

# Introduction aux Blockchains

---

5 octobre 2020

## 7 semaines + projet

1. Introduction aux blockchains
2. Protocoles Économiques et Smart-Contracts
3. Cryptographie et Privacy
4. Algorithmes de consensus
5. Systèmes distribués en milieu adversarial
6. Cas d'étude : Tezos
7. Finalisation du projet

# Domaines dans la blockchain

- Cryptographie
- Réseau
- Bases de données
- Systèmes distribués
- Langages de programmation
- Vérification formelle

# Blockchain : définition

- Base de données distribuée : chaque utilisateur possède une copie locale d'un registre ;
- Les utilisateurs forment un réseau pair-à-pair et s'échangent des blocs (aggrégats d'opérations) pour faire évoluer ce registre ;
- Chaîne de blocs : chaque nouveau bloc doit succéder à un bloc existant et connu du réseau ;
- Protection contre la falsification : tous les participants sont tenus de vérifier la validité des blocs reçus ;
- Implémentation connue : toute personne doit pouvoir vérifier le fonctionnement.

# Une application : les crypto-monnaies (1/3)

Monnaie digitale décentralisée :

- Le registre de la blockchain est un *livre de comptes* associant une adresse à un solde en *jetons*;
- Les comptes sont des objets cryptographiques permettant de signer des transactions et d'en vérifier leur validité.
- Les participants ont une **incitation économique** à participer au réseau : e.g. récompense à la création d'un bloc ;

**Hypothèse** : les participants accordent une valeur aux jetons.

⇒ Les actions nuisibles au réseau doivent se traduire par une perte économique.

# Une application : les crypto-monnaies (2/3)

## Critique des systèmes bancaires

- Centralisation : une seule autorité
- Système en boîte noire
- Aucune garantie de fiabilité

## Avantages de la blockchain

- Décentralisation : chacun participe au système ;
- Sécurité : chaque participant valide localement les changements et peut rejouer l'état ;
- Transparence : chaque action effectuée est connue de tous les participants.

## Problèmes ouverts

- *Privacy* :
  - L'historique de chaque personne est accessible : comment préserver l'anonymat ?
- Passage à l'échelle :
  - Débit réduit d'opérations ;
  - Rapidité du réseau pair-à-pair.
- Dissensions sociales :
  - Fonctionnement et évolution du réseau.

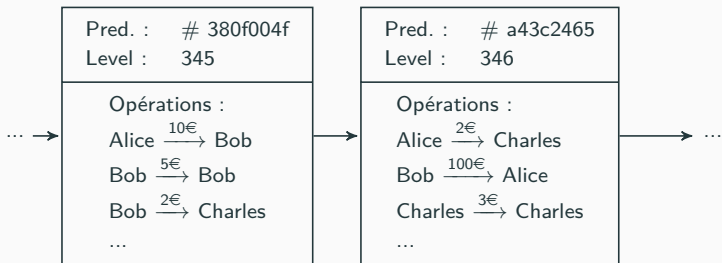
# Quelques projets de cryptomonnaies

- Bitcoin : 2009  $\Rightarrow$  Bitcoin Cash, ...
- Ethereum : 2014  $\Rightarrow$  Ethereum Classic, ...
- Zcash : 2016
- Tezos : 2018
- ...



# Structure de base de la blockchain

## Agrégats d'opérations ordonnés



# État de la blockchain

- L'état (ou registre) d'une blockchain est stocké dans une base de données
- La base de données est modifiée après chaque application de bloc
- L'application d'un bloc revient à appliquer l'ensemble des opérations contenues

# Application d'un bloc

$\mathcal{B}$

Pred. : # 380f004f
Level : 345
Opérations :
Alice $\xrightarrow{10\text{€}}$ Bob
Bob $\xrightarrow{5\text{€}}$ Bob
Bob $\xrightarrow{2\text{€}}$ Charles

