Le petit guide de la Blockchain

Intro

Le petit guide de la Blockchain vous est présenté par les départements Blockchain-Fintech et Legal Design de l'association Assas Legal Innovation qui a pour objectif de sensibiliser les étudiants et professionnels à l'encadrement juridique des nouvelles technologies et à leur impact sur le marché du droit. Il a vocation à aider les juristes à appréhender le fonctionnement d'une Blockchain, une technologie très prometteuse posant de nombreuses questions juridiques. Pour faciliter sa compréhension nous avons tenté d'utiliser au maximum des schémas et des exemples. A noter, ce guide décrit seulement le fonctionnement de la Blockchain publique Bitcoin. En effet, il existe d'autres Blockchain organisées différemment. Pour réaliser ce guide nous nous sommes appuyés sur le site d'Ethereum France que nous remercions pour la qualité de leurs articles.

Avant d'entrer dans le vif du sujet, essayons de donner une définition globale d'une Blockchain publique.

Il s'agit d'un registre transparent, décentralisé et sécurisé permettant d'échanger et de stocker des informations :

- ***Un registre*** *car elle pourrait être assimilée à une base de données ou un grand livre comptable dans lequel des opérations, transactions ou données sont regroupés en block d'où le terme Blockchain.*
- ***Transparent*** *car public et chacun d'entre nous a accès à l'historique de toutes les opérations effectuées par ses utilisateurs depuis sa création.*
- ***Décentralisé*** *car partagé par des ordinateurs dans le monde entier, les nœuds du réseau, sans organe central de contrôle. Une Blockchain est un réseau pair-à-pair.*
- ***Sécurisé*** *car la validation des blocks et des opérations, transactions ou données contenus dans une Blockchain est réalisée par l'intermédiaires de puissants procédés cryptographiques ce qui le rend très difficilement piratable. La cryptographie désigne l'ensemble des techniques permettant de chiffrer une donnée. Cette sécurité est renforcée par le caractère décentralisé de la Blockchain car, chaque nœud du réseau détenant une copie du registre et le protocole reposant sur un algorithme de consensus entre ces derniers, pour corrompre le réseau il faudrait altérer 51% des ordinateurs participants simultanément. Une Blockchain est donc dite infalsifiable. Toutefois, le développement des ordinateurs quantiques pourrait remettre en cause cette caractéristique.*

La particularité des Blockchain publiques est qu'elles fonctionnent avec une crypto-monnaie. Une rémunération des nœuds du réseau est effectivement nécessaire pour les inciter à utiliser leur puissance de calcul permettant la validation des blocks.

La Blockchain s'inscrit donc dans un mouvement de désintermédiation en permettant d'échanger et de stocker de la valeur sans organe central de contrôle. Internet permet déjà d'échanger des informations en pair-à-pair mais non de la valeur tels que des actifs financiers ou des droits de propriétés intellectuelles. En effet, si Harvey Specter souhaitait céder numériquement une valeur sans intermédiaire à Mike Ross sans la sécurité de la Blockchain Harvey Specter pourrait continuer d'en disposer, ce qui est problématique. La Blockchain permet donc de reconstruire au niveau numérique la propriété physique. Le caractère infalsifiable et transparent de la Blockchain en fait également un atout précieux pour des enjeux de traçabilité et de certification.

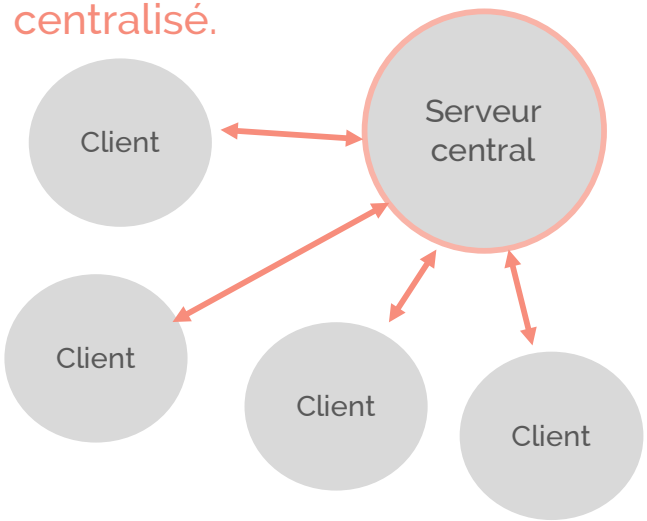
Ainsi, une Blockchain et ce qui en découle (Smart Contract, token, ICO…) peuvent être utilisés dans de nombreux secteurs tels que la banque, l'assurance, l'immobilier, l'industrie musicale, la santé… .

I. L'utilisation d'un réseau pair à pair (P2P)

Le pair à pair est un réseau de partage de données entre plusieurs ordinateurs.

Il doit être distingué du réseau client-serveur.

Dans un réseau client-serveur, **le serveur est la seule source des données**. C'est une entité passive qui attend les requêtes des clients et leur envoie des données. Ce type de réseau est dit propriétaire et centralisé.

