

Département de Génie Logiciel
Faculté des Sciences et Techniques de Tanger
Université Abdelmalek Essaâdi, Maroc
01/04/2026

Traffic Core : Système Intelligent de Gestion du Trafic basé sur la Blockchain Hyperledger Fabric

Encadré par : Pr. Ikram BEN ABDEL OUAHAB
Réalisé par : Mezouri Chaymae, En-najy Chaymae , Idrissi Janati Yassir et Cheairi Hamza

RÉSUMÉ

La gestion du trafic urbain est un enjeu majeur des villes intelligentes, nécessitant des systèmes fiables, sécurisés et transparents. Les solutions centralisées traditionnelles présentent toutefois des limites en matière de traçabilité et de sécurité des données. Ce projet propose Traffic Core, un système de simulation du trafic urbain basé sur une blockchain permissionnée Hyperledger Fabric. Les événements de trafic sont enregistrés sous forme de transactions immuables, garantissant l'intégrité et l'auditabilité des données. Le système repose sur le consensus SmartBFT, assurant la tolérance aux fautes byzantines et la continuité du service. Un smart contract a été développé pour gérer les actifs du trafic et détecter automatiquement les infractions. Un dashboard interactif permet la visualisation en temps réel du réseau et des transactions blockchain. Les résultats obtenus démontrent l'apport de la blockchain pour la fiabilité et la transparence des systèmes de Smart City, tout en ouvrant la voie à des extensions futures basées sur l'intelligence artificielle.

Mots-clés

Blockchain, Smart City, Hyperledger Fabric, SmartBFT, Gestion du trafic, Systèmes distribués, Simulation urbaine, Tracabilité, Sécurité des données, Dashboard interactif, Smart Contract.

1. Introduction

La gestion du trafic urbain constitue un enjeu majeur pour les villes intelligentes, confrontées à une urbanisation croissante et à une augmentation continue du nombre de véhicules. L'optimisation de la mobilité, la sécurité routière et la prise de décision en temps réel nécessitent des systèmes capables de traiter des données fiables et cohérentes issues de multiples acteurs urbains.

Les solutions traditionnelles reposant sur des architectures centralisées présentent toutefois des limites importantes en matière de sécurité, de transparence et de traçabilité des données. La centralisation engendre des risques de manipulation, de perte d'informations et de défaillance unique, compromettant ainsi la fiabilité des systèmes de gestion du trafic.

Dans ce contexte, la blockchain apparaît comme une technologie pertinente pour renforcer la confiance et l'intégrité des données grâce à un registre distribué, immuable et auditible. Ce rapport présente Traffic Core, un système de gestion du trafic urbain basé sur une blockchain permissionnée Hyperledger Fabric, intégrant un mécanisme de consensus tolérant aux fautes byzantines (SmartBFT). Le projet propose un environnement de simulation Smart City dans lequel les événements de trafic sont enregistrés sous forme de transactions vérifiables, accompagnés d'un dashboard interactif permettant la visualisation en temps réel du système.

2. Contexte général du projet

Les villes intelligentes (Smart Cities) reposent sur l'exploitation coordonnée des technologies numériques afin d'améliorer la gestion des services urbains tels que le transport, l'énergie, la sécurité et la mobilité. Ces systèmes s'appuient sur la collecte et le traitement de grandes quantités de données générées en temps réel par des capteurs, des infrastructures connectées et des véhicules intelligents. Dans le domaine du trafic urbain, ces données sont essentielles pour optimiser la circulation, réduire la congestion et améliorer la sécurité routière. Cependant, la majorité des solutions actuelles de gestion du trafic adoptent une approche centralisée, dans laquelle une entité unique contrôle l'ensemble du système et des données. Cette centralisation soulève plusieurs problématiques, notamment en matière de sécurité, de transparence et de fiabilité. Les données de trafic peuvent être vulnérables aux manipulations, aux erreurs humaines ou aux cyberattaques, tandis que l'absence de mécanismes de traçabilité complique l'audit des événements passés et la responsabilisation des différents acteurs. Dans ce contexte, la blockchain se présente comme une technologie capable de répondre à ces défis en introduisant un registre distribué partagé entre plusieurs parties prenantes. Grâce à ses mécanismes de consensus et à l'utilisation de la cryptographie, la blockchain garantit l'immutabilité, la traçabilité et la non-répudiation des transactions. Elle permet ainsi d'instaurer une couche de confiance entre les différents services urbains sans dépendre d'une autorité centrale unique.

Le projet s'inscrit dans cette vision en proposant la conception d'un simulateur de Smart City décentralisée, dans lequel plusieurs modules urbains interagissent à travers une blockchain commune déployée sur Hyperledger Fabric. Chaque module est représenté par des transactions blockchain assurant la validation et l'enregistrement des interactions. Dans ce cadre global, le sous-projet Traffic Core se concentre sur la simulation et la gestion du trafic urbain, en intégrant des mécanismes de consensus avancés et une interface de visualisation permettant d'observer en temps réel le fonctionnement du système.

Ce contexte met en évidence l'intérêt de la blockchain comme infrastructure de confiance pour les Smart Cities et justifie le choix d'une approche décentralisée pour la gestion du trafic urbain, à la fois sécurisée, transparente et extensible.

3. Architecture Générale du Système

L'architecture du système Traffic Core a été conçue selon une approche décentralisée, dans laquelle la blockchain joue le rôle de couche centrale de confiance assurant la coordination, la validation et la traçabilité des interactions entre les différents composants du système. Cette architecture vise à garantir la fiabilité des données de trafic, tout en permettant une visualisation et une simulation en temps réel du fonctionnement de la Smart City.

3.1 Vue d'ensemble de l'architecture

Le système repose sur trois couches principales :

- **Couche Blockchain** : assure l'enregistrement immuable et la validation des événements de trafic.
- **Couche Applicative** : gère la logique métier, la simulation et l'interaction avec la blockchain.
- **Couche Présentation** : permet la visualisation et le contrôle du système via une interface utilisateur.

3.2 Architecture Blockchain

La couche blockchain est basée sur une blockchain permissionnée Hyperledger Fabric, choisie pour son adaptabilité aux environnements institutionnels et aux Smart Cities. Le réseau blockchain est structuré autour des éléments suivants :

- **Orderers** : responsables de l'ordonnancement des transactions et du consensus.
- **Peers** : chargés de l'exécution du smart contract (chaincode) et de la validation des transactions.
- **Organisations** : représentant différents acteurs du système urbain.
- **Canal dédié** : isolant les transactions liées au projet Traffic Core.
- **Smart Contract (Chaincode)** : implémentant la logique métier du trafic urbain.

