REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Université Mohamed Khider – BISKRA
Faculté des Sciences Exactes, des Sciences de la Nature et de la Vie

Département d'informatique

N° d'ordre : RTIC../M2/2018

Mémoire

Présenté pour obtenir le diplôme de master académique en

Informatique

Parcours : Réseaux et TIC

Réalisation d'une architecture pour la protection de possession dans les Big Data

---

Par :

SOUFLI MANEL

Soutenu le 24 Juin 2018, devant le jury composé de :

Mouaki Benani Nawel	M C A	Président
SAOULI Hamza	M A B	Rapporteur
Rahmani Salima	M A A	Examinateur

Remerciements

Nous tenons tout d'abord à remercier Dieu le tout puissant et miséricordieux, qui nous a donné la force et la patience d'accomplir ce Modeste travail.
En second lieu, nous tenons à remercier notre encadreur Mr Saouli Hamza pour son précieux conseil et son aide, ses dirigés durant toute la période du travail.
Nos vifs remerciements vont également aux membres du jury pour l'intérêt qu'ils ont porté à notre recherche en acceptant d'examiner notre travail Et de l'enrichir par leurs propositions.
Nous tenons également à remercier toutes les personnes qui ont participé de près ou de loin à la réalisation de ce travail.

Finalement, nous tenons à exprimer notre profonde gratitude à nos familles qui nous ont toujours soutenues et à tout ce qui participe de réaliser ce mémoire.
Enfin, je remercie tous ceux qui ont contribué de près ou de loin à l'aboutissement de ce travail.

Dédicace

Je dédie ce mémoire à :

· Mes parents :

Ma mère, qui a œuvré pour ma réussite, de par son amour, son soutien, tous les sacrifices consentis et ses précieux conseils, pour toute son assistance et sa présence dans ma vie, reçois à travers ce travail aussi modeste soit-il, l'expression de mes sentiments et de mon éternelle gratitude.

Mon père, qui peut être fier et trouver ici le résultat de longues années de sacrifices et de privations pour m'aider à avancer dans la vie. Puisse Dieu faire en sorte que ce travail porte son fruit ; Merci pour les valeurs nobles, l'éducation et le soutient permanent venu de toi.

Mes sœurs Meriem, Sara et ma petite Iman et mon cher frère qui n'ont cessé d'être pour moi des exemples de persévérance, de courage et de générosité.

Ma grand-mère

Mes chères amies Abir, Hadjer, Linda,Zineb…...

A celui que j'aime beaucoup et qui m'a soutenue tout au long de ce projet, mon fiancé.

Je vous dis merci

Résumé

L'explosion quantitative des données numériques a obligé les chercheurs à trouver de nouvelles manières de voir et d'analyser le monde. Il s'agit de découvrir de nouveaux ordres de grandeur concernant la capture, la recherche, le partage, le stockage, l'analyse et la présentation des données. Ainsi est né le « Big Data ». Il s'agit d'un concept permettant de stocker un nombre indicible d'informations sur une base numérique.

Le terme Big Data signifie méga données, grosses données ou encore données massives. Ils désignent un ensemble très volumineux de données qu'aucun outil classique de gestion de base de données ou de gestion de l'information ne peut vraiment travailler.

Le présent travail traite l'aspect sécurité des Big Data et plus précisément la partie possession de la donnée qui représente un défi vu le volume important des données. Pour ce faire nous avons tenté de mettre en lumière les inconvénients des autres travaux de recherche afin de présenter des solutions susceptibles de combler les défaillances observées. Ensuite, nous avons proposé une architecture de vérification de la possession, cette architecture est constituée d'un ensemble de composants, chacun d'eux contribue au rehaussement du niveau de sécurisation du système.

Mots-clés: Big Data – Possession – Architecture – Travaux connexes – Hadoop – Cloud

Table de matière

Chapitre 4 : Conception et modélisation

Liste de figures

Liste des tableaux

.

Chapitre1 :Introduction Générale

1. Contexte du travail

Chaque jour, nous générons 2,5 trillions d'octets de données. A tel point que 90% des données dans le monde ont été créées au cours des deux dernières années seulement. Ces données provenant de partout : de capteurs utilisés pour collecter les informations climatiques, de messages sur les sites de médias sociaux, d'images numériques et de 4déos publiées en ligne, d'enregistrements transactionnels d'achats en ligne et de signaux GPS de téléphones mobiles, pour ne citer que quelques sources. Cet accroissement de volume est principalement lié, dans les secteurs de la banque, de l'assurance ou encore des opérateurs, à la volonté de ces derniers de sans cesse mieux connaître leurs clients en croisant l'ensemble des informations disponibles sur celui-ci et sur ces actions quelle qu'en soit l'origine.

Le terme même de Bigdata a été évoqué la première fois par le cabinet d'études Gartner en 2008 mais des traces de la genèse de ce terme remontent à 2001 et ont été évoquées par le cabinet Meta Group. Il fait référence à l'explosion du volume des données (de par leur nombre, la 4tesse à laquelle elles sont produites et leur variété) et aux nouvelles solutions proposées pour gérer cette volumétrie tant dans la capacité à stocker et explorer celles-ci que, récemment, la capacité à analyser et exploiter ces données dans une approche temps réel.

Les questions de sécurité et de confidentialité des données mettent nos entreprises au défi de répondre à des problèmes différents mais complémentaires et tout aussi critiques. En effet, la sécurité des données régit l'accès aux données pendant tout le cycle de 4e des données alors que la confidentialité des données définit cet accès en fonction des politiques de confidentialité et des lois — lesquelles déterminent, par exemple, qui peut consulter des données personnelles, financières, médicales ou confidentielles.

2. Problématique et objectifs

En cas d'un désastre naturel ou causé par l'‘être humain ou en cas de faille sécuritaire ou attaque cybernétique sur un fournisseur de ser4ces de stockage d'information :

1- le fournisseur et le client auront besoin de vérifier si les des données qui on été stocké sont toujours enregistré et stocké sur les Datacenter du fournisseur de ser4ce de stockage .... mais il est très couteux (en terme de ressources (machine 4rtuel, cpu, Ram ... etc.)) de parcourir tous les données sachant qu'il s'agit de Bigdata (grand volume avec une grande vérité de données.
2- en cas de perte d'information le contrat stipule que le client doit être remboursé par le fournisseur selon les termes du contrat.

3- donc on a besoin de prouvé ou de désapprouver la possession des données par le fournisseur après un telle désastre ou attaque cybernétique.

L'objectif de ce travail est de proposer une nouvelle architecture qui prend en compte quelque inconvénient des travaux connexes et réaliser une application pour prouver et vérifier la possession des données dans les BigDatas.

3. Structure du mémoire

Le thème s'inscrit dans ce cadre-là, où nous allons essayer de proposer une architecture pour la protection de possession dans les BigDatas.

Ce Mémoire est organisé en quatre chapitres répartis comme suit :
Dans le premier chapitre, nous présentons un état de l'art sur les BigDatas. Notre propos dans ce chapitre est de donnant certaines définitions qui seront utilisées par la suite, ainsi qu'étudier la Sécurité dans les BigDatas et les défis de cette dernière.
Dans le deuxième chapitre, nous présentons un état de l'art sur les approches et les travaux Connexes, dans lequel nous avons étudié des articles qui proposent des solutions pour la Possession dans les BigDatas sur lesquels nous avons construit une table comparative.
Dans le troisième chapitre, nous allons décrire notre contribution qui est la proposition d'une nouvelle architecture de possession dans les BigDatas.

Dans le quatrième chapitre, nous poursuivons par une présentation des langages de programmation et les outils de développement utilisés pour la mise en œuvre du système.
Nous donnant par la suite une description textuelle et graphique de quelques interfaces du système réalisé puis l'architecture de système, aussi qu'une description des parties de code source. Enfin, nous terminons ce mémoire par une conclusion générale, qui récapitule les travaux réalisés.

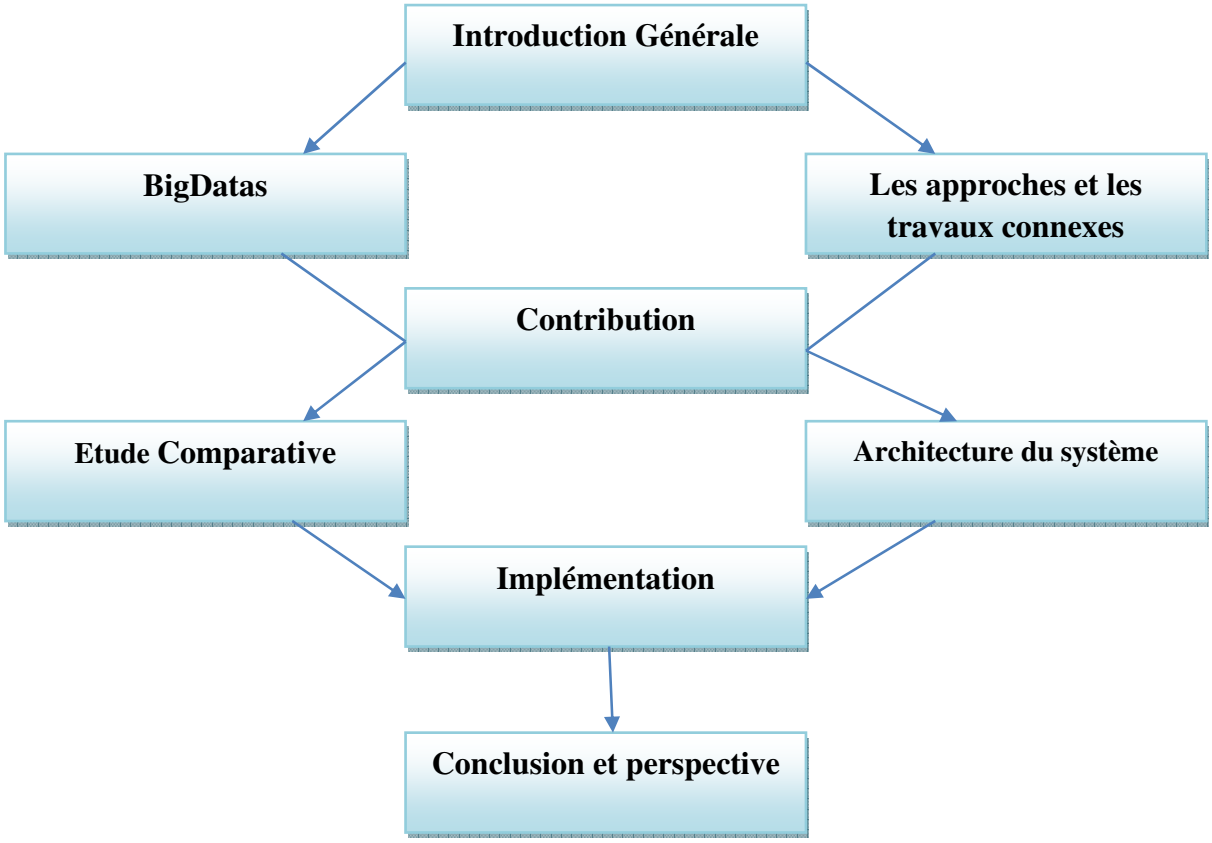

Figure.1 Structure du mémoire

Chapitre 2 :Sécurité des Bigdata

1. Introduction

Dans le déploiement du Bigdata, la sécurité reste le « principal défi », pour ne pas dire obstacle, Aujourd'hui, c'est toujours le cas. La sécurité des projets Bigdata est toujours invoquée comme un élément important.

Les projets Bigdata se mettent en place. Mais parmi les freins, la sécurité figure au premier rang. Elle ne concerne pas seulement les risques de vol de données ou d'espionnage, mais également la conformité avec la réglementation sur la protection des données personnelles.

Dans ce chapitre on va présenter des notions générales et d'autres informations ayant rapport à la sécurité en général, au Bigdata en tant que nouvelle technologie et enfin aux exigences sécuritaires imposées par cette dernière.

2. Bigdata

2.1 Définition

Le terme Bigdata signifie un ensemble des données massives, hétérogène et instructurés. Ces données sont difficiles à être traitées avec des outils et des techniques de gestion traditionnels. Donc il est nécessaire d'utiliser des outils plus sophistiqués et des cadres pour l'organisation efficaces.

Récemment les entreprises, le milieu universitaire s'intéresser au potentiel élevé de Bigdata.[1]

2.2 Modèle 5V

Le modèle 5v est un modèle qui peut décrire un Bigdata, Ce modèle est une extension de la définition précédemment de Modèle 3V, Et se compose de:[2]

Le Volume : Avec la Génération et la collecte de multitude de données, l'échelle de données de4ent de plus en plus gros. Les données produites aujourd'hui sont estimées dans l'ordre de zettabytes, Et ils augmentent autour de 40% chaque année.

La Vélocité : Signifie la rapidité de la collection et l'analyse des données.

La Variété : Indique les différents types de données qui sont : Données semi-structurées, Données non-structurées, Données traditionnelles structurées.

La Valeur : Les données peuvent être une marchandise. Donc il est important de mentionner que aujourd'hui Les données a des coûts.

La Véracité : Vous devez vous assurer de l'exactitude et la qualité des données

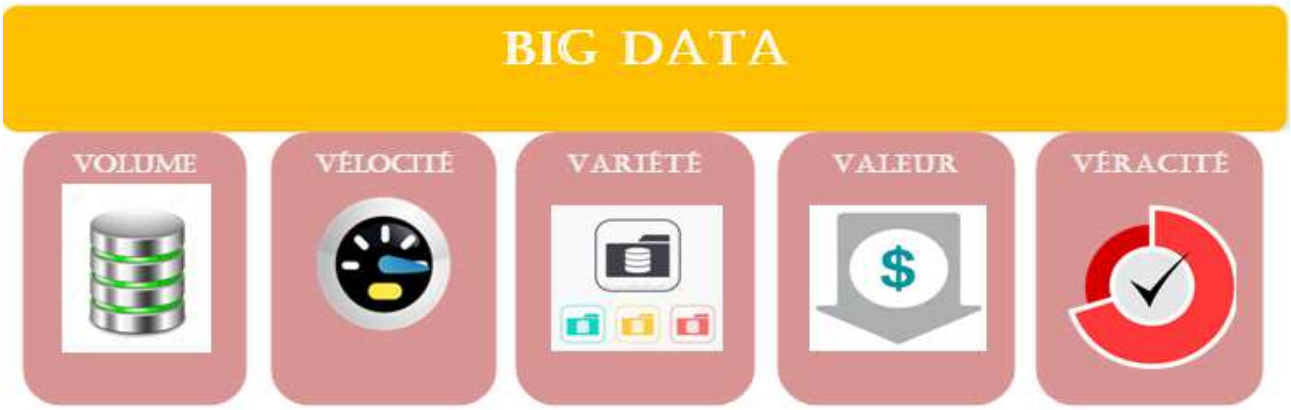

Fig. 1.1 Les 5Vs fonctionnalités de BigDatas

2.3 Méthode de traitement des Bigdata

L'analyse des Bigdata nécessite de nombreuses techniques avancées, Et ils sont les suivants :

- Réseaux de neurones artificiels : Modèles basés sur le principe de l'organisation et du fonctionnement des neurones biologiques.
- Méthodes d'analyse prédictive.
- Les statistiques.
- Traitement du langage naturel.

Les méthodes de traitement des Bigdata englobent différentes Disciplines comme :

- Les mathématiques appliquées.
- Les statistiques.
- L'informatique et l'économie.

Ce sont les bases de techniques d'analyse de données telles que :

- Minutage de données.
- Les réseaux de neurones.
- L'apprentissage par machine.
- Le traitement de signal.

- Les méthodes de 4sualisation.

