

RAPPORT DE STAGE

Effectué au Centre de Recherche



Préparé par

Jmal Doha

Ktata Abderrahim

TITRE DU STAGE :

STOCKAGE ET VISUALISATION DES TRAMES

Encadré par :

M. Tarak Chaari

Encadrant industriel

M. Hassene Mrabet

Encadrant académique

Année Universitaire 2016-2017

Remerciement

*Nous tenons tout d'abord à remercier notre encadrant **Monsieur Hassene Mrabet** et notre encadrante **Madame Olfa ben ayed**, pour sa disponibilité, ses conseils et son aide à la réalisation de ce projet.*

*Nous adressons nos sincères remerciements à **Monsieur Tarak Chaari**, pour son accueil, et sa disponibilité, son aide à la réalisation de ce projet, ses conseils et sa collaboration pour la réalisation et la direction de ce travail.*

Nous ne manquerons pas de remercier toutes les personnes qui nous ont encouragé et aidé de près ou de loin à l'élaboration de ce travail.

Doha & Abderrahim

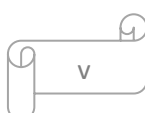
Table des matières

Introduction Générale	1
Chapitre 1 : Présentation du cadre de projet	4
Introduction	5
1 Présentation	5
2 Présentation du projet	5
2.1 Cadre du projet	5
2.2 Problématiques	5
2.3 Objectifs du projet	6
Conclusion	7
Chapitre 2 : Spécification des besoins et conception	8
Introduction	9
1 Spécification des besoins	9
1.1 Les Diagrammes de cas d'utilisations	9
1.1.1 Identification des acteurs	9
1.1.2 Description des cas d'utilisation	10
2 Conception détaillée	14
2.1 Présentation UML	14
2.2 Conception générale de l'application	14
2.2.1 Les Diagrammes de séquences	14
Conclusion	16
Chapitre 3 : Réalisation	17
Introduction	18
1 Environnement de réalisation	18
1.1 Environnement matériel	18
1.2 Environnement logiciel	18
1.3 Langages de programmation	20
2 Vue architecturale du système	21
3 Présentation des interfaces graphiques de l'application	21
3.1 Interface d'accueil	21
3.1.1 Programme Java	21
3.1.2 Hadoop	22
3.1.3 Configuration de l'Apache Drill	23
3.1.4 Apache Drill	24
3.1.5 Connection avec Apache Drill	25
3.1.6 Connection OpenBouquet	27
3.1.7 Visualisation Open bouquet	28

Conclusion	28
Conclusion Générale et Perspective	29

Liste des tableaux

TABEAU 1 - DESCRIPTION TEXTUELLE DU CAS D'UTILISATION « ENVOIE DES TRAMES »	10
TABEAU 2 - DESCRIPTION TEXTUELLE DU CAS D'UTILISATION « ENREGISTREMENT DES TRAMES ».....	11
TABEAU 3 - DESCRIPTION TEXTUELLE DU CAS D'UTILISATION « VISUALISATION DES TRAMES ».....	13
TABEAU 4 - DESCRIPTION TEXTUELLE DU CAS D'UTILISATION « AUTHENTIFICATION »	14
TABEAU 5 - CONFIGURATION TECHNIQUES DES ORDINATEURS.....	18



Liste des figures

FIGURE 1 - DIAGRAMME DE CAS D'UTILISATION DE KAFKA CONSUMER	9
FIGURE 2 - DIAGRAMME DE CAS D'UTILISATION DE L'UTILISATEUR	12
FIGURE 3 - DIAGRAMME DE SEQUENCE DE RECEVOIR LES TRAMES	15
FIGURE 4 - DIAGRAMME DE SEQUENCE DE VISUALISATION OPENBOUQUET	16
FIGURE 5 - VUE ARCHITECTURALE DU SYSTEME	21
FIGURE 6 - PROGRAMME JAVA	22
FIGURE 7 – INTERFACE DE HADOOP	23
FIGURE 8 - CONFIGURATION DU DRILL	24
FIGURE 9 - AFFICHAGE DU TABLEAU	25

Introduction Générale

C'est indéniable, dans le Big Data (ou grandes données), les volumes d'informations manipulées sont colossaux. Voilà pourtant un nom mal choisi. Cette expression fourre-tout désigne des données dont la taille ne permet pas de les caser dans les conteneurs habituels. Elle fait référence à des informations qui sont trop volumineuses pour tenir sur un seul serveur, qui sont dépourvues d'une structure leur permettant d'être stockées dans des bases de données orientées lignes et colonnes, ou qui arrivent sous la forme d'un flux permanent empêchant leur conservation dans un entrepôt de données statique. L'accent est souvent mis sur le volume, mais la complexité des grandes données réside en réalité dans le manque de structure. En général, les ouvrages sur le Big Data indiquent dès les premières lignes la quantité de données disponibles dans le monde. Vous avez certainement déjà lu ces chiffres et connaissez les comparaisons habituelles. Une entreprise de taille normale dispose d'un volume d'informations égal à 427 fois celui de la bibliothèque du Congrès. Le nombre de données photographiques détenues par Facebook est supérieur au nombre de pixels jamais traités par Kodak. Chaque jour produit plus de vidéos que les cinquante premières années de la télévision. Si ces allégations sur les volumes et les types des données sont imaginaires, elles ne sont probablement pas très loin de la vérité.

IoT Tunisia est un Groupement à Intérêt Economique à but non lucratif (GIE) en cours de création qui rassemble académiciens, industriels et donneurs d'ordre ayant un intérêt particulier pour l'Internet des objets (IoT) et aux multiples applications innovantes des villes intelligentes notamment dans les secteurs de l'Agriculture, l'Environnement, l'Énergie, la Santé, la Sécurité et le Transport.

IoT Tunisia a pour mission de:

- Créer un Consortium groupant les différents acteurs (Écoles, universités, laboratoires de recherche, industries, parcs technologiques, associations, experts) afin de :
- Encourager le développement des nouvelles technologies et applications IoT et promouvoir leur utilisation.
- Collaborer dans les réponses aux appels à proposition de projets de recherche et d'innovation à l'échelle nationale et internationale

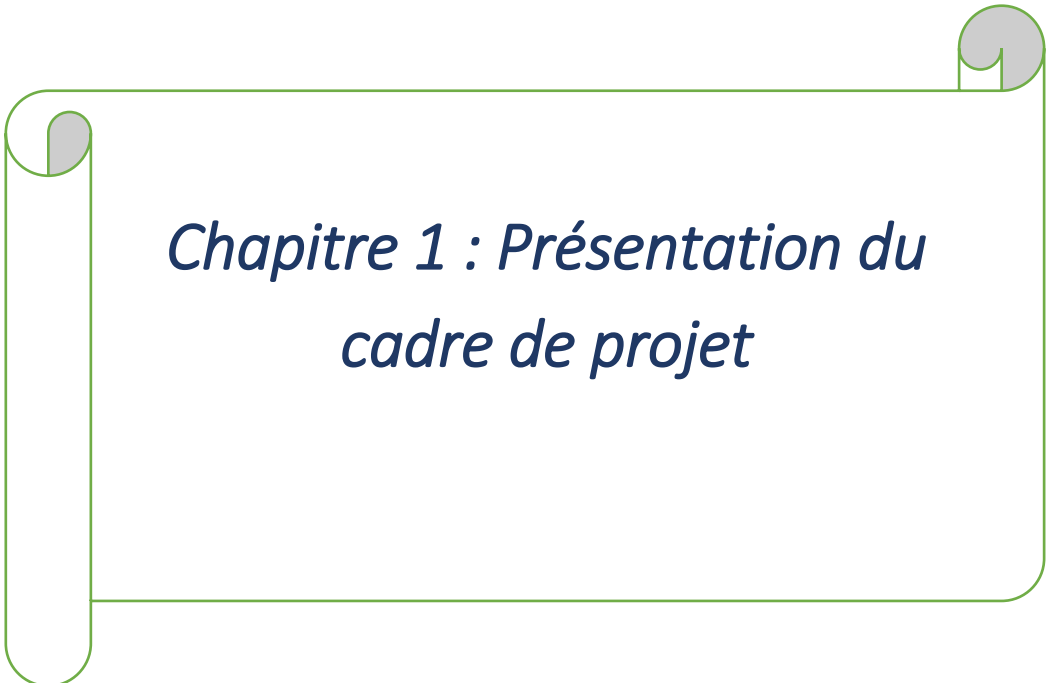
- Coopérer avec les instances nationales et les donneurs d'ordre pour faire de l'IoT un axe stratégique dans le développement de la Tunisie (Lancer des projets smart, simplifier les procédures administratives telles que l'importation des équipements et l'utilisation de bande de fréquences, etc.);
- Valoriser et disséminer les résultats des études et projets de l'IoT à l'échelle nationale et internationale (séminaires, salons, etc.);
- Encourager les jeunes à promouvoir leurs solutions IoT et à créer des start-up en attirant les investisseurs (challenge, concours internationaux, etc.).