Le mécanisme de consensus utilisé est SmartBFT (Smart Byzantine Fault Tolerant), qui permet au réseau de continuer à fonctionner même en présence de nœuds défaillants ou malveillants. Ce consensus renforce la résilience et la sécurité du système par rapport aux mécanismes classiques de type leader-based.

3.3 Couche Applicative et Moteur de Simulation

La couche applicative constitue l'interface entre la blockchain et les utilisateurs. Elle est responsable de :

- La gestion des appels aux smart contracts via le SDK Hyperledger Fabric.
- L'exécution du moteur de simulation du trafic, qui génère automatiquement des événements réalistes (déplacements des véhicules, congestion, incidents, changements de feux).
- La transformation de chaque événement en une transaction blockchain, garantissant la traçabilité complète du système.

Cette couche permet d'assurer une séparation claire entre la logique métier et l'infrastructure blockchain, facilitant ainsi l'extensibilité et la maintenance du système.

3.4 Couche Présentation : Dashboard de Visualisation

La couche présentation est matérialisée par un dashboard web interactif, permettant la supervision et le contrôle du système en temps réel. Cette interface offre :

- Une visualisation du réseau routier (routes, intersections, véhicules).
- Un suivi en temps réel des transactions blockchain.
- Des interfaces CRUD pour la gestion des actifs du trafic.
- Des outils de simulation et de contrôle (démarrage, pause, reprise, réinitialisation).

Le dashboard communique avec la couche applicative via des API sécurisées, assurant une interaction fluide et cohérente avec la blockchain.

3.4 Flux des Transactions

Le fonctionnement global du système repose sur le flux suivant :

1. Un événement est généré par le moteur de simulation ou l'utilisateur.
2. L'événement est transmis à la couche applicative.
3. Une transaction est soumise au smart contract sur la blockchain.
4. Le consensus SmartBFT valide et ordonne la transaction.
5. Les données sont enregistrées de manière immuable dans le ledger.
6. Le dashboard est mis à jour en temps réel.

Ce flux garantit la cohérence, la sécurité et la traçabilité de l'ensemble des opérations du système Traffic Core.

4. Choix Technologiques

Le projet Traffic Core repose sur une stack technologique adaptée aux systèmes distribués et aux environnements Smart City. La couche blockchain est basée sur une blockchain permissionnée **Hyperledger Fabric (v3.1.3)**, permettant la gestion des identités, des canaux et des smart contracts. Le mécanisme de consensus utilisé est **SmartBFT** (Smart Byzantine Fault Tolerant), assurant la tolérance aux fautes byzantines et la fiabilité des transactions.

Le développement des smart contracts (chaincodes) et de la couche applicative est réalisé en JavaScript, avec une interaction à la blockchain via le **SDK Hyperledger Fabric pour Node.js**. Le déploiement du réseau blockchain et des services est assuré par Docker et Docker Compose.

L'interface utilisateur est implémentée sous forme **d'un dashboard web interactif** utilisant HTML5, CSS3 et JavaScript, avec une visualisation dynamique du réseau routier basée sur SVG. Cette stack garantit une intégration cohérente entre la blockchain, la simulation du trafic et la visualisation en temps réel.

5. Implémentation de la Blockchain

L'implémentation de la blockchain constitue le socle du système Traffic Core, assurant la validation, l'ordonnancement et l'enregistrement immuable des événements de trafic. Le réseau blockchain a été conçu de manière à garantir la fiabilité, la tolérance aux pannes et la traçabilité des transactions dans un environnement Smart City.

5.1 Mise en place du réseau Hyperledger Fabric

Le réseau blockchain est basé sur Hyperledger Fabric v3.1.3, une plateforme adaptée aux systèmes distribués institutionnels. L'architecture du réseau comprend plusieurs composants interconnectés :

- Orderers : responsables de l'ordonnancement des transactions et de l'application du consensus.
- Peers : chargés de l'exécution du smart contract et de la validation des transactions.
- Organisations : représentant les différents acteurs du système.
- Canal dédié (trafficchannel) : assurant l'isolation des transactions liées au projet Traffic Core.

L'ensemble des composants est déployé et orchestré à l'aide de Docker et Docker Compose, permettant un démarrage contrôlé et reproductible du réseau.

```
chaymae@DESKTOP-BD9M118:~/hyperledger-fabric/fabric-samples/test-network$ osnadmin channel join \
--channelID trafficchannel \
--config-block ./channel-artifacts/trafficchannel.block \
-o localhost:7053 \
--ca-file "${PWD}/organizations/ordererOrganizations/example.com/orderers/orderer.example.com/msp/tlscerts/tlsca.example.com-cert.pem" \
--client-cert "${PWD}/organizations/ordererOrganizations/example.com/orderers/orderer.example.com/tls/server.crt" \
--client-key "${PWD}/organizations/ordererOrganizations/example.com/orderers/orderer.example.com/tls/server.key"
Status: 201
{
  "name": "trafficchannel",
  "url": "/participation/v1/channels/trafficchannel",
  "consensusRelation": "consenter",
  "status": "active",
  "height": 1
}
```

Figure 1: Jonction du canal trafficchannel par un orderer via l'outil OSN Admin

5.2 Implémentation du Consensus SmartBFT

Le mécanisme de consensus utilisé dans ce projet est SmartBFT (Smart Byzantine Fault Tolerant), intégré dans les versions récentes de Hyperledger Fabric. Ce consensus permet d'assurer la continuité du service même en présence de nœuds défaillants ou malveillants.

Le réseau est configuré avec plusieurs nœuds orderers participant au consensus. SmartBFT garantit :

- L'accord sur l'ordre des transactions.
- La tolérance aux fautes byzantines.
- La suppression du point de défaillance unique.