Ces méthodes sont interconnectées et utilisées simultanément pendant le traitement des données, ce qui augmente considérablement l'utilisation du système. [3]

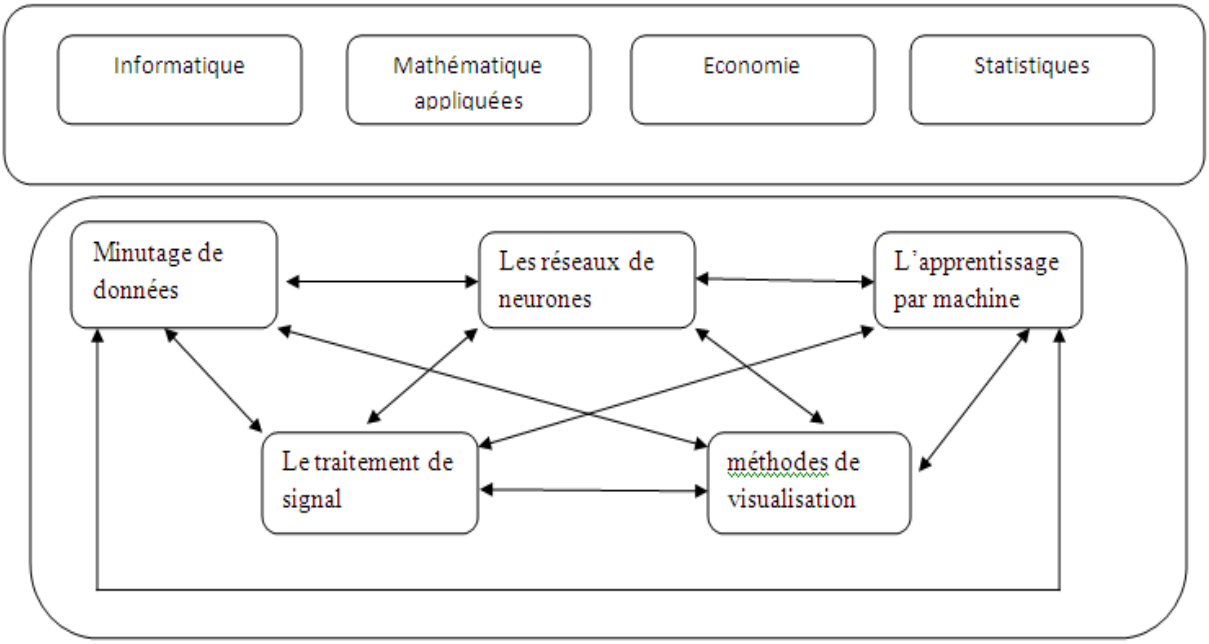

Fig. 1.2 Méthode de traitement des Bigdata

2.4 Qualité des BigDatas

Dix nids de poule sur la qualité de l'information [4]

Les nids de poule	Dimensions DQ affectées
➢ Des sources multiples de la même information produisent Différentes valeurs.	➢ Cohérence et crédibilité.
➢ L'information est produite à l'aide de jugements subjectifs, Conduisant à un biais.	➢ Objecti4té et crédibilité.
➢ Les erreurs systématiques dans la production de l'information entraînent une perte d'information.	➢ Correction, intégralité.
➢ l'accès à une information de4ent plus difficile car le volume des	➢ La représentation concise, ➢ La rapidité, la valeur ajoutée et ➢ Accessibilité.
➢ informations stockées est gros. ➢ Les systèmes hétérogènes distribués entraînent des définitions, des formats	➢ Une représentation cohérente, ➢ Opportunité, valeur ajoutée

et des valeurs incohérents. ➢ L'information non numérique est difficile à indexer	➢ La représentation concise, Valeur ajoutée, accessibilité

Table1. Qualité de bigdata

2.5 Framework d'implémentation

Il existe diverses approches de solutions Big Data. Donc il est nécessaire d'avoir Un (Framework) cadre global uniforme, Ce qui serait utile tant pour les entreprises que pour les fournisseurs pour leur mise en œuvre et leur exécution harmonieuses. Ce cadre se compose de quatre phases de traitement de données, Et sont les suivantes :

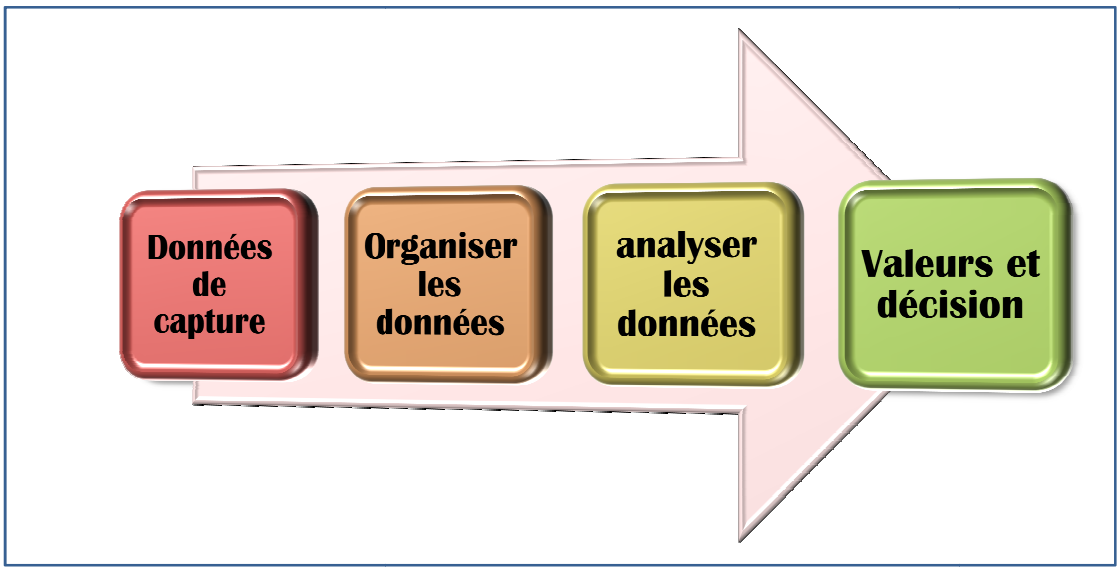

Fig. 1.3 Les phases de traitement de données

2.5.1 Données de capture

Dans cette phase on capture des données des différents systèmes sources. Les données peuvent être de différents types (structuré, semi-structurées et
Non structuré). Selon la variété des données, divers outils comme système de gestion de base de données relationnelle (SGBDR), non seulement SQL (No SQL), DB et Système de fichiers distribués Hadoop (HDFS) peut être utilisé pour stocker des données.
Les différents outils sont : SGBDR,HDFS Hadoop, Not Only SQL

2.5.2 Organiser les données

Dans cette phase Les données sont organisées et di4sé en bases de données relationnelles ou Entrepôt de données en utilisant des outils ou des logiciels comme : Hive, Cloudera.

2.5.3 Analyser les données

Dans cette phase, on utilise les diverses analyses analytiques et d'affaires pour obtenir des données bien organisées. Il existe différents outils analytiques, des programmes R et des outils de BI pour analyser les données de l'entrepôt de données pour extraire des valeurs commerciales.

2.5.4 Valeurs et décisions

Est l'obtention des résultats qui comprennent les rapports d'entreprise, les tableaux de bord, l'analyse ad hoc, les graphiques multidimensionnels, la carte de pointage pour pouvoir prendre des décisions commerciales. [5]

2.6 Domaine d'Application

2.6.1 L'agriculture

L'agriculture est un domaine important pour l'humanité, et pour l'organisation et la gestion de l'agriculture dans le monde entier. Nous avons besoin des bigdata notamment pour la gestion de l'irrigation.

2.6.2 Assurance

Nous avons également besoin de Bigdata dans le domaine de l'assurance en raison de la nécessité de mener des statistiques et d'analyser les risques liés au comportement de millions d'individus.

2.6.3 Marketing

Le marketing nous permet de gérer les énormes masses d'informations provenant de différents sites de réseaux sociaux, des capteurs généraux dans les centres commerciaux, le métro, les aéroports, les universités.

2.6.4 Achat programmatique

De plus en plus, Les gens se tournent vers l'achat programmatique va net, et avec l'utilisation d'une plateforme entre les clients et les fournisseurs Nous avons fait des nombreuses opérations facilement comme la publicité, choix du meilleur prix, payement électronique. Et L'achat programmatique...Etc.

2.6.5 Compétitivité et Innovation de produit

Les entreprises doivent être en mesure (capables) de traiter des grandes données afin de répondre aux besoins de leurs clients et d'accroître leur compétiti4té sur le marché.

2.6.6 Gestion de catastrophes naturelles

Les BigDatas ont la capacité d'analyser les données météorologiques en temps réel et donc de prévoir les conditions météorologiques et d'évter les pertes accidentelles résultant de catastrophes naturelles.

2.6.7 Contrôle d épidémie

Nous pouvons lutter contre les épidémies à l'aide des BigDatas à travers le monde en surveillant la migration des insectes et des rats porteurs des maladies.

2.6.8 Prévention d'attaques cybernétiques

Les techniques d'analyses de données qu'offrent les BigDatas peuvent détecter les intrusions, les failles sécuritaires ainsi que les attaques cybernétiques, Ces techniques nous permettent d'obtenir des schémas relationnels entre les données et effectuer des calculs statistiques qui permettent de surveiller et d'intervenir, en temps réel, sur les menaces et les attaques cybernétiques.

2.6.9 Au-delà du marketing

L'émergence des BigDatas a transformé le monde du marketing. Des nouvelles technologies sont apparues, Des nouvelles techniques ont émergé qui ont facilité le traitement d'énormes masses d'informations tels que les réseaux sociaux, les applications mobiles, les magasins, la télévision, les catalogues, les blogs, la presse, les radios, etc. [6]

Fig. 1.4 Les Domaines d'Application de bigdata

2. 7 Défis et enjeux

- ✓ L'acquisition des données.
- ✓ Le stockage.
- ✓ La gestion des données.
- ✓ L'analyse des données.
- ✓ L'utilisation du matériel est couteuse.
- ✓ La représentation des données.
- ✓ Réduction de redondance et compression de données.
- ✓ Gestion du cycle de vie des données.
- ✓ Confidentialité des données.
- ✓ Gestion de l'énergie.
- ✓ Expendabilité et évolutivité. [7]

3. Sécurité et vie privé

3.1 Sécurité : survole général

Le grand volume d'informations stockées dans les Bigdata doit être protégé, Pour cela plusieurs outils sont développées dans le but de la gestion de ces données massives mais tous ces outils son insuffisants pour la protection complète.

Une solution complète doit répondre à trois exigences :

- La confidentialité : la protection des données contre la divulgation non autorisée.
- L'intégrité : La prévention de la modification non autorisées et des données.
- disponibilité : la prévention et à la récupération des erreurs matérielles et logicielles et des refus négatifs d'accès aux données. [8]

3.2 Les six éléments de sécurité

On a six éléments de sécurité dans le modèle de Framework proposé sont essentiels à la sécurité de l'information. Pour que les informations soient en sécurité il est nécessaire que tous les six éléments être présent.

Et on a six scénarios de pertes d'informations, tous dérivés de cas réels.

3.2.1 La disponibilité

La préservation de la disponibilité doit être acceptée en tant que but de la sécurité de l'information.

La pratique de sécurité ne doit pas se fonder uniquement sur les capacités techniques et les connaissances d'une personne, mais devrait utiliser plusieurs contrôles pour maintenir ou restaurer la disponibilité des données dans l'ordinateur. Il peut être possible de prévenir ou de minimiser la perte grâce à de bonnes pratiques de sauvegarde et à l'utilisation de bons contrôles d'utilisation de l'ordinateur et des données,

La gravité de la perte de disponibilité peut varier. Un auteur peut détruire des copies d'une information de manière à éliminer toute possibilité de récupération. En d'autres situations, les données peuvent être partiellement utilisables, la reprise possible pour une modération.

3.2.2 L'utilité

Dans un tel cas, un employé a crypté systématiquement la seule copie d'informations stockées dans l'ordinateur de son organisation et a ensuite effacé la clé de cryptage accidentellement. L'utilité de l'information a été perdue et ne peut être restaurée que par une cryptanalyse diffuse. Pour préserver l'utilité de l'information, la gestion doit nécessiter des copies de

sauvegarde obligatoires de toutes les informations critiques et devrait contrôler l'utilisation de puissants Mécanismes de protection tels que la cryptographie.

3.2.3 L'intégrité

Plusieurs contrôles peuvent être appliqués pour empêcher la perte d'intégrité de l'information, y compris l'utilisation et la vérification des nombres séquentiels, des sommes de contrôle et / ou des totaux hachis pour assurer L'intégralité et l'intégralité d'une série d'articles.

La gra4té de la perte d'intégrité de l'information varie également. Des parties significatives de l'information peuvent être manquantes ou mal classées (mais toujours disponibles), sans potentiel de récupération. Les informations manquantes ou mal placées peuvent être restaurées, avec un retard et à un coût modéré.

3.2.4 L'authenticité

L'authenticité est le mot qui signifie la conformité à la réalité. Les contrôles qui peuvent être appliqués pour garantir l'authenticité de l'information sont : les transactions confirmant, les noms, les livraisons et les adresses, la validation des produits.

3.2.5 La confidentialité

Les contrôles pour maintenir la dépendance comprennent l'utilisation de la cryptographie, la formation des employés résister aux attaques trompeuses d'ingénierie sociale visant à obtenir leurs connaissances techniques et à contrôler l'utilisation d'ordinateurs et de dispositifs informatiques. Une bonne sécurité exige également que le coût des ressources pour la protection ne dépasse pas la valeur de ce qui peut être perdu, en particulier avec une faible incidence. Par exemple, la protection contre les émanations de radiofréquences dans les guichets automatiques (comme dans ce scénario) n'est probablement pas recommandée, compte tenu du coût du blindage et de la rareté de ces attaques de haute technologie. La gravité de la perte de dépendance peut varier. La perte du scénario le plus défavorable est lorsqu'une partie ayant l'intention et la capacité de causer un préjudice observe les informations sensibles d'une victime.

3.2.6 La possession

La gravité de la perte de la possession peut être variée, dans le cas le moins grave un criminel pourrait prendre des informations pour une certaine période de temps, mais laisser une certaine possibilité pour la récupération à un coût modéré. Mais dans le pire cas un criminel peut prendre des informations, ainsi que toutes les copies de celui-ci, et il peut n'y avoir aucun moyen de récupération.

Nous protégeons la possession de l'information en empêchant les gens de la prise non autorisée, de détenir ou contrôler et de faire des copies, parce que le modèle de sécurité doit inclure la protection de la possession de l'information.

L'utilisation des lois de copyright, la mise en œuvre des limitations physiques et logiques D'utilisation sont des contrôles qui peuvent protéger la possession de l'information. [9]

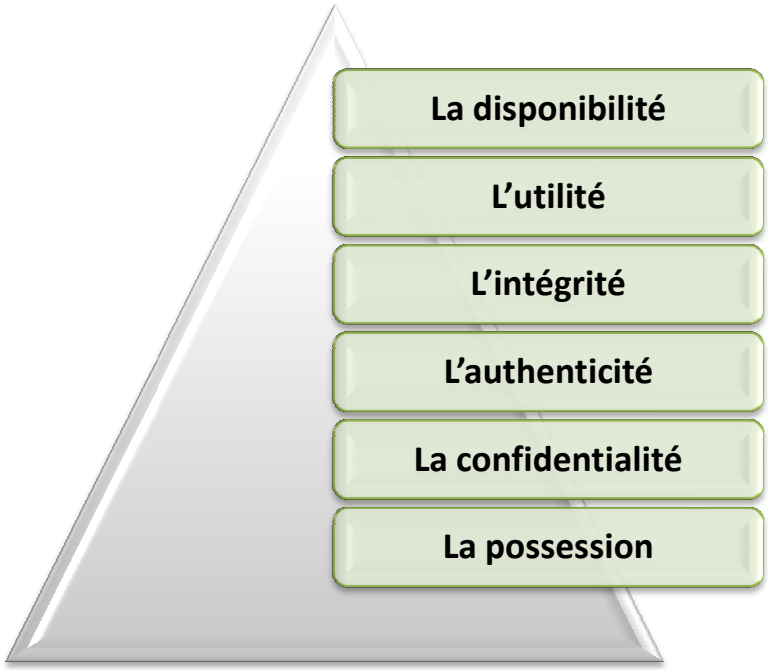

Fig. 1.5 Les six éléments de sécurité

3.3 Les types de sécurité

3.3.1 Sécurité physique

Traite des mesures physiques conçues pour préserver les caractéristiques physiques et les propriétés des systèmes, des espaces, des objets et des êtres humains

3.3.2 Sécurité politique

Traite de la protection des droits acquis, des institutions / structures établies et des choix politiques reconnus

3.3.3 Sécurité socio-économique

Traite des mesures économiques 4sant à protéger le système économique, son développement et son impact sur les individus

3.3.4 Sécurité culturelle

Traite des mesures 4sant à préserver la permanence des schémas traditionnels de la langue, de la culture, des associations, de l'identité et des pratiques religieuses tout en permettant des changements jugés acceptables.

3.3.5 Sécurité environnementale

Traite de mesures conçues pour assurer la sécurité contre les dangers environnementaux causés par des processus naturels ou humains en raison de l'ignorance, de l'accident, de la mauvaise gestion ou de la conception intentionnelle, et se produisent à l'intérieur ou à travers les frontières nationales

3.3.6 Sécurité d'incertitude radicale

S'occupe de mesures visant à assurer la sécurité de la 4olence / des menaces exceptionnelles et rares, qui ne sont pas délibérément infligées par un agent externe ou interne, mais peuvent encore menacer de manière considérable de dégrader la qualité de vie.

3.3.7 Sécurité de l'information

Traite des mesures visant à protéger les systèmes d'information et d'information contre l'accès, l'utilisation, la divulgation, les interruptions, la modification, la lecture, l'inspection, l'enregistrement ou la destruction non autorisés. [10]

3.4 Défis de protection et sécurisation

- L'augmentation des données importantes mener à la complexité de l'analyse et alors le mal fonctionnement de traitement des données.
- Le problème des préoccupations des clients au sujet de leurs données personnelles, et l'utilisation abusive des ce données. Cela oblige les organisations à assurer la protection de la vie privée pour les personnes, et impose des sanctions juridiques font contre tout abus.
- Nous savons que les principes fondamentaux de la protection des données a déjà été établie dans les premiers jours de l'Internet au Royaume-Uni et la loi de l'Union européenne, mais il n'a pas été mis à jour après l'apparition des Bigdata.
- Problème des organisations à acquérir la confiance des utilisateurs de leurs informations personnels.
- Comprendre les caractéristiques des grands volumes de données.
- Identifier les ensembles de données en double.
- Maintenir l'intégrité des données.
- La conception des algorithmes de chiffrement.

- Réduire la surcharge d'information pour les responsables de la sécurité et de l'information.
- Réaliser des comptes fiables dans plusieurs noyaux environnements distribués. [11]

3.5 vie privée : survole général

La vie privée représente tous ce qui est concerne des informations, elle apparaît de plusieurs façons dans la vie quotidienne telle que l'intimité, L'amitié, le jeu de rôle et l'expérimentation créative.[12]

3.6 Types de vie privée

3.6.1 La vie privée de la personne

Englobe le droit de garder les fonctions du corps et les caractéristiques du corps (telles que les codes génétiques et la biométrie). Cet aspect de la vie privée comprend également des intrusions non physiques dans le corps, telles que celles qui se produisent avec les scanners corporels de l'aéroport.