Ce rapport est le témoin de deux mois de stage au sein de centre de recherche Sfax sous la tutelle de **Monsieur Tarak Chaari** dans le cadre d'une 2^{ème} GLID. L'objectif de ce stage était de développer une partie d'application pour IOT Sfax. En effet, cette application de plateforme IOT qui permet de récupérer les trames de lora.

Notre mission consiste en première étape à établir un programme java pour kafka pour qu'il arrive à envoyer des trames à Hadoop HDFS et qui seront enregistrées dans ce dernier dans un fichier. En second lieu , nous configurons Drill pour que puisse comprendre Hadoop HDFS et récupérer les données en les mettant dans un tableau, ceci sera par une facilité d'utilisation à travers les requêtes SQL. Ensuite nous établirons encore une fois le programme java entre Drill et Openbouquet pour identifier .l'URL du apache Drill. Nous terminerons donc par la configuration de l'openbouquet avec cet URL pour eu arriver à la visualisation. Ce rapport est composé de trois Chapitres :

- Le premier Chapitre portera sur l'étude Préalable qui contient :
 - ✓ La première porte sur la présentation du centre de recherche Sfax.
 - ✓ La seconde présente le Cadre du projet.
 - ✓ La troisième présente les Problématiques.
 - ✓ La quatrième présente les objectifs.
- Le deuxième Chapitre sur la Conception du site :
 - ✓ Les diagrammes de cas d'utilisations.
 - ✓ Les diagrammes séquences.
- Le troisième Chapitre sur étudie techniques :
 - ✓ La première porte sur l'environnement matériel.
 - ✓ La seconde prote sur l'environnement Logiciel.
 - ✓ La troisième porte sur les Langages de développement.

- ✓ La quatrième porte sur les principales interfaces graphiques.
- ✓ En fin ,la cinquième partie est composée de la conclusion de ce rapport ,aussi Annexe et la bibliothèque que l'on utiliser.



Chapitre 1 : Présentation du cadre de projet

Présentation du cadre de projet

Introduction

Dans ce chapitre, nous exposons le cadre général du projet. Nous avons commencé dans un premier temps par la présentation de l'organisme d'accueil, ensuite par le travail demandé pour en finir par l'explication de la démarche de travail que nous avons suivie.

1 Présentation

La biotechnologie occupe une place grandissante dans les pays industrialisés. Cette place est essentiellement due à l'importance qu'elle occupe dans les domaines de l'Agriculture, de l'Alimentation, de l'Environnement et de la Santé. Ceci ne fait que rendre plus urgente sa nécessité pour les pays en voie de développement, pour lesquels ces domaines sont vitaux, de se doter des moyens nécessaires pour maîtriser cette technologie.

La Tunisie a franchi ce pas par la mise en œuvre d'un plan Biotechnologique. Ce plan s'est matérialisé entre autres par la création du Centre de Biotechnologie de Sfax (CBS).

Le CBS a été créé par le décret n° 83-1037 du 4 novembre 1983, réorganisant l'Institut National de la Recherche Scientifique et Technique. Depuis janvier 1989, le CBS est devenu une institution indépendante dont le budget est rattaché pour ordre au budget de l'Etat.

2 Présentation du projet

2.1 Cadre du projet

Ce stage s'inscrit dans le cadre d'un projet de 2^{ème} année spécialité génie logiciel et informatique décisionnelle. Le sujet est intitulé « Stockage et Visualisation des trames ».

Ce projet était proposé par Mr **Tarak chaari** qui nous a intégrés au sein de son équipe afin de nous rapprocher de ses ingénieurs et de nous familiariser avec la vie professionnelle.

2.2 Problématiques

L'Internet des objets, ou Internet of Things (IoT) en anglais, transformera l'ensemble de la société, y compris nous-mêmes. À première vue, cette affirmation peut paraître exagérée, mais pensons à l'impact qu'a déjà eu l'internet sur l'enseignement, les communications, les entreprises, la science, les organismes publics et les hommes. L'internet est sans doute l'une des inventions les plus importantes et les plus significatives de toute l'histoire de l'humanité. L'IoT représente la prochaine évolution d'Internet et permettra d'améliorer considérablement

sa capacité à rassembler, à analyser et à restituer des données que nous pourrons ensuite transformer en informations, en connaissances. Des projets IoT déjà en cours promettent de combler les écarts, d'améliorer la distribution des ressources mondiales aux populations défavorisées et de nous aider à comprendre notre planète, ce qui nous permettra d'adopter un comportement plus proactif au lieu de simplement réagir aux événements. Toutefois, plusieurs obstacles menacent de ralentir le développement de l'IoT ,on rappelle comme exemple des milliards de minuscules capteurs qui lui envoient énorme donnée chaque seconde.

2.3 Objectifs du projet

A cause des objets connectés permettant de faire le lien entre le monde réel et le monde numérique. Ils sont souvent nombreux, plus ou moins chers et plus ou moins connectés. Ils génèrent beaucoup de données que l'on veut stocker et traiter. Cette session sera l'occasion de voir comment nous pouvons répondre efficacement à ces problématiques.

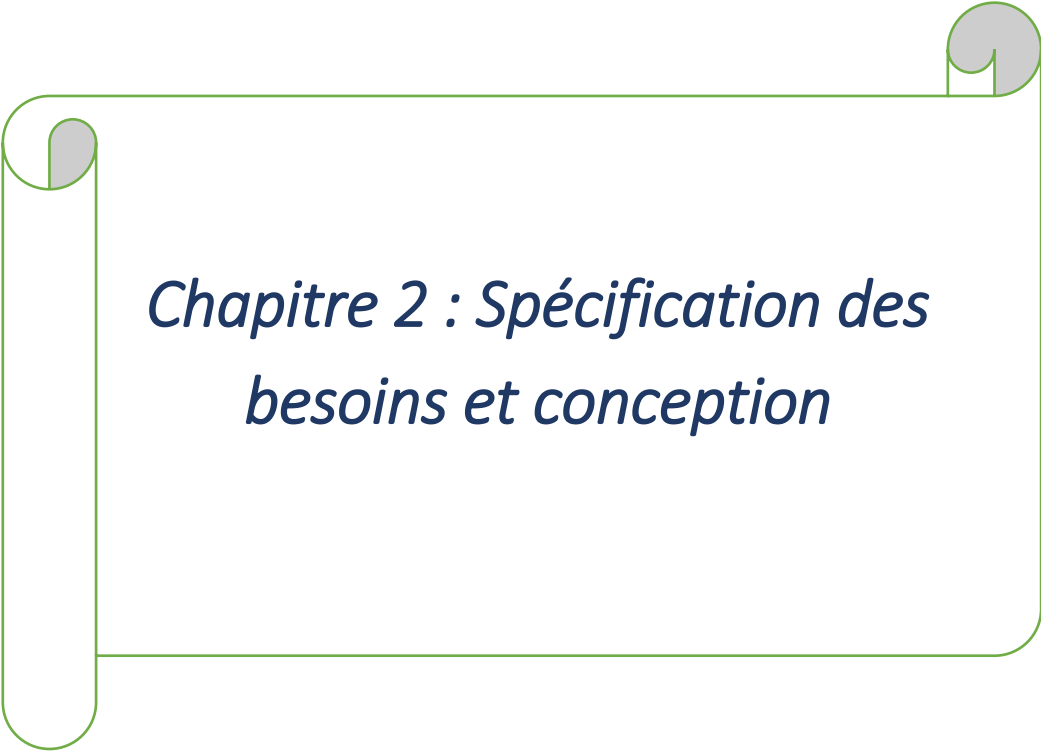
Pour cela nous présentons dans ce qui suit les objectifs à atteindre pour maximiser la performance globale de notre future solution. Ces objectifs se résument dans les points suivants:

- Mise en place Apache Hadoop pour :
 - Garantir la disponibilité et la durabilité des données, par réplication. C'est une approche logicielle à contre-courant des solutions matérielles traditionnelles (RAID, SAN, ...).
 - Garantir une scalabilité linéaire des capacités de stockage et de traitement par simple ajout de machine. Stockage et traitement sont distribués et co-localisés
 - Apporter une capacité à traiter des données peu ou pas structurées.
- Mise en place Apache Drill pour :
 - Augmenter le framework, faciliter l'usage ou tout simplement corriger certaines lacunes qui sont considérées comme un turbo pour Hadoop.
 - Permettre d'effectuer des requêtes interactives SQL au dessus d'Hadoop, afin de donner accès aux utilisateurs.
 - Réaliser des analyses à partir de données stockées dans HDFS et Hadoop.
- Mise en place Openbouquet pour :
 - Utiliser la plateforme analytiques pour la business intelligence(une technologie Big Data puissante pour les entreprises, solide, et collaborative), qui permet aux entreprises de pouvoir construire puis déployer des applications métiers orientées utilisateur.

