

# Introduction à la Blockchain

# Plan

- Histoire du Bitcoin
- Qu'est-ce qu'une blockchain ?
- EVM et smart contracts
- DeFi/CeFi
- Outils de la finance décentralisée
- Miner Extractable Value
- ?

Raphael Westphal (westphal.rafael@gmail.com)

Parcours scolaire/professionnel

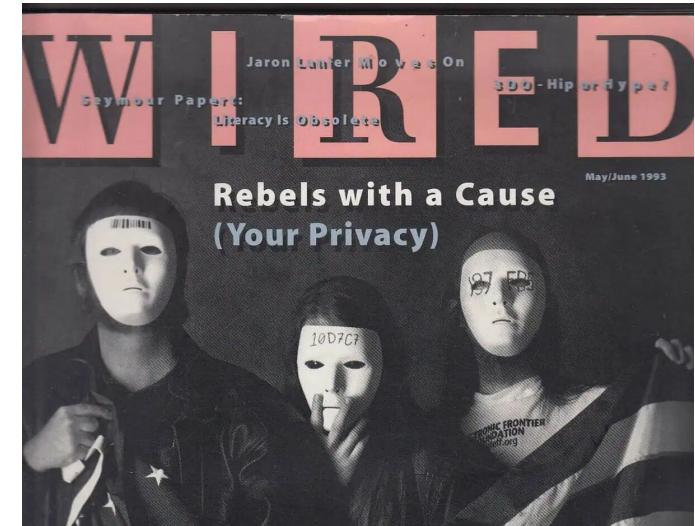
IMT Nord Europe (anciennement Télécom Lille) en alternance

Quatre années à OVH (VoIP et OverTheBox)

# Histoire du bitcoin: 1992, Cypherpunks

Groupe d'individus intéressés dans la cryptographie et les technologies permettant l'amélioration et la préservation de la vie privée.

Les cypherpunks sont également connus pour leur fort engagement en faveur des droits de l'individu, leur opposition à la censure gouvernementale et au contrôle d'Internet. Ils sont souvent associés au mouvement du logiciel libre et à la lutte pour la liberté d'Internet et la vie privée.



# Histoire du bitcoin: 1992, Cypherpunks

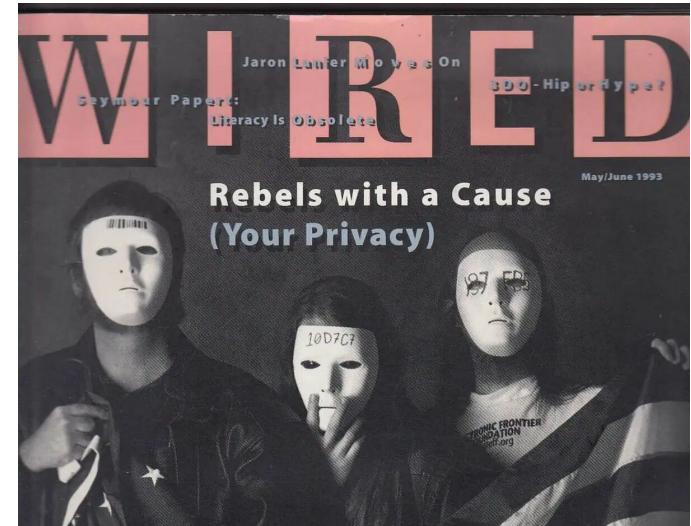
Timothy C. May (ancien ingénieur Intel)

Eric Hughes (mathématicien et programmeur)

John Gilmore (contributeur important du projet GNU et co-fondateur de l'Electronic Frontier Foundation)

Philip Zimmermann (Créateur de PGP)

Julian Assange (Rédacteur en chef et créateur de WikiLeaks)



# Histoire du bitcoin: The Crypto Anarchist Manifesto

Trois constats:

- Le manifeste commence par affirmer que la vie privée est une condition nécessaire pour la liberté individuelle et que la technologie peut être utilisée pour renforcer cette vie privée contre les gouvernements et les entreprises qui cherchent à la violer.
- "Si on peut contrôler l'argent des gens, on peut contrôler tous les aspects de leur vie, si on peut surveiller l'argent des gens, alors on sait tout ce qu'il faut savoir sur eux"
- Les individus doivent avoir le contrôle sur leurs données en étant libre de les diffuser ou non

# Histoire du bitcoin: 1992, Cypherpunks

Le manifeste aborde également les limites de la vie privée, en soulignant que certains renseignements peuvent être partagés volontairement avec d'autres personnes ou organisations pour des raisons légitimes. Cependant, il précise que ces choix doivent être faits de manière consciente et en toute connaissance de cause, et que les individus doivent avoir le contrôle absolu sur les informations qu'ils partagent.

# Histoire du bitcoin: 1992, Cypherpunks

Pourquoi faire confiance à une monnaie ?

Les banques peuvent:

- Imprimer (diluer) votre argent
- Vous refuser un retrait (exemple de la Grèce en 2008 et le Nigéria cette semaine)
- Censurer vos transactions (exemple Wikileaks)

L'argent nous appartient-t-il vraiment ?

# Histoire du bitcoin: 2008, Satoshi Nakamoto

En 2008, Satoshi Nakamoto publie son white paper sur une nouvelle monnaie électronique “Bitcoin” permettant de résoudre le problème de la double dépense.

La première version de bitcoin est publiée en 2009 et permet aux utilisateurs de créer adresses et transactions.

## Bitcoin P2P e-cash paper

Satoshi Nakamoto | Sat, 01 Nov 2008 16:16:33 -0700

I've been working on a new electronic cash system that's fully peer-to-peer, with no trusted third party.

The paper is available at:

<http://www.bitcoin.org/bitcoin.pdf>

The main properties:

Double-spending is prevented with a peer-to-peer network.

No mint or other trusted parties.

Participants can be anonymous.

New coins are made from Hashcash style proof-of-work.

The proof-of-work for new coin generation also powers the network to prevent double-spending.

# Histoire du bitcoin: 2008, Satoshi Nakamoto

Le projet est vite rejoint par d'autres programmeurs (notamment d'autres "Cypherpunks")

Satoshi disparaît d'internet en 2011, jusqu'à maintenant on ne sait toujours qui il(s) est

Source: Le mystère Satoshi



# Qu'est-ce qu'une blockchain ?

La blockchain est une **base de données distribuée** utilisée pour stocker des informations de manière à rendre difficile ou impossible de les modifier, de les pirater ou de **tricher avec le système**. C'est un système **décentralisé** qui permet à plusieurs parties d'enregistrer et de vérifier des **transactions** sans avoir besoin d'une **autorité centrale**. Les informations dans une blockchain sont généralement stockées dans des **blocs** qui sont **liés** ensemble dans une chaîne en utilisant la **cryptographie**. Cela rend les données dans une blockchain sécurisée et transparente, et permet de les partager et d'y accéder par n'importe qui ayant les autorisations appropriées.

# Base de données distribuée

Une base de données distribuée est essentiellement une base de données qui n'est pas limitée à un seul système, elle est répartie sur différents sites, c'est-à-dire sur **plusieurs ordinateurs** ou sur un réseau d'ordinateurs. Un système de base de données distribuée est situé sur différents sites **qui ne partagent pas de composants physiques**. Cela peut être nécessaire lorsqu'une base de données particulière doit être consultée par différents utilisateurs à l'échelle mondiale. Elle doit être gérée de telle sorte que pour les utilisateurs, elle ressemble à une base de données unique.

# Décentralisé

Le stockage décentralisé est un système qui partage entre de nombreux opérateurs indépendants, la conservation de données informatiques.