Ce choix renforce la résilience du système et le rend particulièrement adapté aux infrastructures critiques telles que la gestion du trafic urbain.

```

Rejoining orderer.example.com on port 7053...
Status: 201
    "name": "trafficchannel",
Rejoining orderer2.example.com on port 8053...
Status: 201
    "name": "trafficchannel",
Rejoining orderer3.example.com on port 9053...
Status: 201
    "name": "trafficchannel",
Rejoining orderer4.example.com on port 10053...
Status: 201
    "name": "trafficchannel",
chaymae@DESKTOP-BD9M118:~/hyperledger-fabric/fabric-samples/test-network$ 

```

Figure 2: Jonction de plusieurs nœuds orderers au canal trafficchannel dans le cadre du consensus SmartBFT

5.3 Configuration du Réseau et des Canaux

La configuration du réseau blockchain repose sur la définition des profils de configuration et des politiques de sécurité. Un canal dédié est créé afin d'isoler les transactions du projet Traffic Core et de limiter l'accès aux seules organisations autorisées.

5.4 Déploiement du Smart Contract

Un smart contract (**chaincode**) a été déployé sur le canal blockchain afin de gérer la logique métier du trafic urbain. Le déploiement comprend :

- L'installation du chaincode sur les peers.
- L'approbation par les organisations.
- La validation et l'activation sur le canal.

Chaque transaction soumise au réseau est validée par le consensus SmartBFT avant d'être enregistrée de manière immuable dans le ledger.

5.5 Sécurité et Traçabilité des Transactions

La blockchain garantit la sécurité des données de trafic grâce à :

- L'utilisation de certificats numériques pour l'authentification des entités.
- La validation cryptographique des transactions.
- L'enregistrement immuable des données dans le ledger.

Chaque transaction est associée à un identifiant unique et à un horodatage certifié, permettant une traçabilité complète et un audit fiable des événements de trafic.

Figure 3: Explorateur de blocs et interrogation du World State sur la blockchain Hyperledger Fabric

6. Développement du Smart Contract (Chaincode)

Le smart contract, également appelé chaincode dans Hyperledger Fabric, constitue le cœur de la logique métier du système Traffic Core. Il permet de modéliser les entités du trafic urbain, d'assurer la validation des données et d'automatiser l'exécution des règles métier, tout en garantissant l'immutabilité et la traçabilité des événements enregistrés sur la blockchain.

6.1 Modélisation des Actifs Blockchain

Le chaincode implémente plusieurs types d'actifs représentant les éléments essentiels du système de gestion du trafic. Chaque actif est stocké sous forme de données structurées dans le ledger blockchain.

- **Véhicule (Vehicle)** : représente un véhicule circulant dans le réseau, avec des informations relatives à sa position, sa vitesse et son état.
- **Route (Road)** : décrit les caractéristiques des routes, telles que la limitation de vitesse, le nombre de voies et le niveau de congestion.
- **Intersection** : modélise les croisements routiers et l'état des feux de circulation.
- **Incident** : permet de gérer les événements perturbant le trafic (accidents, travaux, fermetures).
- **Violation (Infraction)** : enregistre les infractions détectées automatiquement par le système.

Cette modélisation permet une représentation fidèle et exploitable des données de trafic urbain.

6.2 Fonctions Principales du Chaincode

Le chaincode offre un ensemble complet de fonctions permettant la gestion des actifs du trafic :

- **Création (Create)** : ajout de nouveaux actifs sur la blockchain.
- **Lecture (Read)** : consultation de l'état actuel des actifs.
- **Mise à jour (Update)** : modification des informations liées aux véhicules, routes ou intersections.
- **Suppression (Delete)** : suppression contrôlée d'actifs existants.

Ces opérations garantissent une gestion flexible et sécurisée des données, tout en conservant un historique complet des modifications.

```
cat ./traffic-core/chaincode-javascript/lib/trafficContract.js | grep -E "async.*\(" | head -20
async InitLedger(ctx) {
  async CreateVehicle(ctx, id, type, currentRoad, currentIntersection, speed, direction) {
    async ReadVehicle(ctx, id) {
      async UpdateVehiclePosition(ctx, id, newRoad, newIntersection, newSpeed, newDirection) {
        async GetAllVehicles(ctx) {
          async GetVehiclesByType(ctx, type) {
            async GetVehiclesByRoad(ctx, roadId) {
              async CreateRoad(ctx, id, name, startIntersection, endIntersection, lanes, maxSpeed, length) {
                async ReadRoad(ctx, id) {
                  async UpdateRoadStatus(ctx, id, newStatus) {
                    async GetAllRoads(ctx) {
                      async CreateIntersection(ctx, id, name, latitude, longitude, connectedRoadsJSON) {
                        async ReadIntersection(ctx, id) {
                          async UpdateTrafficLight(ctx, id, newState) {
                            async GetAllIntersections(ctx) {
                              async GetNetworkStatistics(ctx) {
                                async DeleteVehicle(ctx, id) {
                                  async DeleteRoad(ctx, id) {
                                    async DeleteIntersection(ctx, id) {
```

Figure 4:Extrait des fonctions principales du smart contract Traffic Core implémenté

6.3 Règles Métier Automatisées

L'un des apports majeurs du chaincode réside dans l'implémentation de règles métier automatiques, exécutées à chaque transaction. Parmi ces règles :

- **Détection automatique des excès de vitesse** : comparaison de la vitesse du véhicule avec la limitation autorisée sur la route.

- Création automatique d'infractions : génération d'un actif Violation en cas de non-respect des règles.
- Validation des données : vérification de la cohérence et de l'existence des actifs avant toute mise à jour.

Ces mécanismes garantissent l'objectivité, la cohérence et l'auditabilité des décisions prises par le système.

6.4 Historique et Traçabilité

Hyperledger Fabric permet l'accès à l'historique complet des actifs via des fonctions dédiées du ledger. Cette fonctionnalité offre :

- Une reconstitution précise des déplacements d'un véhicule.
- Une analyse des comportements et des infractions.
- Une traçabilité totale des événements avec identifiant de transaction et horodatage certifié.

Cette traçabilité constitue un élément clé pour les audits, les analyses avancées et l'intégration future de modèles d'intelligence artificielle.

6.5 Sécurité et Validation des Transactions

Chaque interaction avec le chaincode est soumise à des mécanismes de sécurité stricts :

- Authentification des utilisateurs via certificats numériques.
- Validation des transactions par les peers.
- Enregistrement immuable des données après consensus.