Conclusion

Dans ce chapitre introductif, nous avons pu présenter le contexte général dans lequel s'inscrit notre projet. Ce qui nous a permis de souligner les problématiques qui en découlent, de proposer une solution adéquate et de préciser la méthodologie adoptée.

Nous allons entamer dans le chapitre suivant les principaux concepts et les outils de l'informatique ainsi que la méthodologie de travail utilisée pour réaliser notre solution.



Chapitre 2 : Spécification des besoins et conception

Spécification des besoins et conception

Introduction

L'identification des besoins du projet est l'une des étapes importantes dans le cycle de vie d'un logiciel. Elle mène à la détermination des besoins fonctionnels et non fonctionnels avec lequel les utilisateurs s'interagissent avec le système. Nous consacrerons donc ce chapitre à dégager les différentes fonctionnalités de l'application à réaliser. Nous commencerons par l'identification des besoins fonctionnels et non fonctionnels et nous clôturerons ce chapitre par la conception détaillée.

1 Spécification des besoins

Dans ce qui suit, nous allons définir les fonctionnalités que le système doit offrir à ses utilisateurs et les interactions qui peuvent exister entre elles. Ces spécifications ont pour but de mieux comprendre les besoins fonctionnels et les besoins non fonctionnels de l'application.

1.1 Les Diagrammes de cas d'utilisations

Le diagramme de cas d'utilisation est un diagramme UML utilisé pour donner une vision globale du comportement fonctionnel du système logiciel. Ce diagramme décrit les différentes actions qui peuvent être effectuées par un acteur dans un système.

1.1.1 Identification des acteurs

Un acteur représente un rôle joué par une entité externe (utilisateur humain, dispositif matériel ou un système). L'acteur du système est le Kafka Consumer.

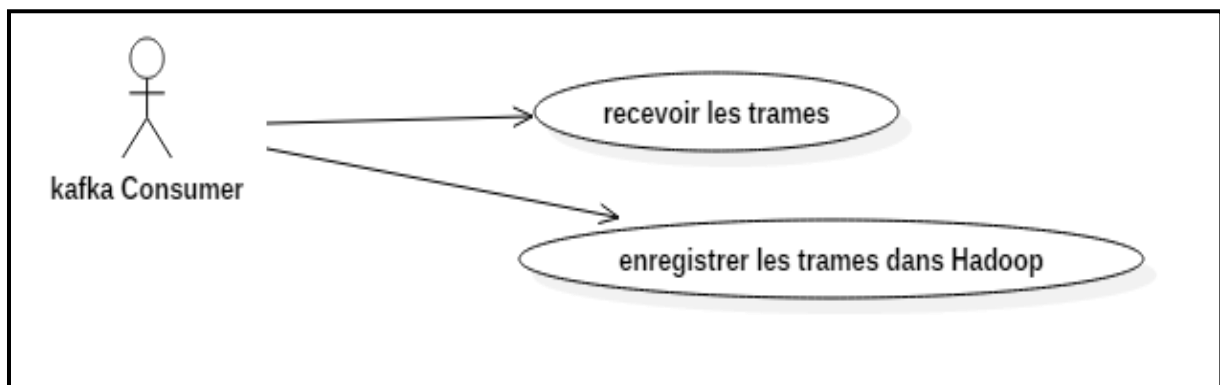


Figure 1 - Diagramme de cas d'utilisation de Kafka consumer

1.1.2 Description des cas d'utilisation

En vue de bien expliquer les cas d'utilisation de notre application, nous commençons par la description des principaux cas d'utilisation dans les tableaux suivants :

Envoie des Trames
Résumé : <i>toutes les trames envoie avec succès</i>
Acteur : <i>Kafka consumer</i>
Précondition : - <i>Tous Lora devices est fonctionnel.</i> - <i>Serveurs configurer et en marche.</i>
DESCRIPTION DU SCENARIO NOMINAL : <i>« DEBUT »</i> 1- <i>Les devices envoient les trames au Gateway chaque 30 secondes qui à son tour les transfère vers le serveur</i> 2- <i>Le serveur ouvre le canal d'écoute pour le Gateway sur un port bien configuré</i> 3- <i>Le serveur est à la réception d'une trame envoyée.</i> <i>« FIN »</i>
DESCRIPTION DU SCENARIO Alternatif : <i>A1 : le serveur tombe en panne.</i> <ul style="list-style-type: none"> - <i>Chercher et vérifier les fonctionnalités possibles pour déterminer ce problème</i> - <i>Une fois le problème est trouvé on le résoudre.</i> - <i>Tester la d'enchaînement native.</i> <i>A2 : le Gateway en panne.</i> <ul style="list-style-type: none"> - <i>Vérifier la sécurité de tous les composants.</i> - <i>Vérifier la batterie.</i> - <i>Détacher le problème</i> - <i>Résoudre et tester le composant</i>

Tableau 1 - Description textuelle du cas d'utilisation « Envoie des Trames »

Enregistrement des Trames
Résumé : <i>toutes les trames enregistrées dans Hadoop</i>
Acteur : <i>Kafka consumer</i>
Précondition : - <i>Toutes les trames envoient avec succès</i> - <i>Serveurs en fonction</i>
DESCRIPTION DU SCENARIO NOMINAL : <i>« DEBUT »</i> 1- <i>Le serveur reçoit les trames.</i> 2- <i>Le serveur divise la trame via un broker kafka.</i> 3- <i>L'utilisateur kafka consumer détermine toutes les trames reçues et enregistre dans hadoop.</i> <i>« FIN »</i>
DESCRIPTION DU SCENARIO Alternatif : <i>A1 : la trame reçue n'est pas enregistrée dans hadoop.</i> - <i>Vérifier que le service hadoop est fonctionnel.</i> + <i>si le service ne fonctionne pas alors le met en marche</i> + <i>si non on vérifie la configuration du service et on le configure selon le besoin</i>

Tableau 2 - Description textuelle du cas d'utilisation « Enregistrement des Trames »

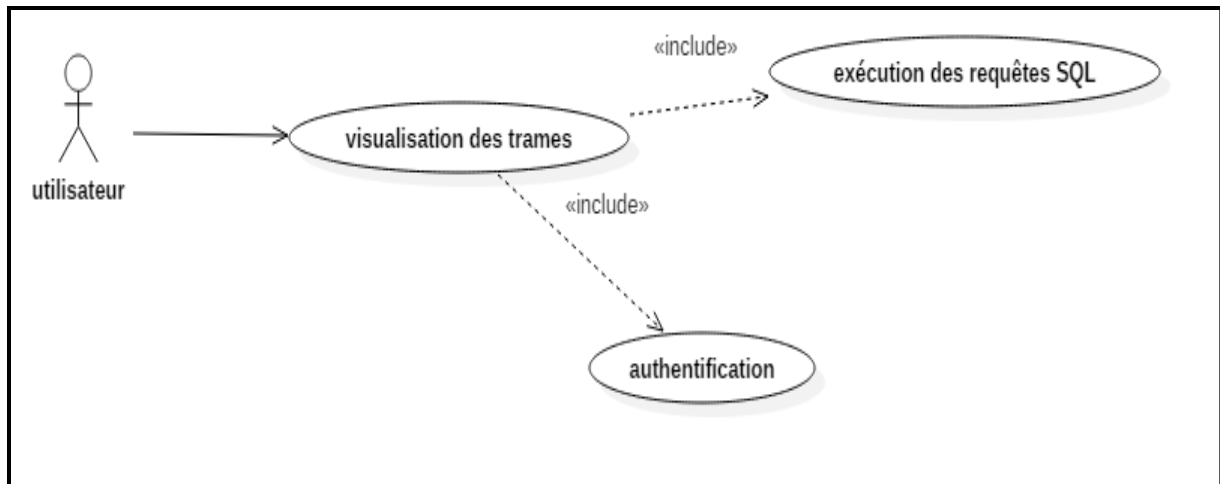


Figure 2 - Diagramme de cas d'utilisation de l'utilisateur

Visualisation des Trames

Résumé : *visualisation de Dashboard*

Acteur : *Utilisateur*

Précondition :

