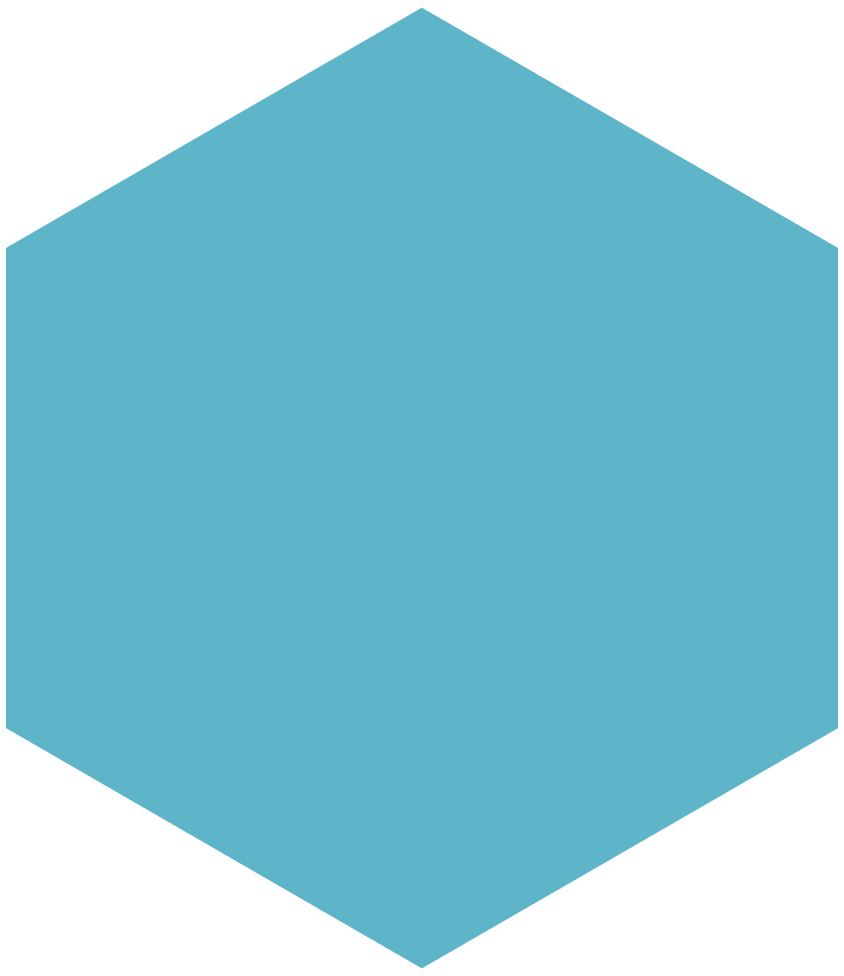


Digital HaïTian Gourde



MASTER 2 MIAGE MBDS

2019-2020

Auteurs :

Thomas BEATINI  
Arnaud FERNANDEZ  
Chloé MACCARINELLI  
Cédric ORTEGA

Tuteurs :

Gabriel MOPOLO-MOKE

Alexandre MAISONOBE

Gaëtan LESCOUFLAIR

Table des matières

[I. Liste des figures 3](#_Toc33444956)

[II. Index 3](#_Toc33444957)

[III. Résumé 4](#_Toc33444958)

[IV. Abstract 4](#_Toc33444959)

[V. Introduction 5](#_Toc33444960)

[VI. Présentation des acteurs 6](#_Toc33444961)

[III.1 Présentation de l’Université Côte d’Azur (MBDS) 6](#_Toc33444962)

[III.1.1 Mobiquité, Base de Données/Big Data et Intégration de Systèmes (MBDS) 6](#_Toc33444963)

[III.2 Présentation de la BRH (Banque de la République d’Haïti) 6](#_Toc33444964)

[VII. Etat de l’art 7](#_Toc33444965)

[III. 1. La blockchain 7](#_Toc33444966)

[III. 1.1. Définition 7](#_Toc33444967)

[III.1.2 Les règles de consensus 7](#_Toc33444968)

[III.2 Les types de blockchains 8](#_Toc33444969)

[III.2.1 PUBLIQUE 8](#_Toc33444970)

[III.2.2 PRIVE ("DISTRIBUTED LEDGER TECHNOLOGY" (DLT)) 8](#_Toc33444971)

[III.2.3 Critères relatifs à notre projet 10](#_Toc33444972)

[III.3 La crypto-monnaie 10](#_Toc33444973)

[III.3.1 Définition 10](#_Toc33444974)

[III.3.2 Les différentes crypto-monnaies 11](#_Toc33444975)

[III.3.3 Critères relatifs à notre projet 11](#_Toc33444976)

[VIII. Etude de l’existant 12](#_Toc33444977)

[IV.1 L’environnement 12](#_Toc33444978)

[IV.1.1 Openchain 12](#_Toc33444979)

[IV.1.2 Hyperledger Fabric, une alternative à Openchain 13](#_Toc33444980)

[IV.1.3 MySQL – API REST NodeJs – Angular 7 15](#_Toc33444981)

[IX. Démarche projet 16](#_Toc33444982)

[VI.1 Gestion de projet 16](#_Toc33444983)

[VI.2 Contraintes, outils et risques 16](#_Toc33444984)

[VI.2.1 Contraintes 16](#_Toc33444985)

[VI.2.2 Outils 17](#_Toc33444986)

[VI.2.3 Risques 17](#_Toc33444987)

[VI.3 Planning 18](#_Toc33444988)

[VI.4 Budget 18](#_Toc33444989)

[X. Exigences fonctionnelles 19](#_Toc33444990)

[VIII.1 Les acteurs de niveau fonctionnel 19](#_Toc33444991)

[VII.2 Les cas d’utilisations 20](#_Toc33444992)

[XI. Exigences non-fonctionnelles 20](#_Toc33444993)

[IX.1 Utilisabilité 20](#_Toc33444994)

[IX.2 Performances 20](#_Toc33444995)

[IX.3 Robustesse 20](#_Toc33444996)

[IX.4 Sécurité 20](#_Toc33444997)

[IX.5 Maintenabilité, évolutivité 20](#_Toc33444998)

[XII. Interfaces détaillées 21](#_Toc33444999)

[XIII. Architectures 21](#_Toc33445000)

[Détails de l’architecture  3 tiers 21](#_Toc33445001)

[Détails de l’architecture technique 21](#_Toc33445002)

[XIV. Les Améliorations 22](#_Toc33445003)

[V.1 Sécurité 22](#_Toc33445004)

[V.2 Client Web et mobile 22](#_Toc33445005)

[XV. Dockerisation 22](#_Toc33445006)

[XVI. Perspectives 23](#_Toc33445007)

[V.2 Client Web et mobile 23](#_Toc33445008)

[V.2.1 Framework Flutter 23](#_Toc33445009)

[V.2.2 Langage Dart 23](#_Toc33445010)

[XVII. Conclusion 23](#_Toc33445011)

[XVIII. Webographie et Bibliographie 23](#_Toc33445012)

# Liste des figures

[**Figure 1 : Fonctionnement d'une blockchain** 7](file:///C:\Users\Perfection\Documents\DIGITAL_GOURDE\Github\projet\doc\RAPPORT_DIGITAL_GOURDE.docx#_Toc33444851)

[**Figure 2 : Les différentes règles de consensus (La Blockchain- Panorama des technologies existantes © 2017 Deloitte SAS)** 8](file:///C:\Users\Perfection\Documents\DIGITAL_GOURDE\Github\projet\doc\RAPPORT_DIGITAL_GOURDE.docx#_Toc33444852)

[**Figure 3  : Différences entre blockchain privée et publique (La Blockchain- Panorama des technologies existantes © 2017 Deloitte SAS)** 9](file:///C:\Users\Perfection\Documents\DIGITAL_GOURDE\Github\projet\doc\RAPPORT_DIGITAL_GOURDE.docx#_Toc33444853)