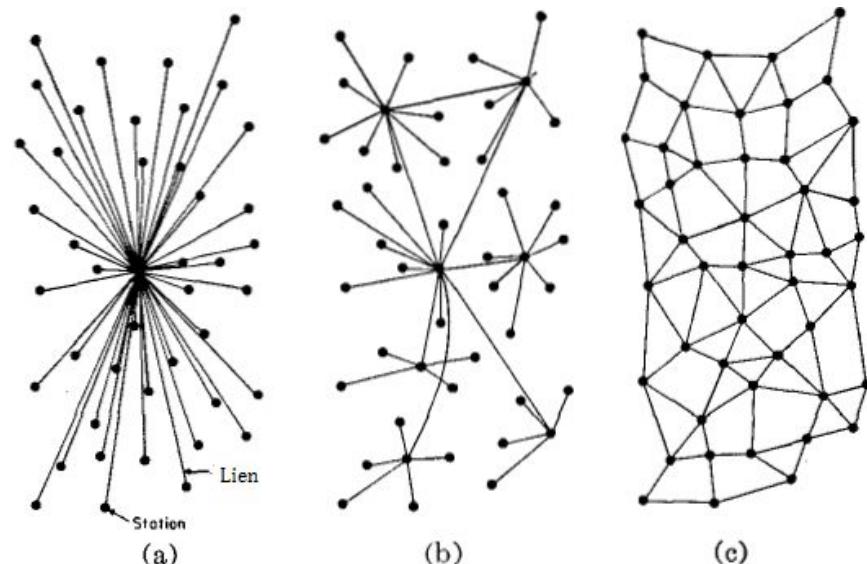


Fig. 1—(a) Centralisé   (b) Décentralisé   (c) Réseaux distribués

# Transaction

En informatique, et particulièrement dans les bases de données, une transaction telle qu'une réservation, un achat ou un paiement est mise en œuvre via une suite d'opérations qui font passer la base de données d'un état A — antérieur à la transaction — à un état B postérieur et des mécanismes permettent d'obtenir que cette suite soit à la fois **atomique, cohérente, isolée et durable (ACID)**

# Atomicité (A.C.I.D)

Le principe d'Atomicité garantit la bonne exécution de la transaction. Les transactions de base de données, comme les atomes, peuvent être décomposées en plus petites parties. Si une partie d'une transaction échoue, toute la transaction sera annulée.

IE: La transaction est exécutée entièrement ou ne l'est pas du tout

# Cohérence (A.C.I.D)

La propriété de Cohérence signifie que seules les données qui suivent des règles prédéfinies peuvent être écrites dans la base de données.

IE: Si je ré-exécute la même transaction sur le même état A, alors je dois retomber sur B

# Isolement (A.C.I.D)

L'isolation fait référence à la capacité de traiter simultanément plusieurs transactions de manière indépendante.

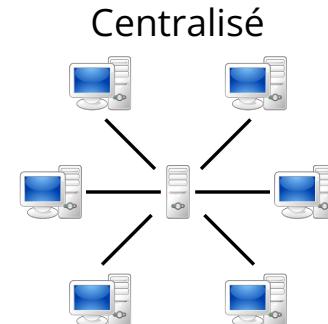
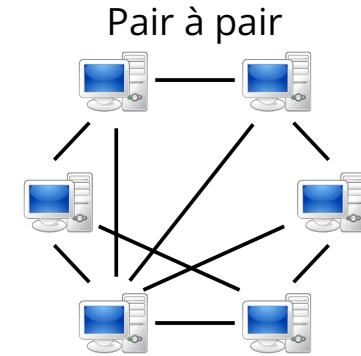
IE: Il n'est possible que d'exécuter "une seule transaction à la fois"

# Durabilité (A.C.I.D)

La durabilité requiert de rendre les défaillances invisibles pour l'utilisateur final. Les données sont sauvegardées une fois la transaction terminée, même en cas de panne de courant ou de défaillance du système.

# Pair à pair (p2p):

Le pair-à-pair définit un modèle de réseau informatique d'égal à égal entre ordinateurs, qui distribuent et reçoivent des données ou des fichiers. Dans ce type de réseau, comparable au réseau client-serveur, chaque client devient lui-même un serveur.



