MA-004.253.1

République Algérienne Démocratique et Populaire

**Ministère de l'Enseignement Supérieur et de la Recherche Scientifique**

Université Saad Dahlab De Blida

Faculté des Sciences

Département d'Informatique

مكتبة الكلية
كلية العلوم

Mémoire présenté par :

**Mr Djamal BELMEBROUK**

**Mr Sidahmed BENKHAOUA**

En vue d'obtenir un diplôme de Master en Informatique

Option : Génie des Systèmes Informatiques

**Sujet :**

# *Conception et mise en place d'un système informatique pour la gestion dynamique du trafic urbain en utilisant les technologies de IoT, Big Data et l'analyse en temps réel*

**Promotrice :** Mme Zahra Fatma Zohra **Encadreur :** Mr BOUSSEBAT Khaled

**Organisme d'accueil :** Icon Software

ICON Software

Soutenue le :

Devant le jury :

- Mr: M. BALA présedant
- Mr: Kamache Examinateur
- M: M. Sidamou,
- Promoteur, Zahra. Fatma Zahra.

***- Session Juin 2015 -***

MA-004-253-1

## *Remerciements*

*Avant de présenter ce travail, nous tenons à remercier Dieu le tout puissant, de nous avoir permis d'arriver à ce niveau d'étude et aussi pour nous avoir donné la force, le courage et la patience pour accomplir ce travail.* ***'Dieu merci'***

*Nous tenons à remercier nos chers parents pour leur encouragement et leur soutien moral et matériel dans le but d'assurer notre réussite.*

*Nous remercions notre promotrice Mme Zahra Fatma Zohra, pour ses précieux conseils. Nous la remercions de nous avoir incités sans cesse à améliorer la qualité de notre travail en gardant à l'esprit l'objectif académique de celui-ci.*

*Nous tenons à remercier également notre encadreur de projet Mr Khaled BOUSSEBAT pour son accueil chez Icon softeware® et de nous avoir proposé ce sujet et pour sa disponibilité, ses conseils et d'avoir partagé avec nous tous ces connaissance et son expérience.*

*Nous remercions aussi tout l'équipe de icon software pour leur disponibilité et Aidez-nous pour quel que soit petit ou grand.*

*Nous tenons à remercier aussi, les membres du jury, Pour l'honneur qu'ils nous ont accordé en acceptant de juger cet humble travail.*

*Enfin Merci à tous ceux qui ont contribués de près ou de loin dans l'élaboration de ce travail.*

*Je dédie ce modeste travail :*

*A mes chers parents qui ont tout fait pour que je réussie dans ma vie et mes études*

*A mes chers frères surtout Ahmed*

*A ma chère sœur*

*A toute ma famille sans exception*

*A mon binôme Sidahmed*

*A tous mes amis en particulier Lahcen, Abderrahim, Omar, Oussama, Rida, Youcef, ….*

*A mes chers amis de la résidence universitaires*

Merci Djamal.

*Je dédie ce mémoire*

*A Ceux, qui sans leur aide, je ne serai pas là, vous qui m'avez appris à continuer de marcher vers la lumière nonobstant les aléas de la vie courante, vous dont le cœur chaleureux à toujours été le seul refuge, pour votre indulgence, clémence et bienveillance, vous dont l'inquiétude exagérée ne décelait que de l'amour, vous les personnes les plus dévouées…, Vous mes chers parents.*

** A Mon frère Boualem et mes sœurs Meriem Nora et Hanane.*

** A Mes chères Nièce Hibat El Rahmane Nourhane et Maria Aïcha.*

** A Ma grand-mère maternelle que Dieu miséricordieux la protège.*
*A mon binôme Djamal*

** A tous Mes amis et camarades qui me sont chers, à tous ceux que j'aime et m'aime: qu'ils trouvent ici l'expression de mes sentiments les plus dévoués et mes vœux les plus sincères.*

** Sans oublier toutes les personnes que j'ai croisées sur le chemin de ma vie, entre autres tous mes professeurs, aux prés desquels, j'ai acquis toutes sortes de connaissances et d'expériences, qui font de moi ce que je suis Aujourd'hui.*

** Que dieux le tout puissant vous préserve tous et vous procure sagesse et bonheur.*

Merci Sid Ahmed.

## ملخص

الازدحام المروري هو حقيقة وأن أي سائق قد واجهته فيمالا يقل عن لحظة من حياته. هذا يؤثر على الملايين من الناس في جميع أنحاء العالم وينجم عنه آثار جانبية على جميع المستويات: الشخصية والمهنية والاجتماعية. لهذه الأسباب ومع تقدم التكنولوجيا بما في ذلك تقنيات الحوسبة والاتصالات ومعالجة البيانات، تسعى شركة أيقونة للبرمجيات لإنشاء نظام لتحليل حركة المرور في المدن.

في هذه المذكرة سوف نقترح بنية باستخدام مختلف التقنيات التحليلية ومعالجة البيانات لإنشاء نظام مراقبة وتسهيل حركة المرور في المدن.



**كلمات البحث:** حركة المرور، CEP، NoSQL، Big Data، IoT،معالجة في الوقت الحقيقي



## Résumé

La congestion routière est une réalité que chaque conducteur doit au moins affronter à un instant de sa vie. Cela affecte des millions d'individus à travers le monde et génère des effets indésirables à tous niveaux : personnels, professionnels et sociaux. Pour ces raisons et grâce à l'avancement des technologies, notamment en informatique, communication et techniques de traitement des données, l'entreprise Icon Software souhaite mise en place un système pour analyser le trafic urbain.

Dans ce travail nous avons proposé une architecture utilisant les différentes technologies d'analyse et traitement des masses de données pour mettre en place un système de suivi du trafic urbain.



**Mots clés :** Trafic routier, IoT, Big data, NoSQL, CEP, analyse temps réel



## Abstract

Traffic congestion is a reality that each driver must confront in at least a moment of his life. This affects millions of people worldwide and generates side effects at all levels: personal, professional and social. For these reasons and with the advancement of technologies, including computing, communication and data processing techniques, the company Icon Software wants to set up a system for analyzing urban traffic.

In this paper, we will propose an architecture using various analysis and data processing technologies masses to establish a urban traffic monitoring system.



**Keywords:** road traffic, IoT, Big data, NoSQL, CEP, real-time analysis

# Table des matières




جامعة البليدة 1 مكتبة الكلية كلية العلوم

**Liste des figures :**

## Liste des tableaux :

# INTRODUCTION GENERALE

## 1. Contexte et problématique

De nos jours, se déplacer est devenu un aspect essentiel de la vie quotidienne : qu'il s'agisse de transports en commun ou de véhicules personnels, le vaste réseau formé de ces moyens de locomotion est immensément complexe à gérer. Le réseau routier a été développé pour répondre aux demandes de déplacements.

Dans le monde et notre pays en particulière, et durant les deux dernières décennies, l'augmentation croissante du nombre de véhicules en circulation, qu'ils soient particuliers, de transport en commun, ou de transport de marchandises, couplée à la modification de la carte de mobilité des citoyens en zone urbaine, a conduit à une congestion de la circulation aussi insupportable humainement qu'handicapante socio-économiquement.

Pour comprendre la congestion, il faut garder présent à l'esprit que c'est un phénomène qui survient lorsque la demande (le nombre de véhicules qui cherchent à utiliser une infrastructure donnée) est supérieure à la capacité de cette infrastructure. Si la demande excède la capacité, alors des véhicules seront ralentis à l'entrée de l'infrastructure, formant ainsi un bouchon. Ces véhicules excédentaires seront à chaque instant plus nombreux qu'à l'instant précédent. Comme chaque véhicule occupe une certaine longueur de voie, la longueur de la file d'attente ne fera que croître en proportion du nombre de véhicules présents dans cette file d'attente. Donc la congestion est un phénomène évolutif, à la fois dans le temps et dans l'espace.

Parmi les conséquences négatives de ce phénomène, en plus de la pollution, il y a les retards de livraison pour les marchandises et, plus sérieux encore, le retard et le stress des usagers du réseau routier passant une grande partie de leur temps dans des embouteillages interminables ce qui montre l'ampleur de l'impact de nos déplacements quotidiens qu'il devient de plus en plus nécessaire d'administrer et gérer. D'où constat de gestion trafic routier s'impose.

Grâce à l'avancement des technologies, notamment en informatique, communication et techniques de traitement des données, il est devenu possible de développer un système pour détecter les perturbations, mesurer les effets et même d'anticiper l'état du trafic afin de mieux adapter les actions d'exploitation.

La gestion de trafic routier considère deux niveaux :

- la surveillance des réseaux et l'information routière en se basant sur le déploiement de systèmes de supervision du trafic, centrés sur le recueil d'information sur les conditions de circulation, les événements, et leur diffusion aux gestionnaires et/ou aux usagers.
- la régulation dynamique de trafic basé sur le traitement des informations collecté en temps réel pour maximiser l'écoulement et d'optimiser l'usage de la voirie.

## 2. Objectifs

L'objectif de ce travail tel qu'il était proposé par la société **Icon Software®** est la mise en place d'un système de transport intégré (logiciel et matériel) pour la gestion dynamique du trafic urbain à travers le contrôle par panneaux de signalisation et/ou la notification des citoyens soit par des composants électroniques installés sur les véhicules.

Le système souhaité est décomposé en deux partie, la première consiste à développer un applicatif mobile et des composants électronique pour la capture des données sur les routes, tendis que la deuxième partie consiste à mettre en place une infrastructure de Big Data (vélocité et volume important de données) et de CEP « Complex Event Processing » (Manipulation d'événements) pour le stockage et traitement des données capté et par la suite des actions de régulation dynamique de trafic urbain.

On suppose que la première partie elle est déjà implémentée, l'objectif est de :

- Etude sur l'état de l'art des technologies de IoT, Big Data, CEP et l'analyse en temps réel,
- Spécification d'un schéma de données dynamique NoSQL orientée colonne pour le stockage de données collectées,
- Utilisation de CEP pour la mise en place des patterns et règles d'optimisation pour suivi du trafic routier,
- Réalisation d'un prototype applicatif.

### 3. Organisation du mémoire

Les chapitres de ce mémoire s'organisent autour cinq chapitres selon une structure logique en débutant dans chapitre 1 par une étude sur l'internet of things de fait que la connaissance de l'état de trafics est un élément essentiel en temps réel pour l'exploitation du réseau routier et notamment pour alimenter les systèmes de la gestion du trafic comme en temps différé pour disposer de statistiques sur l'état de trafic , la connaissance est faite par des composants électroniques de supervision et collecte de données qui sont intégrés dans IoT.

Le deuxième chapitre se focalise sur les technologies de big data notamment en terme de stockage et traitement pour l'exploitation et l'analyse des masse de données générer par les différents composants électroniques implanté sur l'infrastructure routier, Le chapitre expose une étude de technologies NoSQL qui représentent une révolution technologique pour le stockage et certains plateforme d'analyse et traitement.

Le troisième chapitre présent l'aspect de manipulation d'événements, CEPqui représente un moyen de supervision et détection efficace, dans ce chapitre nous présentons les différents terminologies et concepts de CEP qui permet de répondre a la contrainte de temps réel.

Le quatrième chapitre est dédié à l'étude conceptuelle de l'application, il concerne le premier apport de notre travail, notre solution selon une démarche conceptuelle, ce chapitre contient en plus d'étude conceptuel, la modélisation d'événements et la solution proposée à résoudre la problématique de routage de client.

Le cinquième chapitre constitue une présentation des aspects implémentation et réalisation de notre travail.

Enfin, nous terminerons par une conclusion générale et des perspectives de travaux futurs seront présentées.

# I. CHAPITRE I : ETAT DE L'ART : INTERNET OF THINGS

## 1. Introduction

Les technologies les plus profondément enracinées sont les technologies invisibles. Elles s'intègrent dans la trame de la vie quotidienne jusqu'à ne plus pouvoir en être distinguées [1]. L'informatique est en train de transformer notre rapport à l'environnement parce qu'elle s'intervenir dans les objets du quotidien a l'image du smartphone, et sans que on se limite aux outils électroniques, les éléments de notre environnement deviennent "smart", capables de calculs, de l'exécution des applications, de stockage de données et d'accès au réseau sans fil ce qui permet d'utiliser leurs capacités pour inventer des usages innovants. Dans cette partie, nous attacherons à décrire les différents concepts sur lesquels est bâti l'Internet of things (IoT).

## 2. Définitions

L'Internet Of Things (Iot) est une révolution technologique dans le domaine de l'informatique et des télécommunications [2], il s'agit des milliards d'objets varié intelligents et connectés, qui communiquent entre eux en permanence ou à des intervalles réguliers, pour transmettre à des systèmes d'information distants un état, un statut, un besoin, une géolocalisation, etc. L'objective de l'Internet des objets est de permet les objets d'être connecté a n'importe quel moments et places avec n'importe quoi et n'importe qui pour créer ce que on appelle smart environments rendre le transport, énergie et d'autre domaines plus intelligent [3].

l'internet of things est vue comme une infrastructure de réseau dynamique avec les capacités d'auto-configuration basée sur des protocoles standard et interopérables communication où les objets physiques et virtuels ont des identités, des attributs physiques, et des personnalités virtuelles et utilisent des interfaces intelligentes, et sont intégrés de façon transparente dans le réseau d'information,c'est la définition donné par IERC[1][3].

L'IoT peut être vu sous deux angles, soit centré sur l'Internet ou centrer sur l'objet. Quand elle est centrée sur l'Internet, les services sont la composante principale de son architecture et les objets contribuent en l'alimentant par des données. Lorsqu'elle est centrée sur l'objet, le centre de l'architecture devient l'objet et on parle de cloud des objets. Le cloud des objets apparaît donc comme une plateforme d'objets permettant un usage intelligent des infrastructures, des applications et de l'information à coûts réduits [4].

[1] European Research Cluster on the Internet of Things

Internet ne se résume plus à un réseau informatique uniquement accédé et commandé par des êtres humains, les objets peuvent aussi communiquer maintenant entre eux grâce à des capteurs intégrés, s'échanger des informations et agir automatiquement après avoir analysé les informations reçues comme le montre la figure I.1

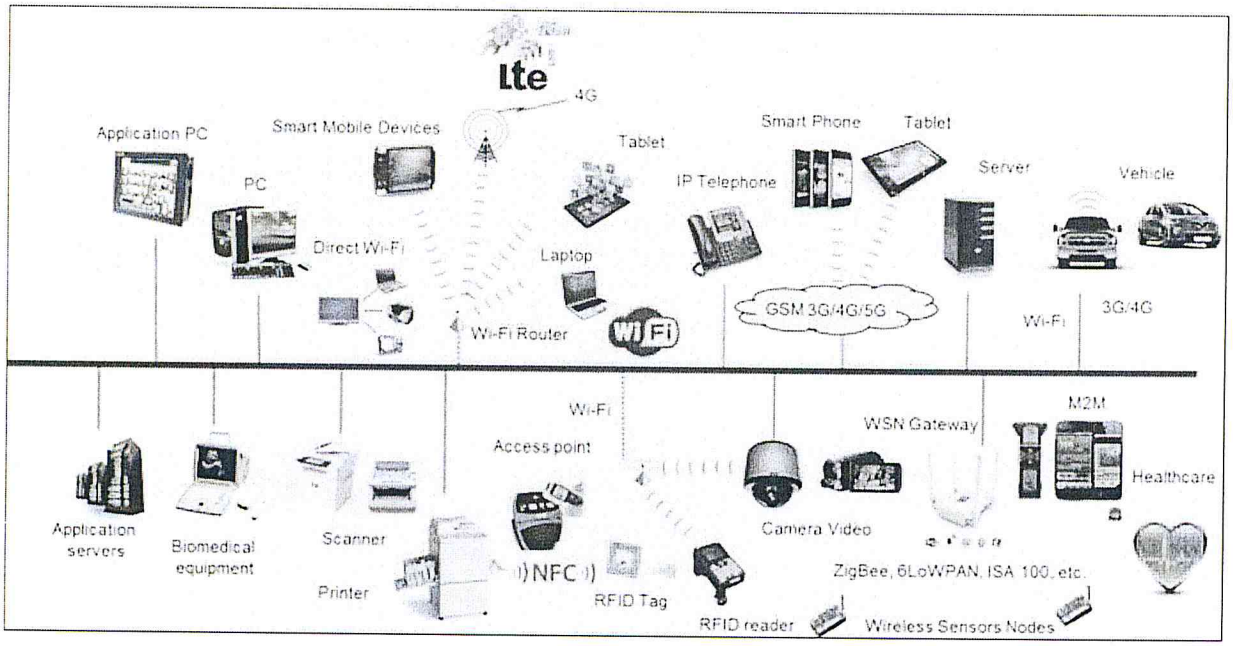



**Figure I-1:** Schéma illustratif de l'internet des objets.

### 3. Les éléments de l'IoT

Trois éléments essentiels sont requise pour la mise en ouvre de l'internet of things [5]:

- Hardware : généralement des capteurs, actionneurs et hardware embarqué communicants.
- Middleware : pour le stockage et outil de calcul et analyse les données.
- Présentation : outil de visualisation et interprétation d'objets lui permettant d'être accessible dans différents plateformes.

### 4. L'architecture de l'internet of things

L'architecture générale de l'IoT se constitue en trois couche : la couche perception, la couche réseau pour la transmission de données et la couche application, comme c'est résumé dans la figure 1.2.

La couche de perception est la source de collection des informations des objets, elle est composé généralement de variété de capteurs.la couche de transmission fait référence a utilisation de réseaux sans fil et de réseaux classiques pour permettre l'échange de données .la

couche application c'est l'interface entre l'internet of things et les utilisateurs.la section suivante détail mieux ces différents couches.

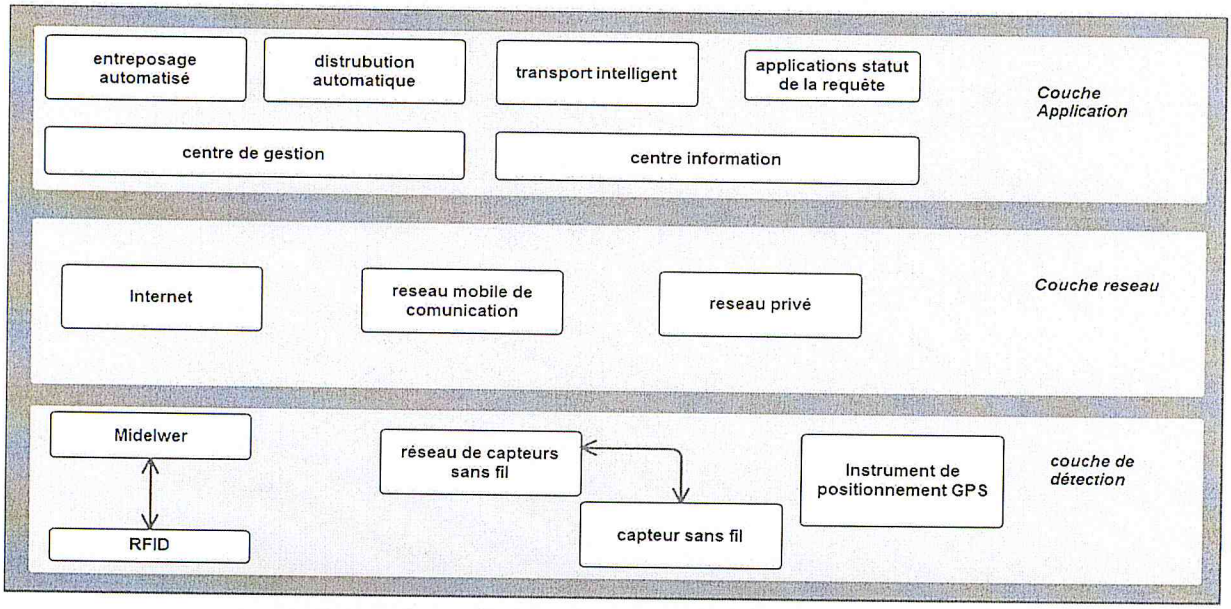



**Figure I-2:** L'architecture générale de l'IoT[6].

## 5. Les technologies de l'IoT

Les domaines technologiques couverts par l'IoT sont larges et variés, dans ce qui suit nous présentons ces différents concepts et technologies de l'IoT :

### 5.1. Les capteurs et Les réseaux de capteurs sans fil

Les capteurs sont généralement les composants essentiels de l'IoT, ils sont de petits appareils disposant de capacités de mesures, voire d'actions, sur leur environnement,par exemple la température, le taux d'humidité, la détection de présences ou de mouvements...etc, équipés de dispositifs sans fil pour l'émission et la réception de données. tandis que Le réseau de capteurs sans fil c'est l'ensemble des capteurs qui s'organise dans un but précise. Dans de nombreux cas, les capteurs sont constitués des unités suivantes:

- Unité d'acquisition, permet le recueil de données environnementales et la conversion analogique vers numérique.
- Unité de calcul, permettant le lancement de procédures, protocoles et autres.
- Unité de communication, permettant la connexion au réseau (émission et réception).
- Unité d'énergie, qui permet la répartition de l'alimentation entre les différents constituants. Dans de nombreux cas, les capteurs sont dispersés dans des zones pauvres en énergie, et sont dotés d'une batterie non-rechargeable et non-renouvelable[7]. Dans le cas du trafic routier, il est néanmoins possible voire nécessaire de s'abstenir de contrainte énergétique.

Les informations collectées dans un réseau de capteurs convergent vers un puits de données (le sink) chargé de la collecte des mesures. Les capteurs doivent donc utiliser la structure de leur réseau sans fil afin d'atteindre ce puits, qui n'est peut être pas, pour une majorité d'entre eux, directement accessible en un saut.

### 5.2. Technologies d'interfaçage dans l'IoT

Plusieurs technologies sont utilisées pour faire communiquer un objet avec l'Internet, parmi lesquelles RFID, NFC, le protocole de communication Zigbee.

La *RFID* est constituée d'un couple lecteur/étiquette. Le lecteur envoie une onde radio, l'étiquette envoie à son tour une trame d'identification. Une fois la puce alimentée, l'étiquette et le tag communiquent suivant le protocole TTF (Tag Talk First) ou ITF (Interogator Talk First). La distance de lecture des puces RFID varie de quelques cm à quelques mètres et elle peut y aller parfois jusqu'à 200 m[8][9].

La technologie *NFC* est le résultat de plusieurs évolutions des microcontrôleurs, des cartes à puce, et des communications à courte portée[8]. NFC est basée sur le même principe que la RFID, c'est-à-dire l'identification par radio fréquence. Elle permet l'échange d'informations à courte distance entre deux objets (un lecteur et une carte) sans contact[5][10]

*Zigbee* quant a lui, c'est un protocole de communication sans fil à bas coût qui permet des échanges à courte distance entre les nœuds d'un réseau sans fil, il est basé sur la norme IEEE 802.15.4 qui spécifie les protocoles de communication entre les couches physiques et liaison de données du modèle OSI[5][11].

### 5.3. Réseaux de communication dans l'IoT

Un défi est créer avec l'avènement de l'IoT, est celui de la mise à disposition des réseaux de communication fiables, tant au niveau des infrastructures qu'au niveau des protocoles de communication, Ce défi est lié entre autres à la mobilité des objets, à l'hétérogénéité des données et des plateformes, à l'accès à l'information depuis n'importe quel lieu, à n'importe quel moment et à travers n'importe quel dispositif , deux approches a mentionné permettant de mettre en ouvre des réseaux adaptés à nouvelles contraintes de l'IoT (la mobilité, l'hétérogénéité, l'accès aux informations des objets et volume de données).

La premier approche est l'utilisation de réseaux sans fil courte portée (Zigbee, Wifi) qui permettent de connecter les objets à l'internet via une passerelle, tandis la deuxième réfère a l'utilisation des réseaux cellulaires classiques large bande (4G, 3G).

### 5.4. L'adressage des objets

La mise en réseau de des objets communicants hétérogènes impliquant l'identification, l'adressage et le nommage de ces objets, renverra nécessairement à l'époque ou l'adressage IP a été mise en place pour permettre l'interconnexion d'un très grand nombre de réseaux hétérogènes. Cependant la technologie IP est relativement gourmande en termes d'utilisation de ressources alors que celles-ci sont rares dans la plupart des objets communicants.

Le groupe 6LoWPAN a défini les mécanismes d'encapsulation et de compression d'entêtes permettant aux paquets IPv6 d'être envoyés ou reçus via le protocole de communication IEEE 802.15.4.[12]

L'adressage IPv6 est le seul qui pourrait satisfait en terme de puissanced'adressage des milliards d'objets communicants, a noté aussi que la technologie IPv6 est de plus en plus connue, se répand et n'entraîne pas de rupture importantepar rapport à l'existant comme internet [13].

### 5.5. Technologies de localisation et géolocalisation dans l'IoT

Il existe trois grandes classes des systèmes de géolocalisation : intérieur, extérieur et mixte. Dans ce qui suit, on cite quelques exemples des systèmes géolocalisation.

#### *5.5.1 Le système GPS*

La puce *GPS* (système de localisation mondial) est le système de repérage le plus utilisé dans le monde. En effet, elle s'intègre facilement dans les dispositifs mobiles, et permet de transmettre la position du mobile en temps réel aux applications dans divers domaines(le transport, les services d'urgence, la météo), le GPS est un système de positionnement extérieur parce que il est destinés à l'utilisation dans les surfaces dégagées.

#### *5.5.2 Techniques de localisation*

Parmi les techniques de localisation nous citons deux principaux algorithmes: La Triangulation et la Trilatération. La trilatération est basée sur La distance entre le mobile et la station de base, la position du mobile est déterminée à partir des distances estimées depuis trois stations de bases distinctes minimum.

La triangulation utilise quant à elle la direction du signal provenant du mobile. La localisation dans ce cas se fait en interprétant les angles que fait la direction du signal et les antennes des stations de base.

### 5.5.3 *Active Badge*

Le système *Active Badge* est un système de géolocalisation en intérieur. Basé sur une architecture propre, il utilise les signaux infrarouges. Dans ce système, un transmetteur infrarouge est placé sur chaque terminal et émet toutes les 15 secondes un signal en diffusion contenant un identifiant unique[14]. Grâce au déploiement d'un réseau de capteurs dans le bâtiment, il est possible d'intercepter les signaux provenant des terminaux. Lorsqu'on identifie les capteurs qui perçoivent un signal particulier, on est en mesure de déterminer la position du terminal qui émet ce signal. Ce système n'est cependant pas très précis.

### 5.5.4 *Radar*

Le système *RADAR* utilise les réseaux sans fil de type 802.11. Donc un radar utilise les ondes électromagnétiques pour détecter la présence et déterminer la position ainsi que la vitesse d'objets. Les ondes envoyées par l'émetteur sont réfléchies par la cible, et les signaux de retour sont captés et analysés par le récepteur, souvent situé au même endroit que l'émetteur. La distance est obtenue grâce au temps aller/retour du signal, la direction grâce à la position angulaire de l'antenne où le signal de retour a été capté et la vitesse avec le décalage de fréquence du signal de retour généré selon l'effet doppler. Les systèmes radar sont des systèmes de positionnements mixtes.

Le Radar de contrôle routier est l'un d'exemples du système radar, le radar de contrôle routier servant notamment à mesurer la vitesse des véhicules circulant sur la route.

Le schéma suivant représente un récapitulatif de classification et des spécificités des systèmes de géolocalisation :

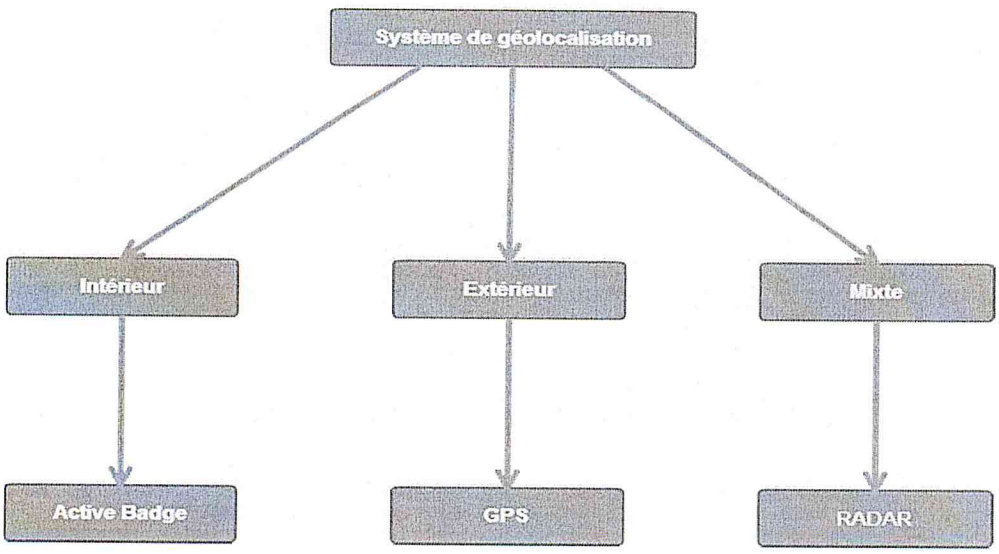



**Figure I-3:** classification et des spécificités des systèmes de géolocalisation

### 6. Domaines d'application de l'IoT

L'objectif majeur de l'IoT est la création des environnements intelligents, nous nous allonsprésenter les domaines d'applications de l'IoT qui serons entouré généralement autour des concept figuré dans la figure 1.5 en vue que il est impossible d'envisager toutes les domainesd'applications.

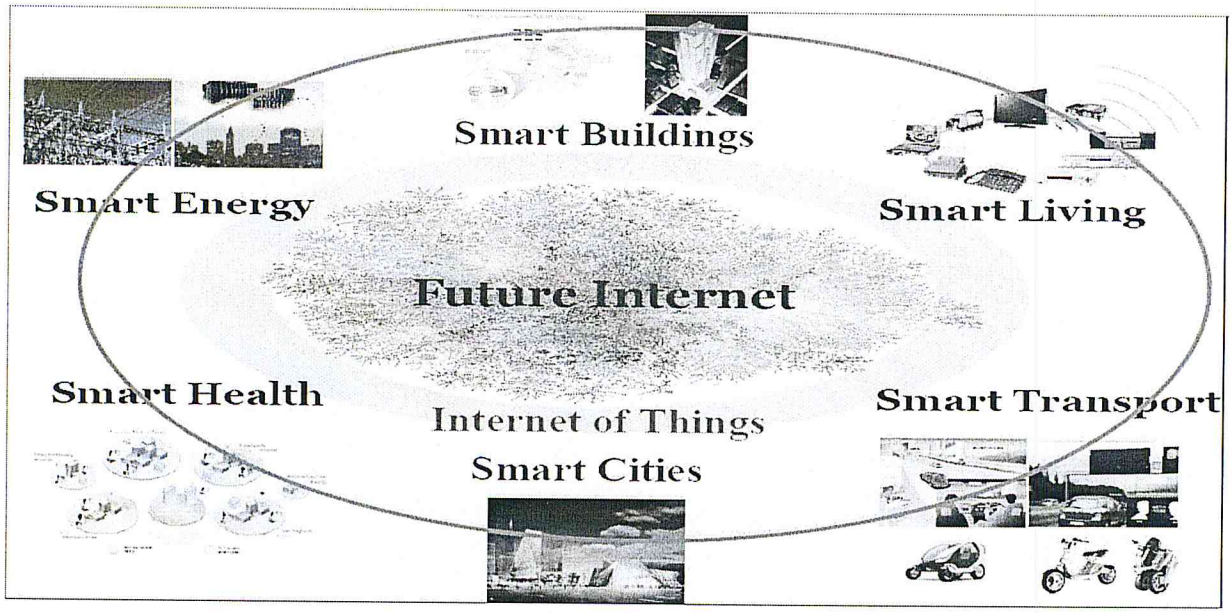



**Figure I-4: les domaines d'application de l'Internet of things[3]**

#### 6.1. Les villes intelligentes

La ville intelligente représente un champ d'application essentiel pour l'Internet of things. Ainsi l'IoT jouer un rôle essentiel dans la maîtrise de circulation automobile dans les villes, contrôler la pollution dans les villes, diminuer la bruit dans les zones urbains central, l'éclairage civil et d'imposé au citoyen une multitude de technologie lui permettant d'interagir et mieux exploiter le différent service public et amélioré son production.

#### 6.2. Les immeubles intelligents

Le déploiement de différents composants électronique qui peuvent être communiqué et interconnecter (TV, téléphone, réfrigérateur, etc.), des capteurs et des actionneurs donnent un véritable sens à la domotique, avec des objectifs de confort, de sécurité, d'économie d'énergie et de partage des contenus.

#### 6.3. Le transport intelligent

Avec l'IoT, des routes intelligents, surveillé et contrôlé en temps réel et messages de notification sont mise a disposition d'usager permettant de la mobilité, de faire disparaitre les

congestions et bouchons de circulation et d'éviter les accidents, de manière générale un transport ou un mouvement facile et sécurisé.

### 6.4. Réseau énergétique intelligent

La maîtrise énergétique est un point stratégique pour les citoyens, les différents objets communicants permettront d'adapter et ajuster le transport et la fourniture d'énergie à la consommation des usagers en particulier dans l'électricité et gaz.

La maîtrise énergétique est un but stratégique no seulement pour les citoyens mais aussi pour les payes, les technologies d'IoT permettront une optimisation de la consommation, un stockage efficace et une meilleure intégration des différentes sources d'énergie y compris l'intégration de la production d'énergies renouvelables.

### 6.5. La santé

Les objets connectés sont en train de devenir des auxiliaires irremplaçables dans le domaine de la prévention, de la communication entre le patient et le soignant, le soin et la surveillance des malades chroniques.

L'IoT représente une plateforme convenable pour e santé, par exemple un corps humain équipé de plusieurs capteurs de mesure de l'état physiologique, ces mesure envoyer via connexion court porté comme Bluetooth a un smartphone qui a son tour envoi ces mesures a un central médical pour le traitement et donne des renseignements aujourd'hui énormément d'application mobile sont disponible. Avec IoT le patient n'a pas besoin de se déplacé, faire un rendez-vous...etc.

