



REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE  
MINISTRE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE SCIENTIFIQUE

UNIVERSITE IBN KHALDOUN - TIARET

# MÉMOIRE

Présenté à :

FACULTÉ MATHÉMATIQUES ET INFORMATIQUE DÉPARTEMENT  
D'INFORMATIQUE

Pour l'obtention du diplôme de :

## MASTER

Spécialité : Réseaux et Télécommunications

Par : **GRIFID Athmane**

Sur le thème

---

**Approche orientée règle de modélisation et d'exécution des  
processus métier flexibles sur Cloud**

---

Soutenu publiquement le 07/ 11/2020 à Tiaret devant le jury composé de :

<b>Mr. MEGHAZI Hadj Madani</b>	Grade MAA	Président
<b>Mr. BEKKI Khadhir</b>	Grade MAA	Encadreur
<b>M. BELKACEMI Nawal</b>	Grade MCB	Examineur

**Année universitaire 2019 – 2020**

## Résumé

Le Cloud Computing est de plus en plus reconnu comme une nouvelle façon d'utiliser, à la demande, les services de calcul, de stockage et de réseau de manière transparente et efficace sur des infrastructures distribuées et hétérogènes. L'évolution et l'accroissement actuels des technologies amènent les entreprises à vouloir se développer plus rapidement afin de rester compétitives et offrir des services à la pointe de la technologie, répondant aux besoins. En effet, les processus offerts étant sujettes à des changements assez fréquents requièrent un haut niveau de flexibilité et d'agilité. Nous nous intéressons à la modélisation et la gestion du changement des processus métier dans le nuage. Nous avons adopté une approche orientée règle pour arriver à cet objectif. L'étude de cas était sur l'implémentation des scénarios d'ordonnancement sur le simulateur CloudSim en se basant sur cette approche

**Mot clé :** Modélisation, vérification, processus métier, flexibilité, cloud Computing, CloudSim.

## Abstract

Cloud computing is increasingly recognized as a new way to use, on demand, computing, storage and network services in a transparent and efficient way on distributed and heterogeneous infrastructures. The current evolution and increase in technology is leading companies to want to grow faster in order to remain competitive and to offer cutting-edge services that meet needs. Indeed, the processes offered being subject to quite frequent changes require a high level of flexibility and agility. We are interested in modeling and managing business process change in the cloud. We have taken a rule-oriented approach to achieve this goal. The case study was on the implementation of scheduling scenarios on the CloudSim simulator based on this approach.

**Keyword:** Modeling, verification, business process, flexibility, cloud computing, CloudSim.

## ملخص

تم التعرف على الحوسبة السحابية بشكل متزايد كطريقة جديدة لاستخدام خدمات الحوسبة والتخزين والشبكات عند الطلب بطريقة شفافة وفعالة على البنى التحتية الموزعة وغير المتجانسة. يقود التطور الحالي والزيادة في التكنولوجيا الشركات إلى الرغبة في النمو بشكل أسرع من أجل الحفاظ على قدرتها التنافسية وتقديم خدمات متطورة تلبي الاحتياجات. في الواقع، تتطلب العمليات المعروضة التي تخضع لتغييرات متكررة مستوى عالٍ من المرونة وخفة الحركة. نحن مهتمون بنمذجة وإدارة تغيير إجراءات العمل في السحابة. لقد اتخذنا نهجًا موجّهًا نحو القواعد لتحقيق هذا الهدف. كانت دراسة الحالة حول تنفيذ سيناريوهات الجدولة على محاكي CloudSim بناءً على هذا النهج.

**الكلمة الرئيسية:** النمذجة، التحقق، العمليات التجارية، المرونة، الحوسبة السحابية، CloudSim

# Table des matières

Introduction Générale.....	1
1 Contexte & Motivation.....	1
2 Problématique et Objectif.....	1
3 Solution proposée .....	<b>Erreur ! Signet non défini.</b>
4 Organisation de mémoire.....	1
<b>CHAPITRE I : LA MODÉLISATION ET LA VÉRIFICATION DES PROCESSUS MÉTIERS .....</b>	<b>3</b>
Introduction .....	4
I. Notions du Processus.....	4
1. Processus .....	4
2. Processus métiers.....	4
2.1 Introduction .....	4
2.2. Définitions .....	5
2.3 Processus métier flexible.....	5
2.4 Différents aspects de processus métier.....	5
3. Modélisation des processus métiers .....	6
3.1. Objectifs de la modélisation .....	6
3.2. Caractéristique de la modélisation.....	7
3.3. Obstacles.....	10
3.4. Les caractéristiques de modélisation .....	10
3.5. Classification des approches de modélisation .....	10
3.5.1. Modélisation basée sur les règles .....	11
3.5.2. Modèle basée sur les graphes .....	12
3.5.3. Comparaison entre les modèles basés sur les graphes et les modèles basés sur les règles.....	12
3.6. Techniques de modélisation des processus métiers.....	13
3.6.1. IDEF .....	14
3.6.2. Réseaux de pétri.....	14
3.6.3. GRAI .....	14
3.6.4. PERA.....	14
3.6.5. CIMOSA.....	15
3.6.6. ARIS .....	15
3.6.7. GERAM.....	15
3.6.8. BPMN.....	15
3.6.9. UML .....	16
3.6.10. Les travaux de modélisation des processus basés sur les règles.....	16

4. Vérification des processus métiers .....	18
4.1. Les techniques de vérification des modèles des processus métier.....	18
4.1.1. La vérification par guides style de modélisation .....	18
4.1.2. La vérification par la simulation.....	19
4.1.3. La vérification par modèles formels.....	19
4.1.4. La vérification par process_mining .....	19
4.2. Synthèse.....	19
4.3. Les erreurs et exceptions d'un processus métier .....	20
4.3.1. Les erreurs fonctionnelles.....	20
4.3.2. Les erreurs opérationnelles.....	20
4.3.3. Les erreurs non fonctionnelles.....	20
5. Conclusion.....	21
<b>CHAPITRE II : CLOUD COMPUTING.....</b>	<b>22</b>
1. Introduction.....	23
2. Définition .....	23
3. Les bases informatiques du Cloud Computing .....	24
3.1. Infrastructure.....	24
3.2. La Virtualisation.....	24
3.3. Interfaces de service.....	25
4. Les modèles de services de cloud computing .....	25
4.1. SaaS (Software as a Service) .....	26
4.2. PaaS (Platform as a Service) .....	26
4.3. IaaS (Infrastructure as a Service) .....	26
5. Les modèles de déploiement du Cloud .....	27
5.1. Cloud privé.....	27
5.2. Cloud communautaire .....	28
5.3. Cloud public.....	28
5.4. Cloud hybride .....	29
6. Les cinq caractéristiques du Cloud Computing .....	29
6.1. Accès réseau universel .....	30
6.2. Mise en commun de ressources.....	30
6.3. Elasticité.....	30
6.4. Libre Service .....	30
6.5. Service mesurable et facturable.....	31
7. La sécurité du Cloud .....	31

7.1.	La confidentialité .....	31
7.2.	L'intégrité.....	32
7.3.	La Disponibilité .....	32
8.	Acteurs du Cloud.....	33
9.	Les avantages et les inconvénients du Cloud Computing .....	33
9.1.	Les avantages du Cloud Computing .....	33
9.2.	Les inconvénients de Cloud Computing.....	34
10.	Les principaux fournisseurs actuels de Cloud Computing .....	34
10.1.	Amazon Web Services (IaaS).....	34
10.2.	Amazon Web Services (PaaS).....	35
10.3.	Microsoft Azure (PaaS).....	35
10.4.	Google AppEngine (PaaS).....	35
10.5.	Google Apps (SaaS) .....	35
11.	Conclusion.....	35
<b>CHAPITRE III : DESCRIPTION ET IMPLÉMENTATION DE NOTRE APPROCHE.....</b>		<b>36</b>
1	Introduction .....	37
2	Définition du problème.....	37
3	Démarche détaillé de réalisation de l'exemple.....	37
3.1	Formalisation du problème .....	38
3.2	Modélisation de notre processus métier par le paradigme ECA.....	39
3.2.1	Règle ECA.....	39
3.2.2	Adaptation de règles ECA .....	39
3.2.3	Configuration des services.....	41
3.2.4	Algorithmes d'ordonnancement .....	42
3.3	Intégration de notre processus dans le Cloud .....	43
4.	Implémentation de notre système .....	44
4.1.	Outils de développement .....	44
4.1.1.	Le langage de programmation JAVA.....	44
4.1.2.	NetBeans.....	44
4.1.3	Pourquoi la simulation ?.....	45
4.1.4	CloudSim.....	45
4.2	La structure du système .....	47
4.3	La description des classes d'application.....	48
4.4	Interfaces de l'application .....	48
5.	Conclusion.....	53

<b>CONCLUSION GÉNÉRALE</b> .....	54
Conclusion générale .....	55
Bibliographie .....	56

## *Liste des figures*

<b>Figure I.1:</b> Différents aspects d'un processus métier.....	6
<b>Figure I.2:</b> Le patron de séquence.....	7
<b>Figure I.3:</b> Le patron du choix.....	7
<b>Figure I.4:</b> Le patron du Boucle.....	8
<b>Figure I.5:</b> Le patron de synchronisation.....	8
<b>Figure I.6:</b> Le patron de parallélisme.....	9
<b>Figure I.7:</b> Le patron de l'Agrégation.....	9
<b>Figure I.8:</b> les approches de modélisation.....	11
<b>Figure II.1 :</b> L'environnement de Cloud Computing.....	24
<b>Figure II.2:</b> Les modèles de services de cloud computing.....	26
<b>Figure II.3:</b> Les modèles de déploiement du Cloud.....	27
<b>Figure II.4 :</b> cloud privé.....	28
<b>Figure II.5 :</b> cloud communautaire.....	28
<b>Figure II.6 :</b> cloud public.....	29
<b>Figure II.7 :</b> cloud hybride.....	29
<b>Figure II.8:</b> Les cinq caractéristiques du Cloud Computing.....	30
<b>Figure II.9:</b> La Sécurité du Cloud Computing.....	31
<b>Figure III.1:</b> Illustration de notre futur système.....	37
<b>Figure III.2:</b> montre l'architecteur globale de notre solution.....	38
<b>Figure III.3:</b> Formalisation du problème.....	39
<b>Figure III.4:</b> Adaptation de règles ECA.....	40
<b>Figure III.5:</b> Tables des Règles ECA.....	41
<b>Figure III.6:</b> Montre le mécanisme de la configuration de ce service.....	42
<b>Figure III.7:</b> illustre le mécanisme détaillé de ce processus d'intégration.....	44
<b>Figure III.8:</b> Architecture générale de CloudSim.....	46
<b>Figure III.9:</b> Les classes du CloudSim.....	47
<b>Figure III.10:</b> Aspet structurel de notre système.....	47
<b>Figure III.11:</b> La fenêtre principale de notre application.....	49
<b>Figure III.12:</b> Liste des Règles ECA.....	49
<b>Figure III.13:</b> Les Règles exécuter.....	50
<b>Figure III.14:</b> L'interface globale.....	50
<b>Figure III.15:</b> Listes des tâches et des VMs.....	51

<b>Figure III.16:</b> Les simulations des politiques.....	51
<b>Figure III.17:</b> Résultat de la politique Min-Min.....	52
<b>Figure III.18:</b> Moyenne de chaque politique.....	52

## *Liste des tableaux*

<b>Tableau I.1:</b> Comparaison entre Langages basés sur les graphes et Langages basés sur les règles...	13
<b>Tableau III.1</b> synthèse de ECA.....	40
<b>Tableau III.2:</b> Description des classes de notre application.....	48

# INTRODUCTION GÉNÉRALE

## Introduction Générale

### 1 Contexte & Motivation

Le Cloud Computing est la technologie émergente dans l'industrie informatique qui repose sur le partage des ressources sur un réseau. Pour optimiser les avantages du Cloud Computing, les développeurs doivent concevoir des mécanismes qui optimisent l'utilisation des modèles architecturaux et de déploiement.

Les utilisateurs de Cloud progressent de jour en jour, ce qui fait que le nombre d'utilisateurs augmente et que cela devient un sérieux problème pour les développeurs de programmer les composants et de planifier les tâches dans le Cloud. Durant les dernières années, la notion de "processus" s'est largement installée car on s'est rendu compte que pour être conforme avec la nouvelle philosophie des affaires, l'organisation doit être conçue pour fournir le flot vertical et horizontal des informations nécessaires à la réalisation des objectifs globaux de l'organisation. L'approche processus consiste à décrire de façon méthodique une organisation ou un ensemble d'activités en processus, de façon à organiser sa contribution à la satisfaction des clients. Il existe plusieurs types de processus (de pilotage, métier et de support).

Ce qui nous intéresse dans le cadre de cette mémoire ce sont les processus métiers leur modélisation et leur flexibilité sur Cloud

### 2 Problématique et Objectif

Plusieurs raisons expliquent l'intérêt accru accordé au processus métier. En fait, les organisations, de nos jours, devraient pouvoir fournir un large éventail de services et de produits. Par conséquent, le nombre de processus au sein des organisations a augmenté, et donc le nombre de produits et de services a également augmenté. [1]

En conséquence, les processus métier actuels sont également soumis à de fréquents changements. En outre, la complexité de ces processus a considérablement augmenté. Plusieurs modèles de processus métiers sont proposés dans la littérature. Or, il y a un manque apparent de moyens de modélisation, analyse et vérification des processus métiers flexibles. Le besoin pour un nouveau composant, pour la modélisation et la vérification de ces processus paraît donc évident.

### 3 Organisation de mémoire

Notre mémoire est organisé en trois chapitres :

**Chapitre 1** : présente une étude générale sur la modélisation et la vérification des processus métiers. et la prise en compte de la flexibilité et du changement dans les ces processus

**Chapitre 2 :** présenter quelques notions fondamentales du Virtualisation et Cloud Computing, ses principes, ses objectifs et ses types.

**Chapitre 3 :** décrit notre approche et son implémentation sur un simulateur CloudSim.

Notre mémoire est conclu par une conclusion générale avec quelques perspectives.

*CHAPITRE I : LA MODÉLISATION ET LA  
VÉRIFICATION DES PROCESSUS MÉTIERS*

## Introduction

Dans ce chapitre, nous rappelons les notions des processus métiers, ainsi nous présentons leur modélisation et vérification.

## I. Notions du Processus

### 1. Processus

- **Une définition utile dit** : "Un processus est un ensemble partiellement ordonné d'étapes exécutées en vue de réaliser au moins un objectif " [2]
  - **La définition officielle de la norme ISO** : (2000) [ISO00] voit le processus comme : « Un ensemble d'activités corrélées ou interactives qui transforment des éléments d'entrée en éléments de sortie". [2]
  - **la structure d'un processus** : « Un processus représente l'organisation d'un ensemble finalisé d'activités effectuées par des acteurs et mettant en jeu des entités".[2]
  - **Une définition plus complète** "Un processus est défini comme un enchaînement partiellement ordonné d'exécution d'activités qui, à l'aide de moyens techniques et humains, transforme des éléments d'entrée en éléments de sortie en vue de réaliser un objectif dans le cadre d'une stratégie donnée".[2]
- On peut donner la définition générale suivante, « Un processus est un ensemble d'activités, entreprises dans un objectif déterminé. La responsabilité d'exécution de tout ou partie des activités par un acteur correspond à un rôle. Le déroulement du processus utilise des ressources et peut-être conditionné par des événements, d'origine interne ou externe. L'agencement des activités correspond à la structure du processus ». [3]

## 2. Processus métiers

### 2.1 Introduction

Un processus est une série de tâches complétées pour atteindre un objectif. Un processus d'entreprise est donc un processus axé sur la réalisation d'un objectif pour une entreprise. Tout d'un processus simple pour faire un sandwich, la construction d'une navette spatiale utilise un ou plusieurs processus métiers. Les processus sont quelques choses que les entreprises passent tous les jours afin d'accomplir leur mission. Le meilleur de leurs processus est le plus efficace de l'entreprise. Certaines entreprises voient leurs processus comme une stratégie pour obtenir un avantage concurrentiel. Un processus qui atteint son objectif d'une manière unique peut distinguer une entreprise. Un processus qui élimine les coûts peut permettre à une entreprise de baisser ses prix (ou de conserver plus de profit). [4]

## 2.2. Définitions

**Définition 1 :** Un processus métier consiste en un ensemble d'activités exécutées en coordination dans un environnement organisationnel et technique. Ces activités réalisent conjointement un objectif d'affaires. Chaque processus métier est mis en œuvre par une seule organisation, mais il peut interagir avec les processus métiers exécutés par d'autres organisations. La gestion des processus métiers couvre non seulement la représentation des processus métiers, mais également des activités supplémentaires. [5]

**Définition 2 :** La gestion des processus métiers comprend des concepts, des méthodes et des techniques pour prendre en charge la conception, l'administration, la configuration, la mise en œuvre et l'analyse des processus métiers [5]

**Définition 3 :** Un système de gestion des processus métiers est un système logiciel générique piloté par des représentations de processus explicites pour coordonner la mise en œuvre des processus métiers. [5]

## 2.3 Processus métier flexible

Un processus métier est considéré flexible s'il est possible de le modifier sans le remplacer dans sa totalité. La flexibilité d'un processus reflète son habilité à couvrir les changements en variant ou en adaptant les parties du processus métier qui sont affectées tout en retenant le format essentiel des parties non affectées par le changement. En conclusion, il est possible de définir la flexibilité par la capacité à supporter et à prendre en charge les changements métier. Un processus métier est flexible s'il est capable d'atteindre ses objectifs en dépit des variations et des stimulis. Lorsqu'une variation apparaît entre la définition et le déroulement actuel du processus, des moyens sont requis pour assurer que le processus évolue en accord avec ses attentes. [6]

## 2.4 Différents aspects de processus métier

Dans une entreprise, un processus métier regroupe plusieurs aspects de base qui peuvent être modélisés et traités indépendamment (*Figure 1.1*). Ces aspects ont inspiré la taxonomie de la flexibilité des processus métier que nous décrirons dans les sections suivantes :

- Aspect Organisationnel : Il décrit la structure organisationnelle de l'entreprise (départements, services, etc.) qui est invoquée pour la réalisation du processus métier à travers les concepts de rôle, de groupe de rôles, d'acteur (ou d'agent), d'équipe, de compétences de chaque membre de l'équipe, de position dans l'équipe ou de responsabilité dans la structure.
- Aspect Informationnel : Il décrit les informations (données, documents, etc.) qui sont manipulées par l'utilisateur (ou acteur) du processus métier pour exécuter l'activité ou le processus.
- Aspect Comportemental : Il correspond à la modélisation de la dynamique du processus métier, c'est à dire, la manière dont les activités sont chronologiquement exécutées et les conditions les déclenchant.
- Aspect Opérationnel : Il décrit l'ensemble des activités participant à un processus métier.
- Aspect Fonctionnel : Il décrit l'objectif à atteindre par le processus métier.[6]

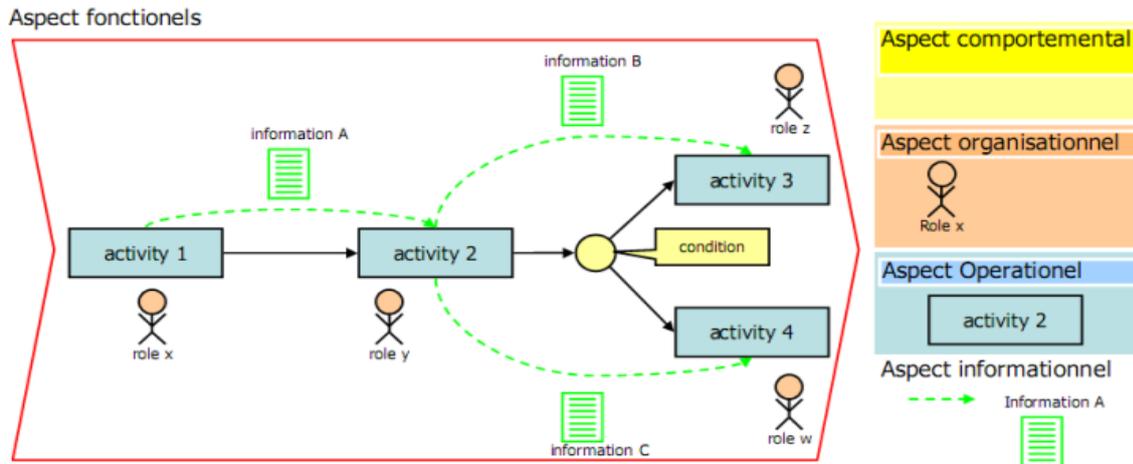


Figure I.1 : Différents aspects d'un processus métier.