[**Figure 4 : Qu'est-ce que la crypto-monnaie ?** 10](file:///C:\Users\Perfection\Documents\DIGITAL_GOURDE\Github\projet\doc\RAPPORT_DIGITAL_GOURDE.docx#_Toc33444854)

[**Figure 5: Des possibilités technologiques révolutionnaires** 11](file:///C:\Users\Perfection\Documents\DIGITAL_GOURDE\Github\projet\doc\RAPPORT_DIGITAL_GOURDE.docx#_Toc33444855)

[**Figure 6 : Schématisation d'un bloc caché** 14](file:///C:\Users\Perfection\Documents\DIGITAL_GOURDE\Github\projet\doc\RAPPORT_DIGITAL_GOURDE.docx#_Toc33444856)

[**Figure 7 : Architecture Hyperledger Fabric** 14](file:///C:\Users\Perfection\Documents\DIGITAL_GOURDE\Github\projet\doc\RAPPORT_DIGITAL_GOURDE.docx#_Toc33444857)

[**Figure 8 : Fonctionnement PoC** 15](file:///C:\Users\Perfection\Documents\DIGITAL_GOURDE\Github\projet\doc\RAPPORT_DIGITAL_GOURDE.docx#_Toc33444858)

[**Figure 9 : Kanban DigitalGourde** 16](#_Toc33444859)

[**Figure 10 : Diagramme de Gantt** 18](#_Toc33444860)

[**Figure 11 : Les acteurs** 19](#_Toc33444861)

[**Figure 12 : Architecture 3 tiers** 21](#_Toc33444862)

[**Figure 13 : Architecture technique** 21](#_Toc33444863)

# Index

API : Application Programming Interface

Cryptographie : C’est une des disciplines de la cryptologie s'attachant à protéger des messages (assurant confidentialité, authenticité et intégrité) en s'aidant souvent de secrets ou clés

Registre distribué : (aussi appelé registre partagé ; en anglais, distributed ledger ou shared ledger) est un registre simultanément enregistré et synchronisé sur un réseau d'ordinateurs, qui évolue par l'addition de nouvelles informations préalablement validées par l'entièreté du réseau et destinées à ne jamais être modifiées ou supprimées. Un registre distribué n'a ni administrateur central ni stockage de données centralisé.

# Résumé

Ce projet consiste à la réalisation d’une cryptomonnaie sociale, monnaie virtuelle d’échange, sécurisée permettant de faciliter et révolutionner le payement et les transferts d’argent.

Il prend principalement en compte ceux qui sont non bancarisés de la population Haïtienne. Cette monnaie pourrait être utilisée dans plusieurs domaines tels que l’agriculture, l’éducation, la santé, etc…

Une technologie appelée blockchain qui, par ses propriétés intrinsèques, apporte une facilité et rapidité des échanges. Cette technologie sera la base de notre moyen d’échange car elle offre un environnement de confiance.

En effet la blockchain est une base de données répliquée, décentralisée donc il n’y pas une autorité centrale. Une fois une transaction validée il est impossible de l’effacer et toute tentative de falsification est rendue extrêmement complexe. De plus il y a l’anonymat des utilisateurs réalisé par la [cryptographie](#_Index).

La blockchain créée sera privée car les transactions seront vérifiées et validées par ceux qui sont autorisés à se connecter sur la blockchain. Cela permet une plus grande efficacité, évolutivité, une faible consommation énergétique. De plus les transactions sur une blockchain privée sont validées beaucoup plus rapidement.

# Abstract

This project consists of the creation of a social cryptocurrency, virtual exchange currency, secure to facilitate and revolutionize the payment and the transfer of money. It mainly considers those who are unbanked of the Haitian population. This currency could be used in several sectors such as agriculture, education, health, etc…

A technology called blockchain which, by its intrinsic properties, brings a facility and rapidity of the exchanges, security of the transactions and an environment of confidence, will be the base of our medium of exchange.

Indeed, the blockchain is a replicated database, decentralized so there is no central authority. Once a transaction is validated it cannot be erased and any attempt to falsify is rendered extremely complex. In addition, there is the anonymity of the users realized by the cryptography.

The created blockchain will be private because the transactions will be verified and validated by those who can connect to the blockchain. This allows for greater efficiency, scalability, low energy consumption. In addition, transactions on a private blockchain are validated much more quickly.

# Introduction

La monnaie nationale d’Haïti, la gourde (HTG), de nature très volatile et en dépréciation continue depuis les vingt dernières années le gouvernement Haïtien en collaboration avec la Banque de la République d’Haïti (BRH) cherchent une solution pour relancer l’économie du pays.

La solution proposée par le projet Digital Gourde (DHTG), est la mise en place d’une cryptomonnaie d’état du même nom qui sera gérée, supervisée et distribuée par la BRH et mis à disposition des institutions financières locales.

Les objectifs à long terme du projet sont de :

1. Réduire puis supprimer le coût de fabrique et d’importation de la monnaie physique

2. Permettre aux personnes non-bancarisées d’Haïti de l’être, sans risques pour les institutions financières

3. Augmenter le taux de change de la monnaie nationale pour combler le déficit financier du pays

L’avantage majeur de l’utilisation des cryptomonnaies dans un cas tel que celui-ci est que la création de la monnaie est instantanée et sans frais. Aussi, chaque opération utilisant cette monnaie (distribution, paiement, virement, etc.) aura un coût extrêmement faible et sera instantanée étant donné que la blockchain sur laquelle elle se base utilise une architecture client-serveur plutôt qu’un système de Proof of Work, contrairement aux cryptomonnaies tels que le Bitcoin.

# Présentation des acteurs

## III.1 Présentation de l’Université Côte d’Azur (MBDS)

L’université Nice Côte d’Azur est une université française pluridisciplinaire, créée en 19653. L'université est implantée à Nice et dans le département des Alpes-Maritimes.

## III.1.1 Mobiquité, Base de Données/Big Data et Intégration de Systèmes (MBDS)

La spécialité MBDS est la proposition d'une restructuration et d'intégration de cette spécialité au sein de la mention MIAGE de Nice. En effet, l'informatique d'entreprise mise en avant par MIAGE est le thème principal naturel de cette spécialité. Dans l'intégration de cette spécialité à la mention MIAGE, nous avons complété la partie très technique du MBDS historique par la double compétence en gestion et par des compétences transverses, indispensables dans les cursus MIAGE.

L'originalité du MBDS est de développer le thème de la mobiquité, source de nombreux travaux de recherche ou produit des entreprises en informatique.

Les objectifs scientifiques se situent à deux niveaux :

• Former des développeurs de Systèmes d'information du futur des bases de données (administration et tuning) au Big Data (gestion et architecture ; Hadoop)

• Intégrer les résultats les plus avancés en recherche sur les services web, les développements sur Smartphone (HTML5 comme code natif) et l’analyse de données (langage R)

La spécialité MBDS proposée est indifférenciée (Professionnelle et Recherche).

## III.2 Présentation de la BRH (Banque de la République d’Haïti)

Nous présentons ici la BRH car c’est elle qui sera le gestionnaire de la gourde électronique. La Banque de la République d'Haïti ou BRH est une institution financière qui joue le rôle de banque centrale pour la République d'Haïti. À travers son Conseil d'Administration, elle a le pouvoir d’énoncer, de diriger et de superviser la politique monétaire. Elle autorise l’impression de billets et la frappe de monnaie et détermine les volumes des émissions en accord avec la loi.

La loi du 17 août 1979 portant création de la BRH dispose en son article 6 que celle-ci est dirigée par un Conseil d’Administration dont les membres sont nommés pour une période de trois ans renouvelables par arrêté du Président de la République. Conformément aux dispositions de la Constitution de 1987, ils sont ratifiés par le Sénat de la République.

