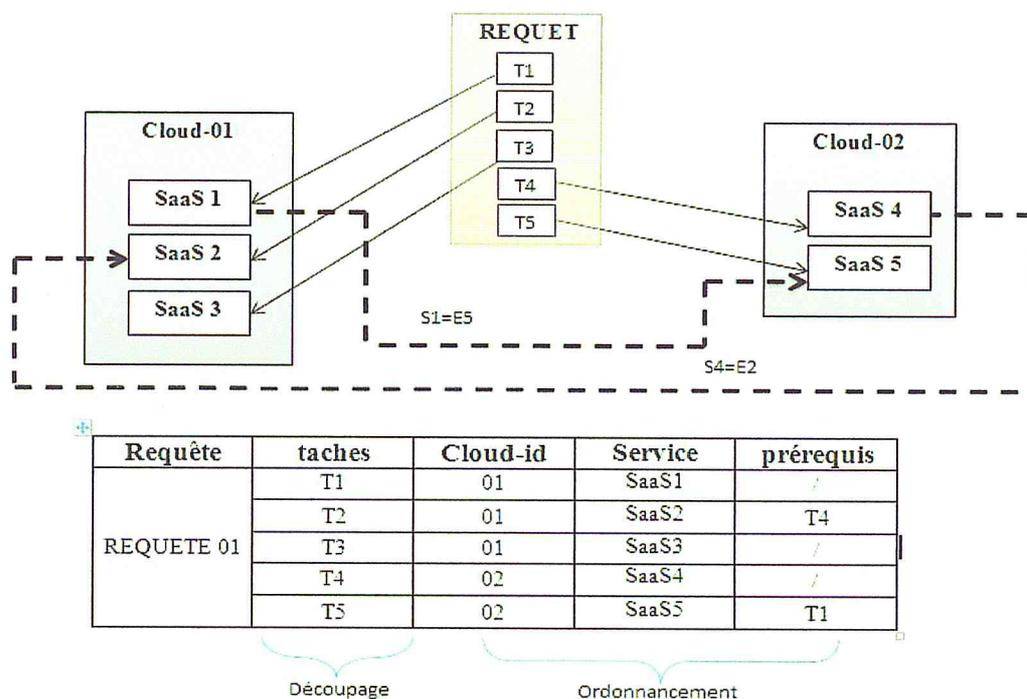


Figure N°32 : Le scénario d'interopérabilité de cloud hybride via un service web broker.

L'exemple de scénario décrit le déroulement de l'interopérabilité dans un Cloud hybride pour le service SaaS. La requête du client subira plusieurs étapes afin de lui rendre sa réponse. Après la validation de la requête, la prochaine étape sera son découpage en plusieurs tâches. Ces dernières seront ordonnées et optimisées selon

leurs priorités et leurs prérequis. Ensuite viendra l'étape de distribution des tâches vers leur Cloud précis. L'échange entre ces Clouds est assuré par le broker web service.

**Scénario 2 :** Nous considérons un convertisseur de devises, il garantit la conversion de plus de 150 monnaies. Nous devrions préalablement convertir notre somme en dollar. Cette 1<sup>ère</sup> opération sera effectuée par Cloud SaaS\_1. La somme obtenue sera déployé ensuite pour le Cloud SaaS\_2 comme 2ème opération de processus de change de monnaie. Ce dernier fait convertir une somme de monnaie en dollar à une monnaie au choix.

Exemple : je possède 1000 DA et j'ai besoin de la convertir en euro. Tout d'abord, le broker envoie un message (soap-requete) au Cloud SaaS\_1, il lui demande de fournir un résultat sous forme de message (soap-résultat) de la somme monnaie convertie en dollar **1000 DA = 9.06 \$**.

La sortie de (Cloud SaaS\_1) sera envoyée au Cloud Saas\_2 comme une entrée pour qu'il puisse changer la somme en euro **9.06\$ = 7.59796214€**.

Au final le broker envoie le résultat obtenu en euro au client.

#### **IV.1.3 discussion :**

Notre solution permet d'atteindre les objectifs suivants :

- Accès aux services dans le Cloud hybride d'une manière automatique via l'interface broker web service qui est décrite d'une manière interprétable par les machines ;
- renforcer l'interopérabilité entre les services du Cloud hybride dans sa couche SaaS d'une manière simple en s'appuyant sur des langages et protocoles indépendants des plates-formes d'implantation et surtout libérer les acteurs et entreprises des fournisseurs de services ;
- Réaliser des interactions faiblement couplées ;
- Ce travail met l'accent sur un mécanisme de fédération entre les Clouds ;
- Ce mécanisme assure une interopérabilité via un système d'échange de messages à base d'un broker web service.

### **IV.3 Conclusion :**

Dans ce chapitre, nous avons donné la description de notre architecture pour la résolution du problème de l'interopérabilité dans un Cloud hybride. La solution proposée repose sur un broker web service, ce dernier assure les interactions entre les divers services du Cloud hybride pour sa couche SaaS et dans l'aspect de fédération horizontale des fournisseurs.

Nous avons décrit l'architecture générale de courtage et l'architecture du broker qui est à base de service web.

Pour illustrer notre solution, nous avons donné deux scénarios pour montrer le déroulement.

Dans le prochain chapitre, nous passerons à l'implémentation et la réalisation de cette solution.

# Chapitre V :

## *Validation*

## **V.1 Introduction**

Ce chapitre décrit la phase d'implémentation de notre solution. Nous allons présenter, dans un premier temps, l'environnement de travail avec les différents langages de programmations. Ainsi que l'environnement matériel et logiciel utilisé dans notre architecture, pour passer par la suite aux étapes de l'implémentation puis la simulation et l'interprétation des résultats.

## **V.2 Langage et environnement de développement**

### **V.2.1 Langages de programmation java**

**Java** est un langage de programmation informatique orienté objet et un environnement d'exécution informatique portable créé par James Gosling et Patrick Naughton employés de Sun Microsystems avec le soutien de Bill Joy (cofondateur de Sun Microsystems en 1982), présenté officiellement le 23 mai 1995 au SunWorld Java est à la fois un langage de programmation et un environnement d'exécution. Le langage Java a la particularité principale que les logiciels écrits avec ce dernier sont très facilement portables sur plusieurs systèmes d'exploitation tels qu'Unix, Microsoft Windows, Mac OS ou Linux avec peu ou pas de modifications... C'est la plate-forme qui garantit la portabilité des applications développées en Java.

Le langage reprend en grande partie la syntaxe du langage C++, très utilisé par les informaticiens. Néanmoins, Java a été épurée des concepts les plus subtils du C++ et à la fois les plus déroutants, tels que l'héritage multiple remplacé par l'implémentation des interfaces. Les concepteurs ont privilégié l'approche orientée objet de sorte qu'en Java, tout est objet à l'exception des types primitifs (nombres entiers, nombres à virgule flottante, etc.). Java permet de développer des applications autonomes mais aussi, et surtout, des applications client-serveur. Côté client, les applets sont à l'origine de la notoriété du langage. C'est surtout côté serveur que Java s'est imposé dans le milieu de l'entreprise grâce aux servlets, le pendant serveur des applets, et plus récemment les JSP (Java Server Pages) qui peuvent se substituer à

PHP, ASP et ASP.NET. Les applications Java peuvent être exécutées sur tous les systèmes d'exploitation pour lesquels a été développée une plate-forme Java, dont

le nom technique est JRE<sup>40</sup> (Environnement d'exécution Java). Cette dernière est constituée d'une JVM<sup>41</sup> (Machine Virtuelle Java), le programme qui interprète le code Java et le convertit en code natif. Mais le JRE est surtout constitué d'une bibliothèque standard à partir de laquelle doivent être développés tous les programmes en Java. C'est la garantie de portabilité qui a fait la réussite de Java dans les architectures client-serveur en facilitant la migration entre serveurs, très difficile pour les gros systèmes.

### V.2.2 Netbeans environnement de développement

NetBeans est un environnement de développement intégré (EDI), placé en open source par Sun en juin 2000 sous licence CDDL et GPLv2 (Common Development and Distribution License). En plus de Java, NetBeans permet également de supporter différents autres langages, comme Python, C, C++, JavaScript, XML, Ruby, PHP et HTML. Il comprend toutes les caractéristiques d'un IDE moderne (éditeur en couleur, projets multi-langage, refactoring, éditeur graphique d'interfaces et de pages Web). Conçu en Java, NetBeans est disponible sous Windows, Linux, Solaris (sur x86 et SPARC), Mac OS X ou sous une version indépendante des systèmes d'exploitation (requérant une machine virtuelle Java). Un environnement Java Development Kit JDK est requis pour les développements en Java.

NetBeans constitue par ailleurs une plateforme qui permet le développement d'applications spécifiques (bibliothèque Swing (Java)). L'IDE NetBeans s'appuie sur cette plateforme. L'IDE Netbeans s'enrichit à l'aide de plugins, Il comprend toutes les caractéristiques d'un IDE moderne [93].

### V.2.3 Java EE and Web Application

Le terme « Java EE » signifie Java Enterprise Edition, et était anciennement raccourci en « J2EE ». Il fait quant à lui référence à une extension de la plate-forme standard. Autrement dit, la plate-forme Java EE est construite sur le langage Java et la plate-forme Java SE, et elle y ajoute un grand nombre de bibliothèques remplissant tout un tas de fonctionnalités que la plate-forme standard ne remplit pas d'origine.