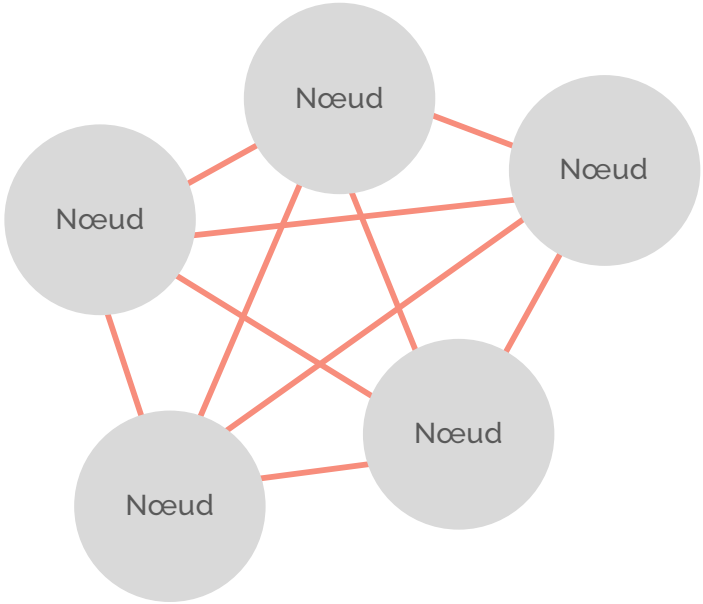


Le réseau client-serveur

Le problème est que si un nombre important de clients envoie une requête au serveur, celui-ci aura du mal à répondre à toutes les demandes.

C'est le problème de la congestion.*

Pour remédier à ce problème est apparu le modèle du pair-à-pair (pair-à-pair, égal à égal). C'est un système décentralisé, distribué.



Le réseau peer-to-peer

Le P2P est un système de répartition de charge. **Les entités sont à la fois clients et serveurs** :

- **Clients** car elles peuvent solliciter des données
- **Serveurs** car elles peuvent être sollicitées pour obtenir des données.

Il n'y a en fait aucun serveur : il suffit de deux ordinateurs, qui constitueront les nœuds du réseau.*

Tous les nœuds ont le même rôle ; il n'y a pas de statut privilégié pour un nœud.

*Congestion

Augmentation du trafic provoquant un ralentissement global du réseau informatique.

*Nœuds

Postes connectés par un protocole réseau pair-à-pair.

Chaque utilisateur décide des partages sur son disque dur et des permissions qu'il octroie aux autres utilisateurs.

Une ressource partagée sur un ordinateur constituant un nœud apparait sur tous les autres ordinateurs qui sont connectés au réseau : c'est le concept de partage arbitraire.

Le système pair-à-pair permet ainsi **l'échange d'objets** (fichiers, flux multimédias continus, le calcul réparti, services…) **entre tous les ordinateurs du réseau sans passer par un serveur central**.

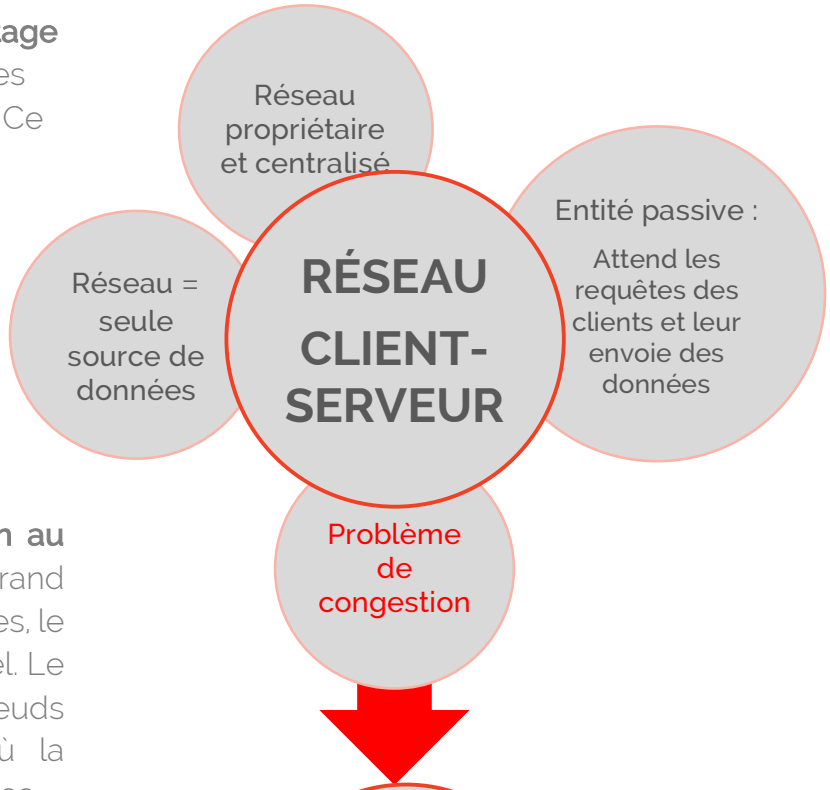
L'avantage : alors que dans un réseau client-serveur une panne du serveur empêche le réseau de fonctionner, **dans un réseau P2P, si un nœud tombe en panne, il reste toujours tous les autres ordinateurs pour servir de ressource.**