### 3. Modélisation des processus métiers

Un modèle de processus métier consiste en un ensemble de modèles d'activités et de contraintes d'exécution entre eux. Un exemple de processus métier représente un cas concret dans le processus opérationnel d'une compagnie, consistant en des instances d'activités. Chaque modèle de processus métier opère comme un plan pour un ensemble d'instances du processus métier, et chaque modèle d'activité opère comme un plan pour un ensemble d'instances de l'activité. Si aucune confusion n'est possible, le terme processus métier est utilisé pour se reporter à aussi bien au modèle de processus métier qu'à une instance de processus métier. D'une manière analogue, l'activité est utilisée pour faire référence au modèle de l'activité ou à une instance de l'activité. Les modèles du processus métier sont les objets principaux pour mettre en œuvre ces processus métier. Cette mise en œuvre peut être faite par des règles d'organisation et des politiques, elle peut aussi être faite par un système logiciel en utilisant un système de gestion de processus métier. Dans ce cas, ce système logiciel est conduit par des représentations explicites du processus. Puisque par définition les processus métier sont exécutés dans une seule organisation, le cheminement d'activités peut être contrôlé par un système de gestion de processus métier comme un composant logiciel centralisé exécuté par le revendeur de la compagnie. Ce contrôle centralisé est très semblable à un conducteur qui contrôle, centralement, les musiciens dans un orchestre ; par conséquent, les processus métier sont aussi appelés des orchestrations de processus. [1]

#### 3.1. Objectifs de la modélisation

La modélisation des processus a beaucoup d'avantages pour les entreprises qui cherchent à améliorer leurs performances. On peut citer :

- Faciliter la communication en utilisant un langage commun.
- Permettre une meilleure compréhension de l'existant.
- Fournir une documentation du processus métier.
- Améliorer la situation actuelle.
- Expérimenter et simuler de nouvelles situations et de nouveaux concepts et leurs impacts sur l'organisation.

- Automatiser le processus, ... [2]

### 3.2. Caractéristique de la modélisation

L'ensemble de critères pour évaluer les modèles des processus métiers

- **Séquencement** : Le déclenchement d'une activité après la terminaison d'une autre activité du même processus. La figure I.2 illustre une séquence d'activités où l'activité A se déclenche en premier et après sa terminaison, l'activité B et l'activité C se déclenchent successivement.[7]

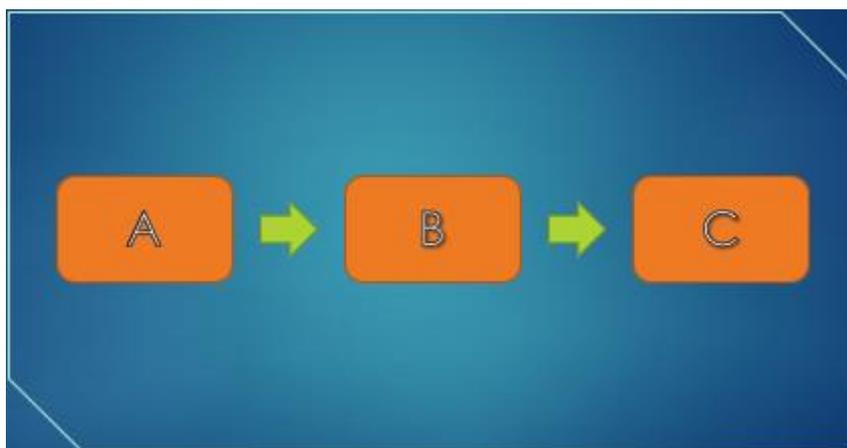


Figure I.2 : Le patron de séquence.

- **Choix** : Le choix d'une seule branche parmi plusieurs pour permettre à une activité d'être exécutée en fonction d'une condition. La figure I.3 illustre le choix (nœud de décision) où soit l'activité B soit l'activité C s'exécute en fonction d'une certaine condition. [7]

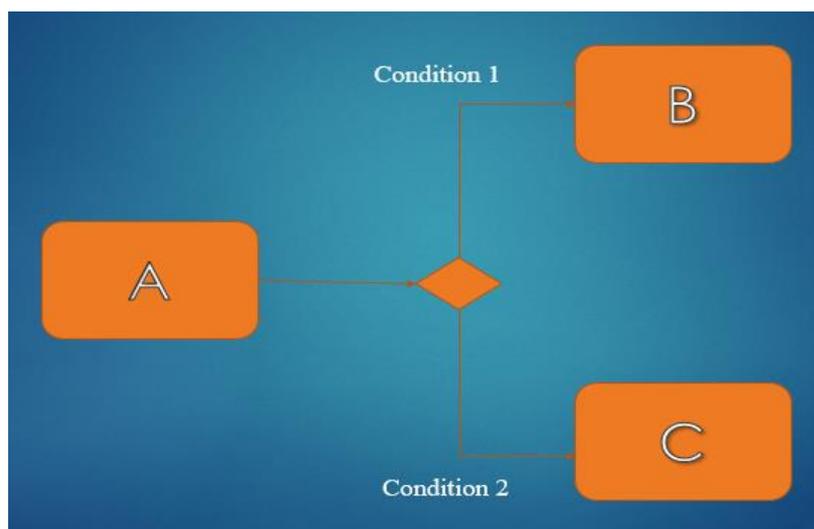


Figure I.3 : Le patron du choix.

- **Boucle** : le déclenchement d'un ensemble d'activités d'une façon à ce qu'elle forme une boucle (exemple « B, C, D »).[7]

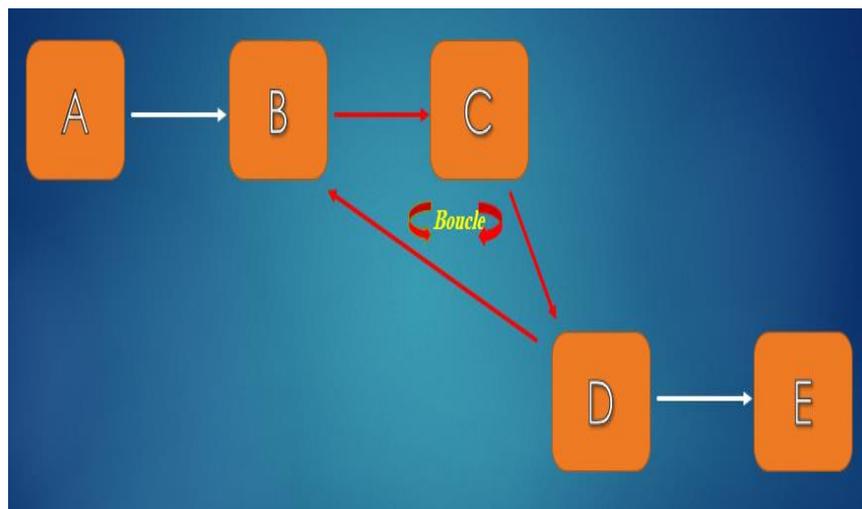


Figure I. 4 : *Le patron du Boucle.*

- **La synchronisation :** La convergence de plusieurs branches d'un processus en une seule branche pour permettre la synchronisation entre les activités. La figure *I.5 illustre* une synchronisation entre deux activités où l'activité C ne se déclenche que si l'activité A et l'activité B se terminent.[7]

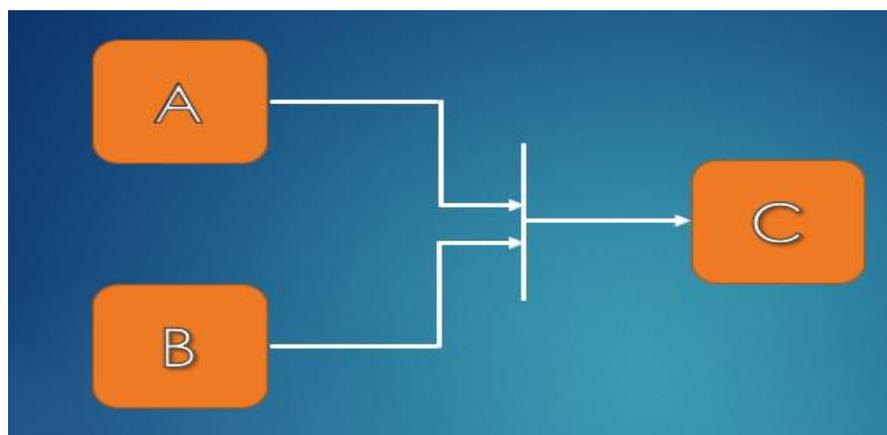
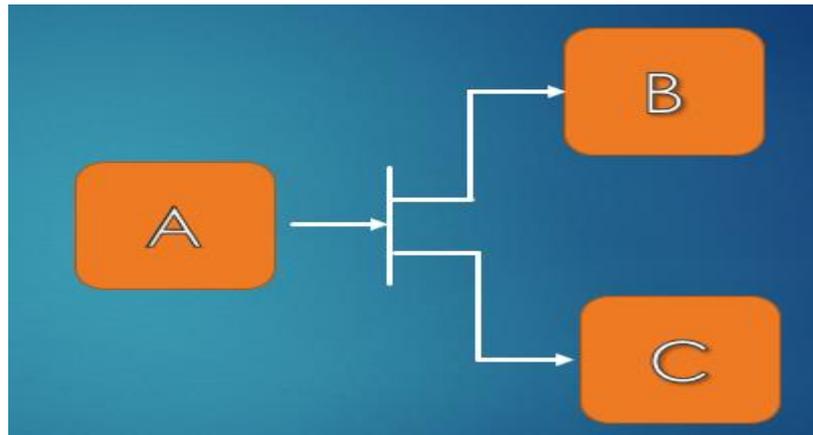


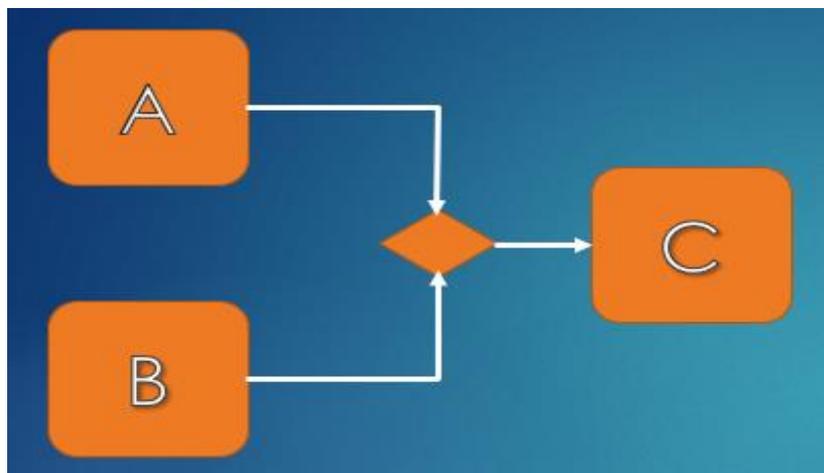
Figure I.5 : *Le patron de synchronisation.*

- **La branche parallèle :** La divergence d'une branche d'un processus en plusieurs branches pour permettre aux activités d'être exécutées d'une manière simultanée (une concurrence d'exécution). La figure *I.6 illustre* deux activités qui s'exécutent en parallèle (l'activité B s'exécute en même temps que l'activité C).[7]



*Figure I.6 : Le patron de parallélisme.*

- **Date limite (en : Deadline) :** peut-être spécifiée en termes de limite de temps absolu quand une activité doit débiter ou finir durant l'exécution du processus. [8]
- **Agrégation :** (La jonction simple) La jointure de plusieurs branches sans synchronisation pour permettre à une instance d'un processus, lors de son exécution, de ne passer qu'une fois par une jonction unique. La figure I.7 illustre la jointure de deux activités (nœud de fusion) où le rassemblement de deux flots alternatifs entrants en un seul flot sortant.[7]



*Figure I.7 : Le patron de l'Agrégation.*

- Doivent avoir un moyen de distinguer les rôles et de les affecter aux différentes tâches.
- Doivent avoir une représentation graphique non ambiguë du langage.
- Doivent avoir un modèle de transaction qui permet la description du comment "le processus peut être non accompli".
- Doivent spécifier comment les instances du processus vont être déclenchées et identifiées durant leur exécution.
- Doivent avoir un moyen de spécifier les caractéristiques du PM qui peuvent intéresser les utilisateurs externes, telles que la qualité de service et le prix.
- Le langage ne doit pas s'embrouiller dans les détails des protocoles de communication.

Les trois premiers critères sont importants pour obtenir les cinq premiers avantages cités plus haut. Les trois suivants sont essentiels pour automatiser l'exécution des processus entre les organisations séparées et qui collaborent entre elles. Le dernier critère garde le modèle dans le niveau d'abstraction adéquat.[2]

### 3.3. Obstacles

Bien que la modélisation des processus métiers ait beaucoup d'avantages, elle l'est rarement faite. Et si c'est le cas, elle ne l'est pas bien faite. Ceci est dû à la nature complexe et floue des processus métiers qui les différencie des autres projets d'ingénieurs qui se font de manière bien structurée. L'architecture, qui est le haut niveau de la conception fonctionnelle des processus métiers est plus un art qu'une science. Cette complexité extrême des processus métiers est due à plusieurs raisons :

- Ils requièrent plusieurs domaines de connaissance.
- Ils opèrent dans des échelles de temps " largement différentes".
- Ils sont souvent indépendants.
- Les gens ont besoin d'années d'entraînement pour les comprendre ou comprendre "leur raison d'être"
- Ils ont beaucoup " de modification non- contrôlées".[2]

### 3.4. Les caractéristiques de modélisation

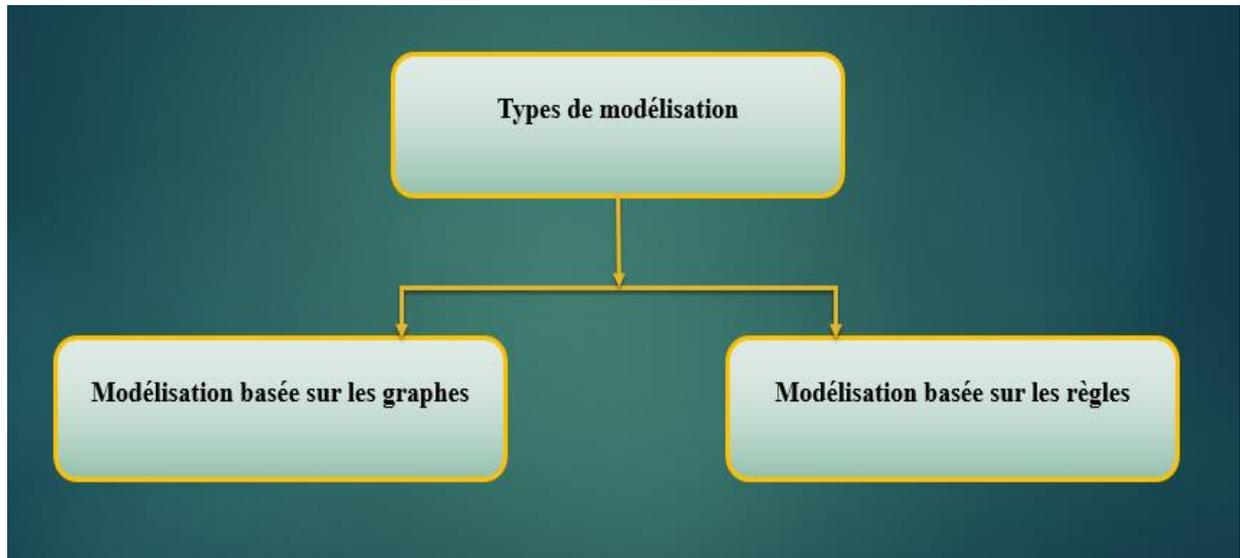
Plusieurs caractéristiques doivent être prises en compte lors de la modélisation de processus métier, parmi [8] :

- ❖ **L'expressivité** : la description complète des différents éléments d'un processus.
- ❖ **La réutilisabilité** : l'habilité à partager des parties du modèle pour les réutiliser.
- ❖ **La complexité** : la difficulté de modéliser et de vérifier un processus.
- ❖ **La flexibilité** : la capacité d'implémenter le changement d'une partie du processus en gardant les autres parties stables.
- ❖ **L'adaptabilité** : l'habilité à réagir aux exceptions aléatoires

### 3.5. Classification des approches de modélisation

Le regroupement des informations collectées reflète l'organisation et son mode de fonctionnement. Le nombre important d'intervenants, de tâches, d'interactions et d'échange de message rendent éventuellement difficile la compréhension au premier regard des informations collectées. La maîtrise des processus métiers nécessite une simplification de cette réalité en structurant l'information sous forme de modèles employés dans toutes les étapes du cycle de vie des processus métiers. Elle a réalisé une étude de R. Lu et S. Sadiq Les approches de

modélisation des processus métier sont classées en deux catégories : les modèles basés sur des règles et les modèles basés sur des graphiques. [8]



*Figure I.8 : les approches de modélisation.*

### 3.5.1. Modélisation basée sur les règles

Dans une approche classique basée sur des règles, la logique du processus est résumée en ensemble de règles, dont chacune est associée à une ou plusieurs activités commerciales, en spécifiant les propriétés de l'activité tel que les pré- et post conditions de l'exécution.

Une règle métier est une déclaration qui vise à influencer ou de guider le comportement de l'information dans une organisation. Les règles métier peuvent être classées en fonction de leurs sources ou leurs structures :

- Des politiques publiées qui doivent être suivies, où se sont les conséquences. Par exemple : le paiement des taxes et le respect de la loi.
- Politiques, normes publiées qui doivent être suivies et adhérent à un comportement d'entreprise acceptable. Les exemples sont les budgets ou les énoncés de mission
- Lignes directrices, les règles qui peuvent ou ne peuvent pas s'appliquer, en fonction des circonstances. Les exemples sont les méthodes et les styles de gestion.

Les différentes catégories structurelles des règles sont :

- Intégrité (ou contraintes) ; Par exemple : chaque projet doit avoir un et un seul gestionnaire de projet.
- Dérivation (conditions conduisant à des conclusions) ; Par exemple : les clients bénéficient d'une réduction de platine à 5%. John Doe est un client de platine. En guise de conclusion, John Doe reçoit un rabais de 5%.
- Réaction (événement, condition, action, Alternative action, post-condition) ; Par exemple : une facture est reçue. Si le montant de la facture est plus de 1000 \$, alors un superviseur doit l'approuver.
- Production (condition, action) ; Par exemple : s'il n'y a pas des défauts dans le dernier lot de voitures alors le lot est approuvé.
- Transformation (changement d'état) ; par exemple : l'âge de l'employé peut être changé de 30 à 31, mais non de 31 à 30.

Les Langages de modélisation des règles métiers sont généralement fondés sur la logique formelle et ont une forte et précise puissance expressive.

La représentation des règles métiers complète les enchaînements des tâches décrits dans les diagrammes des processus métiers. Les règles métiers correspondent à la détermination d'un choix basé sur un critère. Lorsque le jeton se présente à l'entrée de la règle métier, une de ses caractéristiques est vérifiée, provoquant le choix d'une des sorties à prendre.

Le formalisme Event-Condition-Action (ECA) utilisé dans les systèmes de base de données active dans les années 1990, a été adopté par de nombreux langages de modélisation basés sur les règles. Ceci étant, d'autres travaux qui utilisent d'autres formalismes de règles. Un exemple de ces travaux est le travail de Goedertier et al. Dans [13] qui proposent le langage EM-BrA<sup>2</sup>CE basé sur le standard SBVR pour modéliser, d'une manière déclarative, les processus métier. En effet, le SBVR est un méta-modèle, proposé par l'OMG permettant une construction d'un vocabulaire métier (Les concepts et termes) à utiliser dans la définition des règles métier.[8]

### 3.5.2. Modèle basée sur les graphes

Dans un langage de modélisation basé sur les graphes, la définition des processus est spécifiée dans les modèles de processus graphiques.