---

<sup>40</sup> Java Runtime Environment.

<sup>41</sup> Java Virtuelle Machine.

L'objectif majeur de Java EE est de faciliter le développement d'applications web robustes et distribuées, déployées et exécutées sur un serveur d'applications

On utilise java web application pour la création du web service (notre broker).  
[94]

#### **V.2.4 Glass fish**

Apache Tom cat software est un serveur web (http) open source mis en œuvre pour gérer Java Servlet, des Pages Java Server, la Langue d'Expression Java et Java Web Socket des technologies. Il est développé sous le Processus Communautaire Java. [95]

#### **V.2.5 CloudSim**

CloudSim est un nouveau général, et extensible cadre de simulation qui permet la modélisation, la simulation et l'expérimentation de nouvelles cloud Computing infrastructures et de services d'application. Dans le cas Cloud Computing, les outils de Simulations comme CloudSim donne une offre d'importants avantages aux clients et fournisseurs. Pour les clients, il permet de tester leurs services dans un environnement contrôlable avec exempt du coût et de vérifier la performance avant de publier les vrais nuages. Pendant ce temps pour les fournisseurs, de leur permettre de vérifier les types de location en fonction de divers prix et la charge ; En outre, cela permettra d'optimiser le coût de l'accès aux ressources à l'amélioration des bénéfices. Sans ces outils, à la fois des clients et des fournisseurs doit se appuyer sur des évaluations imprécises, ou sur des approches essai-erreur, ces approches peuvent conduire à l'inefficacité performance des services et de réduire la génération de revenus. En outre, CloudSim aide les chercheurs et développeurs basés sur l'industrie pour tester la performance d'un service d'application développée dans un environnement convenable et facile à installer.

Il y a de nombreux avantages de l'utilisation CloudSim pour tester les performances de départ, comme : (i) l'efficacité de temps : il prend très moins de temps et d'efforts pour mettre en œuvre des applications de cloud computing et (ii) la flexibilité : les développeurs peuvent facilement modéliser et tester les performances de leurs applications et ses services dans des environnements hétérogènes (Microsoft Azure, Amazon EC2). [96]

### *V.2.3.1 L'architecture de CloudSim*

La structure logiciel de cloudsim et ses composant est représenter par une architecture en couche comme il montré par la figure 33. Les premières versions du cloudsim utilise Simjava, un moteur de simulation d'évènement discret qui met en œuvre les principales fonctionnalités requises pour des structures de simulation de haut niveau, parmi les fonctionnalités, nous avons la formation d'une file d'attente et le traitement d'évènement, la création de composants système (les services, les machines (host), les centre de données(Datacenter), le courtier (broker), les machines virtuelles), la communication entre les composants et la gestion de l'horloge de simulation.

CloudSim supporte la modélisation et la simulation de l'environnement de Datacenter basé sur Cloud, tel que des interfaces de gestion dédiées aux VMs, la mémoire, le stockage et la bande passante. La couche CloudSim gère l'instanciation et l'exécution des entités de base (VM, hôtes, Datacenter, applications) au cours de la période de simulation. Dans la couche plus haute de la pile de simulation, on trouve le code de l'utilisateur qui expose la configuration des fonctionnalités liées aux hôtes (ex: nombre de machines, leurs spécifications), les politiques d'ordonnancement de Broker, applications (ex: nombre de tâches et leurs besoins), VM, nombre d'utilisateurs.



- **Virtual Machine** : Cette classe modéliser une instance de virtuelle machine (VM), qui est géré et hébergé pendant son cycle de vie par le composant cloud host, un host peut simultanément instancier de multiples VMs et assigner des cœurs à base de politiques prédéfinies de partage de processeur (espace partagé, temps partagé).
- **Datacenter** : Cette classe modéliser l'infrastructure du noyau du service (matériel, logiciel) offert par des fournisseur de ressources dans un environnement de Cloud Computing .il encapsule un ensemble de machine de calcul qui peuvent être homogène ou hétérogènes en ce qui concerne leur configurations de ressources (mémoire , noyau ,capacité et stockage ) .En outre ,chaque composant de Datacenter instancie un composant généralisé d'approvisionnement de ressource qui implémente un ensemble de politiques d'allocation de bande passante, de mémoire et des dispositif de stockage ,l'HOST représente un serveur informatique physique dans un Cloud, Host exécute des actions liées à la gestion des machines virtuelles et a une politique définie pour l'approvisionnement mémoire et BP, ainsi que d'une politique de répartition des PE à des machines virtuelles. Un hôte est associé à un Datacenter. Il peut héberger des machines virtuelles.
- **DataCentreBroker** : Cette classe modéliser le courtier (Broker), qui est responsable de la médiation entre les utilisateurs et les prestataires de service selon les conditions de QoS des utilisateurs et il déploie les tâche de service a travers les clouds. Le Broker agissant au nom des utilisateurs identifie les prestataires de service appropriés du cloud par le service d'information du cloud CIS (cloud information services) en négocie avec eux pour une allocation des ressources qui répond aux besoins de QoS des utilisateurs.
- **VmAllocationPolicy** : C'est une classe abstraite représente une politique de ravitaillement. Il signifie que le moniteur de VM est utilisé pour répartir VM aux hôtes. La fonctionnalité principale de cette classe doit choisir n'importe quel hôte qui est disponible dans Datacenter qui respecte le stockage, la mémoire, l'exigence de disponibilité pour un VM.
- **Ram provisioning** : C'est une classe abstraite qui est utilisée à représente la politique (police) de ravitaillement pour allouer la mémoire principale au VM'S.

L'exécution et le développement de VM sur l'hôte sont faisables seulement quand les composants de Disposition RAM approuvent que l'hôte a exigé la fourmi de mémoire disponible.

- **Bandwidth Provisioner** : C'est une classe abstraites qui est utilisées pour ravitailler de bande passante à VM. La tâche principale de cette classe est d'entreprendre l'allocation de bande passante de réseau à l'ensemble du VM'S qui est développé à travers le Datacenter. Si n'importe quel chercheur veut prolonger cette classe, donc ils peuvent prolonger cette classe avec leurs propres politiques (priorités, QoS). `BandwidthProvisoingSimple` permettent à une VM de réserver autant que la BP<sup>42</sup> que la largeur de bande) comme exigé.

### V.3 Description de l'application :

La version de Cloudsim 3.0.3 n'a pas d'interface graphique, il utilise le mode console pendant l'exécution, ce qui rend difficile à l'utilisateur de profiter pleinement de cloudsim, comme la modification des paramètres de simulation et l'affichage graphique des outputs (résultat de simulation).

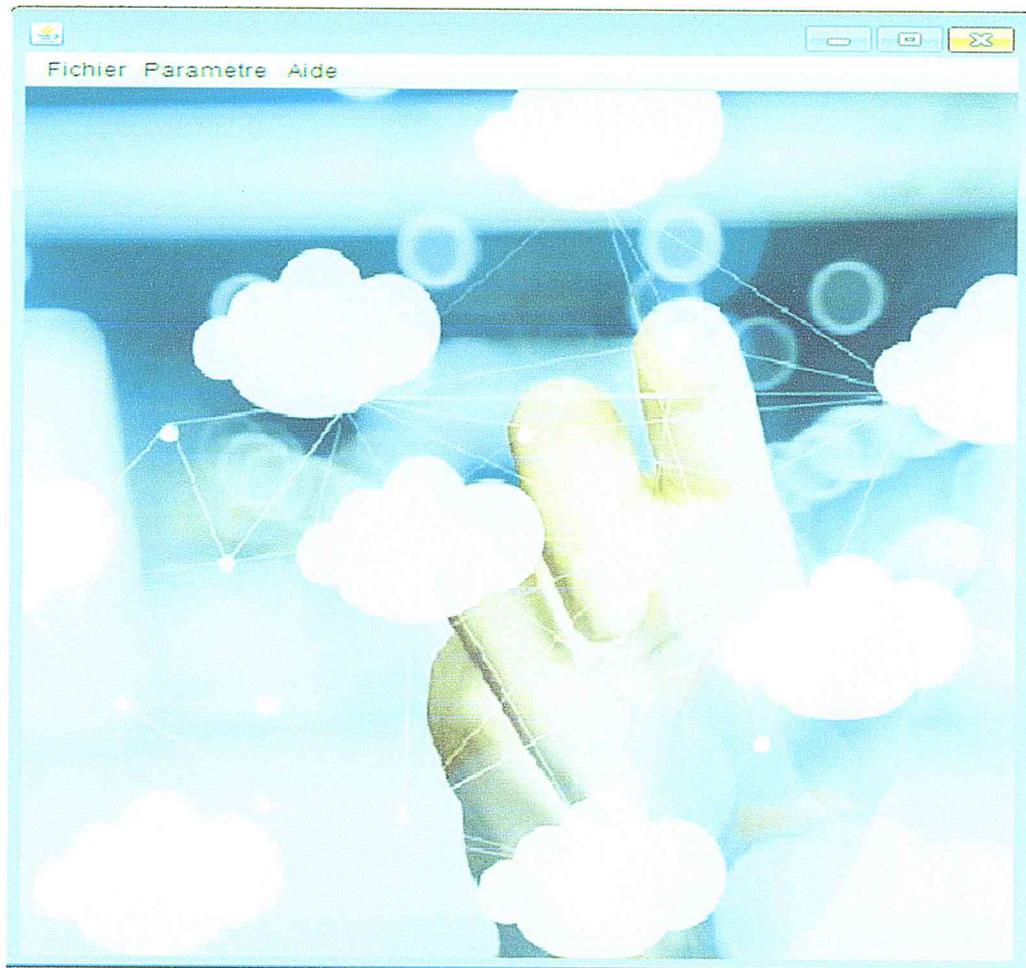
Nous avons créé une interface graphique qui facilite l'accès et la manipulation du simulateur. Nous allons présenter les différentes étapes effectuées pour réaliser notre simulation.