La législation en vigueur assigne quatre rôles fondamentaux à la BRH, lesquels peuvent être énoncés comme suit :

• Défendre la valeur interne et externe de la monnaie nationale

• Assurer l’efficacité, le développement et l’intégrité du système de paiements

• Assurer la stabilité du système financier

• Agir comme banquier, caissier et agent fiscal de l’État

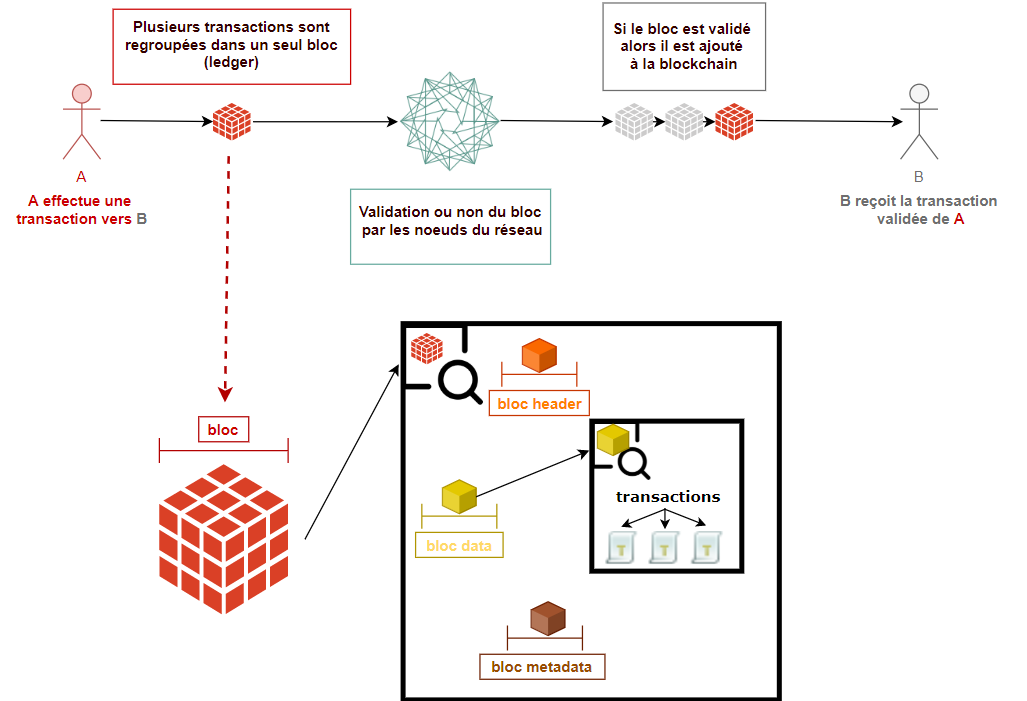
# Etat de l’art

## III. 1. La blockchain

### III. 1.1. Définition

*« La blockchain est une technologie de stockage et de transmission d’informations, transparente, sécurisée, et fonctionnant sans organe central de contrôle.*

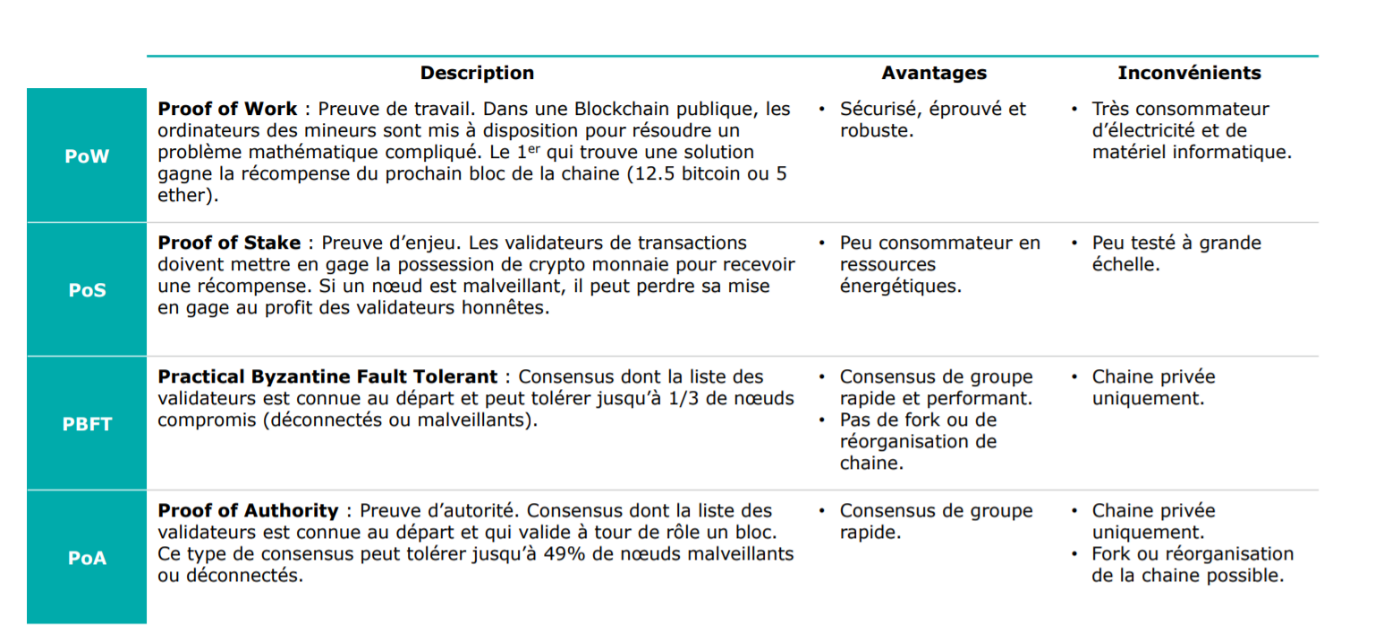
*Par extension, une blockchain constitue une base de données qui contient l’historique de tous les échanges effectués entre ses utilisateurs depuis sa création. Cette base de données est sécurisée et distribuée : elle est partagée par ses différents utilisateurs, sans intermédiaire, ce qui permet à chacun de vérifier la validité de la chaîne. »* *(Définition de Blockchain France)*



**Figure 1 : Fonctionnement d'une blockchain**

### III.1.2 Les règles de consensus

Les règles de consensus désignent le protocole selon lequel un individu sera choisi pour ajouter son bloc à la blockchain. Ce sont les règles de consensus qui assurent la sécurité du réseau et dissuadent la falsification des blocs.



**Figure 2 : Les différentes règles de consensus (La Blockchain- Panorama des technologies existantes © 2017 Deloitte SAS)**

## III.2 Les types de blockchains

### III.2.1 PUBLIQUE

La blockchain dites « publique » est la blockchain d’origine, elle est totalement décentralisée. C’est-à-dire que tout le monde peut lire, effectuer des transactions et participer au processus de validation de blocs, qui seront ou non ajoutés à la blockchain. Tous les acteurs sont égalitaires vis-à-vis de leur participation dans le réseau.

Exemples : Bitcoin, Ethereum, Litecoin, etc.

### III.2.2 PRIVE ("DISTRIBUTED [LEDGER](#_Index) TECHNOLOGY" (DLT))

Depuis 2015 les blockchains partiellement décentralisées ou centralisées se développent et offrent de nombreux avantages :

* Gouvernance simplifiée
* Acteurs connus
* Coûts réduits
* Rapidité
* Confidentialité

Exemples : HYPERLEDGER, Corda, MONEX, B3i, R3, Labchain, etc.

#### BLOCKCHAIN DE CONSORTIUM (SEMI-PRIVE)

La blockchain dites « de consortium » limite et sélectionne le nombre de participants du processus d’approbation et la règle de la majorité ne s’impose pas. La lecture des blocs est publique, réservée aux participants ou hybride. Ce type de blockchain est principalement utilisé dans le secteur bancaire.