### 6.6. L'industrie

Avec l'IoT, les machines devenirons indépendants et capable d'auto-configuré et d'auto-diagnostiqué et de communiqué entre eux, les capteurs vont mesurer la température, détecté des fuites, etc et les actionneurs vont réagir, etc.

l'application de l'IoT dans le domaine industriel permet de :

- ✓ sécurisé les environnements industriels.
- ✓ développement des nouvelles infrastructures de production.
- ✓ réduire la consommation énergétique et perd des produits.

## 7. Conclusion

Dans ce chapitre, Nous avons montrél'Internet des objets qui est une technologie vaste et couvre des domaines variés. Elle peutintégrer divers composants, des protocoles de communication, des middlewares, des données et des services de nature différente. Les données collecté et échangées dans l'IoT entre capteurs,actionneurs, serveurs, et l'Internet proviennent de sources diverses, sont fortement hétérogèneset représentent une masse de plus en plus volumineuse. L'exploitation intelligente de ces données représente un aspect intéressant pour des fins d'analyse. Ce qui sera abordé dans le chapitre suivant.

# II. CHAPITRE II : LES TECHNOLOGIES DU BIG DATA

## 1. Introduction

Avec la montée en puissance de l'Internet des objets, les capteurs embarqués sur des dispositifs mobiles permettent de remonter de plus en plus de données (température, pression, position GPS, vitesse, luminosité, rythme cardiaque, etc.). Cette situation complique de plus en plus la tâche des analystes, les technologies de big data se trouvent face au problème de proposer de nouvelles solutions adaptées à cette forte volumétrie en constante progression.

## 2. Big Data : Définition

C'est l'exploration de très vastes ensembles de données pour obtenir des renseignements utilisables, Le terme Big Data désigne une technologie permet de traiter un volume très important de données aussi bien structurées que non structurées, se trouvant sur des terminaux variés (PC, Smartphones, tablettes, objets communicants...), produites en temps réel(généralement) depuis n'importe quelle zone géographique dans le monde.

Le tableau suivant montrer une comparaison entre les masse de données et les données traditionnel

| | les données traditionnelles | les données de masse |
|---|---|---|
| Volume | Gigaoctet | Pétaoctet/Exaoctet |
| taux de génération | par heur/jour | rapidement |
| Structure | données structuré | semi et non structuré |
| Source | Centralisé | distribuer |
| Intégration | Facile | difficile |
| Stockage | RDBMS | HDFS, NoSQL |

**Tableau II-1: comparaison entre les masses de données et les données traditionnel**

## 3. Big data et 3V

Les grandes caractéristiques qui englobent les problématiques auxquelles le Big Data répond sont le Volume des données, la Vitesse d'acquisition et de traitement des données et la Variété des types de données, c'est pour quoi on parle des trois « V» du Big Data.

- **Volume :** Le volume de données numériques augmente de manière exponentielle, selon IBM 90% de l'ensemble des données aujourd'hui disponibles ont été créées ces deux dernières années [15]. Ces données dépassent les limites de la scalabilité des outilsclassiques, nécessitant des solutions de stockage distribués et des outils de traitement parallèles.

- **Variété :** les données analysées ne sont plus forcément structurées comme dans les analyses antérieures, mais peuvent être du texte, des images, du contenu multimédia, des traces numériques, des objets connectés, etc.[15]
- **Vitesse :** La vitesse décrit la fréquence à laquelle les données sont générées, capturées et analysé. Les données ne sont plus traitées en temps différé, mais en temps réel (ou quasi réel) et par fois il est même possible de ne plus stocker les informations, mais de les analyser en flux (streaming) [15].

Ces trois caractéristiques ou ces trois « V» sont les piliers définissant le Big Data et nombreux responsables informatiques et chercheurs du domaine définir le Big Data selon ces trois V.

L'univers technologique du Big Data s'appuie sur des outils et technologies bien identifiés qui constituent la base innovante de ce mode de traitement.ces outils résument le vocabulaire technologique du Big Data. Ces technologies seront abordées dans ce qui suit.

### 4. Big Data et le traitement massivement parallèle (MPP)

Les ordinateurs parallèles sont des machines qui comportent une architecture parallèle, constituée de plusieurs processeurs identiques concourant tous au traitement d'une seule application. Plusieurs types d'architectures caractérisent MPP: les architectures SVM, MIMD et SIMD, les architectures les deux premier sont les plus représentatives de calcule ou traitement massivement parallèle.

Big Data repose sur des systèmes de traitement de données organisés sur plusieurs nœuds ou clusters parallèles[16]. Ce qui permet de générer des calculs à très haute performance sur des infrastructures (processeurs, serveurs) réduites.

### 5. Traitement IN-MEMORY

On parle de traitement in-memory pour évoquer les traitements qui sont effectués dans la mémoire vive RAM de l'équipement informatique, plutôt que sur des serveurs externes ou les analysé après avoir stocker sur les disques.

L'avantage du traitement in-memory est celui de la vitesse et la rapidité des traitements puisque les données sont immédiatement accessibles. En revanche, ces données ne sont pas stockées sur le long terme, ce qui peut poser des problèmes d'historisation[17].

## 6. Les Bases de données NoSQL

Les bases de données relationnelles ont été la solution la plus adaptée pour stocker des données. elles ne nécessitent que de faire un effort de structuration et modélisation des données pour pouvoir représenter un logiques métier en modèles manipulables dans une base de donnée relationnelle.

### 6.1. De propriétéACID au BASE

Les bases de données relationnelles ne permettent plus de répondre aux besoins de stockage et de traitement de grandes quantités de données, C'est dans ce contexte les technologies NoSQL apparaitre. Le terme NoSQL ne s'agit pas d'un nouveau langage de requête permettant de dialoguer avec un SGBD mais plutôt d'une nouvelle approche du stockage de données.On regroupe derrière le NoSQL l'ensemble des technologies de qui se distinguent par un relâchement des caractéristiques ACID propres aux bases de données relationnelles qui sont[18]:

- Atomicité : l'ensemble des opérations d'une transaction sont réalisées ou toutes sont abandonnées pour revenir à l'état initial.
- Consistance : la base de données conserve un état consistant avant comme après l'exécution d'une transaction.
- Isolation : les autres opérations ne peuvent pas voir l'état intermédiaire des données modifies par une transaction tant que celle-ci n'est pas terminée.
- Durabilité : assure qu'une fois que l'utilisateur a été notifie de la conclusion correcte d'une transaction, l'ensemble des modifications est persistante et ne sera pas annulé.

L'univers du NoSQL introduit l'acronyme BASE correspondant aux propriétés suivantes :

- Basically Available : essentiellement disponible.
- Soft state : état variable dans le temps.
- Eventually consistant : éventuellement consistant, une donnée répliquée peut, à un instant donné, ne pas avoir la même version sur les différents nœuds.

Le théorème CAP[19]permet de faire le lien entre ces deux notions. Ce théorème est plus adapté aux systèmes distribués et qui dit qu'aucun système distribué ne peut satisfaire au maximum que deux des trois caractéristiques suivantes en même temps

- **Cohérence**: les données sont identiques sur lesnœuds sur système au même moment,
- **Disponibilité** : accessibilité des données en cas de panne,
- **Toléranceà la partition**: les données peuvent être partitionnées.

Afin de créer une architecture distribuée on doit donc se résoudre à choisir deux de ces propriétés, laissant ainsi trois designs possibles :

- **CP :** Les données sont consistantes entre tous les nœuds et le système possède une tolérance aux pannes, mais il peut aussi subir des problèmes de disponibilité.
- **AP :** Le système répond de façon performante en plus d'être tolérant aux pannes. Cependant rien ne garanti la consistance des données entre les nœuds.
- **CA :** Les données sont consistantes entre tous les nœuds. Toutes les lectures/écritures des nœuds concernent les mêmes données. Mais si un problème de réseau apparait, certains nœuds seront désynchronisés au niveau des données (et perdront donc la consistance).

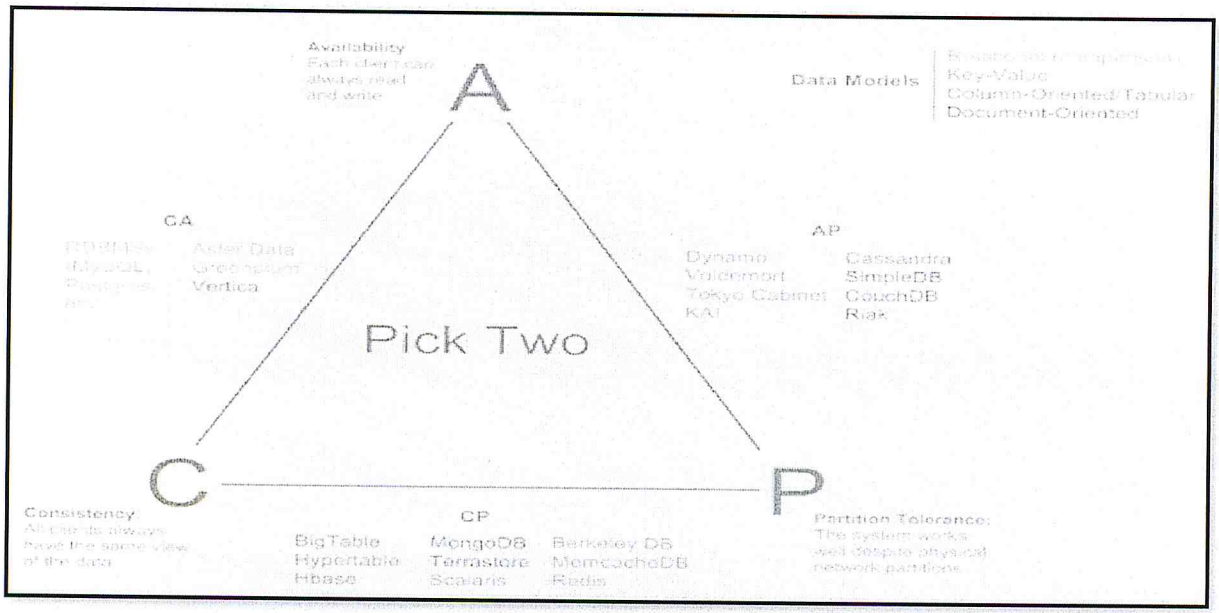



**Figure II-1: un guide visuel au NoSQL**

Les bases de données relationnelles classiques adhérent aux propriétés de Cohérence et Disponibilité(AC), tandis que les bases de données NoSQL sont généralement des systèmes CP (Cohérent et Résistant au partitionnement) ou AP (Disponible et Résistant au partitionnement).

### 6.2. Les catégories de SGBD NoSQL

Les systèmes NoSQL existantes peuvent être regroupés en 4 grandes familles[20] :

#### *6.2.1 Clé/valeur*

C'est le modèle le plus simple et basique[21], il peut être assimilé à une hashmap distribuée. Les données sont donc, simplement représentées par un couple clé/valeur. La valeur peut être une simple chaîne de caractères, un objet sérialisé, cette absence de structure ou de typage a un impact important sur le requêtage. En effet, toute l'intelligence portée

auparavant par les requêtes SQL devra être portée par l'applicatif qui interroge la BD. Cette structure est très adaptée à la gestion des caches ou pour fournir un accès rapide aux données.

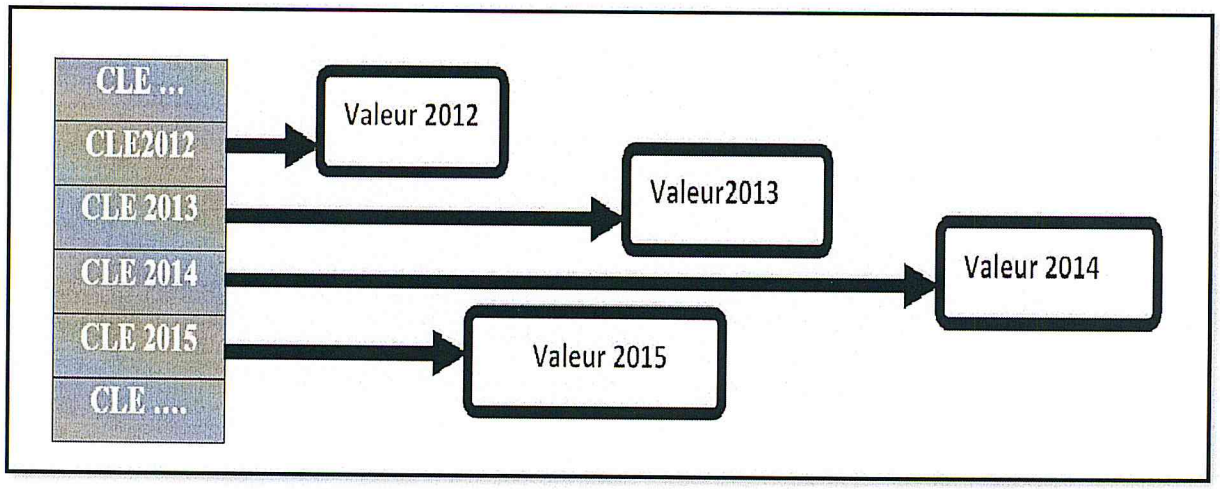



**Figure II-2: Illustration d'une Base de données Clef- valeur**

Parmi les implémentations existant on trouve Redis et Azure Table Storage liée à la plateforme Azure de Microsoft.

### *6.2.2 Orienté document*

Ce modèle se base sur le paradigme clé valeur. La valeur stockée dans ce cas est un document de type XML, Html, texte, etc.[20] Ce modèle est tout à fait adapté à l'indexation Web ou à la gestion documentaire. L'avantage de ce modèle est de pouvoir récupérer via une seule clé, un ensemble d'informations structurées de manière hiérarchique, ce qui implique plusieurs jointures dans le monde relationnel.

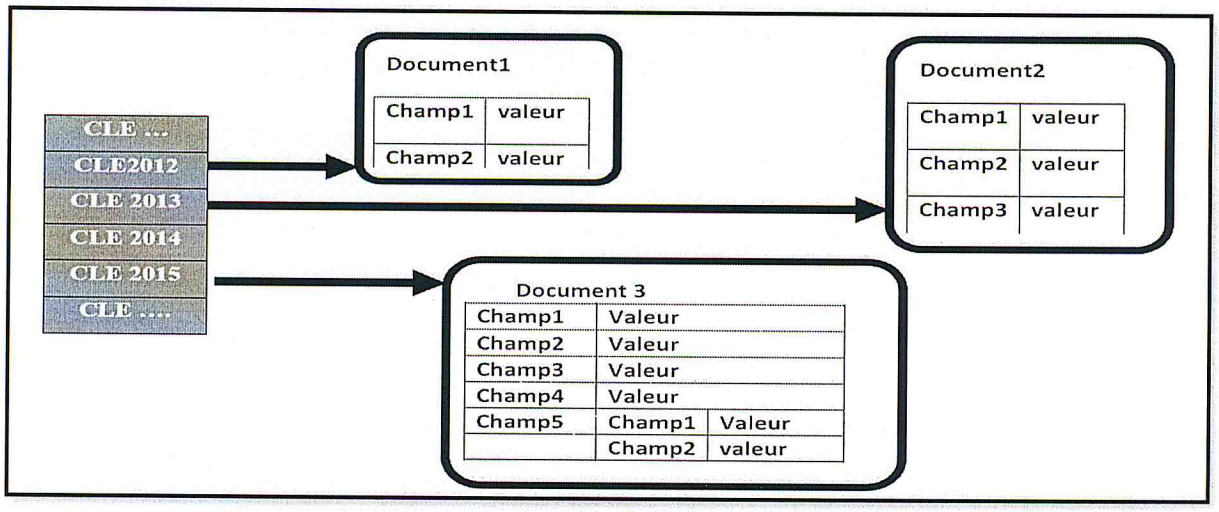



**Figure II-3: Illustration d'une Base de données Orientées Document**

Pour ce modèle, les implémentations le plus populaires est Mongo DB Développé en C++ et CouchDB d'Apache.

### 6.2.3 Orienté graphe

Ce modèle de représentation des données se base sur la théorie des graphes. Il s'appuie sur la notion de nœuds, de relations et de propriétés qui leur sont rattachées. Ce modèle facilite la représentation du monde réel, ce qui le rend adapté au traitement des données des réseaux sociaux.

Les bases de données Neo4j et OrientDB sont des exemples d'implémentations pour cette catégorie.

### 6.2.4 Orienté colonne

Ce modèle ressemble à première vue à une table dans un SGBDR [22]à la différence qu'avec une BD NoSQL orientée colonne, le nombre de colonnes est dynamique. En effet, dans une table relationnelle, le nombre de colonnes est fixé dés la création du schéma de la table et ce nombre reste le même pour tous les enregistrements dans cette table. Par contre, avec ce modèle, le nombre de colonnes peut varier d'un enregistrement à un autre ce qui évite de retrouver des colonnes ayant des valeurs NULL.

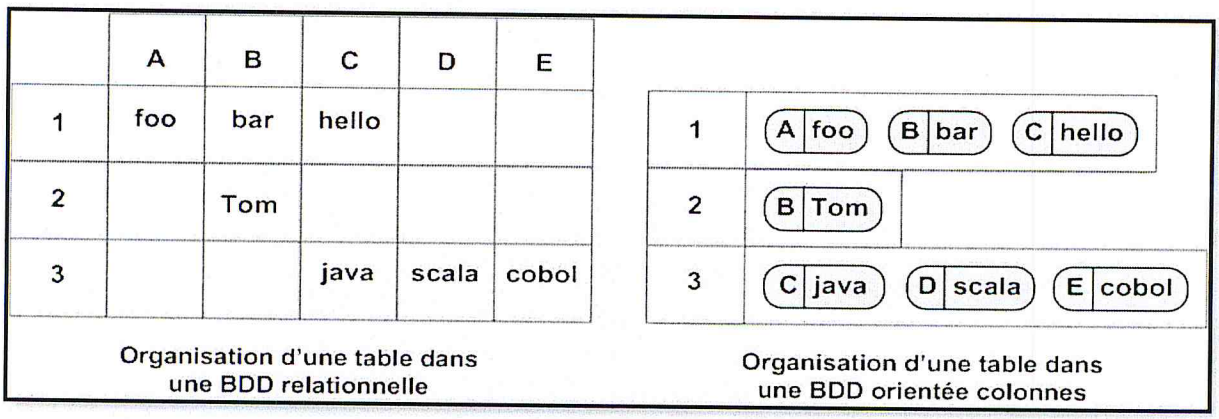



**Figure II-4: la différence entre SGBDR et NoSQL orienté colonne**

Bien que l'on trouve des variations en fonction de la base de données considérée, les concepts essentiels sont les suivants :

**La colonne :** Une colonne est la plus petite unité du modèle de données. C'est un triplet contenant un nom, une valeur et un timestamp (marquer de temps).

**Le super colonne :** Un super colonne est une liste de colonne, Il est vue comme une colonne contenant d'autres colonnes.

**La famille de colonnes** : Une famille de colonnes est un regroupement logique plusieurs colonnes ou super colonnes. Elle se ressemble à une table dans les bases de données relationnelles.

La catégorie SGBD orienté colonne est présente des avantages par rapport les autres catégories parce que :

- Ce modèle proche a une table dans un SGBDR.
- Les bases de données NoSQL orienté colonne sont moins complexes que BD NoSQL des autres catégories.
- Il est plus performant d'extraire des données pour une densité importante d'informations.
- Améliore grandement les performances sur les tris ou agrégations de données car ces opérations sont réalisées via des clés de lignes déjà triées.

Les implémentations les plus populaires pour ce modèle sont : HBase et Cassandra de la fondation Apache.

## 7. Les plateformes les plus répandues de traitement de BIG DATA

### 7.1. Apache hadoop

Apache Hadoop est une plateforme ou un système distribué qui répond au problématiques de stockage et l'analyse des données massives[23]. D'une part, il propose un système de stockage distribué via un système de fichier HDFS (Hadoop Distributed File System), ce dernier offre la possibilité de stocker la donnée en la dupliquant. D'autre part, Hadoop fournit un système d'analyse des données appelé MapReduce. Ce dernier officie sur le système de fichiers HDFS pour réaliser des traitements sur des gros volumes de données[16]. Hadoop est conçu comme un système de traitement orienté batch.

#### *7.1.1 Hadoop Distributed File System(HDFS)*

HDFS est un système de fichiers distribué tolérant au faute conçu pour stocker des gégantes des données soit structurées ou non sur un ensemble de serveurs distribués[24]. HDFS est conçu pour stocker de manière fiable des très grands fichiers et Les blocs d'un fichier sont répliqués pour assurer la tolérance aux pannes.

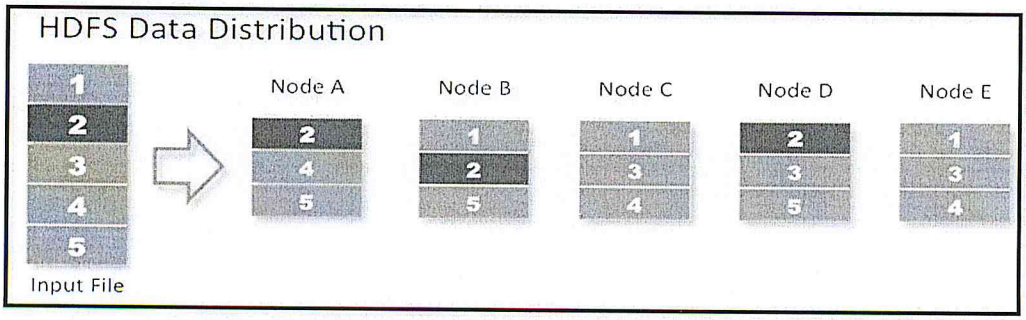



**Figure II-5: la distribution de données dans hadoop**

L'architecture HDFS est de type maitre/esclave qui se décompose essentiellement en un namenode, le maître, un secondary namenode et plusieurs datanodes (noeuds de données).

- **Namenode :** Un cluster HDFS compose d'une seule NameNode, un serveur maître qui gère l'espace de noms de système de fichiers et réglemente l'accès aux fichiers par les clients. Donc le NameNode va orchestrer les données, et contient les informations concernant l'emplacement des différentes répliques et gère les différents nœuds de données (appelé aussi serveurs de région) qui contient une partie des données.
- **Secondary Namenode(backup node) :** Le Namenode dans Hadoop est un point unique de défaillance. Si ce service est arrêté, il n'y a pas moyen de pouvoir extraire les blocs d'un fichier donné. Pour répondre à cette problématique, un Namenode secondaire appelé Secondary Namenode a été mis en place dans l'architecture Hadoop. Son fonctionnement est relativement simple puisque le Namenode secondaire vérifie périodiquement l'état du Namenode principal et copie les métadonnées. Si le Namenode principal est indisponible, le Namenode secondaire prend sa place.
- **Datanode :** le Datanode contient les blocs de données. Les Datanodes sont sous les ordres du Namenode, Ils sont donc sollicités par les Namenodes lors des opérations de lecture et d'écriture. En lecture, les Datenodes vont transmettre au client les blocs correspondant au fichier à transmettre. En écriture, les Datanodes vont retourner l'emplacement des blocs récemment créés.

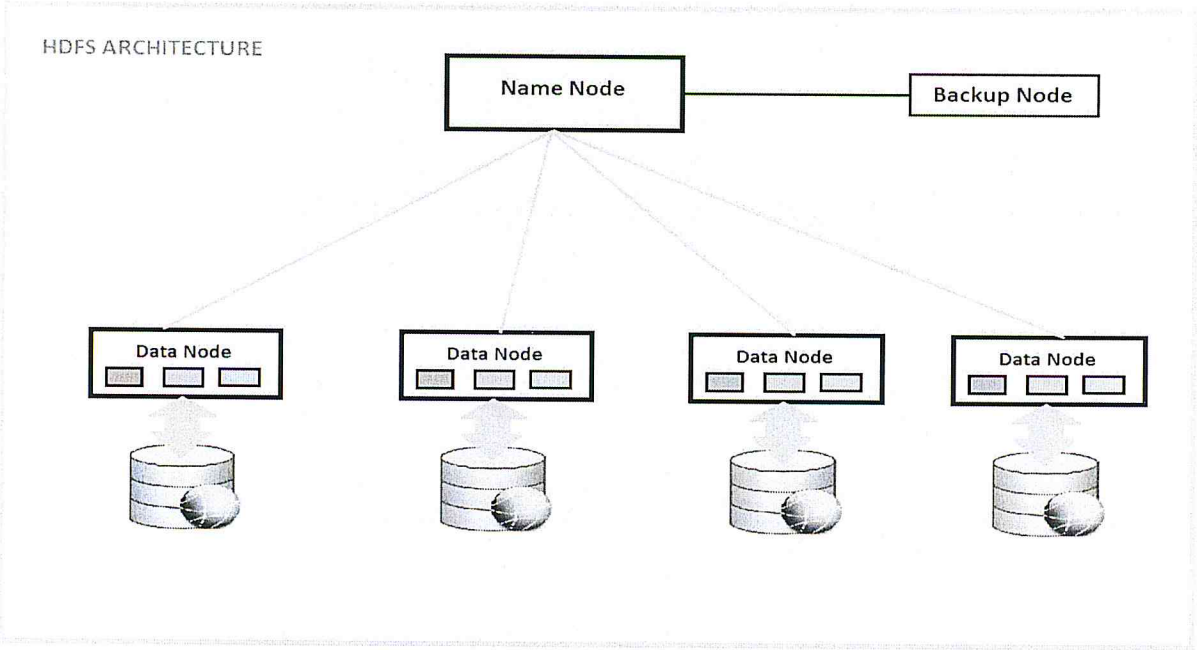



**Figure II-6: un schéma expliquant les différents nœuds du HDFS**

### 7.1.2 MAP REDUCE

MapReduce est un modèle de programmation dérivé des langages de programmation fonctionnels et notamment des fonctions *Map* et *Reduce*[25]. Partant d'un ensemble de données partitionné, un traitement indépendant est appliqué en parallèle sur chacune des partitions. Les résultats de ces traitements sont ensuite agrégés, puis renvoyés à l'utilisateur. L'application de ces fonctions sur une grappe de machines (*cluster)* de taille variable est alors abstraite, tout comme les détails (et potentielles difficultés) techniques dus à la parallélisation d'un programme informatique.

MapReduce implémente les fonctionnalités suivantes :

- Parallélisation automatique des programmes Hadoop.
- Gestion transparente du mode distribué.
- Tolérance aux pannes.

**a) La phase Map**

Le maping est la distribution des données sur plusieurs clusters parallèles où les calculs intermédiaires seront effectués.

**b) La phase Reduce**

Dans cette phase les résultats des calculs intermédiaires distribués sont recentralisé en vue du calcul final.

La figure suivante représente un exemple illustratif de processus MAP/REDUCE :

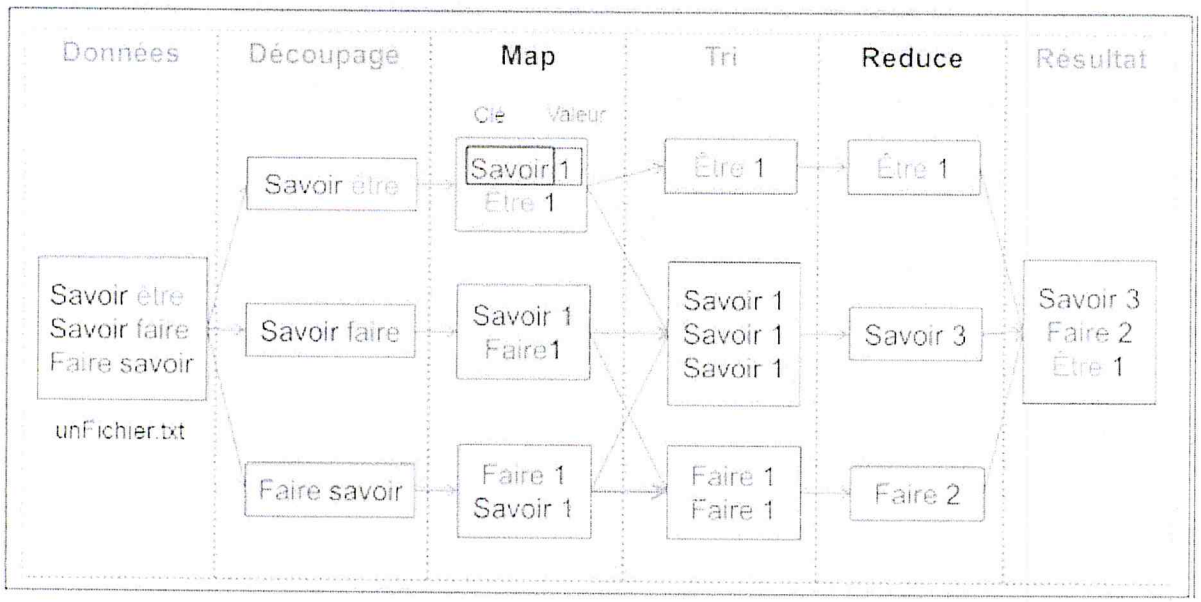



**Figure II-7: exemple illustratif de MapReduce**

### 7.2. Apache Spark

Apache Spark est une platforme de calcul ou un moteur de traitement open source très puissante représente une amélioration de performance par rapport apache hadoop, il est 100x plus rapide que hadoop[26]de fait quelle support le traitement IN MEMORY plutôt que d'utiliser le stockage temporaire dans HDFS, comme Hadoop.fait et aussi et contrairement a hadoop, Spark utilise plusieurs threads au lieu de plusieurs processus pour atteindre le parallélisme sur un nœud de données. Apache spark est un système distribué support le traitement orienté streaming ce qui permet réalisé l'analyse en temps réel.

Spark offre deux principales abstractions pour la programmation ou traitement parallèle: jeux de données répartis élastiques (RDD) et les opérations parallèles sur ces ensembles de données[27].

#### *7.2.1 Resilient Distributed Datasets (RDDs)*

RDD est le concept central du platforme Spark[28]. Un RDD est une collection partitionnée sur un ensemble de machines d'enregistrements en lecture seule qui peut être créée par des opérations à partir de données présentes dans un stockage stable ou à partir d'autres RDDs. Si une partition de RDD est perdue, le RDD dispose de suffisamment d'informations sur la manière dont il a été produit pour recalculer la partition manquante, donc Les RDDs sont tolérants à la panne et proposent des structures de données parallèles qui laissent les utilisateurs :

- persister explicitement les données intermédiaires en mémoire,
- Contrôler leur partitionnement afin d'optimiser l'emplacement des données,
- manipuler les données en utilisant un ensemble important d'opérateurs.

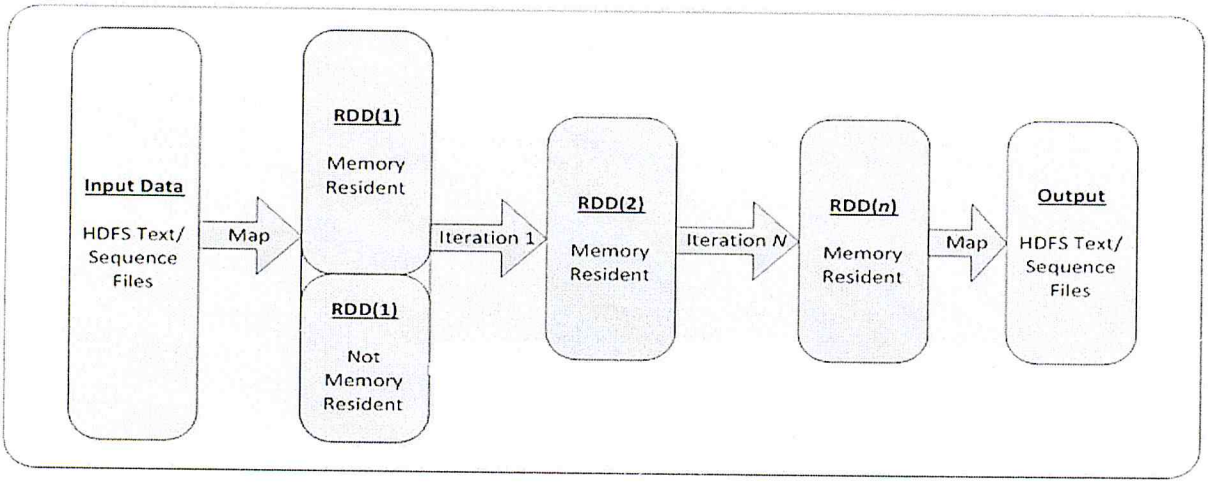



**Figure II-8: le principe de RDD de Spark**

### *7.2.2 Les opérations parallèles sur RDD :*

Les RDD supportent deux types d'opérations :

- Les transformations
- Les actions

#### a) Les transformations :

Les transformations ne retournent pas de valeur mais elles retournent un nouveau RDD. Rien n'est évalué lorsque l'on fait appel à une fonction de transformation, la fonction prend juste un RDD et retourne un nouveau RDD. Parmi les fonctions de transformation nous citons :

**Map :** l'opération Map permet la création d'un nouveau RDD en appliquant une fonction à chaque élément du RDD de départ.

**Filter :** qui retourne un nouveau RDD obtenu en sélectionnant les items de la source pour lesquels la fonction de filtrage est satisfaite.

**GroupByKey :** qui à partir d'un RDD de type (clé, valeur), retourne un RDD de paires (clé, Séquence <valeur>).

#### b) Les actions :

Les actions évaluent et retournent une nouvelle valeur. Au moment où une fonction d'action est appelée sur un objet RDD, les requêtes de traitement des données sont calculées et le résultat est retourné. Parmi les fonctions de transformation nous citons :

**Reduce :** permet d'agréger et combiner les éléments du RDD en utilisant une fonction pour produire des résultats au programme pilote. La fonction utilisée devrait être associative afin de pouvoir être correctement calculée en parallèle.

**Collect:** cette opération permet de regrouper et retourné tous les items du RDD.

**Count :** qui retourne le nombre d'items d'un RDD.

**countByKey :** pour un RDD de type (clé, valeur), retourne l'ensemble de paires (clé, Int) avec le nombre de valeurs pour chaque clé.