3.6.2 La vie privée des comportements et des actions

Comprend des questions délicates telles que les préférences sexuelles et les habitudes, les activités politiques et les pratiques religieuses.

La notion de vie privée du comportement personnel concerne les activités qui se déroulent dans l'espace public et l'espace privé.

3.6.3 La vie privée de communication

vise à éviter l'interception des communications, y compris l'interception du courrier, l'utilisation de bogues, les microphones directionnels, l'interception de la communication par téléphone ou sans fil ou l'enregistrement et l'accès aux messages électroniques.

3.6.4 La vie privée des données et de l'image

Comprend la protection des données d'une personne car elle est automatiquement disponible ou accessible à d'autres personnes et organisations et que les gens peuvent "exercer un degré important de contrôle à ce sujet les données et son utilisation.

3.6.5 La vie privée pensées et sentiments

Est le droit de ne pas partager ses pensées ou ses sentiments ou de révéler ces pensées ou ces sentiments. La confidentialité des pensées et des sentiments peut être distinguée de la 4e privée de la personne, de la même manière que l'esprit peut être distingué du corps

3.6.6 La vie privée De l'emplacement et de l'espace

Signifie que les individus ont le droit de se déplacer dans un espace public ou semi-public sans être identifiés, suivis ou surveillés. Cette conception de la vie privée comprend également un droit à la solitude et un droit à la vie privée dans les espaces tels que la maison, la voiture ou le bureau.

3.6.7 La vie privée D'association

S'intéresse au droit des gens à s'associer à ceux qu'ils souhaitent, sans être surveillé. .[13]

3.7 Défis et enjeux de la vie privée

- L'émergence du Cloud computing et le déploiement de mapredus dans les clouds publics fournissent une nouvelle série de défis dans la vie privée des données. Et le grand problème dans le Cloud est les fuites de données sur le fournisseur du Cloud,
- Protection des données contre les fournisseurs de Cloud adversaires :
- Les adversaires utilisateurs dans le Cloud.
- La difficulté de Cacher les données d'identité des utilisateurs.
- Protection des données des utilisateurs contradictoires :
- Assurer la confidentialité en présence d'un fournisseur de Cloud adversaire qui peut modifier ou supprimer des données.
- Multi utilisateurs sur un seul nuage public.
- Dans le Cloud public chaque utilisateur est capable d'accéder à toutes les données requises, mais aussi que les utilisateurs ne peuvent accéder à des parties de données pour lesquelles ils ne sont pas autorisés. Cela est généralement résolu par les mécanismes d'authentification et d'autorisation qui sont très complexes.[14]

3.8 Sécurité via vie privé

-La sécurité et la vie privée semblent être en désaccord en tout temps et sont très débattues La sécurité vise à réduire les risques. Pour réduire les risques, spécifiques

Des informations sur une ressource sont souvent requises. Dès que la ressource est liée à Entités telles que les particuliers et les entreprises.

-La vie privée (la vie privée sert a anonymiser les données) les problèmes de confidentialité s'inquiètent. Le problème est généralement abordé en anonymisation des données. .[15]

3.9 Cryptage de données

Le cryptage est la technique la plus utilisé pour protéger la confidentialité des données privées sensibles stockées.

Cette technique basé sur des méthodes et des algorithme spécialisés pour les données de petite et moyenne taille non par les données de grande taille, pour cela il est nécessaire de développer des nouvelles approches de cryptographie et des algorithmes efficaces pour les grandes données structurées , semi-structurées et non structurées.

Cela a conduit à des progrès et l'émergence de nouvelles méthodologies pour le chiffrement comme :

3.9 .1 Cryptage recherché :

- Cryptage symétrique de recherche

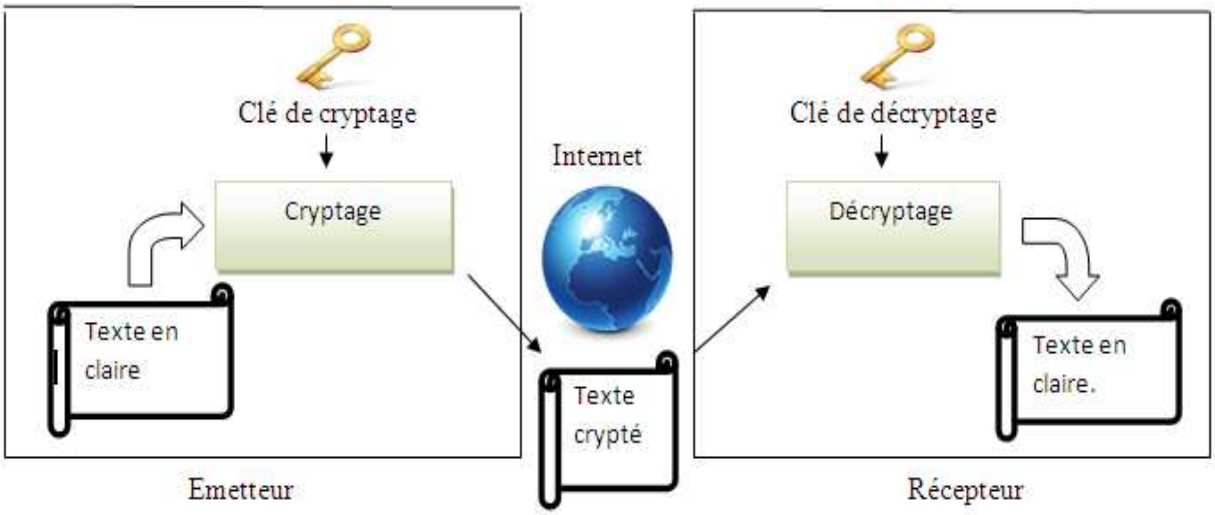

Fig. 1.6 Cryptage symétrique

- Chiffrement recherchable asymétrique / Cryptage recherché basé sur clé publique.

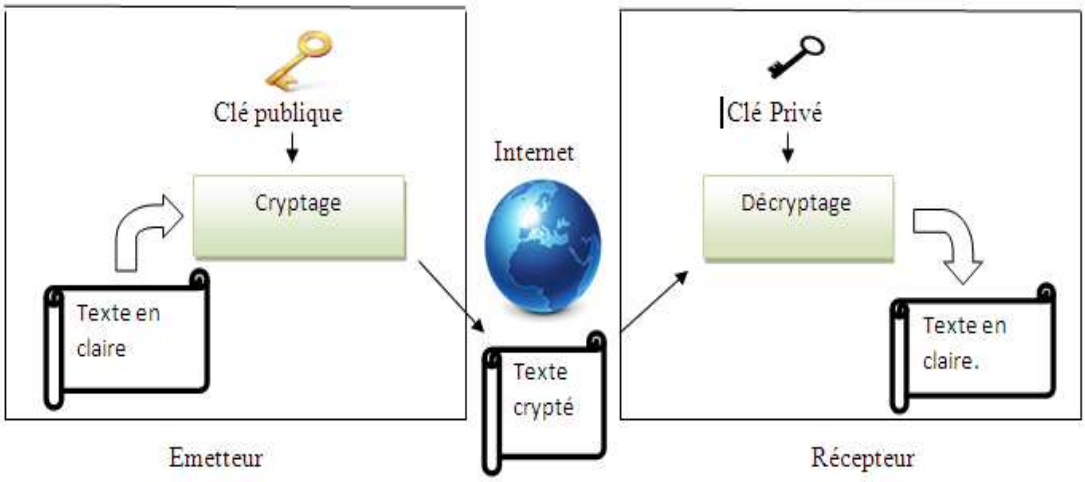

Fig. 1.7 Cryptage Asymétrique

3.9.2Le cryptage préservant l'ordre

Dans ce type de cryptage les messages chiffrés préservent l'ordre de messages clairs. Ce type de cryptage comprend des méthodes de cryptage qui génèrent du texte crypté comme une somme de données selon une distribution prédéterminée.

3.9.3 Cryptage structuré.

Le cryptage structuré peut être considéré comme une généralisation du cryptage symétrique à recherche indexée (SSE).

Un schéma de cryptage structuré crypte les données structurées de manière à pouvoir être interrogé par l'utilisation d'un jeton spécifique à une requête qui ne peut être généré qu'avec la connaissance de la clé secrète.

3.9.4 Cryptage homomorphique.

Le cryptage homomorphique peut être formellement défini comme suit:

- Génération de clés.
- Cryptage.
- Décryptage.
- Évaluation homomorphe. .[16]

3.10 Gestion de la confiance

La confiance a une relation solide avec la sécurité et la 4e privée, assurant la sécurité du système et la confidentialité de l'utilisateur, ce qui permet de gagner la confiance, la confiance ait été étudié dans plusieurs disciplines, y compris la sociologie, la psychologie, l'économie et l'informatique. Il existe deux conditions qui permettent la confiance: le risque et l'interdépendance, La source de risque est l'incertitude quant à l'intention de l'autre partie. L'interdépendance se caractérise par le fait que les intérêts des deux parties

Sont liés et ne peuvent être atteints sans compter l'un sur l'autre. La relation

N'est pas une relation de confiance si ces deux conditions n'existent pas.

La confiance est un concept important pour les utilisateurs du Cloud et aussi pour les fournisseurs de ser4ces de décider quel fournisseur d'infrastructure peut répondre à leurs besoins.

Dans un environnement de Cloud il existe deux types de confiance une confiance difficile et une confiance en douceur :

- La confiance difficile : Il considère les plates-formes de ser4ce fiables si l'existence de primitives de sécurité nécessaires est prouvable.

- La confiance en douceur : est un terme subjectif impliquant des aspects tels que les émotions humaines intrinsèques, les perceptions, les expériences d'interaction, y compris les commentaires et les commentaires des utilisateurs. .[17]

3.11 Définition et Types de vulnérabilité

Une vulnérabilité est une imperfection dans un système. le but des attaquants est la découverte de ces défauts pour faire des cyber-menaces .ces types est :

3.11.1 Vulnérabilités de logiciels

Les vulnérabilités de logiciels peuvent amener les informations stockées dans la base de données à être accessibles à l'attaquant, et l'attaquant peut accéder au système et lancer des cyber-attaques.

3.11.2 Vulnérabilités du personnel

Les vulnérabilités du personnel comprennent : les employés mécontents qui ont un avantage par rapport aux attaquants extérieur. Et les employés naïfs qui sont exploités par les attaquants

3.11.3 Vulnérabilités de planification de récupération après sinistre

Le plan de reprise après sinistre est un document précis les acti4tés qui doivent être sui4es pour récupérer une catastrophe. Ce plan peut contenir plusieurs vulnérabilités.

3.11.4 Vulnérabilités du protocole réseau

La vulnérabilité peut toucher aussi des protocoles de réseaux, Et sont exploités pour perturber ou confondre les sites .par exemple le protocole HTTP est un protocole vulnérable. Donc Il est important de prendre plusieurs mesures pour empêcher l'exploitation de ces protocoles. .[18]

3.12 Infrastructure critique et Bigdata

Plusieurs pays ont pris des mesures explicites pour protéger l'infrastructure critique car elle devenue importante dans nos jours. L'infrastructure critique englobe généralement les installations et les organisations qui fournissent des ser4ces (santé, transport, sureté ...Etc.) ou des produits (production agricole, industrielle...Etc.) au pays. .[19]

3.13 Contrôle de sécurité et protection d'infrastructure

Il existe trois catégories de contrôles de sécurité :

- ✓ La prévention des contrôles : empêchent les incidents de sécurité de se produire.
- ✓ La détection des contrôles : détectent des menaces de sécurité.
- ✓ La correction des contrôles : corrigent les menaces qui ont été détectés.[20]

Fig. 1.8 Les catégories de contrôles de sécurité

3.14 Comment sécuriser les Bigdata

Les défis de sécurité lorsque les organisations commencent à se déplacer des données sensibles à un référentiel Bigdata

3.14.1 Faiblesse de sécurité Hadoop

3.14.1.1 Les défis fondamentaux de sécurité et de la confidentialité

- La sécurité doit être surveillée en temps réel.
- La programmation distribuée doit avoir des calculs sécurisés.
- Avoir un contrôle qui permet de vérifier chaque détail d'accessibilité au système à sécuriser.
- Avoir un audit qui permet de vérifier chaque détail.

3.14.1.2 Faiblesses de sécurité supplémentaires

Les logiciels Open Source comme Apache Hadoop possèdent des faiblesses de sécurités, nous allons énumérer quelque uns dans cette section.

- Les fichiers de configuration sont sans contrôles de validité.
- Les problèmes liés au type d'attaques d'injection SQL se déplacent avec des composants Hadoop comme Hive et Impala.
- Les données sensibles ne possèdent pas un contrôle cryptographique natif.
- Lors de la communication entre un nœud de donnée à un nœud de donnée, les données de texte effacées peuvent être envoyées. [21]

4. Conclusion

Dans la pratique, il existe de nombreux défis pour le traitement et l'analyse des données importantes. Comme toutes les données sont actuellement visualisées par les ordinateurs, elles entraînent des difficultés dans l'extraction de données, suivie de sa perception et de sa

cognition. Ces tâches nécessitent beaucoup de temps et ne fournissent pas toujours de résultats corrects ou acceptables. Dans ce chapitre, nous avons obtenu une classification pertinente des méthodes de grande visualisation des données et nous avons suggéré la tendance moderne aux outils basés sur la visualisation pour le soutien aux entreprises et d'autres domaines importants. Les états passés et actuels de la visualisation des données ont été décrits et soutenus par l'analyse des avantages et des inconvénients. les inconvénients et les stratégies d'optimisation possibles sont discutés. Pour les problèmes de vsualisation abordés dans ce travail, il est essentiel de comprendre les problèmes liés à la perception humaine et à la connaissance limitée. Seulement après cela, le domaine de la conception peut fournir des moyens plus efficaces et utiles d'utiliser Bigdata.

Chapitre 3 :Sécurité des Bigdata

1. Introduction

Dans ce chapitre, nous avons étudié les approches et les travaux connexes qui proposent des solutions pour la possession dans les BigDatas, et possession hors les BigDatas, nous avons cité des inconvénients pour chaque travail, et finalement nous avons construit une table comparative dans laquelle nous avons comparé entre ces diverses solutions en terme de critère de possession.

3. Possession dans les BigDatas

3.1 Utilisation de composant d'audite

3.1.1 Modèle de système

Il existe une méthode de vérification de l'intégrité des données s'appelle l'audit de données, ce processus est effectué par une personne nommée auditeur.

Nous décrivons maintenant un schéma proposé dans le but de soutenir des blocs de données de taille variable, des audits autorisés de tierces parties et des mises à jour de données dynamiques à grain fin.

Le schéma est décrit en trois parties :

1. La configuration

Dans l'étape de la configuration :

- ✓ le client générera des documents de saisie via KeyGen et FileProc (deux algorithmes).
- ✓ puis télécharger les données vers CSS.
- ✓ il stockera une RMHT au lieu d'un MHT en tant que métadonnées.
- ✓ En outre, le client autorisera le TPA en partageant une valeur sigAUTH.

2. **Mise à jour de données vérifiable**

   Dans l'étape de la mise a jour :

   - ✓ le CSS effectue les requêtes de mise à jour fine du client via PerformUpdate.
   - ✓ puis le client exécute VerifyUpdate pour vérifier si CSS a effectué les mises à jour sur les blocs de données et leurs authentificateurs correspondants (utilisés pour l'audit) avec honnêteté.

3. **Défi, Génération et vérification de la preuve**

   Décrit comment l'intégrité des données stockées sur CSS est vérifiée par TPA via GenChallenge, GenProof et Verify.

3.1.2 Les inconvénients de la méthode

- ❖ Fournir une sécurité et une flexibilité améliorées.
- ❖ Le grand nombre de petites mises à jour telles que les applications fréquentes dans les médias sociaux et les transactions commerciales.
- ❖ Améliorer d'autres méthodes de protection côté serveur.
- ❖ La difficulté de la planification des données à l'auditabilité dans le Cloud computing.[23]

3.2 Sécurisation de la possession

On va présenter une méthode pour l'essai de résoudre le problème de « l'incapabilité de faire face de manière adéquate aux problèmes posés par le changement dynamique et la corruption de données » dans les solutions proposées précédemment.

3.2.1 SPDP (Secure Provable Data Possession)

Un schéma SPDP est un type de système de collecte y compris deux algorithmes et un protocole interactif :

KeyGen (1k) : prend un paramètre de sécurité k comme entrée et renvoie une paire de clés publique et secrète correspondantes (pk, sk).

TagObject (pk, sk, F*)* : Il prend en entrée une clé publique pk, une clé secrète sk et un fichier F, et renvoie la métadonnée de vérification Tb, il est exécuté par le client dans le but de la génération des métadonnées de vérification.

Proof (pk, C, P(H0|m)) : Il prend en entrée une clé publique pk et motivé par le défi C et la probabilité postérieure P (H0 | m) d'accepter / rejeter la modification ou l'addition et Il renvoie une preuve de possession V qui est déterminée par le défi C et la probabilité a posteriori. Il est exécuté par le serveur pour la génération d'une preuve de possession.

Balise de variable d'objet est utilisé comme une métadonnée pour la vérification de l'objet.

Pour un client peut vérifier si le serveur possède l'objet correcte ou non, il faut que le serveur utilise une balise de variable d'objet et produire une preuve.