Il existe différentes façons de représenter un même sujet. Chaque perspective représente une vue particulière de l'organisation dépendant de son utilisation. Il existe de nombreux diagrammes selon les perspectives retenues et leur choix dépend de plusieurs critères

- La méthodologie d'élaboration de la cartographie et la modélisation.
- L'outil de modélisation.
- Les parties prenantes ou les intervenants du processus métier.
- Les moyens de communication des informations.
- La spécification de la notation employée précisant le format ainsi que la sémantique des éléments du diagramme.

Il n'existe pas de règles absolues de représentation et de modélisation des processus métiers de l'organisation. Cependant, il existe plusieurs diagrammes et modèles fondamentaux illustrant la situation courante d'une organisation

- Le diagramme de processus représentant les séquences logiques et chronologiques des tâches des processus métiers.
- Le diagramme de la chaîne de valeur représente une perspective macroscopique de la création de valeur dans l'organisation.
- L'organigramme représente la structure organisationnelle hiérarchisée de l'entreprise.

La plupart des langages basée sur les graphes ont leur racine dans la théorie du réseau de Pétri, qui a été appliqué pour la première fois dans la modélisation de workflow. Le point fort de Pétri est qu'il inclut la sémantique formelle malgré sa nature est graphique.

Un Workflow (terme anglais) est un flux d'informations au sein d'une organisation, un processus *workflow* est une représentation d'un processus métier dans un format interprétable par la machine. La structure d'un processus est décrite en termes de tâches et d'ordre d'exécution entre ces tâches c'est-à-dire. Des dépendances logiques ou temporelles entre ces tâches (le flot de contrôle).[8]

### 3.5.3. Comparaison entre les modèles basés sur les graphes et les modèles basés sur les règles

Une comparaison entre ces deux formalismes, le tableau ci-dessous résume cette comparaison.[8]

Critères	Langages basés sur les graphes	Langages basés sur les règles
<b>Expressibilité :</b> c'est la description complète des différents éléments d'un processus.	<ul style="list-style-type: none"> <li>• Capable d'exprimer la structure, les données et les exigences d'exécution</li> <li>• Les modèles de workflow peuvent être exprimés par le représentant à base de graphique.</li> </ul>	<ul style="list-style-type: none"> <li>• Capable d'exprimer la structure, les données, l'exécution, aussi bien que les exigences temporelles.</li> <li>• Les expressions de règle peuvent représenter de plus des modèles de workflow que les langues à base de graphique.</li> </ul>
<b>Flexibilité :</b> C'est la capacité d'implémenter les changements d'une partie du processus en gardant les autres parties stables.	Les processus peuvent seulement être exécutés sur les modèles de processus complets, dans lesquels tous les scénarios d'exécution possibles sont explicitement spécifiés. Les conditions pour chaque scénario doivent aussi être articulées dans le modèle du processus a priori.	<b>Plus flexible</b> comme la spécification incomplète pour la dépendance de tâche est soutenue.
<b>L'adaptabilité :</b> l'habilité à réagir aux exceptions aléatoires.	Les exceptions montent si un peu de conduite opératoire s'est produite qui n'a pas été défini dans le modèle de processus. L'exception Handling doit être définie par un ensemble supplémentaire des politiques (les règles).	Les exceptions prévues peuvent être manipulées en spécifiant des règles supplémentaires en plus que pour exprimer la logique de processus régulière.
<b>Complexité :</b> la difficulté de modéliser et de vérifier un processus.	<ul style="list-style-type: none"> <li>• Langages de modélisation ont une syntaxe plus abstraite et sémantique simple, donc moins de complexité dans la représentation du modèle et de vérification.</li> <li>• Le soutien d'exécution pour les processus dynamiques est plus complexe que la modélisation de constructions supplémentaires et l'effort de vérification est exigé.</li> </ul>	<ul style="list-style-type: none"> <li>• Modèle de langues ont une syntaxe logique et nécessite une certaine expertise dans la modélisation.</li> <li>• Les modèles de processus n'ont pas l'attrait visuel, processus de vérification est plus complexe.</li> <li>• Le Changement de logique de processus est réalisée par la règle de modification et donc moins complexes.</li> </ul>

**Tableau I.1 : Comparaison entre Langages basés sur les graphes et Langages basés sur les règles.**

### 3.6. Techniques de modélisation des processus métiers

Durant les dernières années, la modélisation des entreprises a été un domaine de recherche très actif. En réalité les gens modélisaient les PM depuis plusieurs années, même s'ils n'appelaient pas leurs modèles des " modèles de processus métiers". [2]

### 3.6.1. IDEF

La série IDEF (Icam DEFinition) a été développée dans les années 70 par l'US Air Force dans un projet nommé ICAM (Integrated Computer Aided Manufacturing). Elle regroupe cinq méthodes complémentaires : IDEF0/SADT, IDEF1-IDEF1x, IDEF2, IDEF3. Nous donnerons les grandes lignes de IDEF0, IDEF2, et IDEF3 [2].

### 3.6.2. Réseaux de pétri

Un réseau de Pétri est un outil graphique et mathématique pour la modélisation du comportement des systèmes dynamiques à événements discrets et la description des relations existantes entre des conditions et des événements. Les RDPs permettent d'analyser et de simuler des systèmes. En étant rôle d'outil graphique, il nous aide à comprendre facilement le système modélisé, en plus il nous permet de simuler les activités dynamiques et concurrentes. Avec l'outil mathématique, il nous permet d'analyser le système grâce aux modèles de graphes, aux équations algébriques, etc. Le RDP provient de la thèse de doctorat, réalisé en 1962, du mathématicien allemand Carl Adam Pétri.

Dans sa thèse, C.A. Pétri définit une nouvelle méthode d'étude de système qui se compose en 3 étapes :

- On écrit le système en termes de réseau et on obtient le modèle de réseau de Pétri.
- On analyse le modèle obtenu, on déduit les propriétés comme l'absence de blocages, conflits...etc.
- Finalement, on fait la révision des propriétés pour montrer si le système est correct.[10]

### 3.6.3. GRAI

La méthode GRAI (Graphe de Résultats et Activités Inter-reliés) a été développée par les Professeurs Lucas Pun et Guy Doumeingts de l'Université de Bordeaux [DOUMEINGTS 1984]. Elle est orientée pour la modélisation des systèmes de décision et est basée sur :

- Le modèle de référence (modèle GRAI), contenant les concepts de modélisation et leurs relations permettant de construire le modèle global de l'entreprise ;
- Les formalismes graphiques, représentant les concepts du modèle GRAI et facilitant la communication et l'interprétation : la grille GRAI et les réseaux GRAI ;
- Une démarche structurée et participative dans laquelle toutes les étapes et tous les acteurs de la méthode sont définis, permettant efficacité et gain de temps.[11]

### 3.6.4. PERA

PERA, (acronyme de Purdue Enterprise Reference Architecture) a été développée en 1990 par the Purdue Laboratory for Applied Industrial Control, université de Purdue, USA.

Son objectif est d'établir les bases pour le traitement des fonctions mises en œuvre par l'être humain dans le domaine de l'intégration des entreprises.

La description des tâches et des fonctions de l'entreprise est décomposée en deux grands courants:

- Le courant des informations.
- Le courant de la production.

Dans le modèle de PERA, les classes des fonctions concernant la décision, le contrôle et l'information sont regroupées dans un seul courant appelé fonction d'information.

- Pendant l'implémentation, les deux courants cités ci-dessus sont réaménagés dans trois jeux d'implémentation de tâches et de fonctions:

a- Les activités de l'être humain qui sont:

- Activités d'information.
- Activités de production.

b- Les activités d'information non effectuées par l'être humain.

c- Les activités de production non effectuées par l'être humain.

Différemment de CIMOSA qui définit quatre vues: fonctionnelle, informationnelle, ressources et organisationnelle, PERA se focalise sur seulement deux vues: Une vue fonctionnelle et une vue d'implémentation[2].

### 3.6.5. CIMOSA

CIMOSA (Computer Integrated Manufacturing Open System Architecture) a pour but la construction de systèmes intégrés de production. Le cadre est fondé sur la notion de cycle de vie d'un système et offre un langage de modélisation, une méthodologie et une technologie de soutien pour appuyer ces objectifs [11].

### 3.6.6. ARIS

ARIS (Architecture of Integrated Information Systems) a été développée dans les années 1990 par le professeur August-Wilhelm SCHEER. Elle a été proposée pour les phases d'analyse et de définition des besoins d'un système d'information de gestion [11].

### 3.6.7. GERAM

L'architecture de référence GERAM (acronyme de **G**eneric **E**nterprise **R**eference **A**rchitecture and methodology) a été introduite par le groupe de travail IFAC/IFIP Task Force on Architecture Enterprise Integration. Après avoir analysé les principales architectures de référence disponibles, à savoir CIMOSA, GRAI-GIM et PERA, le groupe a constaté qu'il fallait préserver le meilleur des méthodes de modélisation et architectures existantes pour créer une nouvelle architecture ayant les qualités de ses aînés sans leurs défauts [2].

### 3.6.8. BPMN

En 2000, un consortium d'entreprises impliquées dans le développement du commerce électronique, s'est donné pour objectif de définir un langage de description des processus métiers, qui puisse en traduire la complexité tout en restant accessible. Cela a donné lieu à un formalisme orienté activité, BPMN, en partie inspiré d'UML, et qui en 2005, a été adopté par l'OMG comme UML l'avait été quelques années auparavant.

BPMN est une notation, c'est-à-dire un ensemble de symboles permettant de représenter des processus métiers sous forme graphique. Par rapport aux langages antérieurs, on peut relever que le diagramme d'activités d'UML a été une source d'inspiration, mais BPMN a eu un apport majeur dans la représentation des différents échanges entre processus. Il a, en effet, été conçu pour pouvoir modéliser des processus privés (internes à une entreprise) comme des processus publics (qui impliquent deux ou plusieurs organisations).[3]

### 3.6.9. UML

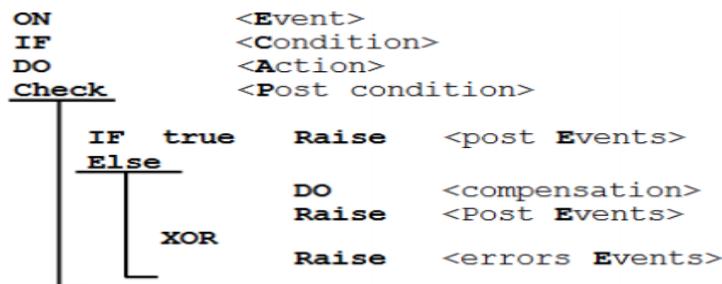
UML (UNIFIED MODELING LANGUAGE) est le résultat de la fusion de trois méthodes d'analyse orientées objet : la méthode OOD (*Object Oriented Design*), la méthode OMT (*Object Modeling Technique*) et la méthode OOSE (*Object Oriented Software Engineering*). À l'initiative de la société Rational Software, les auteurs principaux de ces trois méthodes se sont mis d'accord sur un langage de modélisation unifié. Celui est devenu une référence lorsqu'il a été retenu par un groupement public, l'Object Management Group (OMG) en 1997. UML était initialement un ensemble de diagrammes permettant de représenter un système informatique pour les développeurs travaillant avec une approche orientée objet. Après son évolution en 2004 vers la version UML2, ce langage de modélisation a été utilisé pour décrire un système d'information, notamment au niveau du cahier des charges [3]

UML se veut la continuité et l'unification des approches précédentes. UML introduit notamment la notion de points de vue offrant différentes approches d'un même système. Ceci se traduit par les différents diagrammes d'UML, chacun permettant d'appréhender le système sous un point de vue différent. UML propose une vision plus proche des langages orientés objet que de celui de l'entreprise. UML essaie de réduire la distance entre ces deux mondes afin d'avoir des logiciels de bonne qualité et qui répondent aux besoins métier. Cela reste limité puisque le métier, et particulièrement les processus métier sont décrits de manière souvent éclatée sur tout le modèle après avoir subi une transformation par les concepteurs logiciel afin de décrire les besoins techniques de l'application [9].

### 3.6.10. Les travaux de modélisation des processus basés sur les règles

#### 3.6.10.1. Le formalisme ECAPE

Dans cette section nous présentons un nouveau formalisme appelé ECAPE pour : Evènement – Condition – Action – Post condition – post Evènements [Compensation ou Evènement d’erreurs] (ECAPE-CE abrégé ECAPE). Le plus grand avantage de ce formalisme est qu’il permet, d’un côté, de valider l’exécution de l’action et lancer une compensation pour remplacer le traitement de l’action. D’un autre côté il permet de construire automatiquement un graphe d’exécution du processus utile pour analyser son bon fonctionnement. Une règle ECAPE se définit par :



L’évènement détermine quand une règle doit être évaluée. La condition est un prédicat dont dépend l’exécution de l’action. L’action, quant à elle, spécifie le code à exécuter si la condition est à vrai. La post condition est un prédicat dont la validation dépend de la règle. Le post évènements est l’ensemble d’évènements déclenchés pendant ou après l’exécution de la partie action de la règle. La compensation est l’action qui peut être exécutée, comme première alternative, dans le cas où la partie post condition est non satisfaite. Finalement les évènements d’erreurs sont l’ensemble d’évènements qui peuvent se déclencher comme seconde alternative, si la post condition n’est pas vérifiée.[7]

### 3.6.10.2. ECAPE-L

Le langage ECAPE-L proposé dans [12] s’appuie sur une syntaxe XML pour décrire la logique du processus métier en utilisant le modèle ECAPE ainsi que les différents éléments du processus. Ce langage offre une grande expressivité pour modéliser un processus métier en permettant de couvrir un grand nombre de « Workflow patterns ».

### 3.6.10.3. CECAPENETE

Dans notre démarche, chaque processus métier est spécifié par un ensemble de règles qui utilisent notre formalisme CECAPENETE basé sur ECA. Ce formalisme est défini par : Evènement – Condition – Action – Post condition – post Event, avec l’aspect (C) qu’est désigné l’aspect de sécurité ou interaction, le point (P) pour marquer une exception et avec nombre de fois (NB) pour répéter l’action jusqu’à Nb fois [8].

### 3.6.10.4. Synthèse

Dans ces travaux, les processus métiers sont définis par un ensemble de règles basées sur ECA. Ces règles sont des énoncés exprimant, de manière déclarative, le domaine métier, les politiques d’une entreprise, les règles métiers, ... etc. Le formalisme utilisé dans [7] est

ECAPE pour modélisation et Les réseaux de Pétri colorés pour la vérification. Mais, ce travail traite un seul aspect, c'est l'aspect fonctionnel, ce qui lui rend moins adaptable. Dans [13], les auteurs ont proposé un formalisme CECAPENETE qui fournit une flexibilité et adaptabilité de modélisation et une séparation des préoccupations : fonctionnel et sécurité. Dans d'autres travaux, ils ont utilisé les réseaux de pétri WF-net pour la vérification [14].

## 4. Vérification des processus métiers

La vérification répond à la question « Construisons-nous correctement le modèle ? ("is the system being built right ?")

La vérification est l'ensemble des actions de revue, inspection, test, preuve automatique, ou autres techniques appropriées permettant d'établir et de documenter la conformité des artefacts du développement vis-à-vis des critères préétablis. (ISO 8402) définit la vérification comme étant "la confirmation par examen et apport de preuves tangibles (informations dont la véracité peut être démontrée, fondée sur des faits obtenus par observation, mesures, essais ou autres moyens) que les exigences spécifiées ont été satisfaites".[2]

### 4.1. Les techniques de vérification des modèles des processus métier

La vérification de modèles peut s'appliquer à tout système ayant une sémantique comportementale et une multitude de scénarii d'exécution possibles. Le principe de cette vérification consiste à s'assurer que le système s'exécute conformément à ce qui a été prévu initialement durant la phase de sa conception. Cela passe par l'extraction automatique des comportements inattendus ou erronés dans des délais raisonnables tout en garantissant qu'une propriété est vérifiée par toutes les exécutions possibles du système.

Comme vu précédemment, l'ensemble de ces incohérences des processus métier est dû à différents types d'erreurs qui peuvent souvent être repérées et gérées dans la phase de modélisation, mais aussi dans la phase d'exécution et la phase de monitoring. Dans ce contexte, les travaux de recherche qui existent dans la littérature consistent à fournir un ensemble de techniques et d'outils afin d'assurer une absence totale de toute erreur qui peut induire des incohérences dans les processus métier modélisés par des diagrammes EPC, des diagrammes d'activité UML, des diagrammes BPMN ou encore par un modèle BPEL.

Sadiq *et al*, sont parmi les pionniers dans l'identification des erreurs structurelles, comme les impasses et les boucles infinies, dans les workflows. Leur travail vise principalement à identifier les erreurs syntaxiques avec un manque relatif pour gérer les erreurs sémantiques des workflows. Actuellement, la majorité des travaux qui sont menés dans ce registre se concentre sur l'aspect sémantique des processus métier, par exemple « le processus doit toujours finir par se terminer ? » et d'autres questions similaires. Dans les sous-sections suivantes, nous détaillons les travaux les plus connus dans ce domaine. [15]

#### 4.1.1. La vérification par guides style de modélisation

Dans cette technique, des règles de style ou de bonne pratique sont définies afin de détecter les erreurs dans la phase de modélisation telle que « interdire l'association d'une ressource à

deux activités métier en même temps ». Ces règles sont considérées comme des guides de bonne pratique de modélisation « *good modeling style* » qui sont utilisés afin de guider le concepteur à éviter les éléments qui peuvent conduire à des erreurs ou les éléments qui représentent une source potentielle d'erreurs.

Un exemple de ces règles est proposé dans le travail de Gruhn *et al.* pour la vérification des diagrammes EPC ou encore le travail de Van Dongen *et al* qui vérifie les modèles de références MySAP à travers l'AGL ARIS.[15]

#### 4.1.2. La vérification par la simulation

La vérification par simulation consiste à comprendre, évaluer et comparer le processus modélisé en utilisant plusieurs scénarii possibles. La simulation fournit ainsi des estimations quantitatives sur l'impact qu'une conception de processus est susceptible d'avoir sur son exécution. Plusieurs simulateurs de processus métier ont été proposés dans la littérature tels que ARIS, Protos, etc.

Ces outils permettent aux entreprises d'analyser rapidement leurs processus métier.