## 7.3. Apache Hadoop vs Apache Spark

Ainsi nous sommes arrivés à cette comparaison entre les deux plus connus des platformes de big data.

#### a- La rapidité :

Donc quand on parle de rapidité et comme a été évoqué spark il est de 10 à 100 fois plus rapide que hadoop, tout simplement en réduisant le nombre de lectures et d'écritures sur

ledisque.Dans Spark, le concept de DDR (des jeux de données répartis résilients) permet d'enregistrer les données sur la mémoire et le conserver sur le disque si et seulement si elle est nécessaire. D'après la référence[29]une comparaison dans ce contexte effectuée à donne les résultats illustrés dans la figure

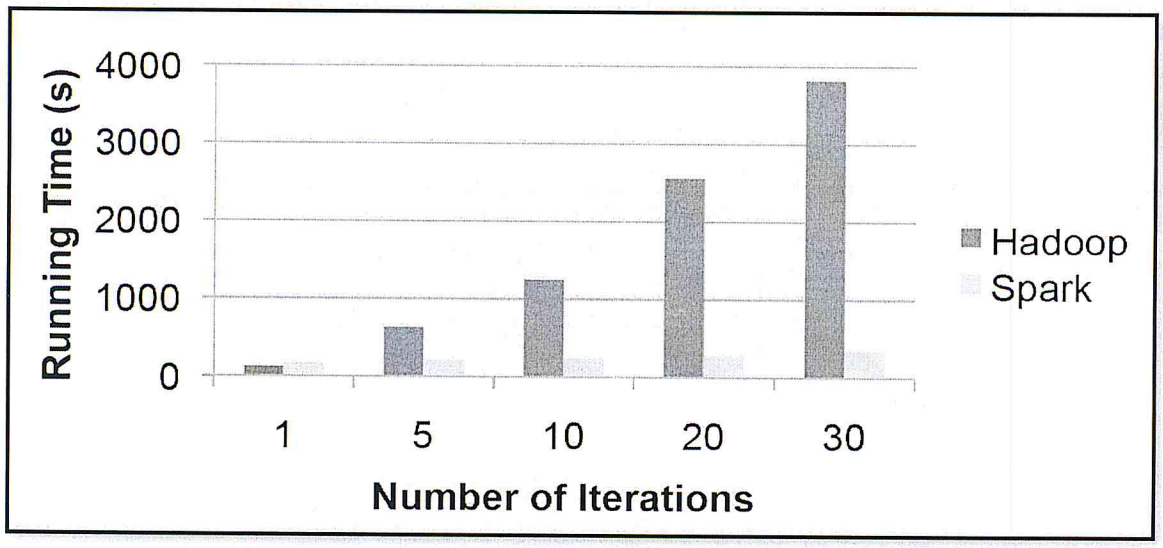



**Figure II-9:comparaison de performance entre spark et hadoop**

**b- L'analyse en temps réel :**

En cas de Hadoop on obtient juste de traitement par lot de données stockées mais avec Spark il est ainsi possible de faire de traitement de données en temps réel ou se que on appelle traitement de flux de données grâce Spark streaming.[30]

**c- La gestion facile :**

Avec Spark il est possible d'effectuer Streaming, Traitement par lots et Machine Learning tous dans le même cluster. ainsi avec Spark il est possible de contrôler différents types de charges de travail, donc si il y a une interaction entre diverses charges de travail dans le même processus, il est plus facile à gérer et sécuriser ces charges de travail qui viennent comme une limitation avec Hadoop MapReduce.

**d- Tolérance de panne :**

Spark et Hadoop ont tous les deux une bonne tolérance de panne, cependant Hadoop MapReduce est légèrement plus tolérants de fait que le MapReduce repose sur les disques durs.

## 8. Conclusion

Dans ce chapitre, Nous avons pu mettre le point sur certaines notions de l'univers technologiques de big data, plus particulièrement les technologies de stockages NoSQL et les plateformes de traitement d'où nous avons constaté que apache spark représente de plus en plus des points forts par rapport d'autre plateforme de traitement.

# III. CHAPITRE III : CEP ET L'ANALYSE EN TEMPS REEL

## 1. Introduction

L'ensemble des objets intégré dans l'IoT génère et émettre de manière continus des événements qui sont désigné par des objets véhiculant des informations. Ces événements de différentes sources exigent un traitement en temps réel. Et aussi, la multiplicité des sources d'événements a rendu complexe la relation entre l'apparition de certains événements d'une part et le traitement associé d'autre part.

## 2. Le concept temps réel

La signification de l'expression « temps réel » varie selon les domaines d'application ou points de vue. Un système temps réel est un système réactif qui accorde une grande importance à l'exactitude des résultats et qui doit respecter un temps de réponse minimal (contrainte temporelle). Le temps de réponse d'un système correspond au temps écoulé entre l'acquisition des données (entrée) et l'envoi des commandes à l'environnement (sortie).

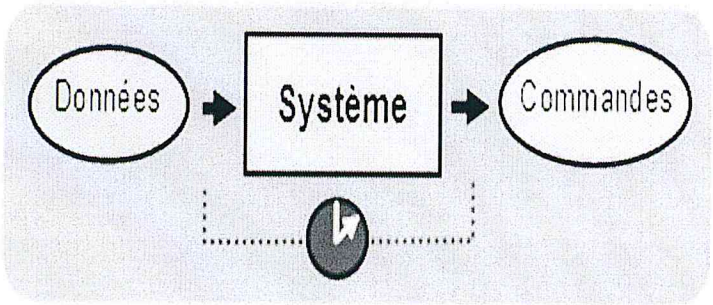



**Figure III-1: Temps de réponse d'un système**

Entre l'entrée et la sortie du système, un traitement de l'information est réalisé et ce temps de traitement correspond principalement au temps de réponse du système. Dans la définition générale d'un système temps réel, aucune valeur maximale pour le temps de réponse n'est spécifiée puisqu'il doit simplement être acceptable pour l'application.

Un système d'analyse en temps réel [31] est un système évènementiel disponible, scalable et stable capable de prendre des décisions et d'actions avec une latence acceptable pour l'application.

## 3. Architecture orientée évènements (EDA)

L'architecture orientée évènements est un modèle de conception d'un système, plus particulièrement d'un système informatique. L'architecture proposée par l'EDA est modulaire et découplée ; les différents composants interagissent entre eux simplement par la diffusion d'évènements.

L'architecture orientée évènements est une forme d'architecture adaptée à des applications ayant des contraintes importantes de temps réel. L'EDA est basée sur l'écoute, la détection et la création d'événements. Des traitements sont ainsi réalisés dès la réception d'un événement.

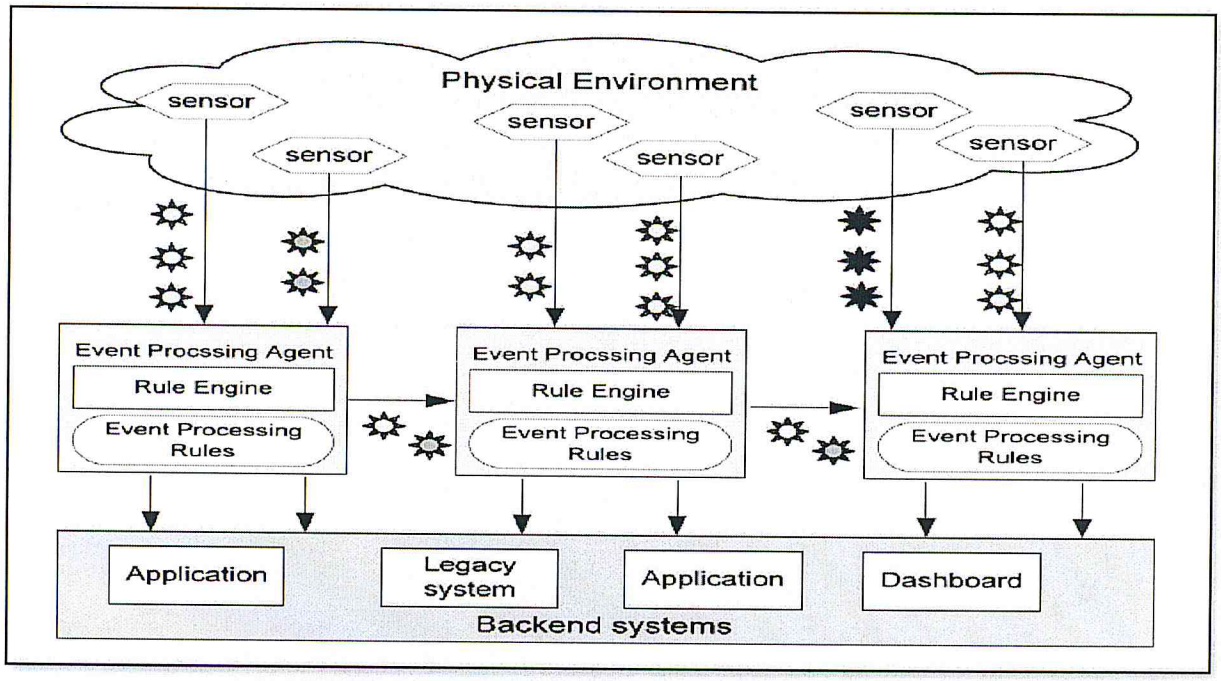



**Figure III-2: Event Driven Architecture [32]**

Le concept n'est pas nouveau, puisqu'il est à la base de n'importe quel système de régulation.Un changement d'état survient, il est détecté par le système, le système y réagit pour le compenser.L'exemple le plus simple est le thermostat, qui régule l'ouverture d'une valve en fonction de latempérature de la pièce. Si on ouvre une fenêtre en plein hiver, le thermostat détectera une fortebaisse de la température et il allumera le chauffage à pleine puissance.

### 3.1. Les composants d'une architecture EDA

Une architecture EDA se composait de différents éléments, parmi les plus importants nous trouverons des évènements, bien sûr. Leur production se fera au moyen de générateurs d'évènements et puisque la vraie puissance de l'EDA réside dans la capacité à pouvoir traiter des évènements et y réagir intelligemment, il faut pour cela un Processeur d'évènements.

#### *3.1.1 Évènement*

Avant de commencer à les manipuler, définissons ce qu'est un évènement. On peut le voir comme quoi que ce soit de notable survenant à l'intérieur ou à l'extérieur du système. Un évènement sera défini comme un changement d'état, c'est à dire une variation suffisamment

significative des paramètres définissant l'état système. N'importe quelle donnée apportant une information inédite peut donc être considérée comme un évènement. Cela peut être : une action effectuée par un utilisateur du système, une donnée envoyée par un capteur, à fréquence fixe …etc. Ces quelques exemples tentent de montrer que la définition d'évènement est volontairement très large et recouvre tout type d'information pertinente pour le système.

Formellement, on peut définir un évènement à plusieurs niveaux, plus ou moins précis, apportant chacun une information propre au système :

- **Notification :** à ce niveau, ce qui nous intéresse est simplement de savoir si l'évènement a eu lieu ou pas, ce qui constitue un fait en soi.
- **Définition :** au deuxième niveau, on analyse le type d'évènement : quel est son genre, sa classe.
- **Détail :** enfin, au troisième niveau, on s'intéresse aux détails de l'évènement. Les données de l'évènement sont appelés attributs ou encore des propriétés.

### *3.1.2 Générateur d'évènements*

Cet élément va s'occuper de faire la transition entre la source de l'évènement (qui peut être complètement externe au système, venant d'une application tierce, fournie manuellement par une personne, etc) et le système EDA. Il va convertir l'évènement dans une représentation standardisée compatible avec le reste du système.

### *3.1.3 Processeur d'évènements*

C'est ici que se situe toute la puissance et l'intérêt d'une solution EDA : le traitement des évènements. Une distinction est fréquemment faite entre trois types de traitements : le simple event processing reçoit un seul évènement notable et réagit à un changement de condition précis. l'Event Stream Processing (ESP), quant à lui, réagit à plusieurs évènements, mais travaille sur un et un seulflux de données. Le Complex Event Processing (CEP), enfin, consiste à corréler les évènements entre eux (de manière causale, temporelle ou spatiale). Àpartir d'un ensemble d'évènements pouvant venir de plusieurs flux, on tente d'inférer l'occurrence d'un évènement complexe. Ceci implique notamment la reconnaissance de motifs. La notion de CEP est présentée en détail dans les sections suivantes.

On remarque que les deux premiers peuvent être vus comme des cas particuliers du troisième.

De manière générale, les moteursde traitement d'évènements que l'on trouvera sur le marché seront qualifiés de moteurs CEP, dont la définition permet d'inclure les trois types de processeurs.

### 3.2. Les avantages d'une architecture orientée événements (EDA)

Quand on parle de EDA, on cite souvent dans les avantages de ces architectures des concepts comme le couplage faible, la réutilisabilité et l'agilité.

- **Le couplage faible :** l'un des avantages majeurs de l'EDA est de permettre un couplage extrêmement faible entre les différents composants. Ce découplage est rendu obligatoire par la communication par évènement.
- **La réutilisabilité :** La maintenabilité/changeabilité se définit comme capacité à maintenir le système fonctionnel au cours du temps et à le faire évoluer en fonction de l'évolution de son environnement, EDA permet de mieux faire évoluer le système, notamment par une réutilisabilité accrue des composants, et de mieux le maintenir donc on peut toucher à un composant du système sans que cela n'ait d'impact majeur sur les autres.
- **Agilité :** d'après les points précédents, on constate que la facilité d'évolution et de maintenance rend le système agile, c'est-à-dire capable d'être adapté rapidement et efficacement aux conditions changeantes du métier.

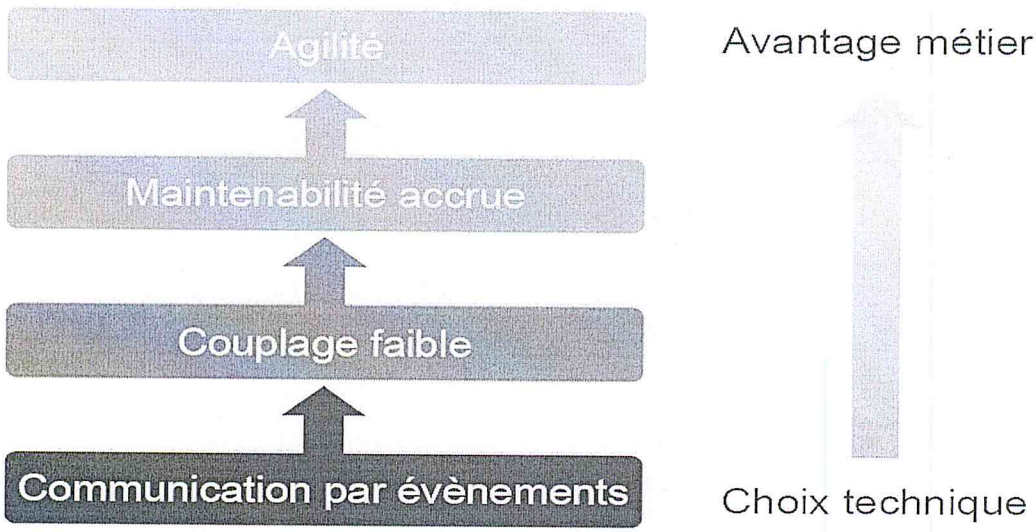



**Figure III-3:** Avantages d'une architecture orientée événements.

## 4. Complex event processing (CEP)

### 4.1. Définition

Le terme CEP signifie traitement des événements complexes. C'est l'analyse de flux d'information désigné par événement, un événement représente l'occurrence de quelque chose Intéressant[33]. Ce type de traitement permet l'extraction de nouvelles informations non pas par l'analyse des événements simples et indépendants, mais par l'analyse d'un ensemble d'événements complexe corrélé et agrégé.

Selon EPTS[2], Complexe Event Processing est tout calcul/traitement exécutant des opérations sur des événements complexes tels que la lecture, la création, la transformation, l'abstraction des données portées par ces événements[34]. En d'autres termes, Complex Event Processing traduit plus des concepts d'analyse d'événements porteurs d'informations prévenant de diverses sources de données que de solutions logicielles.

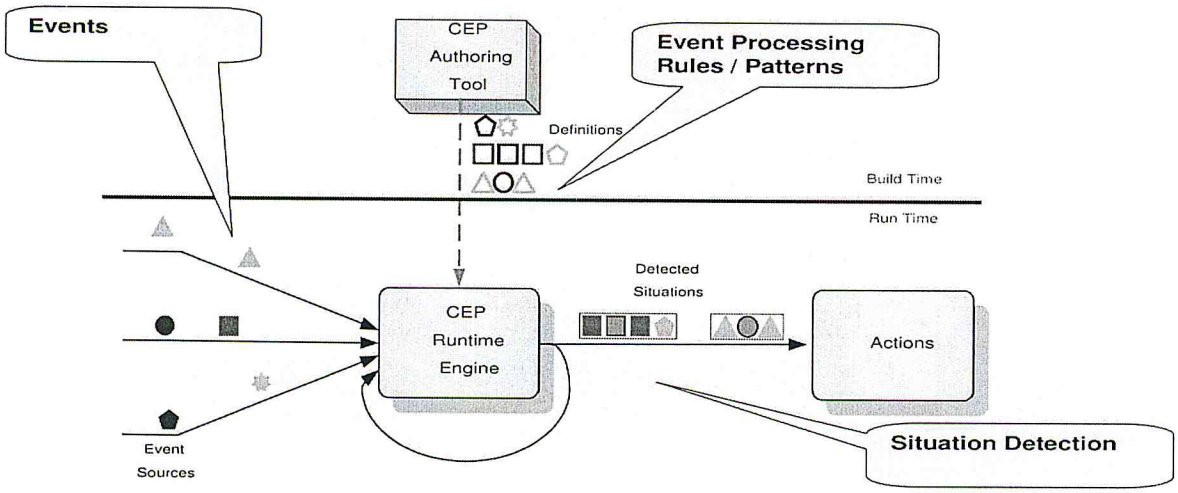



**Figure III-4:** processus de Traitement d'évènements.

La technologie CEP fournit des mécanismes qui peuvent traiter la contraint en temps réel. Le CEP permet capter les événements produisant et analyser en temps réel les informations véhiculées par eux, afin de détecter une séquence d'événements appelée motif, par conséquent une suite d'actions a exécutée.[33]

[2]Event Process Technical Society

### 4.2. Le moteur CEP

Un moteur CEP est une plate-forme permettant le développement et le déploiement d'applications capables de traiter et analyser des données en temps réel, ainsi le principe de CEP est basé sur un moteur d'inférence qui reçoit des flux d'événementsélémentaires, venant par différentes voies et par volume important. Le traitement consiste àeffectuer des corrélations entre ces différents flux afin de détecter une séquence d'événementsappelée motif. un moteur CEP fonctionne, contrairement à d'autres moteurs d'inférence, en temps-réel. Un moteur CEP ne s'intéresse pas au passé, il s'intéresse au présent.

À titre d'exemple si on veut superviser le trafic routier nous pouvons définir un motif pourdétecter l'augmentation du débit (nombre voiture/minute). Nous pouvons définir ce motif par

une expression qui calcule la moyenne des voitures qui passent pendant une minute et qui notifie un opérateur par une alarme dès que le débit dépasse un seuil (par ex :100 voitures/minute).

Les événements arrivent, dans ce cas, des puces RFID installées dans les voitures. Dès qu'elle passe, un récepteur déclenche un événement et l'envoie au moteur CEP.

### 4.3. Terminologie et concepts de base

#### *4.3.1. Événement*

La notion d'événement dans CEP est identique à celle de l'EDA. Un événement est l'enregistrement d'un fait. Sous sa forme la plus simple, il indique le moment et le lieu du fait, accompagné de données connexes.

#### *4.3.2. Événement dérivé, composé ou complexe*

Un événement complexe est l'abstraction d'un ensemble d'événements dits membres. L'occurrence de l'événement complexe se traduit par l'occurrence de tous ses événements membres. Un événement dérivé, par contre est généré à l'issue d'un traitement d'un ou plusieurs événements. Il représente tout simplement le résultat de l'application d'une règle de génération. Un événement composé, quant à lui, est une collection d'événements satisfaisant un patron [35]. Il est généralement construit en utilisant des constructeurs particuliers, comme une conjonction, disjonction, etc.

#### *4.3.3. Un flux ou un nuage d'événement*

Un flux est une séquence d'évènements correctement ordonnés dans le temps. Ce qu'il est important de relever est que l'ordre dont il est question ici est un ordre total et que la

dimension utilisée pour ordonner ces évènements est le temps. Par contre dans un nuage, les évènements arrivent de manière désordonnée et proviennent de sources multiples et c'est au moteur de gérer de manière adéquate la synchronisation temporelle pour pouvoir corréler correctement les données.

#### *4.3.4. Différences entre CEP et ESP*

La différence entre ESP et CEP se rapproche de la différence entre le fonctionnement en flux et en nuage présentée au paragraphe précédent.Le premier concernait le traitementd'un flux de données en temps réel. Le deuxième concernait le traitement complexe d'un nuage de données, c'est à dire de données venant de sources diverses, pas forcément ordonnées dans letemps. Les deux domaines se sont rapprochés, le CEP adoptant l'aspect flux et temps réel, alors quel'ESP étendait progressivement ses capacités d'analyse. Actuellement, un consensus se forme pour affirmer quel'ESP est en fait un cas particulier de CEP.

#### *4.3.5. Patrons d'événement «event pattern»*

La raison d'existence d'un moteur CEP est la recherche de motifs (patterns), un motif ou patron d'événement permet la détection de séquencesd'événements liés par des relations temporelles, booléennes, de similarité, d'indépendanceou de causalité. Lorsqu'une séquence d'événements correspond à un patron, elle constituealors une instance de ce patron.

### 4.4. Types de systèmes CEP

La plupart des solutions CEP peuvent être classés en deux catégories principales, ainsi deuxdéfies [36]:

- **Calcul orienté CEP :** axée sur l'exécution d'algorithmes en ligne en réponse à des événements entrant dans le système. le défie est de pouvoir supporter la charge d'événements et détecter les motifs tout en respectant les contraintes temporelles, Autrement dit l'algorithme de détection doit être efficace et scalable.
- **Détection orientée CEP :** axée sur la détection des combinaisons de modèles d'événements appelés situations. le défie ici est de pouvoir représenter la situation ou le motif qu'en veut détecter. Cela nécessite l'utilisation d'un langage qui supporte des opérations complexes telles que la jointure, l'agrégation et le filtrage temporel.

### 4.5. Expression des règles et des motifs

Un moteur CEP, nous l'avons dit, permet de traiter des évènements au moyen de l'expression de règles permettant de reconnaître des motifs et d'appliquer des actions. Un

aspect essentiel d'un moteur CEP est donc son langage d'expression de ces règles et des motifs que l'on désire reconnaître.

Il existe deux principales catégories de langages : les langages basés sur des règles ECA et ceux basés sur des requêtes. Pour les langages basés sur des règles, la sémantique d'une règle ECA est : lorsqu'un événement E se produit, si la condition C est satisfaite alors exécuter l'action A, tandis que ceux basés sur des requêtes sont généralement dérivé du langage SQL.

### 4.6. Langage de Requêtes d'événement(EQL)

Un système CEP est capable de gérer un très grand nombre de requêtes extrêmement complexes. Il prend en compte le fait que les données à traiter (i.e. les événements) sont vues comme un flux qui entre et sort en permanence. Ceci ressemble à l'interrogation des bases de données mais avec une inversion de la logique. En effet, les requêtes changent dans une base de données alors que les données sont stockées de façon permanente et changent rarement. Dans un système CEP, c'est l'inverse : les données changent fréquemment et les requêtes sont généralement statiques et fixées à l'avance.

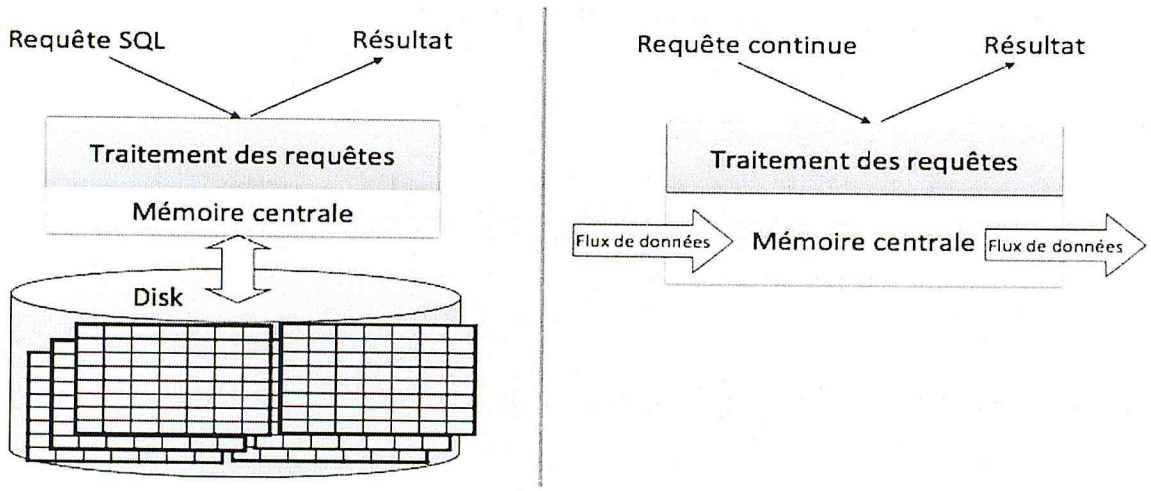



**Figure III-5:** Différence entre un système de bases de données et un système CEP

Cette similarité est à l'origine du fait que la plupart des langages de traitement d'événement, appelés aussi langages EPL «», ont une syntaxe ressemblante à celle du langage SQL utilisé dans les bases de données.

Le langage de requête d'événement est à la base des moteur CEP, les besoins requise pour un langage de requête d'événement peuvent être décrites avec les points suivants :

- **Extraction des données :** les événements contiennent des données qui sont pertinentes pour décider si et comment réagir face à eux. donc l'accès à ces données doit être possible dans conditions de requêtes.
- **Composition :** Il doit être possible de joindre plusieurs événements individuelles de sorte que leurs occurrences combinés donnent au fil du temps un événement complexe. Cette composition doit souvent être sensibles aux données (par exemple, rejoindre uniquement les événements concernant le même client).
- **Relations temporelles entre les événements :** les requêtes d'événements impliquent souvent des conditions temporelles exprimant que les événements doivent se produire dans un intervalle de temps particulier ou dans un ordre particulier.
- **Relation de causalité :** La définition de cette relation peut varier en fonction des moteurs. Néanmoins, il ne suffit pas qu'un évènement en précède un autre pour qu'il en soit la cause. Le glossaire de EPTS propose la définition suivante de la causalité dans le cadre du traitement d'évènements[36]: Un évènement A cause un autre évènement B si A devait survenir pour que B survienne.

### 4.7. Moteurs CEP existants

Nous allons présenté ci-dessous une liste des moteurs CEP bien que cette liste n'est pas exhaustive.

#### a) Tibco BusinessEvents :

Tibco est un des acteurs les plus en vue dans le domaine du CEP, ils ont intégré le traitement d'évènements au moyen de leur produit BusinessEvents.

BusinessEvents est un produit propriétaire pour lequel il n'y a pas de documentation disponible publiquement. Ce produit propose deux langages. Le premier est un langage basé sur des règles, que l'on écrit sous la forme évènement-condition-action (ECA) et qui ressemble à du Java. Le second est un langage basé sur des requêtes, suivant une syntaxe similaire à du SQL.

#### b) Oracle CEP :

Oracle CEP est un serveur Java pour le développement et le déploiement des applications orienté évènement de haute performance, Notons que le produit est propriétaire et fournit un environnement déclarative riche basée sur Oracle continue Query Language (Oracle CQL) et un langage de requête basée sur SQL avec des constructions ajouté lui permettant de supporté les flux de données.

**c) Esper :**

Esper est un moteur CEP sous licence libre (GPL). Il en existe deux versions, écrites respectivement en Java et en C# (NEsper). Esper a la particularité de proposer son moteur sans fioritures : c'est au développeur d'écrire son environnement d'exécution, son système de stockage de règles, etc. Certains y verront un manque de fonctionnalités, mais pour d'autre une grande liberté laissée à l'utilisateur pour l'intégrer à son environnement. Le langage utilisé par Esper est l'EPL, qui est un langage dit de requêtes, basé sur SQL.

### 4.8. Intérêt du CEP

Le CEP représente un moyen de supervision efficace, en étant à l'écoute de tout ce qui se passe dans un environnement et en utilisant le CEP on pourra révéler une situation critique ou une opportunité au moment adéquat ou en temps réel. Cette agilité lui permet d'être le cerveau dans une architecture de type EDA comme l'Internet des objets. Le CEP est alors une clé pour créer un système flexible capable de réagir selon le contexte.

## 5. Conclusion

Nous avons montré, au cours de ce chapitre, l'importance des CEP dans une architecture orientée évènements qui est présenté dans notre cas par l'internet des objets. Ces processeurs détectent des motifs d'évènements et déclenchent des actions en fonction de ceux-ci. Ainsi, en combinant les informations de tous les évènements diffusés dans un système, ces processeurs peuvent évaluer avec précision la situation de ce système et satisfirent la contrainte de temps réel.

# IV. CHAPITRE IV : ETUDE CONCEPTUEL

## 1. Introduction

L'étude conceptuelle vise à mettre en place un système d'information capable à répondre aux objectifs souhaité, pour cela, nous aborderons dans cette partie la modélisation conceptuelle du système en utilisant le langage de modélisation UML. Nous intéresserons au premier temps à trois grands aspects : fonctionnel, dynamique et statique du système, puis au schéma de base de données NoSQL a mis en place pour le stockage de données et modèle structurel d'événements pour le traitement d'événement et nous terminons par la solution élaborée pour le routage des usagers routiers.

## 2. Démarche utilisée

Pour concevoir notre système, nous avons choisi d'utiliser le langage de modélisation UML. Ce dernier permet de définir et de visualiser un modèle, à l'aide de diagrammes qui offrent une vue complète des aspects statiques et dynamiques d'un système. UML comporte treize types de diagrammes, cependant dans notre travail, nous utiliserons seulement trois types de diagrammes : diagramme de cas d'utilisation, diagramme de séquence et diagramme de classe qui sont assez suffisant pour concevoir un système informatique.

## 3. Recueil des besoins

Au niveau Recueil des besoins, notre objectif est une modélisation permettant d'analyser les besoins aux quelle répondre le système. Le modèle devra prendre en compte les éléments suivants :

- **Le phénomène de congestion** : la congestion est un phénomène physique essentiel et spécifique du transport routier selon lequel une concentration accrue de véhicule, par leurs gênes mutuelles, dégrade la vitesse de chacun et la qualité de service de l'ensemble du système.
- **Les perturbations exogènes** : Le réseau routier peut être perturbé parun certain nombre de facteurs exogènes plus ou moins aléatoires tels qu'accident, conditions météorologiques difficiles, chantier sur laroute, etc.
- **La présence de l'information dynamique** : le service d'information dynamique du trafic influence le choix de déplacement des usagers en améliorant leur connaissance de l'état du trafic et/ou leur donnant des conseils de guidage.
- **La gestion du trafic sur le réseau** : le service d'information et d'autres actions d'exploitation sont mutuellement dépendants car les mesures d'exploitation affectent la crédibilité de l'information et la diffusion de l'information, en revanche, conditionne le paramétrage des actions appliquées.

### 3.1. Diagramme des cases d'utilisations

Diagramme des case d'utilisation permet de structurer les besoins des utilisateurs et les objectifs correspondants d'un système, de manière précise ce que le système devrait faire vis à vis ses utilisateurs. Les deux concepts de base du ce diagramme sont : l'acteur et le cas d'utilisation qui modélise un service rendu par le système. La détermination des besoins est basée sur la représentation de l'interaction entre l'acteur et le système.

L'identification des différents acteurs et cas d'utilisation du système est récapitulé dans le tableau qui suit la figure de diagramme des case d'utilisation.