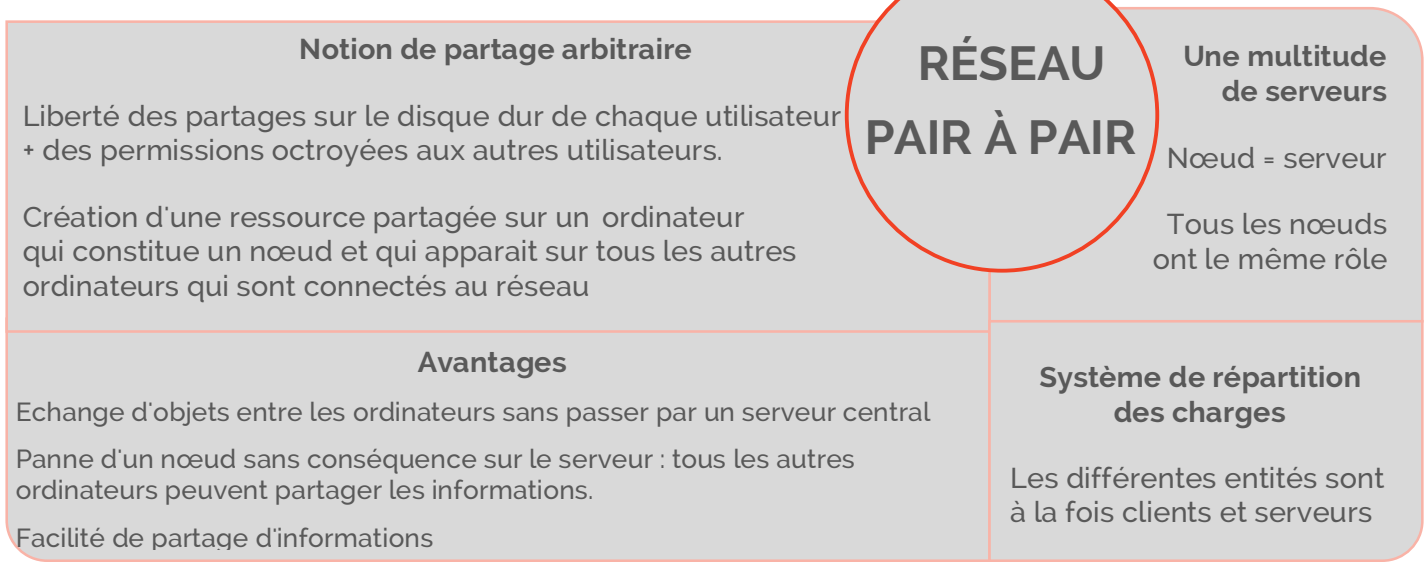
L'utilisation d'un système pair-à-pair nécessite pour chaque nœud l'utilisation d'un **logiciel** qui remplit à la fois les fonctions de client et de serveur. Ce logiciel est appelé servent, ou encore « client ».

Les systèmes pair-à-pair **facilitent le partage d'informations** et rendent la censure ou les attaques légales ou pirates plus difficiles. Ce sont ainsi des outils de choix pour **décentraliser des services qui doivent assurer une haute disponibilité tout en permettant de faibles coûts d'entretien.**

Néanmoins, ces systèmes sont plus complexes à concevoir que les systèmes client-serveur.

Le système P2P favorise **la coopération au sein d'une communauté.** Lorsqu'un grand nombre de nœuds propose des ressources, le système peut atteindre son plein potentiel. Le risque est alors qu'un ou plusieurs nœuds partagent des fichiers corrompus d'où la nécessité de s'équiper d'un firewall efficace.





II. Identifier les émetteurs des transactions

A. La fonction de hachage ou hash

Le HASH est l'empreinte du « block » qui le rend unique.

Chaque « block » est composé, en plus de son propre HASH, du HASH du « block » précédent. C'est ainsi que la **chaîne** est créée.

Composition du block :

BLOCK 54

- **HASH du block 53**
-Données
-Signature
-Preuve de travail
-**HASH du block 54**

BLOCK 55

- **HASH du block 54**
-Données
-Signature
-Preuve de travail
-**HASH du block 55**

Qu'est-ce que le HASH ?

Le HASH est une fonction mathématique qui transforme une donnée quelconque en une valeur numérique à longueur fixe.

La fonction SHA-256 est utilisée pour le Bitcoin. Elle permet de transformer toute donnée d'entrée en une valeur numérique de 256 bits… soit **64 caractères**. Elle utilise pour cela la notation hexadécimale (système de comptage constitué de 16 symboles qui a pour but de faciliter la manipulation des données). Les symboles utilisés sont les chiffres de 0 à 9 et les lettres de A à F. La principale caractéristique d'une telle fonction est qu'un seul changement, même insignifiant, de la chaine d'entrée provoque un changement important dans la chaine de sortie.

Chaîne d'entrée

Ce guide vous a été présenté par Assas Legal Innovation

Application de la fonction SHA-256

Chaîne de sortie

C68169885DC6855
F682B9D760DA556
701DA217DA24A0E
C479019D6DDED
D74FB8

Dans l'exemple, la chaîne d'entrée (input) est une phrase.. **Mais l'objet auquel est appliqué la fonction n'a pour condition que d'être une valeur numérique, dont la taille est d'ailleurs indifférente**. Dès lors, il est possible d'imaginer n'importe quel type de donnée en tant que chaine d'entrée : chiffres, signes, fichier, …

Il est important de se rappeler que l'output a toujours la même taille, quel que soit la taille de l'input.

Quelles sont les propriétés du HASH ?

COMPRESSION

La valeur d'entrée est généralement plus grande que la valeur de sortie.

RÉSISTANCE AUX COLLISIONS

Des valeurs d'entrée différentes ne peuvent aboutir à une valeur de sortie identique.

L'inverse est vrai aussi.

FONCTION À SENS UNIQUE

Il est impossible, en pratique, de trouver une valeur d'entrée à partir d'une valeur de sortie.

Le mode de consensus de la preuve de travail repose sur ce concept.

B. La signature digitale : vérifier que l'émetteur est bien le détenteur de la clé privée

La cryptographie asymétrique est un domaine de la cryptographie où il existe une distinction entre des données publiques et privées. Bitcoin utilise cette cryptographie afin d'identifier les émetteurs et les destinataires des transactions, en se basant sur une **combinaison de clé privée et clé publique**.

La clé privée **est propre à chaque émetteur**. Seul lui peut la connaitre.

La clé publique **est, elle, visible par tous**. Elle est déterminée par application d'une fonction à la clé privée.

Il s'agit d'une **fonction à sens unique**, (l'algorithme ECDSA) i.e. que la même clé publique sera toujours trouvée à partir d'une clé privée mais qu'il est impossible en connaissant la clé publique de retrouver la clé privée.

La clé publique permet de calculer l'adresse. L'adresse est la clé publique présentée d'une façon plus courte et plus lisible.