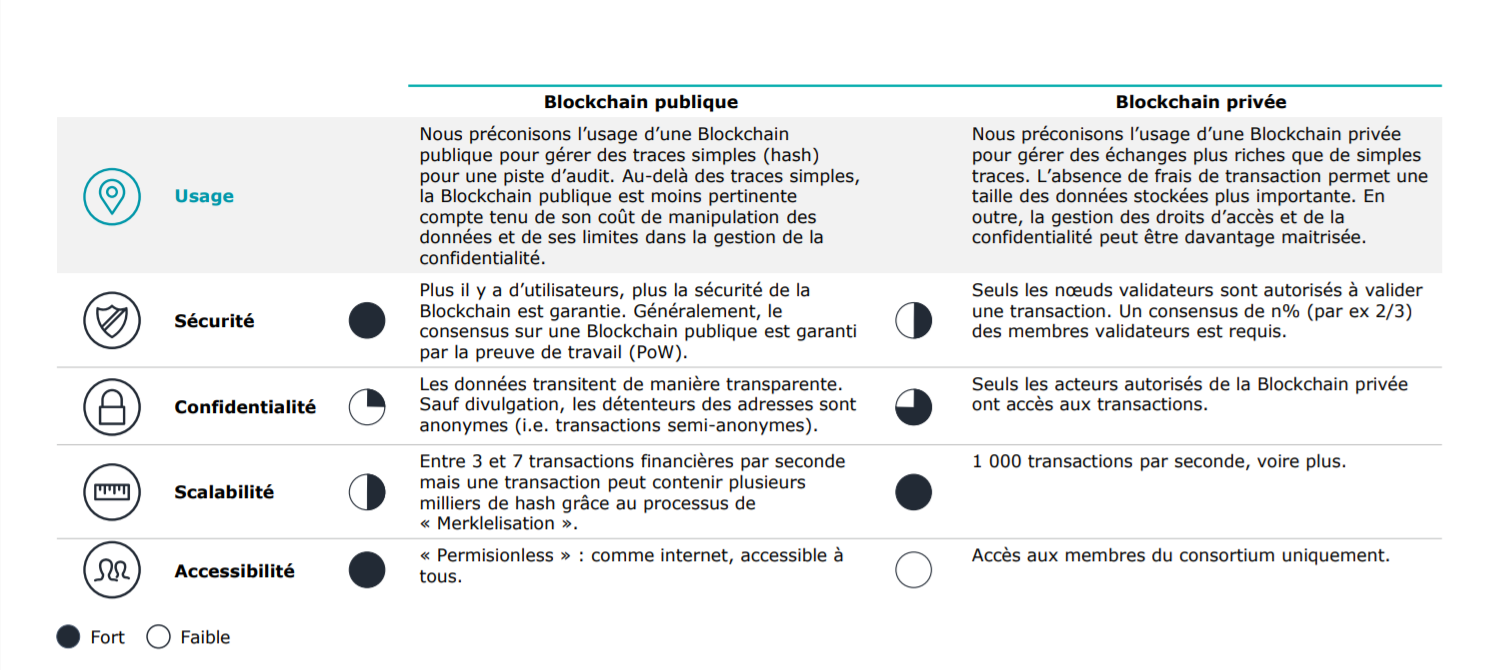
Exemples : R3 (banques), EWF (énergie), B3i (assurance), Corda.

Il y a certains inconvénients, étant donné que la création de bloc est facilitée (une signature suffit), il est possible de créer autant de chaines que possible. De fait si l’on venait à ouvrir la blockchain en lecture à des tiers (clients, auditeurs), ceux-ci n’auraient aucun moyen de vérifier que les données qu’ils consultent soient en provenance de chaine légitime. On pourrait donc se retrouver dans un système de Ponzi (montage financier frauduleux qui consiste à rémunérer les investissements des clients essentiellement par les fonds procurés par les nouveaux entrants).

#### BLOCKCHAIN TOTALEMENT PRIVEE

La blockchain dites « privée » va autoriser un nombre limité et prédéfini d’acteurs. C’est une seule organisation qui va autoriser ou non la possibilité d’effectuer des transactions ou même de participer à la validation de blocs.

Exemples : Hyperledger Fabric, Corda, OpenChain, etc.



**Figure 3  : Différences entre blockchain privée et publique (La Blockchain- Panorama des technologies existantes © 2017 Deloitte SAS)**

### III.2.3 Critères relatifs à notre projet

Nous nous basons sur des blockchains existantes.

Les critères de comparaisons seront les suivants :

1. Type blockchain
2. Accès en lecture
3. Accès en écriture
4. Accès en modification

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Type de blockchain | Lecture | Ecriture | Modification | Exemples d’utilisation |
| Publique | Tout public | Tout public | Tout public | Bitcoin |
| Consortium | Public restreint | Public restreint | Tout public ou Public restreint | Transactions entre plusieurs banques |
| Privé | Public restreint | Administrateur | Administrateur | Transaction au sein d’une seule banque |

## III.3 La crypto-monnaie

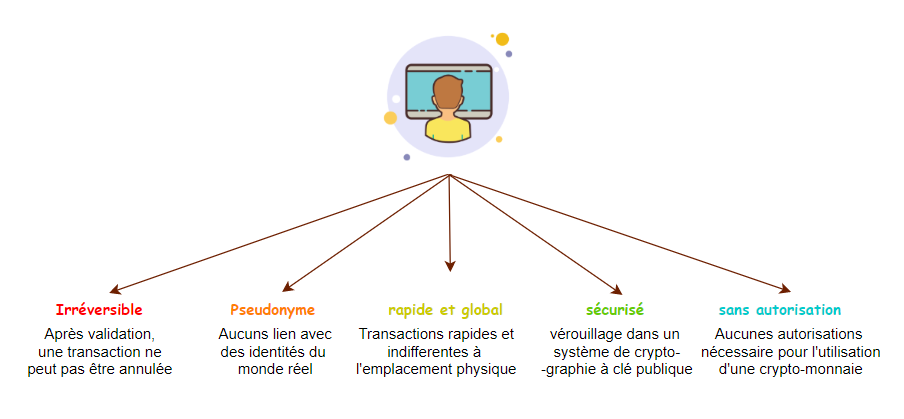
### III.3.1 Définition

Une cryptomonnaie est une monnaie virtuelle, elle ne dispose donc pas de support physique. Elle permet de réaliser des transactions financières, des achats, des virements, ou du stockage de valeur, comme la monnaie traditionnelle.