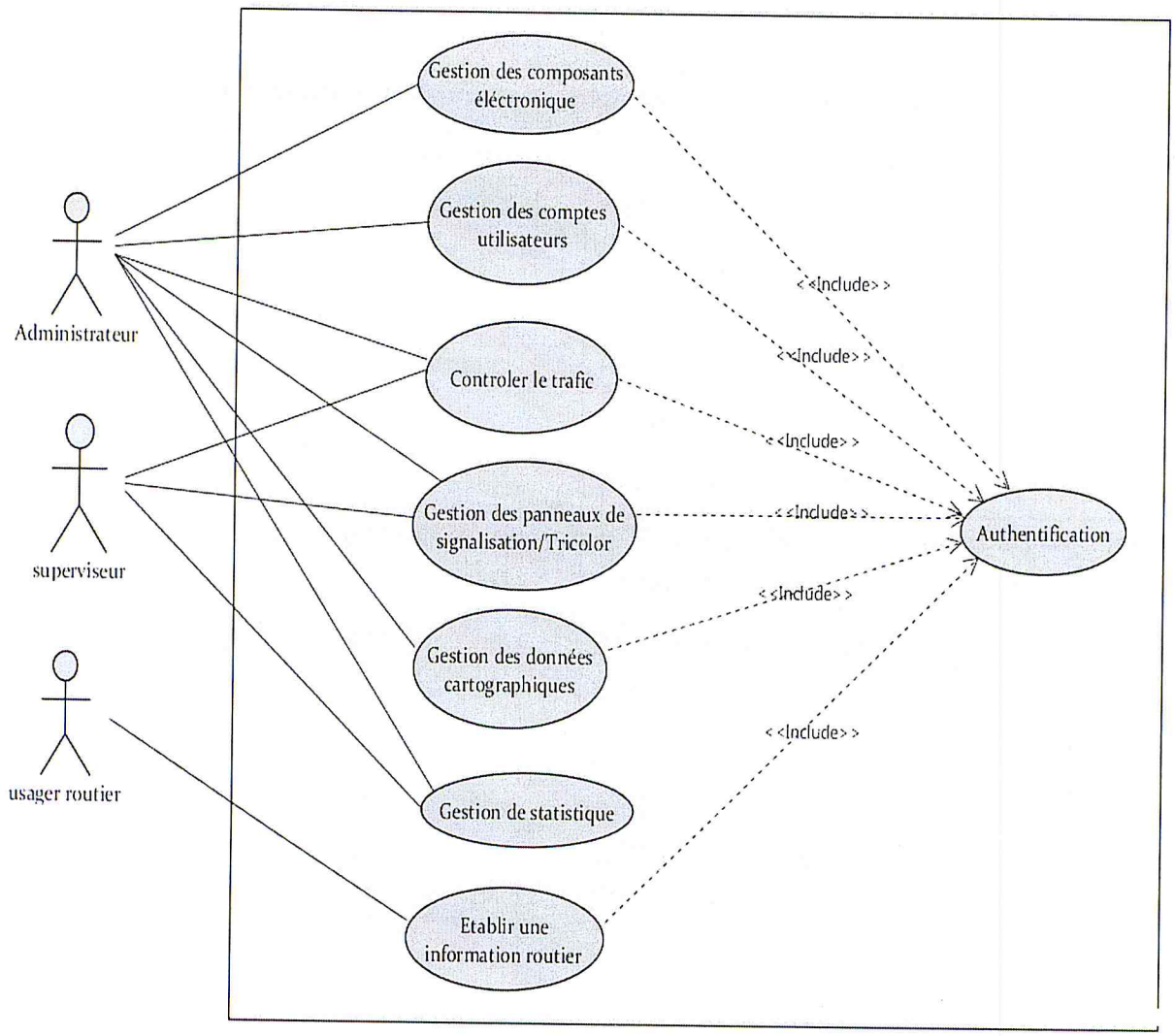



**Figure IV-1:Diagramme de cas d'utilisation global**

### 3.2. Description détaillée des cas d'utilisations

Le diagramme de cas d'utilisation au dessus montre les fonctionnalités globales offertes par le système. Dans ce qui suit, nous détaillons les différents cas d'utilisations.

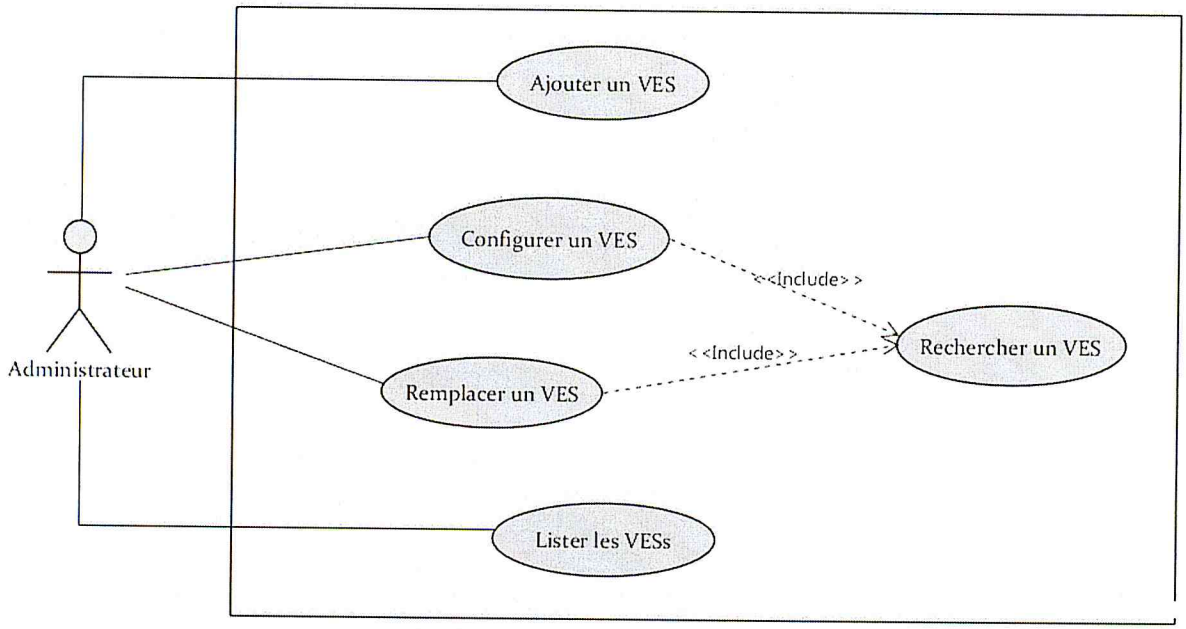



**Figure IV-2: Diagramme de cas d'utilisation "Gestion de recueil de données"**

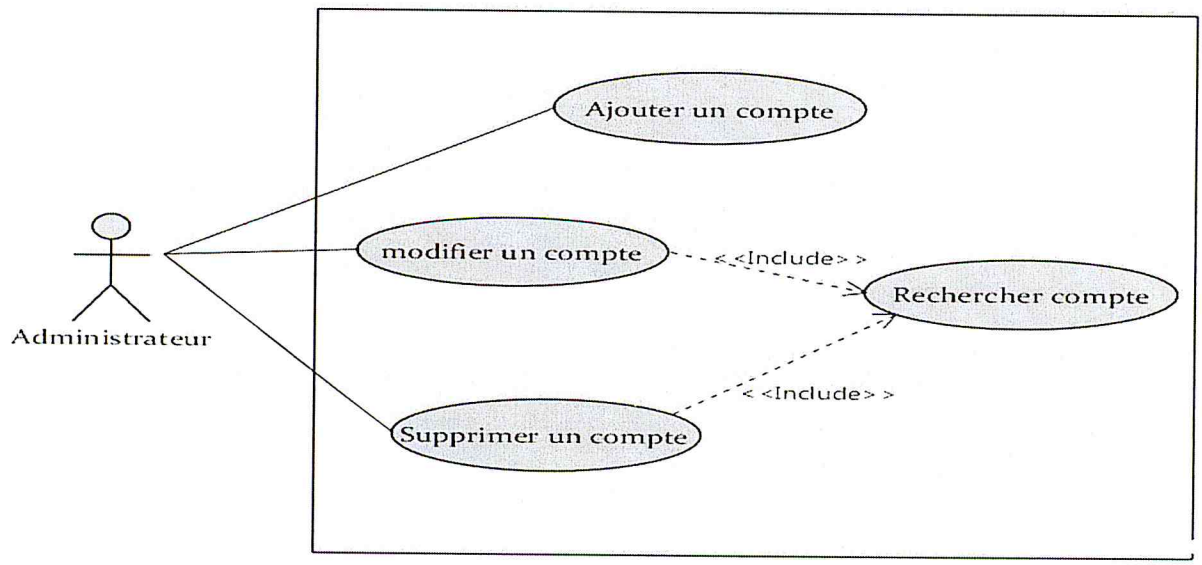



**Figure IV-3: Diagramme de cas d'utilisation "Gestion des comptes utilisateurs"**

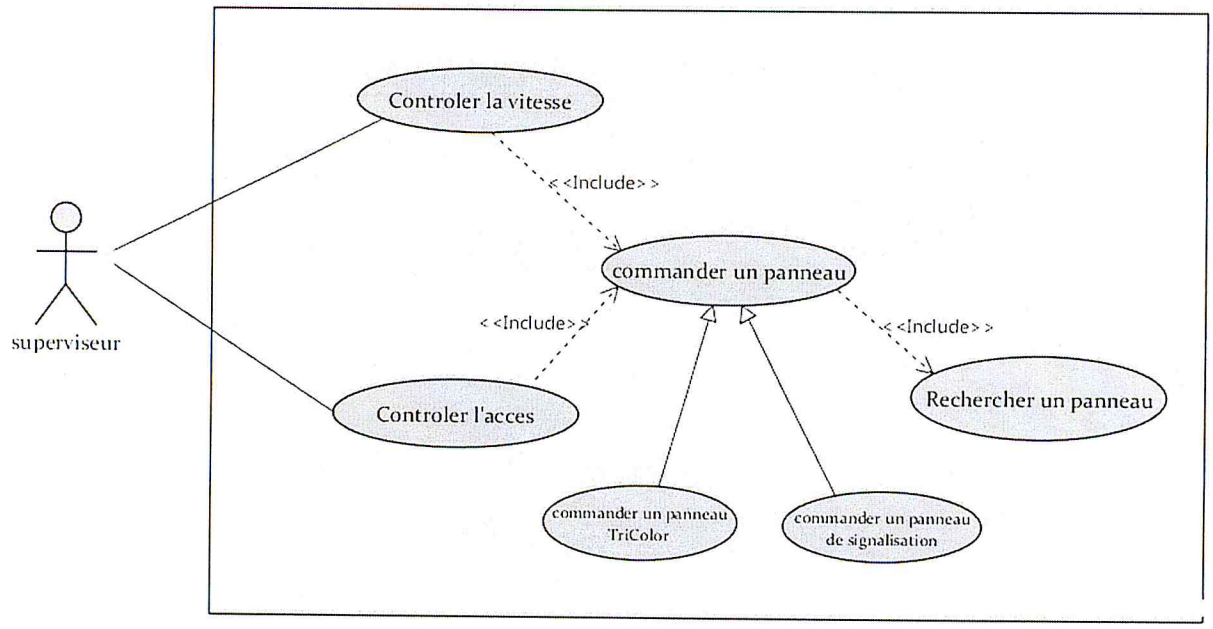



**Figure IV-4: Diagramme de cas d'utilisation "Controler le trafic routier"**

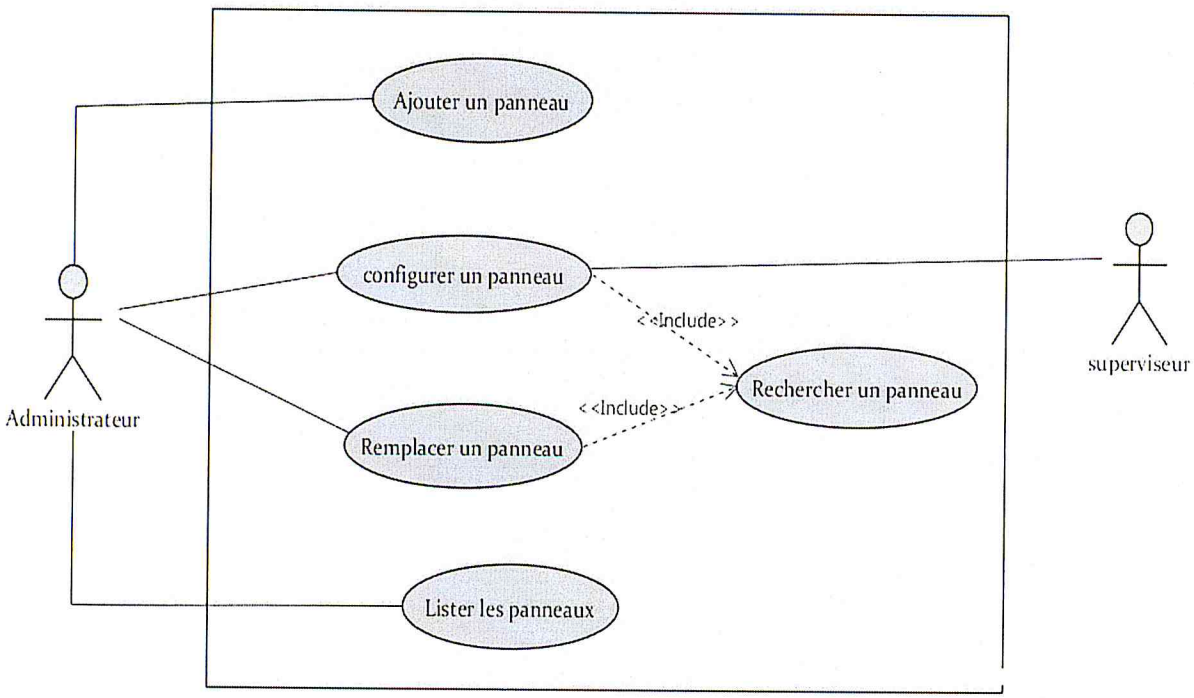



**Figure IV-5: Diagramme de cas d'utilisation "Gestion des panneaux des signalisations"**

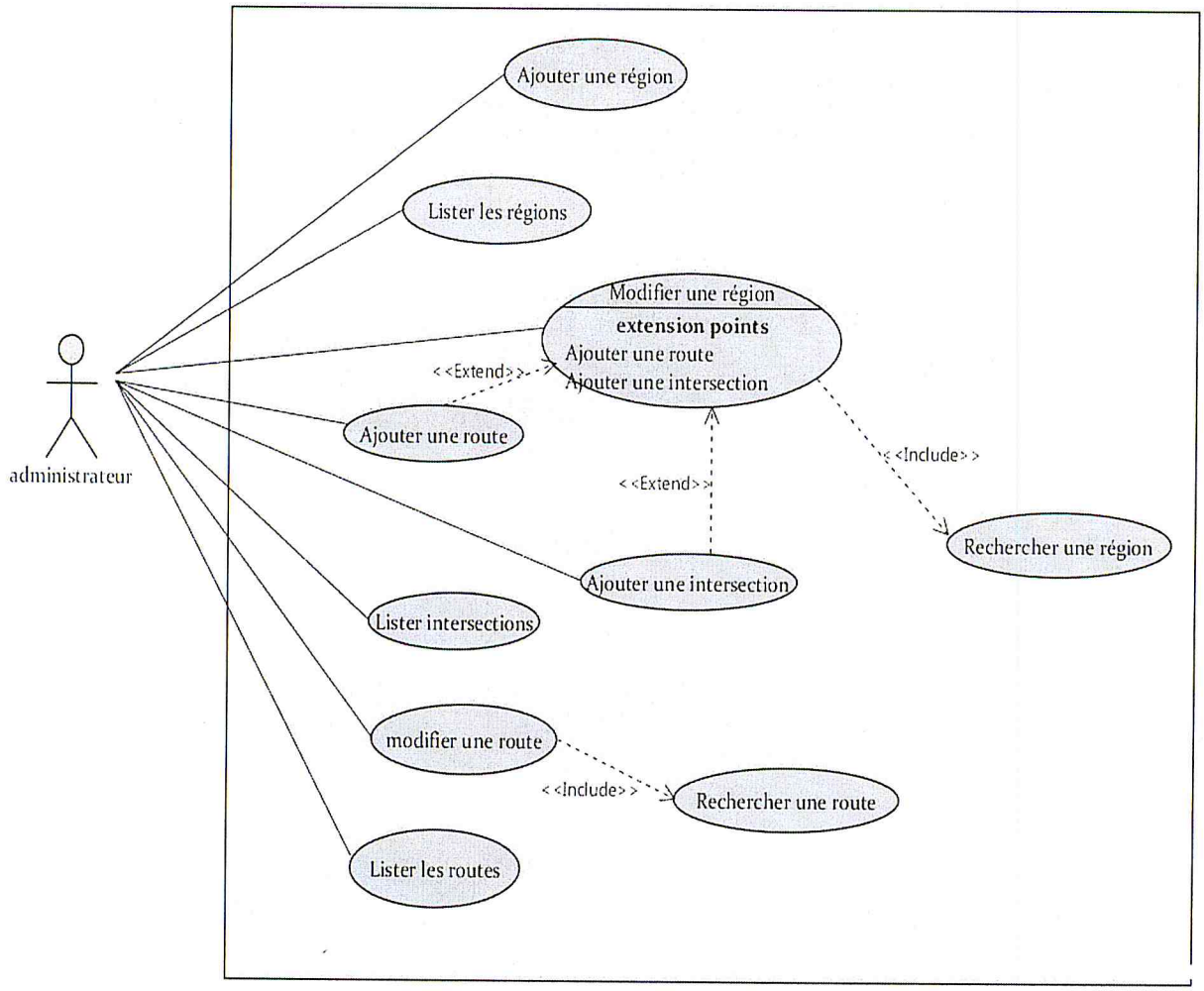



**Figure IV-6: Diagramme de cas d'utilisation "Gestion des données cartographie "**

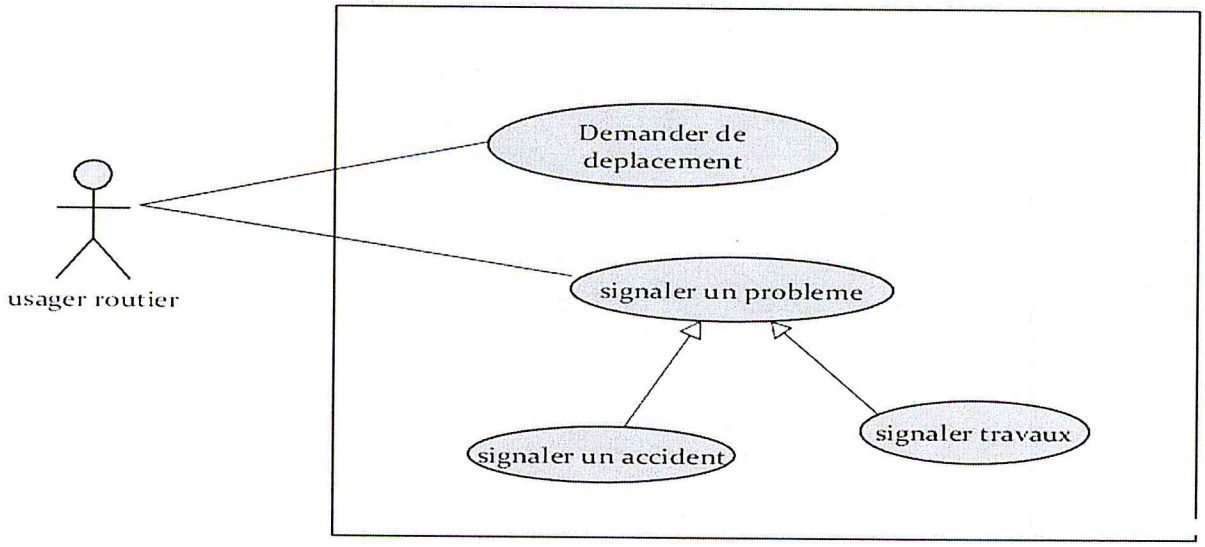



**Figure IV-7: Diagramme de cas d'utilisation "Etablir une information routier "**

## 4. Aspect dynamique du système

Aspect dynamique nous permettons de décrire les interactions entre les objets de manière évolutive selon un point devue temporel pour comprendre les tâches du système, pour appréhender cela nous utilisons le diagramme de séquence.

### 4.1. Diagramme de séquence

Le diagramme de séquence présente une vue dynamique du système, il met l'accent sur l'ordre d'échange des messages au cours de l'exécution du système et nous permettons de présenter les scénarios de cas d'utilisation et opérations du système.

Dans ce qui suit, nous allons présenter les différents diagrammes de séquence.