# <https://blockchaindemo.westphal.fr>

The screenshot shows a web application interface for a blockchain demo. At the top, there is a dark header bar with the text "Blockchain Demo" on the left and several navigation items on the right: "Hash" (which is highlighted in white), "Block", "Blockchain", "Distributed", "Tokens", and "Coinbase". Below the header, the main content area has a title "SHA256 Hash". The interface includes two main input fields: one labeled "Data:" which is currently empty, and another labeled "Hash:" which contains the value "e3b0c44298fc1c149afb4c8996fb92427ae41e4649b934ca495991b7852b855".

Source: <https://github.com/anders94/blockchain-demo>

# Minage de blocs

Minage: Processus durant lequel un **mineur** tente de trouver le bon **nonce** pour avoir le nombre de 0s suffisant pour créer un bloc valide



~ 300 KH/s



~1 GH/s = 100 000 KH/s



~ 110 TH/s = 110 000 000 KH/s

# Minage de blocs



Calculated for  
1 BTC = \$ 16,828.96

**Hashing Power**  
110  
TH/s

**Power consumption (w)**  
3250

**Cost per KWh (\$)**  
0.17

PROFIT RATIO PER DAY		PROFIT PER MONTH	
<b>-50%</b>		<b>\$ -195.56</b>	
Profit per day	Mined/day	Power cost/Day	
<b>\$ -6.52</b> Day	<b>B 0.0004046</b>	<b>\$ 13.26</b>	
Pool Fee \$ 0.06809			
Profit per week	Mined/week	Power cost/Week	
<b>\$ -45.63</b> Week	<b>B 0.002832</b>	<b>\$ 92.82</b>	
Pool Fee \$ 0.4767			
Profit per month	Mined/month	Power cost/Month	
<b>\$ -195.56</b> Month	<b>B 0.01214</b>	<b>\$ 397.80</b>	
Pool Fee \$ 2.04			
Profit per year	Mined/year	Power cost/Year	
<b>\$ -2,379.29</b> Year	<b>B 0.1477</b>	<b>\$ 4,839.90</b>	
Pool Fee \$ 24.85			

# Blockchain

Hash: Chaîne de caractères (chiffre) qui permet de vérifier l'authenticité d'une donnée sans en connaître le contenu. Une fonction de hachage peut prendre en entrée une infinité de données et sortira toujours une chaîne de caractères de la même taille.

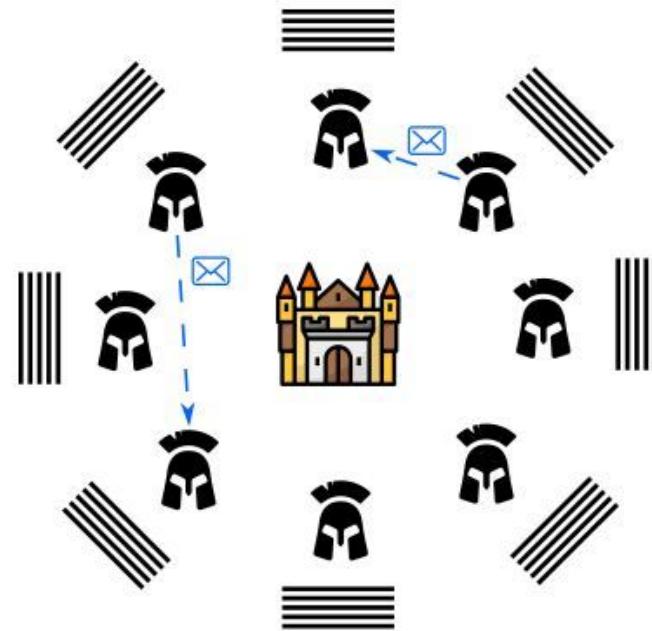
Un bloc contient:

- Un ensemble de données (dans le cas de Bitcoin, des transactions)
- Un numéro d'identification
- Une référence (ou non) à un bloc précédent
- Un nonce
- Une coinbase (identifiant de l'ordinateur ayant trouvé la solution au "problème des zéros")