Dans un certain nombre de ces outils sont comparés par rapport à leur applicabilité dans le domaine du BPM et aux modèles utilisés. Toutefois, la technique par simulation utilise des données théoriques et la qualité des estimations fournies dépend fortement de la batterie de tests utilisée.[15]

#### 4.1.3. La vérification par modèles formels

La technique de vérification par modèle formels exige de formaliser et vérifier la spécification du processus en utilisant un modèle formel donné. L'objectif étant de minimiser les risques qu'un dysfonctionnement puisse se produire en se basant sur les propriétés de vérification offertes par ses modèles comme l'absence de blocage. Plusieurs travaux utilisent cette technique pour vérifier le bon fonctionnement des processus métiers. D'une façon générale, les réseaux de Pétri (RdP) est le modèle le plus utilisé et cela parce qu'il combine les avantages de la représentation graphique avec l'aspect sémantique attribué au comportement du processus modélisé. Le plus grand intérêt d'un réseau de Pétri ne se limite pas à sa capacité de modélisation de comportement. En effet, un RdP offre aussi un large éventail de propriétés mathématiques et des outils qui permettent d'analyser le bon fonctionnement du système. [7]

#### 4.1.4. La vérification par process\_mining

Dans la phase de monitoring, les erreurs opérationnelles et les erreurs non fonctionnelles sont détectées et cela en analysant le processus exécutable dans le but de mesurer les performances opérationnelles et métier en se basant sur les journaux des événements. Le processus exécutable est analysé, aussi, afin de reconstruire le processus métier, à partir des informations cumulées lors de la phase précédente. Le processus résultat sera comparé au processus modélisé pour savoir s'il s'agit du même déroulement de l'ensemble des activités, c'est-à-dire permettre aux analystes métier d'assurer que le processus exécutable correspond aux exigences métier.[7]

### 4.2. Synthèse

La technique par simulation utilise des données théoriques et la qualité des estimations fournies dépend fortement de la batterie de tests utilisée. A son tour, la technique de vérification par guides de style ne peut pas remplacer la vérification par les modèles formels car cette technique tente de limiter les erreurs les plus connues, d'autant plus que cette technique ne détecte pas les erreurs produites lors de l'implémentation du processus. Par ailleurs, le temps que consomment les algorithmes liés à la technique de vérification par les modèles formels dépend de la complexité des algorithmes de traduction de la définition des processus métier en modèles formels et la complexité des algorithmes de détection des propriétés à vérifier. Malgré cet inconvénient, la technique de vérification par les modèles formels s'avère avantageux et cela parce qu'elle augmente la certitude qu'un dysfonctionnement ne se produira pas. Pour cette raison, nous allons détailler, dans la section suivante, les différents modèles formels utilisés pour la vérification des processus métier. [7]

### 4.3. Les erreurs et exceptions d'un processus métier

Une erreur désigne un « *Bug* » dû à une imprécision de modélisation ou au dysfonctionnement lors de l'exécution d'un processus métier. Ces erreurs peuvent concerner des erreurs fonctionnelles, sémantiques ou d'exécution. Une erreur d'exécution qui caractérise toutes situations imprévues qui surgissent pendant l'exécution d'un programme, est appelée une exception. Dans la littérature, plusieurs travaux ont été conduits pour étudier la nature des erreurs de modélisation et les exceptions survenues dans l'exécution des processus métier et cela d'une manière indépendante de leurs approches de modélisation et des technologies utilisées pour leur Exécution D'une manière générale, les erreurs survenues dans un processus métier peuvent être classifiées comme suit : [7]

#### 4.3.1. Les erreurs fonctionnelles

Les erreurs fonctionnelles concernent la cohérence fonctionnelle du processus métier. L'origine de ces erreurs est due à une mauvaise conception comme les boucles infinies (Livelock) qui exprime une situation où l'exécution du processus tourne en rond indéfiniment, l'inter blocage (deadlock) qui exprime une situation où l'exécution du processus est dans une attente infinie, ou encore des activités continuent d'être exécutées après la terminaison du processus. [7]

#### 4.3.2. Les erreurs opérationnelles

Les erreurs opérationnelles concernent les événements aléatoires qui sont susceptible d'entraîner des exceptions. En effet, ces événements sont rares et imprévus comme une panne matérielle du système informatique qui peut déstabiliser le fonctionnement du processus métier ou encore les exceptions générées à cause de l'indisponibilité d'une ou plusieurs ressources au moment où une activité en cours d'exécution veut y accéder ou le cas où la ressource ne satisfait plus les critères d'allocation qu'une activité exige. [7]

#### 4.3.3. Les erreurs non fonctionnelles

Les erreurs non fonctionnelles concernent les erreurs sémantiques qui sont commises par les concepteurs et qui n'a aucune incidence sur l'exécution elle-même du processus, mais le résultat obtenu n'est pas celui attendu. Un exemple de ces erreurs est de définir une activité non atteignable (dead activity), dans ce cas, l'activité définie ne sera jamais exécutée. Un autre exemple est celui de violation de contraintes qui assurent l'intégrité et la cohérence métier des processus (effectuer une commande si le stock est épuisé). En effet, ces contraintes peuvent être spécifiées sur les données, les ressources ou sur les comportements de processus. Cependant un conflit de contraintes, par exemple, peut conduire à une situation invalide. [7]

## **5. Conclusion**

La flexibilité est devenue l'intérêt majeur des entreprises et cela pour améliorer leur productivité, leur fiabilité et leur rapidité d'adaptation aux changements. Dans ce contexte, des langages déclaratifs ont été proposés pour modéliser un processus par un ensemble d'états et un ensemble de contraintes, cette manière offre une flexibilité de modélisation. Nous avons vu de la vérification des processus métier a pour but de montrer que les activités du processus s'exécutent en conformité à son plan de réalisation et qu'elles n'ont pas introduit des erreurs. Pour cela plusieurs techniques sont proposées. Généralement la technique de vérification par réseaux de Pétri est la plus utilisée est cela est du à leur représentation graphique avec l'aspect sémantique attribué au comportement du processus modélisé.

## *CHAPITRE II : CLOUD COMPUTING*

## 1. Introduction

L'avancement rapide des technologies de l'information et de la communication a permis le développement de nouveaux paradigmes informatiques, où les techniques de traitement, de stockage, de communication, de partage et de diffusion de l'information ont radicalement changés. Les individus et les organisations sont de plus en plus recours à des serveurs externes pour le stockage et la diffusion efficace et fiable d'informations.

Le **Cloud Computing**, ou « informatique dans les nuages », est une nouvelle technologie informatique qui permet le déplacement des traitements et fichiers informatiques de l'ordinateur local vers des serveurs distants. Elle consiste à proposer les services informatiques sous forme de services à la demande, accessibles de n'importe où, n'importe quand et par n'importe qui grâce à une connexion internet. Basé sur le principe de fournir les ressources informatiques sous forme de services et de facturer leur utilisation en fonction de leur usage (*pay as you go* en anglais), le **Cloud Computing** permet d'effectuer des économies d'échelle grâce à l'externalisation de ces ressources vers des fournisseurs spécialisés. C'est le cas par exemple de Google App Engine, Amazon EC2 ou Microsoft Azure, les offres permettant d'utiliser les infrastructures de calcul et de communication de Google, Amazon ou Microsoft comme des services utilitaires. Ces offres proposent différents types de service généralement spécifiques à l'usage que les utilisateurs peuvent en faire. Une architecture en couches a été proposée pour catégoriser les différents types de services et les usages correspondants cette architecture est appelée *modèle SPI(Software/Platform/Infrastructure)*. Dans ce chapitre, nous allons présenter globalement l'historique du « Cloud computing » et l'origine de ce terme, suivi d'une définition explicite de ce dernier qui sera basée sur une analyse des définitions proposées par le monde académique. Nous décrivons aussi la virtualisation qui est une partie essentielle dans « l'informatique en nuage », sans oublier les services de Cloud, les types de Cloud et ses acteurs ainsi que les avantages, les inconvénients, les objectifs principaux et les domaines d'utilisation du Cloud computing.

## 2. Définition

Selon la définition du **National Institute of Standards and Technology** (NIST), le cloud computing est l'accès via un réseau de télécommunications, à la demande et en libre-service, à des ressources informatiques partagées configurables (réseaux, serveurs, stockage, applications et services), qui peuvent être provisionnées rapidement et libérées avec un effort de gestion minimale. Ce modèle de nuage est composé de cinq caractéristiques essentielles (service à la demande, Large accès au réseau, Mise en commun des ressources, Élasticité rapide, Service mesuré), trois modèles de services (Software as a Service (SaaS), Platform as a Service (PaaS), Infrastructure as a Service (IaaS)) et quatre modèles de déploiement (Le cloud public, Le cloud privé, Le cloud hybride, Le cloud communautaire). [18]

Pour **CISCO** [19] le Cloud Computing est une plateforme de mutualisation informatique fournissant aux entreprises des services à la demande avec l'illusion d'une infinité de ressources. Alors, le Cloud Computing est un concept qui consiste à transférer des fichiers ou des bases de données sur des serveurs à distance, qui étaient auparavant stockés dans la machine du client. Il permet d'accéder sur demande aux mêmes informations par plusieurs personnes.

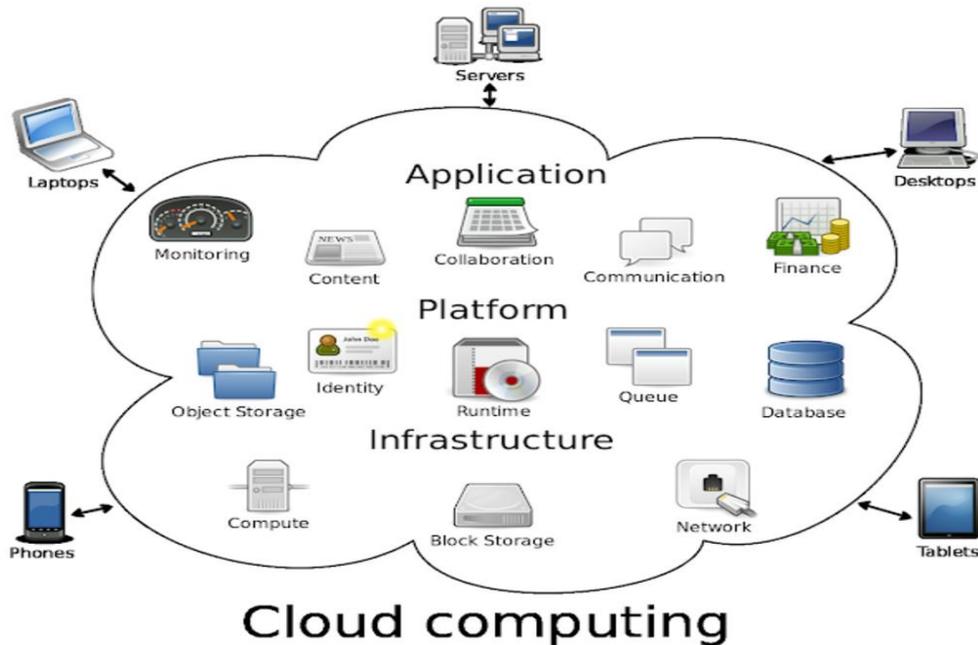


Figure II.1: *L'environnement de Cloud Computing.*

### 3. Les bases informatiques du Cloud Computing

#### 3.1. Infrastructure

L'infrastructure informatique du Cloud est un assemblage de serveurs, d'espaces de stockage et de composants réseaux 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, service level agreement). Par ailleurs, les serveurs d'un Cloud affichent des coûts de fonctionnement moins élevés et ils peuvent être plus fiables s'ils ne sont pas tous équipés de disques internes [20].

#### 3.2. La Virtualisation

Techniquement, le Cloud Computing n'est pas en soi une technologie nouvelle, le Cloud Computing résulte de fruit des technologies existantes comme internet et la virtualisation.

La virtualisation consistant la mutualisation des outils techniques informatiques, cette notion permet une gestion optimisée des ressources matérielles, telle que sur une seule ressource physique on peut exécuter plusieurs machines virtuelles donc plusieurs systèmes

d'exploitation et de variés applications, la virtualisation permet à serveurs, postes de travail, le stockage et d'autres systèmes une indépendance de la couche physique du matériel. Après les structures de mainframe et de clients/serveurs, le Cloud Computing devient comme un nouveau modèle.

La virtualisation des serveurs fournit une grande modularité dans la répartition des charges et la reconfiguration des serveurs en cas d'évolution ou de défaillance passagère.

Comme avantage la virtualisation permet L'emploi optimale des ressources d'un groupe de machines (répartition des machines virtuelles sur les machines physiques en fonction des charges respectives) et l'installation, tests, développements et l'économie sur le matériel (consommation électrique, entretien physique, surveillance). [20]

### **3.3. Interfaces de service**

L'interface 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 (service level agreement) 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 [20].

## **4. Les modèles de services de cloud computing**

Les services de cloud computing offrent des modèles pratiques et payants qui éliminent les dépenses et la maintenance coûteuses. Les fournisseurs de cloud hébergent un choix d'infrastructures, de plates-formes et d'offres de logiciels sur place qu'ils « louent », ce qui donne à votre organisation la flexibilité nécessaire pour transformer les services de cloud computing en fonction de vos besoins changeants.

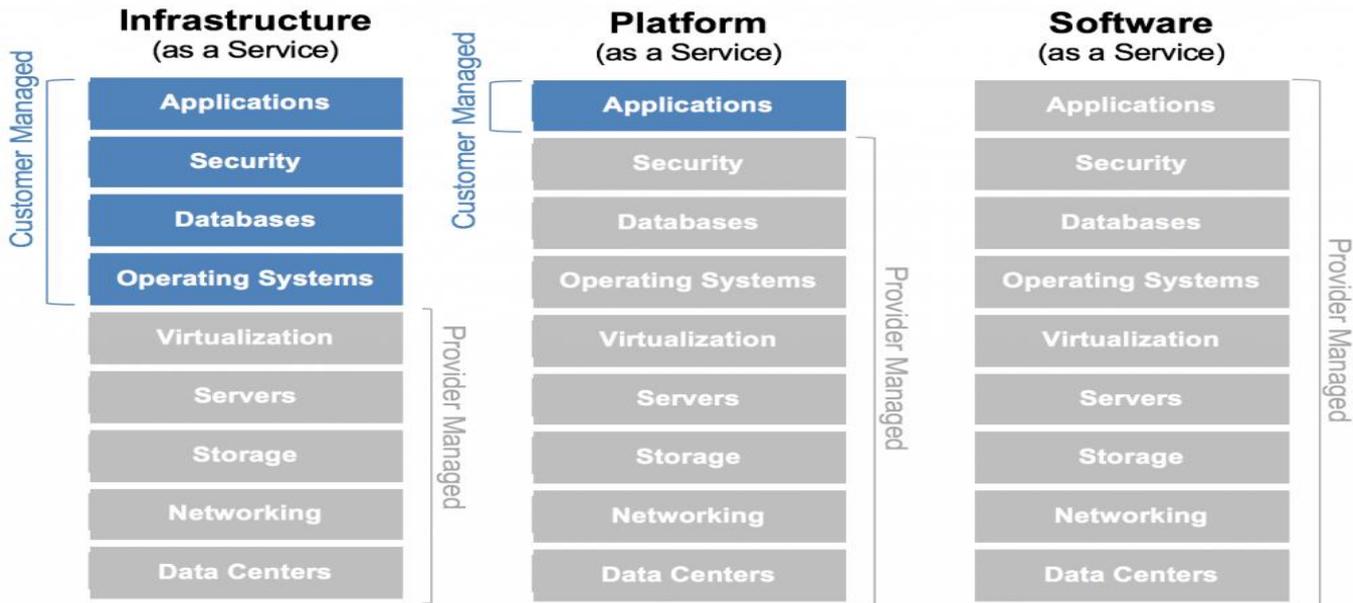


Figure II.2 : Les modèles de services de cloud computing.

#### 4.1.SaaS (Software as a Service)

La couche des « logiciels en tant que service » regroupe les applications accessibles via internet ou le matériel, l’hébergement, l’environnement d’application et le logiciel sont dématérialisés et dont la gestion est en dehors de la responsabilité de l’usager. Les applications de type SaaS sont diverses et variées. On peut citer les applications telles que la messagerie Gmail, l’éditeur de documents Office 365, le réseau social Facebook, les services de stockage de données Google Drive et Dropbox, ou encore le service de cartographie en ligne Google Maps.[21]

#### 4.2.PaaS (Platform as a Service)

La PaaS (ou la plate-forme en tant que service) est un modèle de Cloud correspond principalement à des environnements de développement qui met à disposition des plates-formes d’exécution, de déploiement et de développement d’applications. Pour cela, PaaS fournit un niveau d’abstraction supplémentaire par rapport à IaaS sauf que l’utilisateur n’a plus en charge que les couches de données et d’applications. Une Plateforme de développement sur le Cloud permet d’utiliser et réutiliser un large spectre d’outils, de composant logiciels, et de blocs de codes ce qui permet de réduire considérablement les couts de déploiement des applications avec une grande rapidité de développement et d’hébergement. Les exemples typiques du PaaS sont : Google App Engine, ConPaaS, Windows Azure, Elastic Beanstalk, etc. [22]

#### 4.3.IaaS (Infrastructure as a Service)

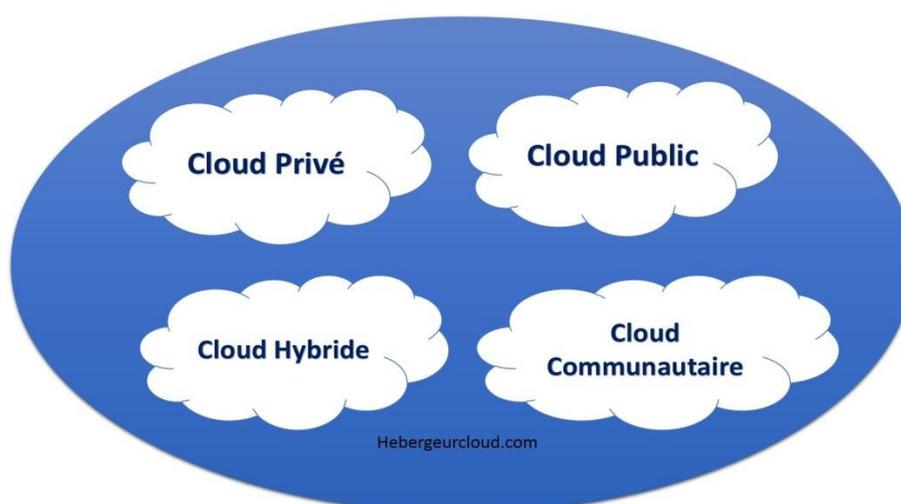
L’IaaS ou l’infrastructure en tant que service offrent des plateformes de virtualisation configurables qui met à la disposition des utilisateurs des ressources informatiques prêtes à

l'emploi telles que des équipements réseau, des serveurs, de l'espace de stockage, etc.

Ce modèle de service est vu comme une abstraction d'un centre de calcul ou de stockage de données sur lequel les utilisateurs déposent leur environnement de production (système d'exploitation, intergiciel, logiciel...). Les serveurs virtuels peuvent être démarrés ou arrêtés à la demande des utilisateurs, et cela permet de concentrer davantage sur la réalisation et le développement des applications sans avoir à se préoccuper de l'acquisition de serveurs ou de la gestion de l'infrastructure.[22]

## 5. Les modèles de déploiement du Cloud

Le NIST définit quatre modèles de déploiement pour le Cloud Computing.



**Figure II.3 : Les modèles de déploiement du Cloud.**

### 5.1. Cloud privé

Il s'agit d'un environnement dont les ressources servent exclusivement l'entreprise utilisatrice. L'infrastructure peut être localisée/gérée sur place ou par un tiers mais l'entreprise est la seule à l'utiliser. Ce modèle offre un haut degré de contrôle et de sensibilité, permettant au propriétaire d'un service à facilement maîtriser et administrer son système en termes de réglementations ou de sécurité, par exemple. [21]

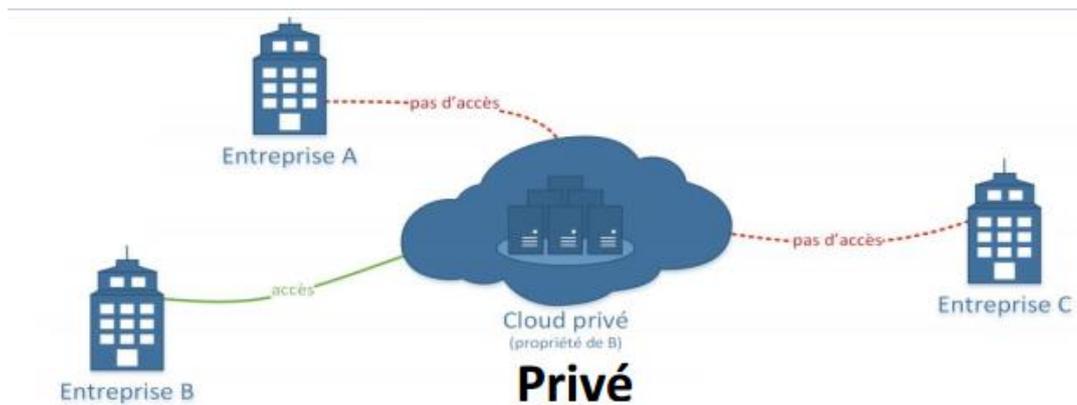


Figure II.4 : *cloud privé*.