- *trames enregistrées dans Hadoop*
- *Base des données testée avec une requête Sql .*
- *Authentification*

DESCRIPTION DU SCENARIO NOMINAL :

« *DEBUT* »

- 1- *L'utilisateur demande une page de visualisation sur Openbouquet*
- 2- *Le système affiche toutes les bases des données*
- 3- *L'utilisateur choisi une base pour faire la visualisation*
- 4- *L'utilisateur doit ajouter les axes de visualisation*
- 5- *L'utilisateur ajoute les opérations de visualisation*
- 6- *L'utilisateur doit choisir deux axes de visualisation et une opération*
- 7- *Le système affiche un dashboard selon les axes choisis*
- 8- *L'utilisateur peut choisir un autre type de dashboard même un tableau*

« *FIN* »

DESCRIPTION DU SCENARIO Alternatif :

A1 : base des données n'existe pas.

- *L'utilisateur doit vérifier l'existence de la base de données avec Hadoop et la tester avec Drill*
- *Si la base des données existe l'utilisateur doit vérifier la configuration. Sinon il doit créer une base et la remplir par des trames ou il doit vérifier la configuration de connexion et il retourne à étape 1.*

Tableau 3 - Description textuelle du cas d'utilisation « Visualisation des trames »

Authentification

Résumé : *permet à l'utilisateur de s'identifier afin d'avoir les privilèges nécessaires pour utiliser l'application.*

Acteur : *Utilisateur*

Précondition : - *L'utilisateur est enregistré dans le système et un couple (login, password) lui sera affecté.*

- *L'utilisateur connaît ses paramètres d'authentification.*

DESCRIPTION DU SCENARIO NOMINAL :

« DEBUT »

9- L'utilisateur veut se connecter à l'application

10- Le système affiche la fenêtre d'authentification

11- L'utilisateur saisit son login et son mot de passe

12- Le système vérifie les informations d'authentification

13- Le système affiche les menus adéquats correspondant à ce type d'utilisateurs.

« FIN »

DESCRIPTION DU SCENARIO Alternatif :

A1 : le champs login et/ou mot de passe est vide

- *Le système affiche un message d'erreur « remplir les champs vides »*
- *Le système reprend au point 3.*

Tableau 4 - Description textuelle du cas d'utilisation « Authentification »

2 Conception détaillée

Dans cette section, nous détaillons la conception de notre application en se basant sur trois phases principales. Dans la première, nous décrivons la conception générale de l'application dans la deuxième, nous étudions la conception de la phase d'intégration des données. Et dans la troisième, nous finissons avec une conception de la phase d'analyse des données. Pour une modélisation conceptuelle parfaite, nous avons choisi le standard UML "Unified Modeling Language".

2.1 Présentation UML

UML est un langage formel couvrant le cycle de développement de logiciel de la spécification à l'implémentation. De plus, il offre aux utilisateurs la possibilité de définir leurs besoins du système à développer : ce qui permet de mieux répondre à leurs exigences et à leurs attentes. Enfin, il décrit plusieurs types de modèles (générique, expressifs, flexibles...). En effet, nous avons choisi UML pour les spécifications des utilisateurs à travers les diagrammes de cas d'utilisation et les interactions des acteurs avec notre système à travers les diagrammes de séquence. Dans ce qui suit, nous allons détailler les notions de bases pour quelques diagrammes que nous avons appliqué durant la phase de conception de notre application.

2.2 Conception générale de l'application

2.2.1 Les Diagrammes de séquences

Un diagramme de séquences est un diagramme d'interaction qui expose en détail la façon dont les opérations sont effectuées : quels sont les messages qui sont envoyés et quand ils le sont. Les diagrammes de séquences sont organisés en fonction du temps.