3.2.2Les inconvénients de la méthode

- Cette méthode fournit une analyse de sécurité moins formelle.(le client est très sollicité pour la réalisation des tests de possession alors que ces clients est mal intentionné)
- Le SPDP peut garantir la sécurité du stockage des objets, mais il doit encore être amélioré en ce qui concerne la protection des données d'origine dans le cadre de la stratégie d'optimisation et de protection des annuaires approuvée. [24]

3.3 Traitement du problème de certificat

On va présenter un système (un schéma) CLPDP (PP-CLPDP) qui assure la protection de la vie privé et également résoudre le problème de gestion des certificats.

3.3.1 Le modèle de système

Il faut qu'un schéma CLPDP soit composé de quatre essentiel éléments :

- KGC (Key génération center) : (partie de confiance) son rôle est de générer toutes les clés privées de tous les utilisateurs, c'est aussi le responsable des paramètres du système.
- Le propriétaire des données : il est un utilisateur dans le serveur, il met ses données dans le serveur et il peut vérifier l'intégrité de ses données avec le vérificateur.
- Le serveur Cloud : (parti semi-confiance) il stocke et gère les données, il peut également produire des preuves pour la vérification de l'intégrité des données.
- Le vérificateur : (parti semi-confiance) il peut faire la vérification de l'intégrité des données des utilisateurs.

3.3.2 Le modèle de système CLPDP proposé

Le schéma CLPDP proposé comprend cinq algorithmes :

La 1 ère étape

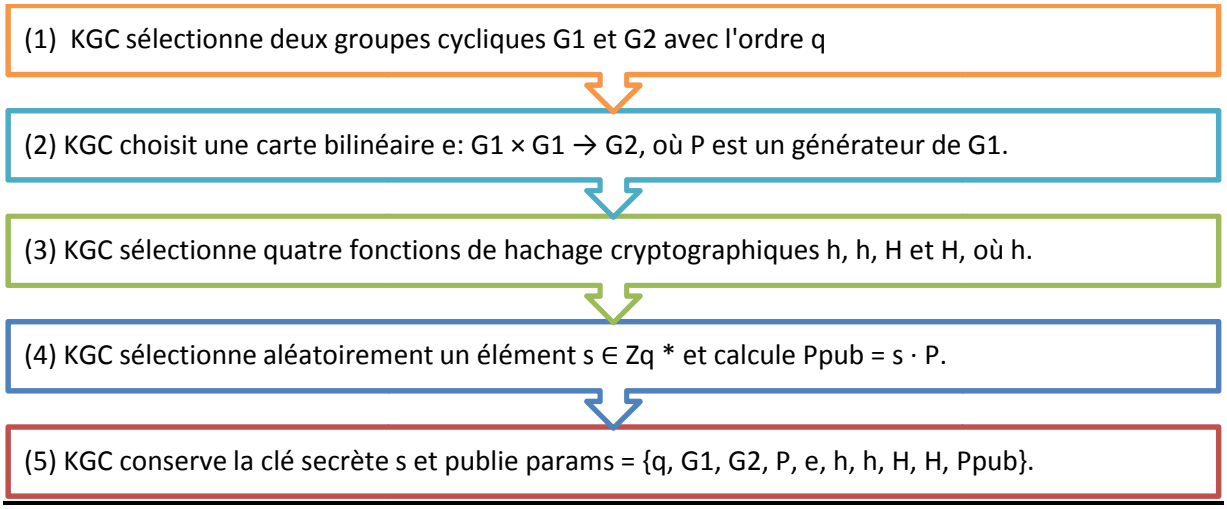

Fig. 2.1 La première étape de modèle CLPDP

La 2éme étape

Ici on se base sur l'utilisation des algorithmes :

Fig. 2.2 La deuxième étape de modèle CLPDP

3.3.3 Les inconvénients de la méthode

- La difficulté de la mise en d'œuvre d'un prototype de système CLPDP dans le Cloud réel. [25]

4. Possession dans le Cloud

4.1 Approche probabiliste

On va présenter deux méthodes qui compensent les fonctions cryptographiques consommatrices de ressources avec des fonctions de hachage simples.

La première méthode basée sur le stockage de client de ses données sur un serveur dans le Cloud, la deuxième méthode basée sur la duplication des données de clients entre plusieurs serveurs distants. Notre solution repose dans la technique de demande périodique.

1) Le client demande au serveur.

2) Et le serveur ne peut répondre correctement que s'il conserve les données du client.

Dans la première méthode :

Le client doit avoir accès aux paires: Eng challenge-respond.

Donc, il doit le stocker soit dans sa propre mémoire, soit sur le serveur externe.

Dans la deuxième méthode :

Le client stocke les données sur deux serveurs et peut comparer les repenses.

Les paramètres utilisés

w	le coût payé par le client pour stocker ses données sur le serveur dans un délai.
v	le coût payé par le client pour une demande.
k	l'amende payée par le serveur au client en cas de perte de ses données
u	le coût engagé par le serveur pour posséder les données du client dans une période donnée.
T	le nombre de périodes où les données ont été stockées dans le nuage.
H	la fonction hash définie par la signature: H: {0, 1} * - → {0, 1} d où d est une longueur de la valeur de sortie.
m	le nombre de fichiers du client stockés sur les serveurs distants.

Table 2. Les paramètres utilisés

4.1.1 Modèle PDP à serveur unique

Dans cette méthode les données de client sont stockées sur un seul serveur, et le protocole fonctionne comme suit :

- Le client prépare un ensemble Zf composé de n paires challenge-response.
  - Exemple : $ZF = \{(ci, ri), i = 1, 2, ..., n\}$.
- Chalenge-response sont générés à l'aide de la fonction H de hachage Donc :
  - $ci = H(F, Ku)$ et $ri = H(F, ci)$.
  - ou Ku : est une clé secrète
  - F : est le fichier vérifié.
- La création de ZF s'effectue par les procédures: PrepareChallenge et PrepareResponse.

- Nous supposons que ci et ri sont beaucoup plus courts que le fichier F.
- L'ensemble ZF est conservé sur l'ordinateur du client.
- Ensuite, le client estime la probabilité p et corrige la distribution S comme suit:

$$S = \begin{cases} 1 \text{ avec probabilité p.} \\ 0 \text{ avec probabilité 1-p.} \end{cases}$$

- À la fin de chaque période de temps j, le client prépare un défi c et l'envoie au nuage en invoquant GenerateChallenge. Selon la distribution S, il peut s'agir d'un défi aléatoire ou d'un défi de l'ensemble ZF. Dans le premier cas, le client ne peut vérifier s'il est trompé par le nuage. Dans le second cas, le client reçoit une réponse r' du nuage . Si r=r' Cela signifie que la vérification est positive et que le client n'est pas trompé.

Algorithm 1 : Generate Challenge(input: *b*, *F*)
if b=0 **then{** Returns a pair: (c, ∅); **}** **if** b=1 **then{** ZF ← ZF \ {(c, r)}; Returns a pair: (c, r); }

Algotithm2: Single PDP Protocol
Ku ← *Setup*(*ξ*); *c* ← *P repareChallenge*(*Ku*); *r* ← *P repareResonse*(*c*); At the end of each time period *j* the following loop is executed **Loop{** Draw a number *b* from the set *{0, 1}* for a file *F* according to the distribution *S*;

(*c, r*) ← *Generate Challenge*(*b, F*);
Send the generated challenge *c* to the cloud *C*;
r ← *CloudResponse*(*c, F, C*);
if *b=1* **then{**
Verify(*r*, *r*);
}
}

4.1.2 Modèle PDP multiserveur

Dans cette méthode le client peut stocker ses données dans plusieurs serveurs indépendants, pour notre situation on suppose que le client stocke ses données dans deux Cloud C1 et C2

Algorithme 4. Multi PDP protocole

Ku ← Setup(ξ);
c ← P repareChallenge(Ku);
r ← P repareResonse(c);
// Les données du client sont réparties en fonction des paramètres: p et m
DataDistribution (p, m);
//A la fin de chaque période j, la boucle suivante est exécutée
{
//Envoyer le défi c au nuage C1 à propos du fichier choisi au hasard F ∈ A;
r'← CloudResponse(c, F, C1);
if F ∈ A ∩ B **then {**
Send a challenge c to C2;
r" ← CloudResponse(c, F, C2);
}

if r' = r" **then {**

Verify();
}
if F / ∈ A ∩ B **then {**
// Envoyer un défi c? au nuage C2 sur le fichier choisi au hasard
F'∈ B \ A ∩ B;

```
}

}
```

4.1.3 Les inconvénients de la méthode

- ❖ Le coût de possession augmente dans la méthode multiserveur car la charge de stockage des données dans deux nuages sera doublée.
- ❖ La difficulté d'exécuter les calculs complexes de cette méthode.
- ❖ L'augmentation de l'utilisation de la mémoire. [26]

4.2 Approche base arbre de dérivation

Le stockage en nuage offre aux clients un ser4ce de stockage de données flexible, dynamique et rentable. Ce nouveau paradigme de ser4ce de stockage de données introduit toutefois de nouveaux défis de sécurité, comme les clients doivent être convaincus que leurs données sont correctement stockées dans le Cloud. Donc Il est impératif de fournir un protocole d'audit dynamique efficace et sécurisé pour vérifier l'intégrité des données dans le nuage.

- Nous analysons d'abord la performance dynamique de certains travaux antérieurs et proposons un nouveau schéma DPDP (Dynamic Provable Data Possession).
- Nous introduisons un schéma de signature sécurisé et la structure de données LBT (Large Branching Tree) dans notre schéma.
- La structure LBT simplifie le processus de mise à jour et le schéma de signature est utilisé pour authentifier à la fois la valeur et la position des blocs de données, ce qui améliore grandement l'efficacité de la communication. L'analyse de sécurité et de performance montre que notre système DPDP est sûrement sûr et efficace.

4.2.1 Grand arbre de ramification (LBT)

LBT a une structure d'un arbre concis. Chaque nœud de l'arbre sauf les feuilles a plus de 2 nœuds enfants.

Un schéma d'authentification LBT produit des signatures qui représentent des chemins reliant des blocs de données à la racine de l'arbre. Le mécanisme d'authentification (validation) fonctionne de manière inductive .La racine authentifie ses nœuds enfants, ces nœuds

authentifient leurs nœuds enfants, et l'authentification procède récursivement aux blocs de données authentifiés par son parent. Et la façon dont les nœuds frères et sœurs sont authentifiés est différente. Comme chaque nœud a plusieurs nœuds frères, nous les étiquetons avec un nombre pour indiquer sa position parmi les frères et sœurs. Et une valeur d'authentification unique qui peut être vérifiée indépendamment a été générée pour la vérification.

4.2.2 Système PDP dynamique

Le système PDP dynamique pour les données externalisées dans le Cloud se compose de trois entités:

1. Client, qui dispose d'une capacité de stockage et d'une capacité de calcul limitées, mais d'une grande quantité de données à stocker dans le Cloud.
2. Cloud Storage Server (CSS), une entité qui dispose d'énormes capacités de stockage et qui est en mesure de fournir la maintenance et le calcul des données.
3. TPA (Third Party Auditor), qui se spécialise dans la vérification de l'intégrité des données externalisées dans le Cloud lorsqu'elles reçoivent une demande du client.

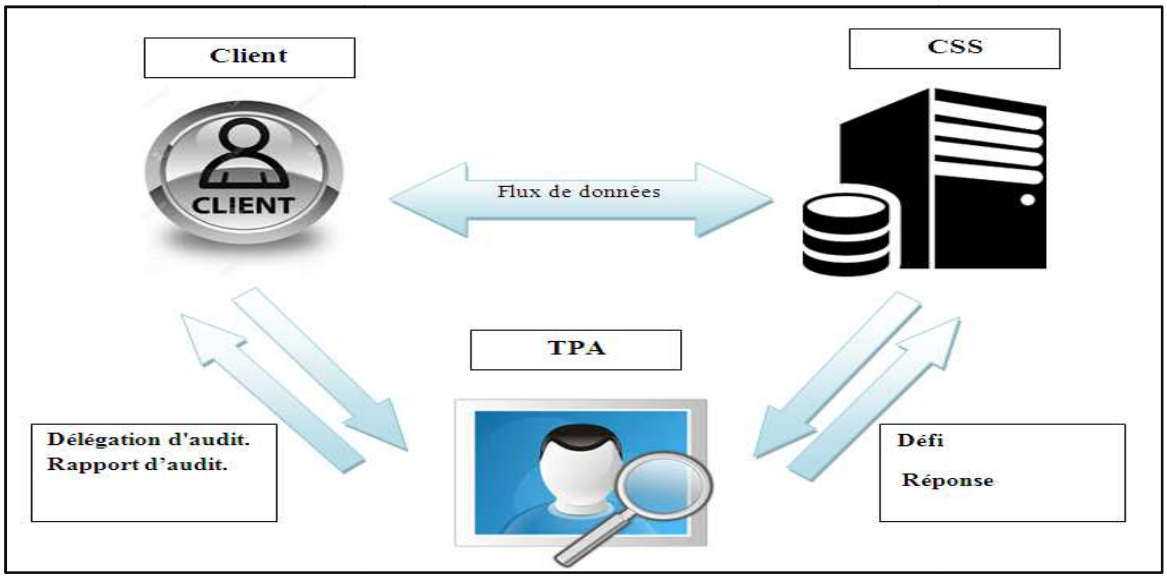

Fig. 2.3 Système PDP dynamique

La structure de données LBT est une extension de MHT, qui est destinée à prouver qu'un ensemble d'éléments sont intacts et non altérés. Naturellement, nous considérons l'utilisation de l'algorithme de hachage utilisé dans la structure MHT pour authentifier les valeurs des nœuds dans LBT, mais cet algorithme apporte des effets indésirables sur la performance.

Pendant le processus de mise à jour, le client modifie, insère ou supprime les données si seulement pour un bloc affecte l'ensemble de la structure de données, ce qui entraîne des frais de calcul O (n) pour le client et CSS. Par conséquent, il est impératif de trouver une meilleure méthode pour authentifier la structure de données LBT. Au lieu d'utiliser des fonctions de hachage, nous utilisons le schéma de signature.

pour améliorer l'efficacité de la vérification des éléments dans le LBT. La complexité de calcul diminue à O (1) dans le processus de mise à jour. En ce qui concerne l'auditabilité publique, nous utilisons les balises homomorphes vérifiables. La raison en est que les HVT permettent de vérifier l'intégrité des données sans blocage. La procédure de notre schéma est résumée en trois phases:

- La configuration.
- L'opération dynamique.
- L'audit périodique.

4.2.3 Les inconvénients de la méthode

- Les coûts de communication élevé.
- Les coûts de calcul pour chaque mise à jour dynamique est très élevé.
- l'analyse de sécurité et à l'analyse de performance n'est pas évalué avec assez de critères. [27]

4.3 Approche basé hiérarchisation de données

Cette méthode présente un schéma de possession de données portable avec une structure hiérarchique. En plus cette approche permet au propriétaire de données d'inscrire et retirer des clients.

4.3.1 Les étapes de la méthode

1. Construit une hiérarchie de données.
2. génère les clés privées des clients et génère des clés publiques avec KeyGen.
3. envoie les clés privées aux clients et publie les paramètres publics sur le Cloud public.
4. le propriétaire de données construit d'abord une hiérarchie de clés de rôle dans RBAC
5. Le propriétaire de donnée alloue différentes données dans des rôles différents en fonction de leurs différentes valeurs.

6. différentes données sont liées à des rôles différents.
7. Les clients, qui sont considérés comme des utilisateurs différents, sont classés dans la hiérarchie des données en fonction de leur paiement pour les données ou les contrats.
8. le propriétaire des données calcule les paires d'étiquettes de blocs en utilisant l'algorithme TagGen.
9. CSS génère un PDP (proof) en mettant en œuvre l'algorithme GenProof en réponse à la requête du client
10. le client exploite CheckProof pour vérifier le PDP.

    Quand on termine ces étapes les données ne peuvent pas être utilisées par les clients si les clients ne peuvent pas valider leur PDP.

4.3.2 Les inconvénients de la méthode

- La procédure d'installation consomme plus de temps car elle implique la construction d'une hiérarchie de données. [28]

4.4 Approche basé cryptographie

4.4.1 PPDP (proxy provable data possession)

Notre system se compose de quatre entités :

- **Le client**

  Le client est la partie qui a un grand volume de données, et il veut les stocker dans le PCS.
- **PCS (Public Cloud Ser4ce)**

  Il doit être caractérisé par un grand espace de stockage et des ressources de calcul pour la maintenance des données client.
- **Le concessionnaire**

  Est celui qui stocke les données de client dans le PCS
- **Le mandataire (proxy)**

  il est le responsable de la vérification de la possession des données des clients.

  La structure PPDP utilisée est comme suivant :

  Soit g un générateur de G1.

  Un fichier codé en utilisant un code de correction d'erreur.

  Le fichier soit de4sé en n blocs (m1, m2, m3,…, mn) ou $m \in Z*$.

On utilise les fonctions suivantes

$f : Zq* \times \{1, 2,\ldots n\} \rightarrow Zq*$
$\Omega : Zq* \times \{1, 2,\ldots.. n\} \rightarrow Zq*$
$\pi : Zq* \times \{1, 2, \ldots.n\} \rightarrow \{1, 2,\ldots. \cdot, n\}$
$H : \{0, 1\} * \rightarrow Zq*$
$h : Zq* \times Zq* \rightarrow G1$

Où f et Ω sont deux fonctions pseudo-aléatoires, et π est une permutation pseudo-aléatoire, H et h sont des fonctions de hachage cryptographiques. Pour la structure d'accès générale, la construction du protocole PPDP se compose de six phases: SetUp, TagGen, CertVry, CheckTag, GenProof, CheckProof. [29]

4.5 Approche basé protocole distant

Dans le Cloud computing le client confronte un gros problème, ce problème est l'assurance de l'intégrité de ses données distantes. Pour la résolution de ce problème nous proposons un protocole amélioré (amélioré de protocole de Hao et al) de vérification de l'intégrité des données distantes. Ce protocole est sûr et peut résister a l'adversaire actif, de plus il peut prendre en charge des blocs de taille variable.