Le hash du bloc n'est pas stocké dans celui-ci mais en est dérivé

# Problèmes des généraux byzantins

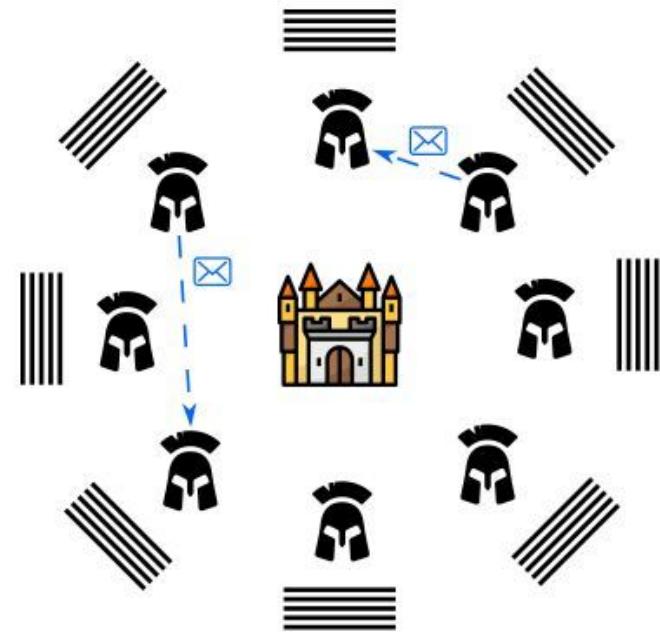
Théorisé par Leslie Lamport en 1982, il pose l'analogie la coordination d'une attaque en généraux byzantins et le bon fonctionnement d'un ordinateur ayant un ou plusieurs composants défectueux



# Problèmes des généraux byzantins

Des généraux byzantin veulent coordonner une attaque sur une ville. Pour gagner la bataille, il faut que toutes les armées s'accordent pour attaquer le même jour à la même heure.

Problème: les canaux de communication ne sont pas fiable et des traîtres se sont glissés parmi les messagers

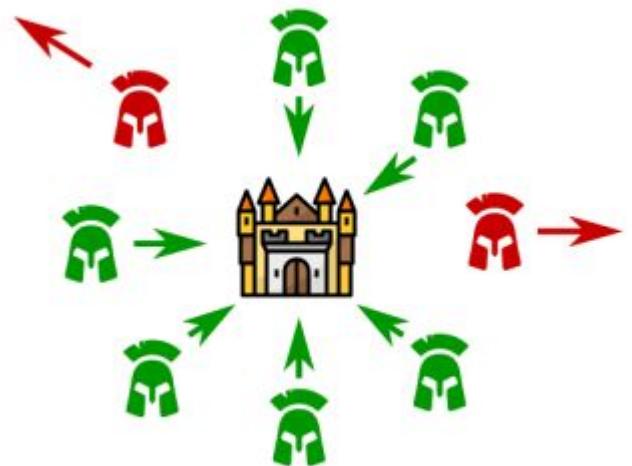


# Problèmes des généraux byzantins

Comment réussir à atteindre le **consensus** sur la date et l'heure d'attaque ?

(Le jour de l'attaque importe peu, ce qui compte est que tout le monde attaque en même temps)

