Chapitre X

NoSQL

Les succès des BD relationnelles

- Modèle solide et bien standardisé.
- Technologie bien au point et implémentée efficacement.
- Gestion des transactions offrant des garanties fortes (ACID).
- Permet l'intégration de données en servant de support unique à plusieurs applications.

La contestation

- Entreprises émergentes (Amazon, Google, Linkedin, …) avec des besoins particuliers :
  - Très vastes quantités de données (petabyte 10^{15}, hexabyte 10^{18}, zettabyte 10^{21}, …),
  - Utilisation de très grandes grappes *(clusters)* de machines,
  - Besoin de rapidité face à un usage intense et fiabilité par réplication pour des données dont le schéma n'est pas figé,
  - Coût très élevé des systèmes relationnels.
- Développement de nouveaux systèmes de gestion de données NoSQL (Not, Not only SQL)

Que sont les BD noSQL

- Modèle de données
  - Données organisées en agrégats.
  - Un agrégat est un ensemble des données que l'on consulte habituellement ensemble, par exemple l'information concernant un client et ses commandes.
  - On accède à un agrégat à partir d'une clé.
  - Les données sont donc vues comme des ensembles de paires *(clé, valeur)* où les *valeur* peut avoir un schéma plus ou moins complexe, fixé ou non.

BD NoSQL - Implémentation

- Tables *hash* distribuées:
  - Les paires *(clé, valeur)* sont réparties sur les différentes machines d'une grappe. On utilise le terme *sharding* (éclatement).
  - Les mêmes données peuvent apparaître plusieurs fois pour assurer la fiabilité et la rapidité d'accès.
  - Mais, se pose alors le problème de la cohérence: que se passe-t-il lors d'une mise à jour?

BD NoSQL - Transactions

- La référence dans le modèle relationnel est la transaction ACID (Atomicité, Cohérence, Isolation, Durabilité).
- Dans les bases de données NoSQL l'exigence de transactions ACID est abandonnée au profit de transactions BASE:
  - ***B****asically* ***A****vailable* (la disponibilité est assurée),
  - ***S****oft-state* (l'état du système peut évoluer, même sans operations effectuées),
  - ***E****ventually consistent* (la cohérence est atteinte à terme).

Comparaison NoSQL - Relationnel

- Dans le relationnel les données doivent être recherchées dans plusieurs tables. Comment les répartir dans une grappe?
- Les transactions ACID deviennent très coûteuses à garantir dans un contexte de réplication. Il faut synchroniser toutes les copies.
- Les garanties fortes offertes par les transactions du relationnel ne sont pas toujours nécessaires.
- NoSQL privilégie un temps de réponse court pour tous les utilisateurs par rapport à une cohérence parfaite.
- Utiliser des données pas (ou moins) structurées a ses avantages.

NoSQL pour tous?

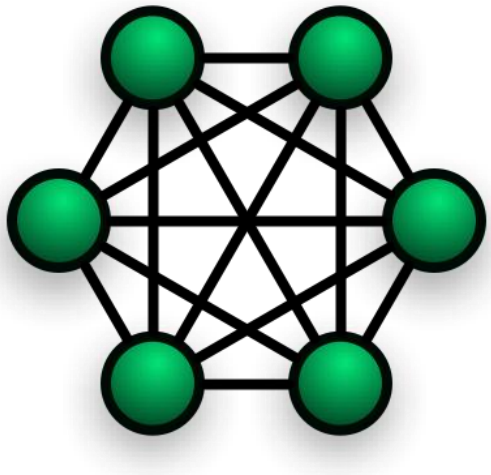
- NON
  - La cohérence est souvent indispensable.
  - Il n'y a pas de modèle standardisé pour noSQL, d'où l'existence d'une multitude de systèmes incompatibles.
  - NoSQL est mal adapté aux situations où les mêmes données sont utilisées dans des applications ayant des besoins différents.
  - NoSQL a ses applications, mais les systèmes relationnels restent largement les plus utilisés.