## 5.2. Cloud communautaire

L'infrastructure est partagée entre plusieurs organisations et peut être gérée par le groupe ou par un tiers. Ce modèle est généralement adopté par les institutions gouvernementales, hôpitaux, hôtels, etc. afin de profiter de ressources communes (réseau, stockage, sécurité, etc.), ce qui assure un fonctionnement efficace des différents déploiements concernés. [21]

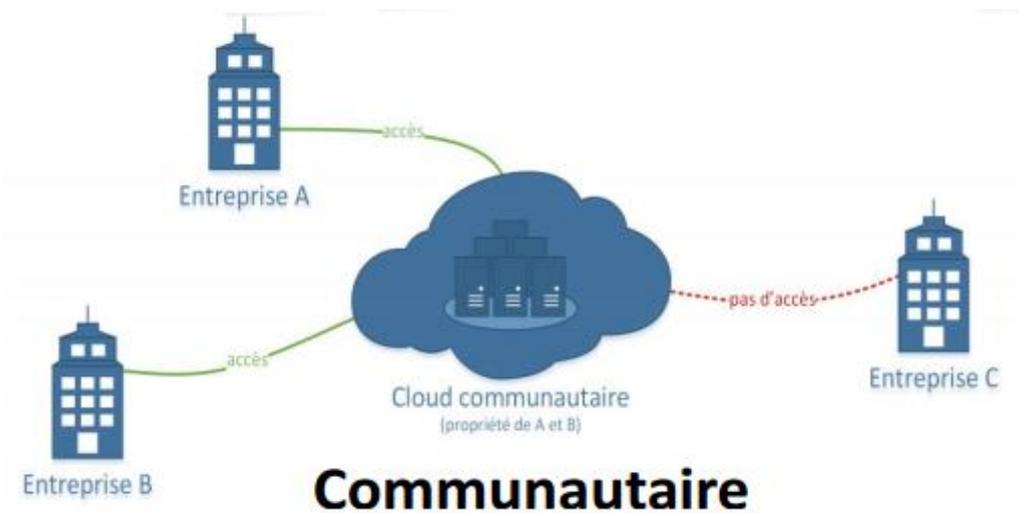


Figure II.5 : *cloud communautaire*.

## 5.3. Cloud public

Les ressources sont fournies par un prestataire, propriétaire de celles-ci, et mutualisées pour un usage partagé. Ce modèle est largement adopté par les fournisseurs de IaaS afin d'offrir une large offre de ressources partagées, à faible coût et gérées de manière élastique. [21]

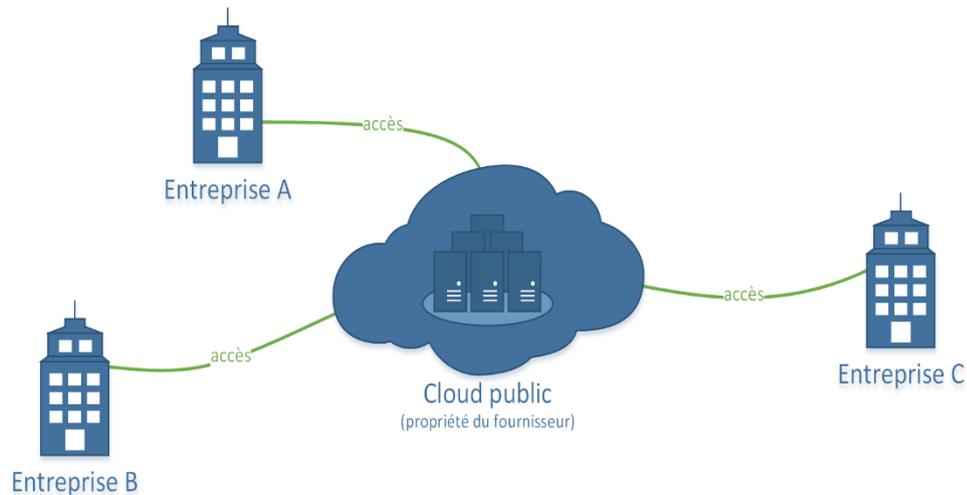


Figure II.6 : *cloud public*.

#### 5.4. Cloud hybride

Il s'agit de la combinaison de plusieurs clouds indépendants publics ou privés. Ceux-ci doivent respecter des standards et des technologies communes pour assurer la portabilité des applications entre les clouds. Dans ce modèle de déploiement, une entreprise peut, par exemple, tirer parti du faible coût d'un cloud public pour l'hébergement de certains services classiques, tout en recourant à un cloud privé pour gérer des services plus sensibles (confidentialité, sécurité, etc.). [21]

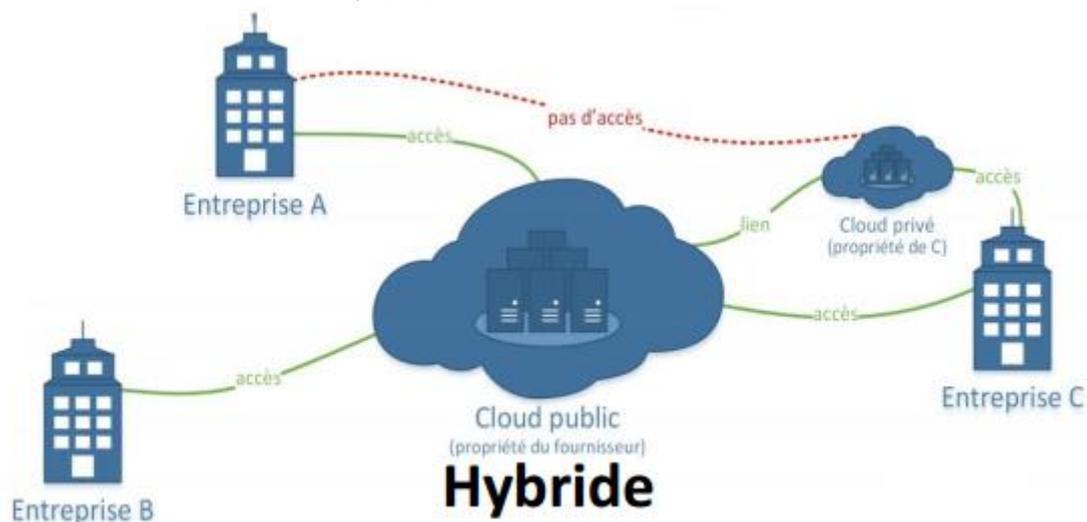


Figure II.7 : *cloud hybride*.

## 6. Les cinq caractéristiques du Cloud Computing

Le modèle Cloud Computing se différencie par les cinq caractéristiques essentielles suivantes :



**Figure II.8 : Les cinq caractéristiques du Cloud Computing.**

### 6.1. Accès réseau universel

Un environnement de type Cloud Computing est accessible via le réseau, quel que soit le périphérique (PC, Mac, tablette, SmartPhone, ...). [23]

### 6.2. Mise en commun de ressources

Dans un environnement de type Cloud Computing, on ne pense pas en nombre de serveurs, taille de disques, nombre de processeurs..., mais en puissance de calcul, capacité totale de stockage, bande passante disponible. [23]

### 6.3. Elasticité

Grâce au Cloud, il est possible de disposer de plus de ressources très rapidement pour soutenir une forte demande (par exemple pour garantir une bonne expérience d'achat sur une plateforme web d'e-commerce durant les fêtes de fin d'années). Inversement, au-delà de la provision de ressources, il est possible avec le Cloud de diminuer les ressources utilisées (par exemple en cas de baisse d'activité sur cette même plateforme web d'e-commerce) si celles-ci sont supérieures à ce qui est nécessaire. [23]

### 6.4. Libre Service

Dans un environnement de type Cloud Computing, il est possible à un utilisateur de consommer les services ou les ressources sans pour autant nécessiter une demande d'interventions auprès du fournisseur : équipe IT ou fournisseur externe (par exemple, un développeur qui souhaite tester son application sur une machine virtuelle représentative d'un

poste standardisé de son entreprise peut, au travers d'un portail web, provisionner ou utiliser une machine). [30]

### 6.5. Service mesurable et facturable

Dans un environnement de type Cloud Computing, le fournisseur de la solution est capable de mesurer de façon précise la consommation des différentes ressources (CPU, stockage, bande passante, ...) ; cette mesure lui permet de facturer à l'usage le client.[23]

## 7. La sécurité du Cloud

La sécurité consistant de garantir la confidentialité, l'intégrité, l'authenticité et la disponibilité des informations.



*Figure II.9 : La Sécurité du Cloud Computing.*

### 7.1. La confidentialité

La confidentialité garantit que les données d'un utilisateur ne soient atteintes que par les personnes autorisées et à empêcher d'atteindre par les mauvaises personnes, Le propriétaire des données doit savoir quelles données sont protégées et dans certains cas être capable de leur suppression. Les outils de Cloud Computing comportent des manières de confidentialité comme la gestion des identités et des accès, le cryptage ou l'isolation ; Les contrôles d'accès les plus sécurisés ne sont d'aucune protection contre un attaquant qui gagne l'accès à des

informations d'identification ou des clés (exemple : un numéro de compte ou numéro de routage). La plupart des échanges internes ou externes au Cloud sont encapsulés en SSL (Secure Sockets Layer) et authentifiés avec un certificat généré par le client. Ce certificat n'est lié à aucune autorité de certification racine de confiance, mais plutôt auto-signé par le client lui-même. Tant que ce dernier assure le contrôle de sa clé privée, le mécanisme permet un degré élevé d'assurance : seuls les clients autorisés et présentant cette clé peuvent accéder à des aspects spécifiques du service. Le cryptage est un mécanisme défini pour assurer la confidentialité des données. Le client peut utiliser le chiffrement des données pour le stockage dans le cloud, Les identifiants et mots de passe constituent une procédure standard ; authentification à deux facteurs est en train de devenir la norme et la vérification biométrique est une option aussi. Aussi dans la plupart des cas, le Cloud Computing est conçu sur la virtualisation. Un des points critiques est l'isolement de la machine virtuelle avec les autres machines virtuelles. Actuellement, les fournisseurs de Cloud Computing bâti sur les technologies Microsoft ou VMware, assurent une haute isolation des systèmes invités grâce à la forte expérience des produits de virtualisation de ces différents éditeurs (Virtual PC, Virtual Server et plus récemment Hyper-V chez Microsoft, VMWare Workstation...).[20]

## 7.2.L'intégrité

Les utilisateurs qui veulent à externaliser leurs données peuvent être protégés contre les modifications non autorisées, Intégrité consiste à garder la cohérence, l'exactitude et la fiabilité des données sur son cycle de vie. Dans le cloud, il est nécessaire d'assurer l'intégrité des données pendant un transfert ou un stockage. Des mesures doivent être employées pour s'assurer que les données ne peuvent pas être modifiées par des personnes non autorisées (par exemple, dans une violation de la confidentialité). En outre, des moyens doivent être mis en place pour détecter tout changement dans les données qui pourraient survenir à la suite d'événements non-humaines telles qu'une impulsion électromagnétique ou plantage du serveur. Si un changement inattendu se produit, une copie de sauvegarde doit être disponible pour restaurer les données affectées à son état correct.[20]

## 7.3.La Disponibilité

Parmi les importants avantages fournis par des plates-formes de Cloud Computing est la disponibilité forte des ressources ,La disponibilité est mieux assurée par le maintien rigoureusement tout le matériel, les réparations de matériel immédiatement en cas de besoin, fournir un certain degré de redondance et de basculement, fournissant une bande passante de communication adéquate et prévenir l'apparition de d'étranglement, la mise en œuvre des systèmes d'alimentation de secours d'urgence et tenir au courant de toutes les mises à niveau du système nécessaire ;La condition de la connectivité, Une perte de connectivité au niveau de votre entreprise, de votre fournisseur d'accès à Internet ou de votre fournisseur de cloud rend ces services indisponibles.[20]

## 8. Acteurs du Cloud

Selon le NIST, on distingue cinq acteurs majeurs au centre du fonctionnement du paradigme Cloud Computing : le fournisseur, l'utilisateur, l'auditeur, le courtier et le transporteur de Cloud. Ces acteurs représentent une entité physique ou morale (une personne ou une organisation) qui prend part à un processus ou transactions dans le Cloud, de la manière suivante :[21]

- *Le fournisseur (cloud provider)* : Représente une personne ou un organisme mettant un ensemble de services cloud à la disposition d'utilisateurs (ou consommateurs) potentiels.
- *L'utilisateur (cloud customer)* : Représente une personne, un organisme ou une entité qui se prête à l'utilisation des différents services mis à disposition par le fournisseur, généralement via un contrat client.
- *L'auditeur (cloud auditor)* : Représente une entité dont la tâche est de mesurer et d'évaluer les services Cloud en termes de performances, sécurité, coûts, etc., tout en restant indépendant du fournisseur et des utilisateurs du cloud.
- *Le courtier (cloud broker)* : Représente un parti qui s'occupe de la gestion et la prestation de service cloud en négociant les relations entre les utilisateurs et le fournisseur Cloud.
- *Le transporteur (cloud carrier)* : Représente un organisme ou une entité intermédiaire qui s'occupe de fournir la connectivité et la disponibilité des services Cloud, en les transportant du fournisseur vers le consommateur.

## 9. Les avantages et les inconvénients du Cloud Computing

### 9.1. Les avantages du Cloud Computing

L'utilisation du Cloud Computing présente de nombreux avantages [24]

- **Réduction des coûts d'infrastructure** : Avec le Cloud Computing, il n'est plus requis de mettre en place une infrastructure entière ou d'augmenter le nombre de serveurs physiques proportionnellement aux charges. Toutes les ressources nécessaires peuvent être allouées facilement et avec un coût à l'utilisation afin de répondre à des montées de besoins et en temps réel.
- **Réduction de la responsabilité de gestion des systèmes** : La gestion et la maintenance des serveurs sont prises en charge par le fournisseur de service Cloud. Ce qui permet donc d'alléger considérablement les tâches des utilisateurs.
- **Mises à jour logicielles automatiques** : La mise à jour des applications est systématique, et le fournisseur décharge les clients de toute responsabilité de maintenance. Ainsi les utilisateurs de service n'ont pas besoin d'acheter aucun logiciel ou licence.
- **La flexibilité des ressources** : L'utilisateur peut augmenter ou réduire la capacité de son infrastructure sans investissement majeur et dans un délai réduit.

- **Puissance de calcul et stockage illimités** : contrairement aux grilles, les ressources dans un Cloud peut être ajouté à tout moment ce qui met à dispositions des ressources de grandes capacités.
- **Fiabilité** : le Cloud Computing garantie aux utilisateurs la qualité de service (QoS), l'équilibrage de la charge, la sauvegarde et la récupération des données.
- **Commodité de l'accès** : l'utilisateur peut à tout moment et à partir de n'importe quel appareil se connecter à ses applications et ces données.
- **La sécurité des données** : La sécurité des données est le principal frein d'adoption du Cloud Computing. Dans ce contexte, les fournisseurs garantissent aux utilisateurs un très haut degré de sécurité des données avec le chiffrement des données, la surveillance logicielle et la sécurisation des lieux de stockage (Centres de données).
- **Partage de données** : Les données peuvent être partagées, puisque tout utilisateur du Cloud Computing peut aisément rendre disponibles ses données à un ou plusieurs autres utilisateurs du Cloud Computing. Il est donc possible de créer une plateforme virtuelle collaborative en un temps record.
- **Informatique verte (Green Computing)** : Les infrastructures gérées en interne sont souvent sous-utilisées, alors que l'infrastructure d'un Cloud mutualise l'ensemble de ressources pour un grand nombre de consommateurs. Elle permet alors de minimiser le nombre des équipements et d'augmenter le taux son utilisation.

## 9.2. Les inconvénients de Cloud Computing

L'accès au Cloud passe nécessairement par une connexion internet : par conséquent, si votre connexion est coupée, momentanément ou durablement, vous n'avez plus accès à vos documents ou service.

**Sécurité** : la plateforme Cloud, si elle est externe (non installée sur le réseau interne ou avec une ouverture extérieure) doit être suffisamment sécurisée pour éviter le risque d'intrusion, de vol des données par piratage. [25]

## 10. Les principaux fournisseurs actuels de Cloud Computing

### 10.1. Amazon Web Services (IaaS)

Est une division du groupe américain de commerce électronique Amazon.com.

**EC2** : Elastic Compute Cloud, est un service Web qui met à disposition sur le cloud de la capacité de traitement ajustable.

**EBS** : Elastic Block Storage, fournissent du stockage persistant de type NAS.

**CloudWatch** : est un service de surveillance pour les ressources du cloud AWS et les applications que vous exécutez sur AWS.

**Simple DB** : est un service Web simple qui fournit les fonctionnalités essentielles d'une base de données (consultation élémentaire en temps réel, requêtes simples de données structurées).

**CloudFront** : est un service web qui accélère la distribution des contenus web statiques et dynamiques, tels que les fichiers .html, .css, .js et image, aux utilisateurs.

**SQS** : Simple Queue Service, est un service de file d'attente de messagerie entièrement géré qui facilite le découplage et la mise à l'échelle des micro services, des systèmes décentralisés et des applications sans serveur.

**RDS** : Relational Database Service, Avec Amazon RDS, configurez, gérez et dimensionnez facilement une base de données relationnelle dans le cloud. [26]

## 10.2. Amazon Web Services (PaaS)

**S3** : Simple Storage Service, est un service de stockage sur Internet. Il est conçu pour faciliter l'informatique à l'échelle d'Internet pour les développeurs. [26]

## 10.3. Microsoft Azure (PaaS)

Est la plate-forme applicative en nuage de Microsoft.

**Azure Compute** : hébergement scalable d'applications ASP.NET ou batches .NET /natifs.

**Azure Storage** : stockage hautement scalable pour les données non structurées.

**SQL Azure** : base de données relationnelle (SQL Server) scalable dans le Cloud.

**Access Control** : fédération d'identité pour SSOs dans le Cloud. [26]

## 10.4. Google AppEngine (PaaS)

Est une plateforme de conception et d'hébergement d'applications web basée sur les serveurs de Google.

**Cloud Hosting** : hébergement scalable d'applications Python ou Java, avec stockage et base de données plate BigTable. [26]

## 10.5. Google Apps (SaaS)

Version professionnelle de Gmail et Google Apps : suite bureautique en ligne, consommable à la demande. [26]

## 11. Conclusion

Le Cloud est un modèle de distribution et de partage de ressources à grand échelle, qu'il offre aux utilisateurs un environnement plus simple et plus efficace pour travailler, parmi d'intérêt de Cloud : facilité le travail, augmente la collaboration, réduire le coût de matériel.

*CHAPITRE III : DESCRIPTION ET  
IMPLÉMENTATION DE NOTRE APPROCHE*

## 1 Introduction

Dans ce chapitre, nous expliquons notre démarche de modélisation flexible d'un processus métier sur cloud. Nous allons présenter l'exemple modélisé, le simulateur CloudSim et ses fonctionnalités, l'implémentation de cet exemple et les outils utilisés avec quelques interfaces de notre application.

## 2 Définition du problème

Le problème du classement est formalisé comme suit :

Nous considérons les tâches P1, P2, ..., Pn qui arrivent avec un ordre pour être exécuter dans un service cloud. Le fournisseur de cloud doit adopter une politique d'ordonnancement enfilade qui permet aux processus exécuter le plus possible en respectant les critères d'ordonnancement.

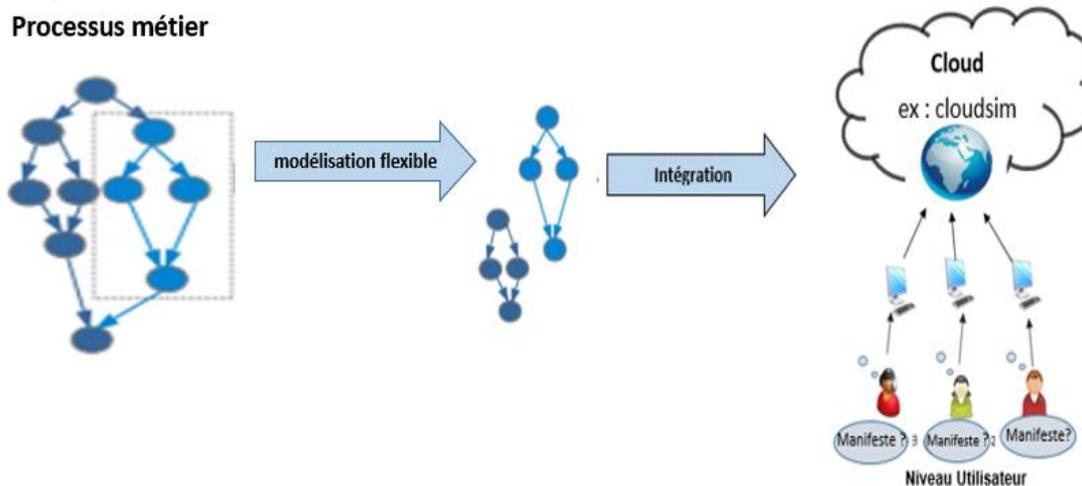


Figure III.1 : Démarche de l'implémentation du l'exemple.