#### V.3.1 Interface principale

Dès le lancement de notre application, la figure 34 suivante apparaît en premier, elle est constituée d'une barre de menu contenant les menus suivants : le menu Fichier, Paramètre, et Aide, chaque menu contient un ensemble d'items.

---

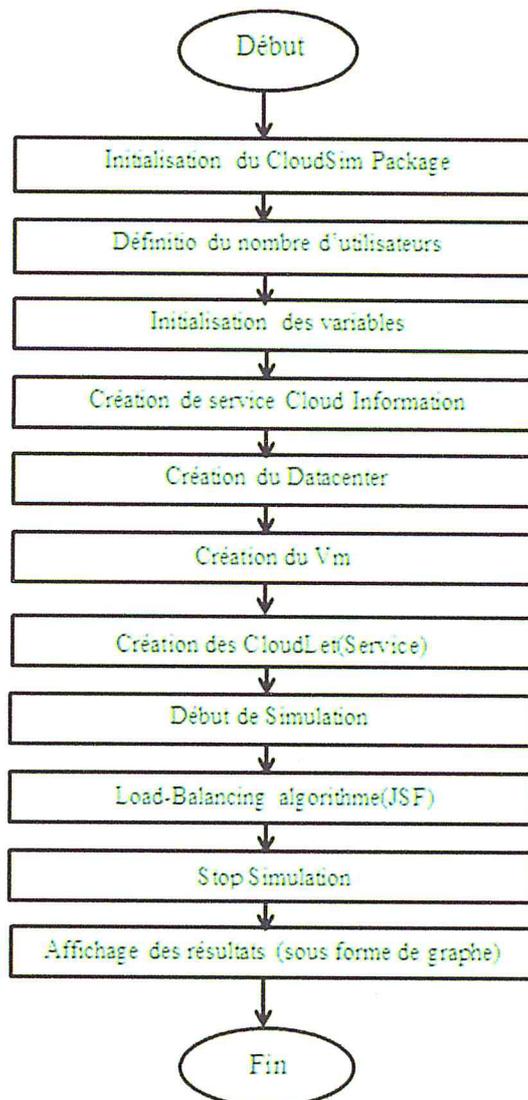
<sup>42</sup> Bande Passante.



*Figure 34 : interface principale*

### **V.3.2 Simulation d'un seul cloud**

Le cycle de vie de la simulation dans l'environnement CloudSim est illustré dans le processus suivant (figure N°35) :



*Figure 35 : processus de simulation*

### ***Configuration de Datacenter***

Cette étape consiste à saisir les paramètres nécessaires propres à la topologie du réseau (figure 36) comme le nombre d'hôte de Datacenter, le nombre de core (CPU) et leur mesure (MIPS), la taille de la mémoire, le cout de traitement , le cout de la mémoire , le cout de disque dur , le cout de stockage , la bande passante et son cout, et le fuseau horaire. Le nombre de Datacenter dans notre travail de simulation est égal à 1.

Lors de la configuration des Datacenter ils doivent respecter la valeur appropriée de chaque paramètre, tel que NbHost représente le nombre de machine

physique existante dans le centre de données ; pour le Stockage représente la taille de stockage de l'host Nombre de 10000 et plus (host Storage (MB)) ; La bande passante BW à offrir prend le nombre 1000 et plus (BW = 1000 et plus)

The screenshot shows a configuration window for a DataCenter. At the top, there are three tabs: 'DataCenter' (selected), 'VM', and 'Cloudlet'. Below the tabs, there are several input fields for configuring a VM. The fields are arranged in a grid-like format with checkboxes on the left and values in text boxes on the right.

<input type="checkbox"/> Nb hôte	1	<input type="checkbox"/> RAM	2048
<input type="checkbox"/> Stockage	1000000	<input type="checkbox"/> BP: Bande Passante	10000
<input type="checkbox"/> CPU			4
<input type="checkbox"/> MIPS:millions d'instructions par seconde			1000
<hr/>			
Architecture	x64	Coût / BP	0.1
Coût / CPU	3	fuseau horaire	10
Coût / mémoire	0.05	Coût / stockage	0.05
Système opérationnel			Windows
Gestionnaire de machine virtuelle			Xen

*Figure 36 : Configuration de Datacenter*

### ***Configuration de Machine Virtuelle***

Cette étape consiste à configurer les paramètres nécessaires propres aux machines virtuelles (figure 37). La définition de nombre de machine virtuelle VM, et le CPU dans une VM, la spécification de la RAM, la taille de disque dure et la bande passante.

La configuration des machines virtuelles représente l'affectation de chaque paramètre par les valeurs concernées.

Option	Value
<input type="checkbox"/> Nb machine Virtuelle	
<input type="checkbox"/> Stockage	10000
<input type="checkbox"/> RAM	512
<input type="checkbox"/> BP	1000
<input type="checkbox"/> MIPS	250
<input type="checkbox"/> CPU	1

*Figure 37 : configuration des VM*

### ***Configuration de Cloudlet***

Dans cette partie, on va travailler avec les cloudlets telle qu'en définissent les caractéristiques tâches qui attribue par Cloudlet. Cela signifie que nous pouvons configurer le comportement du cloudlet : Longueur cloudlet (paramétré par MIPS - qui influent sur le temps d'exécution de la tâche dans une machine virtuelle), fichier de sortie (que d'impact sur des simulations avec le stockage) ,le nombre de CPU (PEs) utilisé par cloudlet (qui aura un impact sur la quantité MIPS utilisé pour traiter la tâche), le modèle d'utilisation (qui définit quelle comportement que vous tâche fera lorsque son utilisation PEs, mémoire et bande passante). La figure (38) montre les paramètres configurer sur l'onglet Cloudlets : Nb pour le nombre du cloudlet a exécuter dans le cloud ; La taille du fichier de entrée/sortie de la cloudlet avant/après l'exécution (ex long file Size = 350) ; longueur du cloudlet : la taille du cloudlet à exécuter dans le Cloud Ressource (ex long length = 40000) ;

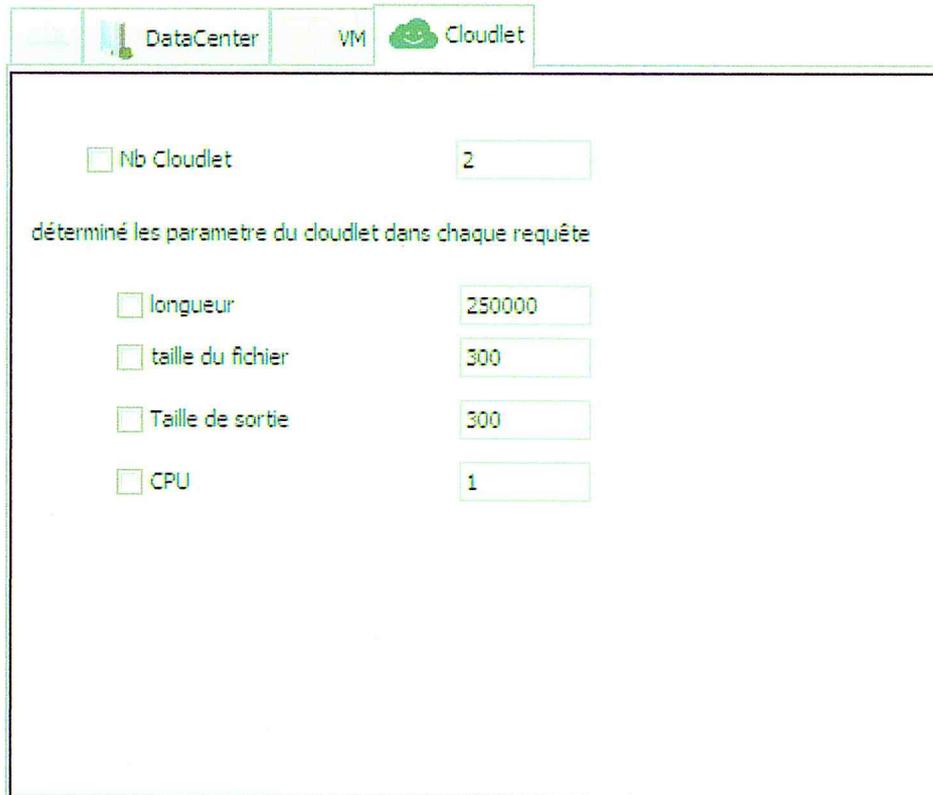


Figure 38 : configuration du Cloudlet.