Blockchain et monnaie virtuelle

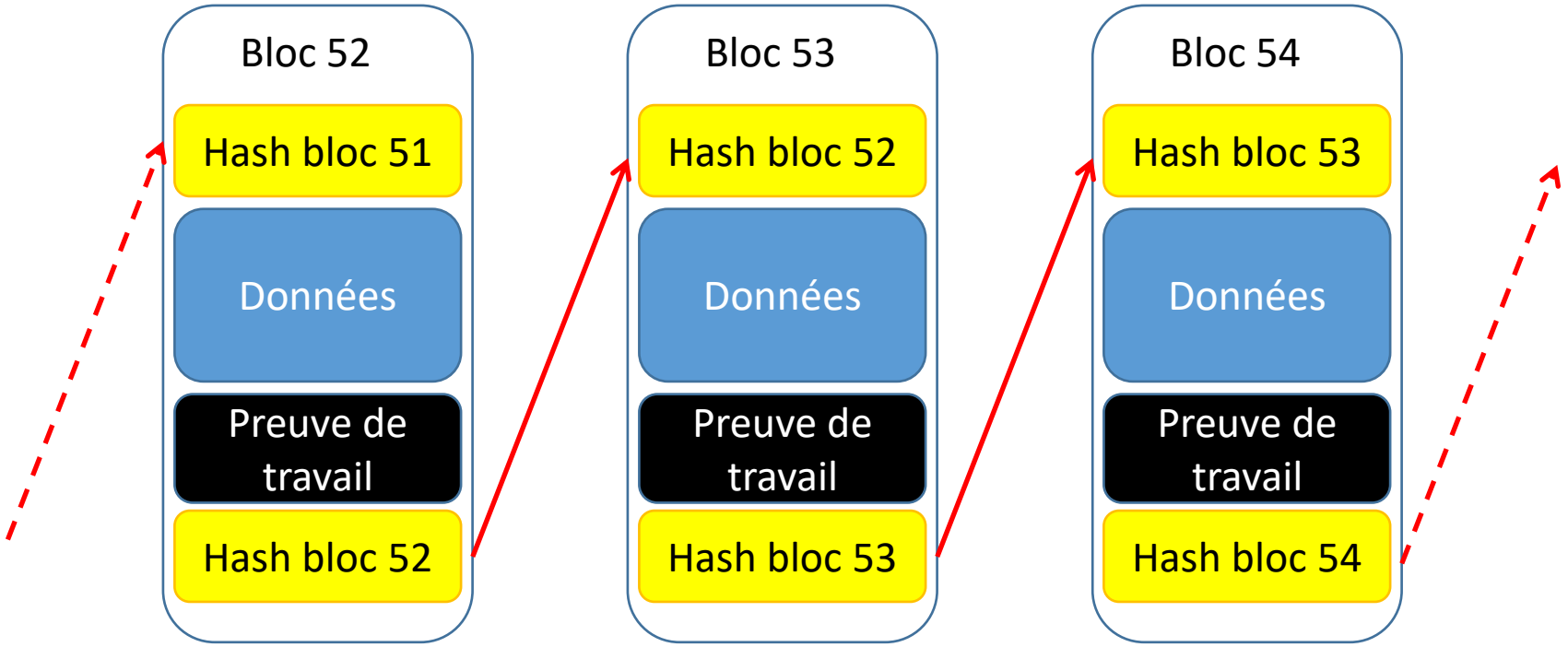
Chapitre 10b

Principe du blockchain

- Stockage de données numérique
  - Pérenne
    - Les données restent de manière permanente, ce qui peut aussi poser un problème de taille de données
  - Infalsifiable
    - Ou, du moins, très difficile à falsifier
  - Distribué
    - On parle plutôt de réseau maillé



Contenu d'une blockchain



Hash

- Résultat de la fonction de hashage SHA-256 sur l'ensemble du bloc (hash du bloc précédent compris)

- 2 propriétés intéressantes :

  - Un petit changement de valeur d'entrée implique un code de sortie complètement différent
    - SHA-256(« J'aime les bases de données ») = D86D1CAFC662F8762C18B6FD821306694DF49087B5DD86D37EE3788E0483AC7D
    - SHA-256(« J'aime les bases de données! »)= 11C181F698361C0ED0FB45A18E531421D37D35F8C041A430B8C6C1636AAAC786