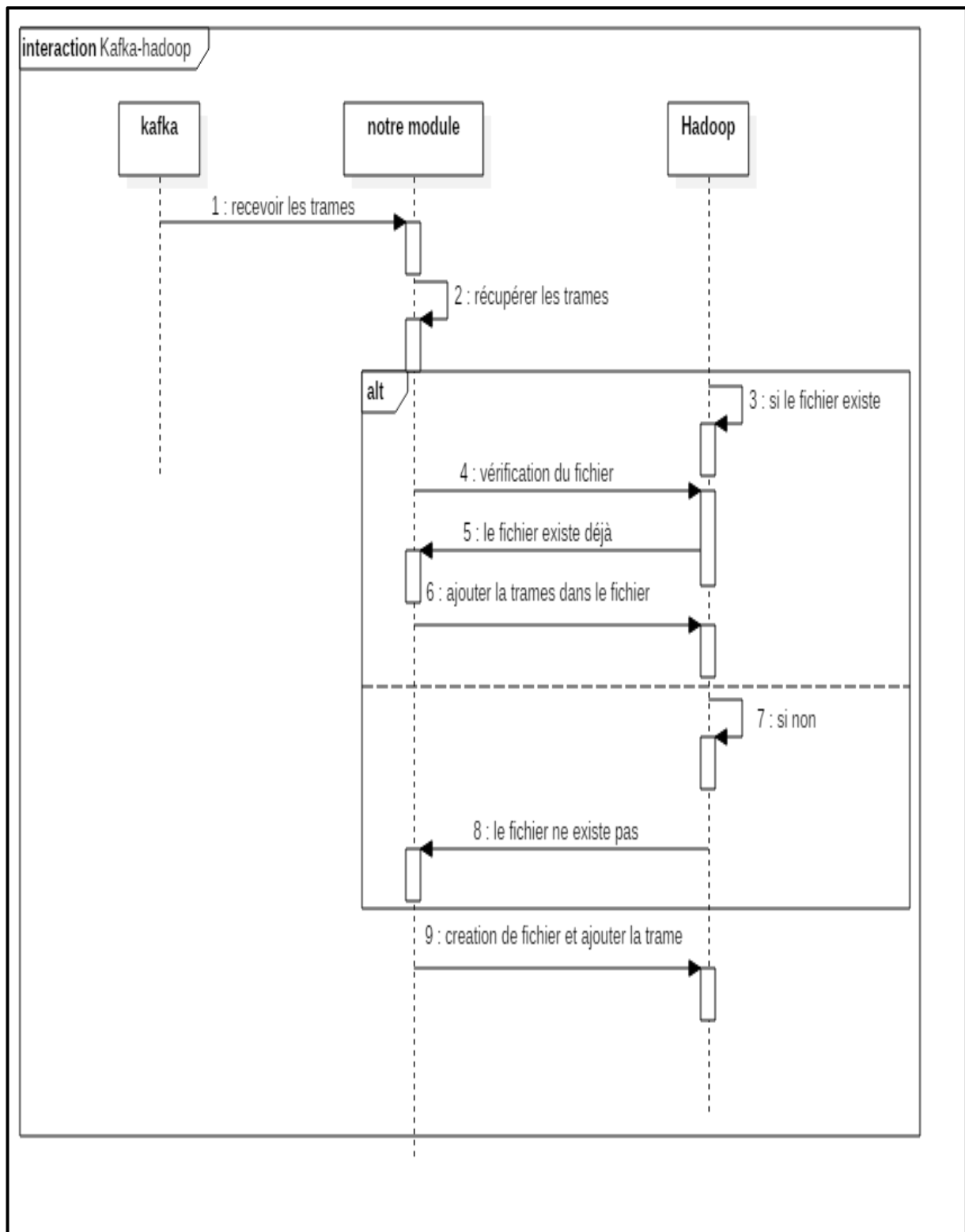


Figure 3 - Diagramme de séquence de recevoir les trames

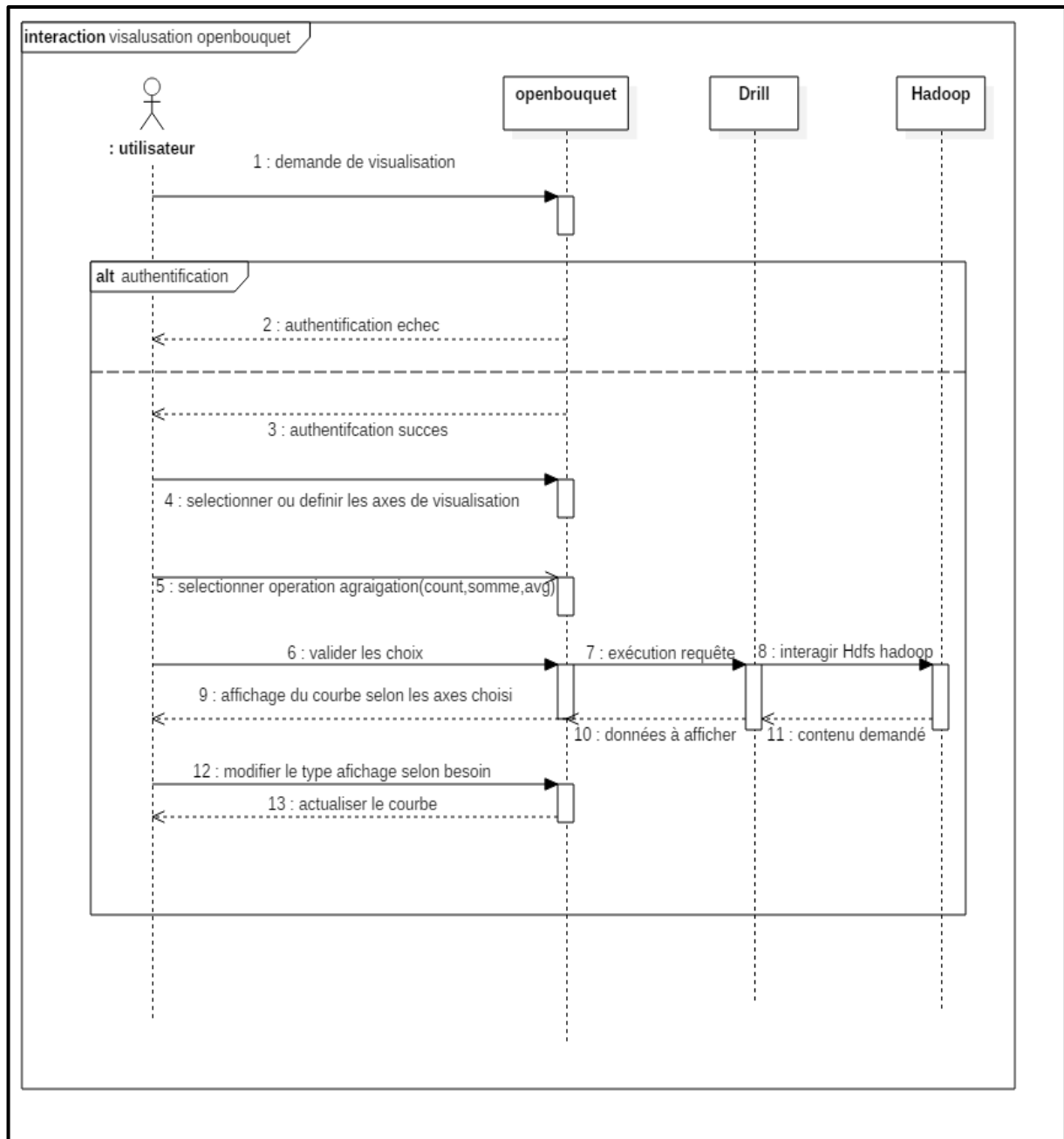


Figure 4 - Diagramme de séquence de Visualisation Openbouquet

Conclusion

Dans ce chapitre, nous avons mis une conception détaillée dans laquelle nous avons présenté le diagramme de cas d'utilisation, les diagrammes de séquences, les diagrammes d'activités et le diagramme de navigation de notre système.

Dans le chapitre suivant, nous allons présenter l'environnement de réalisation de notre solution et décrire d'une façon précise les étapes nécessaires pour réaliser la migration.



Chapitre 3 : Réalisation

Réalisation

Introduction

Dans ce dernier chapitre, nous mettons l'accent sur l'étape de la réalisation du projet en se basant sur l'étude de la conception traitée dans le chapitre trois. Nous commencerons par la présentation du choix de l'environnement de réalisation. Ensuite, nous présenterons le choix technologique, scénario d'exécution et les différentes interfaces de l'application à travers des captures d'écran.

1 Environnement de réalisation

Dans ce qui suit, nous présentons l'environnement matériel et logiciel ainsi que les différents langages de programmation que nous avons utilisé pour réaliser notre solution.

1.1 Environnement matériel

Tout au long des différentes étapes d'implémentation de notre application, à savoir l'étude théorique, la spécification des besoins, la conception, le développement et le test, deux postes de travail ont été utilisés pour le développement de l'application. Ces postes ont les configurations suivantes :

Caractéristiques	Processeur	Mémoire vive	Disque dur	Système d'exploitation
ASUS	Intel core i7	12 Go	500 GB	Windows Vista
ASUS	Intel core i7	8 Go	1 TB	Windows Vista

Tableau 5 - Configuration Techniques des Ordinateurs

1.2 Environnement logiciel

Durant la réalisation de l'application nous avons utilisé les outils suivants :

Windows 10 : est le système d'exploitation de Microsoft qui a vu le jour en 2015. Cette nouvelle version a été pensée pour fonctionner aussi bien



sur un ordinateur standard que sur une tablette tactile. Elle a été très bien accueillie par les utilisateurs et a rattrapé le fiasco engendré avec Windows 8.

Apache Kafka : est un système de messagerie distribué, originellement développé chez LinkedIn, et maintenu au sein de la fondation Apache depuis 2012. Son adoption n'a cessé de croître pour en faire un quasi de-facto standard dans les pipelines de traitement de données actuels. Bien plus qu'un simple concurrent des outils conçus autour des standards JMS ou AMQP, Kafka a pour ambition de devenir **la plateforme centralisée de stockage et d'échange de toutes les données émises par une entreprise en temps réel**.



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.



Apache Drill : est un framework logiciel open-source qui supporte les applications temps réel distribuées pour l'analyse interactive des jeux de données à grande échelle. Drill est la version open source du système Dremel de Google qui est disponible comme un service d'infrastructure appelé Google BigQuery. Drill supporte de nombreuses bases NoSQL et système de fichiers comme HBase, MongoDB, MapR-DB, HDFS, et fichiers locaux. Une seule requête peut joindre des données d'entrepôts de données différents. Par exemple, vous pouvez joindre le profil utilisateur présent dans une collection sur MongoDB avec les logs d'Hadoop.



Eclipse est un projet, décliné et organisé en un ensemble de sous-projets de développements logiciels, de la fondation Eclipse visant à développer un environnement de production de logiciels libre qui soit extensible, universel et polyvalent, en s'appuyant principalement sur Java.



Boîte à outils d'analyse open-source pour explorer, partager et connecter vos données à des applications et des visualisations.



Slack est certes un peu difficile à définir. Il s'agit d'un mélange entre une salle de chat AOL, Hangouts Google et des messages texte. Comme les internautes utilisent déjà Hangouts, iMessage, des d'autres plateformes, Slack semble être un excès.



La gestion de projet est une démarche visant à organiser de bout en bout déroulement d'un projet. C'est tout l'opérationnel et le tactique qui fait qu'un projet aboutit dans un triangle présentant l'équilibre qualité-coût-délai (QCD). Le management de projet assume le pilotage stratégique du projet.



Git, outil de gestion de versions de code source, s'est répandu très rapidement dans la communauté Open Source, de part sa rapidité, sa flexibilité et sa fiabilité. Les solutions d'hébergement en ligne de projets Git connaissent un énorme succès, mais leur utilisation est bien souvent payante pour l'hébergement de projets privés. GitLab est un clone à Github. Pour celles et ceux qui ne connaîtraient pas Github, GitLab est une solution gratuite et Open Source pour l'hébergement de projets Git privés comparable aux géants tels que GitHub et BitBucket et qui convient aux entreprises.