#### *4.1.1. Diagramme de séquence du cas « Authentification »*

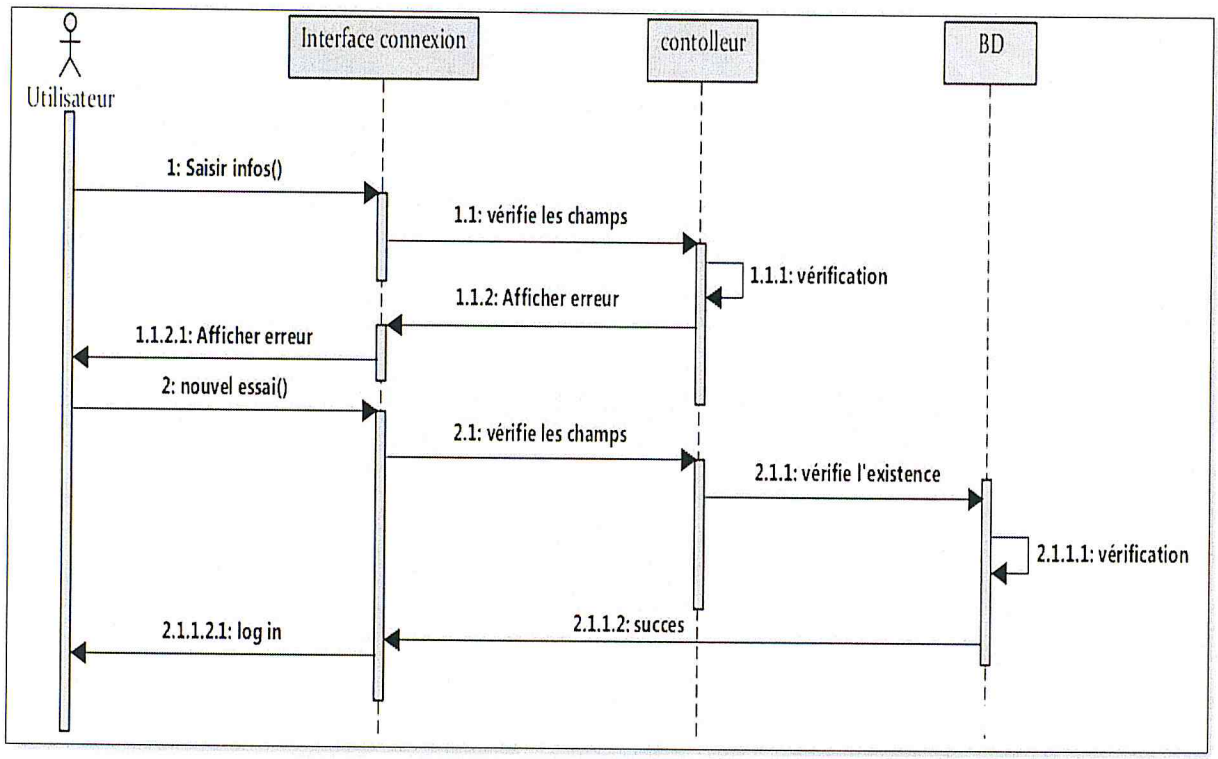



**Figure IV-8: schéma d'authentification**

### 4.1.2. Diagrammes de séquence du cas « Gestion des comptes utilisateurs »

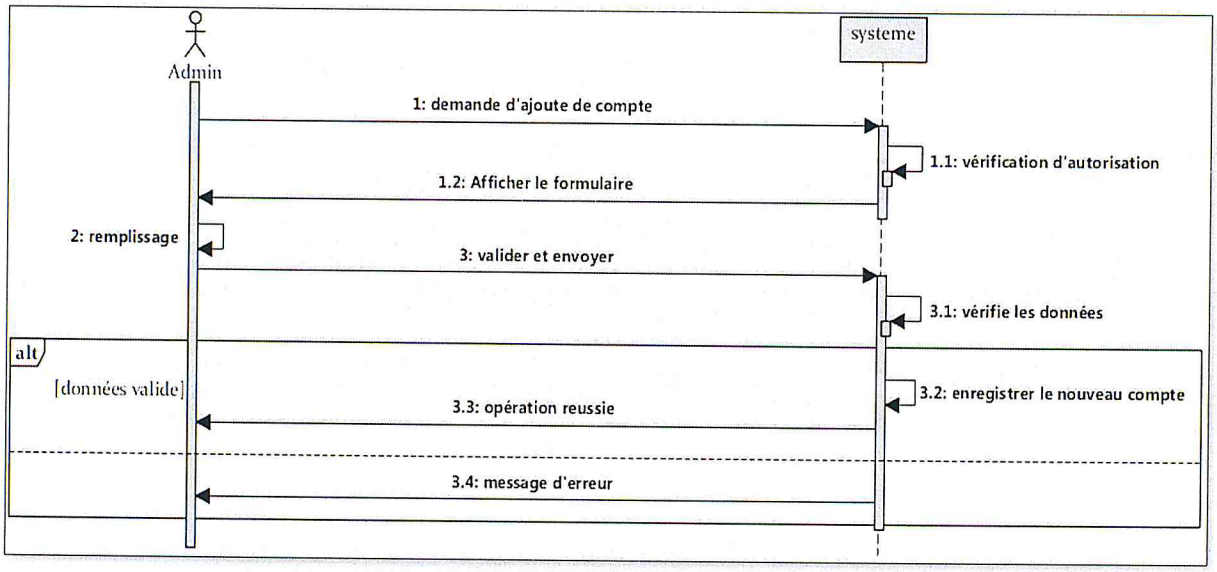



**Figure IV-9:diagramme de séquence de cas: ajouter compte**

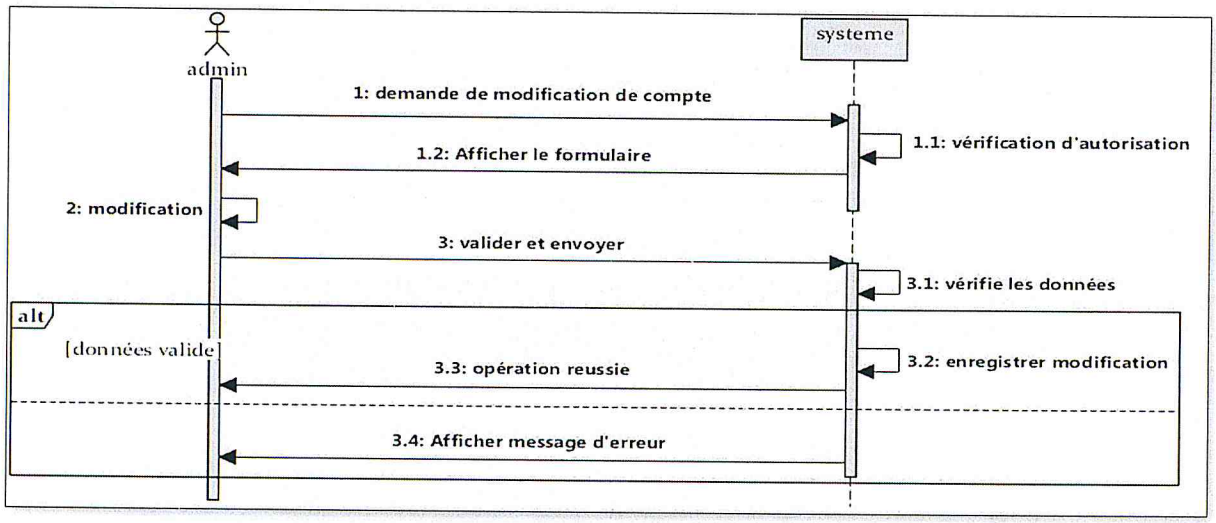



**Figure IV-10:diagramme de séquence de cas: modifier compte**

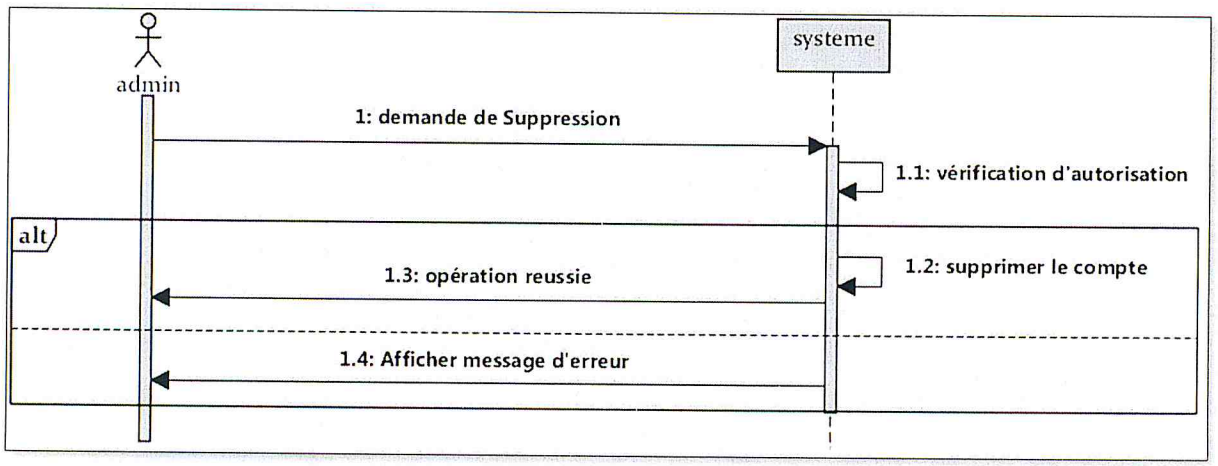



**Figure IV-11: diagramme de séquence de cas: supprimer compte**

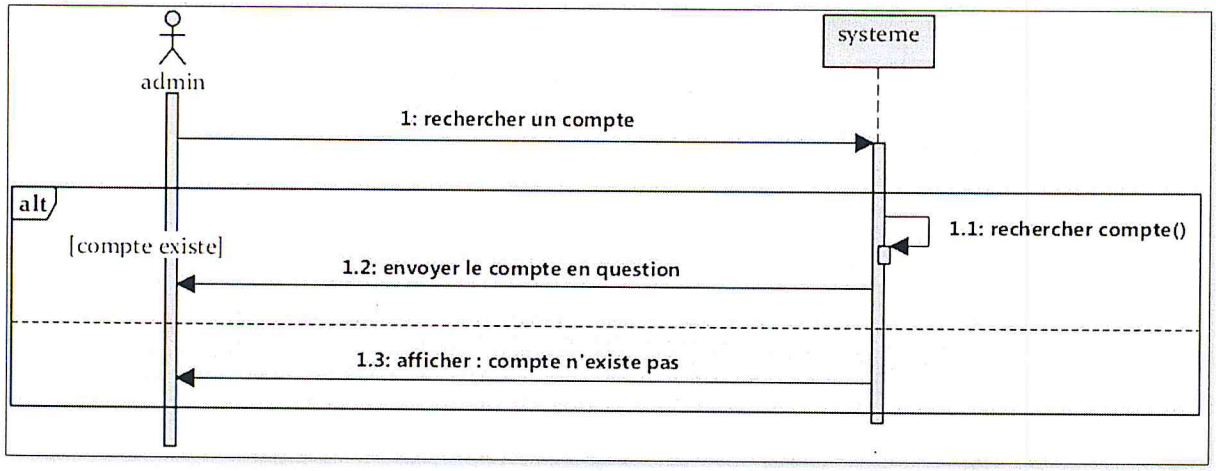



**Figure IV-12:diagramme de séquence de cas: rechercher compte**

### *4.1.3. Diagrammes de séquence du cas « Gestion des composants électroniques VES »*

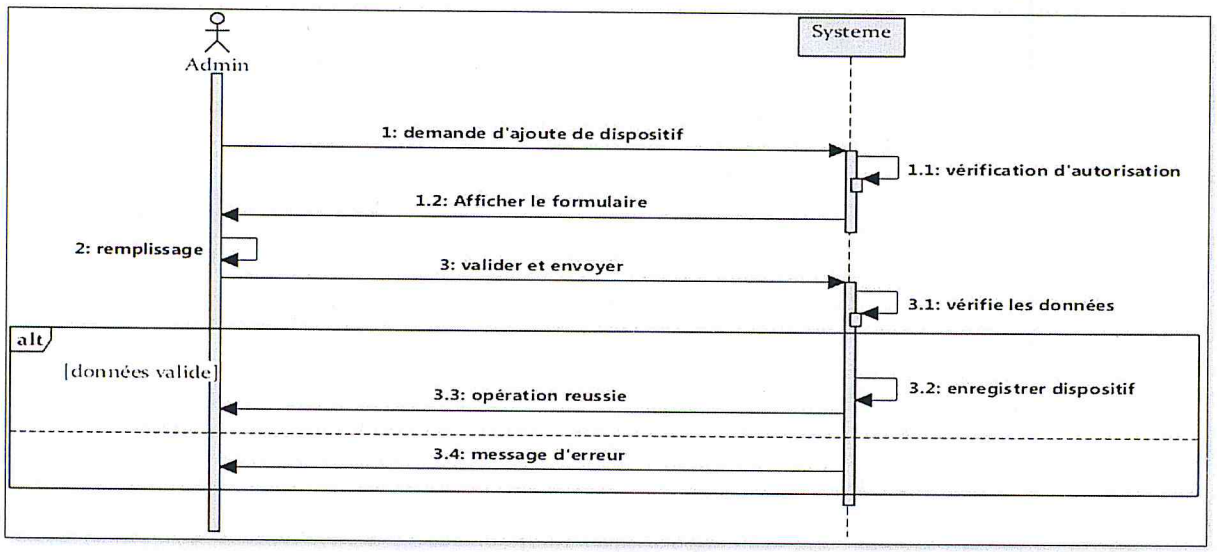



**Figure IV-13:diagramme de séquence de cas: ajouter dispositif VES**

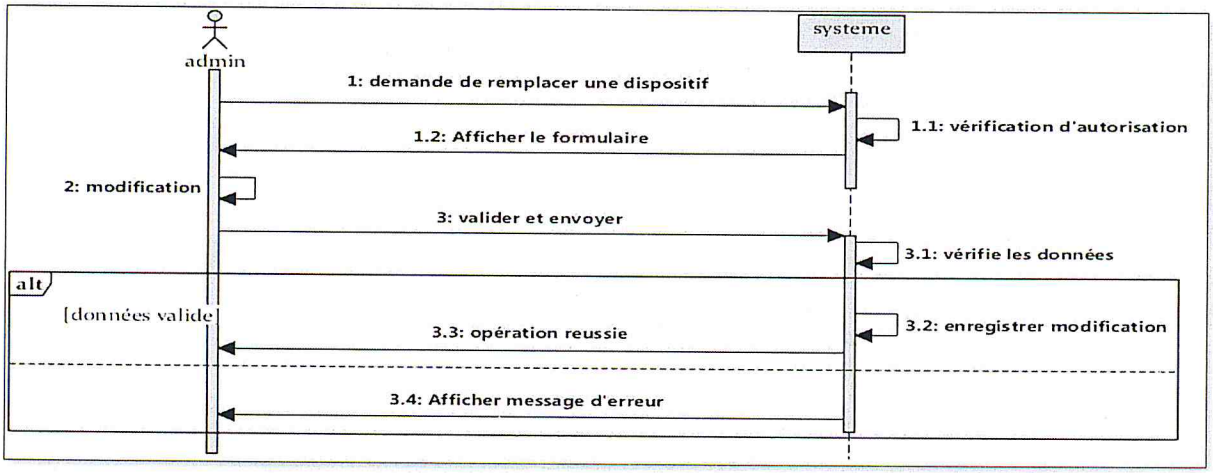



**Figure IV-14:diagramme de séquence de cas: remplacer dispositif VES**

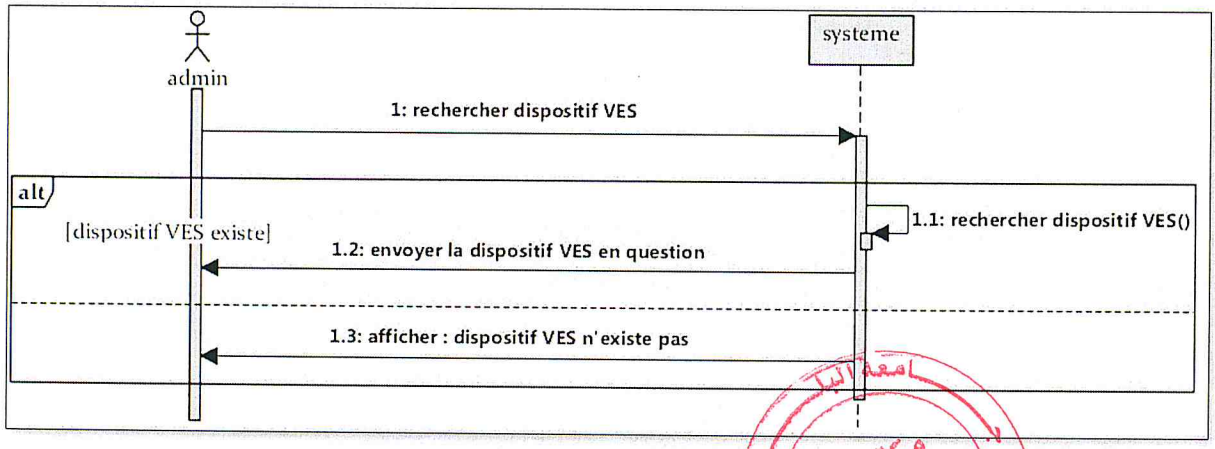



**Figure IV-15:diagramme de séquence de cas: rechercher dispositif**

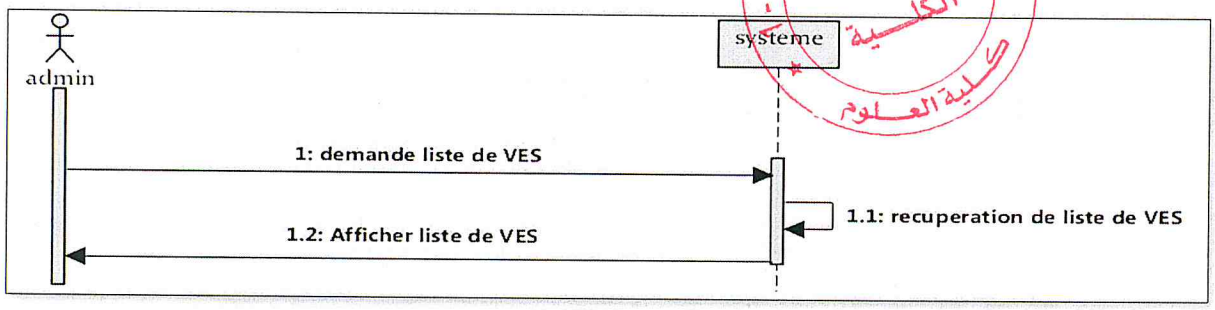



**Figure IV-16:diagramme de séquence de cas: lister les dispositifs**

### *4.1.4. Diagrammes de séquence du cas « Contrôler le trafic routier»*

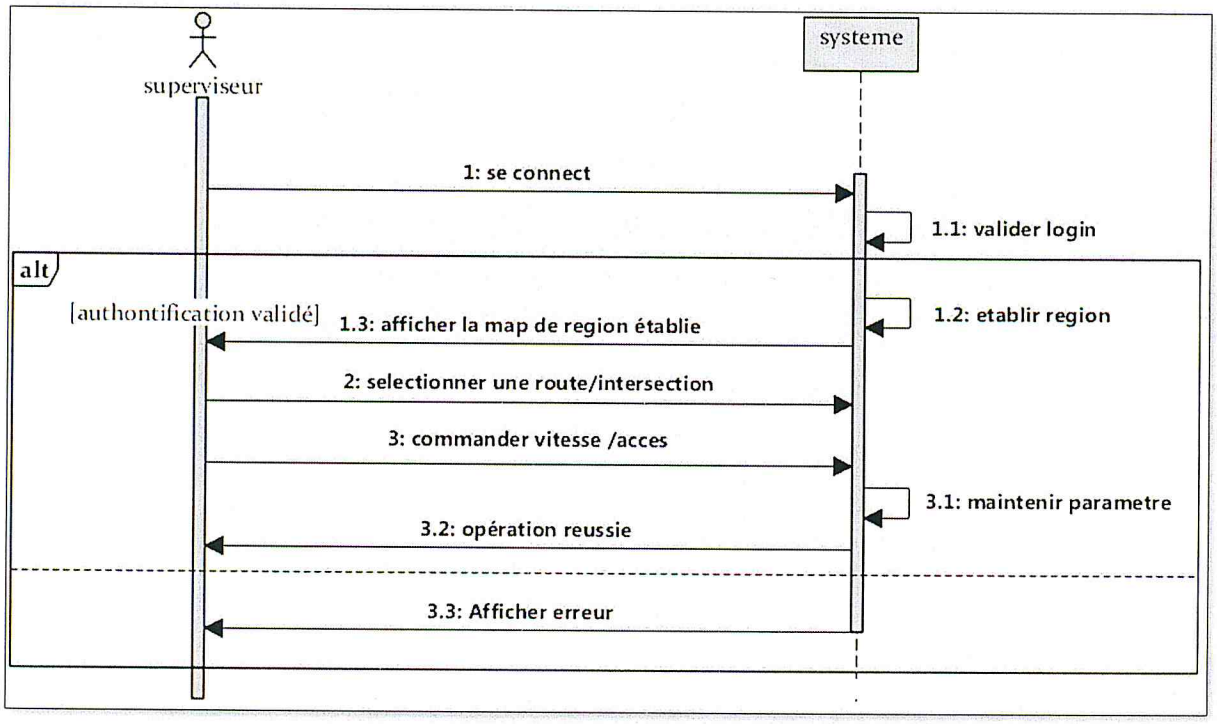



**Figure IV-17:diagramme de séquence de cas: gérer trafic routier**

### 4.1.5. Diagrammes de séquence du cas « Gestion des panneaux» :

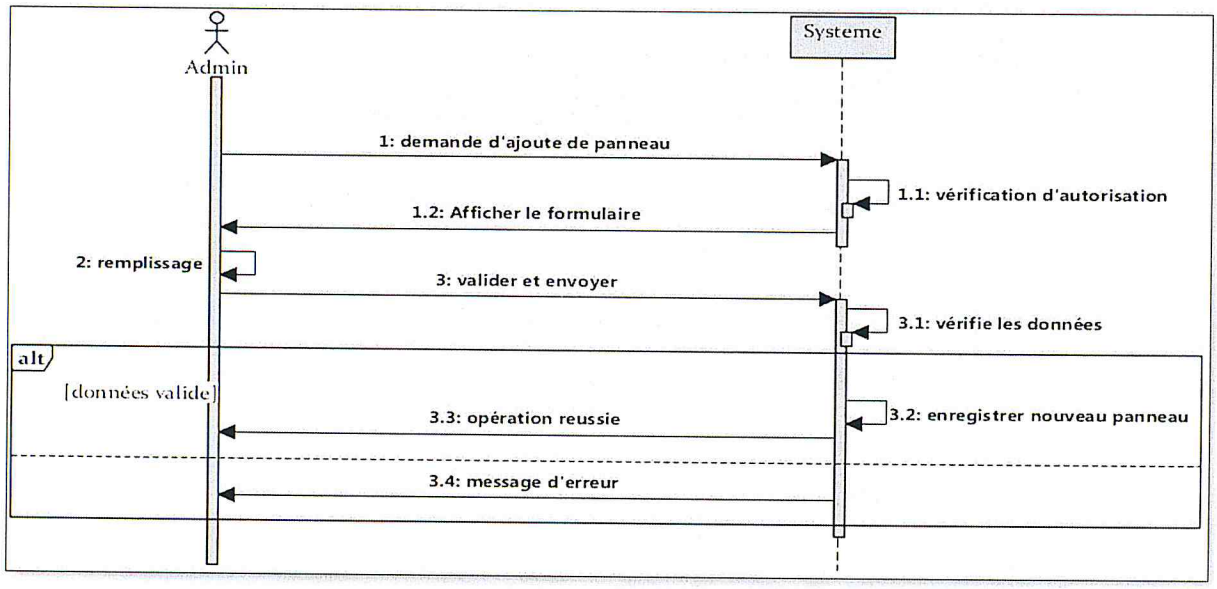



**Figure IV-18:diagramme de séquence de cas: ajouter panneau**

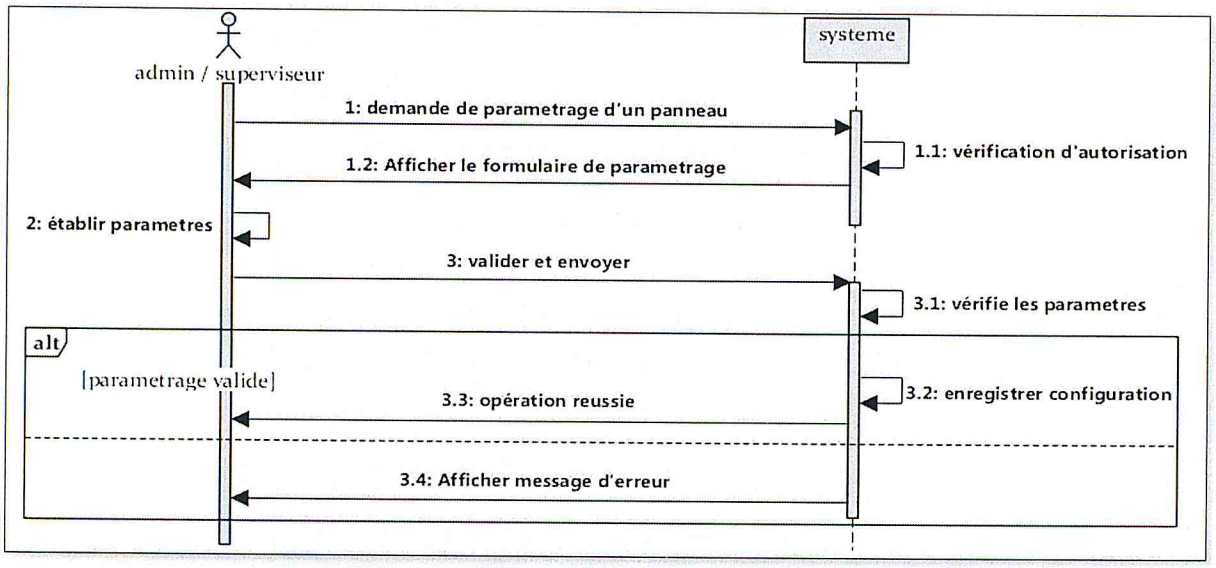



**Figure IV-19:diagramme de séquence de cas: configurer un panneau**

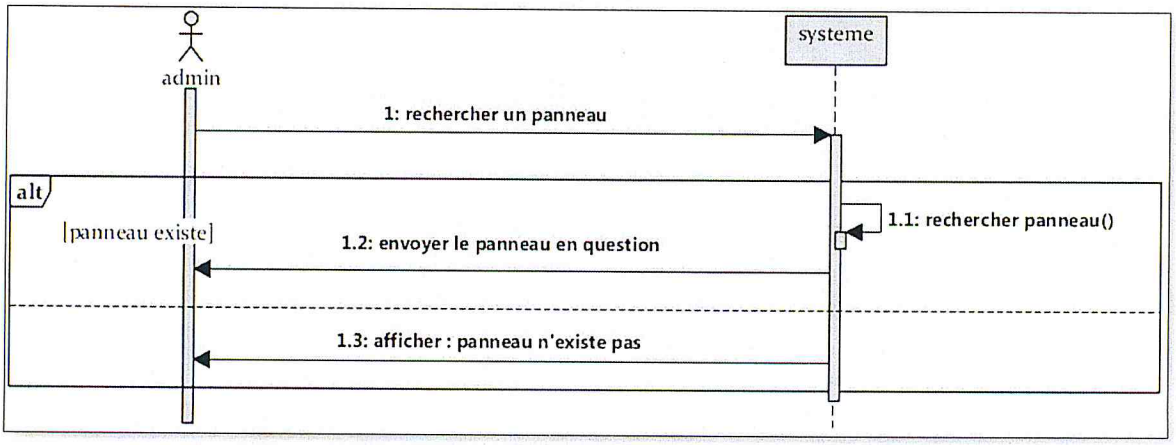



**Figure IV-20:diagramme de séquence de cas: rechercher un panneau**

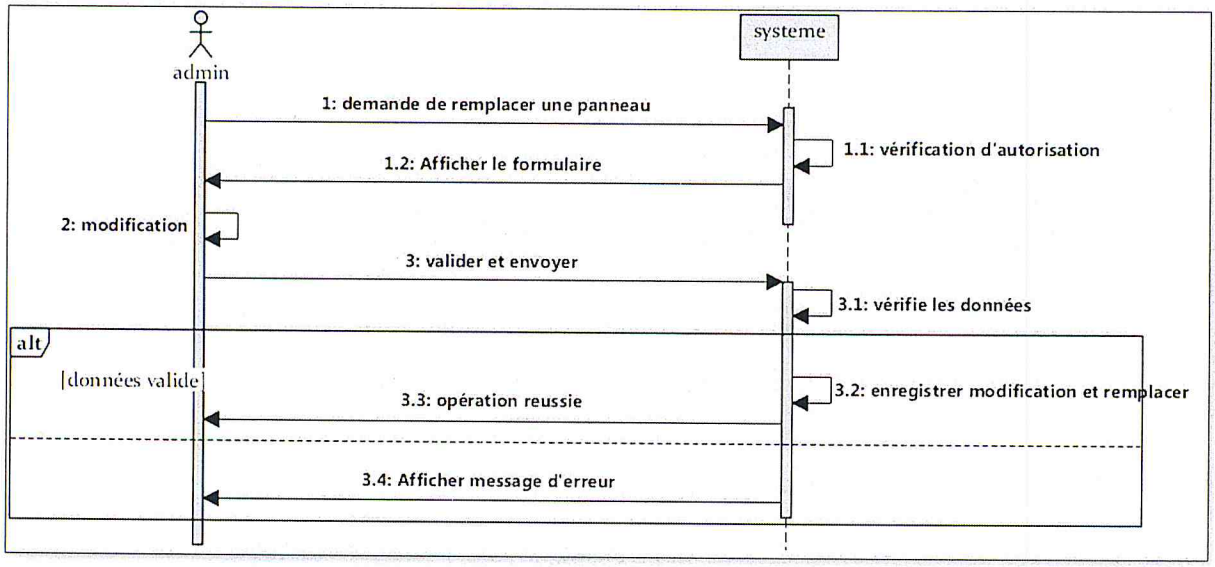



**Figure IV-21:diagramme de séquence de cas: remplacer panneau**

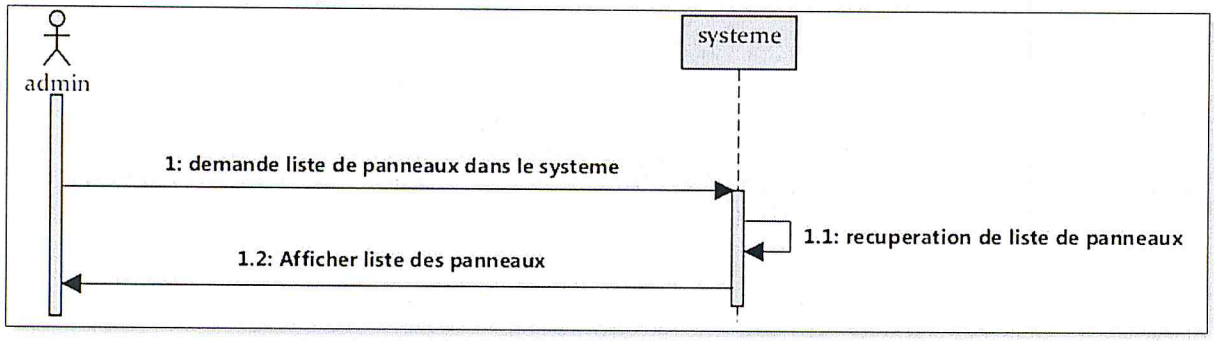



**Figure IV-22:diagramme de séquence de cas: lister panneaux**

### *4.1.6. Diagrammes de séquence du cas « Gestion des données cartographie»*

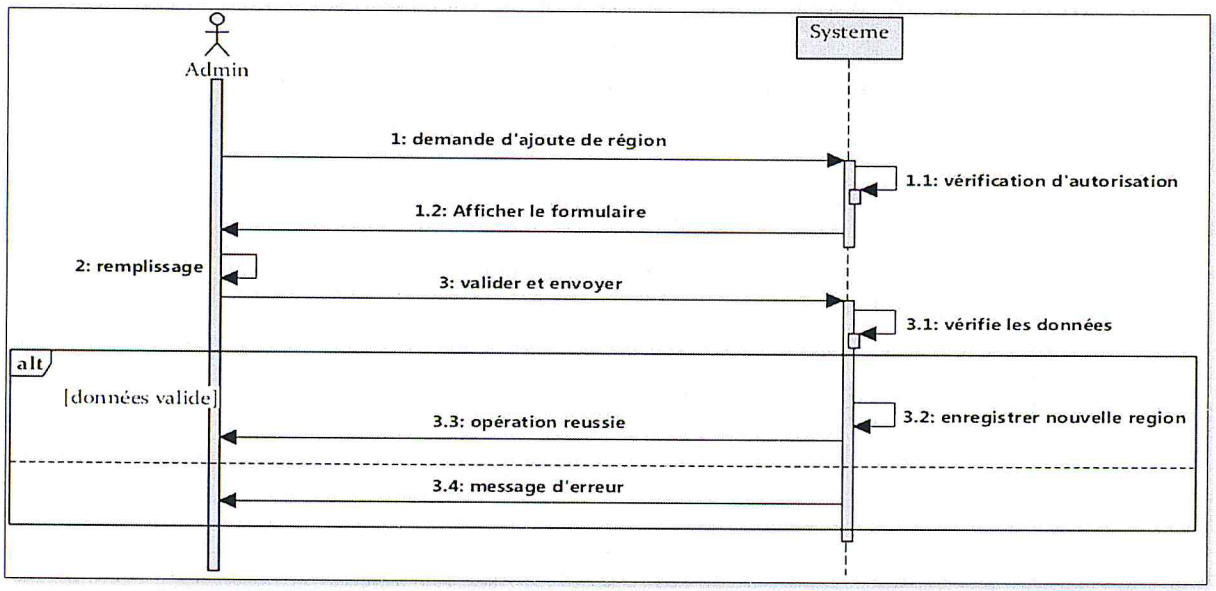



**Figure IV-23:diagramme de séquence de cas: ajouter région**

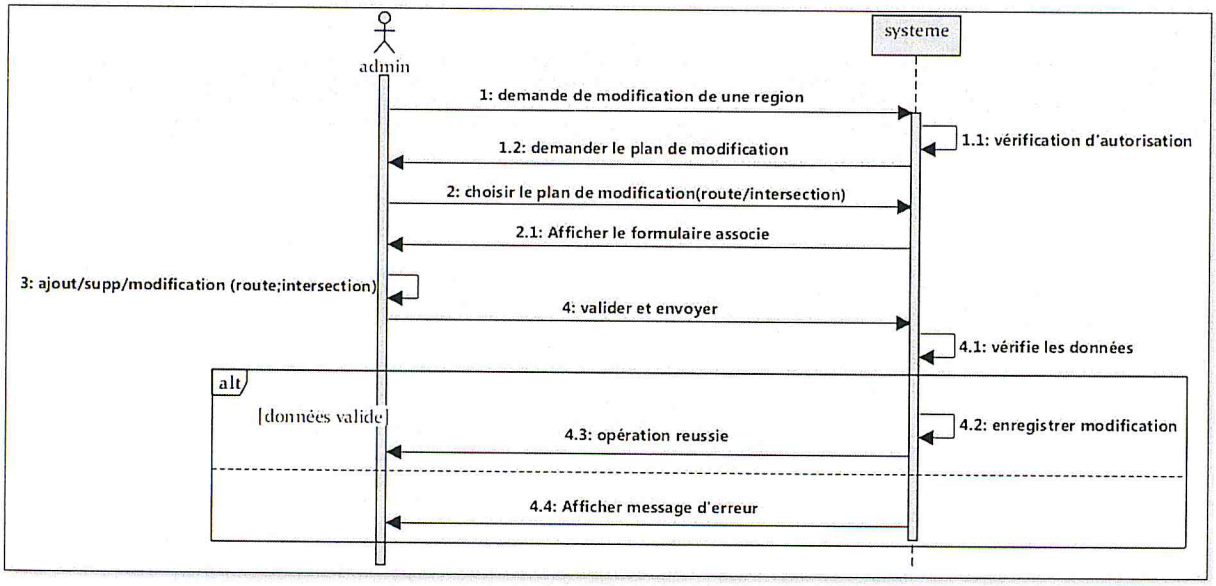



**Figure IV-24:diagramme de séquence de cas: modifier région**

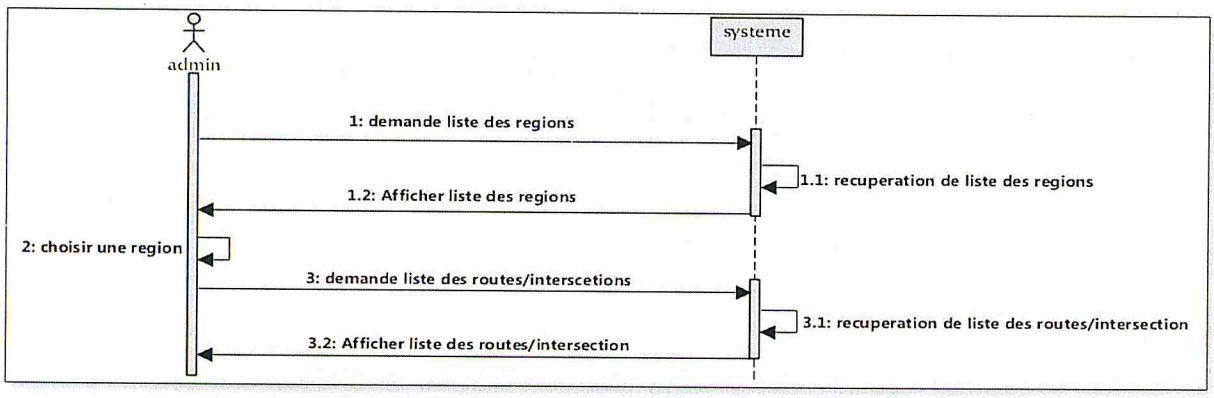



**Figure IV-25:diagramme de séquence de cas: lister régions**

### *4.1.7. Diagrammes de séquence du cas « Etablir une information routier»*

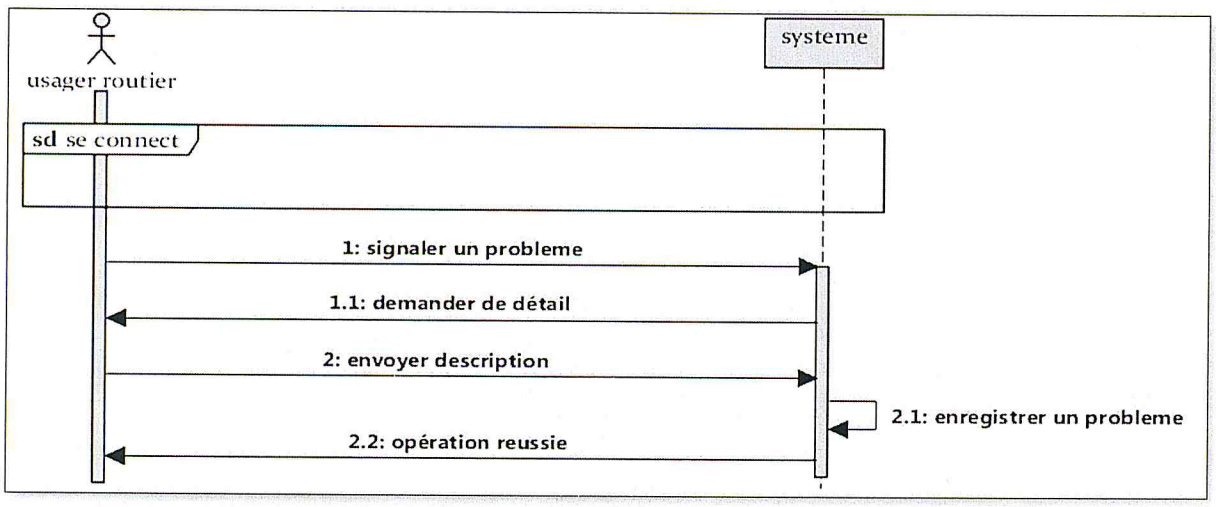



**Figure IV-26:diagramme de séquence de cas: signaler problème**

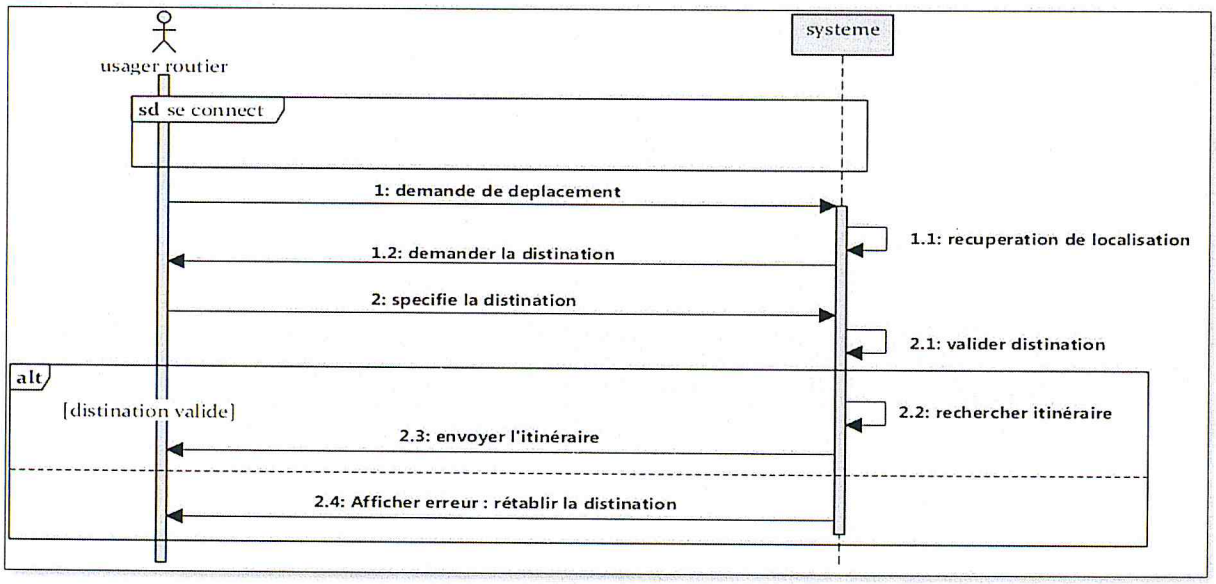



**Figure IV-27:diagramme de séquence de cas: demander déplacement**

### *4.1.8. Diagrammes de séquence du cas « Gestion de statistiques»*

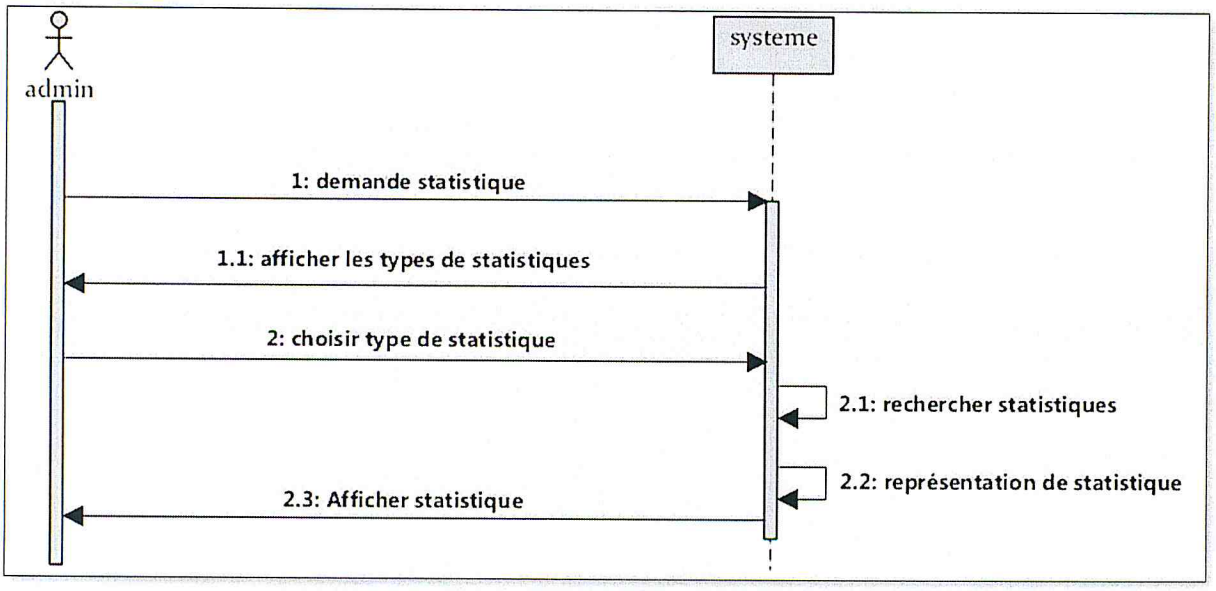



**Figure IV-28:diagramme de séquence de cas: gérer statistiques**

### 5. Aspect statique du système

Dans cette vue et a l'opposé aux diagrammes de cas d'utilisation qui décrivent le système d'un point de vue externe, nous intéressons a l'aspect structurel interne du système en utilisant le diagramme de classe. Elle est appelé une vue statique car on ne tient pas en compte le facteur temporel du système.

### 5.1. Diagramme de classe

Le diagramme de classe permet d'identifie la structure des classes d'un système, y compris les propriétés et les méthodes de chaque classe et aussi les différents relations, telles que la relation d'association et d'héritage, qui peuvent exister entre les classes, les différents class de notre système sont énumérés dans le tableau suivant :

| ***Classe*** | ***Description*** |
|---|---|
| Utilisateur | Désigner un personne utilisateur de système, Chaque utilisateur il est caractérisé par son identificateur *id*, *son nom*, son *prénom, son username et* son *password, etc.* |
| Administrateur | Désigner l'administrateur du système qui représente un cas spécial de la classe Utilisateur. |
| Usager routier | Désigner un usager routier qui représente un cas spécial de la classe Utilisateur. |
| Superviseur | Désigner un superviseur dans système qui représente un cas spécial de la classe Utilisateur. |
| Route | Modélise une route dans une région. |
| Région | Modélisé une région géographique qui se caractérise par un identificateur *id* et un *nom.* |
| Panneau Tricolore | Désigné l'outil utilisé pour gérer le trafic dans les carrefours. |
| Panneau Signalisation | Désigné l'outil utilisé pour gérer le trafic dans les routes. |
| Intersection | Modéliser les Intersections (les carrefours) dans une région. |
| VéhiculeSys | Désigné l'outil qui envoi de données utilisé pour mesurer le trafic, c'est un composant électronique installé chez l'usager routier. |
| Déplacement | Désigner un déplacement sollicité par usager routier. |
| Annotation | Désigner un événement sur une route, elle est caractérisé par une désignation de type (accident, chantier, etc.). |
| Detaille | Représente une classe associative entre Route et Annotation. |

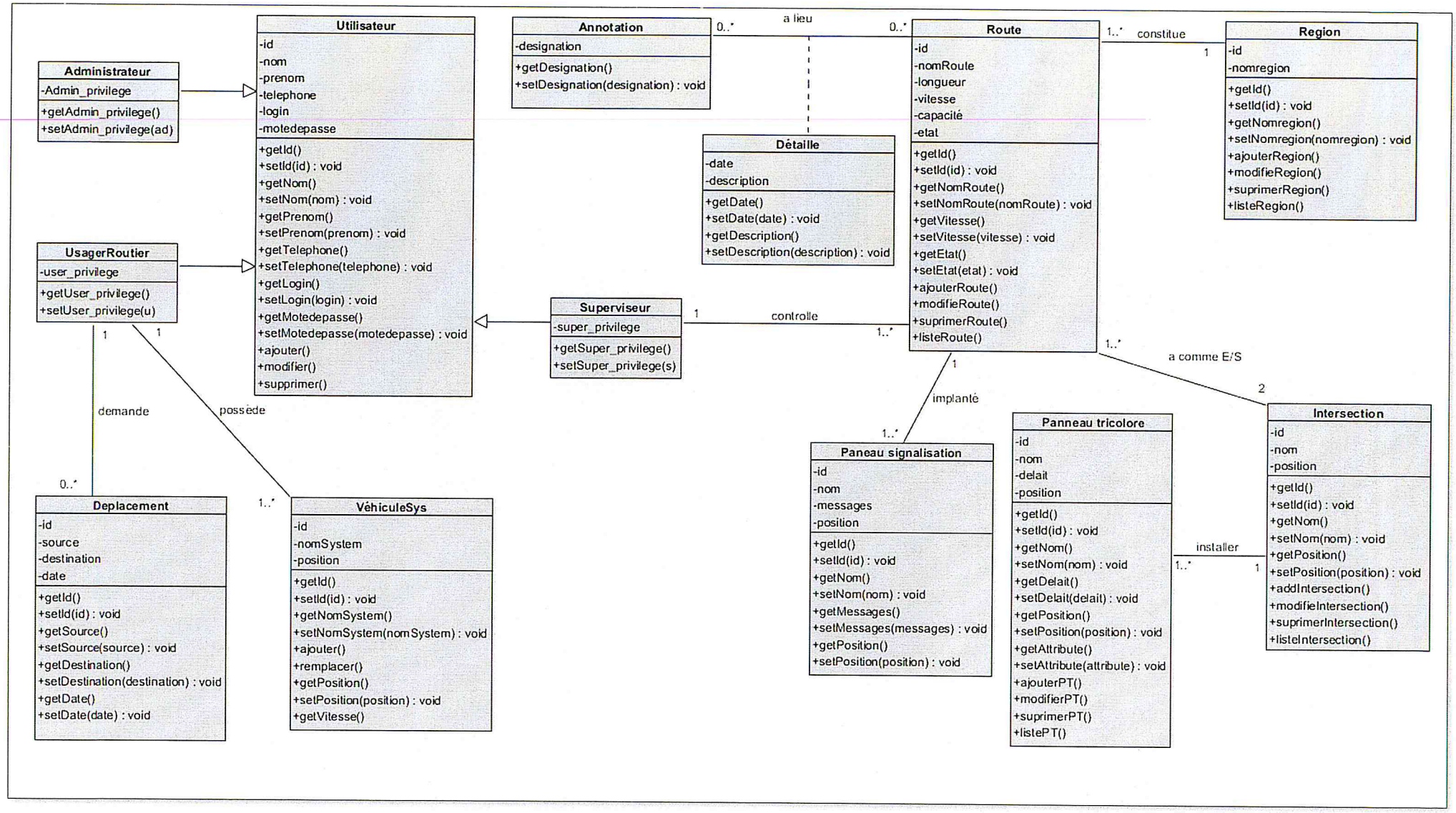



*Figure IV-29: Diagramme de classe du système*

## 6. Spécification d'un schéma de données pour le stockage des données dans une baseNoSQL orientée colonne

### 6.1. Passage au schéma de données orienté colonne

Rappelons que nous allons travaillons dans environnement big data où nous allons utiliser un SGBD NoSQL orienté colonne c'est pour quoi nous allons effectuer un passage a un schéma données de type qui n'est pas relationnel mais plutôt NoSQL orienté colonne.