## 3 Démarche détaillé de réalisation de l'exemple

Afin de réaliser une solution au problème précédent, nous avons proposé une approche constituée de trois étapes principales : (i) Formalisation ; (ii) Modélisation. (iii) Intégration.

Dans la phase de formalisation, nous avons décrit le processus métier de l'ordonnancement sous forme des entres, sorties et contraintes. Ce dernier est modélisé par des modèles d'ordonnancement basé sur les algorithmes comme l'algorithme de RR et l'algorithme PSO et l'algorithme Min-Min et l'algorithme Max-Min. Les algorithmes sont intégrés comme un service dans le cloud pour être mesurés.

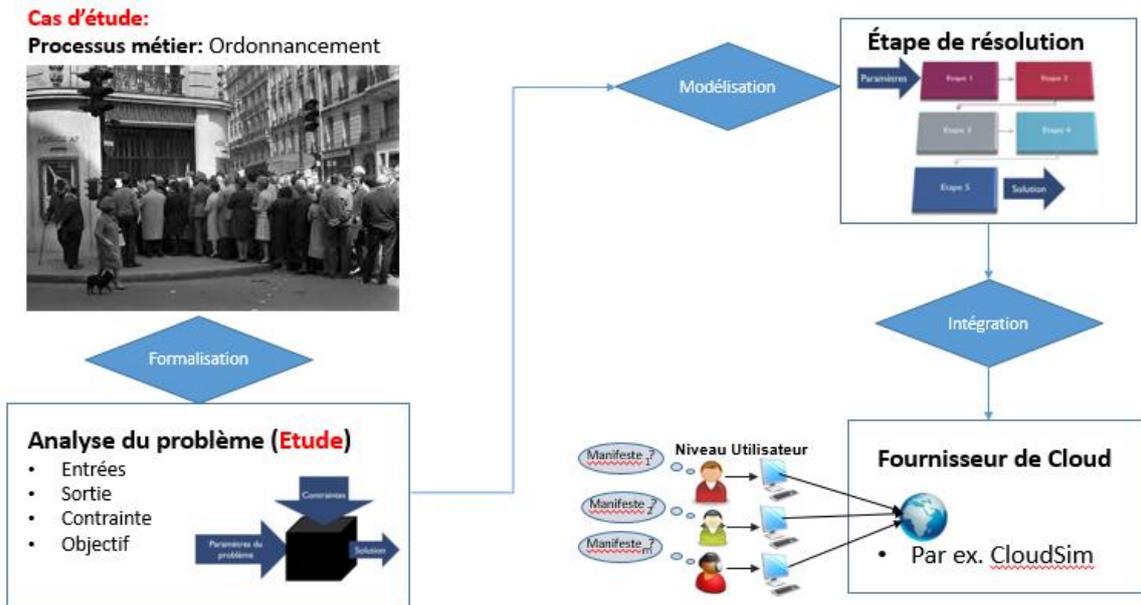


Figure III.2 : Vue globale de notre solution.

### 3.1 Formalisation du problème

Notre problème est formalisé comme suit :

- **Entrées** : ensemble des processus pour le traitement. **P1. P2 ...Pn** (P1, P2, ..., Pi ..., Pn); chaque processus Pi est caractérisé par  $T_{att}, T_{exe}, T_{arr}, T_{db}, T_{fin}$

$T_{att}$  : temps d'attente pour chaque processus.

$T_{exe}$  : temps d'exécution.

$T_{arr}$  : temps d'arrivée de processus.

$T_{db}$  : temps début d'exécution.

$T_{fin}$  : temps de fin d'exécution.

$$T_{fin}(Pi) \geq T_{db}(Pi) \quad Pi \in P$$

$$T_{arr}(Pi) \geq 0$$

$$T_{att}(Pi) = T_{db}(Pi) - T_{arr}(Pi)$$

- **Contraintes** : l'ordonnancement des processus est régi par des contraintes qui peuvent être
  - Des contraintes temporelles.
  - Des contraintes des ressources : performances et disponibilités des ressources.
  - Quantité d'information disponible.
- **L'objectif** : l'utilisation du cloud pour faire l'exécution du processus métier d'ordonnancement d'une manière optimale qui respecte les contraintes.

- **Sortie** : une planification des processus exécutables suivant les contraintes.

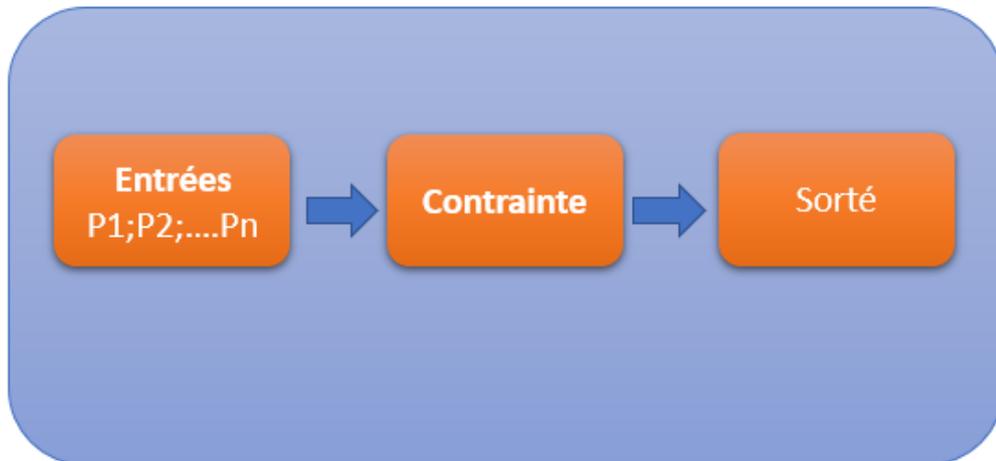


Figure III.3 : Formalisation du problème.

### 3.2 Modélisation de notre processus métier par le paradigme ECA

La gestion de l'ordonnancement exige la satisfaction des contraintes temporelles et des contraintes d'allocation de ressources. Ces contraintes doivent être modélisées et respectées durant l'exécution de ce processus métier. Dans ce travail, nous avons modélisé notre processus métier (Ordonnancement) par le paradigme ECA (Event-Action condition).

#### 3.2.1 Règle ECA

La notion de réagir à des situations dans les systèmes actifs est supportée par des règles actives de la forme Événement-Condition-Action (les règles ECA). La sémantique d'une règle ECA est :

*Lorsqu'un évènement E se produit Si la condition C est satisfaite Alors exécuter l'action A*

- La partie *Évènement* caractérise un fait significatif pour le déclenchement de la règle.
- La partie *condition* examine le contexte d'occurrence de l'évènement
- La partie *Action* spécifie la tâche à réaliser si l'évènement a eu et si la condition a été validée.

#### 3.2.2 Adaptation de règles ECA

Nous avons appliqué les règles ECA sur notre processus métier comme suit : La définition des évènements qui provoque les actions, les contrôles des contraintes temporelles et les contraintes des ressources (voir figure III.4)

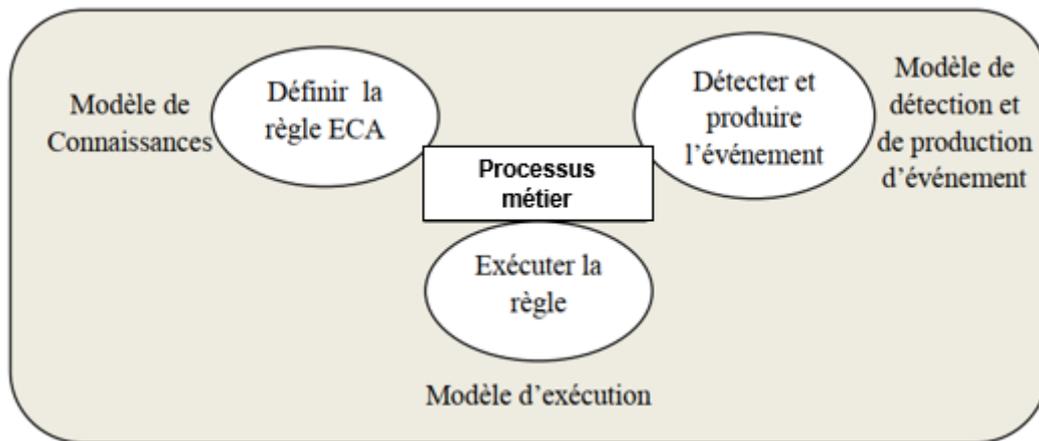


Figure III.4 : Adaptation de règles ECA.

Le tableau III.1 montre une synthèse de ECA selon notre cas d'étude du processus métier d'ordonnement.

Evènement	Condition	Action
Arriver d'une tâche: <i>Stop (Task): Boolean</i>	Vérifier si la fille d'attente est vide Vérifier si la ressource est disponible Voir les conditions de la politique de service	Bloquer (): tâche dans la fille d'attente Exécuter (): immédiatement la tâche
Arriver d'une tâche prioritaire Voir New (Task)	Vérifier la priorité des processus en instance	Exécuter (): la tâche
Concurrence sur les ressources	Vérifier le problème de l'interférence Vérifier l'interblocage Vérifier processus famine <b>BooleanexistConcurrere (P, R)</b>	Exécuter (): la tâche <b>BooleanRun</b> (process pi; politique algorithme)
Personnaliser le mode de gestion de la fille d'attente	Vérifier les conditions d'application de cette stratégie	Appliquer (/ignorer (): la stratégie Recommandation de stratégie d'application
Arrêter un service	Vérifie d'autre service prioritaire La fin de temps de service Modifie le service	Modifie (/annuler (/exécuter ())
Ajouter une contrainte de ressources	La disponibilité de ressource La fonctionnalité	Appliquer ()
Ajouter une contrainte temporelle	Le service Suspendu pour developper	Développer ()

Tableau III.1 : synthèse de ECA.

La formalisation des ces règles est illustrés dans la *Figure III.5 suivants* :

Nom	ONEVENT	CONDITION	ACTION	TRIGGER
R1	prédécesseur-tâche	true	trouver la tâche	tâche arrive
R2	accès ressource	false	autoriser l'accès aux ressource	besion ressource
R3	evenment asynchrons plustot	true	planifier la tâche	interruption
R4	evenment asynchrons plustard	true	planifier la tâche	interruption
R5	contrainte temporelle strict	true	planifier la tâche	interruption
R6	choix algorithme	true	selection algorithme	selectionalgorithme
R7	algoPSO	true	traitement des tâches	fin tâche
R8	algoMin-min	true	traitement des tâches	fin tâche
R9	algoRoundRobin	true	traitement des tâches	fin tâche
R10	algoMax-min	true	triatement des tâches	fin tâche

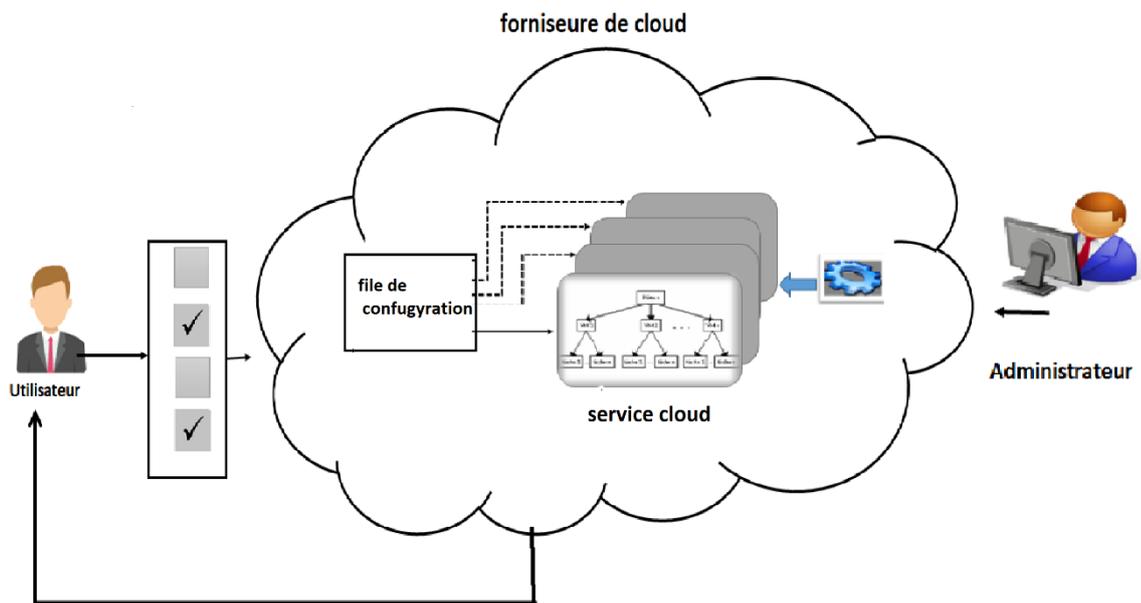
**Figure III.5 : Tables des Règles ECA.**

### 3.2.3 Configuration des services

Afin que la gestion de ce processus métier assure une variabilité et une flexibilité ; nous avons proposé un mécanisme qui permet aux utilisateurs de configurer le service en fonction de besoin en se basant sur les règles ECA.

La configuration est basée sur les opérations suivantes :

- **Include** : Signifier que la fonctionnalité doit être inclus dans le processus métier de l'ordonnancement
- **Exclude** : Signifier que la fonctionnalité doit être non inclus dans le processus métier de l'ordonnancement
- **Obligatoire** : signifier que le processus doit respecter une contrainte obligatoire comme par exemple la relation de précédence entre les tâches : la tâche i ne commerce que la tâche j est terminé.
- **Optionnelle** : signifier que l'usage de la fonctionnalité est optionnel dans le processus par exemple comme la propriété de l'précédence préemptif ou non préemptif



FigureIII.6 : Montre le mécanisme de la configuration de ce service.

### 3.2.4 Algorithmes d'ordonnancement

Le processus de l'ordonnancement est piloté par quatre algorithmes principales :

- **Algorithme de Min-Min**

L'utilisation efficace des ressources peut être augmentée en utilisant un algorithme d'équilibrage de charge. Ceci est réalisé en utilisant les ressources d'un processeur inutilisé (inactif) tout en libérant les ressources des processeurs ayant une charge importante.

L'algorithme d'équilibrage de charge distribue la charge parmi toutes les ressources disponibles. L'algorithme Min-min commence par un ensemble de tâches. Il y a un ensemble de tâches  $T$ . Avec l'ensemble des tâches, il y a aussi des ressources. Après l'implémentation de l'algorithme, l'ensemble de tâches sera alloué ou mappé au jeu de ressources. L'algorithme choisit la tâche ayant la taille minimum. La ressource sera allouée à la tâche ayant le temps d'achèvement minimum. Une fois l'affectation terminée, la tâche sera supprimée de l'ensemble de tâches.[27]

- **Algorithme de RRS (Round-Robin Scheduling)**

C'est l'un des algorithmes d'ordonnancement les plus anciens, les plus simples, les plus équitables et les plus utilisés, spécialement conçu pour les systèmes à temps partagé. Une petite unité de temps, appelée tranches de temps ou quantum est définie. Tous les processus exécutables sont conservés dans une file d'attente circulaire. Le planificateur de CPU contourne cette file d'attente, en allouant le processeur à chaque processus pour un intervalle de temps d'un quantum. De nouveaux processus sont ajoutés à la queue de la file d'attente. Le planificateur CPU sélectionne le premier processus de la file d'attente, met un temporisateur à interrompre après un quantum et distribue le processus. Si le processus est toujours en cours à la fin du quantum, le processeur est

préempté et le processus est ajouté à la queue de la file d'attente. Si le processus se termine avant la fin du quantum, le processus libère volontairement le CPU. [28]

- **Algorithme "PSO" Particle Swarm Optimization**

Particle Swarm Optimization (PSO) ou Optimisation de l'essaim de particules (en français) est un algorithme intelligent basé sur la température, influencé par le comportement social des animaux, comme un troupeau d'oiseaux trouvant une source de nourriture ou une école se protégeant contre les prédateurs. Une particule dans PSO est analogue à un oiseau ou un poisson volant à travers un espace de recherche (problème). Le mouvement de chaque particule est coordonné par une vitesse qui a à la fois la grandeur et la direction. Chaque position de particule à chaque instant est influencée par sa meilleure position et la position de la meilleure particule dans un espace problématique. La performance d'une particule est mesurée par une valeur d'aptitude, qui est spécifique au problème. L'algorithme PSO est similaire à d'autres algorithmes à évolution. Dans PSO, la population est le nombre de particules dans un espace de problèmes. Les particules sont initialisées de manière aléatoire. Chaque particule a une valeur de comportement qui sera évaluée par une fonction d'aptitude à optimiser à chaque génération. Chaque particule connaît sa meilleure position et la meilleure position jusqu'ici parmi le groupe entier de particules. Le meilleur résultat d'une particule est le meilleur résultat (valeur d'aptitude) atteint jusqu'à présent par la particule. [29]

- **Algorithme de Enhanced Max-Min**

Parfois, la tâche la plus importante est trop grande par rapport aux autres tâches dans Mata-task, dans ce genre de cas, le makespan (temps d'exécution total) global est augmenté car une tâche trop importante est exécutée par la ressource la plus lente en premier tandis que les autres tâches sont exécutées par une ressource plus rapide ou lorsqu'il existe une différence majeure entre la ressource la plus lente et la plus rapide équilibre dans le contexte de la vitesse de traitement ou de la bande passante cette tâche la plus importante est exécutée par la cause de ressource la plus lente augmentant dans makespan et le déséquilibre de charge à travers ressources. Par conséquent, au lieu de sélectionner la tâche la plus grande si nous sélectionnons la tâche moyenne ou la plus proche supérieure à la moyenne, alors le makespan est réduit et la charge entre les ressources. [30]

### 3.3 Intégration de notre processus dans le Cloud

Afin d'intégrer notre processus métier dans un service cloud (par ex CloudSim) ; la première étape consiste à expliciter les fonctionnalités, les opérations, les contraintes et la qualité de service liées à ce processus métier. Après la description détaillée de ce processus métier, tous les éléments (fonctionnalités opération etc.) doivent être déployés sous forme des tâches qui assure le service à distance pour les différents acteurs : client cloud ou bien un développeur Cloud, Administrateur etc.

L'intégration de ce processus métier est effectué dans un cadre organisé sous forme des package d'une manière modulaire de telle sorte chaque module dépend d'un autre module avec différentes relations de dépendance : requière ; include etc.

La figure suivante illustre le mécanisme détaillé de ce processus d'intégration

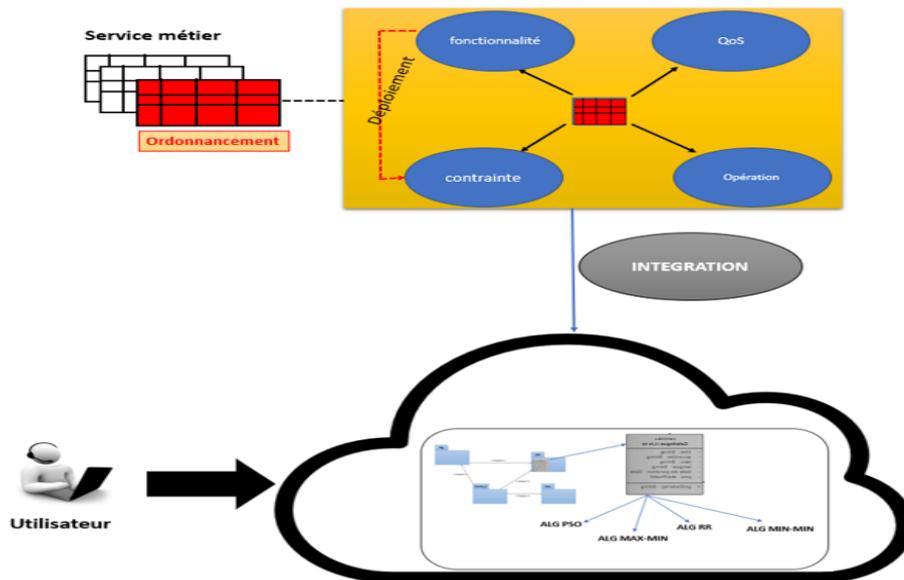


Figure III.7 : illustre le mécanisme détaillé de ce processus d'intégration.