1.3 Langages de programmation

C'est un langage de programmation orienté objet, développé par Sun Microsystems. Il permet de créer des logiciels compatibles avec de nombreux systèmes d'exploitations (Windows, Linux, Macintosh, Solaris). Java donne aussi la possibilité de développer des programmes pour téléphones portables et assistants personnels. Enfin, ce langage peut être utilisé sur internet pour des petites applications intégrées à la page web (applet) ou encore comme langage serveur (jsp).



2 Vue architecturale du système

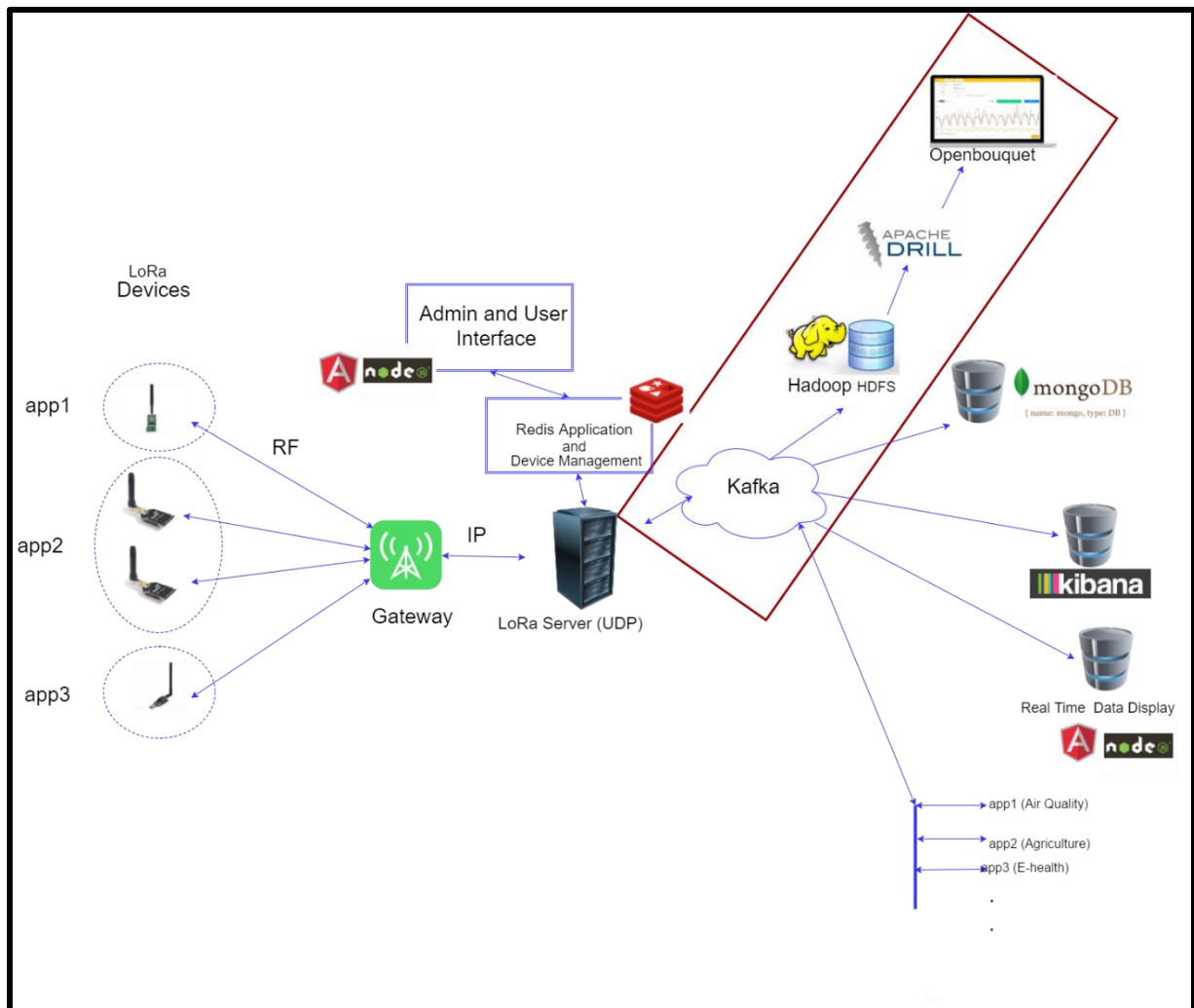


Figure 5 - Vue architecturale du système

3 Présentation des interfaces graphiques de l'application

Dans cette partie, nous allons présenter les captures d'écran de notre application.

3.1 Interface d'accueil

3.1.1 Programme Java

Nous avons créé un programme java intermédiaire entre Kafka et Hadoop, il fait la récupération de la trame et la sauvegarder dans Hadoop dans un fichier json.

```

package hadoop;
import java.io.IOException;
import java.io.OutputStream;
import java.io.PrintStream;
import java.io.PrintWriter;
import java.net.URI;
import java.net.URISyntaxException;

import org.apache.hadoop.conf.Configuration;
import org.apache.hadoop.fs.FSDataOutputStream;

import org.apache.hadoop.fs.FileSystem;

import org.apache.hadoop.fs.Path;
import org.apache.hadoop.hdfs.DFSOutputStream;
import org.apache.hadoop.util.Progressable;

public class Hdfs {
    String fs_defaultFS = "hdfs://localhost:9000";
    String USERNAME = "hdfs";

    FSDataOutputStream outputStream = null;
    Path path;
    FileSystem hdfs;

    public Hdfs(String fileName) throws IOException, URISyntaxException {
        String sURI = fs_defaultFS + "/" + fileName;
        System.setProperty("HADOOP_USER_NAME", USERNAME);
        path = new Path(sURI);
        hdfs = FileSystem.get(new URI(fs_defaultFS), new Configuration());

        hdfs.setReplication(path, (short) 1);
    }

    public boolean write(String trame) throws IOException, URISyntaxException {
        try {
            if (hdfs.exists(path)) {
                // outputStream = hdfs.append(/n);

                outputStream = hdfs.append(path);
                System.out.println("append");
            } else {
                outputStream = hdfs.create(path);
                System.out.println("create");
            }
            (new PrintStream(outputStream)).println(trame);
            System.out.println("new");

            return true;
        } finally {
            if (null != outputStream) {
                outputStream.close();
            }
        }
    }

    public static void main(String[] args) throws IOException,
        URISyntaxException {
        Hdfs hd = new Hdfs("/hadoop/hdfs.json");
        try {
            hd.write("hdfs");
        } catch (URISyntaxException e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
        }
    }
}

```

Figure 6 - Programme Java

3.1.2 Hadoop

A la suite de l'exécution de programme JAVA et la récupération de donner, tous les trames sont enregistrées dont Hadoop, Hadoop affiche l'existence de fichier avec les

permissions de l'utilisateur, il peut afficher aussi le contenu avec le découpage en bloc du fichier.