### Expérience 01 :

Dès le lancement de l'application, et après le choix d'une simulation simple. Nous voulons mesurer le temps de réponse par rapport à la longueur du cloudlet (length) et en attribuer la série de simulation par les mêmes caractéristiques, cette simulation a été réalisée avec les paramètres de simulation suivant :

**Nombre de simulation = 4 ;**

**Nombre Cloudlet = 5 ;**

**Cloudlet File size (taille fichier) = 300**

**Cloudlet Output Size (taille de fichier de sortie) = 300 ;**

**Machine Virtuelle Nb=5 ; Mips = 250 ; Ram = 512 ; bP = 1000 ; CPU=1**

**Datacenter (host caractéristique) Ram = 2048 ; Storage = 1000000 ; BW = 10000 ; CPU=4 ; Mips = 1000.**

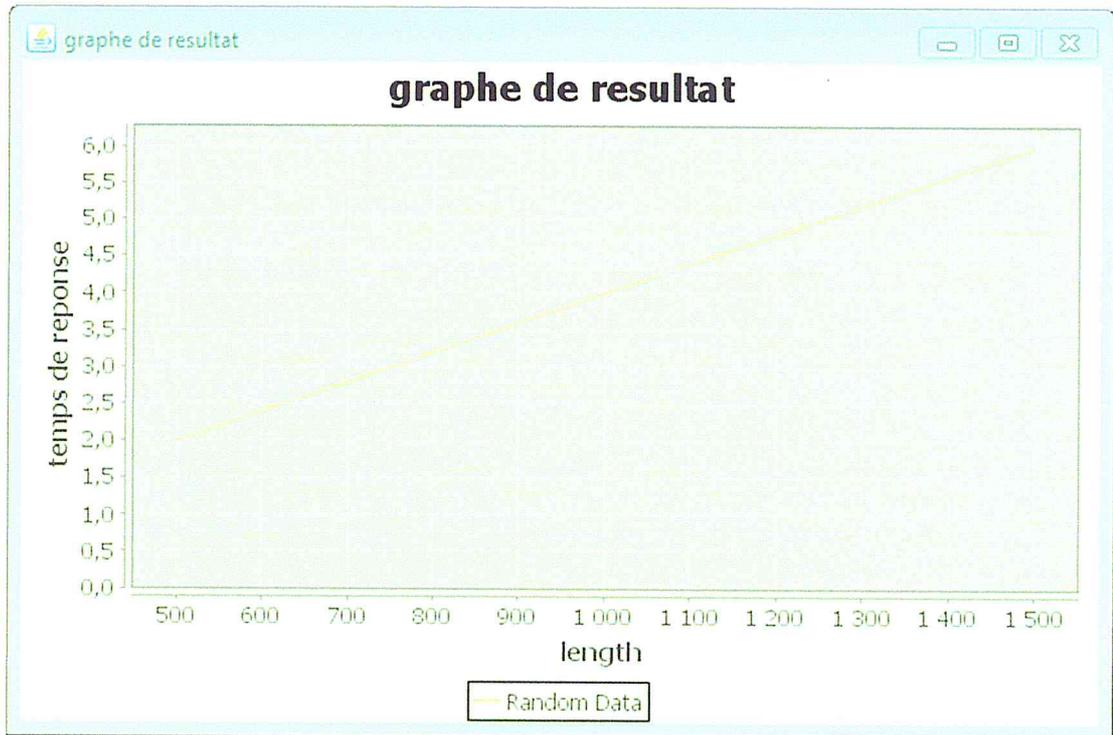


Figure 39 : graphe résultat de la simulation d'un seul Cloud

Le graphe de la figure 39 représente les résultats de simulation par rapport à la longueur du cloudlet, on remarque une augmentation très élevée du temps de réponse en fonction du length (longueur du cloudlet).

### V.3.3 Simulation de la solution (Interopérabilité)

La simulation de la solution nécessite :

#### V.3.3.1 configuration du Cloud Hybride

La figure 41 est l'interface où l'utilisateur entre les paramètres pour la configuration du Cloud 01 et Cloud 02 (comme décrit dans la partie V.3.2) :

- création des Datacenter (les hôtes) ;
- puis les machines virtuelles ;
- création des Cloudlet, dans notre cas, c'est les SaaS où on va exécuter nos tâches, les paramètres des SaaS sont représentés dans la figure N° 40.

- prérequis {
  - . si = 0 ; SaaS n'a pas de prérequis
  - . si < > 0 ; SaaS a une prérequis qui est un sortie d'un autre service SaaS qui se trouve dans l'autre cloud dans ce cas c'est sont l'identifiant
- output « sortie » → de type long elle peut être une entré dans un autre service
- longueur « length » → de type long c'est la taille du SaaS

Figure 40 : Les paramètres du SaaS à simuler

 Cloud\_1

 Cloud\_2

**Configuration du 1er Cloud**

**Hote**

Nb° hote	<input type="text"/>	Nb° CPU	<input type="text"/>
RAM	<input type="text"/>	Stockage	<input type="text"/>
MIPS	<input type="text"/>	BP	<input type="text"/>

**VM**

Nb° VM	<input type="text"/>	Stockage	<input type="text"/>
RAM	<input type="text"/>	BP	<input type="text"/>
MIPS	<input type="text"/>	CPU	<input type="text"/>

**SaaS**

Nombre du service dans le 1er Cloud

Taille du service

Taille de fichier de sortie

Prérequis

Figure 41 : Fenêtre pour la configuration du cloud 1

### V.3.3.2 création du BROKER

La création du broker implique la création d'un web service en utilisant java web application.

Notre web service assure l'interopérabilité entre les deux Clouds (Cloud hybride) grâce à ses méthodes web (opération) offre.

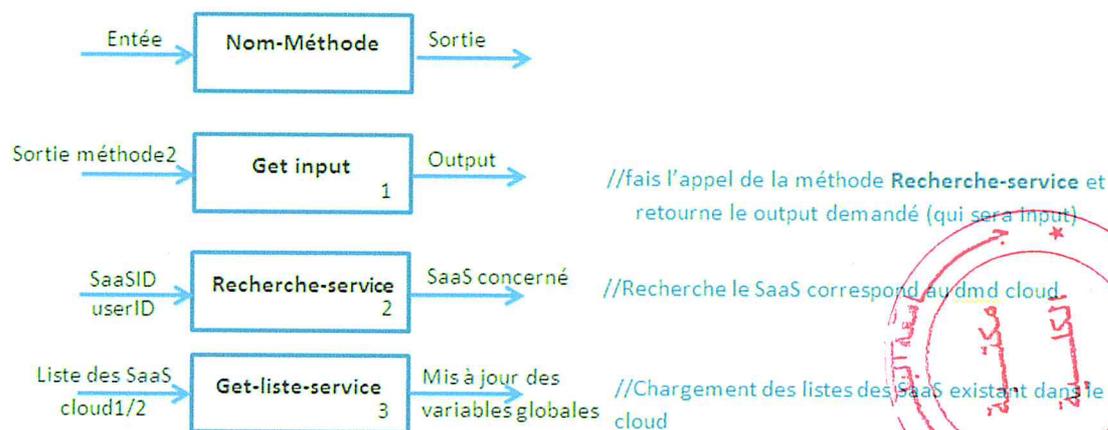


Figure 42 : Les méthodes web du broker

Après la définition du web service et le déploiement sur le serveur tom cat on sera capable de récupérer la description en WSDL du broker sous forme de fichier XML (figure 43).

This XML file does not appear to have any style information associated with it. The document tree is shown below.

```
<!-- Published by JAX-WS RI at http://jax-ws.dev.java.net. RI's version is
JAX-WS RI 2.2-hudson-740-. -->
-->
<!-- Published by JAX-WS RI at http://jax-ws.dev.java.net. RI's version is
JAX-WS RI 2.2-hudson-740-. -->
<!--
Generated by JAX-WS RI at http://jax-ws.dev.java.net. RI's version is JAX-
WS RI 2.2-hudson-740-.
-->
<definitions xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-
200401-wss-wssecurity-utility-1.0.xsd" xmlns:wsp="http://www.w3.org/ns/ws-
policy"xmlns:wsp1_2="http://schemas.xmlsoap.org/ws/2004/09/policy" xmlns:wsam
="http://www.w3.org/2007/05/addressing/metadata" xmlns:soap="http://schemas.xml
soap.org/wsdl/soap/"xmlns:tns="http://bROKER.Cloud/" xmlns:xsd="http://www.w
3.org/2001/XMLSchema" xmlns="http://schemas.xmlsoap.org/wsdl/" targetNamespa
ce="http://bROKER.Cloud/" name="WSbrokerService">
```

```

<Types>
  <xsd:schema>
    <xsd:
import namespace="http://bROKER.Cloud/" schemaLocation="http://localhost:8080
/WebServiceBroker/WSbroker?xsd=1"/>
    </xsd:schema>
  </types>
  <message name="getInput">
    <part name="parameters" element="tns:getInput"/>
  </message>
  <message name="getInputResponse">
    <part name="parameters" element="tns:getInputResponse"/>
  </message>
  <message name="recherche">
    <part name="parameters" element="tns:recherche"/>
  </message>
  <message name="rechercheResponse">
    <part name="parameters" element="tns:rechercheResponse"/>
  </message>
  <message name="getCloudList1">
    <part name="parameters" element="tns:getCloudList1"/>
  </message>
  <message name="getCloudList1Response">
    <part name="parameters" element="tns:getCloudList1Response"/>
  </message>
  <message name="getCloudList2">
    <part name="parameters" element="tns:getCloudList2"/>
  </message>
  <message name="getCloudList2Response">
    <part name="parameters" element="tns:getCloudList2Response"/>
  </message>
  <portType name="WSbroker">
    <operation name="getInput">
      <input wsam:Action="http://bROKER.Cloud/WSbroker/getInputRequest" mes
sage="tns:getInput"/>
      <output wsam:Action="http://bROKER.Cloud/WSbroker/getInputResponse"
message="tns:getInputResponse"/>
    </operation>
    <operation name="recherche">
      <input wsam:Action="http://bROKER.Cloud/WSbroker/rechercheRequest" me
ssage="tns:recherche"/>
      <output wsam:Action="http://bROKER.Cloud/WSbroker/rechercheResponse"
message="tns:rechercheResponse"/>
    </operation>
    <operation name="getCloudList1">
      <input wsam:Action="http://bROKER.Cloud/WSbroker/getCloudList1Request
" message="tns:getCloudList1"/>