## 4. Implémentation de notre système

Dans cette section on va présenter notre implémentation.

### 4.1. Outils de développement

Cette partie est consacrée pour la présentation et l'argumentation de nos choix des principales technologies et outils que nous avons utilisés pour réaliser notre application.

#### 4.1.1. Le langage de programmation JAVA

Java est un langage de programmation et une plate-forme informatique qui ont été créés par Sun Microsystems en 1995. Beaucoup d'applications et de sites Web ne fonctionnent pas si Java n'est pas installé et leur nombre ne cesse de croître chaque jour. Java est rapide, sécurisé et fiable. Des ordinateurs portables aux centres de données, des consoles de jeux aux superordinateurs scientifiques, des téléphones portables à Internet, la technologie Java est présente sur tous les fronts. [31]

#### 4.1.2. NetBeans

NetBeans est un environnement de développement intégré (IDE) pour Java, placé en open source par Sun en juin 2000 sous licence CDDL (Common Développement and Distribution License). En plus de Java, NetBeans permet également de supporter différents autres langages, comme Python, C, C++, XML 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). NetBeans est disponible sous Windows, Linux, Solaris (sur x86 et SPARC), Mac OS X et Open VMS. [32]

### 4.1.3 Pourquoi la simulation ?

Dans un système distribué, il existe des enjeux à résoudre tels que la gestion et l'ordonnancement des applications car, ces tâches sont compliquées et il n'existe pas une solution optimale pour répondre à ces issues. D'autre part, dans l'environnement d'un système distribué comme le Cloud, il est difficile d'effectuer les différents scénarios avec différent nombres de ressources et utilisateurs afin d'évaluer la performance des algorithmes du partage de charge, Broker, gestion des ressources...etc. Afin de résoudre cette issue, les chercheurs utilisent les simulateurs pour effectuer leur scénario avant de les effectuer au sein d'un système distribué réel. Dans ce qui suit, on va présenter l'un des outils de simulation le plus connu et utilisé dans le domaine de simulation du Cloud [27].

### 4.1.4 CloudSim

Il y a plusieurs outils de simulation de systèmes distribués. Les plus connus sont GridSim, CloudSim, Simgrid. Dans ce qui suit, dans ce mémoire le Framework utiliser est CloudSim, qui permet de la simulation de l'environnement du Cloud Computing. CloudSim est une boîte à outils développée par le laboratoire GRIDS de l'Université de Melbourne. Il fournit une modélisation systémique et comportementale des composants de l'informatique en nuage , Vues architecturales La couche de simulation CloudSim permet de modéliser et de simuler des environnements de datacenters cloud virtualisés, nous sommes également en mesure de modéliser les politiques d'hébergement, de courtage de services, de planification et d'allocation d'une plate-forme Cloud à grande échelle. Ainsi, CloudSim permet la simulation de scénarios modélisant des services IaaS, PaaS et SaaS . Entre tous les outils de simulation en ligne de commande du Cloud Computing [27].

#### 4.1.4.1 Pourquoi le CloudSim ?

CloudSim est un simulateur qui s'appuie sur la recherche et le développement dans le domaine émerge du Cloud Computing. Nous avons opté pour ce simulateur grâce à ses fonctionnalités :

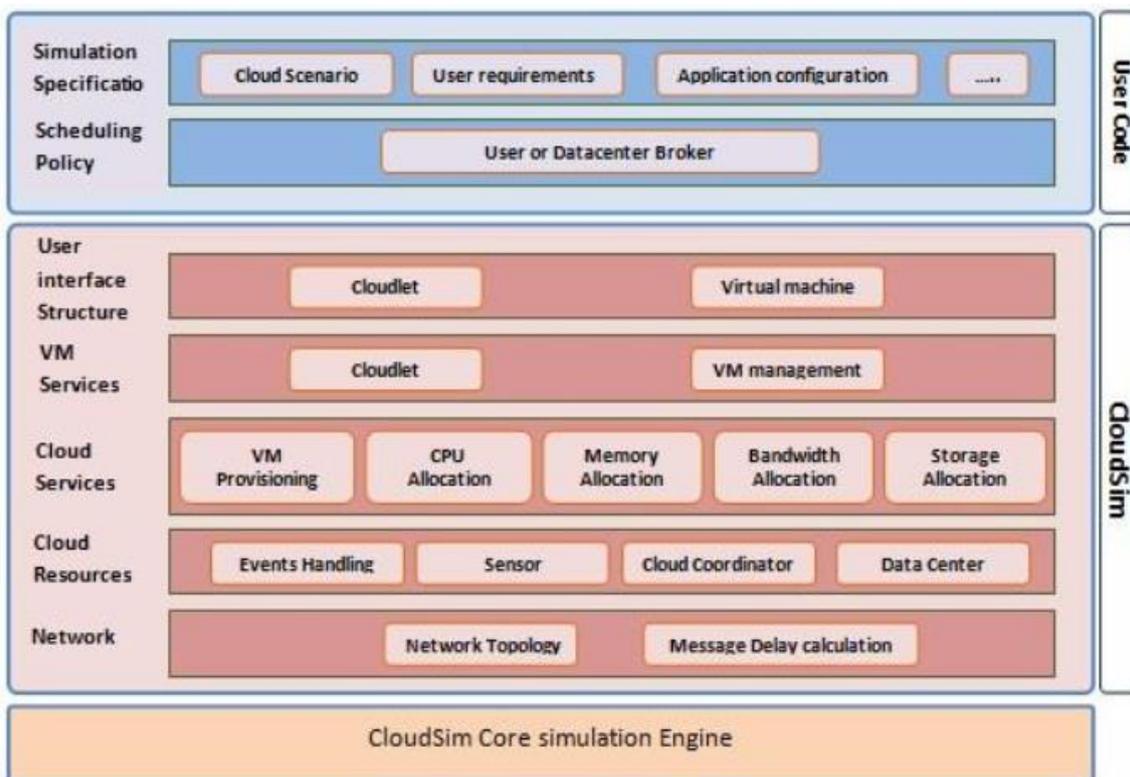
- Un support pour la modélisation et la simulation d'environnement à grande échelle du Cloud Computing y compris les centres des données (Data Centers).
- Une plate-forme autonome pour la modélisation du Cloud Computing des Data Centers, des Brokers, de la planification et des politiques d'allocation des ressources.
- Un support pour la simulation des connexions réseau entre les éléments du système de simulation.
- La disponibilité d'un moteur de virtualisation qui facilite la création et la gestion indépendante des services ainsi que l'hébergement des services virtualisés sur un nœud d'un centre de données.
- La flexibilité pour commuter entre l'allocation en espace partagé et temps partagé et la prise en charge de la répartition des cœurs de traitement aux services virtualisés.
- Simulation de la définition de matériel de centre de données en termes de machines physiques composées de processeur, de dispositifs de stockage, de mémoire et de largeur de bande interne.
- Simulation des spécifications, de la création et de la destruction de machines virtuelles.

CloudSim est la meilleure boîte à outils. Nous l'avons donc choisi en tant que représentant du simulateur de ligne de commande du cloud computing et comparez-le avec les outils graphiques les meilleurs ou les plus familiers de ce schéma.

CloudSim est Framework modélise et simule l'environnement du Cloud Computing et ses services, a été réalisé en Java. Afin que nous puissions simuler cet environnement, il faut connaître l'architecture de ce Framework [27].

#### 4.1.4.2 Architecture générale de CloudSim

La *Figure III.8* montre une architecture multicouche de la structure logicielle CloudSim et ses composantes. La couche CloudSim fournit un support pour la modélisation et la simulation des Data Centers dans l'environnement des clouds computing y compris les interfaces de gestion dédiées aux machines virtuelles, la mémoire, le stockage et la bande passante. L'affectation des VMs à des hôtes, la gestion d'exécution des applications et le suivi de l'état du système dynamique sont traités par cette couche. Un fournisseur Cloud doit implémenter ses stratégies dans cette couche afin d'étudier l'efficacité des différentes politiques qui permettent l'attribution des VMs à ses hôtes. Au niveau supérieur de la couche CloudSim, nous avons le code utilisateur (user code) fournissant les entités de base pour les hôtes (nombre de machines, leur spécification et ainsi de suite), les applications (nombre de tâches et leurs exigences), VMs, nombre d'utilisateurs, types d'application et les politiques d'ordonnancement du Broker. Un développeur d'applications Cloud peut générer plusieurs activités parmi lesquelles nous citons : Des Scénarios de disponibilité des modèles clouds, effectuer des tests robustes basés sur les configurations personnalisées supportées par le CloudSim et implémenter des techniques de provisionnement des applications personnalisées dans les clouds.



FigureIII.8 : Architecture générale de CloudSim.

### 4.1.4.3 Les classes de CloudSim

Le simulateur CloudSim est composé de plusieurs classes que nous pouvons classer en deux catégories : des classes qui modélisent les entités comme le Data Center, le Broker, etc. et des classes modélisant les politiques d'allocation.

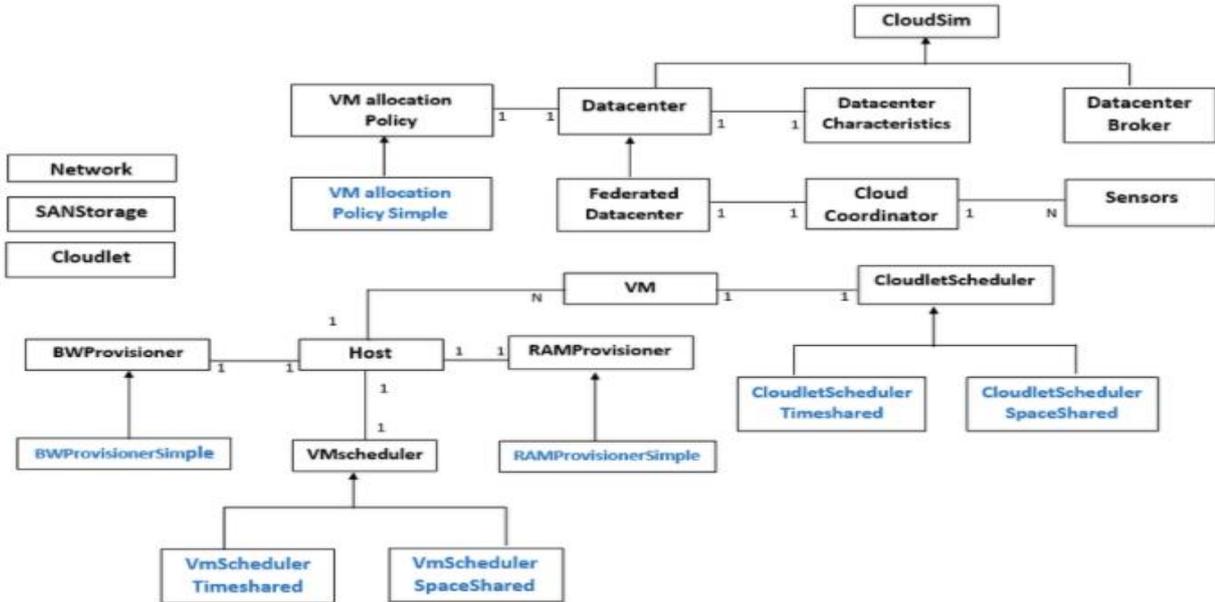


Figure III.9 : Les classes du CloudSim.

### 4.2 La structure du système

L'aspect structurel du système, elle contient deux interfaces, La première interface est affichée en rouge (NewJframe2) et hérite de (NewJframe).

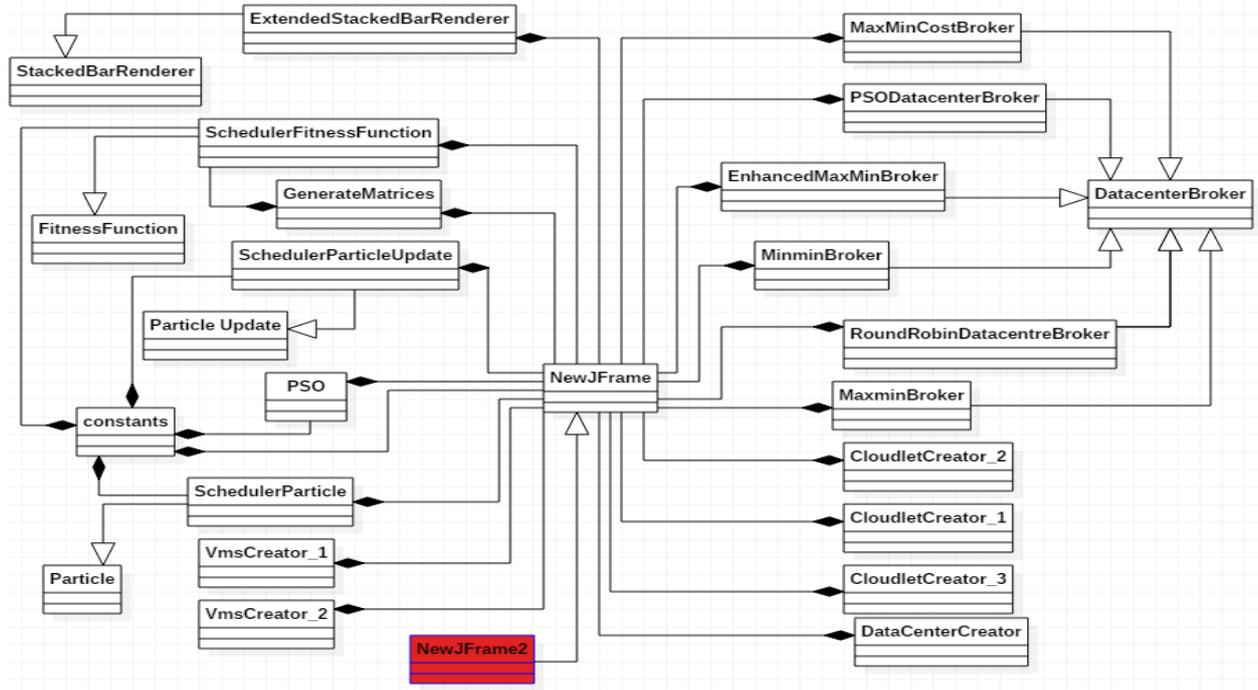


Figure III.10 : structure de notre système.

### 4.3 La description des classes d'application

Nous avons synthétisé le rôle quelques classes corps de la plateforme CloudSim.

Le tableau ulster les rôles de chaque classe

Classe	Description
EnhancedMaxMinBroker	Il s'agit d'une classe qui construit un service cloud et crée des ordinateurs virtuels et un courtier (Broker) qui planifie les tâches sur les machines virtuelles selon PAPS (Premier arrivé / Premier servi) Politique de planification
MaxMinCostBroker	Un courtier (Broker) qui planifie les tâches sur les machines virtuelles conformément à la politique de planification PAPS (FCFS)
MaxminBroker	Trouver le processus de l'algorithme Maxmin qui ont déjà dans le cloudlet
MinminBroker	Trouver le processus de l'algorithme Minmin qui ont déjà dans le cloudlet
PSO	Le fonctionnement de l'algorithme PSO
PSODatacenterBroker	Choisi une VM dans le datacenter par le broker pour l'algorithme
RoundRobinDatacentreBroker	Choisi une VM dans le datacenter par le broker pour l'algorithme
CloudletCreator_1	Crée des Cloudlets selon les exigences de l'utilisateur
CloudletCreator_2	Crée des Cloudlets selon les exigences de l'utilisateur
CloudletCreator_3	Crée des Cloudlets selon les exigences de l'utilisateur
Constants	Nombre de Cloudlets/nombre de Datacenters/nombre de particules
DataCenterCreator	Crée un centre de données selon les exigences de l'utilisateur
VmsCreator_1	Crée des listes de VM selon les exigences de l'utilisateur
VmsCreator_2	Crée des listes de VM selon les exigences de l'utilisateur

*Tableau III.2 : description des classes De notre application.*

### 4.4 Interfaces de l'application

La figure suivante nous montre après avoir exécuté le programme, et il contient quatre boutons (Afficher, OK, Aucun, Tout). Le bouton (Tout) sélectionne toutes les règles Le bouton (Aucun) ne sélectionne aucune règle.



Figure III.11 : La fenêtre principale de notre application.