Application de la fonction :

Algorithme ECDSA (Clé privée) = Clé publique.

Il reste à prouver que l'émetteur de la transaction est le détenteur de la clé privée.

Dans un système centralisé, cela est vérifié par l'intermédiaire (l'établissement de crédit) qui vérifie que l'émetteur est bien le détenteur des fonds.

Dans un système décentralisé, cette preuve est fournie par la **signature digitale**.

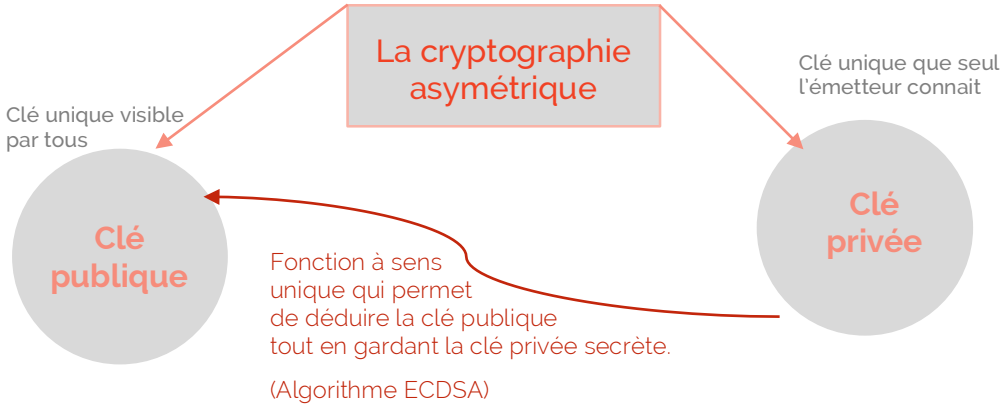
Au lieu de fournir la clé privée au réseau, l'émetteur fournit la signature digitale (déterminée par application d'une fonction à la clé privée et à la transaction). **La signature digitale est donc la preuve de la détention de la clé privée qui a généré la clé publique, et donc l'adresse.**

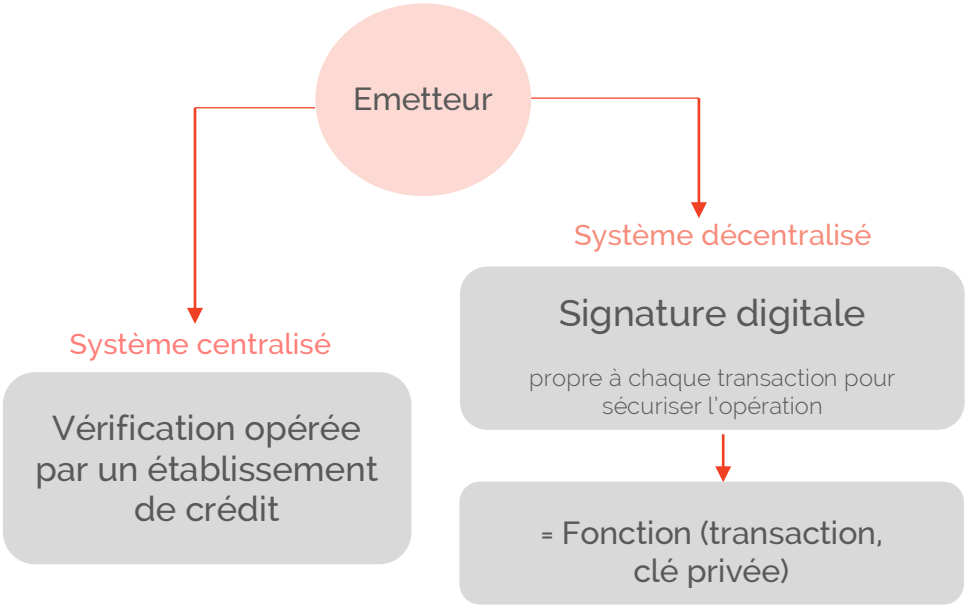
Application de la fonction :

Fonction (Clé privé + message émis) = Signature digitale.

Il s'agit toujours d'une fonction à sens unique.

La signature digitale est différente à chaque transaction émise. Il est donc impossible d'utiliser une même signature digitale pour valider deux transactions distinctes.





Pour vérifier que chaque signature est valide, **chaque nœud applique une fonction prenant en paramètre :**