```

```

    <output wsam:Action="http://bROKER.Cloud/WSbroker/getCloudList1Respo
nse" message="tns:getCloudList1Response"/>
  </operation>
  <operation name="getCloudList2">
    <input wsam:Action="http://bROKER.Cloud/WSbroker/getCloudList2Request
" message="tns:getCloudList2"/>
    <output wsam:Action="http://bROKER.Cloud/WSbroker/getCloudList2Respo
nse" message="tns:getCloudList2Response"/>
  </operation>
</portType>
<binding name="WSbrokerPortBinding" type="tns:WSbroker">
  <soap:binding transport="http://schemas.xmlsoap.org/soap/http" style="docum
ent"/>
  <operation name="getInput">
    <soap:operation soapAction=""/>
    <input>
      <soap:body use="literal"/>
    </input>
    <output>
      <soap:body use="literal"/>
    </output>
  </operation>
  <operation name="recherche">
    <soap:operation soapAction=""/>
    <input>
      <soap:body use="literal"/>
    </input>
    <output>
      <soap:body use="literal"/>
    </output>
  </operation>
  <operation name="getCloudList1">
    <soap:operation soapAction=""/>
    <input>
      <soap:body use="literal"/>
    </input>
    <output>
      <soap:body use="literal"/>
    </output>
  </operation>

```

```

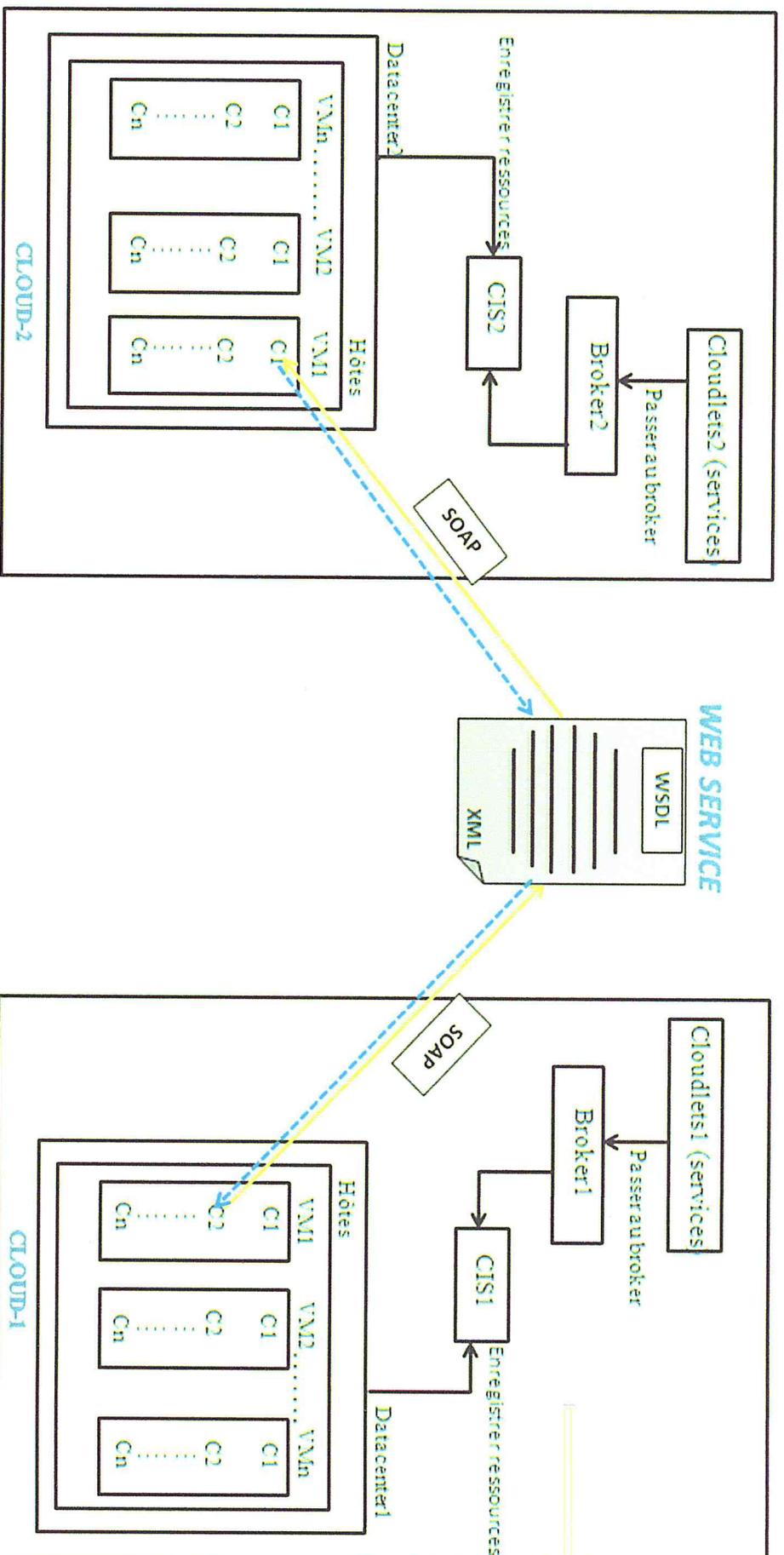
<operation name="getCloudList2">
  <soap:operation soapAction=""/>
  <input>
    <soap:body use="literal"/>
  </input>
  <output>
    <soap:body use="literal"/>
  </output>
</operation>
</binding>
<service name="WSbrokerService">
  <port name="WSbrokerPort" binding="tns:WSbrokerPortBinding">
    <soap:address location="http://localhost:8080/WebServiceBroker/WSbr
oker"/>
  </port>
</service>
</definitions>

```

Figure 43 : Description du broker en WSDL

### V.3.3.3 Description des interactions Cloud/broker

Le schéma (figure44) suivant représente le cycle de vie de la solution sous CloudSim.



C1 ..... Cn : cloudlet (ou bien les services) // CIS1-2 : Cloud Information Service // Cloudlets1-2 : la liste des cloudlets // VM1 ..... VMn : des virtuelles machines  
 Les hôtes et les VM contiennent des configurations cotées hardware : CPU, MIPS, R.A.M. // → SOAP demande // ← SOAP réponse

Figure 44: Vue globale sur l'implementation de la solution proposée

La figure 45 définit un exemple du message SOAP (requête) pour la méthode get-Input.

La demande du prérequis par le Cloud 1 au broker c'est la demande de la sortie (output) du service numéro 3 dans le Cloud 2 et la récupération de résultat est dans un message réponse qui retourne 300 qui est la sortie du service 3 et en même temps c'est l'entrée du service demandé figure 46.

#### SOAP Request

---

```
<?xml version="1.0" encoding="UTF-8"?> <S:Envelope xmlns:
S="http://schemas.xmlsoap.org/soap/envelope/" xmlns: SOAP=
ENV="http://schemas.xmlsoap.org/soap/envelope/">
<SOAP-ENV:Header/>
<S:Body>
  <ns2:getInput xmlns:ns2="http://Example/">
    <ServiceID>3</ServiceID>
    <UserID>1</UserID>
  </ns2:getInput>
</S:Body>
</S:Envelope>
```

---

*Figure 45 : Enveloppe SOAP requête au Cloud pour la méthode getInput*

#### SOAP Response

---

```
<?xml version="1.0" encoding="UTF-8"?> <S:Envelope
xmlns:S="http://schemas.xmlsoap.org/soap/envelope/" xmlns: SOAP=
ENV="http://schemas.xmlsoap.org/soap/envelope/">
<SOAP-ENV:Header/>
<S:Body>
  <ns2:getIpuResponse xmlns:ns2="http://Example/">
    <return>300</return>
  </ns2:getInputResponse>
</S:Body>
</Envelop>
```

---

*Figure 46 : enveloppe SOAP réponde au Cloud pour la méthode getInput*

Dès la création des services (cloudlet par rapport à CloudSim), s'ils n'ont pas des prérequis ils seront exécutés directement, sinon (s'ils ont des prérequis) le Cloud envoie un message SOAP (requête) au broker grâce aux méthodes décrites dans le fichier WSDL, le prérequis ou bien l'entrée du service à la base c'est une sortie d'un autre service SaaS.

Le message SOAP arrive, le broker recherche le service dans l'autre Cloud et récupère sa sortie (output) et envoie sous forme d'un message SOAP une réponse au même temps au service demandeur qui reste en attente en attendant l'arrivée du prérequis.

Nous allons simuler la solution selon le critère du temps de l'interopérabilité « temps de communication / le temps d'attente » par rapport à la longueur (length) du service SaaS, Tell que :

$$\text{Temps de Communication} = \text{temps de réponse final} - \sum \text{temps d'exécution}$$

*Equation 1 : Calcule du temps de l'interopérabilité.*

Le temps de réponse final est le temps de fin d'exécution de simulation des deux Cloud et le temps d'exécution est le temps de traitement de cloud1+cloud2.