**Figure 4 : Qu'est-ce que la crypto-monnaie ?**

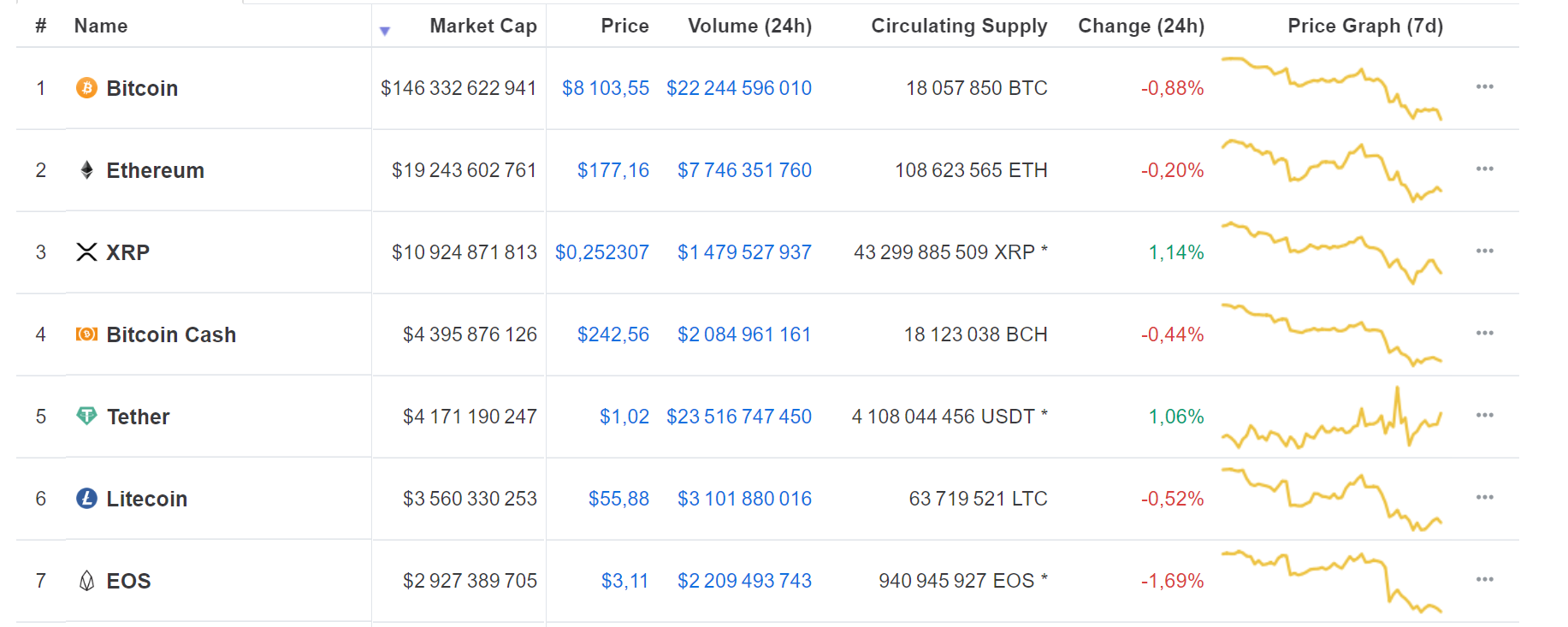
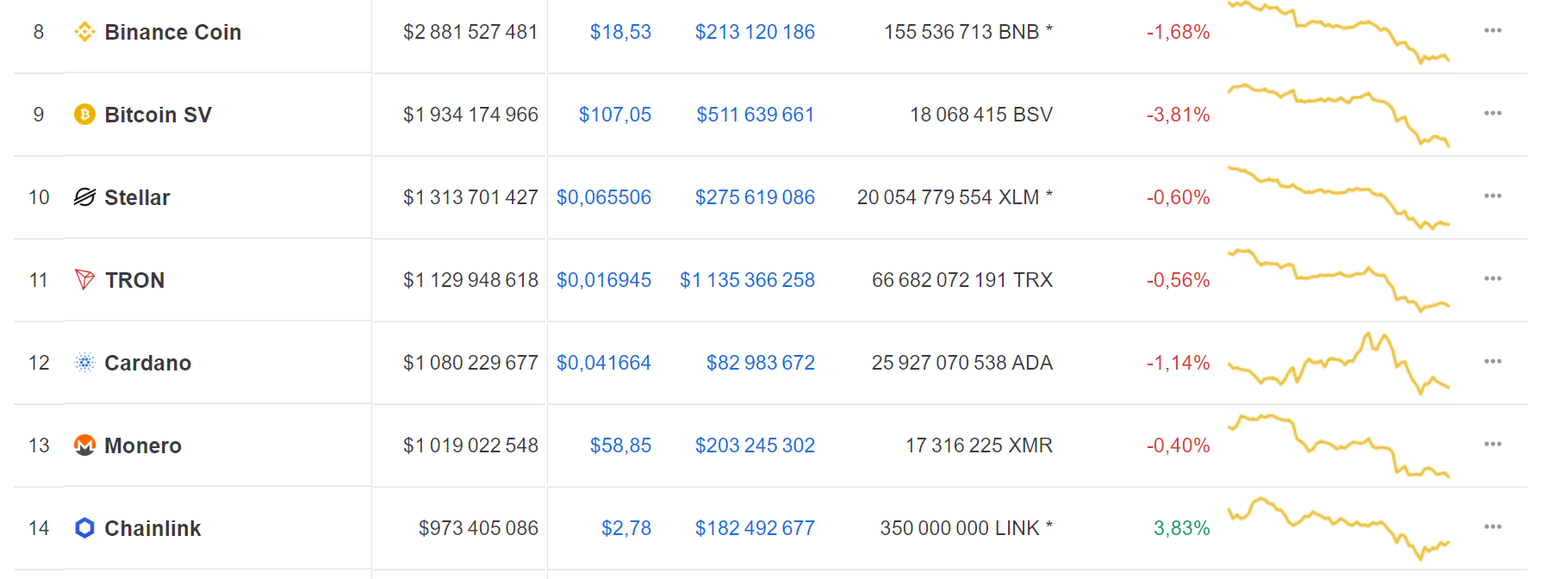
La cryptomonnaie est cryptée et peut être utilisée uniquement par les personnes détenant le code de décryptage. Il peut s’agir d’un mot de passe, d’une empreinte digitale ou de tout autre élément permettant de s’identifier. Contrairement à la monnaie classique, les transactions sont très rapides, très peu couteuses et se font dans l’anonymat total. Grâce au système de cryptographie, les transactions ne peuvent pas non plus être falsifiées.



**Figure 5: Des possibilités technologiques révolutionnaires**

### III.3.2 Les différentes crypto-monnaies

Ci-dessous un classement des crypto-monnaies en fonction de leur valorisation boursière.



### III.3.3 Critères relatifs à notre projet

Nous nous basons sur des crypto-monnaies d’états en circulation.

Les critères de comparaisons seront les suivants :

* Plateforme : indique sur quelle blockchain la crypto-monnaie s’appuie
* Objectif : indique dans quel cadre la crypto-monnaie a été créée et quels sont les objectifs du projet
* Type blockchain : privée, publique ou de consortium
* Caractère social : indique si le projet pour lequel a été créé la crypto-monnaie est à but non lucratif

# Etude de l’existant

## IV.1 L’environnement

### IV.1.1 Openchain

En se basant sur les versions précédentes du projet, la blockchain choisie est Openchain.

#### IV.1.1.1 Fonctionnement

Openchain est une technologie de blockchain open source, ou plutôt chaîne de transaction. En effet, Openchain n’utilise pas le concept de bloc vu plus haut, les transactions sont directement groupées entre elles et non via des blocs. Cela permet de gagner en temps et les transactions sont validées en (quasi) temps réel.

**Caractéristiques propres à Openchain :**

* Validation instantanée des transactions
* Pas de frais de minage
* Scalabilité extrêmement élevée
* Sécurisé via signatures numériques
* Immutabilité ***(1)***
* Attribution d’alias aux utilisateurs au lieu d'utiliser des adresses en base 58
* Plusieurs niveaux de contrôle :
  + Un registre entièrement ouvert pouvant être rejoint anonymement
  + Les participants doivent être approuvés par l'administrateur
  + Certains utilisateurs jouissent de plus de droits que les utilisateurs anonymes
* Système de hiérarchie
* Transparence des transactions
* Gestion de la perte ou du vol de clés privées pour les utilisateurs finaux
* Possibilité d'avoir plusieurs instances d'Openchain

***(1) Engager une ancre dans la blockchain Bitcoin à l'irréversibilité de sa PoW***

#### IV.1.2 API

Le serveur Openchain expose une [API](#_Index) HTTP pouvant être utilisée pour interagir avec les données. L'URL d'une opération est construite à partir de l'URL de base du nœud final et du chemin d'accès relatif de l'opération.

Par exemple, si l'URL de base est https://www.openchain.org/endpoint/, pour appeler l'opération/record (interroger un enregistrement), l'URL complète sera <https://www.openchain.org/endpoint/record> .