4.5.1 Le protocole amélioré de Hao et al

Dans cette amélioration on prend en charge les blocs de taille variable, Le fichier m est segmenté en m = {mij}, i ∈ [1, n], j ∈ [1, li], li ∈ [1, lmax]

Le fichier m comprend n bloc de données, chaque bloc possède des segments l , tous les segments l d'un bloc ont la même taille, tous les segments d'un bloc sont inferieur ou égale à lmax.

SetUp (1k → (pk, sk)) : soit p et q deux nombres premiers

Et N =p*q, N est un module publique de RSA

QRN est tous les résidus quadratiques Modulo N

QRN : Le groupe cyclique multiplicatif.

H : une fonction de hachage cryptographique

{g1, g2…..glmax}: des générateurs de QRN.

La clé publique est pk = (N, g1, g2, ..., glmax)

La clé secrète est sk = (p, q).

TagGen (pk, sk, m) → Dm. : mi, i ∈ [1, n]

$$Di = (g1^{mi1} \cdot g2^{mi2} \ldots glmi\ ili) \bmod N \quad [30]$$

4.6 Approche multi fonction

4.6.1 Modèle de système

Multifonction prouvable data possession (MF-PDP)

Le schéma de MF-PDP se base sur cinq algorithmes

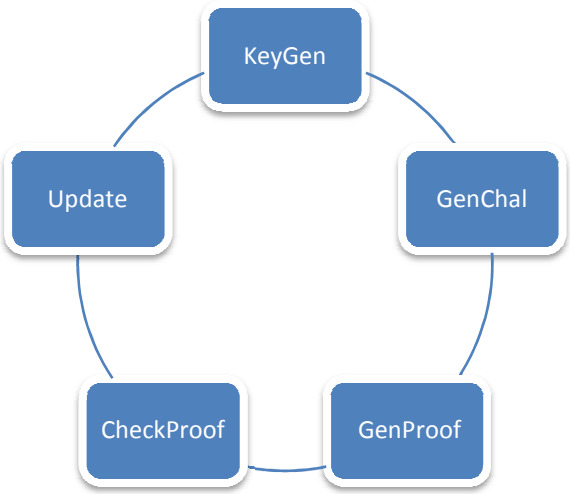

Fig. 2.4 cinq algorithmes de MF-PDP

Le MF-PDP est une version améliorée de PDP, mais il prend en charge de nouvelles exigences dans environnement de nuage en utilisant de nouvelles idées. En fonction exigence: il prend en charge la dynamique des données en supprimant l'index de bloc de données lorsque calculer l'étiquette de bloc de données et la vérification publique en utilisant une clé publique dans l'étape de vérification. En exigence de performance, il utilise l'étiquette homomorphe vérifiable pour réduire le coût de communication et la vérification d'échantillonnage aléatoire pour réduire les coûts de calcul.

Dans l'exigence de sécurité, la procédure de vérification dépend de la preuve de deux parties l'information qu'une partie vient du prouveur et l'autre vient du vérificateur, cette idée non seulement assurez-vous de la validité de la preuve mais aussi de résister à l'attaque de remplacement de la preuve.

4.6.2 Les inconvénients de la méthode

La difficulté d'étudier la relation entre les différentes fonctions

La réponse aux différentes exigences de l'en4ronnement informatique en nuage.

Le stockage dans des différents endroits. (Distance géographique coute du temps de transfert de donnée et de cout). [31]

4.7 Approche de préservation de 4e privée

Cette méthode appelé vérification de possession de données à distance avec des authentifiants préservant la confidentialité pour le stockage en nuage.

Elle prend en compte la confidentialité des authentifiants , Dans cette méthode :

- Le fournisseur de ser4ces de Cloud et le vérificateur public n'ont pas accès aux authentiques authentifiant (signatures) pour les données de Cloud.
- Cette méthode utilise un nouvel authentificateur appelé HMI (Homomorphic In4sible Authenticator).

4.7.1 Modèle de système

- **L'utilisateur Cloud :** L'utilisateur du Cloud possède une grande quantité de données qui seront externalisées dans le Cloud.
- **Le Cloud :** fournit un espace de stockage énorme pour l'utilisateur du Cloud.
- **L'auditeur tiers (TPA) :** il a des capacités de calcul et de communication, et il est délégué par l'utilisateur du Cloud pour vérifier la possession des données du nuage.
- **L'autorité de confiance (TA**) : L'AT est une organisation reconnue par le Cloud et par l'utilisateur du Cloud, qui peut être une institution publique ou une organisation non gouvernementale. La clé publique de l'AT est utilisée par l'utilisateur du Cloud

pour générer des authentifiants préservant la confidentialité de ses données externalisées.

4.7.2 Les inconvénients de la méthode

La fonction de préservation de la confidentialité de l'authentificateur entraîne des frais généraux négligeables pour les entités impliquées. [32]

5. Possession dans un en4ronnement non approuvé

5.1 Approche basé balises homomorphes

5.1.1Provable Data Possession (PDP)

Les schémas PDP ont plusieurs utilisations comme :

- Ils échantillonnent le stockage du serveur.
- Ils accédant à un sous-ensemble aléatoire de blocs.
- On peut appliquer le schéma PDP dans le contexte de résistance.
- Le problème est d'empêcher les changements non autorisés par la technologie ou les techniques connues sous le nom de balises homomorphes.
- En raison de la propriété homomorphique, les balises sont calculées pour plusieurs blocs de fichiers pouvant être combinés en une seule valeur. A un moment plus long, le client peut vérifier qu'il possède le fichier en générant une attaque aléatoire contre un ensemble de blocs sélectionné aléatoirement.
- Le PDP efficace est un schéma fondamental qui subit une introspection archi4stique qui développe une partie de conservation à long terme des données d'astronomie.
- Les schémas PDP efficaces s'assureront que les exigences de la vérification de données à distance n'induisent pas indûment les sites de stockage à distance.
- Les performances de PDP efficace et d'autres protocoles de contrôle de données à distance seront implémentées. Les expériences montrent que la possession probabiliste garantit la possession de grands ensembles de données. Bien que le PDP est difficile à crypter et décrypter. La version mise à jour de PDP est dérivée comme CDPD.

5.1.2 Modèle de menace

- l'échec de le faire représente une perte de données.
- Le serveur n'est pas approuvé.

- La mauvaise conduite du stockage peut être abandonnée.
- les données qui ne peuvent pas ou rarement être consultées.

5.1.3 Exigences et paramètre

- Complexité du calcul.
- Bloquer la complexité de l'accès.
- Complexité de la communication.

La quantité de calcul et l'accès au bloc sur le serveur doivent être maintenus. L'efficacité de la bande passante doit être minimisée. La complexité chez le client est de moindre importance.

5.1.4 Analyse de sécurité

Nous avons utilisé La fonction de hachage homomorphe pour l'authentification car La PDP n'est pas capable pour la technique d'authentification de source.

Le protocole homomorphique utilisé pour composer le bloc on sous ensemble (plusieurs sous blocs).

Et la définition de sécurité est comme suit :

- Un simulateur S met en place un système PDP et choisit ses paramètres de sécurité.
- L'adversaire sélectionne les valeurs et les envoie au simulateur.
- L'adversaire peut interroger l'oracle aléatoire au n'importe quel moment. Pour chaque entrée dans le oracle aléatoire le simulateur répond avec un valeur aléatoire et stocke l'entrée et sortie correspondante dans le tableau.
- Lors de la phase de challenge, le simulateur défis A sur la valeur et envoie un valeur aléatoire .A réponses avec une chaîne.

5.1.5 Les inconvénients de la méthode

- Le calcul sur le serveur et la communication avec le serveur du client. Ils entraînent une charge faible au niveau du serveur et nécessitent une quantité faible et constante de communication..
- l'utilisation de la fonction de hachage sur des fichiers volumineux est couteuse. [33]

6. Modèle de comparaison (tableau comparatif)

Approche critère	Utilisation de composant d'audite	Sécurisation de la possession	Traitement du problème de certificat	Approche probabiliste	Approche basé arbre de dérivation	Approche basé hiérarchisation de données	Approche basé cryptographie	Approche basé protocole distant	Approche multi fonction	Approche basé balises homomorphes	Approche de préservation de 4e privée
Possession de données	√	√	×	√	–	√	√	×		√	√
Soutenir l'échantillonnage	√	–	√	–	×	×	–	×	√	√	×
Type de garantie	**Det**	**Det**	**Det**	–	**Det**	×	–	**Det**	-	**Pro**	**Det**
Type de preuve	**Det**	**Pro**	–	**Pro**	–	**Det**	–	–			**Det**
PDP (single…)	**No PDP**	**single**	**single**	**Sngle/ multi**	**single**	**single**	**single**	**No PDP**	**Single**	**Single**	**No PDP**
Vérification publique [34]	√	√	√	√	√	√	√	√		√	√
Dynamique des données [34]	**Yes**	**Yes**	–	–	**Yes**	**Yes**	–	–		–	**Yes**
Confidentialité de la vérification[34]	**Yes**	–	**Yes**	**Yes**	**Yes**	**Yes**	–	–		–	**Yes**
Anti-attaque de remplacement [35]	**No**	–	**Yes**	–	–	–	–	–	√	–	**Yes**
Anti replay Attack[35]	**No**	**No**	**No**	–	–	–	–	–	√		–
Opération de bloc [35]	**Yes**	**No**	**No**	**No**	**Yes**	**Yes**	–	**Yes**		**Yes**	**No**
Volume de données[36]	–	–	**x**	√	√	–	√	√		√	√

Variété[36]	–	–	x	X		√	x	x	x	x	√
Véracité[36]	–	–	x	x	√		x	√	x	√	√
Valeur[36]	–	–	x	X		√	x	x	x	√	–
Velocité[36]	–	–	x	x	√	×	√				×

Table 3. Tableaux Comparatif

Conclusion

Cloud computing fournit une infrastructure flexible et de grande capacité de stockage pour les applications Bigdata. Mais, comme toute nouvelle technologie elle a besoin de nombreuses améliorations et de la mise en place de normes précises pour éviter les risques. La sécurité est souvent considérée comme le frein principal à l'adoption des services du Cloud computing. C'est ainsi que de nombreux travaux ont été consacrés à la recherche de solutions pour remédier à ce problème.

Les chercheurs ont présenté une analyse sur les techniques de vérification de possession des données stockées dans un environnement externe.

Chapitre 3 :
Approches et travaux connexes

3 Introduction

L'objectif de notre projet de fin d'étude était de concevoir et implémenter une architecture de protection de la possession des données dans les Bigdata. Nous avons étudié des travaux de recherche Pour tirer parti de leur savoir-faire et présenter des solutions susceptibles de combler les défaillances observées. Dans ce chapitre on va présenter la conception détaillée, dans laquelle nous avons fixé la structure globale de l'application.

4 Conception général du système proposé

4.2 Architecture globale

Dans cette section on va présenter l'architecture globale du système proposé.

On a proposé une architecture qui nous permet :

- ✓ De faire le bon choix du Cloud approprié.
- ✓ De vérifier périodiquement les données stockées afin d'être sûr que ces dernières ne soient ni modifiées ni altérées.
- ✓ De contrôler l'accès des utilisateurs aux données stockées.
- ✓ Au utilisateur de vérifier la possession de ses données dans le Cloud.

On va proposer un ensemble de composants qui représente le système qui assure la protection la possession des données de clients dans les BigData et le rôle de chaque composant :

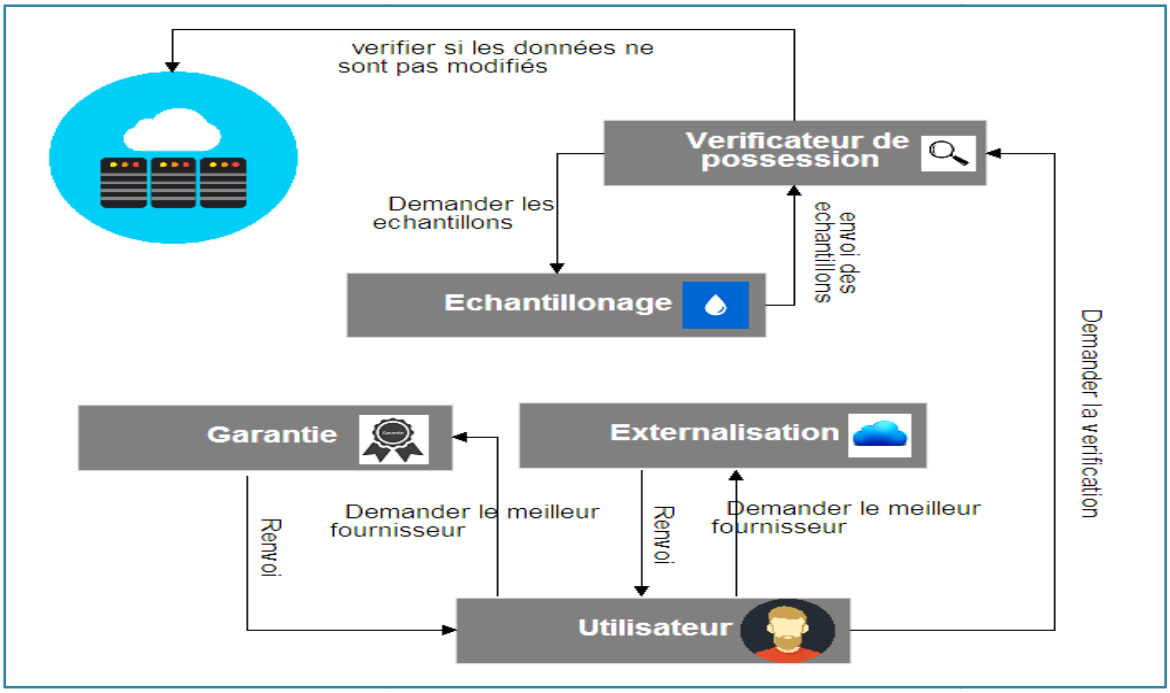

Fig. 3.1 Architecture de système proposé

2.2 Architecture détaillé

Pour qu'un utilisateur puisse protéger la possession de ses données dans le Cloud, nous avons proposé une architecture globale qui nous permet de:

- ✓ De faire le bon choix du Cloud (fournisseur) approprié.
- ✓ D'assurer la transmission correcte des données vers le Cloud.
- ✓ De vérifier périodiquement les données stockées afin d'être sûr que ces dernières ne soient ni modifiées ni altérées.
- ✓ De contrôler l'accès des utilisateurs aux données stockées.
- ✓ d'obtenir des échantillons des données.
- ✓ De vérifier la possession des données.

On va proposer un ensemble de composants qui représente le système qui assure la protection d'intégrité des Big Data et le rôle de chaque composant :

2.2.1 Composant d'échantillonnage

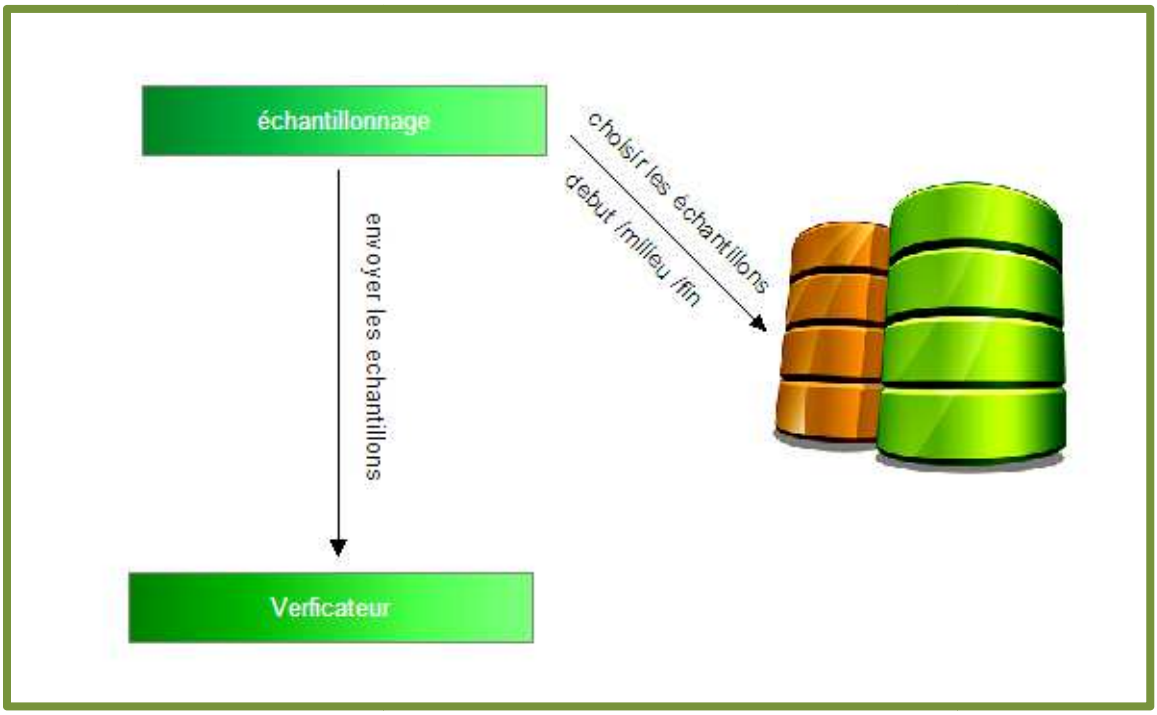

Fig. 3.2 Architecture de composant D'échantillonnage

➢ Le rôle de composant d'échantillonnage

- ✓ L'échantillonnage consiste à s'intéresser à des petits parties des données afin d'identifier des informations significatives qui concernent globalement l'ensemble de données.
- ✓ Le composant d'échantillonnage choisit aléatoirement un ensemble des 'échantillons des données qu'il vérifiera par la suite.
- ✓ Après un nombre donné de vérification, le composant d'échantillonnage échantillonne de nouveaux blocs pour les soumettre à une vérification afin de diminuer le risque d'erreur.
- ✓ Le volume des échantillons prélevés peut atteindre un pourcentage déterminé de la masse totale des données.
- ✓ Si les échantillons analysés sont fiables, on considère que toutes les données sont fiables et l'analyse sera abandonnée.