**Ou**

**Expérience :** la simulation de la solution a été réalisée avec les paramètres suivant :

**Nombre de simulation = 4 ;**

**Cloud1 :**

- **Nombre Cloudlet = 3 ;**
- **Cloudlet File size (taille fichier) = 300**
- **Cloudlet Output Size (taille de fichier de sortie) = 300 ;**
- **Machine Virtuelle Nb=5 ; Mips = 250 ; Ram = 512 ; bP = 1000 ; CPU=1.**
- **Datacenter (host caractéristique) Ram = 2048 ; Storage = 1000000 ; BW = 10000 ; CPU=4 ; Mips = 1000.**

### Cloud2 :

- **Nombre Cloudlet = 2 ;**
- **Cloudlet File size (taille fichier) = 300**
- **Cloudlet Output Size (taille de fichier de sortie) = 300 ;**
- **Machine Virtuelle Nb=5 ; Mips = 250 ; Ram = 512 ; bP = 1000 ; CPU=1.**
- **Datacenter (host caractéristique) Ram = 2048 ; Storage = 1000000 ; BW = 10000 ; CPU=4 ; Mips = 1000.**

### Prérequis :

- S1 : sans prérequis ;
- S2 : S4 ;
- S3 : sans prérequis ;
- S5 : S3 ;

### Résultat de l'expérience :

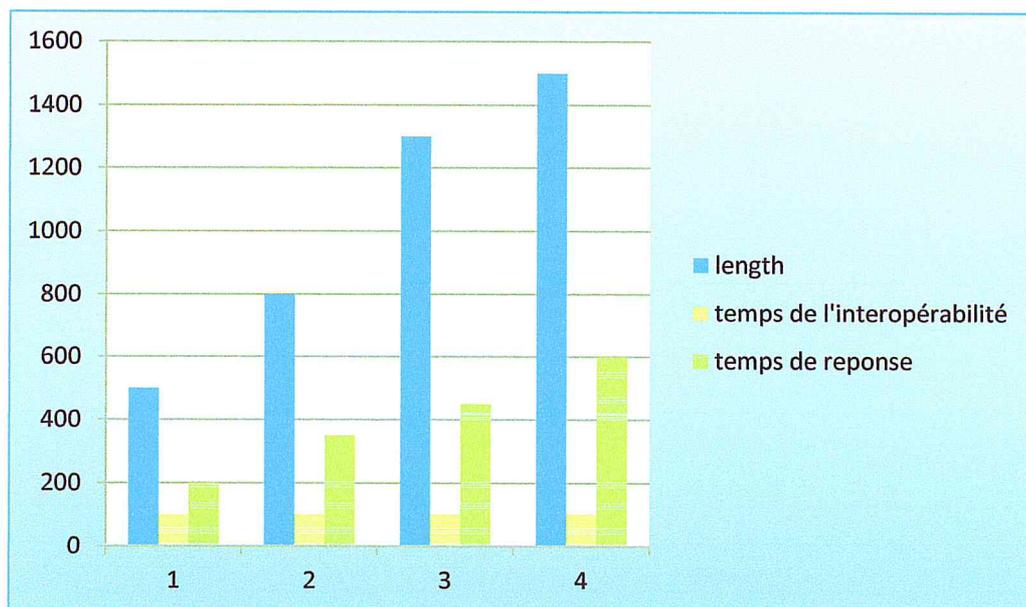


Figure 47 : Résultats de la simulation

#### V.3.3.4 Discussion :

Pour valider notre solution, et puisque il n'existe pas encore des benchmarks pour l'interopérabilité dans le Cloud Computing, nous avons choisi d'étudier un paramètre qui est le temps d'interopérabilité. Pour cela, nous avons réalisé une série de tests pour voir l'évolution du temps de l'interopérabilité par rapport à la taille du service « length » dans le but d'étudier et comparer l'évolution de chacun.

Dans la partie V.3.2 et après la simulation d'un seul Cloud (sans interopérabilité) on observe que :

1. le temps de réponse augmente par l'augmentation de la taille du service.

Dans notre solution et après simulation, on observe que le temps de l'interopérabilité augmente d'une manière très légère.

2. Le temps de communication reste stable par rapport à la taille du service.
3. Si le temps de réponse augmente cela implique que le temps de réponse final augmente aussi.

Et selon l'équation 1 on a :

4. Temps de Communication = temps de réponse final -  $\Sigma$  temps d'exécution.

D'après 1, 2, 3 et 4 on déduit que :

Malgré le changement dans la taille de service, le temps de communication reste stable.

## V.4 Conclusion

Dans ce chapitre, nous avons présenté la mise en œuvre de la solution proposée en utilisant le langage JAVA avec l'outil de simulation de Cloud Computing CloudSim. Nous avons réalisé une série de tests en simulant le temps de l'interopérabilité par rapport à la taille de cloudlets.

Les résultats montrent que l'utilisation de notre solution assure l'interopérabilité avec un temps de communication réduit.

## Conclusion Générale

Actuellement, il y a un grand problème d'interopérabilité entre les différents fournisseurs des solutions Cloud Computing. Le manque d'interfaces communes et/ou standardisées (API) freine l'interopérabilité, la migration de ressources (machines virtuelles, données et applications) d'un Cloud à un autre et l'interaction entre eux.

Dans ce mémoire, nous nous sommes penchés sur l'apport que pourrait avoir l'utilisation d'un modèle à base de broker pour l'interopérabilité dans un Cloud hybride de type SaaS.

Pour atteindre cet objectif, nous avons commencé dans un premier temps par étudier les notions de base du Cloud Computing, après nous avons exploré le problème d'interopérabilité dans les systèmes d'information en général avant de passer à l'interopérabilité dans les Clouds. Ce passage était nécessaire pour bien comprendre l'interopérabilité dans les Clouds car un Cloud est un système d'information qui déploie ses services à travers internet. Ensuite, nous avons fait une recherche bibliographique des travaux de recherche qui sont en relation avec notre problématique. Après cette étape, nous avons réalisé une étude comparative entre les solutions existantes.

L'interopérabilité des services Cloud de type logiciel en tant que service a été choisie car la plupart des solutions et des modèles existants mettent l'accent sur l'infrastructure en tant que service et plate-forme en tant que service. La recherche pour l'interopérabilité pour les services Cloud de type logiciel en tant que service est encore très immature.

Notre solution consiste à utiliser un broker de type service web jouant le rôle de médiateur dans un Cloud hybride. Le but étant de générer un fichier WSDL en introduisant un ensemble de méthodes d'interactions permettant l'interopérabilité des services (SaaS) entre Clouds.

Pour l'expérimentation de la solution proposée, nous avons utilisé le simulateur CloudSim. Les tests effectués ont donné de bons résultats surtout en termes de temps d'interopérabilité.

Plusieurs perspectives s'ouvrent à ce travail. En effet, pour cette première proposition nous avons choisi le critère de temps d'interopérabilité pour les tests de validation. D'autres critères peuvent être étudiés dans les futurs tels que la qualité et le coût d'interopérabilité.

## Bibliographie :

- [1] Peter Mell and Timothy Grance. The NIST Definition of Cloud Computing. National Institute of Standards and Technology, 2011.  
<http://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-145.pdf>.
- [2] Floerecke and Lehner. An Enhanced Cloud Ecosystem Model, la conférence européenne sur les systèmes d'information (ECIS), İstanbul, Turquie, 2016.
- [3] Vivansa ([www.vivansa.com](http://www.vivansa.com)). Rapport de veille technologique Cloud Computing. ref.: 13514-DLV-19833\_RPT Version: 1.01, (28-Sep-2009).
- [4] Jeff Barr, Le Cloud Computing avec Amazon Web Service, Collection : Référence, édition : Pearson France. 27 mai 2011.
- [5] Thomas Paviot. Méthodologie de résolution des problèmes d'interopérabilité dans le domaine du Product Lifecycle Management. PhD thesis, École Centrale Paris, 2010.
- [6] Nicholas. Carr. Rough type. Available online at: [Http://www.roughtype.com/p=279](http://www.roughtype.com/p=279). 2006.
- [7] Bhaskar Prasad Rimal & Admela Jukan & Dimitrios Katsaros & Yves Goeleven. Architectural Requirements for Cloud Computing Systems: An Enterprise Cloud Approach. Journal of Grid Computing, vol. 9, no. 1, pages 3–26, 2010.
- [8] Cor-Paul Bezemer & Andy Zaidman. Multi-tenant SaaS applications maintenance dream or nightmare. In Proceedings of the Joint ERCIM Workshop on Software Evolution (EVOL) and International Workshop on Principles of Software Evolution (IWPSE), pages 88–92. ACM, 2010.
- [9] Xi Wang. Development of an Interoperable Cloud-based Manufacturing System. PhD thesis, University of Auckland, 2014.
- [10] A. Khan. 2010. Evolution of Service Delivery Cloud Platform.
- [11] John W. Rittinghouse and James F. Ransome. CLOUD COMPUTING: Implementation, Management, and Security. First CRC Press, Inc. Boca Raton, FL, USA 2009.