**Quelques opérations utiles :**

* Soumettre une transaction (/submit)
* Interroger un enregistrement (/record)
* Flux de transactions (/stream)
* Récupérer les informations sur la chaîne (/info)
* Interroger un compte (/query/account)
* Interroger une transaction (/query/transaction)
* Interroger une version spécifique d'un enregistrement (/query/recordversion)
* Interroger toutes les mutations ayant affecté un enregistrement (/query/recordmutations)
* Interroger les enregistrements dans un compte et ses sous-comptes (/query/subaccounts)
* Interroger tous les enregistrements avec un type et un nom donné (/query/recordsbyname)

Dans ce projet, le traitement des requêtes HTTP venant du client AngularJS passe par une API Restful constituée de :

1. Un serveur avec Node.js et le Framework Express.js

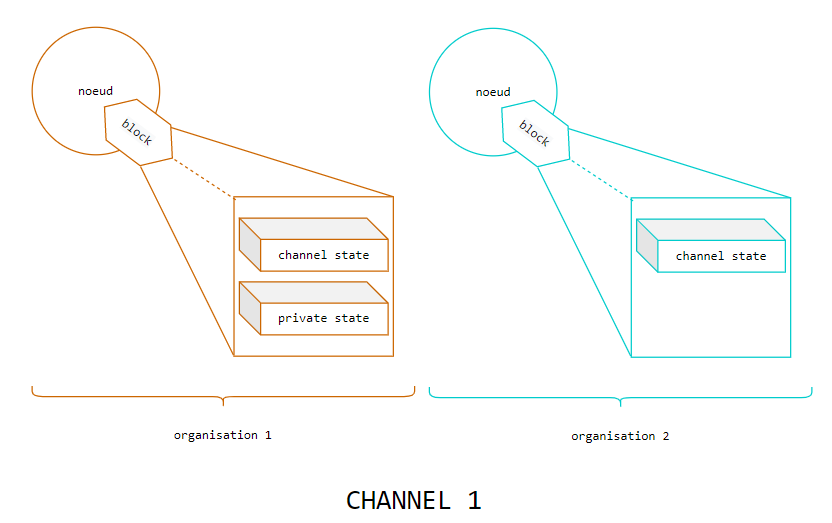
2. Une base de données MySQL

### IV.1.2 Hyperledger Fabric, une alternative à Openchain

#### IV.1.2.1 Concepts

Hyperledger Fabric est une blockchain Open Source, extrêmement modulaire, proposée par la Linux Foundation et spécialement conçue pour un usage privé et professionnel. Elle s’organise autour de plusieurs concepts :

* Organisation : acteur de la chaîne
* Noeud : un nœud du réseau, appartient à une organisation qui est chargée de maintenir son fonctionnement
* Brick : chargée d’organiser la vie du réseau
* Channel : « sous blockchain » dans laquelle sont inscris les blocs en fonction du sujet
* CA : brick chargée de vérifier les certificats
  + Ajouter des identités
  + Générer des certificats
  + Renouveler ou révoquer des certificats
* Blocs cachés (private data) : permet de créer des données privées dans un channel au lieu de créer un nouveau channel, on peut donc partager un même channel entre plusieurs organisations
* Cycle de vie du Chaincode



**Figure 6 : Schématisation d'un bloc caché**

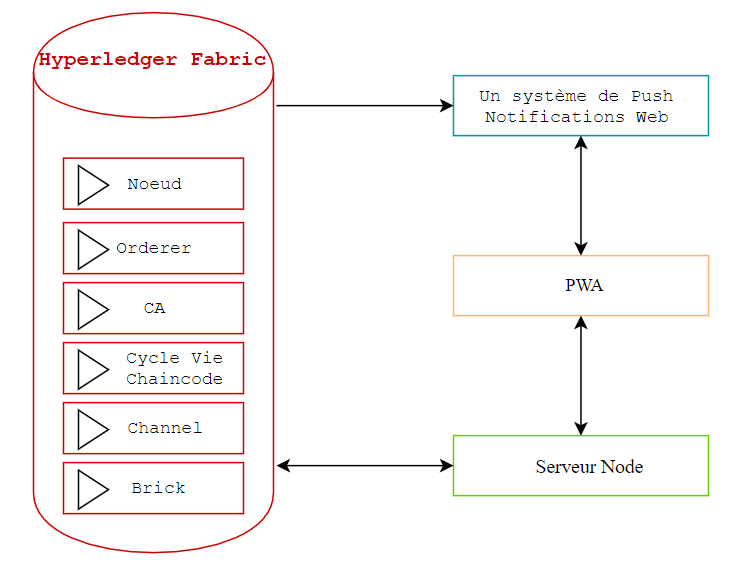
#### IV.1.2.2 Architecture



**Figure 7 : Architecture Hyperledger Fabric**

A la différence de OpenChain, Hyperledger Fabric contient un « bloc » contenant les états finaux de chaque objet afin d’alléger la charge d’un bloc (contient tous les fichiers, transactions, etc.)

#### IV.1.2.3 Architecture du PoC



**Figure 8 : Fonctionnement PoC**

Le PoC est composé de différents éléments :

* Une instance Hyperledger vierge
* Un server Node.js
* Un système de Push Notifications Web
* Une PWA

#### IV.1.2.4 Conclusion

Hyperledger Fabric semble convenir parfaitement à ce que nous voulons dans notre projet.

Dans notre cas la blockchain est déjà implémentée et le temps imparti ne nous permet pas de changer de technologie. Or, elle serait intéressante à mettre en place dans une future version de l’application.

### IV.1.3 MySQL – API REST NodeJs – Angular 7

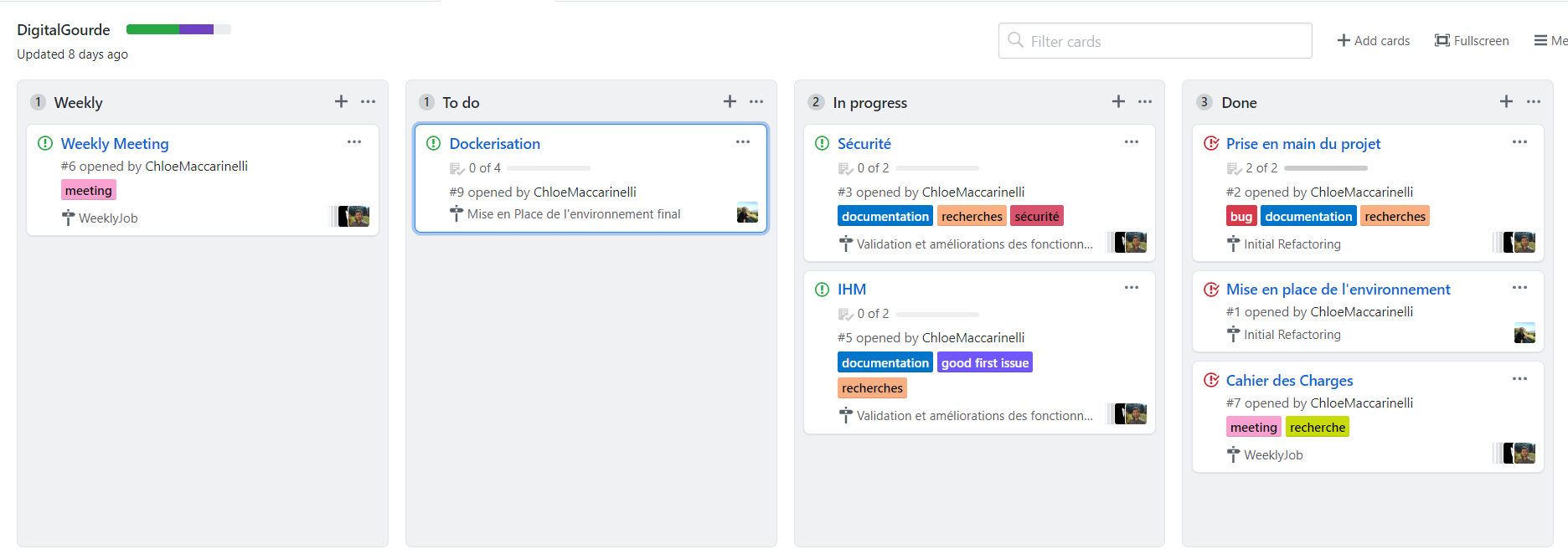
Les détails des environnements utilisés sont disponibles en [Annexe : Documentation Technique de l’environnement existant](Annexes.docx)

# Démarche projet

## VI.1 Gestion de projet

Dans le cadre de ce projet nous n’avons pas suivi une méthode précise de gestion de projet. Mais afin de s’assurer de l’avancement du projet de manière évolutive et très organisée il a été divisé en sprint.