$\mathcal{S}$

Alice	1230€
Bob	5432€
Charles	543€

$\mathcal{B}(\mathcal{S})$

Alice	1220€
Bob	5440€
Charles	545€

**Il faut vérifier ce que l'on reçoit !**

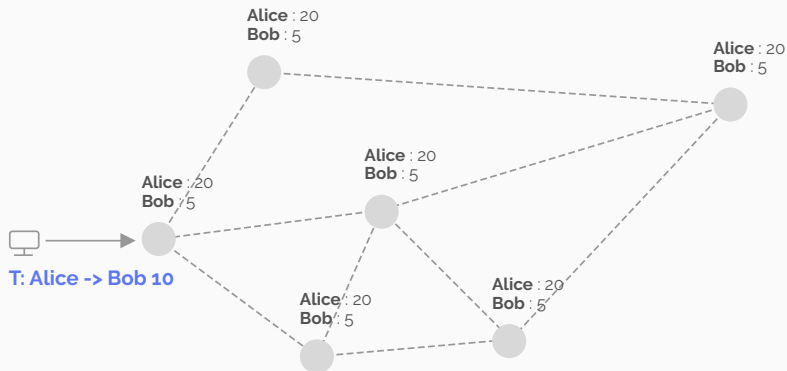
# Base de donnée répliquée

- Chaque utilisateur possède une copie locale de l'état de la blockchain
- Cela permet de vérifier localement ce que les autres utilisateurs envoient
- Cela permet également de créer des transactions cohérentes avec l'état actuel de la chaîne

# Réseau pair-à-pair

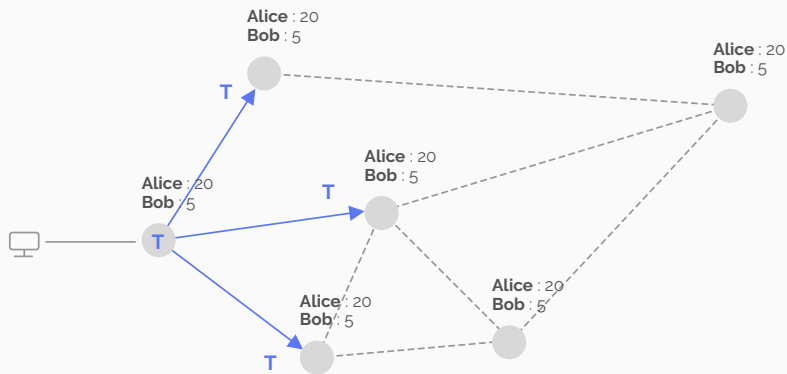
- Les utilisateurs doivent rester synchronisés entre eux
- Ils communiquent via un réseau pair-à-pair en utilisant un **nœud**
- Chaque participant se connecte à un nœud agissant comme un client pair-à-pair
- Plus il y a de nœuds dans le réseau, plus le réseau est sûr

# Propagation dans le réseau



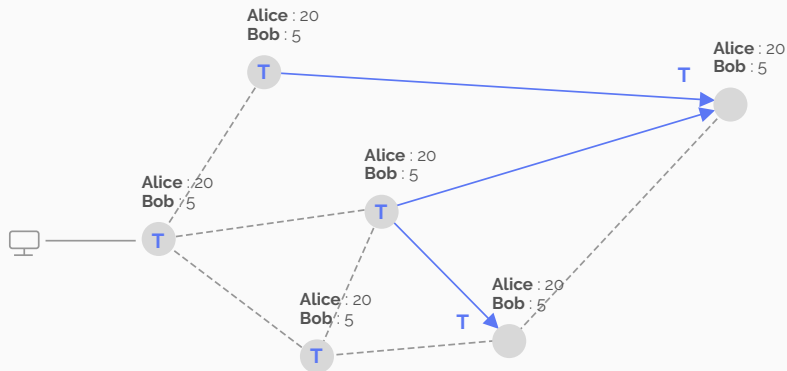
L'utilisateur crée et envoie une transaction à son nœud

# Propagation dans le réseau



Le nœud propage la transaction à ses pairs qui la mette dans leur liste de transactions