Ces mécanismes assurent la fiabilité et l'intégrité du système Traffic Core.

```
2028-01-03 17:00:18.840 +01 004d INFO [chaincodeCmd] ClientWait -> txid [664b64c71d0d91533fd541d38a238b60b31fc4e59a2a66
11b1d5ca37b34786d] committed with status (VALID) at localhost:9051
2028-01-03 17:00:18.840 +01 004e INFO [chaincodeCmd] ClientWait -> txid [664b64c71d0d91533fd541d38a238b60b31fc4e59a2a66
11b1d5ca37b34786d] committed with status (VALID) at localhost:7051
chaynae@DESKTOP-BD9M118:~/hyperledger-fabric/fabric-samples/test-network$ _
```

Figure 5: Validation et commit d'une transaction du smart contract sur plusieurs nœuds orderers

7. Développement du Dashboard

Le dashboard Traffic Core constitue l'interface principale entre l'utilisateur et le système blockchain. Il permet de visualiser, superviser et contrôler en temps réel l'ensemble des éléments du trafic urbain simulé, tout en offrant une transparence totale sur les transactions enregistrées sur la blockchain. Ce dashboard joue un rôle central dans la démonstration fonctionnelle du projet.

7.1 Architecture du Dashboard

Le dashboard repose sur une architecture client–serveur assurant une communication fluide avec la blockchain Hyperledger.Fabric.

Il est composé de :

- Un frontend web permettant l'affichage et l'interaction utilisateur.
- Un backend applicatif assurant la communication avec la blockchain via le SDK Hyperledger Fabric.
- Des API REST permettant l'exécution des transactions blockchain et la récupération des données du ledger.

Cette architecture garantit une séparation claire entre la présentation, la logique applicative et la couche blockchain.

7.2 Vue d'Ensemble et Visualisation du Réseau

Le dashboard propose une vue globale du réseau routier, représenté sous forme d'une carte interactive. Les routes, intersections et véhicules sont affichés dynamiquement, avec des codes couleur permettant d'identifier l'état du trafic et le niveau de congestion.

Les véhicules sont animés en temps réel afin de refléter leur déplacement dans le réseau, tandis que l'état des feux de circulation est mis à jour automatiquement à partir des données enregistrées sur la blockchain.

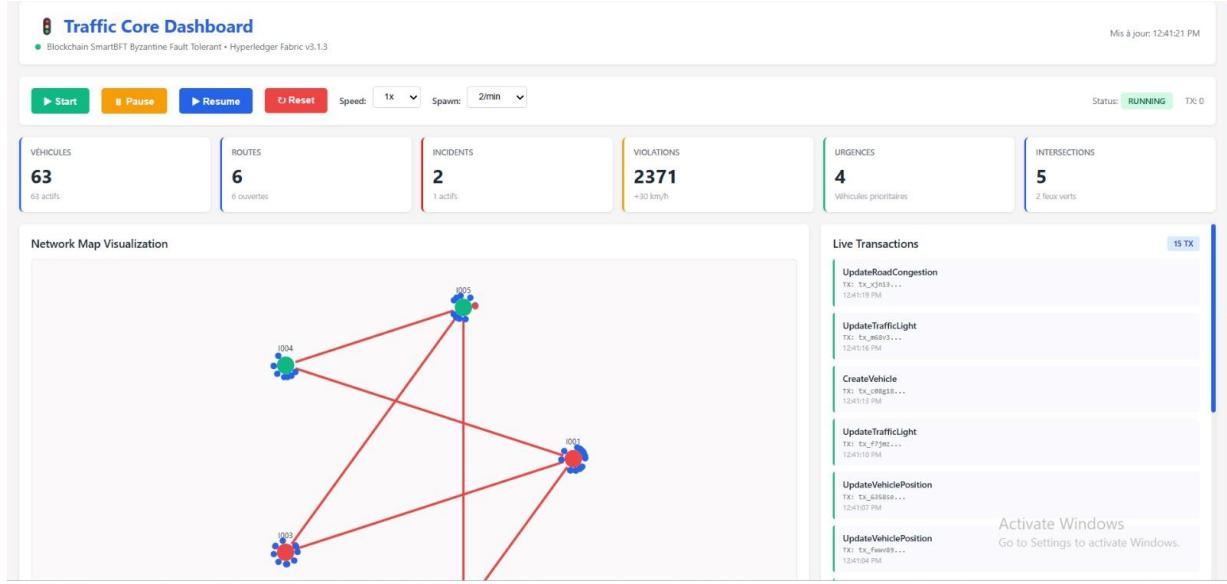


Figure 6: Vue d'ensemble du Traffic Core Dashboard avec visualisation du réseau et des transactions blockchain en temps réel

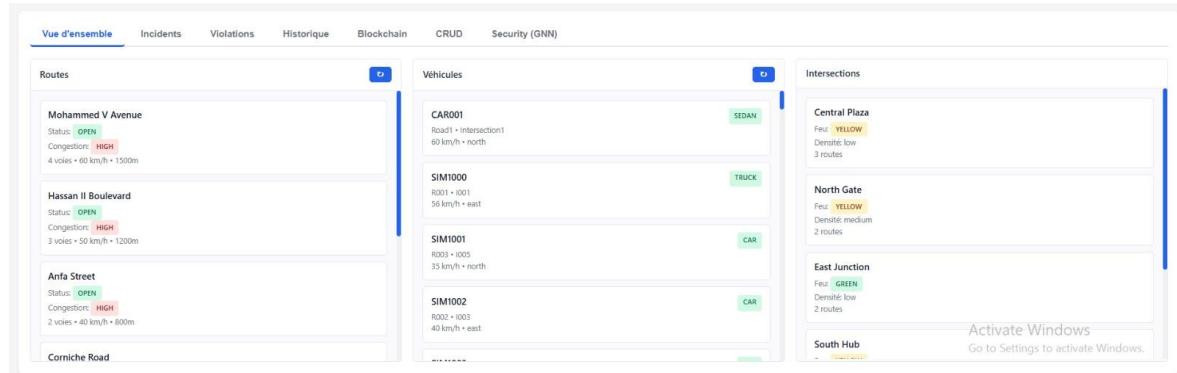


Figure 7: Visualisation des routes, véhicules et intersections dans le Traffic Core Dashboard

7.3 Gestion des Incidents et des Infractions

Une interface dédiée permet la gestion des incidents (accidents, travaux, fermetures de routes) ainsi que la consultation des infractions détectées automatiquement par le smart contract.

- Les incidents peuvent être créés, suivis et résolus via le dashboard.
- Les infractions sont affichées en temps réel, avec des détails sur le véhicule, la localisation et le type de violation.