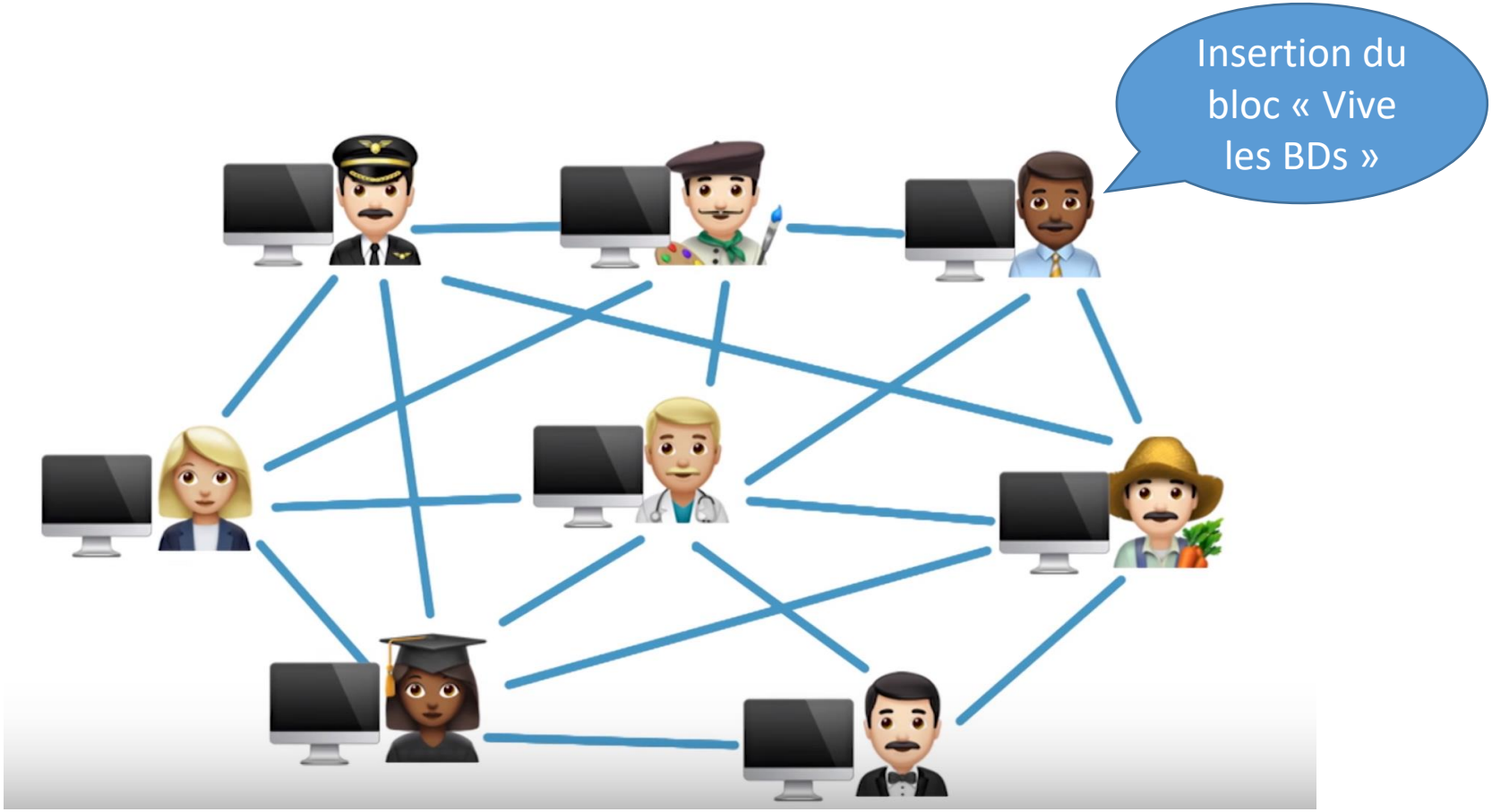
  - Collisions possibles
    - Mais aucune trouvée encore à ce jour