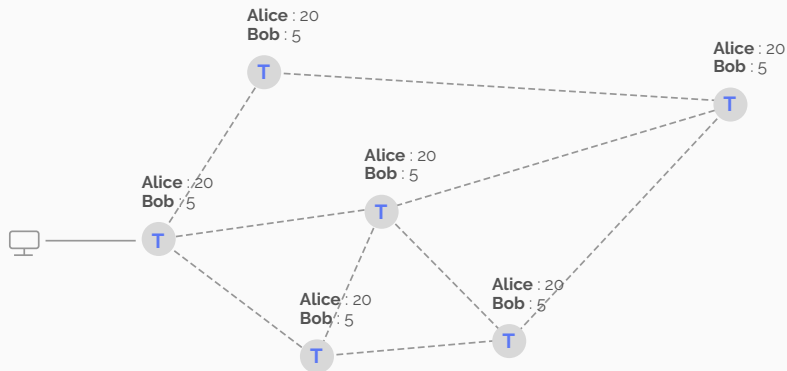
# Propagation dans le réseau



Les autres nœuds informent à leur tour leur pairs...

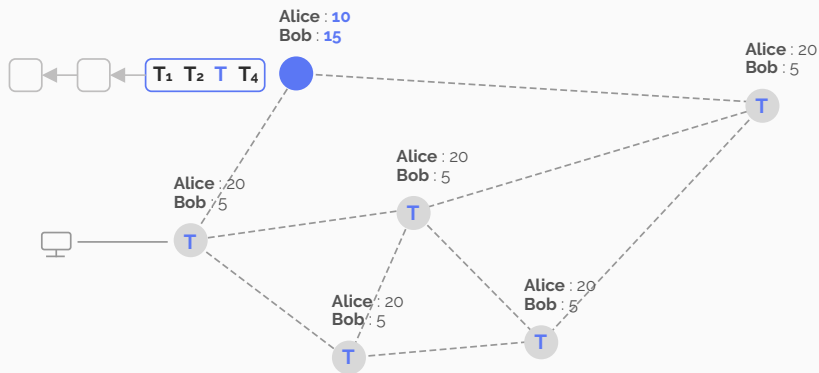


# Propagation dans le réseau



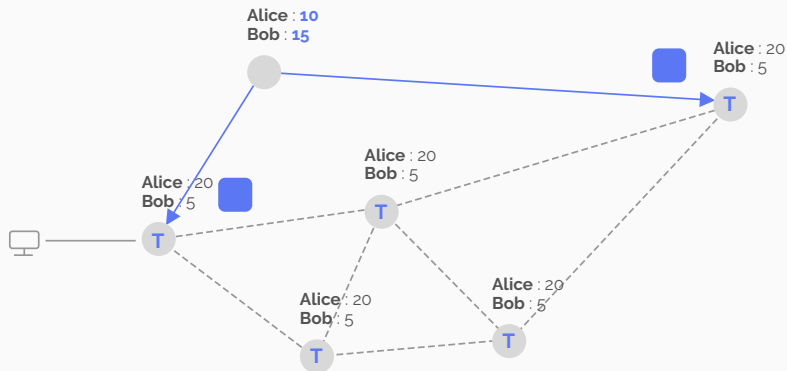
... jusqu'à la propagation globale dans le réseau

# Propagation dans le réseau



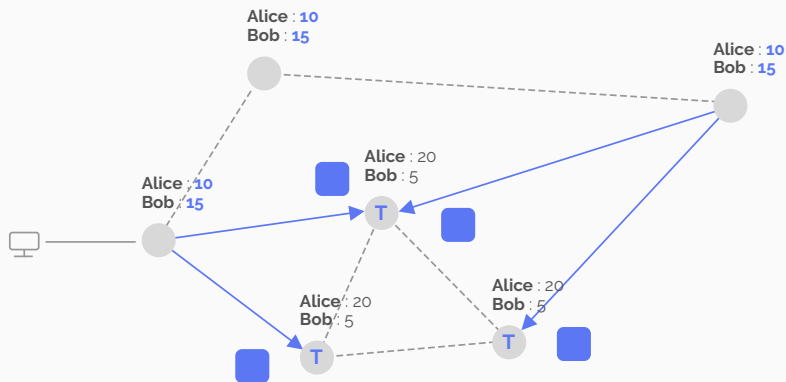
À un moment, un nœud va décider de créer un bloc avec les transactions reçues