+ l'adresse publique de l'émetteur

+ le message émis

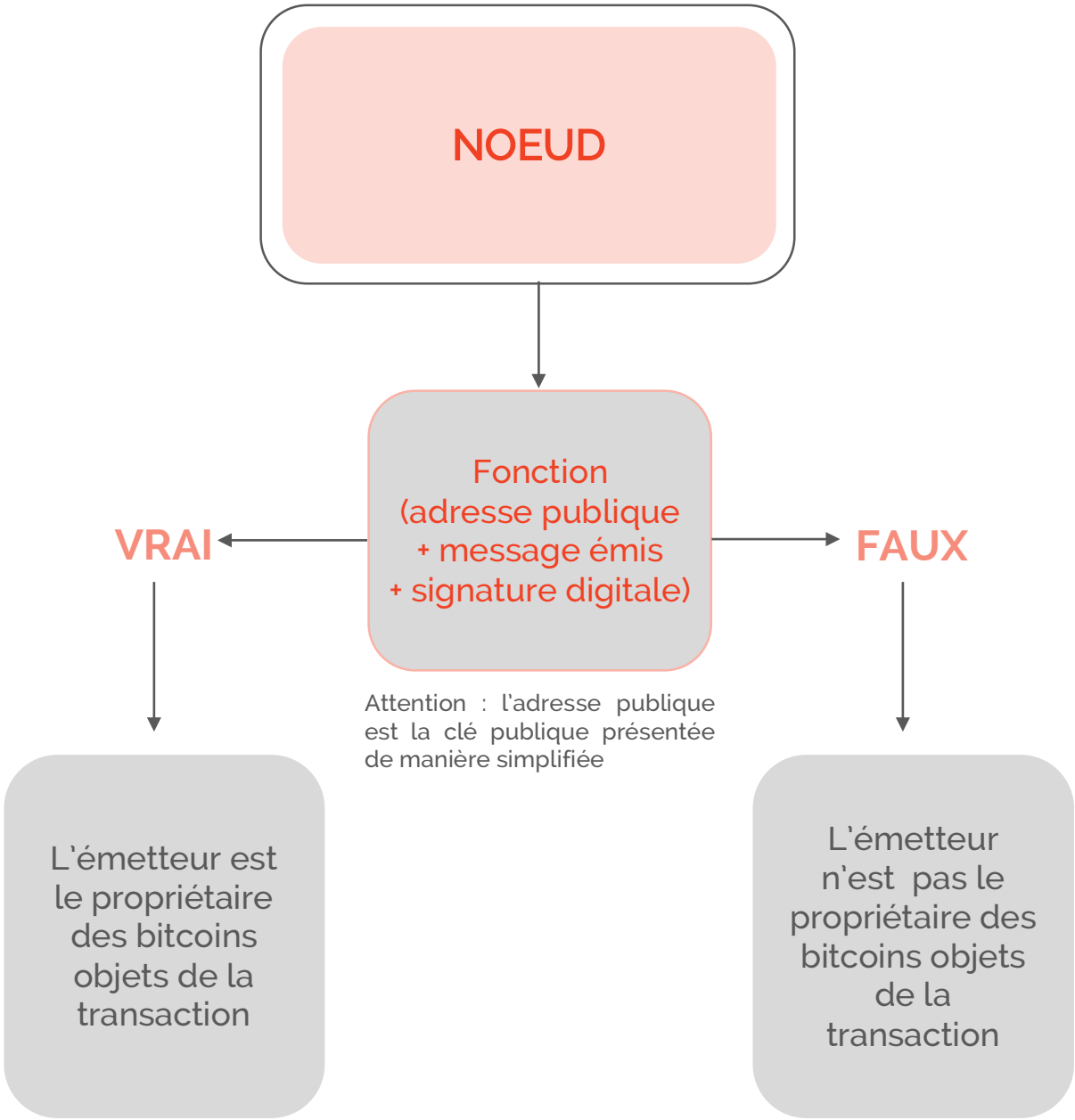
+ la signature digitale de la transaction considérée.

Cette fonction renvoie **vrai ou faux.**

Application de la fonction par le nœud :

Fonction (adresse émetteur + message émis + signature digitale) = Vrai ou Faux

De cette manière, **le nœud peut vérifier que l'émetteur est bien le détenteur de la clé privée sans prendre connaissance de la clé privée.**



C. Les UXTOs : vérifier que l'émetteur a suffisamment de fonds

Une fois que l'émetteur est considéré par le nœud comme légitime, pour que la transaction soit valide, **il faut que le nœud vérifie également que l'émetteur ait suffisamment de fonds.**

Dans un système centralisé, l'établissement de crédit vérifie en effet que chaque émetteur possède suffisamment de fonds sur son compte.

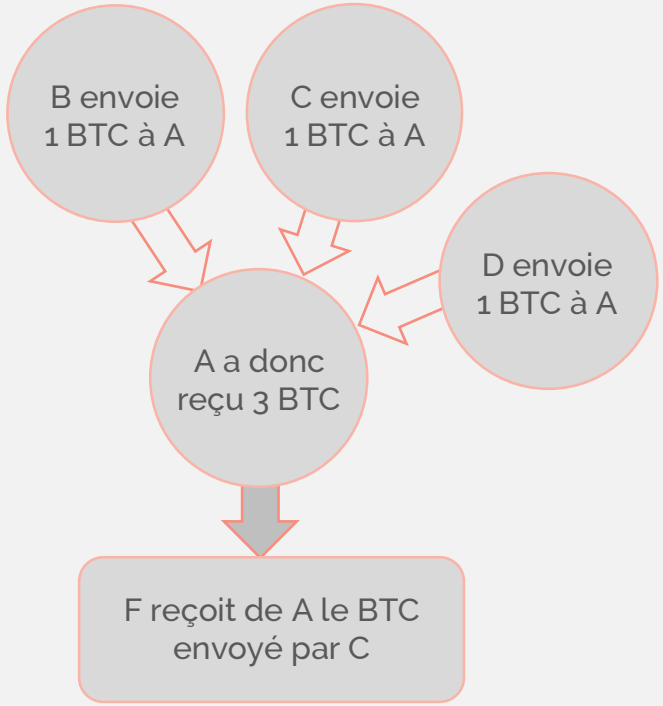
Dans un système décentralisé, les nœuds ne connaissent pas la balance associée à chaque adresse.

Pour calculer la balance de chaque compte, un nœud doit alors faire la **somme des sorties de transaction non dépensées – UTXO – associée à l'adresse du compte considéré.**

Autrement dit, afin de vérifier la validité de la transaction, **le nœud doit vérifier que toutes les transactions qui avaient été adressées à l'émetteur (autrement dit tous les fonds qu'il a pu recevoir) n'ont pas déjà été dépensées.**

Cela est possible parce que chaque transaction doit référencer tous les UXTOs.*

Il faut donc que tous ces **UXTOs aient une valeur supérieure ou égale au montant de la transaction dont la validité est vérifiée.**



La chaîne trace chaque Bitcoin individuel depuis sa création, et toutes les adresses auxquelles il a été envoyé. Ici, le système vérifie donc que le Bitcoin envoyé à F par A avait bien été envoyé par C auparavant, et non encore envoyé à quelqu'un d'autre.