La gestion du projet s’effectue directement sur GitHub avec les outils intégrés.



**Figure 9 : Kanban DigitalGourde**

## VI.2 Contraintes, outils et risques

### VI.2.1 Contraintes

L’application doit être scalable et facilement utilisable par tout utilisateur, elle doit donc répondre à certaines spécifications.

De plus, certaines règles de logiques doivent être introduites dans le processus de développement pour respecter le fonctionnement classique d’une banque.

LES REGLES SUIVANTES DOIVENT ETRE RESPECTEES

Nous supposons que le propriétaire de l’application est la banque centrale d’Haïti (BRH)

Toutes les opérations doivent faire l’objet de vérifications

Le solde d’un portefeuille ne peut être négatif :

* La banque centrale ne peut distribuer aux institutions financières plus que le montant restant émis en gourde électronique disponible dans son portefeuille
* Une institution financière ne peut distribuer à ses clients plus que ce qu’elle possède dans son portefeuille
* Un particulier ou commerçant ne peut virer ou retirer plus que ce qu’elle possède dans son portefeuille

La banque centrale peut à tout moment faire une nouvelle émission de monnaies électroniques

Quel que soit l’opération, on débite toujours un portefeuille pour créditer un autre portefeuille

Un portefeuille est toujours rattaché à une banque

### VI.2.2 Outils

Pour la réalisation de ce projet nous avons utilisé les outils suivants :

• draw.io

• IntellIj

• Github

• Visual Studio Code

• GanttProject

### VI.2.3 Risques

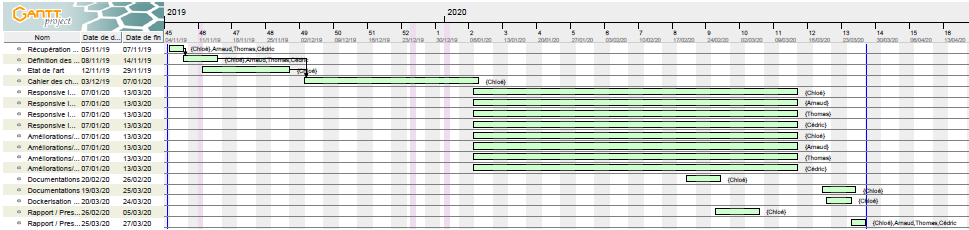
Par définition, un risque est un danger éventuel, plus ou moins prévisible, inhérent à une situation ou à une activité. L’idée du plan de risque est d’anticiper et de limiter au maximum les risques qui pourraient survenir pour tenter de réduire leurs impacts sur le bon déroulement du projet, vu qu’il est fort probable que le projet ne se réalise pas selon les prévisions et que le risque zéro n’existe pas.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Libellés | Priorité | Facteurs | Actions | Statut / date début du risque | Coût |
| Ne plus disposer de la maintenance de l’outil Open Source (Openchain) | 3 | Maintenance annulée par les contributeurs | -Formation sur un nouvel outil et implémentation  -Consultant pour le changement d’outils  (Prévoir un budget supplémentaire) | Plus tard | 20 JH  Ou \*\* |
| Ne pas pouvoir réaliser le prototype à temps | 5 | -Cahier des charges non définies  -Le temps impartie pour la réalisation du projet | -Implémenter les cas d’utilisation les plus important  -Délai supplémentaire | En cours /  20/02/2020 | 20JH |

\*\* Pour une blockchain privée ou semi-privée déjà existante type Hyperledger, Quorum ou **Corda**, les coûts seront liés aux développements nécessaires pour implémenter les modules et les fonctionnalités voulues ainsi que des **coûts de licence** et de maintenance. Pour l’implémentation d’une blockchain sur mesure “from scratch”, les coûts sont liés à du développement et au **maintien des serveurs** principalement.

## VI.3 Planning

Vous trouverez ci-dessous la planification sous forme de diagramme de Gantt, et le planning prévisionnel.



**Figure 10 : Diagramme de Gantt**

|  |  |
| --- | --- |
| Élaboration du cahier des charges  Etat de l’art  Prise en main du projet  Mise en place de l’environnement et nouvelle technologie (Flutter) | 15/11/2019 au  01/02/2020 |
| Recherche de faille de sécurité, amélioration et implémentation du niveau de sécurité de certaines fonctionnalités | 01/02/2020 au 20/02/2020 |
| Amélioration de la partie client web et implémentation du client mobile | 20/02/2020 au 20/03/2020 |
| Documentation, test et déploiement de l’application | 20/03/2020 au  30/03/2020 |

La nature prévisionnelle de ce calendrier pourrait cependant l’amener à être modifié au niveau des dates estimées, elles restent donc plus ou moins flexibles.

Avec une estimation de travail de 18 Jours Homme par mois sur 5 mois, on estime le coût du projet à 18JH \* 5 mois = 90JH.

## VI.4 Budget

A voir

# Exigences fonctionnelles

Les exigences fonctionnelles définissent les caractéristiques fonctionnelles que l'application fournit aux entités extérieures interagissant avec elle, appelées "acteurs".

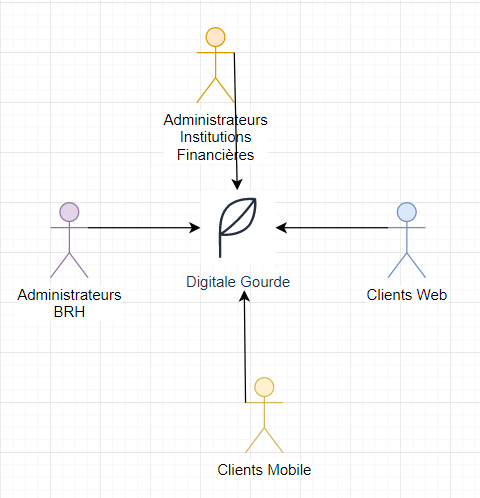
Une exigence fonctionnelle définit :

• Une information gérée par l'application

• Un comportement ou une fonction réalisée par l'application, appelé "cas d'utilisation".

Les cas d‘utilisation décrivent les interactions de l'application avec ses acteurs.

## VIII.1 Les acteurs de niveau fonctionnel



**Figure 11 : Les acteurs**

**Client Web et mobile :** personne qui utilise le système pour effectuer des transactions

**Administrateur BRH :** Ce sont les plus grands administrateurs, ce sont eux qui font les premières transactions et déterminent le nombre de cryptomonnaie en circulation avec l’accord de la BRH.

**Administrateur Institutions Financières :** personne responsable de la paramétrisation du système et de la distribution des droits d’accès de l’instance validateur dont il contrôle.

## VII.2 Les cas d’utilisations

ouverture de portefeuille

# Exigences non-fonctionnelles

## IX.1 Utilisabilité

• Fonctionnalités accessibles au client Web et au client mobile

• L’application doit être accessible de n’importe quel support

• L'application doit être accessible par plusieurs utilisateurs à la fois

• La création d’une transaction doit pouvoir se faire en 3 clics

• La création d’un portefeuille doit pouvoir se faire en 3 clics

## IX.2 Performances

• La blockchain doit être capables de valider plus de 1000 transactions par /s

• L'application web se doit d'être capable de présenter les résultats à au moins 1000 utilisateurs simultanément.

• La blockchain doit être à faible consommation énergétique.