# Propagation dans le réseau



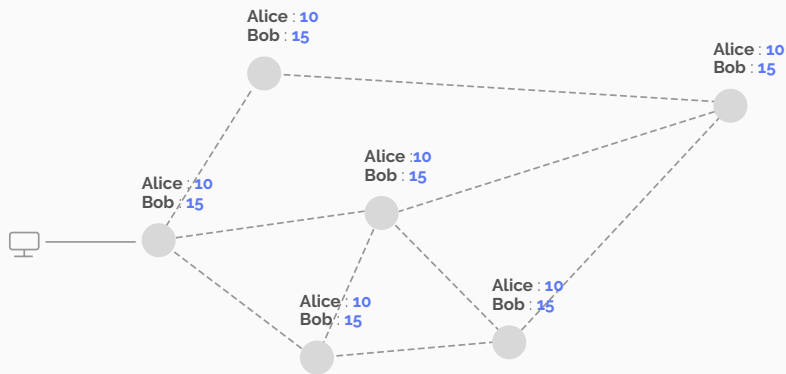
Ce nouveau bloc est propagé et appliqué dans le réseau

# Propagation dans le réseau



Chaque participant vérifie et applique localement ce bloc à sa base de données

# Propagation dans le réseau



Au bout d'un moment, chaque participant a reçu le bloc et la majorité du réseau est à jour

# Architecture du système

Une blockchain a besoin de deux composants principaux :

- Un **nœud**
- Un **protocole économique**

## Une couche réseau et un stockage

- Couche réseau P2P : *Gossip*
  - Propagation des opérations
  - Propagation des blocs
  - Découverte de pairs réseau
- Stockage :
  - Stocke les blocs et les opérations pour les envoyer au besoin aux autres participants
  - Maintient les états passés de la chaîne pour être capable de revenir en arrière

# Protocole économique

Le protocole économique est l'ensemble des règles de la chaîne

Chaque participant accepte de suivre ses règles pour maintenir le bon comportement du réseau

Il détermine notamment :

- la validité des transactions
- la validité des blocs
- attribue un score aux blocs appliqués
- les récompenses à la participation
- ...



# Interaction entre le nœud et le protocole

- Le protocole valide les blocs et les opérations
- Le nœud stocke les blocs et les états de la chaîne

## Scénario : un nouveau bloc arrive du réseau

1. Le nœud demande au protocole de valider ce bloc ;
2. Le protocole valide le bloc et retourne un **score** ou si le bloc est invalide rejette le bloc et *kick* au besoin le pair ;
3. Le nœud teste si le nouveau **score** est meilleur que le score du bloc résultant de son état courant :
  - Si oui : le bloc est sélectionné comme nouvelle tête de chaîne et propagé dans le réseau,
  - Sinon : le bloc est ignoré.

# Problème des chaînes concurrentes

Il est possible qu'à un moment le réseau ne soit plus d'accord sur l'état de la chaîne : quel bloc est en tête ?



Pour résoudre ce problème, il est nécessaire d'établir un **algorithme de consensus** que tous les participants sont encouragés à appliquer pour le bon fonctionnement du réseau.

# Algorithme de consensus

Un algorithme de consensus doit définir deux notions clés :

- Établir un score à chaque bloc
- Déterminer la validité d'un bloc

Mais pourquoi avoir envie de publier des blocs ?

**Incitation économique** : chaque personne participant au bon fonctionnement de la chaîne (vivacité) reçoit une récompense

# Proof-of-Work

La majorité des implémentations de blockchains actuelles dispose d'un algorithme de consensus basé sur le **proof-of-work**

- Le premier qui arrive à résoudre un puzzle "cryptographique" a le droit de publier un bloc
- Bonnes propriétés : dur de tricher, facile à vérifier, ...

# Fonctions de hachage cryptographique

Une fonction de hachage  $h$  prend une **donnée** en entrée de taille arbitraire  $n$  et produit une image de taille constante  $c$  en sortie : un **hash**

$$h : \{0, 1\}^n \mapsto \{0, 1\}^c$$

Propriétés attendues :

- Déterminisme :  $x = y \Rightarrow h(x) = h(y)$
- Uniformité :  $x \neq y \Rightarrow h(x) \neq h(y)$  (impossible en pratique)
- Non-inversibilité : pour un hash donné, aucune information sur l'entrée ne doit pouvoir être déduite (cryptographie)