Ces fonctionnalités renforcent la transparence et facilitent l'analyse des événements critiques.

Figure 8: Interface de gestion et de signalement des incidents de trafic

Figure 9: Interface de visualisation des violations de vitesse détectées

7.4 Interface CRUD des Actifs

Le dashboard intègre une interface CRUD complète permettant la gestion des actifs du système :

- **Création et mise à jour des véhicules.**
- **Gestion des routes (limitations de vitesse, congestion).**
- **Gestion des intersections et des feux de circulation.**

Chaque action effectuée via l'interface génère une transaction blockchain, garantissant la traçabilité de toutes les modifications.

Figure 10: Interface CRUD des actifs Traffic Core

7.5 Transactions Blockchain en Temps Réel

Le dashboard affiche un flux de transactions blockchain en temps réel, permettant de suivre l'ensemble des opérations effectuées sur le système. Chaque transaction est accompagnée de son identifiant unique, de son horodatage et de la fonction exécutée.

Cette fonctionnalité permet de visualiser concrètement le fonctionnement de la blockchain et de vérifier l'enregistrement immuable des événements de trafic.

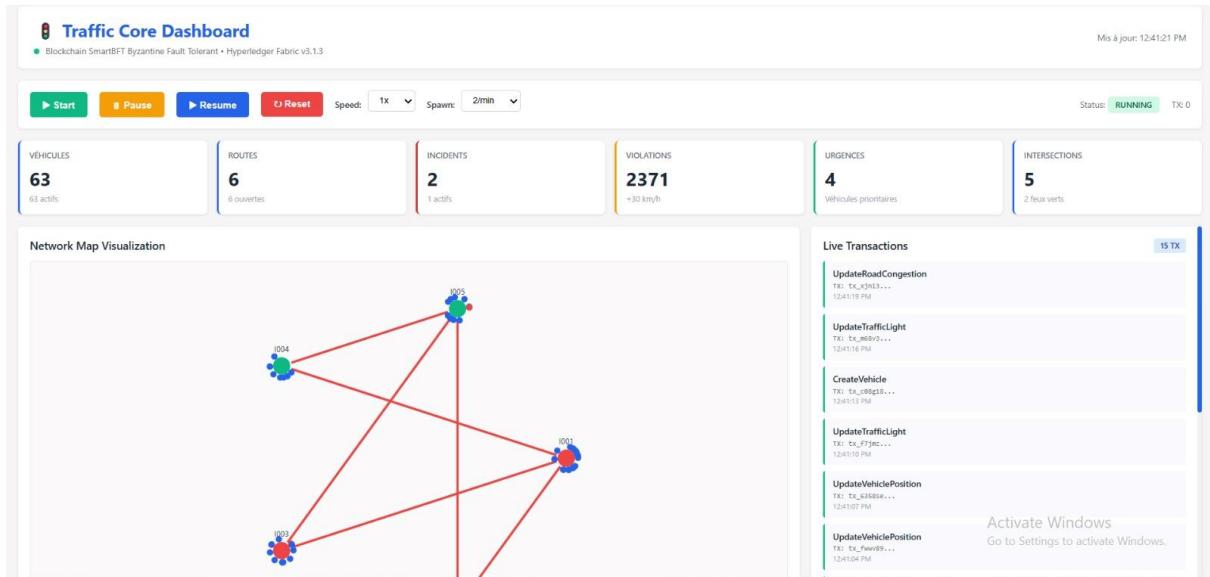


Figure 11: Feed des transactions blockchain en temps réel et visualisation du réseau

7.6 Interaction et Ergonomie

Afin d'améliorer l'expérience utilisateur, le dashboard intègre des modals interactifs permettant d'afficher les détails des véhicules, routes et intersections. Ces modals remplacent les alertes classiques et offrent une présentation claire et structurée des informations.

L'interface propose également des contrôles de simulation permettant de démarrer, mettre en pause, reprendre ou réinitialiser la simulation du trafic, offrant ainsi une interaction complète avec le système.

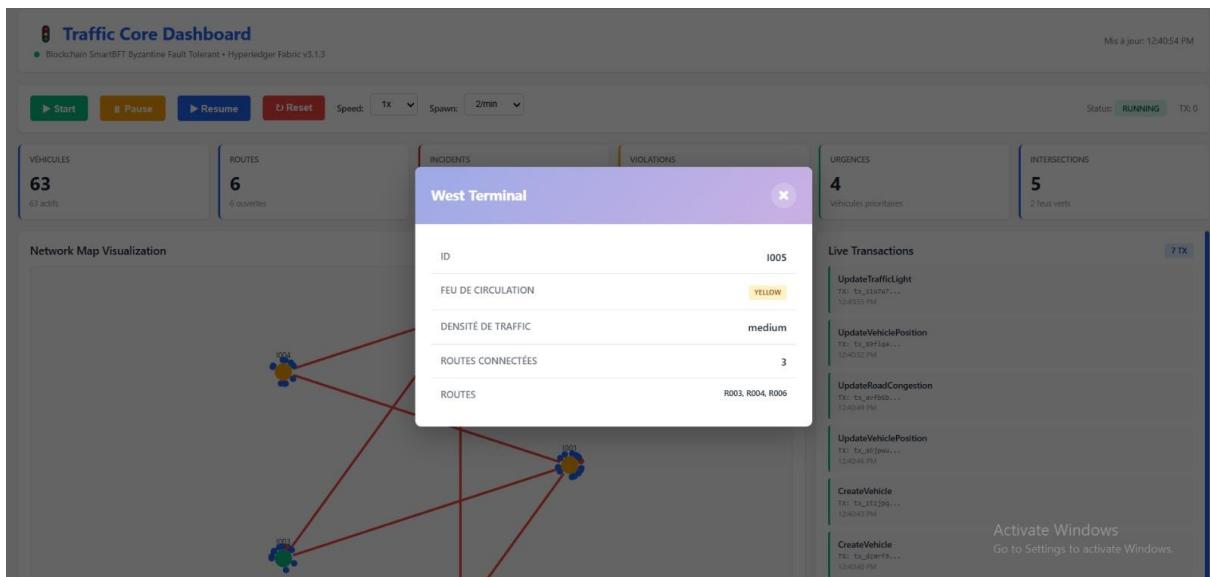


Figure 12: Modal interactif affichant les détails d'une intersection (West Terminal)