Difficulté

- Chaque personne possède une version de la blockchain
- N'importe qui peut rajouter un bloc
  - Cela peut très vite devenir chaotique
- On ajoute la notion de difficulté : un bloc ne peut être ajouté que si son hash est inférieur à une certaine valeur (nommée « cible » ou « target »).
- C'est en modifiant la preuve de travail que l'on peut essayer d'obtenir un hash suffisamment faible.
- Il faut donc un grand nombre d'essais pour y arriver

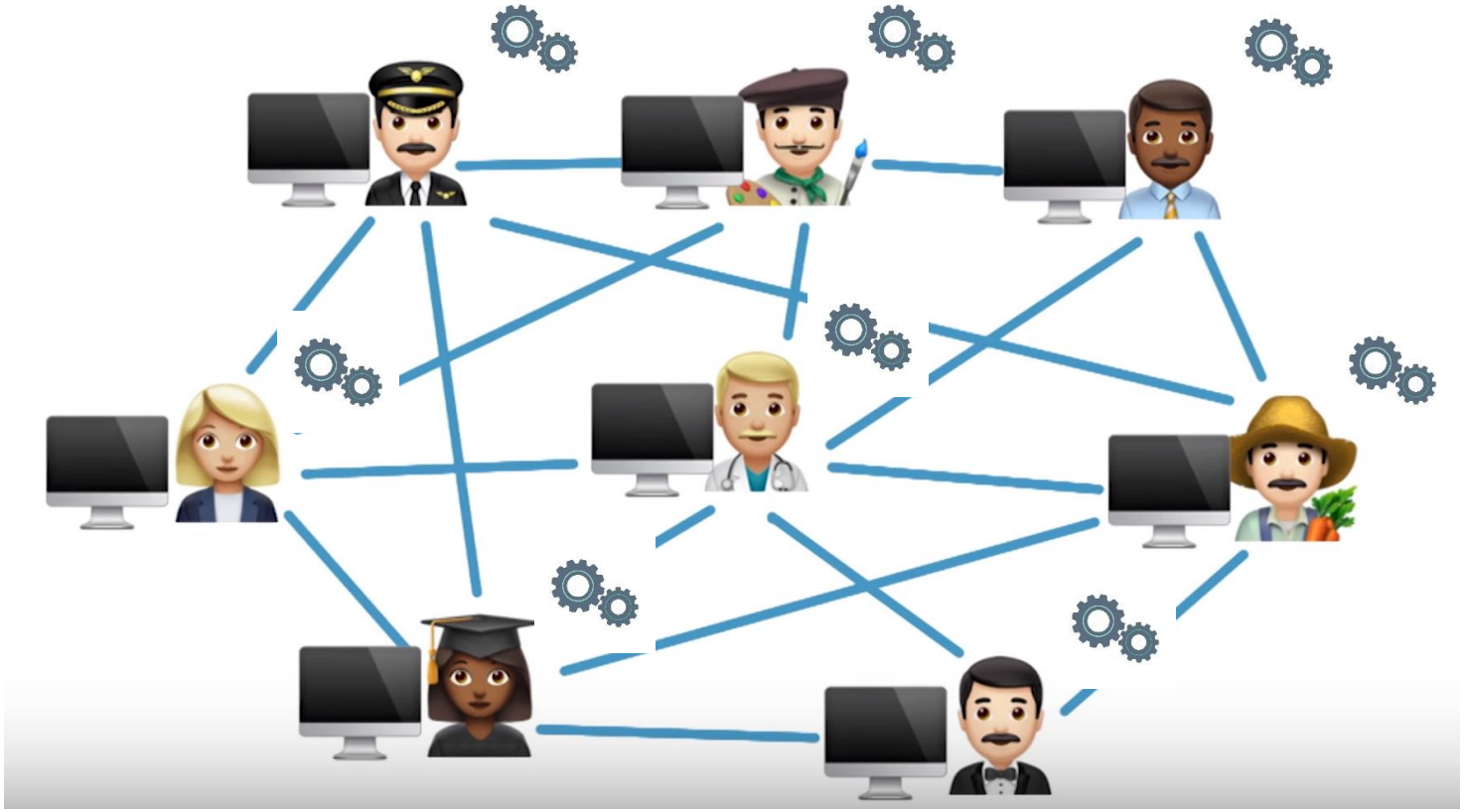
Ajout d’un bloc

- Etape 1 : envoi de la requête à tout le monde



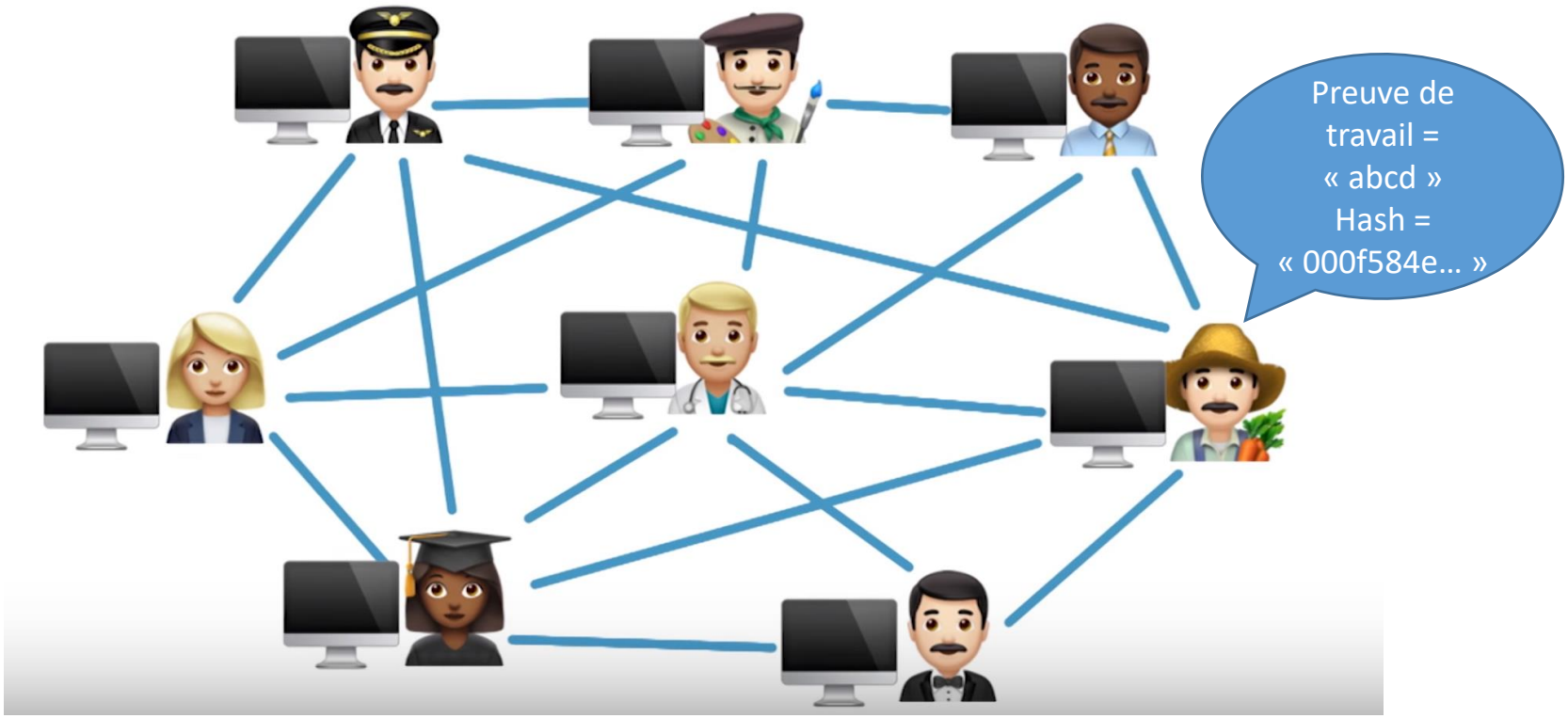
Ajout d’un bloc

- Etape 2 : calcul d’un hash inférieur à la cible



Ajout d’un bloc