2.2.3 Composant auditeur

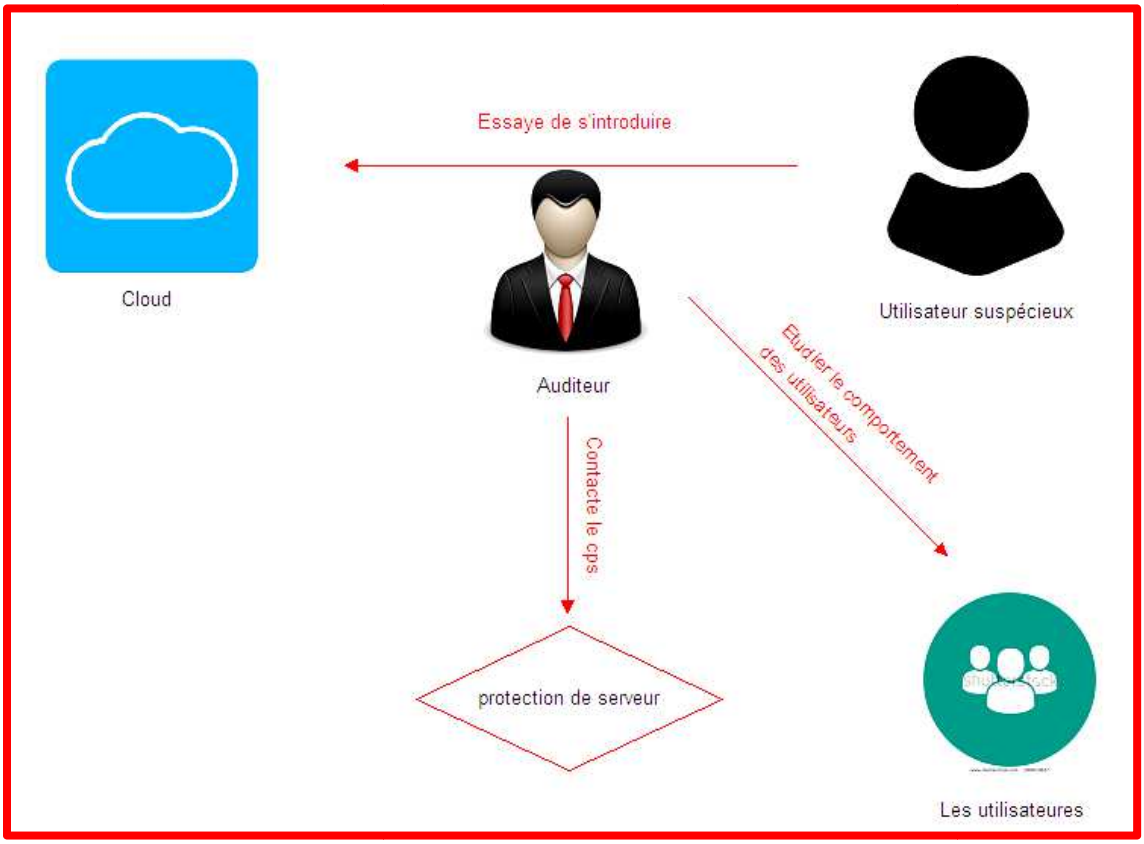

Fig. 3.3 Architecture de composant Auditeur

Le rôle de composant d'audit

- ✓ Analyse le comportement de l'utilisateur
- ✓ Comparaison des analyses avec les profils normaux
- ✓ Accepte ou refuse l'accès de l'utilisateur .

2.2.5 Composant fourniture de garantie et proof

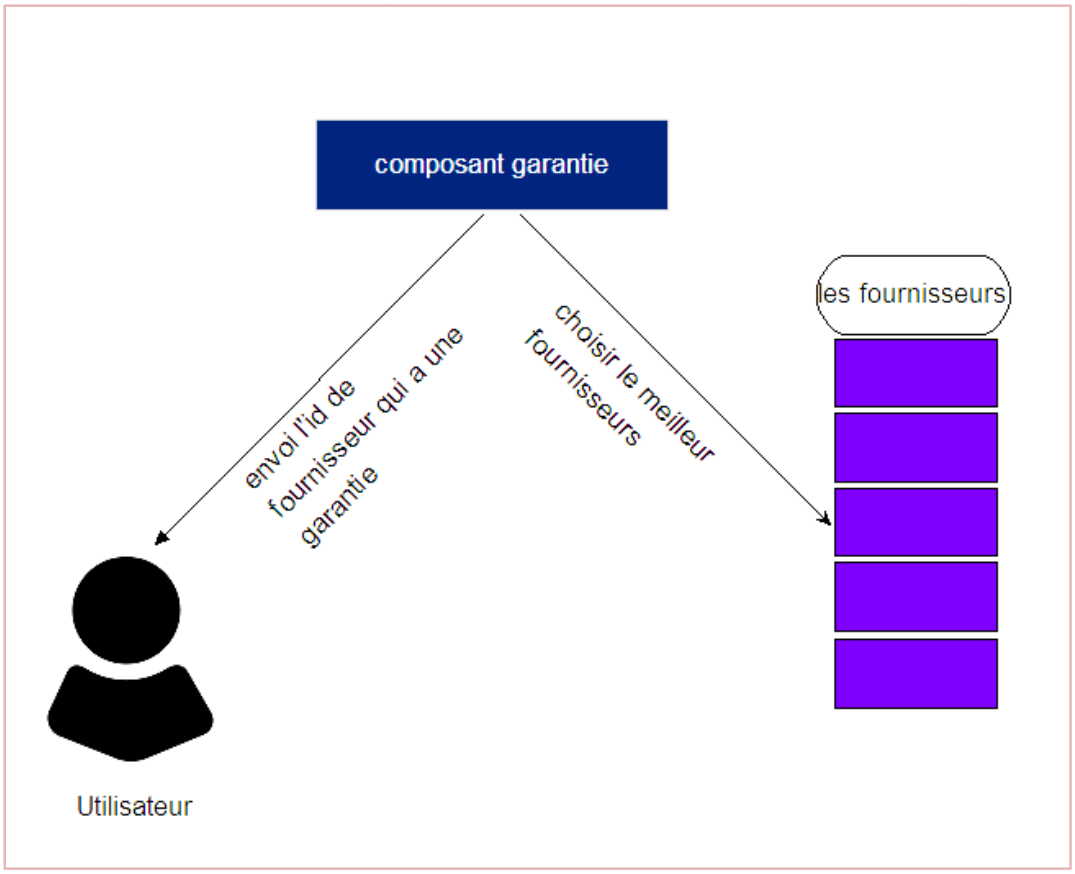

Fig. 3.4 Architecture de composant De Garantie

- ➢ **Le rôle de composant fourniture de garantie et proof**
  - ✓ Vérifier le fournisseur Cloud (fiable ou non).
  - ✓ Classer les fournisseurs selon la sécurité (la garantie).

2.2.6 Composant vérificateur de possession

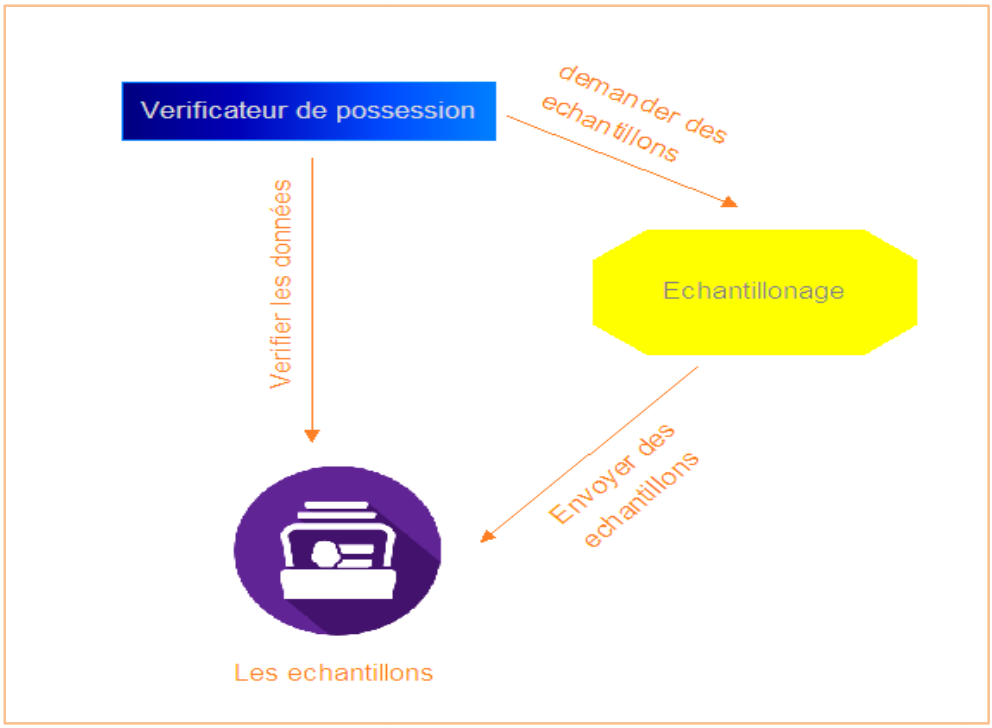

Fig. 3.5 Architecture de composant Vérificateur de possession

- **Le rôle de composant vérificateur de possession**
  - ✓ Réduire le rôle du client
  - ✓ Demande des échantillons des données
  - ✓ Vérifier la possession des données avec l'utilisation d'hachage.

2.2.7 Composant d'externalisation

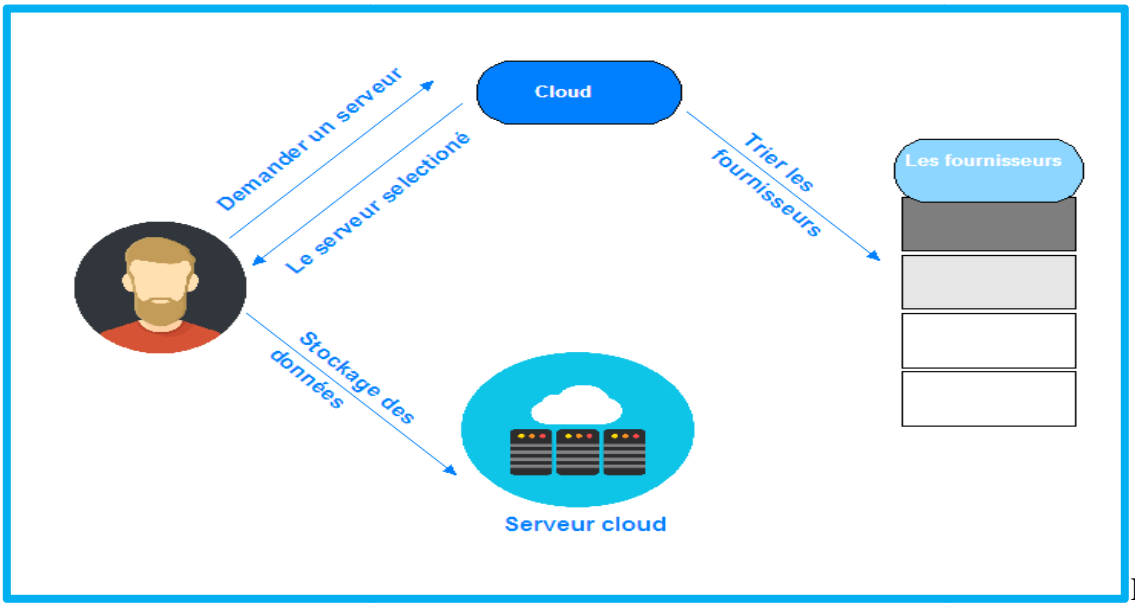

Fig. 3.6 Architecture de composant D'externalisation

- **Le rôle de composant d'externalisation**

✓Citer les meilleurs fournisseurs

✓Vérifier les demandes de l'utilisateur

✓Proposer le meilleur fournisseur selon les besoins de l'utilisateur

3. Projection sur Hadoop

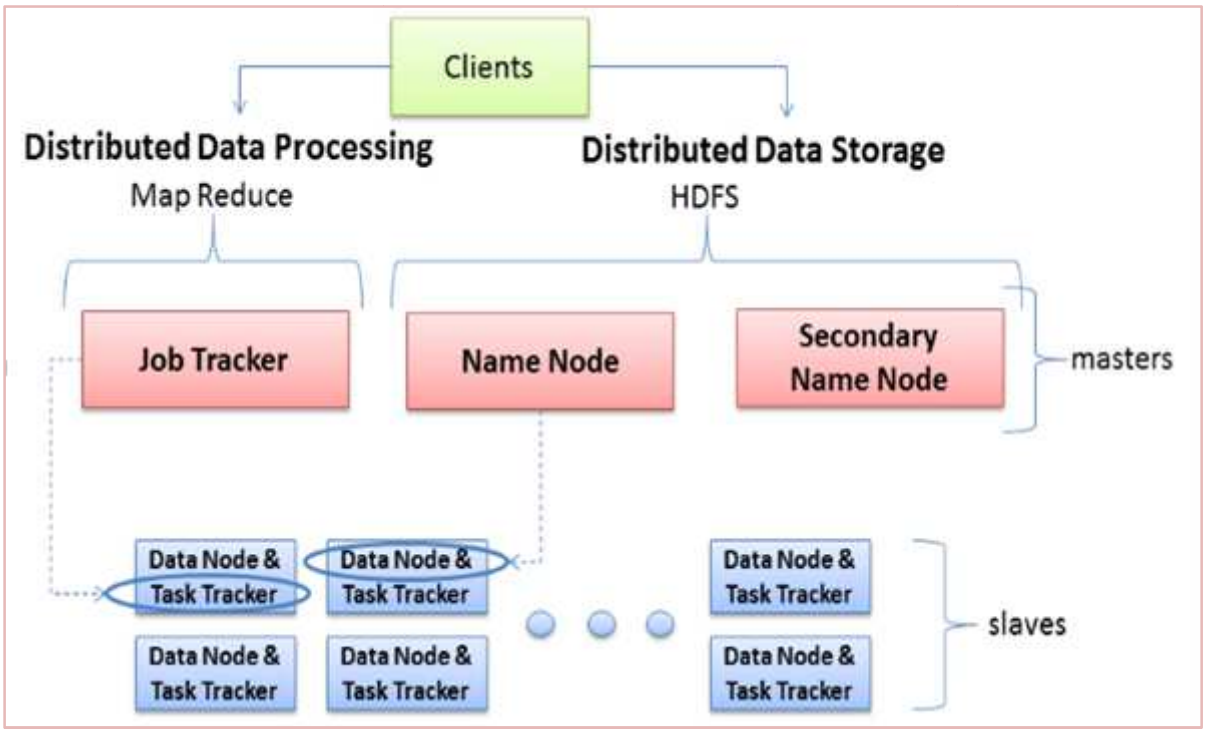

Figure.3.7 Rôles de serveur Hadoop

Les trois principales catégories de rôles de la machine dans un déploiement Hadoop sont les machines client, les nœuds *Maîtres* «Masters» et les nœuds esclaves « Slave». Les noeuds*Maîtres* super4sent les deux pièces fonctionnelles clés qui composent Hadoop : stockant beaucoup de données (HDFS) et exécutant des calculs parallèles sur toutes ces données (MapReduce). Le Name Node super4se et coordonne la fonction de stockage de données (HDFS), tandis que Job Tracker super4se et coordonne le traitement parallèle des données à l'aide de MapReduce. Les nœuds esclaves constituent la grande majorité des

machines et fait le stockage des données et d'exécuter les calculs. Chaque slave tourne à la fois un Data Node et TaskTracker qui communiquer et de recevoir des instructions de leurs nœuds Maître.

1.1 NameNode

Le NameNode dans Hadoop est le noeud où Hadoop stocke toutes les informations de localisation des fichiers dans HDFS.

1.2 Secondary Name Node

Le secondarynamenode est responsable de l'exécution des fonctions d'entretien périodiques pour leNameNode. Il ne crée que des points de contrôle du système de fichiers présents dans le NameNode.

1.3 DataNode

Le DataNode est chargé de stocker les fichiers dans HDFS. Il gère les blocs de fichiers dans le noeud. Il envoie des informations au NameNode sur les fichiers et les blocs stockés dans ce nœud et répond à la NameNode pour toutes les opérations du système de fichiers.