Lorsqu'on clique sur sur le bouton (Afficher), la liste des règles s'affiche

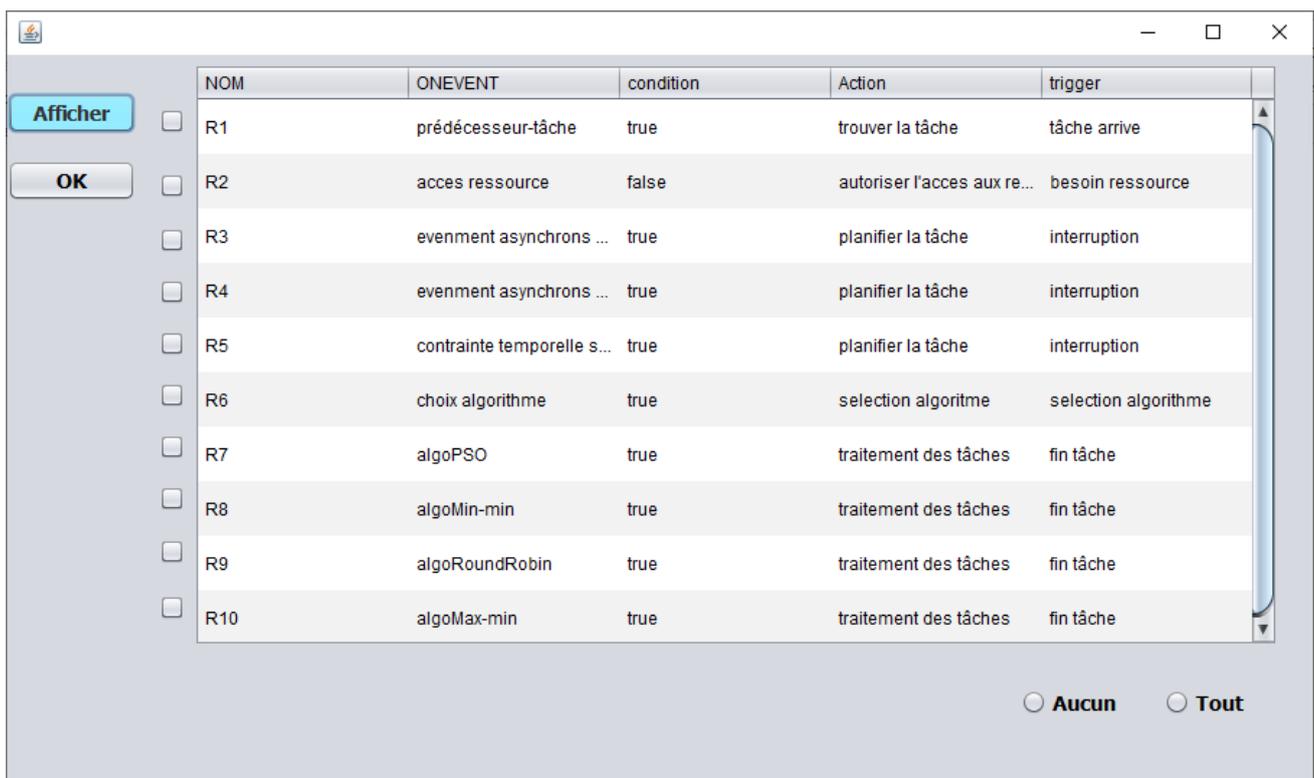
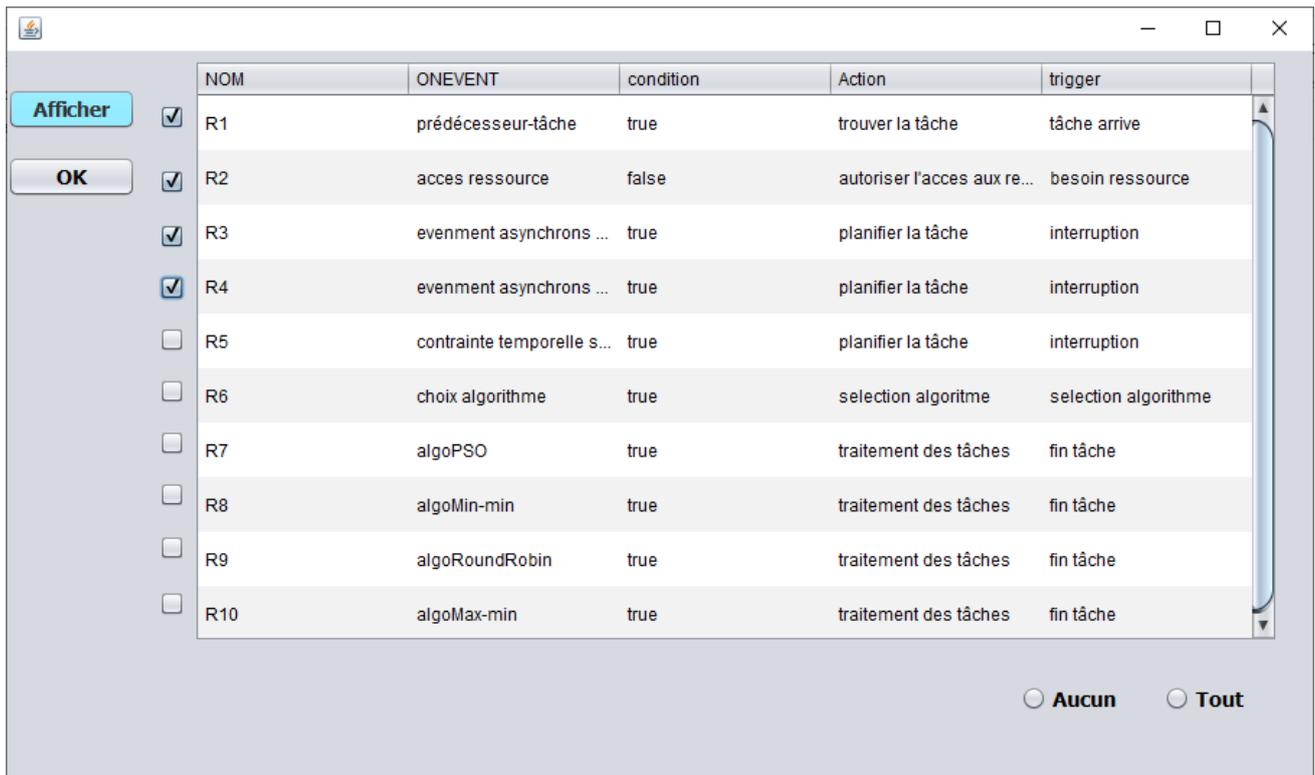


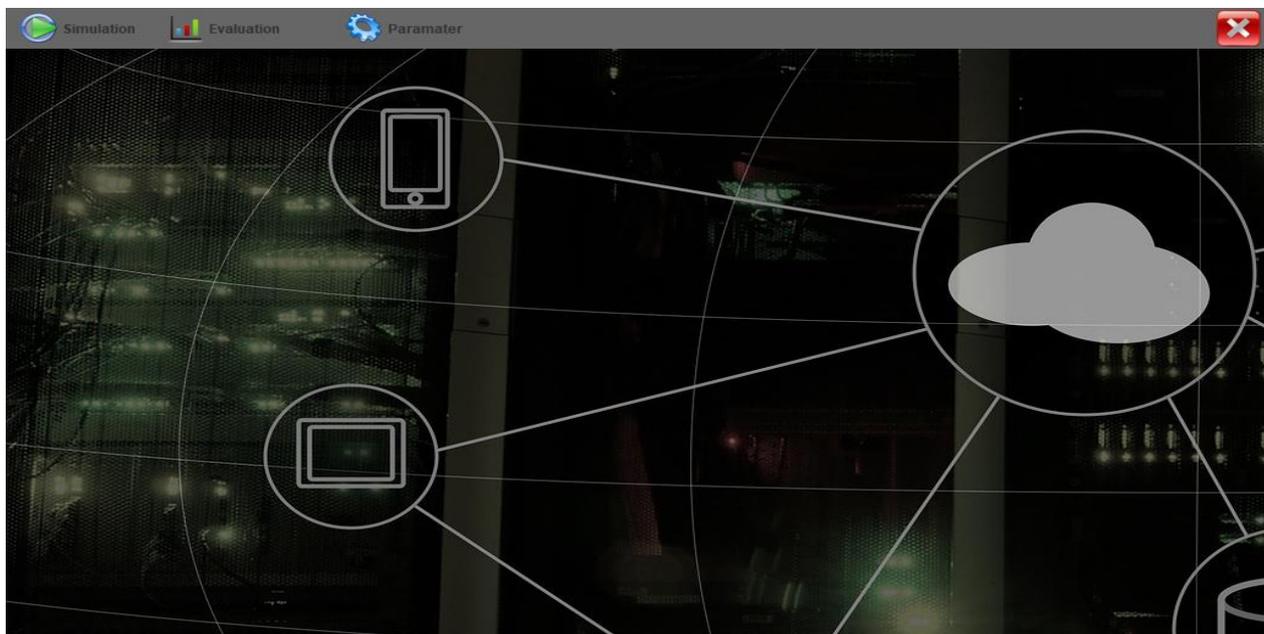
Figure III.12 : Listes des Règles ECA.

Nous sélectionnons les règles pour mettre rendre l'exécution flexible du process d'ordonnancement.



**Figure III.13 : Les Règles exécuter.**

La figure suivante représente l'interface globale de notre application développée, elle contient toutes les fonctionnalités de notre application y compris les boutons "Simulation", "Evaluation" et "Paramètre".



**Figure III.14 : L'interface globale.**

Lorsque on clique sur le bouton de Paramètre, la fenêtre des listes des tâches et des VMs s'affiche.



Figure III.15 : Listes des tâches et des VMs.

Quand on clique sur le bouton " Simulation ", les simulations de chaque algorithme s'affichent ci-dessous :



Figure III.16: Les simulations des politiques.

Lorsqu'on clique sur "Actualiser" la simulation va réinitialisée, le résultat de chaque politique (ex : Min-min pour 10 tâches (Task) et 04 VM) s'affiche lorsque on clique sur le bouton « A ».

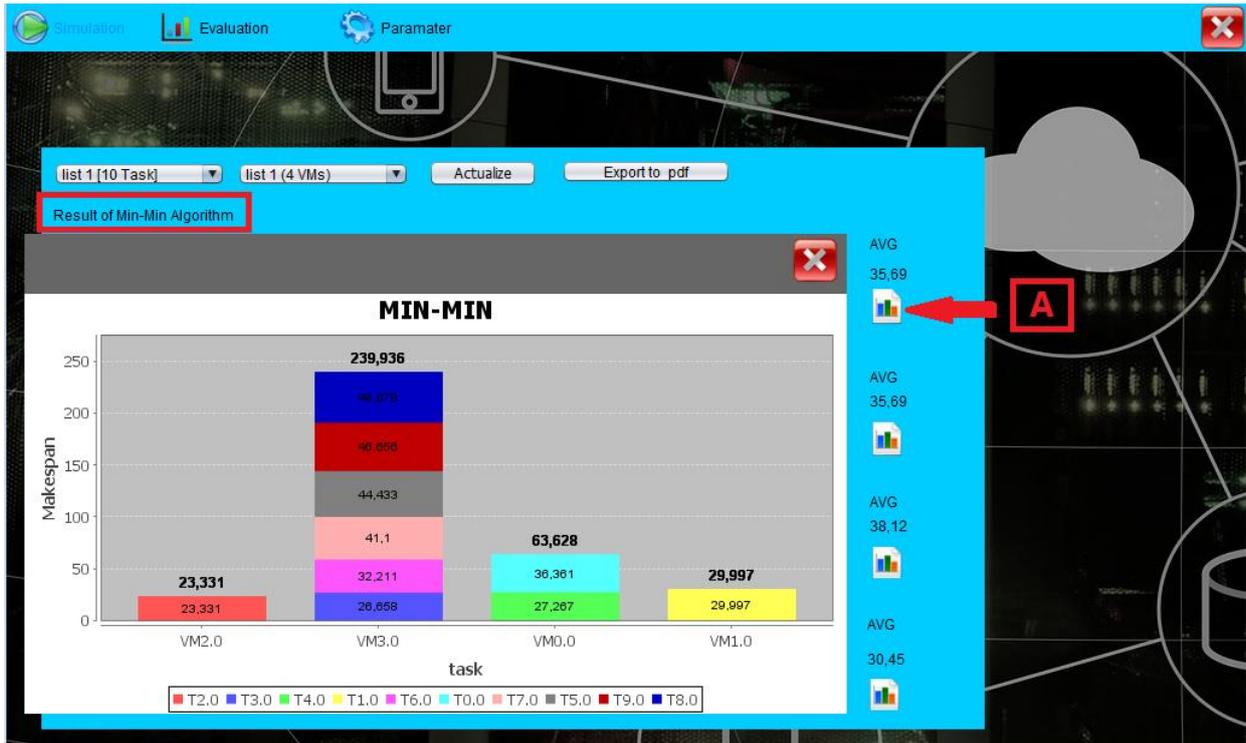


Figure III.17: Résultat de la politique Min-Min.

La politique Min-min choisit les tâches ayant la taille minimum. La ressource (CPU) sera allouée à la tâche ayant le temps d'achèvement minimum. Et après nous pouvons exporter le résultat si on clique sur Exporter en PDF.

La figure suivante calcule la moyenne de chaque politique avec différents nombres des paramètres (tâches et VMs).

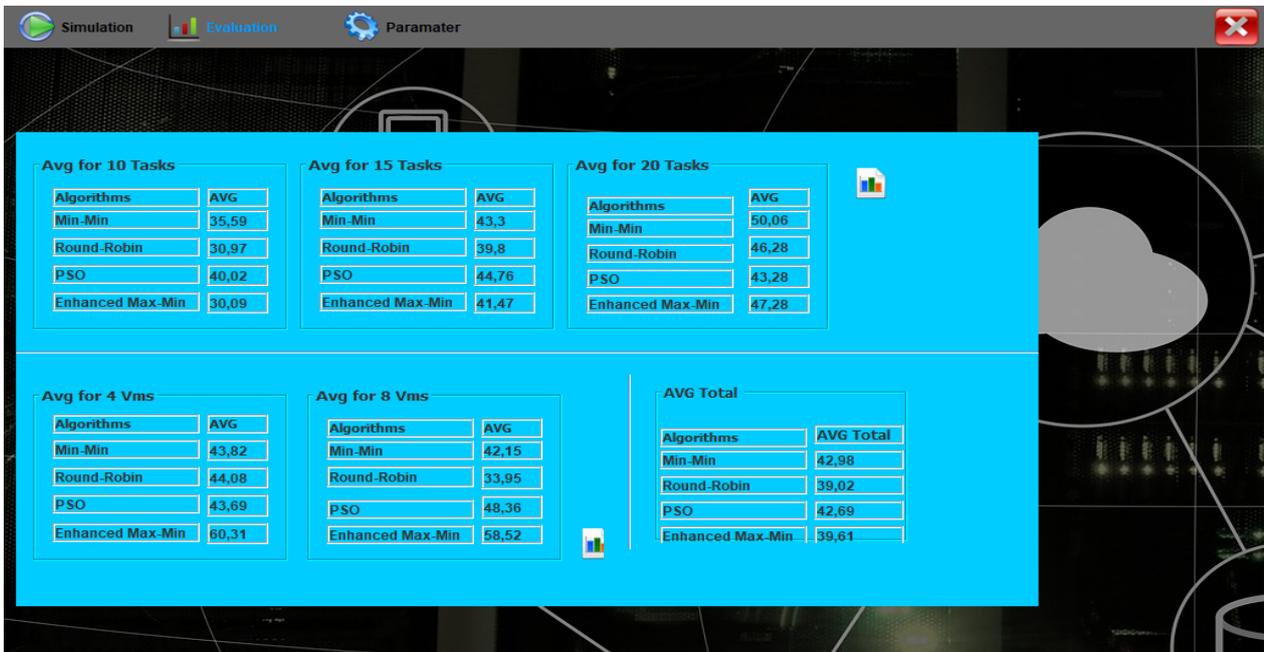


Figure III.18: Moyenne de chaque politique.

## 5. Conclusion

Dans ce chapitre, nous avons présenté notre approche sur un exemple d'ordonnancement et décrit le Framework CloudSim utilisé pour implémenter notre approche. Grâce aux règles ECA, nous avons pu rendre notre processus d'ordonnancement flexible et les résultats sont également conformes aux règles choisies.

# CONCLUSION GÉNÉRALE

## Conclusion générale

Dans ce mémoire nous avons modélisé le processus métier de l'ordonnancement d'une manière flexible et adaptable de tel sorte l'utilisateur peut facilement décrire et modifier l'exécution des processus. La flexibilité est devenue l'intérêt majeur des entreprises et cela pour améliorer leur productivité, leur fiabilité et leur rapidité d'adaptation aux changements. Dans la modélisation, la logique du processus est décrite par un ensemble de règles. Toutefois, parmi ces règles, le formalisme ECA s'avère être le plus adapté pour améliorer la flexibilité d'un processus métier car ces règles permettent de spécifier le flux de contrôle d'un processus d'une manière flexible en utilisant les événements. Et en plus, les règles ECA sont faciles à maintenir et permettent d'intégrer tous les types de règles (contraintes, déviations, productions, et transformations). Pour l'implémentation, nous avons utilisé le Framework CloudSim.

Ce travail ouvre plusieurs perspectives entre eux nous pouvons citer :

- Configurer l'exécution des autres algorithmes d'ordonnancement par les règles ECA.
- Développer la partie « Front-End » par l'utilisateur finaux (utilisateur de cloud).
- Tester notre solution avec d'autres plateformes cloud.
- Ajouter une partie vérification de la modélisation basée sur les règles.

### Bibliographie

- [1] Zohra S, « Contribution à la modélisation et à la vérification de processus Workflow », Thèse de doctorat, Paris, Nov 2010 page(16,25).
- [2] Raida ElMansouri, « Modélisation et Vérification des processus métiers dans les entreprises virtuelles : Une approche basée sur la transformation de graphes », Thèse de doctorat, Université Mentouri Constantine, 2010 page(9,13,14,15,24,25,40).
- [3] Chantal M, Marie B F et Yves G, « Processus métiers et systèmes d'information Gouvernance, management, modélisation », Dunod 3ème édition, 2011 page(49,198,185).
- [4] P. David T. Bourgeois, «Information Systems for Business and Beyond», Syalor foundations, 2014 page(85).
- [5] M. Weske, « Business process Management », Springer, 2007.
- [6] Mounira Zerari, « Une approche pour l'amélioration de la flexibilité des processus métier basée sur les techniques du process mining », Thèse de doctorat , Université Mentouri - Constantine Faculté des Sciences et de l'ingénieur,2012 page(25,11).
- [7] Mohamed Boukhebouze, « Gestion de changement et vérification formelle de processus métier : une approche orientée règle », Thèse de doctorat, Ecole Doctorale Infomaths de Lyon, 2010 page(28,29,79,61,62,59).
- [8] Makhloufi D, « Conception et mise en œuvre d'un outil de modélisation et de vérification des processus métiers flexibles », Mémoire de master, Master GI, Université Ibn Khaldoune, Tiaret 2013 page(6,5,11,7,9,13).
- [9] Amen S, « Modélisation centrée sur les processus métier pour la génération complète de portails collaboratifs », Thèse de doctorat, Université Lille 1, 2013 page(27).
- [10] DRISSI Meriem, « Génération des réseaux de Pétri à partir Des modèles OSSAD », Mémoire de Master, université Abdelhamid ibn badis Mostaganem 2016 page(27).
- [11] Xabier H, « Extensions de BPMN 2.0 et méthode de gestion de la qualité pour l'interopérabilité des données » Thèse de doctorat, Université de bordeaux 2018 page(39,44,32).
- [12] Madani M, « Approche pour le développement des applications orientées services Sécurisées », Mémoire de Master, Université Ibn Khaldoune, Tiaret 2010 .
- [13] Bekki K, Belbachir H, « Towards a Flexible and Adaptable Modeling of Business Processes », International Journal of Information Technology and Web Engineering, pp. 57-67, Avril-Juin 2011.
- [14] Brahim S, « Conception et mise en œuvre d'un outil de vérification des workflows », Mémoire de Master, Université Ibn khaldoune, Tiaret, 2011 page(36).
- [15] Mohammed Oussama Kherbouche, « Contributions à la gestion de l'évolution des processus métier » Thèse de doctorat, Université du Littoral Côte d'Opale ,2013 page(50,58,59).
- [18] The NIST Definition of Cloud Computing, sur NIST Special Publication 800-145. Par Peter Mell, Timothy Grance September 2011 page(2,3).

## Bibliographie

---

- [19] Cisco Cloud Computing Data Center Strategy, Architecture, and Solutions, 1er edition 2009, par cisco systems. page(4).
- [20] LANANI S, « Une approche BPM (Business Process Managment) par composition d'applications dans le cloud computing », Diplôme de Magister Université Mohamed khider de Biskra 2015 page(47,48,49,50).
- [21] Khaled K, « formalisation et évaluation de stratégies d'élasticité multi-couches dans Le cloud », Thèse de doctorat, Université de Pau et des Pays de l'Adour et Université Constantine 2 – Abdelhamid Mehri 2019 page(12,14,15).
- [22] YAGOUB M, « Une approche basée agent pour la sécurité dans le Cloud Computing », Thèse de doctorat, Université Mohamed khider, Biskra 2019 page(15,14).
- [23] DJEBBAR E, « Optimisation d'ordonnancement et d'allocation de ressources dans les cloud computing », Thèse de doctorat, Université Ahmed Ben Bella, Oran 2016 page(15,16).
- [24] BOUZARA Ayoub, « Sécurité dans le Cloud Computing », Mémoire de Master Université Ibn Khaldoune, Tiaret, 2018 page(21).
- [25] Messabih k et Metahri k, « Déploiement d'application sur une architecture haute performance « Cloud computing » », Mémoire de Master, Université Ibn Khaldoune, Tiaret, 2016 page(29).
- [26] MANSOUR K et SAFIR M, « Etat de l'art sur le Cloud Computing : Etude comparative et mise en place d'une solution cloud privé », Mémoire de Master, Université Ibn Khaldoune, Tiaret, 2018 page(21,22).
- [27] HANET Naima, HAMDANI Fatima, « Étude comparative entre le(s) politique(s) de planification des tâches dans les environnements Cloud » Mémoire de Master Université Ibn Khaldoune, Tiaret, 2018 page(28,43).
- [28] R. KAUR et S. KINGER, Analysis of Job Scheduling Algorithms in Cloud Computing, volume 9 number 7, (IJCTT) Mar 2014 page(383)
- [29] A. Toumi, Optimisation par Essaim de Particules : application à la restauration supervisée d'image, UMKBA, 2012.
- [30] U. BHOI et T PURVI N. RAMANUJ, « Enhanced Max-min Task Scheduling Algorithm in Cloud Computing » Volume 2, (IJAIEEM) April 2013
- [31] [https://www.java.com/fr/download/faq/whatis\\_java.xml](https://www.java.com/fr/download/faq/whatis_java.xml) (Consulté le : 07 Décembre 2020)
- [32] <https://doc.ubuntu-fr.org/netbeans> (Consulté le : 07 Décembre 2020)