- Etape 3 : Diffusion du hash trouvé et ajout du bloc

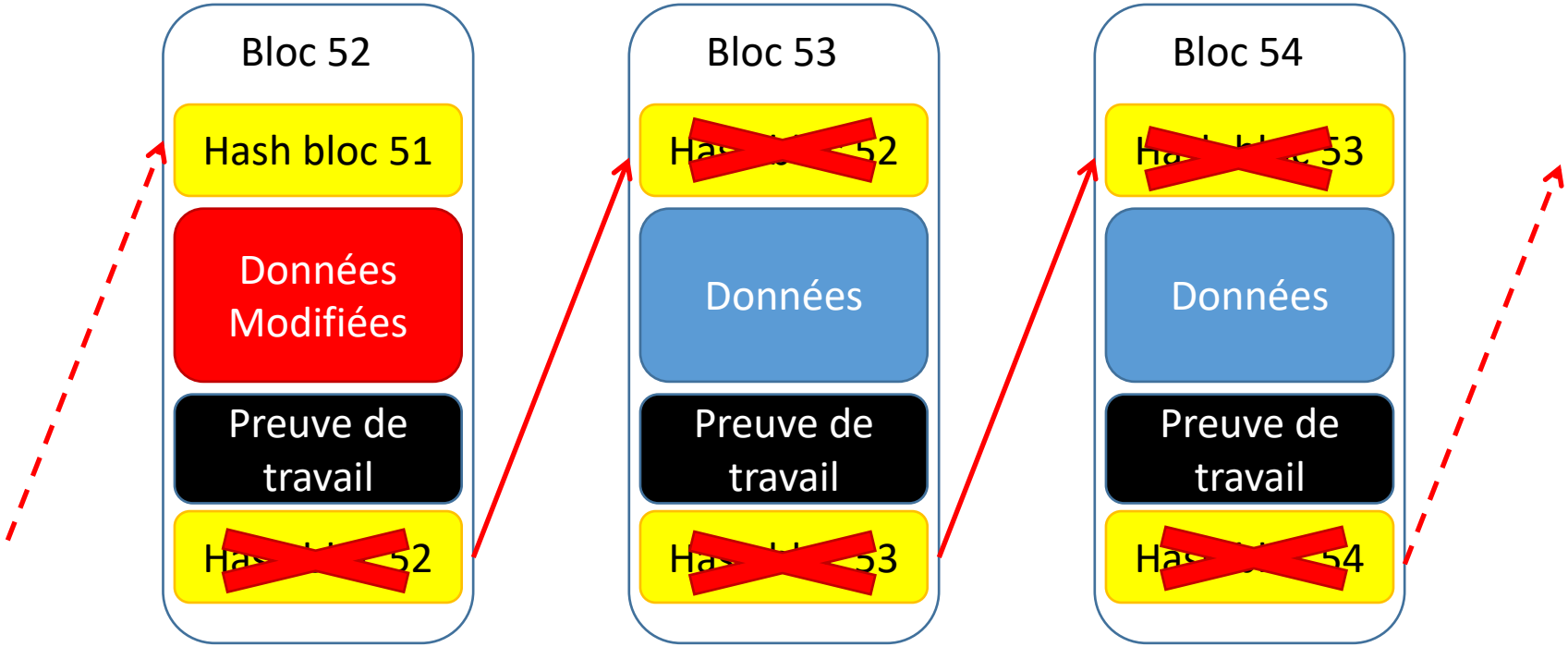


Duplicata

- Il n’est pas impossible que deux personnes trouvent des preuves de travail différentes, mais avec un hash valide.
- La chaine aurait donc deux branches.

- Avec le temps, une branche sera plus utilisée que l’autre, conservation de la plus longue.