1.4 JobTracker

JobTracker est chargé de prendre des demandes d'un client et l'attribution des TaskTrackers avec les Task à effectuer. Le JobTracker tente d'assigner des tâches à TaskTracker sur le Data Node où les données sont présentes localement (Data Localité). Si cela est impossible, il va au moins essayer d'assigner des Task à TaskTrackers dans le même rack. Si, pour une raison quelconque, le node échoue au Job Tracker affecte la tâche à l'autre TaskTracker où la réplique des données existe depuis les blocs de données sont reproduits à travers les DataNodes. Cela garantit que le travail ne manque pas même si un nœud échoue au sein du cluster.

1.5 TaskTracker

Task Tracker accepte Task (Map, Reduce and Shuffle) de la JobTracker. Le TaskTracker continue à envoyer un message de heart beat à un Job Tracker de notifier qu'elle est 4vante. Avec le rythme cardiaque il envoie aussi les emplacements libres disponibles à l'intérieur pour traiter des tâches. TaskTracker démarre et surveille le Map&ReduceTasks et envoie progrès / informations d'état vers le Job Tracker. Le système externe travaille avec le

HDFS (Name Node, Data Node), et le système interne : l'agent scanner travaille avec Secondary Name Node et Acceslevel agent travaille avec MapReduce (JobTracker, TaskTracker).

4. Diagramme de séquence général

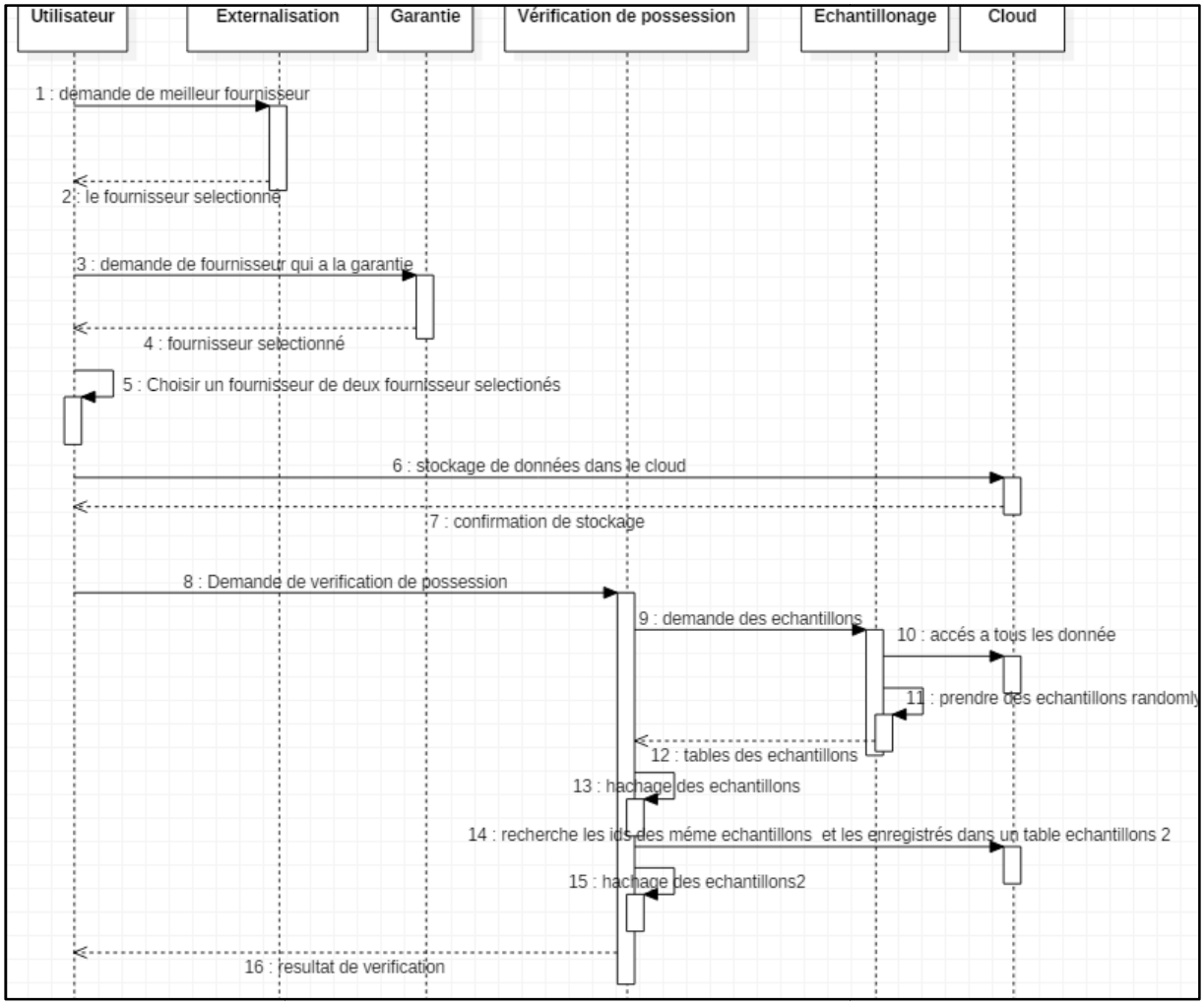

Fig. 3.8 Diagramme de séquence générale

III.1. Conclusion

Dans ce chapitre nous avons présenté notre système de protection de la possession des données. Notre architecture est composée d'un ensemble de composants. Le composant d'échantillonnage a un rôle de nous donne un nombre aléatoire des échantillons. Le composant de la garantie donne le Cloud le plus confiant. le composant Externalisation permet nous de savoir le meilleur fournisseur qui assure la sécurité. Finalement le composant de verification de possession permet au client de verifier ses données qui sont stockées dans le Cloud.

Chapitre 4 :implimentation

4.1 Introduction

Le chapitre suivant est consacré à la description des détails d'implémentation d'architecture de la possession dans le Big data. Nous commençons par la présentation des langages de programmation et les outils de développement utilisés pour la mise en œuvre du système conçu dans le chapitre précédent. Nous donnons par la suite une description textuelle et graphique de quelques interfaces du système réalisé puis l'architecture de système, la description des interfaces graphiques et enfin les principaux codes source.

4.2 Outils et langages de programmation utilisés

Pour la résiliation du système de possession dans le Big data, nous avons utilisé un langage de programmation, et quelques en4ronnements de développement. Nous les décrivons brièvement dans les sous sections suivantes.

4.2.1 Langages de programmation

- **Java**

Java est un langage de programmation informatique orienté objet créé par James Gosling et Patrick Naughton, employés de Sun Microsystems, avec le soutien de Bill Joy (cofondateur de Sun Microsystems en 1982), présenté officiellement le 23 mai 1995 au SunWorld.

La société Sun a été ensuite rachetée en 2009 par la société Oracle qui détient et maintient désormais Java.

La particularité et l'objectif central de Java est que les logiciels écrits dans ce langage doivent très facilement portables sur plusieurs systèmes d'exploitation tels que UNIX, Windows, Mac OS ou GNU/Linux, avec peu ou pas de modifications. Pour cela, divers plateformes et frameworks associés 4sent à guider, sinon garantir, cette portabilité des applications développées en Java. Outre son orientation objet, le langage Java a l'avantage d'être modulaire (on peut écrire des portions de code génériques, c'est-à-dire utilisables par plusieurs applications), rigoureux (la plupart des erreurs se produisent à la compilation et non à l'exécution) et portable (un même programme compilé peut s'exécuter sur différents environnements).

Java est un langage interprété, ce qui signifie qu'un programme compilé n'est pas directement exécutable par le système d'exploitation mais il doit être interprété par un autre programme, qu'on appelle interpréteur.

Un programmeur Java écrit son code source, sous la forme de classes, dans des fichiers dont l'extension est .java . Ce code source est alors compilé par le compilateur javac en un langage appelé bytecode et enregistre le résultat dans un fichier dont l'extension est .class. Le byecode ainsi obtenu n'est pas directement utilisable. Il doit être interprété par la machine 4rtuelle de Java qui transforme alors le code compilé en code machine compréhensible par le système d'exploitation. C'est la raison pour laquelle Java est un langage portable : le bytecode reste le même quelque soit l'en4ronnement d'exécution.

Figure 4.1 : Logo de java

4.2.2 Outils de développement

- **Netbeans**

Netbeans IDE : est un en4ronnement de développement intégré (EDI), placé en open source par Sun en juin 2000 sous licence CDDL (Common Développement and Distribution License) et GPLv2. En plus de Java, Netbeans permet également de supporter différents autres langages, comme C, C++, JavaScript, XML, Groovy, PHP et HTML de façon native ainsi que bien d'autres (comme Python ou Ruby) par l'ajout de greffons. Il comprend toutes les caractéristiques d'un IDE moderne (éditeur en couleur, projets multi-langage, refactoring, éditeur graphique d'interfaces et de pages Web).

Figure 4. 2 : Logo de netbeanse

- **Hadoop**

Hadoop est un framework libre et open source écrit en Java destiné à faciliter la création d'applications distribuées (au niveau du stockage des données et de leur traitement) et échelonnables (scalables) permettant aux applications de travailler avec des milliers de nœuds et des pétaoctets de données. Ainsi chaque nœud est constitué de machines standard regroupées en grappe. Tous les modules de Hadoop sont conçus dans l'idée fondamentale que les pannes matérielles sont fréquentes et qu'en conséquence elles doivent être gérées automatiquement par le framework.

Hadoop a été inspiré par la publication de MapReduce, GoogleFS et BigTable de Google. Hadoop a été créé par Doug Cutting et fait partie des projets de la fondation logicielle Apache depuis 2009.

Le noyau d'Hadoop est constitué d'une partie de stockage: HDFS (Hadoop Distributed File System), et d'une partie de traitement appelée MapReduce. Hadoop fractionne les fichiers en gros blocs et les distribue à travers les nœuds du cluster.

Figure 4.3 : Logo d'hadoop

- **MySQL**

MySQL est un système de gestion de bases de données relationnelles (SGBDR). Il fait partie des logiciels de gestion de base de données les plus utilisés au monde. MySQL fait référence au Structured Query Language, le langage de requête utilisé.

Figure 4.4 : Logo de MySQL.

- **La base de bigdata**

  Nous avons téléchargé une base de données pour l'utiliser comme exemple dans notre application.

4.3 Description des Interfaces Graphiques

1.1 Interface Admin login

Cette interface est pour que l'administrateur accède à son compte :

Figure 4.5 : Interface Admin login

1.2 Interface User login

Cette interface est pour que l'utilisateur accède à son compte :

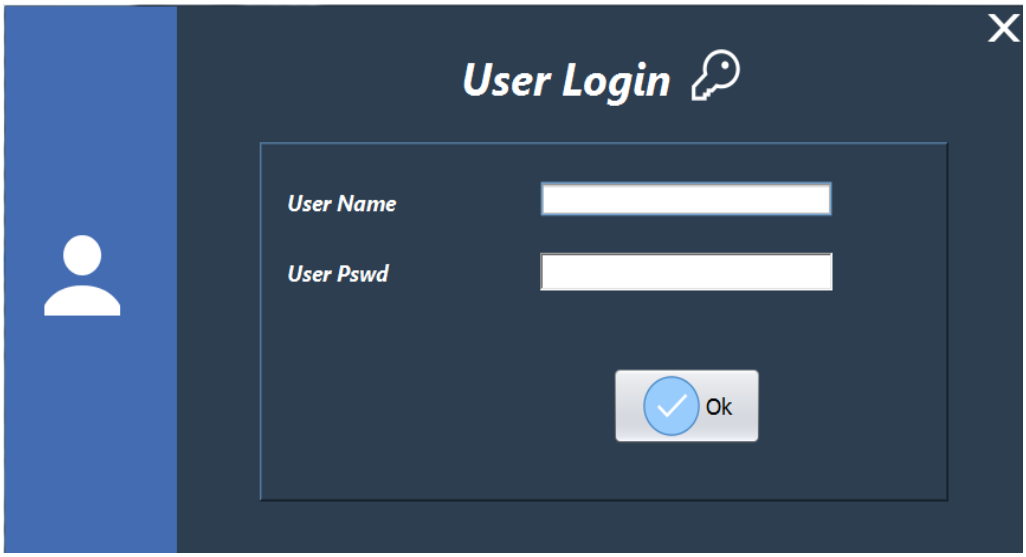

Figure 4.6 : Interface User Login

1.3 Interface Externalisations

Dans cette interface l'utilisateur peut connaitre le meilleur fournisseur :

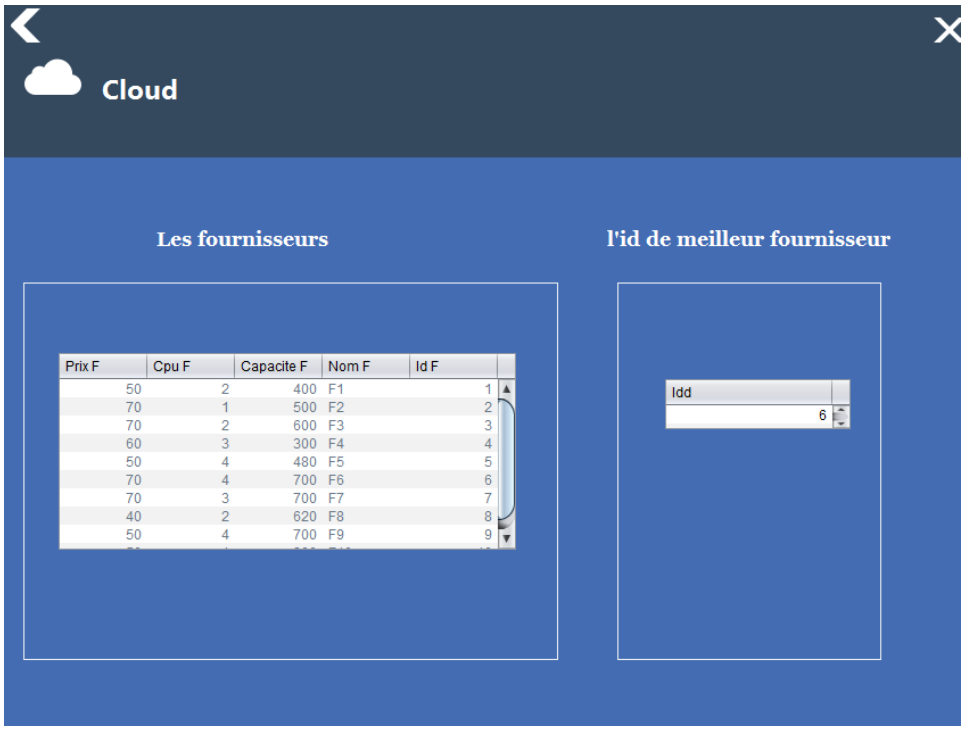

Figure 4.7 : Interface Externalisations

1.4 Interface Garantie

Dans cette interface l'utilisateur peut connaitre le meilleur fournisseur qui a une garantie

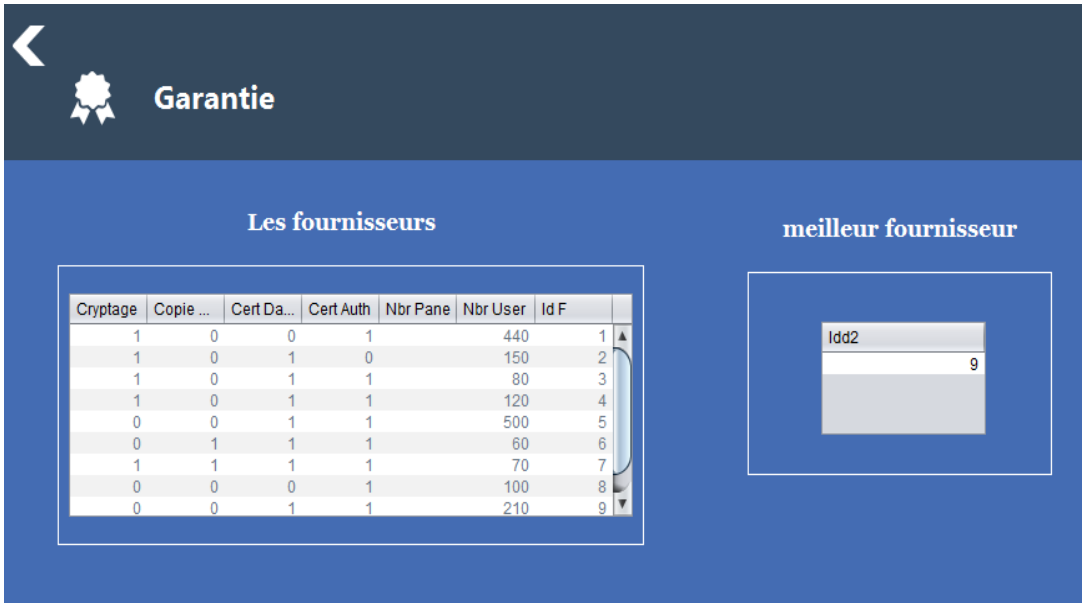

Figure 4.8 : Interface Garantie

1.5 Interface Meilleur fournisseur

Dans cette interface l'utilisateur peut connaitre les meilleur fournisseurs, voir les échantillons et verifier la possession des données.