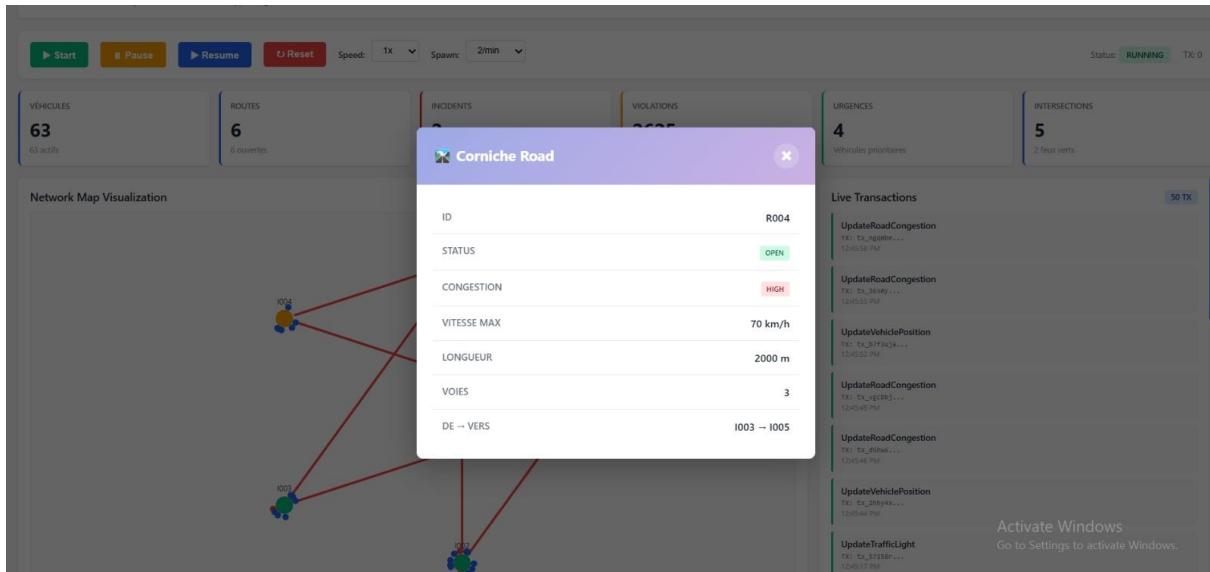


Figure 13: Modal interactif affichant les détails d'une route (Corniche Road)

8. Moteur de Simulation

Le moteur de simulation constitue un composant central du projet Traffic Core, permettant de reproduire des scénarios réalistes de trafic urbain et de démontrer le fonctionnement du système blockchain dans des conditions proches du réel. Il génère automatiquement des événements de trafic qui sont ensuite validés et enregistrés sous forme de transactions sur la blockchain.

8.1 Architecture du Moteur de Simulation

Le moteur de simulation est implémenté au niveau de la couche applicative, côté serveur, afin d'assurer une cohérence globale du système et d'éviter les conflits liés à une exécution distribuée côté client. Il fonctionne de manière continue selon une boucle de simulation, générant des événements à intervalles réguliers. Chaque action simulée (déplacement de véhicule, changement d'état, mise à jour de congestion) déclenche une transaction blockchain, garantissant la traçabilité complète de la simulation.

8.2 Algorithme de Déplacement des Véhicules

Le moteur de simulation implémente un algorithme permettant de simuler le déplacement dynamique des véhicules à travers le réseau routier. À chaque cycle de simulation, les étapes suivantes sont exécutées :

1. Récupération de l'état actuel du véhicule.
2. Vérification de la route et de l'intersection associées.
3. Analyse de l'état du feu de circulation.
4. Calcul de la nouvelle position et du statut du véhicule.
5. Soumission d'une transaction de mise à jour sur la blockchain.

Cet algorithme permet de reproduire des comportements réalistes tels que l'arrêt aux feux rouges ou la variation de vitesse selon la congestion.

8.3 Gestion Automatique de la Congestion

Le moteur de simulation calcule dynamiquement le niveau de congestion des routes en fonction du nombre de véhicules présents et de leurs caractéristiques. Le niveau de congestion est ajusté automatiquement et enregistré sur la blockchain lorsqu'un seuil est dépassé.

Cette approche permet de refléter en temps réel l'état du trafic et d'observer l'impact des variations de densité sur la circulation urbaine.

8.4 Détection des Infractions en Temps Réel

Le moteur de simulation interagit avec le smart contract afin de détecter automatiquement les infractions, notamment les excès de vitesse. Lorsqu'un véhicule dépasse la limitation autorisée sur une route donnée, une transaction est générée et une infraction est enregistrée sur la blockchain.

Cette automatisation garantit l'objectivité et la fiabilité du processus de détection des infractions, tout en illustrant l'intérêt de la blockchain pour les systèmes de contrôle du trafic.

```
:haymae@DESKTOP-BD9M118:~/hyperledger-fabric/fabric-samples/test-network$ curl http://localhost:3000/api/blockchain/latest
{"height":50095,"latestBlockNumber":50094,"currentBlockHash":"SprBBHar40PYeGmDYd19Ly/Hcs0+W3Fg6Urb3ZMztSk=","previousBlockHash"
"/QzhFCdWxuzGluUkbJ3RgtMOEl3bVybYI7/zzANs="}haymae@DESKTOP-BD9M118:~/hyperledger-fabric/fabric-samples/test-network$
```

Figure 14:Infraction détectée lors de la simulation

The screenshot shows a table with three rows of detected violations:

Violation ID	Route	Limit	Speed
SIM1005	Coroniche Road (8000)	Limite: 70 • Réel: 107	+37 KM/H
SIM1006	Mers Sultan Road (8000)	Limite: 40 • Réel: 71	+31 KM/H
SIM1005	Coroniche Road (8000)	Limite: 70 • Réel: 106	+36 KM/H

Figure 15:Infraction détectée lors de la simulation Dashboard

8.5 Contrôle et Paramétrage de la Simulation

Le moteur de simulation peut être contrôlé directement depuis le dashboard. L'utilisateur dispose de plusieurs options :

- Démarrage et arrêt de la simulation.
- Mise en pause et reprise.
- Réinitialisation complète de l'environnement.
- Ajustement de la vitesse de simulation.

Ces fonctionnalités facilitent l'expérimentation de différents scénarios de trafic et l'analyse du comportement du système.