Les différents concepts du modèle de données NoSQL orienté colonne sont :

**Colonne :** Une colonne est la plus petite unité du modèle de données NOC. C'est un triplet contenant un nom, une valeur et un timestamp sert à déterminer la mise à jour la plus récente (marquer de temps).

| NOM |
|---|
| VALEUR |
| TIMESTAMP |

**Figure IV-30: Description d'une colonne**

**La ligne (row):** Une ligne est composée d'un ensemble de colonnes et identifiée par une clé, Il est possible d'utiliser des colonnes comme clé primaire.

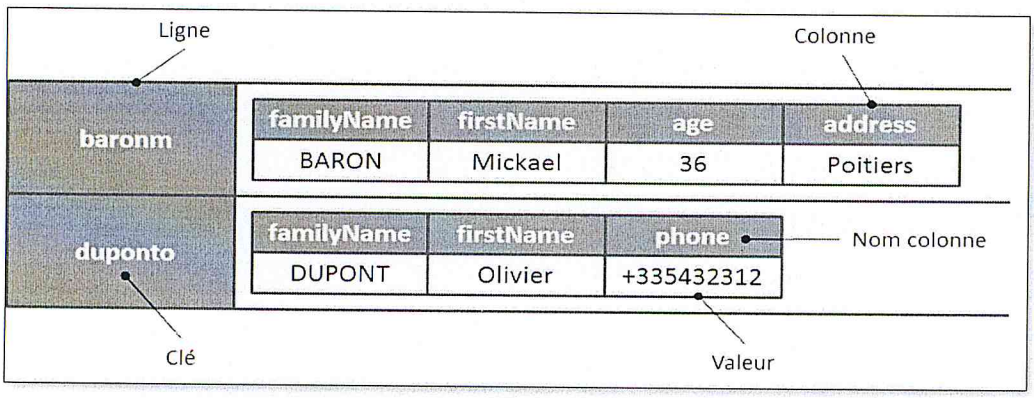



**Figure IV-31: Exemple descriptif d'une ligne**

**Famille de colonnes (Column Family) :**

Une famille de colonnes est un regroupement logique de lignes elle se ressemble a une table dans les bases de données relationnelles. on distingue deux types de famille de colonnes:

- statique : les colonnes sont définies lors de la création ou modification de la famille de colonnes,
- dynamique : les colonnes sont définies lors de la création ou modification d'une ligne.

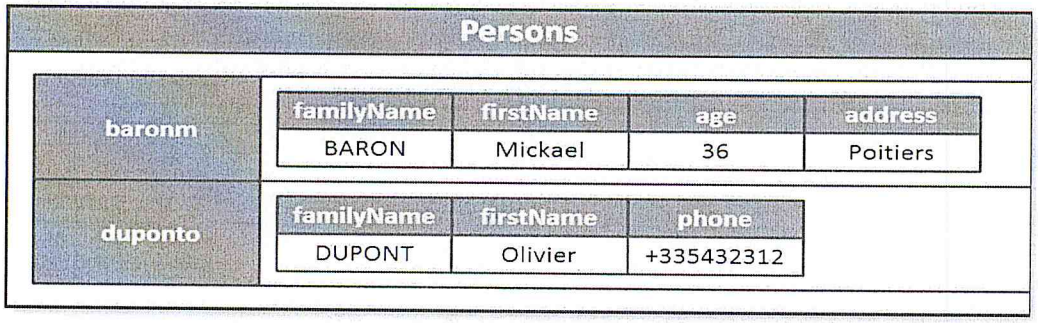



**Figure IV-32: Exemple pour décrire une famille de colonnes**

**Keyspace :**

Le keyspace est un regroupement de famille de colonnes. Il se ressemble schéma dans le monde des bases de données relationnelles.

Le model de données de BDNOC est maintenant est connue, il ne reste que faire la traduction ou une transformation à un schéma de BDNOC.La traduction consiste a présenté les équivalents des concepts du diagramme de classe en commençant par déterminer l'équivalent en BDNOC d'une classe et ensuite de chaque type d'associations notamment un à un, plusieurs à un et enfin plusieurs à plusieurs.

### *6.1.1. Traduction d'une classe*

La règle est simple, une classe aura pour équivalent une famille de colonnes du même nom. Et chaque attribut de la classe donnera lieu à une colonne (super ou simple) selon les besoins et les spécificités de l'attribut. L'exemple ci-dessous illustre ce principe.

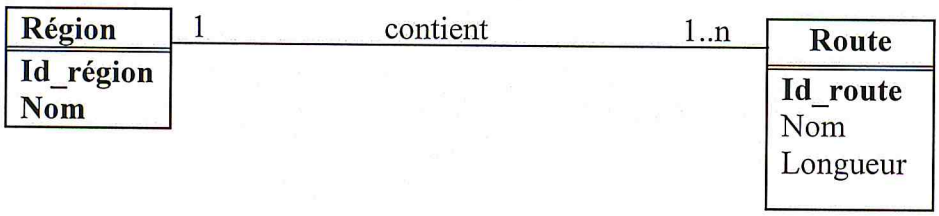
| **Route** |
|---|
| Id_route |
| Nom |
| Longueur |
| Vitesse |
| Capacité |
| Etat |

| **ColumnFamily : Route** | | | | | | |
|---|---|---|---|---|---|---|
| **Key** | **Valeur** | | | | | |
| **Id_route** | Colonne | Nom | Longueur | Vitesse | Capacité | Etat |
| | Valeur | | | | | |

La traduction des classes en leur équivalent BDNOC étant désormais effectuée,il faut maintenant aborder la traduction des différents types de d'associations.

### *6.1.2. Traduction d'une association de plusieurs à un*

Soit l'exemple suivant :




L'équivalent du schéma ci-dessus en BDNOC doit pouvoir pour une Région donné, lister les routes qui la contiennent. Soit :

| | |
|---|---|
| région x | **Route id …Route id12**<br>**Route id31Route id …** |
| région y | **Route id …Route id 80Route id …** |
| ⋮ | ⋮ ⋮ |

Le modèle de données orienté colonne équivalent à la situation de la relation un à plusieurs se présentecomme suit :

| **ColumnFamily : Routes par Région** | | | | |
|---|---|---|---|---|
| **Key** | **Valeur** | | | |
| **Id_région** | Colonne | Id_route | …….. | Id_route |
| | Valeur | | …….. | |

Donc la règle est la suivante : une association de type un à plusieurs se traduit en BDNOC en une famille de colonne unique qui sera constitué des attributs d'entité maître comme colonne simple et une Super colonne qui regroupe les attributs de l'entité esclave. Si l'association entre les deux entités a des attributs, ils sont traduits en colonnes dans la famille de colonne en BDNOC.

### *6.1.3. Traduction d'une association plusieurs à plusieurs*

Une association de type plusieurs à plusieurs, se traduit en BDNOC en deux familles de colonne. Un type d'association plusieurs à plusieurs est perçu comme une association symétrique de un à plusieurs. On applique donc la traduction de un à plusieurs dans les deux sens. C'est à dire soient les entités A et B qui donneront lieu ainsi que nous l'avons dit précédemment, à deux familles de colonnes FCA et FCB dont les colonnes seront respectivement les attributs des entités A et B. Aussi, il sera ajouté à chacune des familles de colonnes FCA et FCB, un super colonne SCB respectivement SCA qui contiendra la liste des objets de la famille de colonne FCB respectivement FCA.

### *6.1.4. Traduction d'une association un à un*

En base de données, une relation de un à unsignifie que pour chaque enregistrement d'une table correspond un et seul enregistrement de l'autre table qui lui soit lié et réciproquement. Une association de type un à un étant une relation d'égale à égale entre entité, dont la traduction elle donne lieu à une famille de colonne dans le modèle BDNOC, dont les colonnes sont l'ensemble des attributs des deux entités qui participent à l'association. Donc on regroupe dans la même famille de colonne les attribues de deux entités qui soit utiles.

En fin il faut noter que dans les bases de données NoSQL il n'y a pas de jointeur, ni de normalisation, en monde NoSQL le schéma de données est dé-normalisé et quasi impossible de le normalisé et le schéma de données est modélisé selon les exigences de l'application.

## 6.2. Schéma de données NoSQL orienté colonne du système

En appliquant les différentes règles de traduction établies plus haut on obtient le modèle de données équivalent en BDNOC.

Le schéma de donnéesest constitué de 10 familles de colonnes statiques à savoir : *Utilisateur, Route, Région, VES, Annotation, Intersection, Déplacement, Panneau, Routes_par_région, Panneaux_par_route.*

Et trois familles de colonnes dynamiques qui sont : *Trafics par Route, Annotations par Route, Déplacements par Utilisateur.*

**La famille de colonnes Utilisateur :**

| **ColumnFamily : Utilisateur** | |
|---|---|
| **Clé** | **Valeur** |
| userID | Colonne \| Nom \| Prénom \| Téléphone \| Login \| Password<br>Valeur \| | | | | | |

**La famille de colonnes Route :**

| **ColumnFamily : Route** | |
|---|---|
| **Clé** | **Valeur** |
| routeID | Colonne \| Nom \| Longueur \| Vitesse \| Capacité \| Etat<br>Valeur \| | | | | |

**La famille de colonnes Déplacement:**

| **ColumnFamily : Déplacement** | |
|---|---|
| **Clé** | **Valeur** |
| déplacementID | Colonne \| Source \| Destination \| date<br>Valeur \| | | |

**La famille de colonnes Région :**

| **ColumnFamily : Région** | |
|---|---|
| **Clé** | **Valeur** |
| régionID | Colonne \| Nom<br>Valeur \| |

**La famille de colonnes VES:**

| **ColumnFamily : VES** | |
|---|---|
| **Clé** | **Valeur** |
| VES_Id | Colonne \| Nom<br>Valeur \| |

**La famille de colonnes Annotation :**

| **ColumnFamily : Annotation** | | | | |
|---|---|---|---|---|
| **Clé** | **Valeur** | | | |
| annotationID | Colonne | désignation | description | Time |
| | Valeur | | | |

**La famille de colonnes Intersection :**

| **ColumnFamily : Intersection** | | | |
|---|---|---|---|
| **Clé** | **Valeur** | | |
| intersectionID | Colonne | Nom | Nbr_sortie |
| | Valeur | | |

**La famille de colonnes Panneau :**

| **ColumnFamily : Panneau** | | | |
|---|---|---|---|
| **Clé** | **Valeur** | | |
| panneauID | Colonne | Type | Position |
| | Valeur | | |

**ColumnFamily Routes par Région :**

| **ColumnFamily : Routes par Région** | | | | |
|---|---|---|---|---|
| **Clé** | **Valeur** | | | |
| régionID | Colonne | routeID | ..... | routeID |
| | Valeur | | ..... | |

**ColumnFamily Panneaux par Route :**

| **ColumnFamily : Routes par Région** | | | | |
|---|---|---|---|---|
| **Clé** | **Valeur** | | | |
| routeID | Colonne | panneauID | ..... | panneauID |
| | Valeur | message | ..... | message |

**La famille de colonnes Trafic par Route :**

| **Clé** | **Valeur** |
|---|---|
| routeID | Colonne: time, time, ..... / Valeur: tauxTrafic, tauxTrafic, ..... |

**La famille de colonnes Annotations par Route :**

| **Clé** | **Valeur** |
|---|---|
| routeID | Colonne: désignation, désignation, ..... / Valeur: time, time, ..... |

**La famille de colonnes Déplacements par Utilisateur :**

| **Clé** | **Valeur** |
|---|---|
| userID | Colonne: déplacementID, déplacementID, ..... / Valeur: time, time, ..... |

### 7. Modélisation de sous système manipulateur d'événements

Nous abordons maintenant l'aspect de traitement d'événement, L'approche fait référence a un prototype EDA pour détecté les embouteillages, les différents événements produit dans le système sont des informations sur les routes qui doit être transformé en événements plus abstrait et sophistiquer pour évaluer l'état de trafic actuel et mise en place des actions.

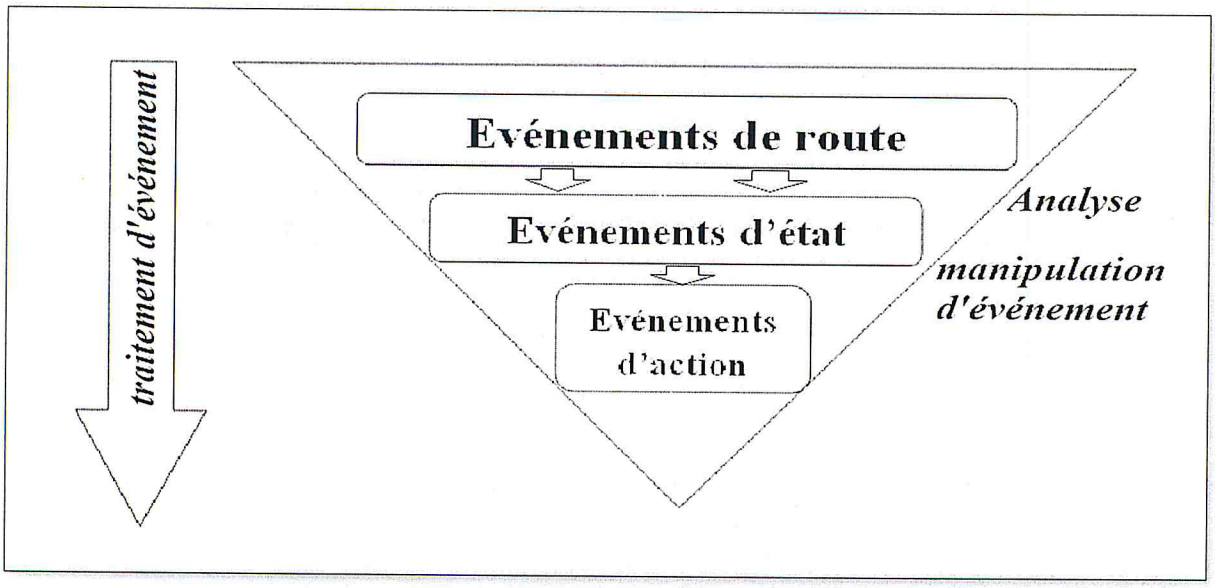



**Figure IV-33:l'hiérarchie générale d'événement**

### 7.1. Modélisation structurelle d'événements

Les événements sont un concept clef pour un système CEP et doit être définie formellement pour comprendre les différents types d'événements, leurs propriétés, contraintes et dépendances. L'hiérarchie d'événement reflète directement sur la séquence de traitement d'événement

- ***Evénements de route:*** les événements d'entrés sont des informations pour évaluer l'état de trafic de chaque route.
- ***Evénements d'état:*** dans cette étape les événements de route sont synthétisés dans événements d'états qui vont caractériser un état réel dans l'environnement.
- ***Evénements d'action:*** on se basant sur les événements de situation (état), un plan d'action est nécessaire pour définir les actions approprie pour certain situations.

### 7.2. Spécification d'événements

Le modèle structurelle d'événements définie en haut est raffiné et ajusté pour la gestion de trafic en spécifiant les classes d'événement.

- *Evénements de route*: l'événement de route est spécifie par l'événement de trafic dont le contenu est données d'une route, plus précisément le taux d'occupation et la vitesse moyenne.
- *Evénements d'état*: on appliquant sur les événements de trafic des motifs d'événements (*event pattern*) pour identifier la situation actuelle de trafic, l'événement d'état est spécifie par l'événement de congestion.
- *Evénements d'action*: enfin des événements d'action ont lieu selon les événements d'état, événement d'action est un événement d'alerte.

Ainsi on obtient un schéma ou un modèle structurelle d'événements du système de gestion de trafic routier suivant:

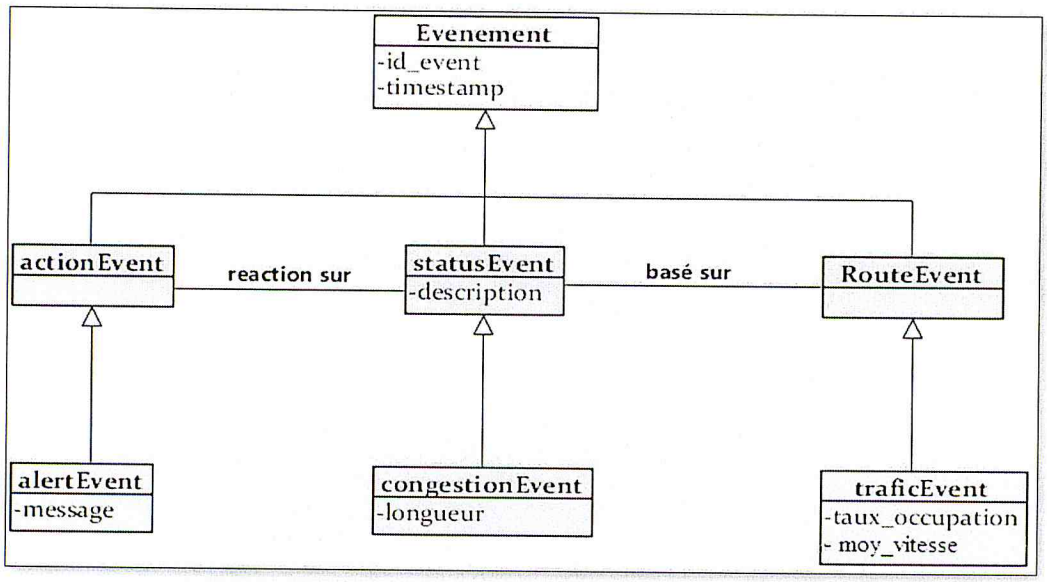



**Figure IV-34:modèle structurel d'événement du système**

### 7.3. Le réseau de traitement d'événement (EPN)

L'hiérarchie générale d'événement définie dans la modèle structurel d'événement a une correspondance avec une séquence d'étape de traitement d'événement, ce dernier est n'est que de transformation d'événement, une étape de transformations est réalisé par un agent de traitement d'événement (EPA), qui est connecté a un réseau de traitement d'événement. Donc chaque EPA applique certaine traitement à un niveau d'abstraction d'événement donné.

Les objectifs de chaque EPA sont listés comme suit :

- ***Agent de Diagnostique de congestion (ADC) :*** l'agent de diagnostique de congestion lit de manière continue les flux d'événements de route, exécute et applique des motifs d'événement sur eux, quand ADC détecte un taux d'occupation élevé et vitesse moyenne bas dans une fenêtre de temps il génère événements de congestion. Par résultat un nouveau flux d'événements.
- ***Agent de Planification d'Action (APA) :*** c'est l'agent de réaction, prend en entrée événements de congestion et génère des événements d'alerte comme actions pour réagir sur l'état actuel de trafic, plus précisément, une congestion.

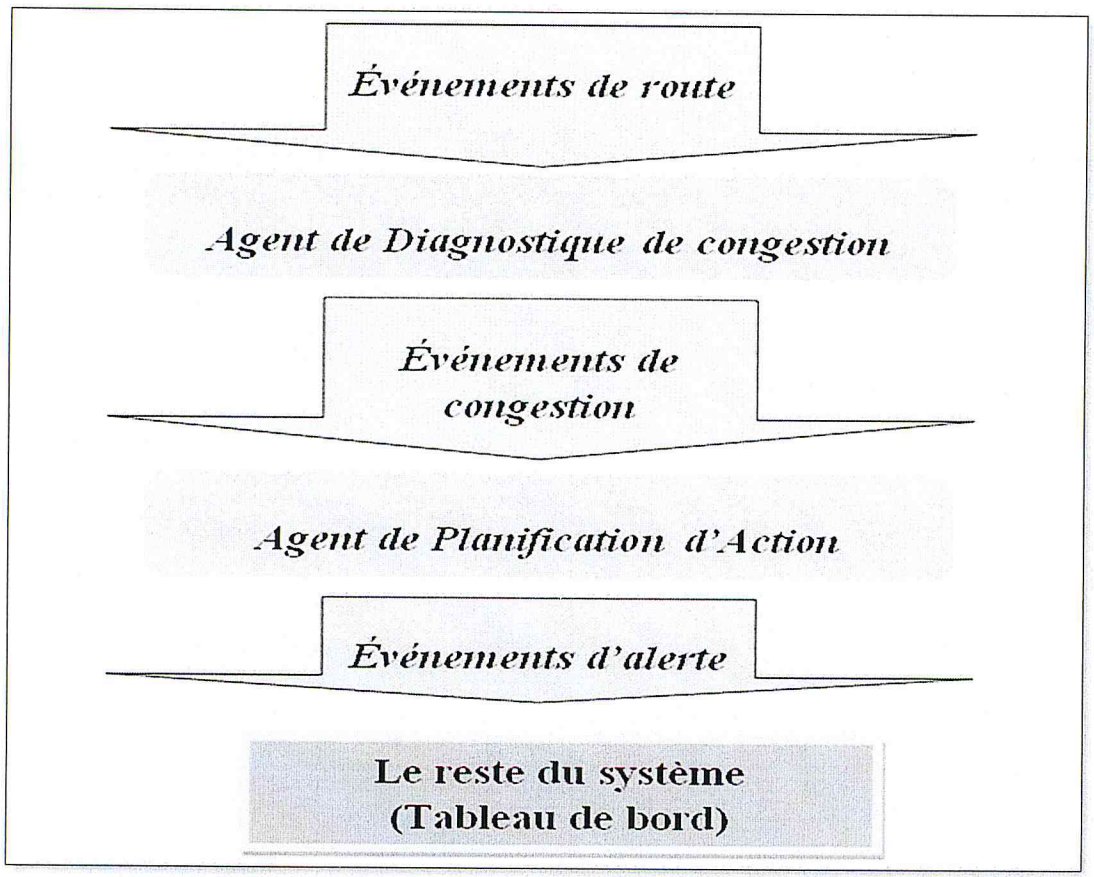



**Figure IV-35: le schéma EPN du système**

### 8. Architecture général du système

Après avoir clôturé la partie conceptuelle et la revue des technologies de IoT, big data et CEP, l'architecture du système adopté est basé sur la coopération entre plateforme big data de traitement et stockage de données et moteur CEP dans la façon suivante :

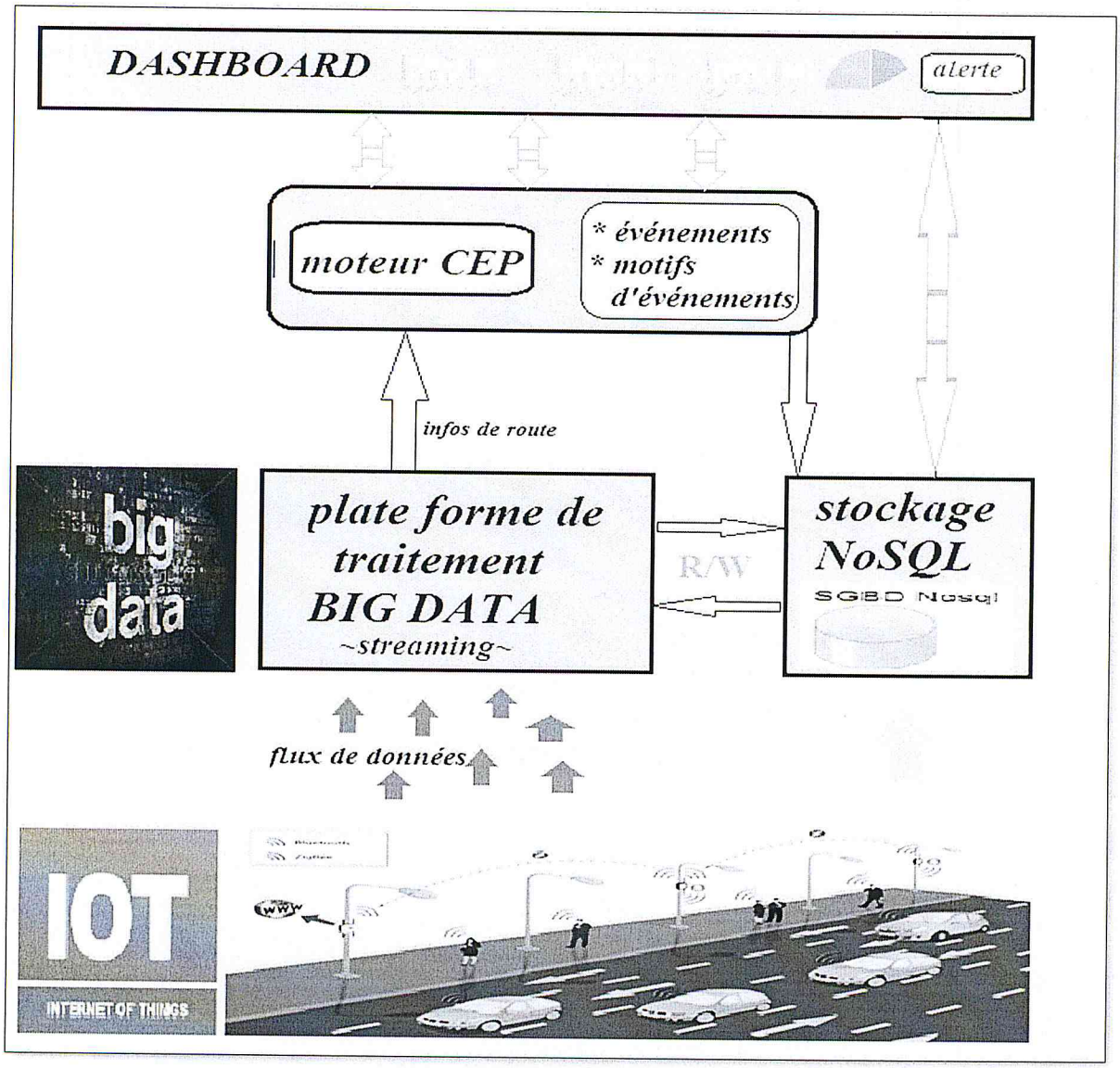



**Figure IV-36: Architecture générale du système**

Les données générées par les composants électroniques installés sur les véhicules sont traité en mode streaming par une plateforme Big Data d'analyse de données, par la suite le résultat est transmit a un moteur CEP pour détecté des motifs sur notamment l'état de trafic routier et définir l'action pour réagir sur le contexte.

### 9. Le plan de routage sur un réseau routier

Sur le plan de routage la solution que nous avons opté est basé sur certains algorithmes bien connus de la théorie des graphes, ces algorithmes constituent en général les fonctionnalités de routage sur un réseau routier notamment pour le calcul de l'itinéraire optimal entre deux lieux Parmi les algorithmes de routage ou encore fonctionnalités deroutage, on peut citer :

- Plus court chemin Dijkstra : c'est un algorithme de routage sans heuristique,
- Plus court chemin A-Étoile : routage pour grand un ensemble de données,
- Plus court chemin A-Étoile utilisant deux directions,
- Plus court chemin Shooting-Étoile : routage prenant en compte le sens giratoire.

**Algorithme de Dijkstra :**

Dans théorie des graphes, Dijkstra a proposé un algorithme permet de résoudre leproblème du plus court chemin. Cet algorithme permet de déterminer le plus courtchemin pour se rendre d'un point à un autre connaissant le réseau routier d'un endroit donné. L'algorithme de Dijkstra à un graphe connexe dont le poids affecté àchaque arête est un nombre positif.
Dans la théorie de la Complexité, l'algorithme de Dijkstra est un algorithme polynomial et ne nécessite pas développement des heuristiques.

**Algorithme A-Étoile :**

L'algorithme A-Étoile est aussi un algorithme permet de résoudre leproblème du plus court chemin, il est basé sur les sommets. Cet algorithme ajoutel'information sur la position du début et de la fin de chaque tronçon, Cela permet une recherche privilégiant les tronçons proches du point d'arrivée de la recherche. A* apport de rapidité par rapport a dijkstra quant le graphe devient grande de fait qu'il est basé sur les heuristiques.

**Algorithme Shooting-Étoile :**

L'algorithme du plus court chemin Shooting-Étoile est basé sur les arrêtes. Cetalgorithme est spécialisé dans la recherche d'un parcours d'un tronçon à un autreetnon d'un sommet à un sommet comme c'est le cas avec les algorithmes de Dijkstra etA-Étoile, ainsi Shooting-Étoile permet prise en charge de plusieurs paramètres dans la recherche de chemin optimal.il aussi basé sur les heuristiques.

En effectuant une comparaison entre ces solutions, L'algorithme de Dijkstra est très puissant, même si il admit quelques limites notamment quand le graphe devient grand et non prise en compte des poids négatif (qui n'a aucune sens dans notre cas de fait que les routes vont toujours avoir des poids positive), il trouve toujours le chemin ayant le poids le plus faible ainsi que la complexité est polynomial qui n'est pas le cas avec le A-Étoile qui présente l'avantage de rapidité et Shooting-Étoile qui permet de prendre plusieurs paramètre, ces deux dernier nécessite développement des heuristiques et ils ne donnent pas toujours la solutions exact.

La solution retenue est basé sur l'algorithme de dijkstra de fait que est une solution du plus court chemin sans heuristiques, les contraintes de notre cas ce que on ne recherche pas juste le plus court itinéraire mais aussi optimal en terme de nombre de vihécules et la vitesse de l'écoulement de trafic, et aussi que nous travaillons dans un environnement distribué ce qui implique le parallélisme de traitement.

| Vocabulaire des graphes | Vocabulaire routier |
|---|---|
| *Graphe (pondéré)* | *Réseau routier* |
| *Sommet ou nœud* | *Intersection de routes* |
| *Arête* | *Route* |
| *Poids* | *Distance, la vitesse moyenne et le taux d'occupation* |
| *Chemin le plus court* | *Itinéraire le plus fiable ou rapide* |

**Tableau IV-3: Vocabulaire routier vis à vis graphes**

### 9.1. Déduction de poids a affecté à la route

Comme c'est évoqué en haut dans l'algorithme de plus courts chemin, un seul paramètre est affecté au Arête, dans notre contexte ce paramètre sera le temps de parcours chaque route, par la suite le chemin retourné c'est le plus rapide.

Ce paramètre représente la relation entre les trois pointes :

- La longueur de la route, (L)
- Nombre de véhicules dans la route, (N)
- La vitesse moyenne de la route. (V)

La vitesse moyenne : $V = \frac{\sum_{i=0}^{n} Vi}{N}$ $V_i$ :représente la vitesse de véhicule i

Le temps de parcours c'est la distance par rapport la vitesse, soit :$T = \frac{L}{V}$

Enfin la valeur retourné c'est le temps de parcours, qui sera affecté à la route au lieu de la distance

Algorithme de dijkstra est auto-suffisant et idéal s'il sera déroulé et implémenté sur une seule machine, cependant la deuxième contrainte exige le parallélisme.

La nouvelle version de l'algorithme pour le traitement de plus fiable chemin sera basée sur le paradigme de MapReduce qui permet traité de grandes données d'une façon parallèle et distribués

### 9.2. La solution de routage retenue

La solution retenue est un algorithme basé sur l'algorithme Dijkstra, Pour calculer le chemin plus rapide dans un réseau routier nous utiliser un algorithme itératif. Notre approche estbasée sur le modèle de programmation MapRedure pour le traitement de grands ensembles de données avec un algorithme parallèle.