## Fonction de hachage cryptographique (2)

Il existe un grand nombre d'algorithmes de hachage cryptographiques

### Secure Hash Algorithm (SHA)

```
$ echo "bla" | sha256sum  
00e3261a6e0d79c329445acd540fb2b0 ...
```

```
$ echo "bli" | sha256sum  
5f2c0653c7f703abf9ac2b083c256a7f ...
```

# Bitcoin – Algorithme de consensus

Dans Bitcoin, un bloc est dit valide si les bits de son hash commencent par  $n$  zéro (**difficulté**)

Block  $\mathcal{B}$

Opérations :

Alice  $\xrightarrow{1bc}$  Bob

Roger  $\xrightarrow{5bc}$  John

$h(\mathcal{B}) = 001010111010101\dots$

Si  $n = 2$  alors **valide**

Si  $n = 5$  alors **invalide**

**Comment peut-on rendre ce bloc valide ?**

## Bitcoin – Algorithme de consensus (2)

On ajoute au bloc une valeur (**nonce**) facilement modifiable

Block  $\mathcal{B}$

Nonce : <entier>

---

Opérations :

Alice  $\xrightarrow{1bc}$  Bob

Roger  $\xrightarrow{5bc}$  John

$\mathcal{B}_i$  = Bloc  $\mathcal{B}$  avec  $i$  le *nonce* :

- $h(\mathcal{B}_0) = 10101011... \Rightarrow \times$
- $h(\mathcal{B}_1) = 00011101... \Rightarrow \times$
- ...
- $h(\mathcal{B}_{23}) = 0000010... \Rightarrow \checkmark$



## Bitcoin – Algorithme de consensus (2)

On ajoute au bloc une valeur (**nonce**) facilement modifiable

Block  $\mathcal{B}$

Nonce : <entier>

---

Opérations :

Alice  $\xrightarrow{1bc}$  Bob

Roger  $\xrightarrow{5bc}$  John

$\mathcal{B}_i$  = Bloc  $\mathcal{B}$  avec  $i$  le *nonce* :

- $h(\mathcal{B}_0) = 10101011\dots \Rightarrow \times$
- $h(\mathcal{B}_1) = 00011101\dots \Rightarrow \times$
- ...
- $h(\mathcal{B}_{23}) = 0000010\dots \Rightarrow \checkmark$

**Plus  $n$  est grand, plus il est “difficile” de trouver un bloc valide**

## Bitcoin – Algorithme de consensus (3)

Bitcoin vise à ce que le réseau dispose d'un bloc toutes les 10 minutes. Pour arriver à ce but, la difficulté est adaptée dynamiquement :

- Après la publication d'un bloc, la difficulté pour le bloc suivant démarre très haut et décroît chaque minute ;
- Si un bloc valide est trouvé en dessous de 10 minutes, la difficulté sera accrue pour les blocs futurs ;
- Au contraire, si un bloc est publié au-delà de 10 minutes, la difficulté sera réduite.

## Critiques

- Course au calcul : premier arrivé, premier servi
- Favorise les personnes disposant d'un matériel dédié
- Consomme énormément d'énergie
- Centralise le réseau

## Quelques chiffres

- 1 Bitcoin  $\approx$  10.000\$
- La récompense pour un bloc est actuellement 12,5 Bitcoin

# Différents algorithmes de consensus

Proof-of-Work : coûteux mais simple et robuste

D'autres algorithmes de consensus existent :

- Proof-of-Stake : droit de créer un bloc selon l'investissement des utilisateurs
- Proof-of-Space : similaire au PoW mais espace disque plutôt que calcul
- Proof-of-Authority : droit de créer un bloc basé sur la réputation

## Thèmes abordés

- Intuition générale de la blockchain
- Architecture globale
- Éléments de l'algorithme de consensus du Bitcoin

## Protocoles économiques

- Architecture des protocoles économiques ;
- Vote et amendements ;
- Smart-contracts.

## Cryptographie et Privacy

- Algorithmes de signature ;
- Structures cryptographiques ;
- Préserver l'anonymat dans la blockchain.

## Consensus

- Détail des principaux algorithmes de consensus ;
- Nature des attaques potentielles.

## Systemes distribués adversarial

- Interactions entre les différents acteurs du système ;
- Architecture d'un nœud ;
- Résistance aux attaques.