The dashboard displays various metrics and controls:

- Controls: Start, Pause, Resume, Reset, Speed (Tx), Spawns (2/min), Status (RUNNING), TX: 0.
- Metrics: VÉHICULES (101), ROUTES (6), INCIDENTS (2), VIOLATIONS (4544 +30 km/h), URGENCES (7), INTERSECTIONS (5).

Figure 16:Contrôles de simulation disponibles sur le dashboard

8.6 Apport du Moteur de Simulation

Le moteur de simulation joue un rôle essentiel dans la validation du projet Traffic Core. Il permet de tester le système sous différentes charges, de vérifier la robustesse du consensus blockchain et de démontrer l'enregistrement immuable des événements de trafic. Il constitue ainsi un outil clé pour l'évaluation des performances et la démonstration académique du système.

9. Résultats et Validation

Cette section présente les résultats obtenus à l'issue de l'implémentation du système Traffic Core et valide le bon fonctionnement de l'architecture blockchain, du moteur de simulation et du dashboard. Les

expérimentations réalisées permettent d'évaluer la fiabilité, la cohérence et la robustesse du système dans un contexte de Smart City décentralisée.

9.1 Validation du Réseau Blockchain

Le réseau Hyperledger Fabric configuré avec le consensus SmartBFT a démontré un fonctionnement stable et fiable tout au long des tests. Les transactions soumises par la couche applicative ont été correctement ordonnancées et validées par les nœuds orderers, puis enregistrées de manière immuable dans le ledger. Les tests de tolérance aux pannes ont montré que le système reste opérationnel même en cas de défaillance d'un nœud participant au consensus, confirmant ainsi la résilience du mécanisme SmartBFT.

```
2026-01-03 00:29:14.109 +01 0045 INFO [grpc] AddTraceEvent -> [core] [Channel #13]Channel Connectivity change to CONNECTING
2026-01-03 00:29:14.109 +01 0046 INFO [grpc] AddTraceEvent -> [core] [Channel #13 SubChannel #14]Subchannel created
2026-01-03 00:29:14.109 +01 0047 INFO [grpc] AddTraceEvent -> [core] [Channel #13]Channel exiting idle mode
2026-01-03 00:29:14.109 +01 0048 INFO [grpc] AddTraceEvent -> [core] [Channel #13 SubChannel #14]Subchannel Connectivity change to CONNECTING
2026-01-03 00:29:14.109 +01 0049 INFO [grpc] AddTraceEvent -> [core] [Channel #13 SubChannel #14]Subchannel picking address "localhost:7050" to connect
2026-01-03 00:29:14.113 +01 004a INFO [grpc] AddTraceEvent -> [core] [Channel #13 SubChannel #14]Subchannel Connectivity change to READY
2026-01-03 00:29:14.113 +01 004b INFO [grpc] Infof -> [pick-first-leaf-lb] [pick-first-leaf-lb 0xc0003b26c0] SubConn 00044a5f0 reported connectivity state READY and the health listener is disabled. Transitioning SubConn to READY
2026-01-03 00:29:14.113 +01 004c INFO [grpc] AddTraceEvent -> [core] [Channel #13]Channel Connectivity change to CONNECTED
2026-01-03 00:29:14.218 +01 004d INFO [chaincodeCmd] ClientWait -> txid [239dbefe63dce876c92462de2e63b0796c37c3ca7d75800e5e63b06] committed with status (VALID) at localhost:9051
2026-01-03 00:29:14.222 +01 004e INFO [chaincodeCmd] ClientWait -> txid [239dbefe63dce876c92462de2e63b0796c37c3ca7d75800e5e63b06] committed with status (VALID) at localhost:7051
chaymae@DESKTOP-BD9M118:~/hyperledger-fabric/fabric-samples/test-network$
```

Figure 17:Logs de validation et de commit des transactions blockchain via le consensus SmartBFT

9.2 Transactions Blockchain en Temps Réel

Le dashboard affiche un flux continu de transactions blockchain générées par la simulation du trafic et les actions de l'utilisateur. Chaque transaction est associée à un identifiant unique et à un horodatage certifié, garantissant une traçabilité complète des événements.

Les principales transactions observées incluent :

- La mise à jour des positions des véhicules.
- La modification des niveaux de congestion.
- Les changements d'état des feux de circulation.
- La création automatique des infractions.



Figure 18:Feed des transactions blockchain en temps réel

9.3 Validation Fonctionnelle du Système

Les scénarios de simulation exécutés ont permis de valider l'ensemble des fonctionnalités du système :

- Déplacement dynamique des véhicules dans le réseau routier.
- Mise à jour cohérente de l'état des routes et intersections.
- Détection automatique et fiable des infractions.
- Synchronisation en temps réel entre la blockchain et le dashboard.

Ces résultats confirment la bonne intégration entre le moteur de simulation, le smart contract et l'interface de visualisation.

9.4 Traçabilité et Auditabilité des Données

L'un des apports majeurs du système réside dans la capacité à consulter l'historique complet des actifs de trafic. Les fonctions d'historique de la blockchain permettent de reconstituer précisément les événements passés, tels que les déplacements d'un véhicule ou l'évolution de la congestion d'une route.

Cette traçabilité renforce l'auditabilité du système et constitue une base fiable pour des analyses avancées et des décisions éclairées.

Figure 19: Historique d'un actif de trafic extrait du ledger

9.5 Gestion des Conflits et Cohérence

Lors des simulations à haute fréquence, des conflits de concurrence de type MVCC (Multi-Version Concurrency Control) ont été observés. Ces conflits, gérés automatiquement par la blockchain, empêchent les modifications concurrentes incohérentes et garantissent l'intégrité des données.

La gestion correcte de ces conflits démontre le bon fonctionnement des mécanismes de contrôle de concurrence et la robustesse du système face à une charge élevée.

9.6 Synthèse des Résultats

Les résultats obtenus confirment que le système Traffic Core répond aux objectifs initiaux du projet. L'architecture blockchain assure la sécurité, la transparence et la fiabilité des données de trafic, tandis que le moteur de simulation et le dashboard permettent une démonstration claire et interactive du fonctionnement du système. Cette validation met en évidence l'intérêt de la blockchain pour les applications Smart City critiques.

10. Comparaison des Mécanismes de Consensus

10.1 Objectif de la comparaison

L'objectif de cette section est de comparer les mécanismes de consensus utilisés ou étudiés dans le cadre du projet *Traffic Core*, conformément aux exigences du sujet.