**Victoire**

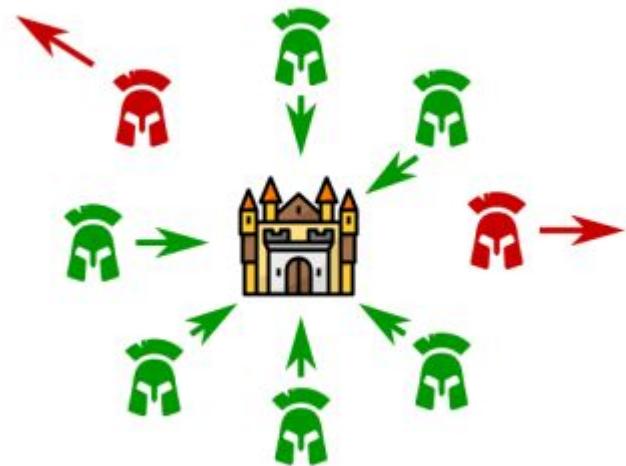


 Général loyal

 Traître

# Problèmes des généraux byzantins

**Victoire**

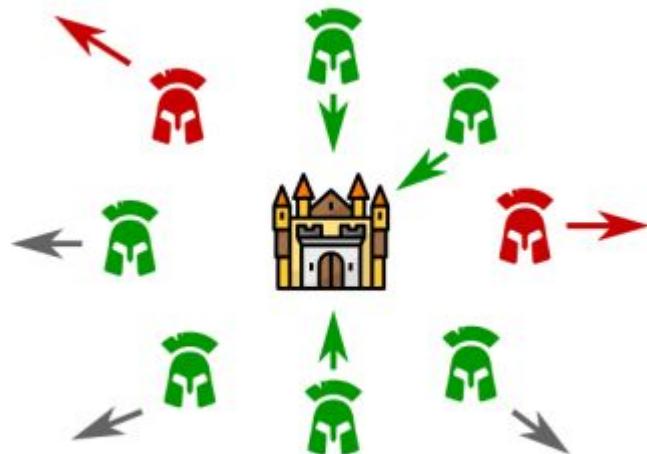


Général loyal



Traître

**Défaite**

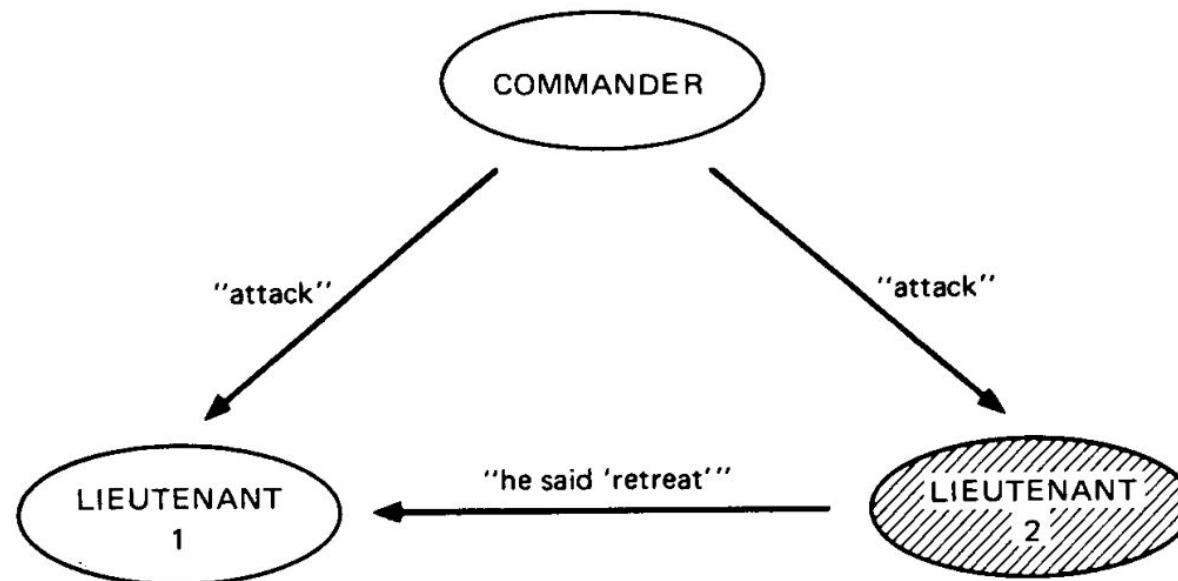


Général loyal

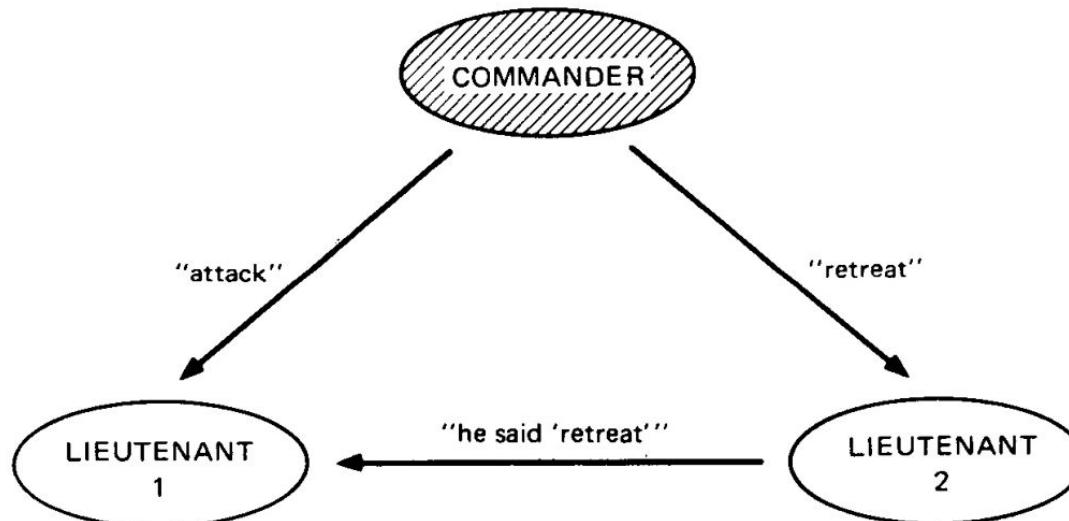


Traître