L'algorithme procède comme suit :

***1. initialisation :*** pour chaque nœud de graphe sauf le nœud de départ,

$\forall s \in S$ $T[i] = \infty$ , $T[s0]=0$

***2. Itérations :*** A chaque itération, l'algorithme calcule pour chaque nœud s ∈ Sle poids et listé ses voisins puis gardé le poids le plus petit. La technique MapReduce est utilisée pour trouver le poids le plus petit de chaque nœud de réseau. Ainsi, nous parallélisons la phasede recherche sur les différentes Map pour trouver les différents poids de chaque nœud. Le reducer permetde retourné le poids le plus petit de nœud parmi les autres.

a. *La fonction Map:*

La fonction Map est appliquée en parallèle à chaque nœud dans l'ensemble de graphe d'entrée. Chaque Map reçoit en entrée un nœud. Pour chaque nœud elle calcule le poids et ses voisines dans le graphe en se basant soit sur la matrice d'adjacente ou sur la liste d'adjacents. Cela produitune liste de paires pour chaque appel. La Map générée est sous la forme suivante : < nœud , < poids ,liste_voisin (nœud , temps) >>

Le but de map est de modéliser les données en < key, value > paires et au reducer de regrouper les paires émises < Key, value >. Cequi signifie essentiellement que, les paires avec la même clé sont regroupées et transmises a une seule machine, qui ensuite exécutera lereducer sur eux.

b. *la fonction reduce :*

La fonction Reduce traite toutes les temps de parcours possiblesvers un nœud donné et sélectionne le plus petit.

***3. Arrêt :***L'algorithme s'arrête lorsqu'il le temps de parcours le plus petit à chaque nœud ne change plus.

### *9.2.1. ALGORITHME MAP/REDUCE DE PLUS FIABLE CHEMIN*

**Entrée :**

un graphe G, N_départ et N_arrivé deux sommets de G.

**Représentation de données :**

**Clé :** nœud n

**Valeur :** T, liste d'adjacente

**Initialisation :**

Pour chaque Nœud :T[i] ← ∞

T[n0] ← 0

**Tant que** (changement = 'oui') **faire appliqué** MapReduce

La fonction Mapp se présente comme suit :

**Procédure Map (n_id n, nœud N)**

T← N.temps

émet (n_id, N)

**pourchaque**nœud_id m ∈N.liste_adjacente

**faire :** émet (m_id, T[m]+ temps(n,m))

**fait ;**

La fonction Reduce se présente comme suit :

**Procédure Reduce (n_id m, [T1, T2, …])**

$T_{min}$← ∞

**pourchaque** T ∈ensemble **[T1, T2,…]** /*les temps possibles vers un nœud m */

**faire :Si** $T < T_{min}$

**alors** $T_{min}$ ← T

**fsi ;**

M.temps ← $T_{min}$

émet (n_id m, Nœud M)

**fait ;**

## 10. Conclusion

Dans ce chapitre, nous avons présenté la conception de notre système à travers les différents diagrammes UML, ainsi que la modélisation de sous système événementiel, et nous avons finie par illustration de l'architecture générale du système ainsi que la solution adopté pour le routage de client.

# v. CHAPITRE V : ASPECTS IMPLEMENTATION

## 1. Introduction

Après une profonde dans modélisation de notre système nous arrivons dans ce chapitre a effectué la mise en ouvre du système, L'implémentation de l'application nécessite des langages et outils technologiques capable de répondre aux nos exigences notamment l'analyse en temps réel des grandes données.

## 2. Langages de programmation utilisée

### 2.1. Java

Java est un langage de programmation orienté objet développé par Sun Microsystems, Java est à la fois un Langage de programmation et un environnement d 'exécution. Il est très performant et très utilisé par un grand nombre de développeurs grâce à plusieurs caractéristiques parmi les quelle on cite :

- **Portabilité** : Permet de crée des applications compatible avec des nombreux systèmes d'exploitation grâce a JVM « java virtuelle machine », java donne aussi la possibilité de développer des applications sur mobile, desktop et sur le web.
- **Sécurisé** : La sécurité fait partie intégrante du système d'exécution et du compilateur. Un programme Java planté ne menace pas le système d'exploitation.
- **Libre** : java est open source et avec pleine de documentations, ainsi que la JVM est gratuite.

### 2.2. Java Entreprise Edition (JEE)

Java EE est une plate forme de développement et de déploiement des applications distribuer, la plateforme JEE est constitue des services, d'interfaces de programmation d'application et des protocoles qui fournissant les fonctionnalités nécessaire de développement d'application distribué.

### 2.3. Scala

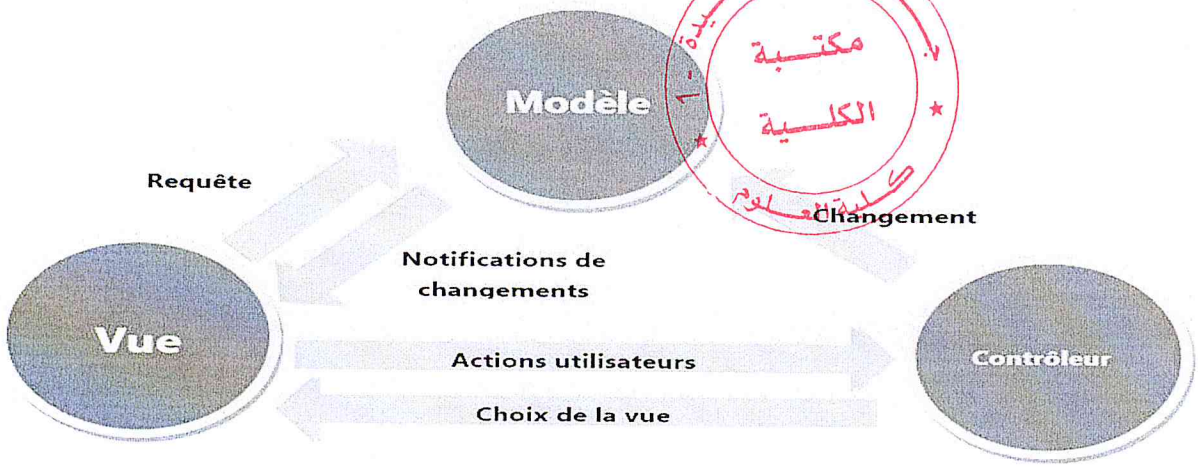
Scala est un langage de programmation qui intègre les paradigmes de programmation orienté objets et de programmation fonctionnelle. Scala est proche de Java et facilement s'interfacer avec un code Java, il peut être compilé en java bytecode exécutable sur la JVM.

### 2.4. HTML5

Html5 est la dernière version de langage html, il nous permettons le développement des applications web moderne.

### 4. Type Architecteur de l'application

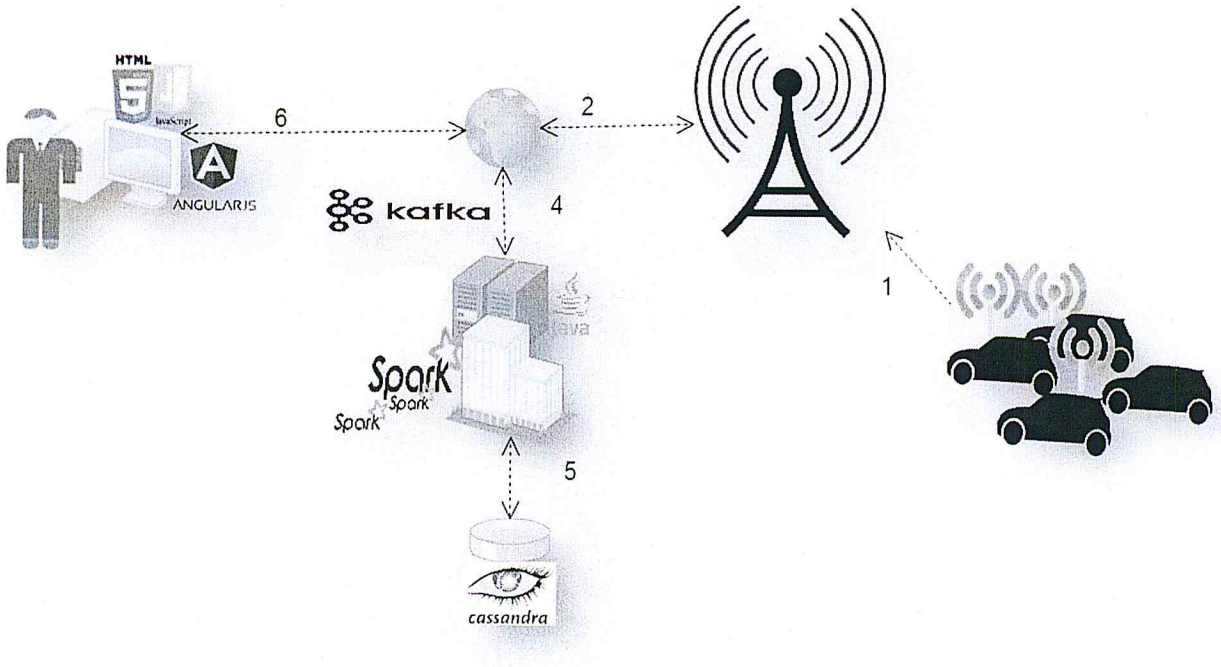
Notre prototype applicatif est baser sur Le modèle MVC (*Model-View-Controller*), ce dernier est un modèle destiné à répondre aux besoins des applications interactives en séparant les problématiques liées aux différents composants au sein d'une application, l'idée de l'architecture MVC est de bien séparer les données, la présentation et les traitements.




مكتبة الكلية

**Figure V-1: le modèle MVC**

Afin de mettre en œuvre notre système, et répondre au contrainte temps réel et stockage, nous avonsbesoin d'utiliser plusieurs technologiesqui sont illustrées dans la figure :




**FigureV-2:les technologies utilisé pour réalisé le prototype**

### 5. Simulation de partie IoT du système

Pour simulé la partie physique de l'infrastructure routier nous avons opté a un simulateur. Ce simulateur contient une carte cartographie qui représente une région ou bien une zone géographiqueconstitue des routes où des véhicules circulent, et des intersections contenant un feu pour gérer le carrefour.

Ce simulateur est un projet open source sous licence **MIT**, qui a été développer par**Artem Volkhin,** le projet est téléchargeable depuis le site Git HUB.Le simulateur à subir a des modifications de code source pour satisfait nous exigences.

Sur le plan de choix de simulateur a été difficile a trouvé un simulateur idéal puisque la majorité des simulateurs ne sont pas accessible en terme de modification de code source et par fois aussi en terme de licence.

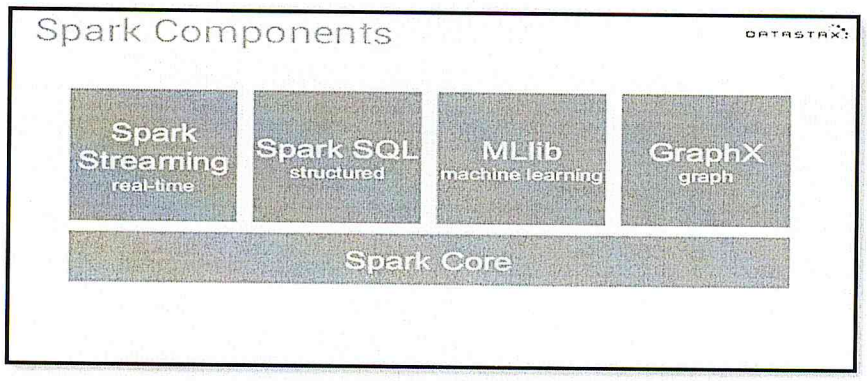
### 6. Les outils utilisés

Dans la réalisation de notre travaille on distingue deux vue des technologies utilisé, la premier fait référence a l'utilisation des outils technologiques très évolue permettant d'analysé et traité des masse de données et en temps réel, la deuxième vue représente l'utilisation des outils de développement agréable nous a facilité l'implémentation du prototype.

### 6.1. Outils technologiques de traitement

#### *6.1.1. Apache Spark*

Apache Spark est une plateforme de traitement distribué répartie sur plusieurs nœuds, il est open source et très puissante basé principalement sur le traitement IN MEMORY pour atteindre une rapidité de traitement, la plateforme Spark est constitué de composants illustré dans la figure suivante :




**FigureV-3:Les composants d'apache spark**

### *6.1.2. Apache Cassandra*

Apache Cassandra est une base de données No SQLorienté colonne distribuée qui permet de stocker et manipuler une grande quantité de données grâce à sa sociabilité horizontale, basé sur deux concepts de base de données existantes, Le première Bitable de Google, pour son modèle de données orienté colonne et le seconde Dynamo, créé par Amazon, pour son architecture distribuée sans nœud maître. Elle possède les caractéristiques suivantes :

- **Tolérance aux pannes:**les données d'un nœud sont automatiquement répliquées vers d'autres nœuds. Ainsi, si un nœud est hors service les données présentes sont disponibles à travers d'autres nœuds.
- **Décentralisé:**Dans un cluster tous les nœuds sont égaux. Il n'y pas de notion de maitre et esclave, ni de processus qui se charge la gestion des différents nœuds.
- **Élastique:** la scalabilité est linéaire c.-à-d. le débit d'écriture et de lecture augmente de façon linéaire lorsqu'un nouveau nœud est ajouté dans le cluster

**Cassandra Query Langage (CQL) :**

CQL est un langage d'interrogation de données pour. CQL est assez proche du SQL standard du point de vue syntaxe qui permet une prise en main rapide par les développeurs tout en présentant des particularités propres à Cassandra que l'on ne retrouve pas dans le monde relationnel.

### *6.1.3. Spark Job Server*

Spark job server est un api REST pour spark crée par l'entreprise Ooyala en 2013. Cette outil nous permettons de crée des service REST et les déployé dans apache spark , ces services pouvait être invoque via http par un navigateur ou bien un code de n'importequelle langage de programmation.Le résultat des servicesdéployésest sous format fichier JSON structuré qui permet d'échange rapide de données, l'outil spark job server permet aussi d'allouer des cores et mémoire pour l'exécution très rapide.

### *6.1.4. Apache Kafka*

Apache Kafka est un système distribué orienté message de type publication/abonnement (pub-sub) hautement performant et tolérant aux panes implémenté comme système de tracesdistribué, adapté pour la

consommation de messages en ligne et hors ligne. Il s'agit d'un système orienté message développé à l'origine à LinkedIn pour la collection et la distribution de volumes élevés d'évènements et de données de trace à latence faible.

Apache Kafka diffère de système de messagerie traditionnelle :

- Il est conçu comme un système distribué qui est très facile à monter en charge.
- Il offre haut débit à la fois pour la publication et l'abonnement.
- Il prend en charge multi-abonnés et équilibre automatiquement les consommateurs en cas de panne.
- Il persiste les messages écrits sur le disque et peut donc être utilisé pour la consommation a lots (batching), en plus des applications en temps réel.

### *6.1.5. Apache zookeeper*

Apache Zookeeper est un service de coordination de haute performance pour les systèmes distribuées, qui permit de stocker et de gérer les paramètres de configuration de l'ensemble du système et de ses composants, ZooKeeper est tolérant au panne et le stockage de la configuration est centralisée, qui est toujours en ligne, permet de maintenir la configuration dynamique du système distribué et éviter les problèmes avec l'intégrité des données.

### *6.1.6. Esper CEP*

Esper est une plateforme Java dédiée au traitement d'événement (CEP) permet le développement rapide d'applications qui analysent de grands volumes de messages ou des événements. Esper apport une grande liberté nous permette de l'intégrer facilement à notreenvironnement de réalisation.

## 6.2. Les outils de développement

### *6.2.1. Eclipse IDE*

Eclipse IDE est un environnement de développement intégré, extensible, universel et polyvalent, permettant potentiellement de créer des projets de développement mettant en œuvre n'importe quel langage de programmation. Eclipse IDE est principalement écrit en Java (à l'aide de la bibliothèque graphique SWT, d'IBM), et ce langage, grâce à des bibliothèques spécifiques, est également utilisé pour écrire des extensions.

### 6.2.2. Wildflay 8

Wildflay est le successeur de JBoss, Wildflay est un serveur d'application développé par Red Hat et fonctionne sur de multiples plateformes. Il prise en charge la dernier version Java EE (JEE7) qui apport des richesses de fonctionnalités, ainsi WildFly support les deniers standards pour le développement web moderne.

### 6.2.3. Apache Shiro

Apache Shiro est un framework de sécurité Java très puissant et facile à utiliser, qui prend en compte la gestion de l'authentification, l'autorisation, la cryptographie et la gestion des sessions qui peut être utilisé pour sécuriser une application. L'architecture de shiro est facile à comprendre ce qui nous aidé à implémenter rapidement la sécurité dans notre application.

### 6.2.4. JRebel

JRebel est un plugin JVM (java virtuelle machine) qui permet aux développeurs Java de voir instantanément tout changement de code apportées à une application sans redéployer ou redémarrage du serveur.JRebel est une solution alternative à la mise à jour des classes qui ne nécessite pas une session de débogage à démarrer. Au contraire, elle surveille le système de fichiers pour les changements et les mises à jour des classes en mémoire. Cela signifie que seules les classes compilées pour fichiers ". Classe" seront mis à jour et les changements de classes dans les fichiers JAR seront ignorés.

### 6.2.5. RazorSQL

RazorSQL est un logiciel conçu pour manipuler une base de données. Il intègre plusieurs fonctionnalités pratiques pour faciliter ce genre de tache. RazorSQL prend désormais en charge les bases de données Cassandra, et il nous a donné la possibilité de se connecter, requête, parcourir et modifier les bases de données Cassandra via le GUI RazorSQL.

### 6.2.6. AngularJS

AngularJS est un framework JavaScript qui étend le HTML pour le rendre dynamique, et permet de développer des propres balises et attributs HTML. C'est un framework qui se veut extensible et qui pousse vers un développement structuré, le but n'étant pas d'ajouter de simples animations au DOM, mais bien d'apporter un aspect applicatif au front-office.

## 7. MISE EN OEUVRE DU PROTOTYPE

La mise en œuvre ou le déploiement d'application a été fait dans deux parties.

### 7.1. Premier partie : traitement de données

Les données générer de simulateur sont envoyer au serveur JMS apache Kafka, ce dernier réalise une interface entre le simulateur et apache spark, par la suite spark récupérer les données générer de simulateur depuis Kafka pour faire des analyse sur eux, le principe est basé sur publication/abonnement (pub-sub), le simulateur publier les données a un topic dans kafka puis spark s'abonner a ce topic et récupérer les données. De manière générale les données sont envoyées via kafka au spark en temps réel.La mise en place de Kafka nécessité l'utilisation d'apache Zookeeper pour la coordination de différents brokers et les producteurs.

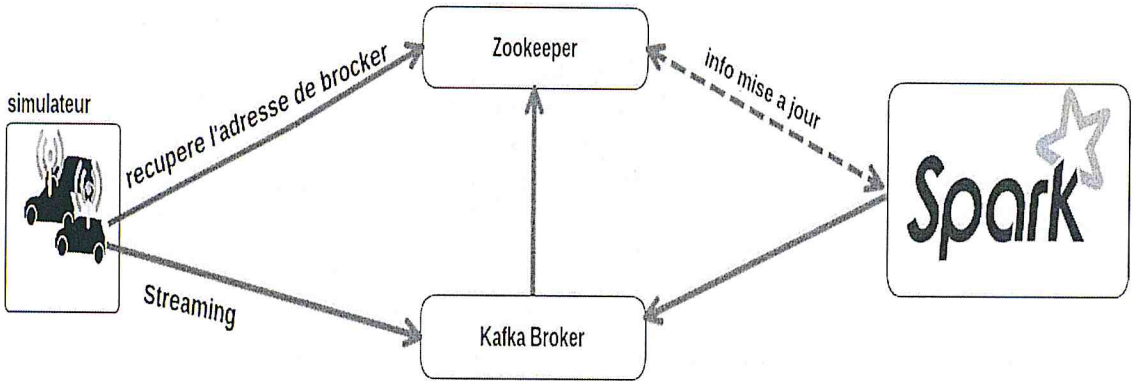



**Figure V-4 : intégration de zookeeper, kafka et spark**

Par la suite spark et en mode streaming, reçoit et analysé les données, il distribuer les données sous forme de RDDs et exécuté sur eux des opérations parallèles, puisles résultatssont à la fois stocker dans la base de données Cassandra et envoyer au Esper CEP pourappliqué des règles et détecté l'embouteillage.

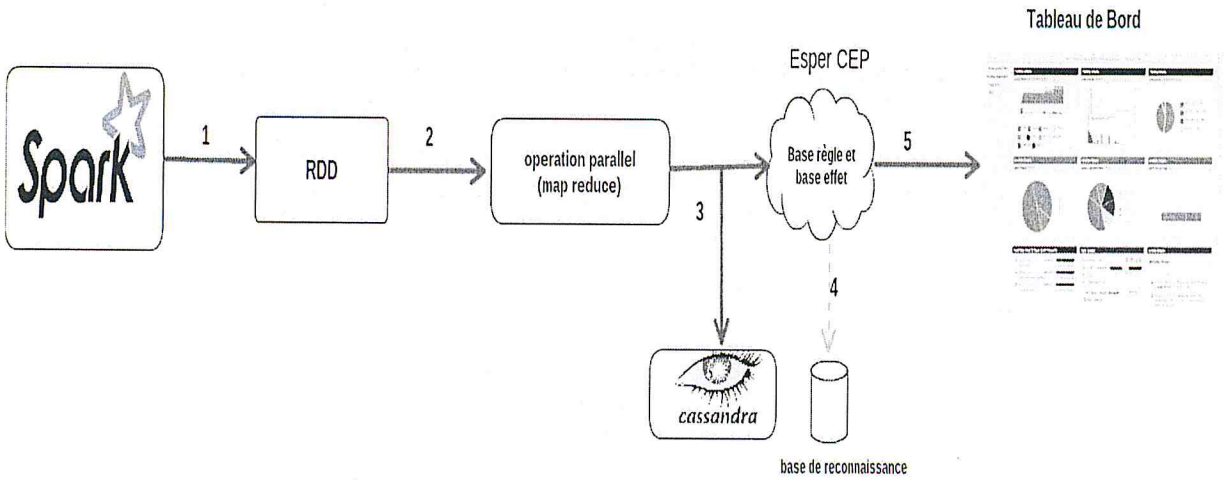



**Figure V-5 : traitement de données dans spark**

### 7.2. Deuxième partie : application web

Nous avons choisi java EE7 pour le développement de notre application, Nous avons développé des méthodes pour l'invocation des services REST qui sont publie dans Spark via spark job server

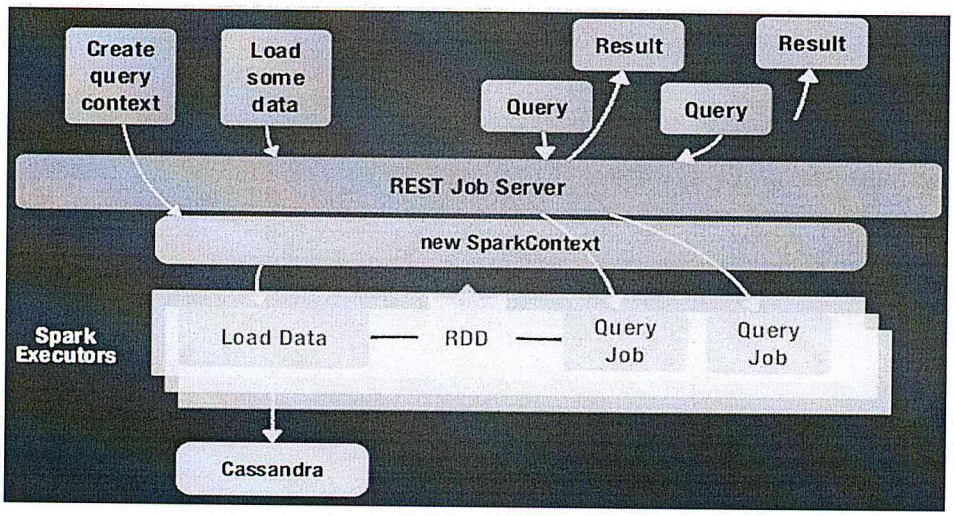



**Figure V-6:illustration de REST job server**

Nous avons aussi utilisé Web socket qui a une grande valeur dans l'application puisque il permet de diffusé les donnée reçue par le serveur en temps réel pour permet de contrôlé le trafic.

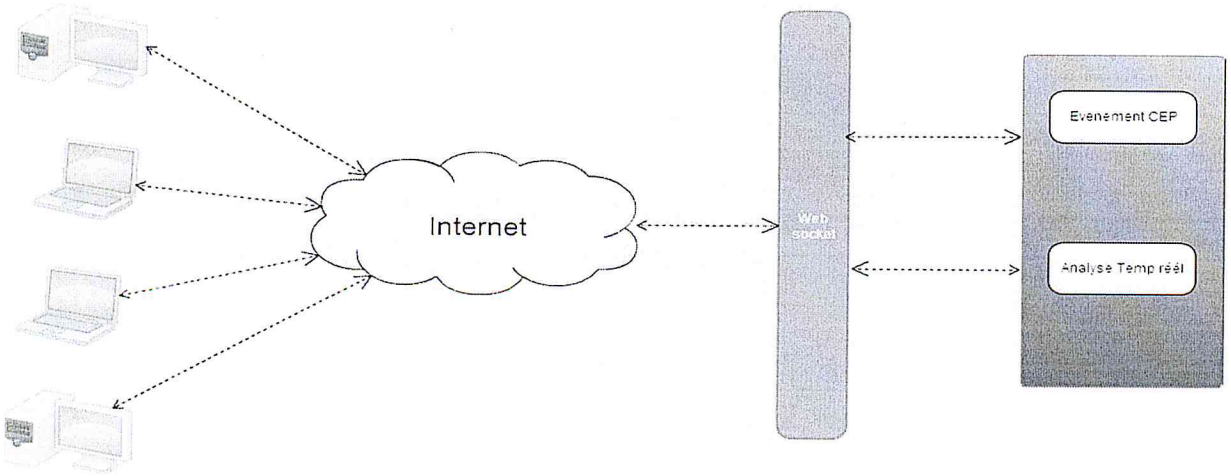



**Figure V-7:web soket et diffusion de données**

Enfin dans la partie visualisation nous avons utilisé HTML5 et java script. Et nous avons utilisé aussi angularjs pour rende l'application léger mêmes si il contient plusieurs composant web et interaction des données reçue et c'est le cas dans la partie table de borde qui a une grande interaction avec le serveur qui va rendre la page web assez lourd.

## 8. Présentation de l'application

Nous présentons dans cette section quelque interface principale de notre réalisation