Exemple : Matteo souhaite envoyer 15 BTC à Axel. La transaction considérée référence toutes les UXTOs, i.e. toutes les transactions que Matteo a reçues et qu'il n'a pas dépensées. La valeur totale de ces UXTOS doit être supérieure ou égale à 15 BTC

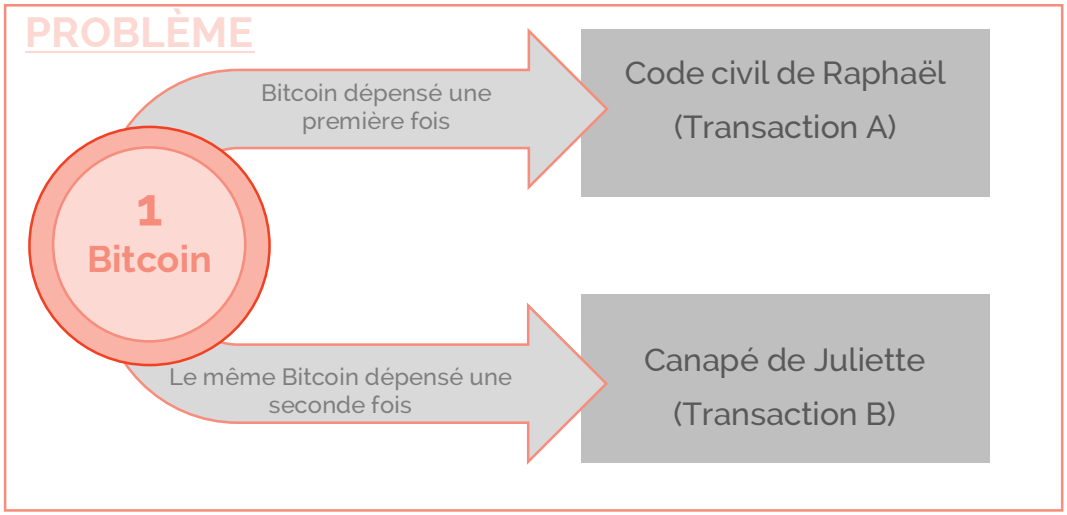
Ce mécanisme conduit à la formation d'une chaine de transactions : toute transaction indique une sortie de transaction non dépensée.

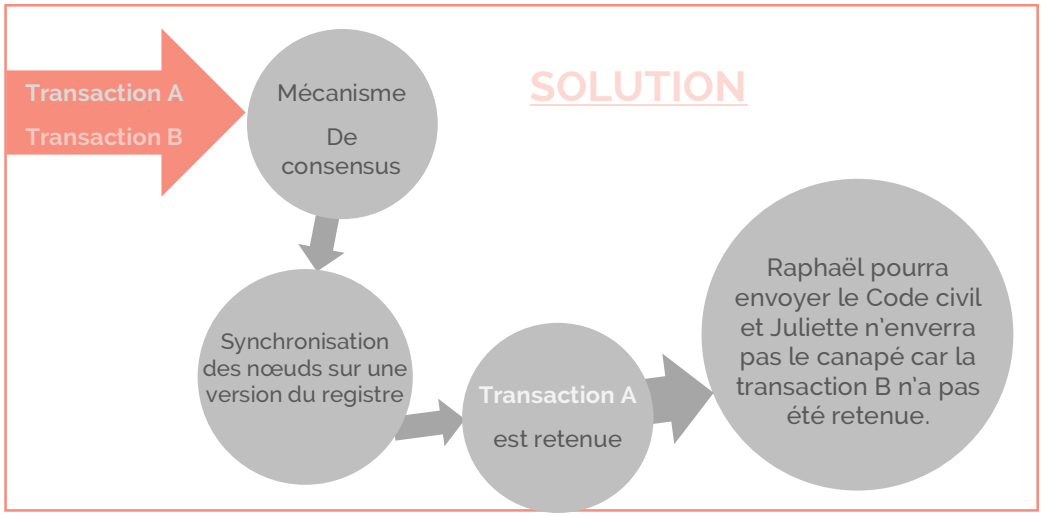
Pour chaque Bitcoin individuel se crée une chaîne de transactions : chaque nouvelle transaction contient et répertorie toutes les transactions précédentes.

*UXTO
« Unspent Transaction Outputs »
Transaction non dépensée.

III. La preuve par consensus pour remplacer la confiance

A. La résolution du problème de la double dépense





Dans un système décentralisé l'enjeu est de trouver un **consensus** entre les différents participants, les nœuds du réseau. Il faut donc que le réseau s'accorde sur la dernière version actualisée du registre distribué afin de déterminer qui possède quoi. En effet, il se pourrait qu'un nœud mal intentionné procède à une **double dépense**.

Le système bancaire traditionnel centralisé palie à cette difficulté par l'intermédiation. La Blockchain, elle, **ordonne et horodate** les transactions dans **une chaîne de blocks avec laquelle chaque nœud se synchronise.**

Chaque block est constitué d'un ensemble de transactions et **toutes les transactions d'un même block sont considérées avoir eu lieu au même moment.**

Toutes les 10 minutes un nouveau block est ajouté à la chaîne de blocks. **Tous les nœuds possèdent une copie identique du registre blockchain et chaque block référence le block précédent**, ce qui forme une chaine de blocks, qui s'étend jusqu'au premier block créé.

Il est donc impossible d'hacker un block sans hacker tous les blocks précèdent, ce qui est source d'une **grande sécurité du réseau**.