Figure 4.9 : Interface Meilleur Fournisseurs

4.4 Les principaux codes source

A. Composant Échantillonnage

```
int [][] Echantionnagedatabase = new int [2][100];


    Random random= new Random();
    int nb;
    nb=random.nextInt(10)+1;
   int j=0;


    for (int i = 0; i< 100; i++)
    {
      if (Globaldatabase[1][i]>=1 & Globaldatabase[1][i]<=10)
     {
       Echantionnagedatabase[0][j]=Globaldatabase[0][i];
```

```
        Echantionnagedatabase[1][j]=Globaldatabase[1][i];
        j++;

       }
       if (j==nb)
       {
          break;
       }

     }


for (int i = 0; i< 100; i++)
    {
      if (Globaldatabase[1][i]>=11 & Globaldatabase[1][i]<=20)
      {
       Echantionnagedatabase[0][j]=Globaldatabase[0][i];
       Echantionnagedatabase[1][j]=Globaldatabase[1][i];
       j++;
      }
      if (j==2*nb)
      {
         break;
      }
    }
 for (int i = 0; i< 100; i++)
    {
      if (Globaldatabase[1][i]>=21 & Globaldatabase[1][i]<=30)
      {
       Echantionnagedatabase[0][j]=Globaldatabase[0][i];
       Echantionnagedatabase[1][j]=Globaldatabase[1][i];
       j++;

      }
```

```
          if (j==3*nb)
          {
             break;
          }
        }
    for (int i = 0; i< 100; i++)
        {
          if (Globaldatabase[1][i]>=31 & Globaldatabase[1][i]<=40)
          {
           Echantionnagedatabase[0][j]=Globaldatabase[0][i];
           Echantionnagedatabase[1][j]=Globaldatabase[1][i];
           j++;
          }
          if (j==4*nb)
          {
             break;
          }
        }
for (int i = 0; i< 100; i++)
        {
          if (Globaldatabase[1][i]>=41 & Globaldatabase[1][i]<=50)
          {
           Echantionnagedatabase[0][j]=Globaldatabase[0][i];
           Echantionnagedatabase[1][j]=Globaldatabase[1][i];
           j++;
          }
          if (j==5*nb)
          {
             break;
          }
        }



   for (int i = 0; i< 100; i++)
```

```
    {
      if (Globaldatabase[1][i]>=51 & Globaldatabase[1][i]<=60)
      {
       Echantionnagedatabase[0][j]=Globaldatabase[0][i];
       Echantionnagedatabase[1][j]=Globaldatabase[1][i];
       j++;
      }
      if (j==6*nb)
      {
         break;
      }
    }
for (int i = 0; i< 100; i++)
    {
      if (Globaldatabase[1][i]>=61 & Globaldatabase[1][i]<=70)
      {
       Echantionnagedatabase[0][j]=Globaldatabase[0][i];
       Echantionnagedatabase[1][j]=Globaldatabase[1][i];
       j++;
      }
      if (j==7*nb)
      {
         break;
      }
    }
for (int i = 0; i< 100; i++)
    {
      if (Globaldatabase[1][i]>=71 & Globaldatabase[1][i]<=80)
      {
       Echantionnagedatabase[0][j]=Globaldatabase[0][i];
       Echantionnagedatabase[1][j]=Globaldatabase[1][i];
       j++;
      }
      if (j==8*nb)
```

```
        {
            break;
        }
    }
for (int i = 0; i< 100; i++)
    {
      if (Globaldatabase[1][i]>=80 & Globaldatabase[1][i]<=90)
      {
       Echantionnagedatabase[0][j]=Globaldatabase[0][i];
       Echantionnagedatabase[1][j]=Globaldatabase[1][i];
       j++;
      }
      if (j==9*nb)
      {
         break;
      }
    }
for (int i = 0; i< 100; i++)
    {
      if (Globaldatabase[1][i]>=91 & Globaldatabase[1][i]<=100)
      {
       Echantionnagedatabase[0][j]=Globaldatabase[0][i];
       Echantionnagedatabase[1][j]=Globaldatabase[1][i];
       j++;
      }
      if (j==10*nb)
      {
         break;
      }
    }
```

B. Composant Externalisation

```
int maxcapacité;int idmax=0;
  int k=0;
  int ii=0;
  //maxcapacité=temp[1][0];
    maxcapacité=700; // a parcourire la colone concerné pour connaiotre le maximlmum afin de fair l'initialisation

  while(k<10)
  {
  System.out.println(temp[1][k]);
  if(temp[1][k]>= maxcapacité){ // toujours égale après une initialisation correct

    // maxcapacité=temp[1][k];
    //idmax=temp[0][k];

    // System.out.println("le max de capacité est:  "+ maxcapacité);
    bestpro4derstorage[ii]=temp[0][k];
    ii++;

  }
  k++;
  }



    int zz=0;
  //int maxcpu=temp[2][bestpro4derstorage[0]];
   int maxcpu=0; // a parcourire la colone concerné pour connaiotre le maximlmum afin de fair l'initialisation

  for (int jj=0; jj<ii; jj++)
```

```
 {
    if(temp[2][bestpro4derstorage[jj]-1]>= maxcpu){ // toujours égale après une initialisation correct
       bestpro4dercpu[zz]=temp[0][bestpro4derstorage[jj]-1];
       maxcpu = temp[2][bestpro4derstorage[jj]-1];
       zz++;
    }

 }
        //   System.out.println("best one :"+bestpro4dercpu[zz-1]);


  int yy=0;
 //int maxprix=temp[3][bestpro4dercpu[0]];
 int minprix=1000; // a parcourire la colone concerné pour connaiotre le minimum afin de fair l'initialisation

 for (int jj=0; jj<zz; jj++)
 {
    if(temp[3][bestpro4dercpu[jj]-1]<= minprix){ // toujours égale après une initialisation correct
       bestpro4derprix[yy]=temp[0][bestpro4dercpu[jj]-1];
       minprix = temp[3][bestpro4dercpu[jj]-1];
       yy++;
    }

 }
           //System.out.println("best one :"+bestpro4dercpu[yy-1]);


  for (int jj=0; jj<yy; jj++)
  {
    System.out.println("best one :"+ jj +"is:"+ bestpro4derprix[jj]);
```

```
   }
```

C. Composant Garantie

```
int maxnbrUser;int idmax=0;
  int k=0;
  int ii=0;
  //maxcapacité=temp[1][0];
     maxnbrUser=700;

     while(k<10)
  {
  System.out.println(temp[1][k]);
  if(temp[1][k]>= maxnbrUser){ // toujours égale après une initialisation correct

     // maxcapacité=temp[1][k];
     //idmax=temp[0][k];

     // System.out.println("le max de capacité est:  "+ maxcapacité);
     bestpro4derNbrUser[ii]=temp[0][k];
     ii++;

  }
  k++;
  }
 // System.out.print("ffffffffffffffff"+ bestpro4derNbrUser[0]);

  int zz=0;
  //int maxcpu=temp[2][bestpro4derstorage[0]];
    int maxcertAUTH=0; // a parcourire la colone concerné pour connaiotre le maximlmum afin de fair l'initialisation
```

```
 for (int jj=0; jj<ii; jj++)
 {
    if(temp[2][bestpro4derNbrUser[jj]-1]> maxcertAUTH){ // toujours égale après une initialisation correct
       bestpro4derCertauth[zz]=temp[0][bestpro4derNbrUser[jj]-1];
       maxcertAUTH = temp[2][bestpro4derNbrUser[jj]-1];
       zz++;
    }
      }

// System.out.print("ffffhhhhhhhhhhhhhhhhh"+ bestpro4derCertauth[0]);
 // System.out.print("rrrrrrrrr"+ temp[3][bestpro4derCertauth[0]]);

int zz2=0;
 //int maxcpu=temp[2][bestpro4derstorage[0]];
  int maxcertdaccr=0; // a parcourire la colone concerné pour connaiotre le maximlmum afin de fair l'initialisation

 for (int jj=0; jj<zz; jj++)
 {
    if(temp[3][bestpro4derCertauth[jj]-1]> maxcertdaccr){ // toujours égale après une initialisation correct
       bestpro4derCertdaccr[zz2]=temp[0][bestpro4derCertauth[jj]-1];
       maxcertdaccr = temp[3][bestpro4derCertauth[jj]-1];
       zz2++;
    }
      }

 //System.out.print("gggggggggggggggg"+  bestpro4derCertdaccr[0]);

 int zz3=0;
 //int maxcpu=temp[2][bestpro4derstorage[0]];
```

```
  int maxcopiered=0; // a parcourire la colone concerné pour connaiotre le maximlmum afin de fair l'initialisation

 for (int jj=0; jj<zz2; jj++)
 {
    if(temp[4][bestpro4derCertdaccr[jj]-1]> maxcopiered){ // toujours égale après une initialisation correct
       bestpro4derCopiered[zz3]=temp[0][bestpro4derCertdaccr[jj]-1];
       maxcopiered = temp[4][bestpro4derCertdaccr[jj]-1];
       zz3++;
    }
      }

 int zz4=0;
 //int maxcpu=temp[2][bestpro4derstorage[0]];
  int maxcryptage=0; // a parcourire la colone concerné pour connaiotre le maximlmum afin de fair l'initialisation

 for (int jj=0; jj<zz3; jj++)
 {
    if(temp[5][bestpro4derCopiered[jj]-1]> maxcryptage){ // toujours égale après une initialisation correct
       bestpro4dercryptage[zz4]=temp[0][bestpro4derCopiered[jj]-1];
       maxcryptage = temp[5][bestpro4derCopiered[jj]-1];
       zz4++;
    }
      }


  int zz5=0;
 //int maxprix=temp[3][bestpro4dercpu[0]];
 int minnbrpane=1000; // a parcourire la colone concerné pour connaiotre le minimum afin de fair l'initialisation
```

```
 for (int jj=0; jj<zz4; jj++)
 {
    if(temp[6][bestpro4dercryptage[jj]-1]< minnbrpane){ // toujours égale après une initialisation correct
       bestpro4derNbrpane[zz5]=temp[0][bestpro4dercryptage[jj]-1];
       minnbrpane = temp[6][bestpro4dercryptage[jj]-1];
       zz5++;
     }

 }


 System.out.println("best one :"+ bestpro4derNbrpane[0]);
```

4.5 Conclusion

Dans ce chapitre nous avons présenté les étapes de la mise en œuvre de notre projet avec tous les outils, les langages et les plateformes utilisés ainsi que la présentation avec l'explication du rôle de chaque outil.

Pendant l'implémentation de ce travail nous suggérons le développement de la possession dans le big Data sous les nouveaux plateformes et outils, des nouvelles technologies qui représentent la tendance aujourd'hui.

Conclusion

Les BigDatas, c'est avant tout une formidable opportunité pour les entreprises d'innover, de Développer leurs ventes, leurs bénéfices, leurs marchés, d'adresser de nouveaux clients, et de Créer de nouvelles offres.

Pour les clients de ces entreprises et les consommateurs, c'est l'assurance d'une meilleure Expérience client dans toutes leurs interactions que ce soit au niveau marketing, commercial ou au niveau du service client.

La démarche Bigdata permet de collecter, de stocker, et d'analyser toutes ces données à des coûts nettement plus raisonnables qu'avant grâce à des technologies nouvelles de stockage et surtout d'analyse.

Contribution

Dans notre projet, nos principales contributions sont :

- ✓ Nous avons étudié la possession dans et hors les BigDatas dans des divers articles pour pouvoir faire une étude comparative entre toutes les démarches.
- ✓ Déceler quelques inconvénients pour chaque approche.
- ✓ Nous avons proposé une nouvelle architecture pour supporter Hadoop afin de réaliser la bonne solution possible pour la possession dans les BigDatas.

Perspectives

Comme perspectives, L'ajout d'un composant de contrôle à l'architecture pour pouvoir :

- ✓ Faire des tests sur tous les autres composants.
- ✓ Vérifier le bon fonctionnement des autres composants.
- ✓ Déterminer ceux défaillants et vérifier les nœuds des données.

Erratum

- On s'excuse du non clarté de quelque signe.
- On remercie d'avance toute personne qui nous signalera, les erreurs qu'il pourrait déceler, à l'adresse suivante : souflimanel@gmail.com

Bibliographie

[1]

[2] Lisbeth Rodríguez-Mazahua , Cristian-Aarón Rodríguez-Enríquez, José Luis Sánchez-Cervantes , Jair Cervantes, JorgeLuis García-Alcaraz ,Giner Alor-Hernández ,(2015), "A general perspective of Big Data: applications, tools, challenges and trends".

[3] Ekaterina Olshannikova, Aleksandr Ometov, Yevgeni Koucheryavy ,Thomas Olsson,(2016),), " 4sualizing Big Data" ,p104-107.

[4] Ismael Caballero, Manuel Serrano, and Mario Piattini (2014), " A Data Quality in Use Model for Big Data(Position Paper)", p67-68.

[5] Manas Kumar Sanyal, Sajal Kanti Bhadra and Sudhangsu Das,(2016), "A Conceptual Framework for Big Data Implementation to Handle Large Volume of Complex Data", p459-464.

[6] Hamza Saouli, Kazar Okba, Dounya Kassimi, (2016), « Applications et enjeux des Big Data dans le contexte des défis mondiaux ».

[7] Min Chen · Shiwen Mao · Yunhao Liu, (22 January 2014), " Big Data: A Survey", p175,176.

[8] Shuyu Li and Jerry Gao,(2016), " Security and Privacy for Big Data", p304-307.

[9] Seymour bosworth, M.E.Kabay, Eric Whyne,(2014), "COMPUTER SECURITY HANDBOOK", 111-115.

[10] Jan Camnisch,Simone fischer-Hubner,(2014),"Privacy and identity Management for the futur internet in the Age of globalisation "p42-43.

[11] "Big data and data protection", (1998), p40-42.

[12] Sithu D Sudarsan, Raoul P Jetley, Srini Ramaswamy, (2015), "Security and Privacy of Big Data", p134-135,131-132, 124-125, 126-128.

[13] Camnisch,Simone fischer-Hubner,(2014)," Privacy and identity Management for the futur internet in the Age of globalisation ", p41-42.

[14] Leslie P. Francis , " Introduction :Technology and New Challenges for Privacy".

[15] Sithu D Sudarsan, Raoul P Jetley, Srini Ramaswamy, (2015), "Security and Privacy of Big Data", p124-125.

[16] Shuyu Li and Jerry Gao,(2016)" Security and Privacy for Big Data",p283-292.

[17] Shuyu Li and Jerry Gao,(2016)" Security and Privacy for Big Data",p300-304.

[18] Stephen Flowerday, Tamir Tsegaye,(2014)" Controls for Protecting Critical Information Infrastructure from Cyberattacks",p24-25.

[19] Sithu D Sudarsan, Raoul P Jetley, Srini Ramaswamy, (2015), "Security and Privacy of Big Data", p126-128.

[20] Stephen Flowerday, Tamir Tsegaye,(2014)" Controls for Protecting Critical Information Infrastructure from Cyberattacks",p27-28.

[21] Ajit Gaddam, " Securing Your Big Data En4ronment ".

[22] Kjell Johan Sæbø, (2009), "Possession and pertinence: the meaning of have".

[23] Chang Liu, Jinjun Chen, Senior Member, IEEE, Laurence T. Yang, Member, IEEE, Xuyun Zhang, Chi Yang, Rajiv Ranjan, and Ramamohanarao Kotagiri, (SEPTEMBER 2014), "Authorized Public Auditing of Dynamic Big Data Storage on Cloud with Efficient Verifiable Fine-Grained Updates".

[24] Z. Zou1 and Q. Kong2,3 *1Dongguan Research Institute of CASIA, Cloud Computing Center, Chinese Academy of Sciences, Dongguan, China 2The State Key Laboratory of Management and Control for Complex Systems, Institute of Automation, Chinese Academy of Sciences, Beijing, China 3Qingdao Academy of Intelligent Industries, Qingdao, China*, (2017), " Secure provable data possession for big data storage ".

[25] Debiao He , Neeraj Kumar , Huaqun Wang , Lina Wang , Kim-Kwang Raymond Choo , (2017), " Privacy-preser4ng certificateless provable data possession scheme for big data storage on cloud ".

[26] Lukasz Krzywiecki, Krzysztof Majcher, and Wojciech Macyna,(2016) "Efficient Probabilistic Methods for Proofof Possession in Clouds".

[27]Ge Yao1, Yong Li1, Linan Lei1, Huaqun Wang, and Changlu Lin,(2016)" An Efficient Dynamic Provable Data Possession Scheme in Cloud Storage".

[28] Changlu Lin, Fucai Luo, Huaxiong Wang and Yan Zhu,(2016)" A Provable Data Possession Scheme with Data Hierarchy in Cloud".

[29] Huaqun Wang, and Debiao He,(2016)" Proxy Provable Data Possession with General Access Structure in Public Clouds".

[30] Enguang Zhou and Zhoujun Li,(2014)" An Improved Remote Data Possession Checking Protocol in Cloud Storage".

[31] Xiajun Yu,Qiaoyan Wen," A Multi-Function Provable Data Possession Scheme in cloud Computing".

[32] Wenting Shen, Guangyang Yang, Jia Yu, Hanlin Zhang, Fanyu Kong, and Rong Hao,(2017)" Remote Data Possession Checking with Privacy-Preser4ng Authenticators for Cloud Storage".

[33] Giuseppe Ateniese, Randal Burns,Reza Curtmola, Joseph Herring, Lea Kissner , Zachary Peterson,Dawn Song ,(2007)" Provable Data Possession at Untrusted Stores".

[34] Giuseppe Ateniese, Randal Burns,Reza Curtmola, Joseph Herring, Lea Kissner , Zachary Peterson,Dawn Song ,(2007)" Provable Data Possession at Untrusted Stores",p600.

[35] Changlu Lin, Fucai Luo, Huaxiong Wang and Yan Zhu,(2016)" A Provable Data Possession Scheme with Data Hierarchy in Cloud",p320.

[36] Xiajun Yu,Qiaoyan Wen," A Multi-Function Provable Data Possession Scheme in cloud Computing",p17.

.