### 8.1. Interface d'authentification

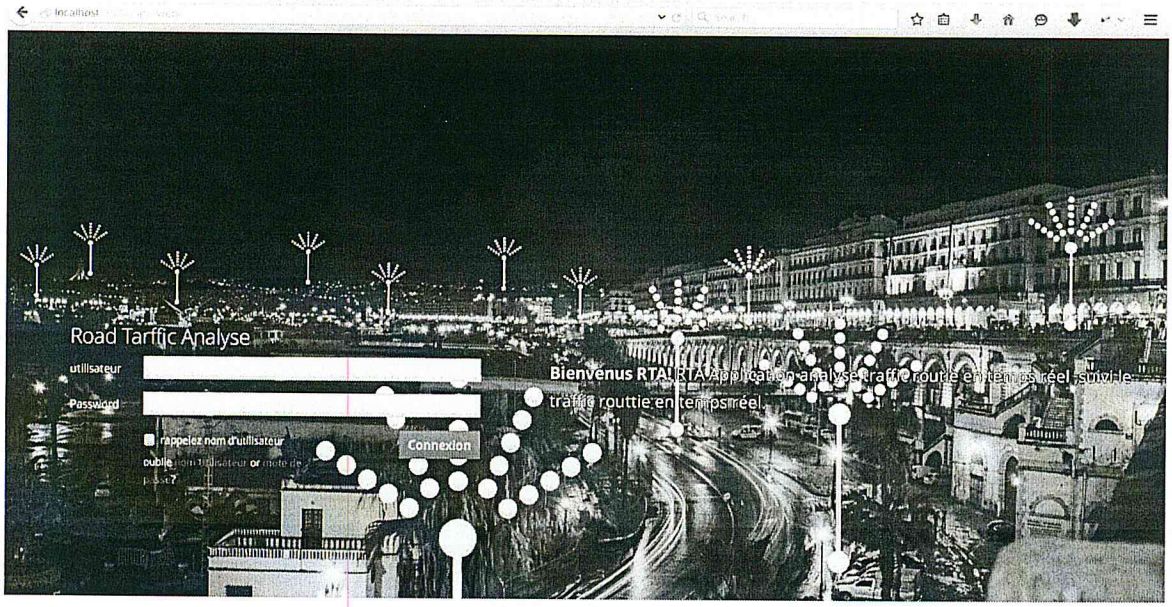



**Figure -V-8:interface d'authentification**

### 8.2. Interfaces d'administration :

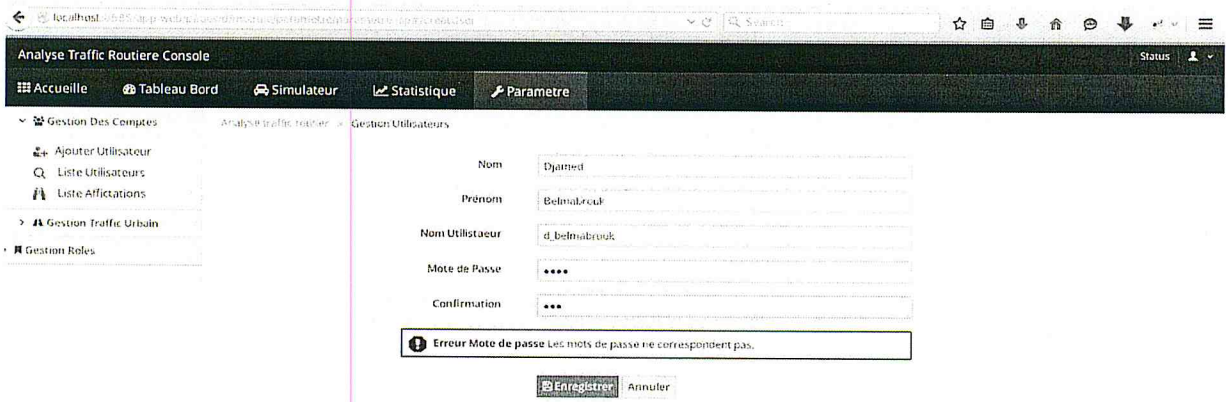



**Figure V-9:création d'un compte**

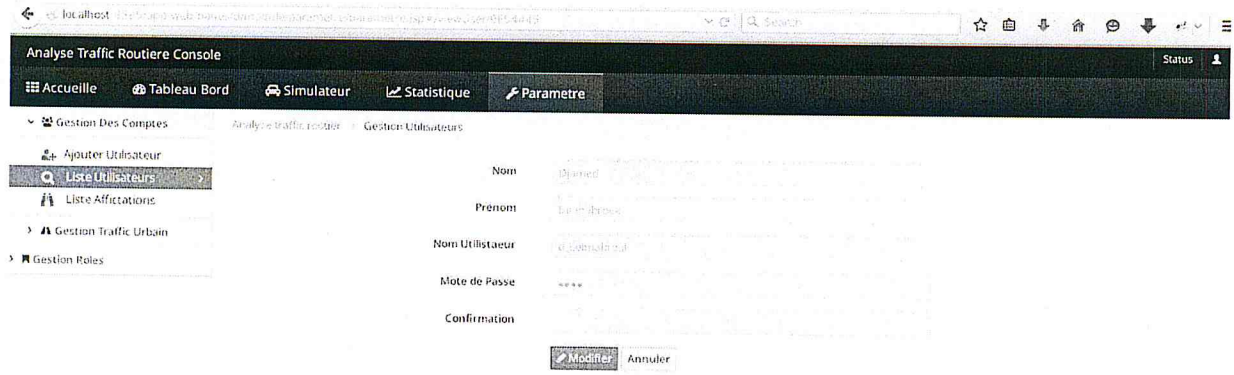



**Figure V-10:modification d'un compte**

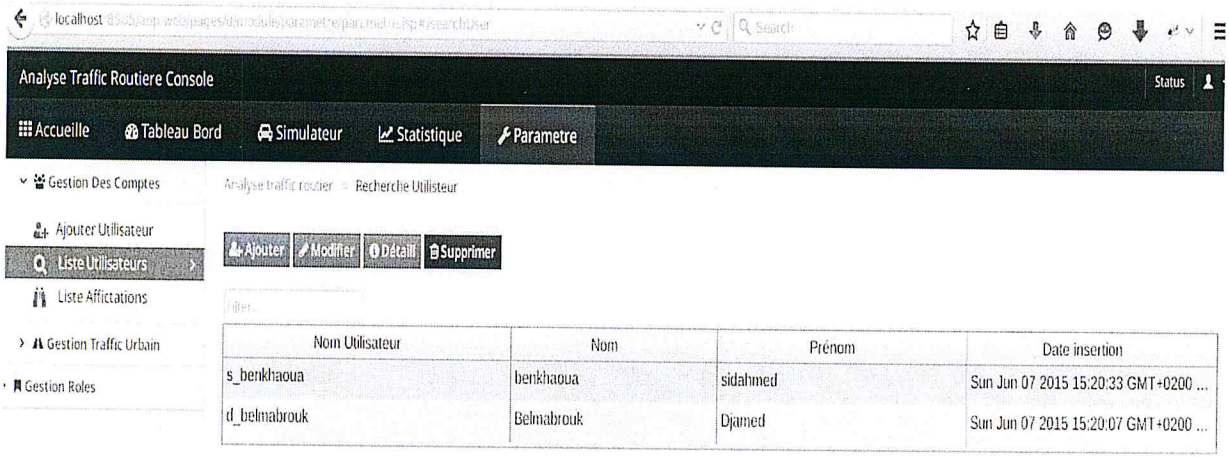



**Figure V-11:lister les utilisateurs**

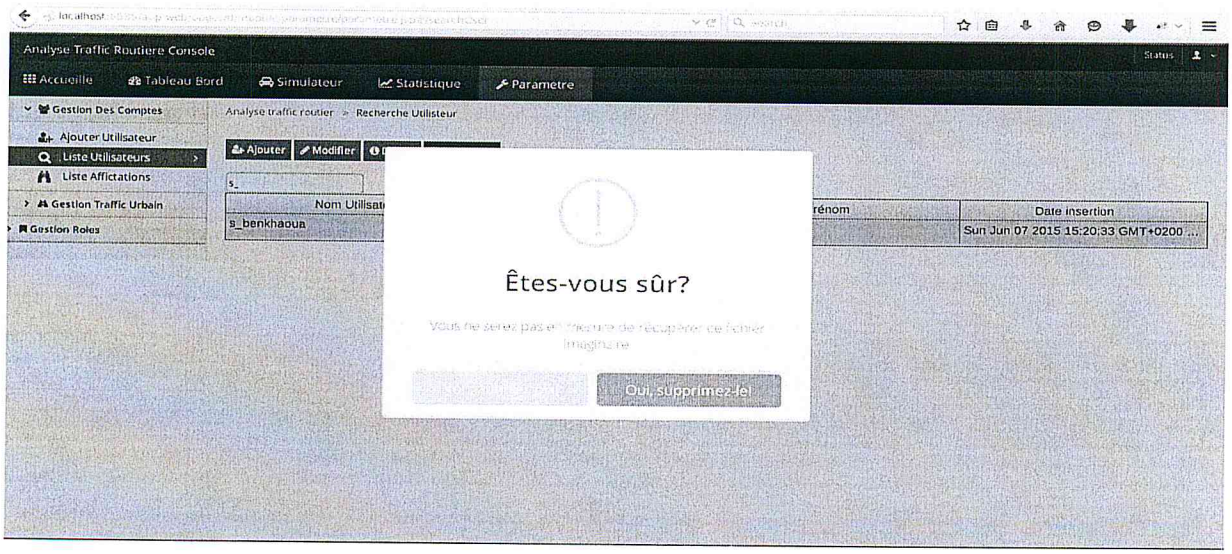



**Figure V-12:suppression d'un utilisateur**

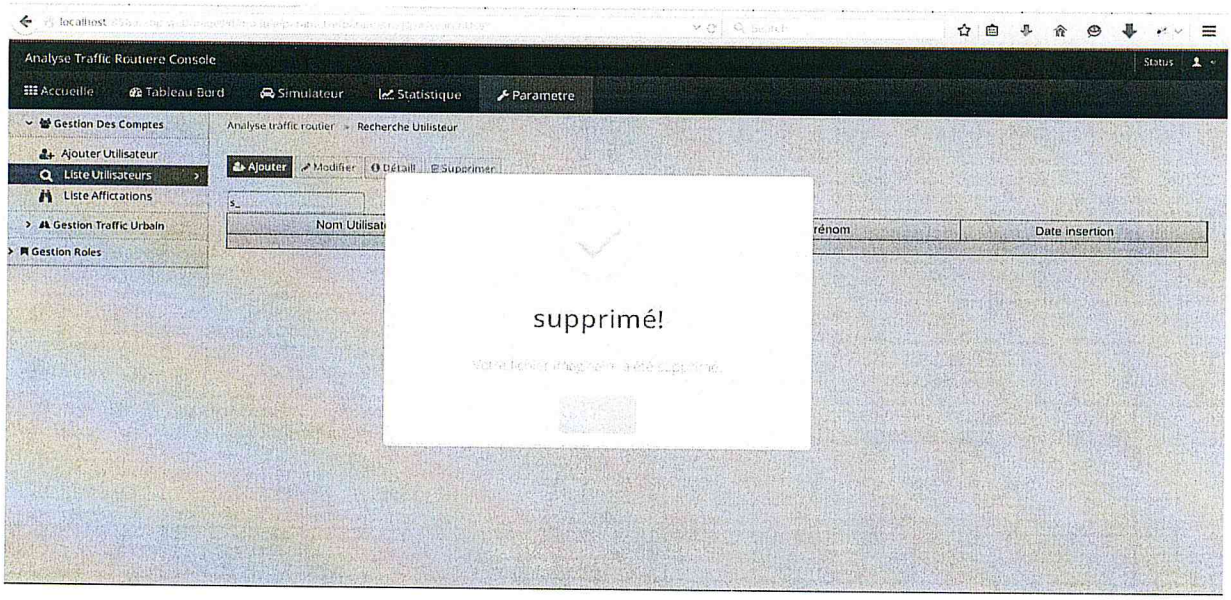



**Figure V-13:utilisateur supprimé**

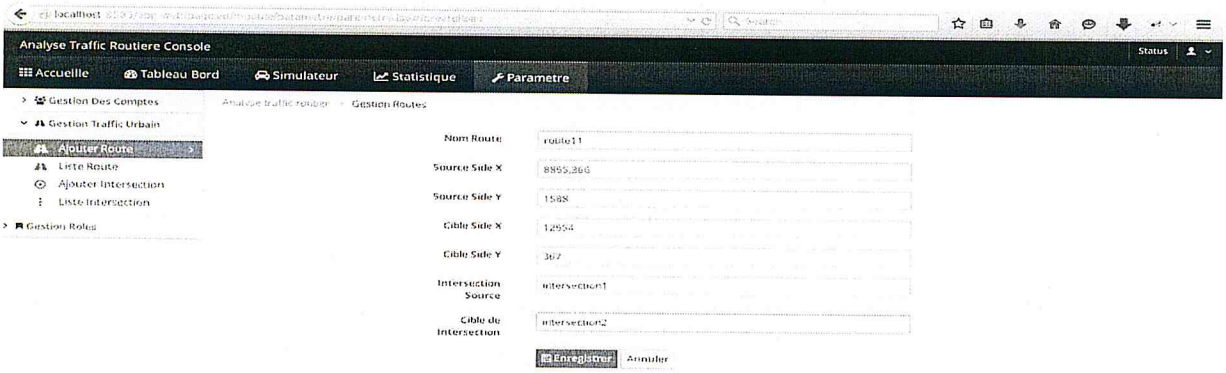



**Figure V-14:ajouter une route**

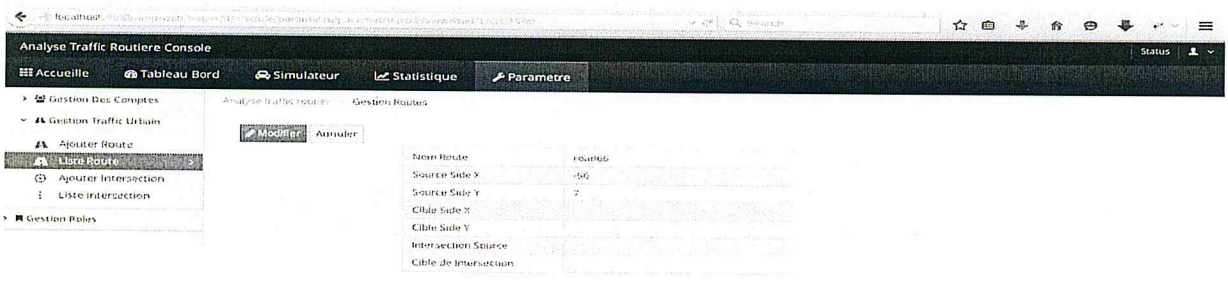



**Figure V-15:détail d'une route**

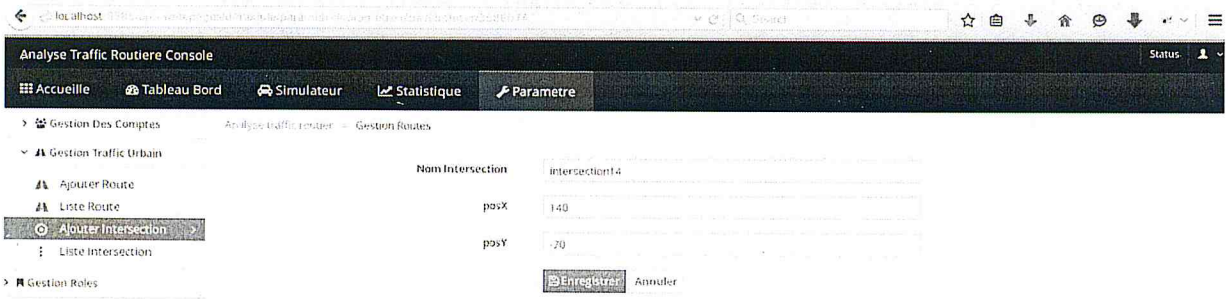



**Figure V-16:ajouter intersection**

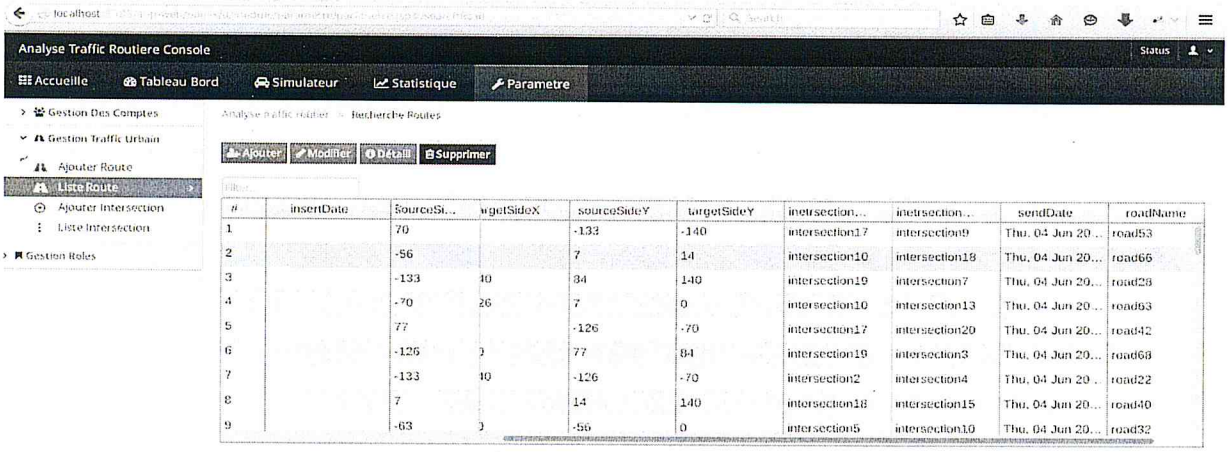



**Figure V-17:lister routes**

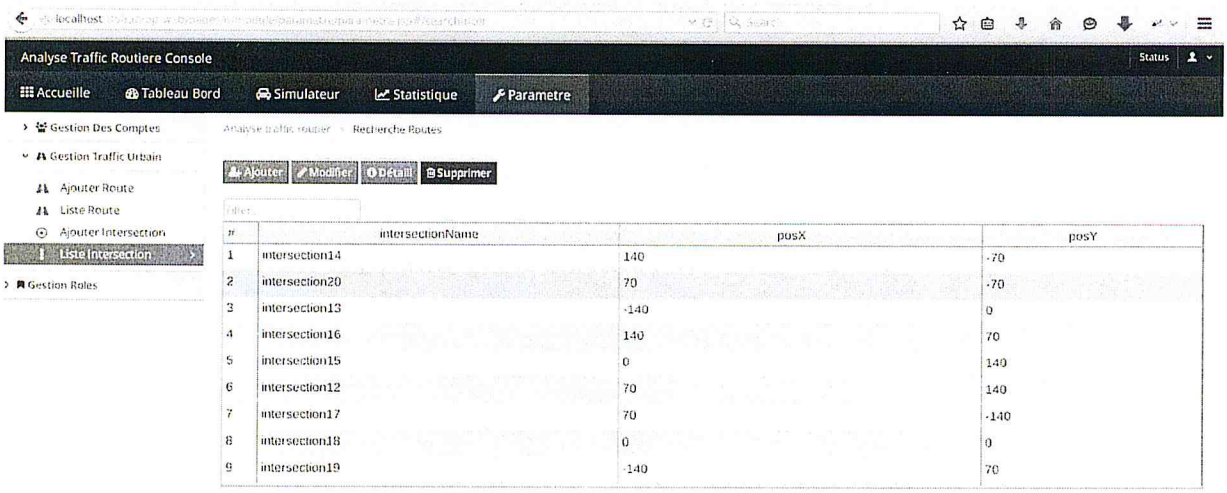



**Figure V-18:lister intersections**

## 8.3. Tableau de bord :

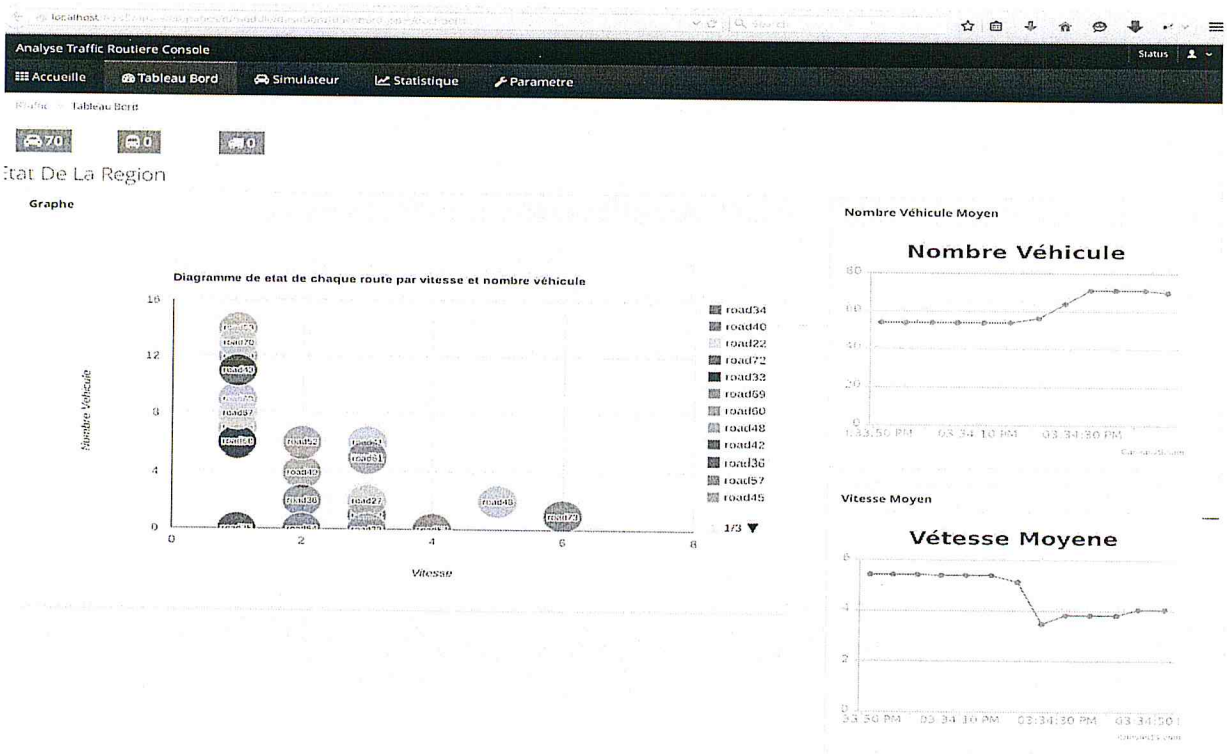



**Figure V-19:tableau de bord de suivi l'état de trafic**

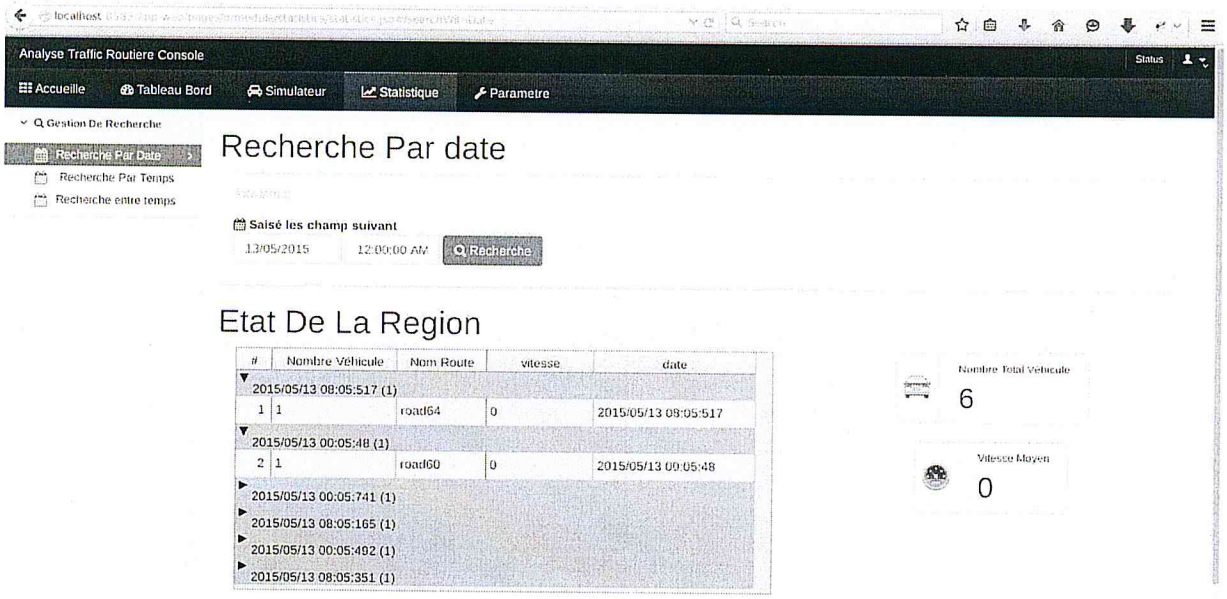



**Figure V-20:statistique par date**

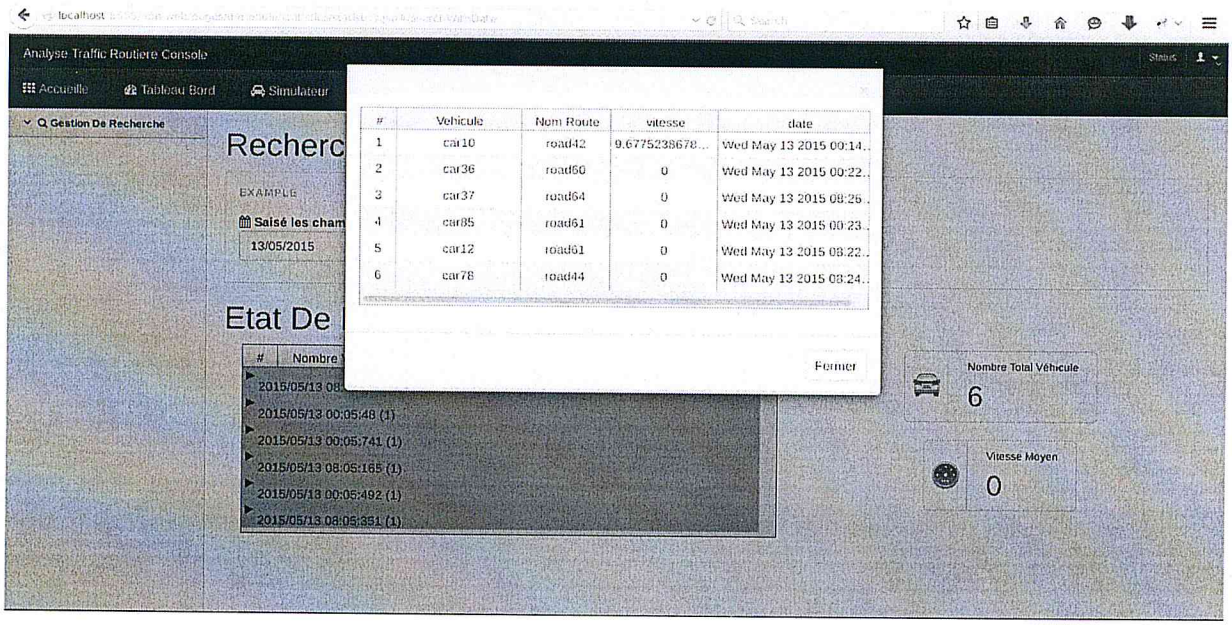



**Figure V-21:lister les véhicules pour chaque route pour une date**

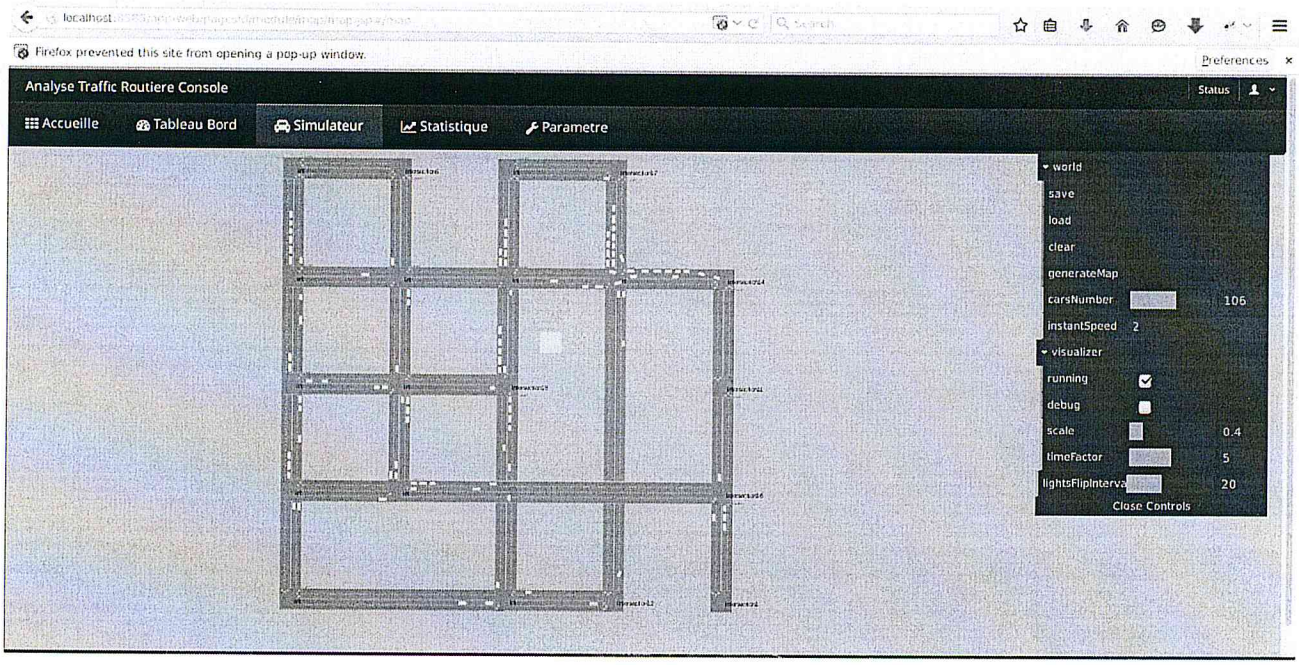



**Figure V-22:page de simulateur et de contrôle**

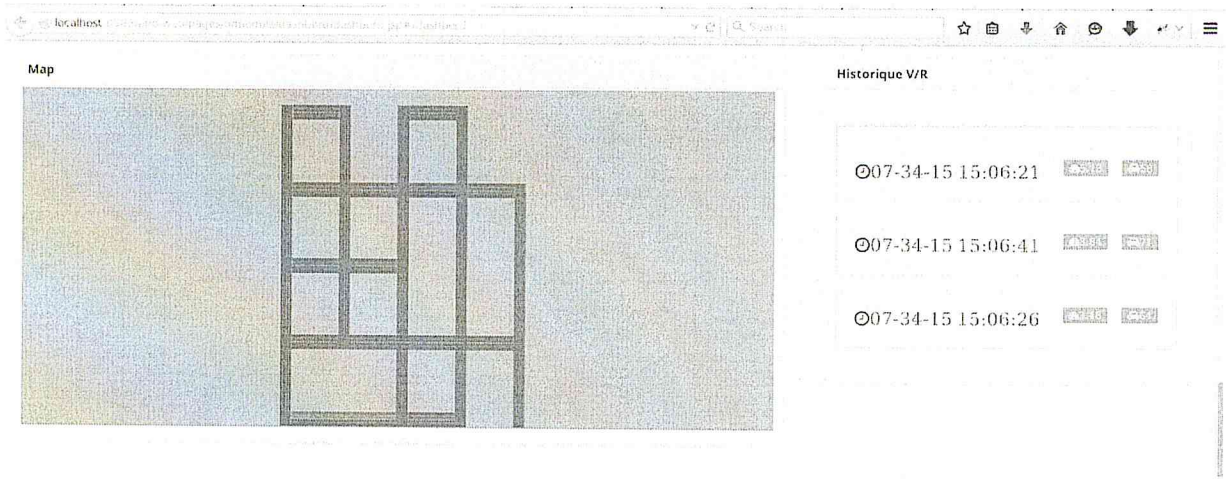



**Figure V-23: alerte de congestion**

## 9. Conclusion

Dans ce chapitre, les solutions que nous avons proposées aux problèmes posées ont été présentées d'un point de vue technique, Plusieurs technologies se sont avérées nécessaires pour la mise en œuvre du prototype développé. Et enfin nous avons présenté quelque interface de notre prototype développé.

# vi. Conclusion générale et perspective

## 1. Conclusion générale

Dans ce projet, notre mission a été de mise en place un prototype de système de gestion et suivi de trafic routier, le défi était alors d'exploiter tout la puissance des nouvelles technologies de Big Data pour analysé en temps réel l'état de trafic urbain et mise en place des plans de gestion.

Au fil de ce mémoire, nous avons vu plusieurs aspects, nous avons commencé par une présentation relative de IoT, de Big Data et analyse en temps réel en faisant un tour des technologies existantes et des principes de ces tendances qui évoluent quotidiennement. Ensuite, nous nous sommes recentrés sur l'aspect conceptuel et implémentation.

Lors de la réalisation de notre projet, nous avons été confrontés à plusieurs problèmes, parmi eux, de trouvé un simulateur qui permet de simuler l'infrastructure routier réel et génère la mase de données attendus. L'autre besoin majeur sur lequel nous nous sommes concentrés est la réalisation d'une application web dynamique. Cette application doit notamment offrir des services de gestion et statistiques de trafic.

Ce projet sa donné lieu à un système de suivi et détection des congestions dans un réseau routier, un tableau de bord réalisé est basé sur une gamme technologique derrière très évolué et moderne, permit au superviseur d'être au courant de ce qui passe dans un réseau routier et prend aisément d'action de control, pour l'usager routier des informations de circulation en temps réel lui permet de se déplacer confortablement.

Ce travail nous a, non seulement, donné l'opportunité de manipuler de nouvelles technologies dédiées au à la collecte des données (Iot) et au stockage et traitement des données de masse, mais nous a aussi permit d'approfondir nos connaissance en matière d'analyse et de conception des systèmes.

## 2. Perspectives

Le sujet dans lequel nous nous sommes lancés est très vaste et le travail réalisé peut être amélioré et suivi afin d'en faire un système plus dynamique et autonome. Parmi les perspectives à prendre en compte pour améliorer le fonctionnement du système, plus particulièrement sur les plans de régulation du trafic tel que la régulation d'accès, la limitation dynamique de vitesse et feux de signalisation nécessite des algorithmes, heuristiques et recours au demain de l'automatique et mathématiques appliquées et recherche opérationnel, il sera intéressant d'investiguer sur l'apport des méthode méta-heuristiques pour la commande

prédictive et exploiter ces domaines de recherche pour mettre en place un plan de circulation rendre le trafic plus fluide.

# vii. Bibliographie


[1] Mark Weiser, "The Computer for the 21st Century," no. 3, 94 - 104, 1991.

[2] David R, Ahmed Nait-Sidi-Moh, David Durand, et Jérôme Fortin, "Internet des objets et interopérabilité des flux logistiques: état de l'art et perspectives".

[3] Ovidiu Vermesan et Peter Friess, *Internet of Things: Converging Technologies for Smart Environments and Integrated Ecosystems*. Denmark: River Publishers, 2013.

[4] Sunil K, Timalsina, et Rabin Bhusal, "NFC and Its Application to Mobile Payment: Overview and Comparison," 2012.

[5] Jayavardhana Gubbi, Rajkumar Buyya, Slaven Marusic, et Marimuthu Palaniswami, "Internet of Things (IoT): A vision, architectural elements, and future directions," no. 1645–1660, 2013.

[6] Ma Zhonggui, Shang Xinsheng, fu Xinxi, et luo feng, "THE ARCHITECTURE AND KEY TECHNOLOGIES OF INTERNET OF THINGS IN LOGISTICS," no. 464-467, 2012.

[7] Akyildiz F, Su W, Sankarasubramaniam Y, et Cayirci E, "Wireless sensor networks : a survey," no. 393–422, 2002.

[8] Abiodu Olonibua et kayode Akingbade, "Wireless Transmission of Biomedical Signals Using the Zigbee Technology," 2013.

[9] Cheng Hong, Ni Wancheng, et Li Na, "A Systematic Scheme For Designing RFID Systems With High Object Detection Reliability," 2008.

[10] Ali Alshehri et Steve Schneider, "Formally defining NFC M-coupon requirements, with a case study," *The 8th International Conference for Internet Technology and Secured Transactions (ICITST)*, pp. 52-58, 2013.

[11] Choi JinSoo et Zhou MengChu, "Performance Analysis of ZigBee-based Body Sensor Networks," pp. 2427-2433, 2010.

[12] Shelby Z et Bormann C, "6LoWPAN : The Wireless Embedded Internet," 2010.

[13] Ziegler Sébastien et al., "Iot moving to an ipv6-based future iot," pp. 161-172, 2013.

[14] Want Roy et Hopper Andy, "ACTIVE BADGES AND PERSONAL INTERACTIVE COMPUTING OBJECTS," vol. 38, no. 1, pp. 11-19, février 1992.

[15] Marie-Pierre Hamel et David Marguerit, Analyse des big data Quels usages, quels défis ?, 2013.

[16] Gabriele Cavallaro et al., "SMART DATA ANALYTICS METHODS FOR REMOTE SENSING APPLICATIONS," pp. 1405-1408, 2014.

[17] Blandine LAFFARGUE, guide du big data, 2013.

[18] Fabio Coelho, Francisco Cruz, Ricardo Vilaca, José Pereira, et Rui Oliveira, "pH1: A Transactional Middleware for NoSQL," *2014 IEEE 33rd International Symposium on Reliable Distributed Systems*, pp. 115-124, 2014.

[19] Eric A Brewer, Towards Robust Distributed Systems, 2000.

[20] Tuan Dinh Le, Seong Hoon Kim, Minh Hoang Nguyen, et Daeyoung Kim, "EPC Information Services with No-SQL datastore for the Internet of Things," *IEEE International Conference on RFID (IEEE RFID)*, pp. 47-58, 2014.

[21] HAN HU, YONGGANG WEN, TAT SENG CHUA, et XUELONG LI, "Toward Scalable Systems for Big Data Analytics:A Technology Tutorial," vol. 2, 2014.

[22] le NoSQL. (2015) www.philippebechet.com/.

[23] Apache hadoop. http://hadoop.apache.org.

[24] CHENG ZHANG PENG, ZE JUN JIANG, XIAO BIN CAl, et ZHI KE ZHANG, "REAL-TIME ANALYTICS PROCESSING WITH MAPREDUCE," 2012.

[25] Mingyue Luo et Gang Liu, "Distributed log information Processing with Map-Reduce," 2010.

[26] Apache spark. http://spark.apache.org/.

[27] Mohiuddin Solaimani, Mohammed Iftekhar, Latifur Khan, et Bhavani Thuraisingham, "Spark-based Anomaly Detection Over Multi-source VMware Performance Data In Real-time," 2014.

[28] Matei Zaharia, Mosharaf Chowdhury, Tathagata Das, et Ankur Dave, "Resilient Distributed Datasets: A Fault-Tolerant Abstraction for In-Memory Cluster Computing".

[29] Matei Zaharia, Mosharaf Chowdhury, Scott Shenker, et Ion Stoica, "Spark: Cluster Computing withWorking Sets".

[30] spark streaming. streaming.spark.apache.org.

[31] L'analyse décisionnelle en temps réel. (2012) http://blog.octo.com/.

[32] Jürgen Dunkel, "On Complex Event Processing for Sensor Networks".

[33] Andrew Berry et Zoran Milosevic, "Real-time analytics for legacy data streams in health," *17th IEEE International Enterprise Distributed Object Computing Conference*, 2013.

[34] David Luckham et Roy Schulte, "Event Processing Glossary," 2008.

[35] Etzion Opher. (2008) http://epthinking.blogspot.com/2008/07/on-relationships-among-derived-event.

[36] Michael Eckert et Francois Bry, "Complex Event Processing (CEP)," 2009.

# viii. Glossaire

***6***

6LoWPAN
IPv6 Low power Wireless Personal Area Networks · 10

***A***

ACID
Atomicité, Cohérence, Isolation, Durabilité · 18

***B***

BD
Base de Données · 21

***C***

CAP
Consistency, Availability, Partition Tolerance · 18
CEP
Complex Event Processing · 32

***E***

ECA
Evènement-Condition-Action · 37
EDA
Event-Driven Architectures · 30
EPL
Event Processing Languages · 37
ESP
Event Stream Processing · 36

***G***

*GPS*
Global Positioning System · 10

## *H*

Html
Hyper Text Markup Language · 20

## *I*

IEEE
Institute of Electrical and Electronics Engineers · 10
IEEE 802.15.4
est un protocole de communication défini par IEEE déstiné aux dispositifs faible débit et faible consommation · 9
IoT
Internet of Things · 6
IP
Internet Protocol · 10

## *J*

json
Un document JSON a pour fonction de représenter de l'information accompagnée d'étiquettes permettant d'en interpréter les divers éléments, sans aucune restriction sur le nombre de celles-ci. · 77

## *M*

Middleware
intergiciel servant d'intermédiaire de communication entre des applications · 7
MIMD
Multiple Instructions Multiple Data · 17

## *N*

NFC
Near Field Communication · 9
NoSQL
Not only SQL · 18

## *R*

RAM
Random Access Memory · 17
RDBMS
système de gestion de base de données relationnel · *Voir*

جامعة البليدة 1
مكتبة
الكلية
كلية العلوم

## S



## X