Exemple : Harvey Specter, créé une transaction d'un montant de 1 BTC à destination de Mike Ross, qui en échange lui envoie un Code civil. Une fois le Code civil reçu, Harvey Specter créé une seconde transaction en envoyant ce même BTC à destination de lui-même.

Il suffit que Mike Ross attende que la première transaction soit incluse dans un block avant d'envoyer le code civil. La blockchain étant identique pour chaque noeud, il n'y aura pas de désaccord. Si le réseau a retenu la transaction 1 (ce qui devrait être le cas) il sera bien propriétaire du BTC et enverra le Code Civil à Harvey Specter. Au contraire, s'il retient la transaction 2 il ne sera pas propriétaire du BTC et ne lui enverra donc pas le Code civil.

B. La vérification de la chaîne de blocks : la preuve de travail (PoW)

Comment un block est-il validé ?

Les nœuds du réseau qui reçoivent de nouvelles transactions les placent dans leur **liste de transactions non confirmées** (chaque nœud possède sa propre liste).

Puis ils vérifient la **validité des transactions** qu'ils reçoivent. Ils forment un set de transactions valides (qui constitue le contenu d'un block).

Les nœuds du réseau vérifient ensuite la **validité du block**.

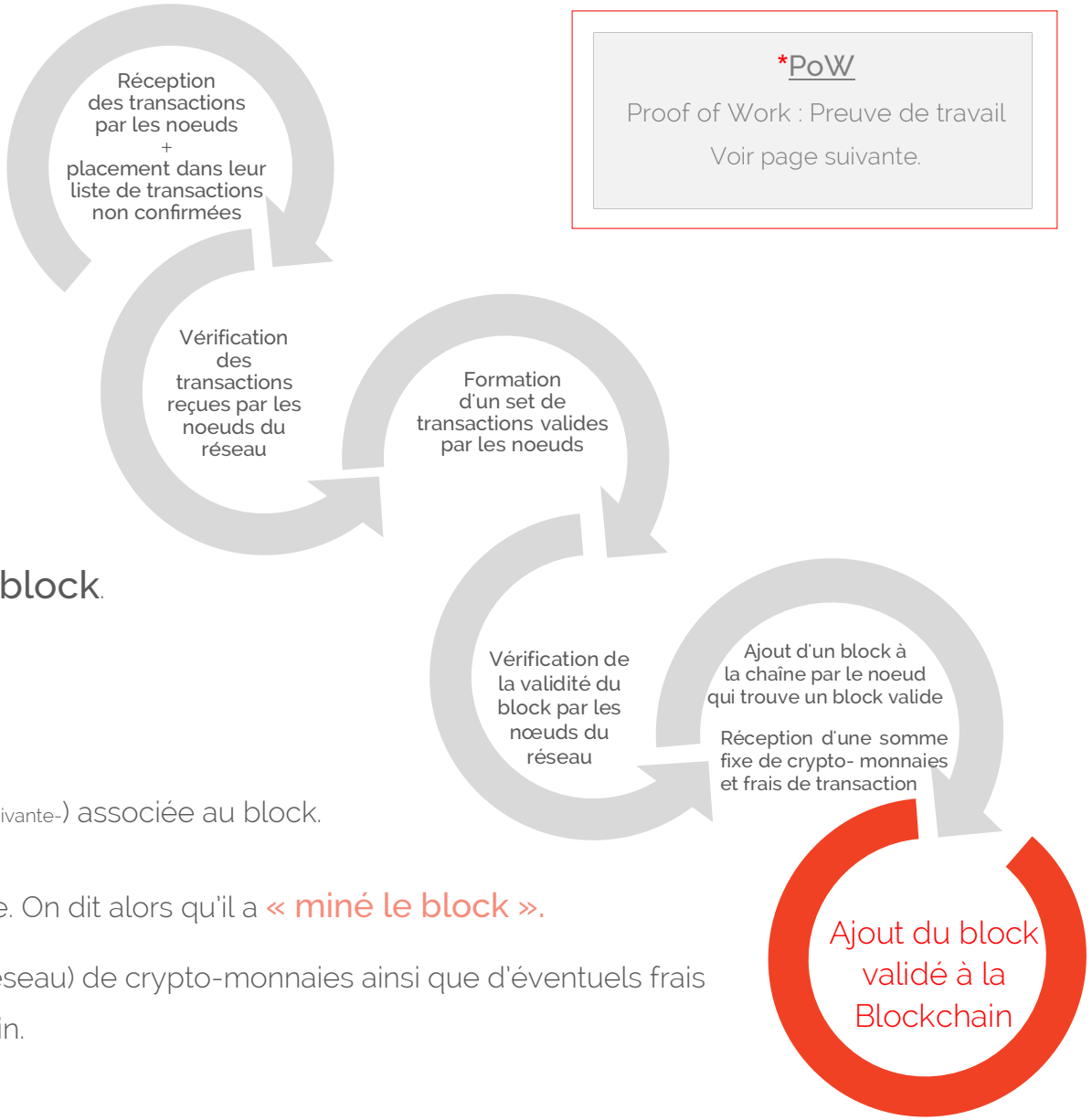
Les **conditions de validité du block** sont :

-la validité du set de transactions

-la validité de la preuve de travail (PoW* -voir page suivante-) associée au block.

Le nœud qui trouve un block valide ajoute son block à la chaîne. On dit alors qu'il a « miné le block ».

Il reçoit en compensation une somme fixe (déterminée par le réseau) de crypto-monnaies ainsi que d'éventuels frais de transaction. Le block une fois validé est ajouté à la Blockchain.