- [12] Lien : <http://cloudcomputing.sys-con.com/node/1203562> [date dernière consultation 03/03/2017]
- [13] P. Costa, «Bridging the Gap Between Applications and Networks in Data Centers,» LADIS Workshop 12 Madeira, Portugal, 2012.
- [14] Evert Duipmans, Business Process Management in the cloud: Business Process as a Service (BPaaS). Literature Study. University of Twente. April 1, 2012.
- [15] T. Dillon, C. Wu and E. Chang. Cloud Computing: Issues and Challenges. 24 th IEEE International Conference on Advanced Information Networking and Applications (AINA), 2010. DOI : [10.1109/AINA.2010.187](https://doi.org/10.1109/AINA.2010.187)
- [16] Cloud Computing : une remise en question du rôle du SI dans l'entreprise ?  
<https://lalettreducloud.com/2016/04/05/cloud-computing-une-remise-en-question-du-role-du-si-dans-lentreprise/>
- [17] Cloud Computing : NOUVEAUX MODELES !, LIVRE BLANS, MARS 2012. <https://syntec-numerique.fr/sites/default/files/Documents/20120328-Infra-livreblanc-cloud-computing-nouveaux-modeles.pdf>
- [18] J. R. Winkler. La sécurité dans le Cloud. Pearson Education France. 2011. [https://www.pearson.fr/resources/titles/27440100447280/extras/2509\\_chap01.pdf](https://www.pearson.fr/resources/titles/27440100447280/extras/2509_chap01.pdf)
- [19] URL : [http://www.memoireonline.com/01/13/6713/m\\_Etude-et-mise-en-place-dunesolution-cloud-computing--privee-dans-une-entreprise-moderne-cas7.html](http://www.memoireonline.com/01/13/6713/m_Etude-et-mise-en-place-dunesolution-cloud-computing--privee-dans-une-entreprise-moderne-cas7.html)
- [20] 10 th Biyani International Conference (BICON-15) ISBN: 978-93-83462-78-0  
<https://www.coursehero.com/file/p51msse/10-th-Biyani-International-Conference-BICON-15-ISBN-978-93-83462-78-0-Data/>
- [21] WILLIAM VOORSLUYS, JAMES BROBERG, and RAJKUMAR BUYYA Introduction to cloud computing -White Paper. Chapter 01, 2011. <https://www.dialogic.com/~~/media/products/docs/whitepapers/12023-cloud-computing-wp.pdf>
- [22] Advanced Robotics (ICAR), 16th International Conference 25-29 Nov. 2013.

- [23] Lien <http://erods.liglab.fr/icar2013/docs/icar2013-panorama-cloud-computing.pdf>
- [24] M. Armbrust and al. A view of cloud computing. *Communications of the ACM*. Volume 53 Issue 4, Pages 50-58, April 2010.
- [25] CHEKAOUI Faïza. L'autonomie dans le Cloud Computing guidée par l'approche IDM. Mémoire magister, Université A.MIRA-BÉJAÏA, février 2015.
- [26] M. Alzubaidel and A. M. Elmogy. Cloud Computing Antecedents, Challenges, and Directions. *Proceedings of the International Conference on Internet of things and Cloud Computing. ICC '16*, March 22-23, 2016, Cambridge, United Kingdom, 2016.
- [27] Asma Talhi. Proposition d'une modélisation unifiée du Cloud Manufacturing et d'une méthodologie d'implémentation, basées sur les ontologies d'inférence. Thèse de doctorat. Ecole nationale supérieure d'arts et métiers - ENSAM, Paris, mai 2016.
- [28] F. F. Moghaddam and al. Cloud Computing Challenges and Opportunities: A Survey. *First International Conference on Telematics and Future Generation Networks (TAFGEN)*, 2015.
- [29] N. Leavitt. Is Cloud Computing Really Ready for Prime Time? *Computer*, Vol. 42, no. 1, pages 15–20, Jan 2009.
- [30] Jack M Germain. How to Avoid Cloud Vendor Lock-In, 2013. <http://www.linuxinsider.com/story/79417.html>
- [31] Ahmed Shawish & Maria Salama. Cloud Computing: Paradigms and Technologies. In *Inter-cooperative Collective Intelligence: Techniques and Applications*, volume 495, Pages 39–68. Springer, 2014
- [32] B. Kezia Rani, B. Padmaja Rani, A-Vinaya Babu. Cloud Computing and Inter-Clouds - Types, Topologies and Research Issues. *Second International Symposium on Big Data and Cloud Computing (ISBCC'15)*.
- [33] H. S. Jagdev & K.-D. Thoben, Anatomy of enterprise collaborations, *Journal Production Planning & Control*, The Management of Operations Volume 12, 2001 - Issue 5.
- [34] Lindeke, L.L. & Block, D.E. Maintaining professional integrity in the midst of interdisciplinary collaboration. *Nursing Outlook*, 46(5), pp.213–218. 1998.

- [35] Baxter, P. The CCARE model of clinical supervision: bridging the theory-practice gap. *Nurse education in practice*, 7(2), pp.103–11. 2007.
- [36] L. Bouzguenda, *Coordination multi-agents pour le Workflow interorganisationnellâche*, thèse de doctorat, IRIT, 2006.
- [37] Camarinha-Matos, L., et H. Afsarmanesh. *Collaborative Networks: Reference Modeling*. Élite par L. Camarinha-Matos ET H. Afsarmanesh. Springer-Verlag New York Inc., 2008.
- [38] D. Vanderhaeghen, D. Werth, T. Kahl, P. Loos, *Service-and Process-Matching- An Approach towards Interoperability Design and Implementation of Business Networks*, *Enterprise interoperability : new challenges and approaches*, Springer edition. ISBN: 978-1-84628-713-8, p. 189-198, 2006.
- [39] B. Bauer, J.P. Müller, S. Roser, *a Decentralized Broker Architecture for Collaborative Business Process Modelling and Enactement*, *Enterprise Interoperability: New Challenges and Approaches- Springer Verlag - ISBN-10:1846287138*, 2006.
- [40] N.Denos – K.Silini, *Travailler dans un environnement numérique évolutif*, C2i groupe de travail Interop de l'AFUL 16 août 2011
- [41] Hervé PINGAUD, *Cartes blanches au comité scientifique, Rationalité du développement de l'interopérabilité dans les organisations*, Centre de Génie Industriel Ecole des Mines d'Albi-Carmaux.
- [42] IEEE, *IEEE: Standard Computer Dictionary - A Compilation of IEEE Standard Computer Glossaries*, 1990.
- [43] Carney, D., D. Fisher, ET P. Place. «Topics in Interoperability: System-of-Systems Evolution. «Technical Note, University of Pittsburgh, Software Engineering Institute, 2005.
- [44] COM. «communication de la commission au conseil et au parlement europeen-L'interopérabilité des services paneuropéens d'administration en ligne.» 45 finals, Bruxelles, Belgique, 2006.
- [45] EIF. «European Interoperability Framework (EIF) for European public services» 2010.
- [46] EIF. « European Interoperability Framework for pan-European eGovernmentServices, Interoperable Delivery of European eGovernment Services to public Administrations, Businesses and Citizens (IDABC). » 2004.

- [47] Chen, D. & Daclin, N., 2006. Framework for enterprise interoperability". In Bordeaux, France: IFAC 2nd International Workshop on Enterprise Integration, Interoperability and Networking (EI2N'2006).
- [48] Obrst, L.J., 2004. Ontologies and Semantic Web for Semantic Interoperability. In McLean, USA: Semantic Technologies for e-Government Conference, pp. 366–369.
- [49] C4ISR. Architecture Working Group Final Report, (April) 1998.
- [50] NATO. NATO Allied Data Publication 34 (ADatP-34), NATO C3 Technical Architecture (NC3TA), Version 4.0.), 2003.
- [51] D Chen, N Daclin. Framework for Enterprise Interoperability - Proc. of IFAC Workshop EI2N, 2006 - researchgate.net
- [52] L Obrst, Technologies for semantically Interoperable Systems - Proceedings of the twelfth international conference 2003 - dl.acm.org
- [53] Amir Pirayesh Neghab, Evaluation basée sur l'interopérabilité de la performance des collaborations dans le processus de conception. thèse de doctorat. UNIV-LORRAINE 2014.
- [54] Buyya R, Yeo CS, Venugopal S, Broberg J, Brandic I. Cloud computing and emerging IT platforms: vision, hype, and reality for delivering computing as the 5th utility. *Future Generation Computer Systems* 2009; 25(6):599–616.
- [55] Lewis G. Basics about cloud computing. *Technical Report September*, Software Engineering Institute – Carnegie Mellon, 2010
- [56] Kelly K. The Technium: A Cloudbook for the Cloud. Available from: [http://www.kk.org/thetechnium/archives/2007/11/a\\_cloudbook\\_for.php](http://www.kk.org/thetechnium/archives/2007/11/a_cloudbook_for.php)
- [57] Buyya R, Ranjan R, Calheiros RN. InterCloud: utility-oriented federation of cloud computing environments for scaling of application services. In *Proceedings of the 10th International Conference on Algorithms and Architectures for Parallel Processing*. Springer-Verlag: Busan, Korea, 2010; 13–31.
- [58] Amazon. Summary of the Amazon EC2 and Amazon RDS Service Disruption. Available from: <http://aws.amazon.com/message/65648> [dernière consultation 04/09/2017]