Le consensus SmartBFT a été implémenté et utilisé dans le système afin d'assurer la validation et l'ordonnancement des transactions blockchain.

À des fins d'analyse comparative, ce mécanisme est comparé au consensus Raft, utilisé comme mécanisme de référence dans Hyperledger Fabric.

La comparaison repose sur des métriques bien déterminées, issues à la fois des observations réalisées lors de l'implémentation de SmartBFT et des propriétés théoriques des protocoles de consensus décrites dans la littérature.

10.2 Présentation des mécanismes de consensus

10.2.1 Consensus SmartBFT

SmartBFT (Smart Byzantine Fault Tolerant) est un mécanisme de consensus byzantin intégré aux versions récentes de Hyperledger Fabric.

Il repose sur une variante optimisée du protocole PBFT (Practical Byzantine Fault Tolerance), permettant au réseau de continuer à fonctionner même en présence de nœuds défaillants ou malveillants.

Dans le cadre du projet Traffic Core, SmartBFT a été déployé avec plusieurs nœuds orderers participant au consensus. Les transactions sont validées de manière distribuée et enregistrées de façon immuable dans le ledger blockchain, garantissant un haut niveau de sécurité et de fiabilité.

10.2.2 Consensus Raft

Raft est un mécanisme de consensus Crash Fault Tolerant (CFT), historiquement utilisé comme consensus par défaut dans Hyperledger Fabric.

Il repose sur une architecture leader–followers, dans laquelle un leader est élu pour proposer l'ordre des transactions.

Raft offre de bonnes performances en termes de latence et de débit, mais ne protège pas contre les comportements malveillants des nœuds. Il est principalement adapté aux environnements de confiance, où les défaillances sont supposées être accidentnelles plutôt qu'intentionnelles.

10.3 Métriques de comparaison

La comparaison des deux mécanismes de consensus est réalisée à partir des métriques suivantes :

- Type de fautes supportées
- Tolérance aux pannes
- Niveau de sécurité
- Latence de validation
- Débit de transactions
- Scalabilité
- Complexité du consensus
- Adaptation aux systèmes Smart City

Les métriques relatives à SmartBFT sont issues des observations expérimentales réalisées dans le projet, tandis que celles de Raft sont basées sur ses propriétés théoriques connues.

10.4 Analyse comparative

Métrique	SmartBFT	Raft
Type de consensus	Byzantine Fault Tolerant (PBFT modifié)	Crash Fault Tolerant
Fautes supportées	Crash + Byzantines	Crash uniquement
Tolérance aux pannes	$n \geq 3f + 1$	$n \geq 2f + 1$
Sécurité	Très élevée	Moyenne
Latence	Plus élevée	Faible
Débit (Throughput)	Moyen	Élevé
Scalabilité	Moyenne	Élevée
Complexité	Élevée	Faible
Adaptation Smart City	Très bonne	Limitée

Les résultats observés montrent que SmartBFT offre un niveau de sécurité et de résilience supérieur, au prix d'une complexité et d'une latence légèrement plus élevées.

Raft, en revanche, privilégie la performance, mais reste vulnérable aux comportements malveillants.

10.5 Discussion et choix du consensus

Dans le contexte d'un système de gestion du trafic urbain, la fiabilité, la sécurité et la tolérance aux comportements non fiables constituent des critères essentiels. Les observations réalisées lors de l'implémentation de SmartBFT dans le projet Traffic Core ont montré une validation correcte des transactions, un fonctionnement stable du réseau et une traçabilité complète des événements de trafic.

Bien que Raft présente de meilleures performances en termes de latence et de débit, son incapacité à gérer les fautes byzantines limite son utilisation dans des systèmes critiques ouverts à plusieurs acteurs.

10.6 Synthèse

La comparaison réalisée met en évidence que le consensus SmartBFT constitue un choix plus adapté aux systèmes Smart City critiques, où la sécurité et la confiance sont prioritaires. Raft reste un mécanisme performant et pertinent pour des environnements contrôlés, mais ne répond pas aux exigences de résilience avancée imposées par la gestion décentralisée du trafic urbain.

Cette analyse confirme que l'utilisation de SmartBFT dans le projet Traffic Core est cohérente avec les objectifs initiaux du projet et les contraintes des systèmes urbains intelligents.

11. Conclusion

Ce rapport a présenté la conception et l'implémentation de Traffic Core, un système intelligent de gestion du trafic urbain basé sur une blockchain permissionnée Hyperledger Fabric, s'inscrivant dans la vision des Smart Cities décentralisées. L'objectif principal du projet était de démontrer l'apport de la blockchain pour garantir la fiabilité, la traçabilité et la transparence des données de trafic dans un environnement distribué et critique. L'architecture proposée repose sur une séparation claire entre la couche blockchain, la couche applicative et la couche de visualisation. L'utilisation du consensus SmartBFT a permis de mettre en place un réseau robuste, tolérant aux fautes byzantines et capable d'assurer la continuité du service même en cas de défaillance partielle des nœuds. Le développement du smart contract (chaincode) a permis de modéliser les principaux actifs du trafic urbain et d'automatiser des règles métier essentielles, telles que la détection des infractions, tout en garantissant l'immutabilité des enregistrements.

Le dashboard interactif et le moteur de simulation ont joué un rôle central dans la validation du système, en offrant une visualisation en temps réel du réseau routier, des transactions blockchain et des événements simulés. Ces outils ont permis de démontrer concrètement l'intégration entre la blockchain et les services de gestion du trafic, ainsi que l'impact du mécanisme de consensus sur les performances et la stabilité du système. La comparaison de plusieurs mécanismes de consensus a mis en évidence les compromis entre performance, sécurité et tolérance aux pannes. Les résultats obtenus confirment que le consensus SmartBFT constitue un choix particulièrement adapté aux systèmes Smart City, où la fiabilité et la confiance sont des critères déterminants, tandis que des consensus plus simples peuvent être envisagés pour des scénarios moins critiques. En perspective, ce projet ouvre la voie à de nombreuses extensions, notamment l'intégration de techniques d'intelligence artificielle pour la prédition et l'optimisation du trafic, l'interconnexion avec d'autres services urbains (stationnement, transport public, énergie) et l'amélioration des performances à grande échelle. Traffic Core démontre ainsi que la blockchain peut constituer une infrastructure de confiance essentielle pour les systèmes urbains intelligents du futur.