• Permettre la validation presqu’instantanée des transactions et à faible coût

## IX.3 Robustesse

• Accès Internet : disponibilité 24H/24H

• Les informations métier être doivent disponibles pendant au moins 10 ans, sans aucune perte

## IX.4 Sécurité

• Une transaction ne peut être validée sans la signature du client possédant la clé privée

• Aucune modification ne peut se faire que par l’administrateur

• Les mots de passe doivent être cryptées

## IX.5 Maintenabilité, évolutivité

• Facilité de déploiement

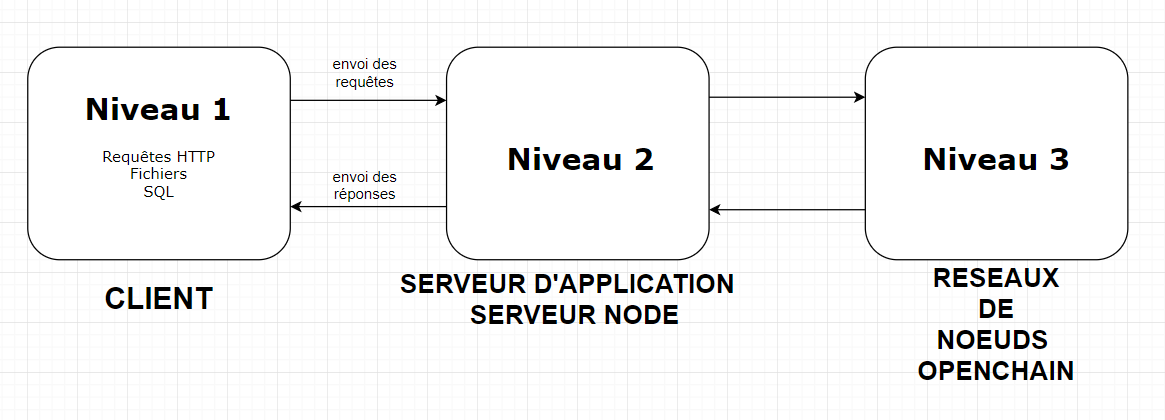
• Application appelée à évoluer sur 10 ans

# Interfaces détaillées

# Architectures

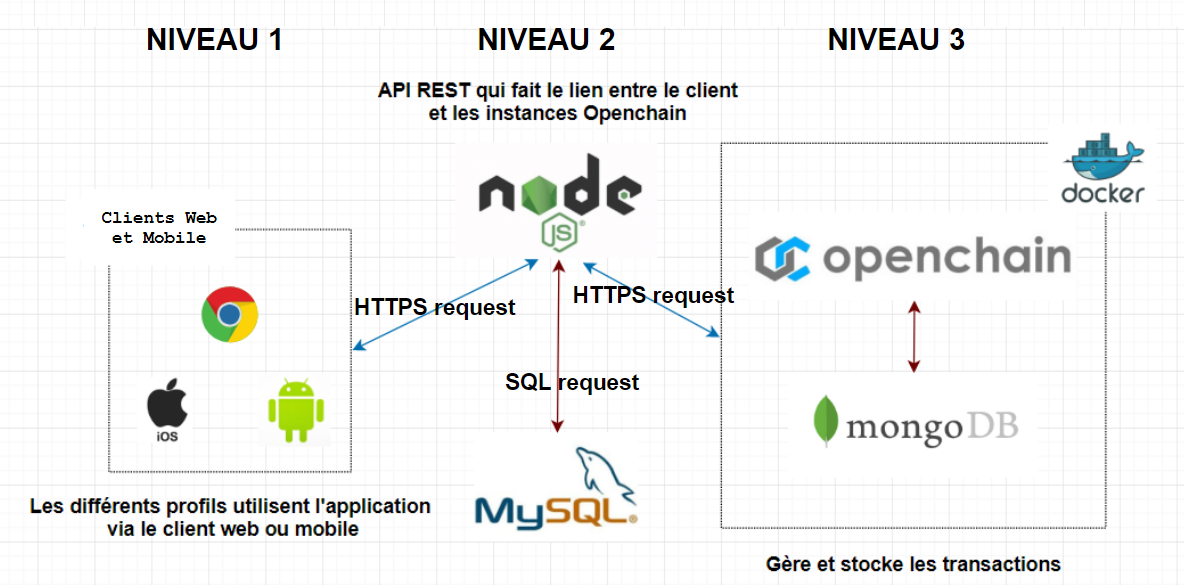
## Détails de l’architecture  3 tiers

Nous choisissons une architecture 3 tiers, constituée du client (web et mobile), du serveur d’application (digital Gourde) et du réseau blockchain.



**Figure 12 : Architecture 3 tiers**

## Détails de l’architecture technique



**Figure 13 : Architecture technique**

**Niveau 1 :** Clients web et mobile par lesquels nos divers profils pourront accéder aux services de DHTG. La connexion à ceux-ci peut être soit via le web, soit via une application mobile, c’est ici que les demandes d’opérations seront effectuées.

**Niveau 2 :** Les clients web et mobile communiquent avec une API REST (node js) qui fait le lien entre la blockchain OpenChain (transactions, portefeuilles, cryptomonnaie), les données des banques et des utilisateurs (stockées dans une base de données MySQL) et les clients web et mobile

**Niveau 3 :** Finalement, l’instance OpenChain qui gère les transactions de la blockchain est hébergée dans un conteneur Docker avec un base de données MongoDB où seront stocker les données liées aux transactions

# Les Améliorations

## V.1 Sécurité

Dans l’application existante, un seul portail permet d’accéder à la fois à l’espace administrateur et particulier/commerçant.

En séparant les deux interfaces on assure une première couche de protection.

url*/portail : espace particuliers*

url*/brh/accueil : espace BRH*

A COMPLETER

## V.2 Client Web et mobile

A COMPLETER

# Dockerisation

Après avoir récupérer le projet nous avons rencontré de nombreuses difficultés à le faire fonctionner. La solution la plus simple est donc de le dockeriser.

Avantages :

* L’application est isolée
* Possibilité de création d’un réseau de containers
* Léger
* Lancement de l’application simplifié

# Perspectives

## V.2 Client Web et mobile

Afin d’optimiser au mieux le temps de développement de la solution, ainsi que le coût de développement, nous avons décidé d’utiliser le framework « Flutter » qui nous permet de développer une seule application qui sera utilisable sur tous les supports (Client Web, mobile, tablette, …).

### V.2.1 Framework Flutter

Flutter, est un Framework open source développé par Google qui a pour objectif de rassembler les points fort de nombreux outils existants. Flutter se fait connaître pour sa capacité à concevoir des applications natives multiplateforme (Android, IOS, Windows, Mac, Linux).

### V.2.2 Langage Dart

Ce langage « oublié » a comme principale avantage d’offrir deux modes de fonctionnement

* AOT (Ahead Of Time)
* JIT (Just In Time)

#### V.2.2.1 AOT

Dart permet de générer une application native pour chaque plateforme. Le code est donc optimisé pour l’architecture sur laquelle il fonctionne.

#### V.2.2.2 JIT

Dart propose la fonctionnalité de Hot Reaload qui permet de réduire considérablement le temps de développement. En effet, le principe de Hot Reload est de réduire le temps entre chaque build, qui est relativement long quand l’on développe une application Android native. On passe donc à quelques millisecondes (pire des cas) entre chaque build.

De plus Dart est performant pour gérer les allocations mémoire et le « Garbage collector ».

# Conclusion

# Webographie et Bibliographie

* <https://blockchainfrance.net> , Blockchain France
* [https://docs.openchain.org/](https://docs.openchain.org/en/latest/index.html) , documentation OpenChain
* [https://hyperledger-fabric.readthedocs.io/](https://hyperledger-fabric.readthedocs.io/en/release-1.4/) , documentation Hyperledger Fabric
* <https://flutter.dev/> , site web officiel de Flutter
* , Rapports et travaux réalisés par , année 2018-2019, disponibles sur : <https://freedcamp.com/MBDS_6di/Projet_DIGITAL_G_LzG/todos>