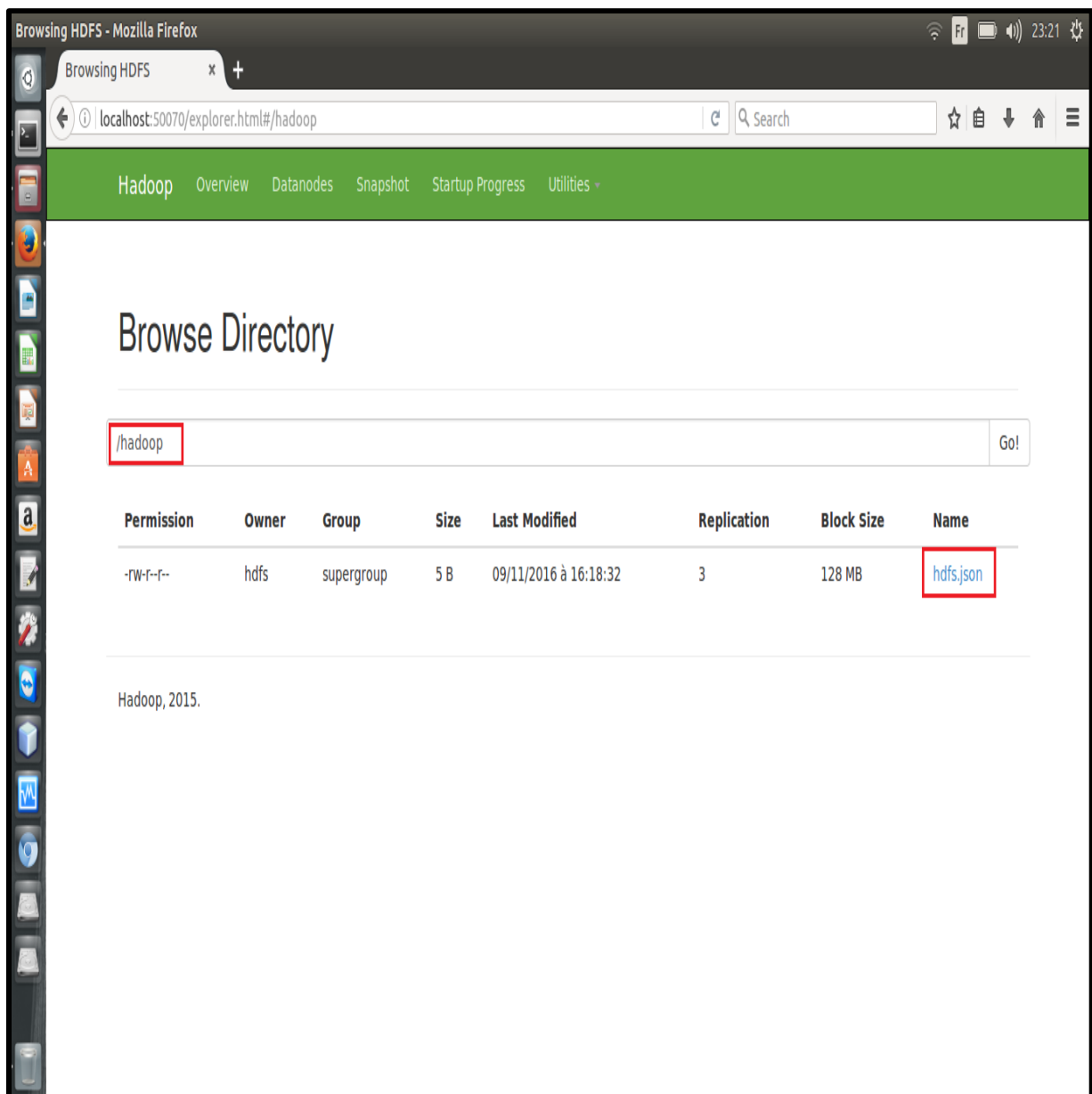


Figure 7 – Interface de Hadoop

3.1.3 Configuration de l'Apache Drill

Cette figure représente la configuration avec Hadoop dans l'apache Drill pour l'interaction avec des requête SQL.

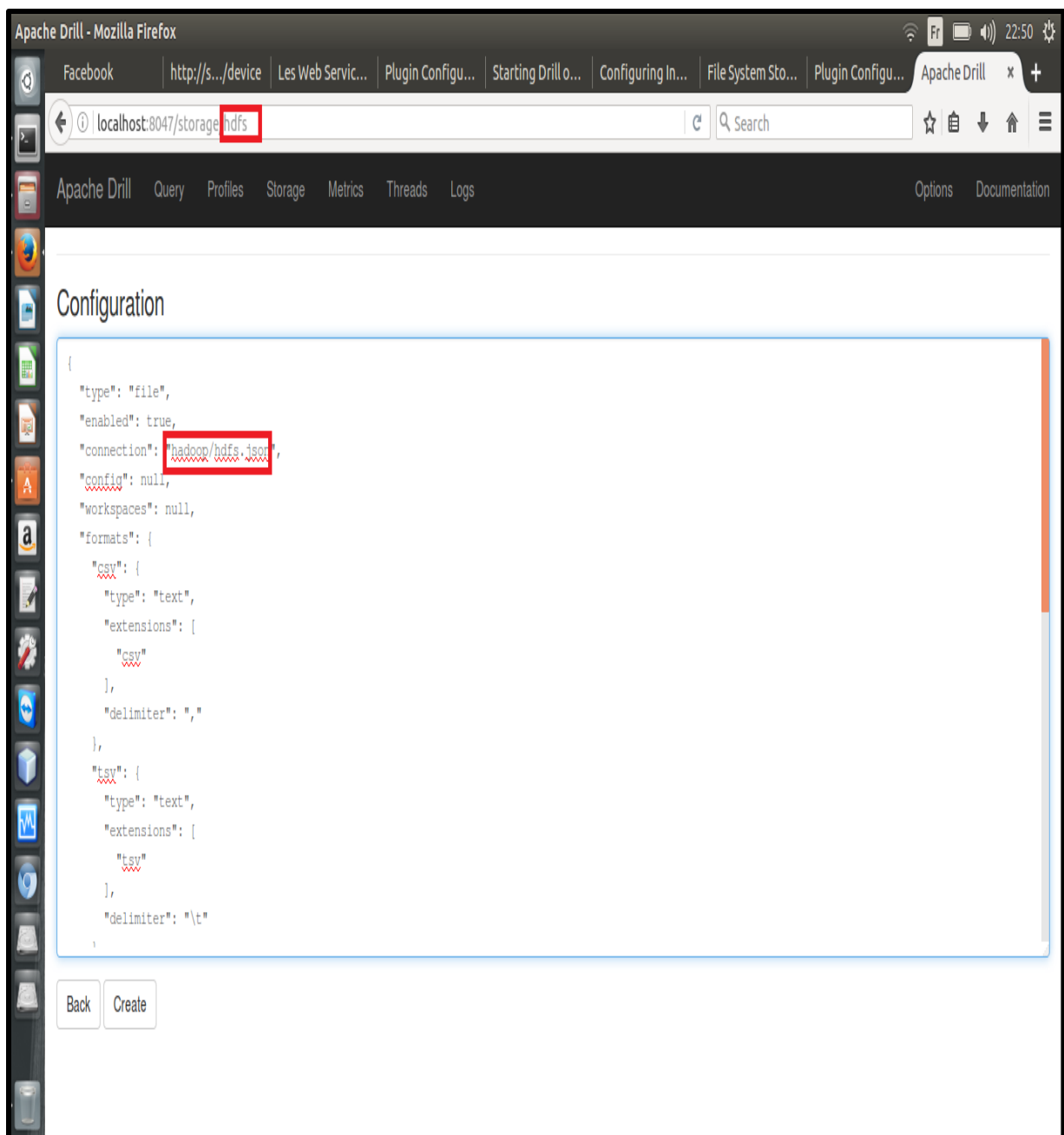


Figure 8 - Configuration du Drill

3.1.4 Apache Drill

Cette figure représente un exemple d'exécution de requête sql dont apache drill. Tous ces données sont retirées de Hadoop

Figure 9 - Affichage du tableau

3.1.5 Connection avec Apache Drill

Durant le cursus de ce stage nous avons trouvé plusieurs problèmes dont la dernière phase "Visualisation de donnée en fonction de courbe" alors on à essayer avec d'autre méthode, la première étape établir une connections avec apache Drill en programme JAVA.

```

package drill;

import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;

public class ApacheDrillJDBCClient {
    static final String JDBC_DRIVER = "org.apache.drill.jdbc.Driver";
    static final String DB_URL = "jdbc:drill:drillbit=localhost:31010";

    static final String USER = "admin";
    static final String PASS = "admin";

    public static void main(String[] args) {
        Connection conn = null;
        Statement stmt = null;
        try{
            Class.forName(JDBC_DRIVER);
            conn = DriverManager.getConnection(DB_URL,USER,PASS);

            stmt = conn.createStatement();
            /* Perform a select on data in the classpath storage plugin. */
            String sql = "select name, format,tmst,date_time FROM hdfs.`hdfs.json` ";
            ResultSet rs = stmt.executeQuery(sql);
            System.out.println("Name" + " " + "FORMAT " + "TMST " + " " + "Date Time");
            System.out.println("-----");
            while(rs.next()) {
                int name = rs.getInt("name");
                String format = rs.getString("format");
                String tmst = rs.getString("tmst");
                String date_time = rs.getString("date_time");
                System.out.println(name+ " : " + format+ " : " + tmst + " : " + date_time);
            }

            rs.close();
            stmt.close();
            conn.close();
        } catch(SQLException se) {
            //Handle errors for JDBC
            se.printStackTrace();
        } catch(Exception e) {
            //Handle errors for Class.forName
            e.printStackTrace();
        } finally {
            try{
                if(stmt!=null)
                    stmt.close();
            } catch(SQLException se2) {
            }
            try {
                if(conn!=null)
                    conn.close();
            } catch(SQLException se) {
                se.printStackTrace();
            }
        }
    }
}

```

Figure 10 - Connection avec apache Drill

3.1.6 Connection OpenBouquet

Cette figure représente d'établir une nouvelle connexion avec apache Drill pour la visualisation. C'est la configuration de l'open bouquet, on a utilisé l'URL que nous avons créé pour déterminer la configuration de l'open bouquet et relie celle-ci par Drill.