*PoW

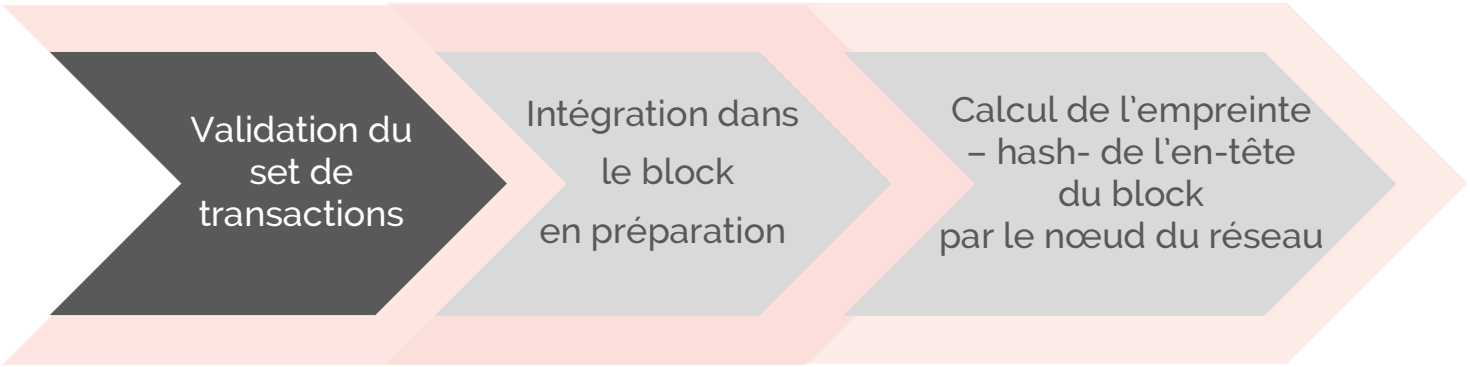
Proof of Work : Preuve de travail

Voir page suivante.

Comment fonctionne la PoW (Proof of Work) ?

La **preuve de travail** est la preuve que le mineur a résolu le problème mathématique de non-dépassement de seuil correspondant au block. C'est l'un des mécanismes de sécurité fondamentaux du réseau.

Une fois que le set de transactions validé est inclus dans le block en préparation, le nœud du réseau calcule **l'empreinte (le hash)** de l'en-tête du block. Cet en-tête contient la version du block, l'empreinte du block précédent, la racine de l'arbre de transactions, la difficulté et un nonce*



Le **hash de l'en-tête du block** doit, pour que la preuve de travail soit valide, être **inférieur** à un certain nombre que l'on nomme seuil.

Pour obtenir un **hash valide** il faut tester différents nonces jusqu'à obtenir un **hash inférieur à ce seuil.** Pour cela les nœuds du réseau utilisent la puissance de calcul de leur ordinateur.

Plus un mineur possède une puissance de calcul élevée, plus la probabilité qu'il trouve le nonce qui produira le hash du block est élevée.

NB 1 : Précédemment nous avons vu qu'un hash est un nombre hexadécimal composé de 64 chiffres. Une fois le hash obtenu, on doit le convertir en base décimale. Sa valeur maximale est de 16^{64}. Le hash doit donc être inférieur à la difficulté qui elle aussi est un nombre. Ce nombre peut avoir également une valeur maximum de 16^{64}. Ainsi, plus le nombre qui représente la difficulté est faible plus il est difficile de trouver un nonce permettant d'obtenir un hash valide.

NB 2 : À l'échelle du réseau, la difficulté est adaptée de telle sorte qu'il faut en moyenne 10 minutes pour qu'un nœud trouve un block valide. Ainsi plus la puissance de calcul au sein du réseau est élevée plus la difficulté sera élevée afin de conserver cette moyenne de 10 minutes.

*Nonce

Nombre arbitraire utilisé une seule fois dans une communication cryptographique.

Comment est calculé le HASH du block ?

Une fois le hash obtenu, il doit être **converti en base décimale**. Il faut alors multiplier chaque chiffre du hash par 16^i, i étant la position du chiffre dans le hash, en partant de la droite et à partir de 0.

A chaque lettre, est associé un chiffre (A = 10, B = 11, C = 12, D = 13, E = 14, F = 15).

Exemple :

2BA5 = $2 \times 16^3 + 11 \times 16^2 + 10 \times 16^1 + 5 \times 16^0$.

Considérant que **pour la blockchain Bitcoin, un hash est composé de 64 chiffres**, sa valeur est ainsi majorée par 16^{64}.

Le **seuil** auquel le hash de l'en-tête du block doit être inférieur est déterminé par le protocole (par exemple **protocole « Blockchain Bitcoin »**). Il est défini en fonction de la *difficulté** du réseau.

Le seuil fixe une **limite inférieure à** 16^{64}. L'ordinateur devra donc calculer tous les nonces jusqu'à en trouver un inférieur à 16^{64} (à chaque échec, l'ordinateur **incrémentera** le nonce).

Comment le hash peut-il varier si les données d'entrée sont réelles ?

Si les données du block sont invariables parce que réelles, en revanche la preuve de travail ne l'est pas. **C'est un « nonce », une variable aléatoire, qui viendra se rajouter à l'information réelle** pour que, à l'application de la fonction, la valeur numérique sortie puisse être différente.

Dans le cas de la fonction SHA-256, le *nonce* est un chiffre.

*Difficulté du réseau

Representation de la difficulté à trouver un block.

Plus la puissance de calcul totale du réseau augmente, plus la difficulté augmente pour conserver un temps de block égal

Le petit guide de la Blockchain a été réalisé par

(dans l'ordre alphabétique) :

Abir Bouattour

Axel Bédat-Passi

Dumitru Parvan

Giovanni Novi

Hulé Kechichian

Juliette Bérard

Magali Cadoret

Matteo Olekhnovitch

Raphaël di Vita

Source : Ethereum France - Comprendre la Blockchain Ethereum : Bitcoin, première implémentation de la blockchain - Gautier Marin