- [59] Microsoft. Windows Azure Service Disruption Update. Available from: <http://blogs.msdn.com/b/windowsazure/archive/2012/03/01/windows-azure-service-disruption-update.aspx> [dernière consultation 04/09/2017]
- [60] Armbrust M, Fox A, Griffith R, Joseph A, Katz R. Above the Clouds: A Berkeley View of Cloud Computing. *Technical Report*, Electrical Engineering and Computer Sciences, University of California, Berkeley, 2009.
- [61] Beloglazov A, Buyya R, Lee YC, Zomaya A. A taxonomy and survey of energy-efficient data centers and cloud computing systems. *Advances in Computers*, M. Zelkowitz (ed.) 2011; 82:47–111.
- [62] Global Inter-Cloud Technology Forum. Use Cases and Functional Requirements for Inter-Cloud Computing. *Technical Report*, Global Inter-Cloud Technology Forum, 2010.
- [63] Sakashita Y, Takayama K, Matsuo A, Kurihara H. Cloud Fusion Concept. *FUJITSU Scientific & Technical Journal (FSTJ)* 2012; 48(2):143–150.
- [64] M.HAMZE. Autonomie, sécurité et QoS de bout en bout dans un environnement de Cloud Computing. Thèse doctoral. école doctorales sciences pour l'ingénieur et micro techniques UNIVERSITÉ DE BOURGOGNE
- [65] 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.
- [66] Ferrer AJ, Hernández F, Tordsson J, Elmroth E, Ali-Eldin A, Zsigri C, Sirvent R, Guitart J, Badia RM, Djemame K, Ziegler W, Dimitrakos T, Nair SK, Kousiouris G, Konstanteli K, Varvarigou T, Hudzia B, Kipp A, Wesner S, Corrales M, Forgó N, Sharif T, Sheridan C. OPTIMIS: a holistic approach to cloud service provisioning. *Future Generation Computer Systems* 2012; 28(1):66–77.
- [67] Keahey K, Tsugawa M, Matsunaga A, Fortes J. Sky computing. *Internet Computing* 2009; 13(5):43–51.
- [68] Petcu D, Crciun C, Neagul M, Panica S, Di Martino B, Venticinque S, Rak M, Aversa R. Architecturing a sky computing platform. In *Proceedings of the International Conference Towards a Service-Based Internet*

*ServiceWave '10*, Vol. 6569, Cezon M, Wolfsthal Y (eds). Springer-Verlag: Ghent, Belgium, 2011; 1–13.

[69] «Cloud federation » En ligne

<http://searchcloudprovider.techtarget.com/definition/Whatis-cloud-federation>

[dernière consultation 04/09/2017]

[70] Matthew B. Becker. Interoperability Case Study Cloud

Computing.berkmen. Research Publication No. 2012-11 April 2012

[71] IEEE STANDARDS ASSOCIATION.

<http://standards.ieee.org/develop/project/2301.html>. Septembre 2011.[dernière consultation 04/09/2017]

[72] OASIS Cloud Application Management for Platforms (CAMP).

<http://docs.oasisopen.org/tosca/TOSCA/v1.0/TOSCA-v1.0.html>, Août 2012.

[73] Topology and Orchestration Specification for Cloud Applications.

<http://docs.oasis-open.org/tosca/TOSCA/v1.0/TOSCA-v1.0.html>, Janvier 2012.

[74] DMTF to develop standards for managing a cloud computing environment. <http://www.dmtf.org/standards/cloud>, Juin 2011.

[75] EDMONDS ANDY, METSCH THIJS, PAPASPYROU ALEXANDER ET RICHARDSON ALEXIS: Toward an open cloud standard. *Internet Computing, IEEE*, 16(4):15–25, 2012

[76] SLIK D, SIEFER M, HIBBARD E, SCHWARZER C, YODER A, BAIRAVASUNDARAM LN, BAKER S, CARLSON M, NGUYEN H ET RAMOS R: Cloud data management interface (cdmi) v1. 0, 2011

[77] Nam-Luc Tran, Sabri Skhiri, and Zimanyi Esteban. EQS: An Elastic and Scalable Message Queue for the cloud. In 2011 IEEE Third International Conference on Cloud Computing Technology and Science, pages 391–398. IEEE, November 2011

[78] MOSCATO FRANCESCO, AVERSA ROCCO, DI MARTINO BENIAMINO, FORTIS T ET MUNTEANU VICTOR: An analysis of mOSAIC ontology for Cloud resources annotation. In Federated Conference on Computer Science and Information Systems (FedCSIS), pages 973–980. IEEE, 2011.

- [79] PAWLUK PRZEMYSŁAW, SIMMONS BRADLEY, SMIT MICHAEL, LITOIU MARIN ET MANKOVSKI SERGE: Introducing stratos: A cloud broker service. In IEEE CLOUD, pages 891–898, 2012
- [80] YANGUI SAMI, MARSHALL IAIN-JAMES, LAISNE JEAN-PIERRE ET TATA SAMIR CompatibleOne : The Open Source Cloud Broker. Journal of Grid Computing, pages 1–17, 2013
- [81] Rogers, O., Cliff, D.: A financial brokerage model for Cloud computing. J. Cloud Comput. Adv. Syst. Appl. 2012, 1, 2 (2012).  
<http://www.Cloud-casa.com/content/1/1/2> [dernière consultation 04/09/2017].
- [82] Wu, F., Zhang, L., Huberman, B.: Truth-telling reservations. Lect. Notes Comput. Sci. 3828, 8091 (2005).doi:10.1007/11600930 9.
- [83] Grozev & R. Buyya. Inter-Cloud architectures and application brokering: taxonomy and survey. Published online 12 December 2012 in Wiley Online Library (wileyonlinelibrary.com). DOI: 10.1002/ spe.2168. *Softw. Pract. Exper.* 2014; 44:369–390.
- [84] <http://www.virtustream.com/> [dernière consultation 04/09/2017].
- [85] <http://www.zdnet.fr/actualites/cloud-hybride-ibm-achete-le-cloud-broker-gravitant-39827638.htm> [dernière consultation 04/09/2017].
- [86] <http://www.informatiquenews.fr/cloud-broker-dhpe-un-bon-moyen-deviter-le-shadow-it-42264> [dernière consultation 04/09/2017].
- [87] <http://www.apriori-fx.com> [dernière consultation 04/09/2017].
- [88] <http://www.scoop.it/t/cloudnews/p/4024344068/2014/07/09/prologue-un-cloud-broker-francais> [dernière consultation 04/09/2017].
- [89] Jean-Louis Benard, Laurent Bossavit, Regis Medina, Dominic Williams. L'eXtreme Programming. Eyrolles, Paris, 2002.
- [90] D. A. Taylor. *Object Technology (2Nd Ed.): A Manager's Guide*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1998.
- [91] W. Emmerich and N. Kaveh. Component technologies: Java beans, com, corba, rmi, ejb and the corba component model. In *Proceedings of the 8th European Software Engineering Conference Held Jointly with 9th ACM SIGSOFT International Symposium on Foundations of Software Engineering, ESEC/FSE-9*, pages 311–312, New York, NY, USA, 2001. ACM.

- [92] C. Szyperski. *Component Software: Beyond Object-Oriented Programming*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 2nd edition, 2002.
- [93] W3C. W3c working group note. [Http://www.w3.org/TR/ws-gloss/](http://www.w3.org/TR/ws-gloss/). 2004.
- [94] langage java definition et historique [En ligne] Available at [http://www.comoria.com/3516/Java\\_%28langage%29](http://www.comoria.com/3516/Java_%28langage%29).
- [95] Installer NetBeans IDE 7.3.1 sous Debian 7 : [En ligne] Available at <http://blog.erlem.fr/divers/27-linux/50-installer-netbeans-ide-7-3-1-sous-debian-7>.
- [96] <https://javaee.github.io/glassfish> [dernière consultation 04/09/2017].
- [97] <http://www.oracle.com/technetwork/java/javaee/overview/index.html> . [dernière consultation 04/09/2017].
- [98] PierreYves Gibello, Web Services SOAP / WSDL JAXWS / JAXB / SAAJ, Janvier 2009.
- [99] OBERLE KARSTEN ET FISHER MIKE: ETSI CLOUD–initial standardization requirements for cloud services. In *Economics of Grids, Clouds, Systems et Services*, pages 105–115. Springer, 2010.
- [100] PETCU DANA, MACARIU GEORGIANA, PANICA SILVIU ET CRCIUN CIPRIAN: Portable cloud applications-from theory to practice. *Future Gener. Comput. Syst.*, 29(6):1417–1430, août 2013.
- [101] YANGUI SAMI, MARSHALL IAIN-JAMES, LAISNE JEAN-PIERRE ET TATA SAMIR : CompatibleOne : The Open Source Cloud Broker. *Journal of Grid Computing*, pages 1–17, 2013.
- [102] ANEDDA PAOLO, LEO SIMONE, MANCA SIMONE, GAGGERO MASSIMO ET ZANETTI GIANLUIGI: Suspending, Migrating and Resuming HPC virtual clusters. *Future Generation Computer Systems*, 26(8):1063–1072, 2010. 55
- [103] ROCHWERGER BENNY, BREITGAND DAVID, LEVY ELIEZER, GALIS ALEX, NAGIN KENNETH, LLORENTE IGNACIO MARTÍN, MONTERO RUBÉN, WOLFSTHAL YARON, ELMROTH ERIK ET CACERES JUAN : The reservoir model and architecture for open federated cloud computing. *IBM Journal of Research and Development*, 53(4):4–1, 2009.
- [104] EDMONDS AND Y, METSCH THIJS, PAPASPYROU ALEXANDER ET RICHARDSON ALEXIS: Toward an open cloud standard. *Internet Computing*, IEEE, 16(4):15–25, 2012.