The screenshot shows a web browser window with the URL `openbouquet.io/apps/release/ob1/index.html?apiUrl=http://localhost:8080#navigator/dictionaries/5835bd276087a3220df6106a`. The main content area displays the 'OB1 Navigator' interface. A modal dialog titled 'New datasource' is open, showing the configuration for a new data source. The configuration includes:

- Datasource connection configuration:** A dropdown menu set to 'Apache Drill'.
- Hostname:** A text input field containing 'localhost'.
- Port:** A text input field containing '31010'.
- Database:** A text input field containing 'hdfs'.
- Url:** A text input field containing 'jdbc:drill:drillbit=localhost:31010', which is highlighted with a red border.
- Username:** A text input field containing 'admin'.
- Password:** A text input field containing masked characters (dots).

At the bottom right of the dialog, there are two buttons: 'Cancel' and 'Check Connection'.

Figure 11 - Configuration open bouquet

3.1.7 Visualisation Open bouquet

Suite d'établir la connexion et choix du base de donner, on a essayé de visualisation de donner en fonction de courbe, pour garantir la visualisation il faut choisir obligatoirement 2 pivot et une opération de visualisation (svg, sum...)

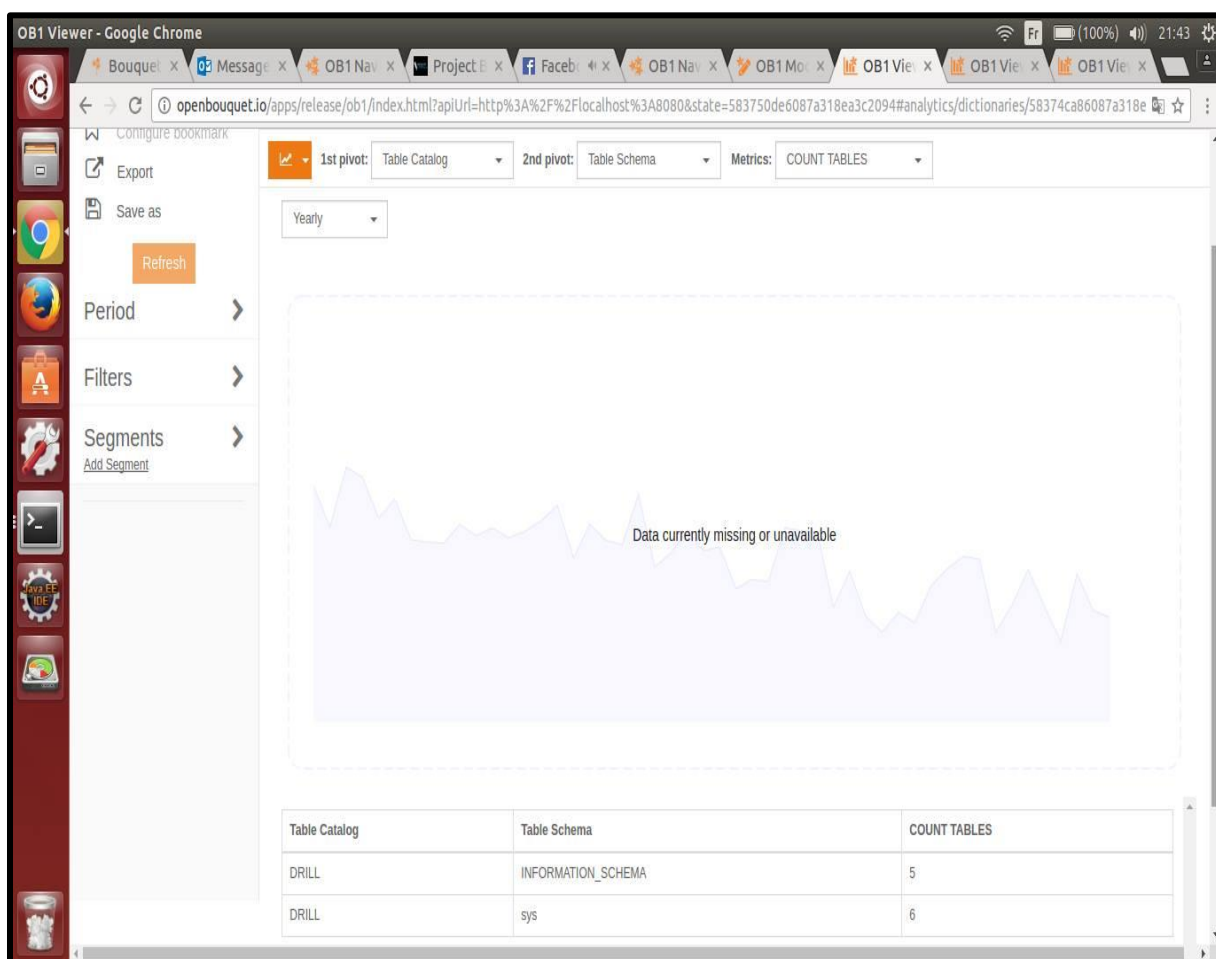


Figure 12 - Visualisation open bouquet

Conclusion

Ce chapitre nous a donné une idée sur l'ensemble des logiciels et matériels qui ont été utilisés dans le développement d'une application qui réalise une migration entre plusieurs technologies.

Nous avons clôturé ce chapitre par la présentation des différentes interfaces et fonctionnalités du notre application.

Conclusion Générale et Perspective

Plusieurs projets de recherche en cours favorisent le développement Stratégies et approches réalisables basées sur des technologies sémantiques et Paradigmes validant les avantages des ontologies pour améliorer l'interopérabilité dans l'Internet des choses. Dans la section précédente, une brève liste est fournie, Mais il manque beaucoup parce que la liste complète est assez grande. Tout ce qui souligne L'intérêt du milieu universitaire et des secteurs industriels dans de telles approches.

L'intérêt pour les utilisateurs de participer et d'inclure leurs propres capteurs Seront les avantages des outils et des applications qui sont en collaboration Développé, en conjonction avec les solutions que la communauté définit. Les outils seront principalement les outils de dataBig Data pour l'analyse des données, Les outils de planification pour la construction de la logique dynamique, les outils de visualisation pour la plates-formes mobiles et plates-formes de stockage de données.

La multiplication des données, dans un volume toujours plus important, et leur traitement, les problématiques, les nouvelles possibilités et les usages qui en découlent sont couverts par l'expression « Big Data ».

Grâce à Hadoop, même des structures limitées en taille et ressources, peut facilement avoir accès à des fortes capacités de calcul ,des déploiements à bas coût de clusters en interne ou location de temps d'exécution via les services de cloud computing.

Enfin, l'intérêt de l'utilisateur pour l'atteinte des solutions plus puissantes, Une meilleure analyse des données, une modélisation plus précise des données, une prise de conscience de la situation, Et de meilleures solutions définitives sera grâce à la collaboration et à la L'intégration de multiples sources de données, il peut être. Par exemple, concernant des scénarios tels que Villes intelligentes et l'automatisation des bâtiments.

DEPARTEMENT INFORMATIQUE OU TELECOMMUNICATION

STOCKAGE ET VISUALISATION DES TRAMES

KTATA ABDERRAHIM ET JMAL DOHA

RESUME :

Nous avons établi un programme java pour kafka pour qu'il arrive à envoyer des trames à Hadoop HDFS et qui seront enregistrées dans ce dernier dans un fichier. Ensuite, nous configurons apache Drill pour que puisse développer Hadoop HDFS et récupérer les données en les mettant dans un tableau, ceci sera par une facilité d'utilisation à travers les requêtes SQL. Nous terminerons donc par la configuration de l'openbouquet pour arriver à la visualisation.

ABSTRACT:

We have established a java program for kafka so that it can send frames to Hadoop HDFS and that will be saved in the latter in a file. Then we configure apache drill so that can develop Hadoop HDFS and retrieve the data by putting them in an array, this will be by ease of use through SQL queries. We will therefore end with the configuration of the openbouquet to arrive at the visualization

MOTS CLÉS:HADOOP,KAFKA,DRILL,OPENBOUQUET