Sécurité



- Admettons que je veuille modifier les données du bloc 52
  - Le hash est donc invalide
  - Mais ce hash intervient aussi dans le bloc 53
  - Le hash du bloc 53 sera alors invalide, et ainsi de suite
  - Je dois donc recalculer l'ensemble des hashs à partir du bloc modifié et, en pratique, avoir plus de puissance de calcul que l'ensemble des participants
  - Le dernier bloc pourrait être falsifié, mais plus on remonte loin dans la chaine, plus il sera difficile de falsifier un bloc

Cryptomonnaies

- Blockchains stockant des informations monétaires
- Les deux plus connues : Bitcoin et Ethereum

- Problèmes à résoudre:
  - Que faire en cas de duplicata et comment les limiter?
  - Comment inciter les gens à calculer les hash des blocs?
  - Comment garantir la sécurité?

Limitation des duplicatas

- La difficulté de hashage est telle qu'il faut, en moyenne, environ 10 minutes pour qu'un nouveau bloc soit trouvé.

- Duplicatas toujours possibles, mais peu fréquents.
  - Dans tous les cas, le principe de base s'applique : la chaine la plus longue est conservée.

- Comment garantir ce temps moyen de 10 minutes?
  - Si je mets en production un gros cluster, j'apporte plus de capacité de calcul mondialement et le temps moyen sera inférieur à 10 minutes.

- Solution : Modification de la difficulté du hashage tous les 2016 blocs ajoutés.

Minage

- Lorsque quelqu'un trouve une preuve de travail satisfaisante, il reçoit 6,25 bitcoins (= environ 280.000€ au taux actuel).

- Cependant, cela est très rare. Sachant que, pour être valide, un hash doit commencer par 72 zéros, on peut estimer qu'il faut calculer la fonction de hashage environ 2^71 fois, ce qui peut prendre des milliers d'années.
  - D'où groupement de mineurs pour augmenter la probabilité, et partage des récompenses.

- Miner du bitcoin est-il intéressant?
  - Aujourd'hui, peut-être, mais soit en groupe, soit avec une ferme à bitcoin.
  - La récompense est divisée en deux tous les 210,000 blocs. Avant le 11 mai 2020, miner un bloc rapportait 12,5 bitcoins.
    - Mais la valeur du bitcoin a **quintuplé** depuis…
  - Prochain *halving* estimé au printemps 2024

Sécurité du contenu

- Le monde entier ne peut générer, en moyenne, qu'un bloc toutes les 10 minutes.
- Falsifier le dernier bloc impliquerait d'avoir plus de puissance de calcul que l'ensemble de la planète…
- … ou d'avoir de la chance.

- D'autre part, le contenu de chaque bloc est un ensemble de transactions (env. 2000) de type « *A* donne *X* bitcoins à *B* ».
  - Si *A* a rédigé ce message, c'est parfait
  - Si, c'est *B*, c'est problématique.

Cryptographie

- Chaque utilisateur possède deux clés cryptographiques
  - Une clé privée et une clé publique

- La transaction « *A* donne *X* bitcoins à *B* » est signée par *A*.
  - *A* utilise sa clé privée pour chiffrer un hash du message.

- Tout utilisateur peut dès lors utiliser la clé publique de *A* pour vérifier si le message n'a pas été altéré.

- Même si une modification est possible, c'est très peu probable en pratique.

Les problèmes à résoudre

- Dans ce type de base de données, on ne fait qu'ajouter de l'information.
  - La blockchain grandit d'environ 50Go chaque année, et atteint aujourd'hui environ 330Go
  - Comment repartir d'une chaine plus petite tout en assurant la sécurité?

- Quid de la protection de la vie privée?
  - Fonctionne très bien avec bitcoin, car l'argent ne transite que d'un compte à l'autre, mais on ne sait pas à qui appartient le compte.
  - Pour d'autres types de données (ex. médicales), les informations apparaitraient en clair.

- Pour bitcoin : Le nombre de bitcoins à miner est fini (21 millions de bitcoins).
  - Le dernier bitcoin sera miné aux alentours de l'année 2140.
  - Comment dès lors inciter les gens à continuer à calculer les hashs?