# Problèmes des généraux byzantins



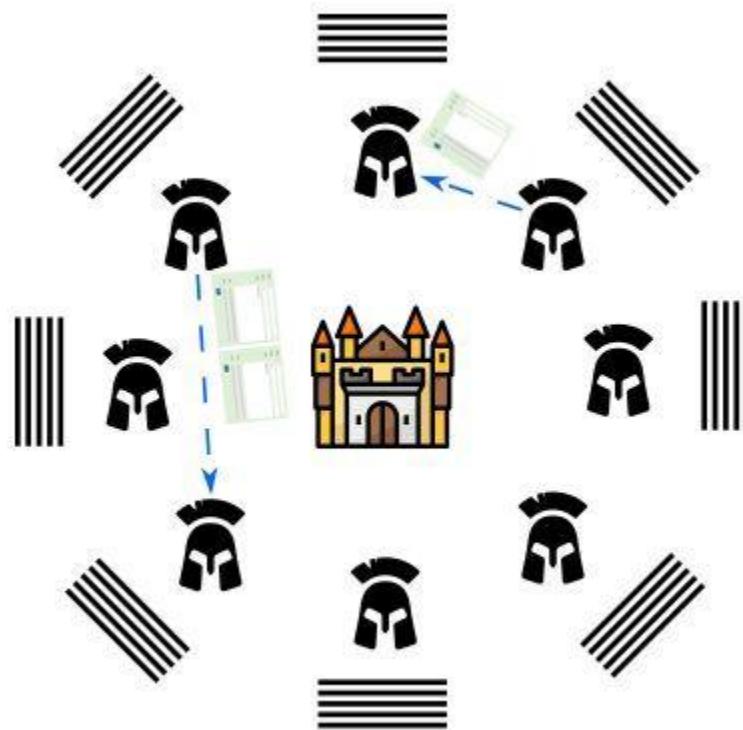
# Problèmes des généraux byzantins



# Problèmes des généraux byzantins

Block:	#	2
Nonce:	22810	
Data:	Je suis un général et je vais attaquer Lundi	
Prev:	000099fbb7bdcf37249d096a0b917f61c319aecb5114762d03fet	
Hash:	0000d2a84e2fd428ad08163bf98c53a09f0bd579dc9237ffd93bc	
Mine		

# Problèmes des généraux byzantins



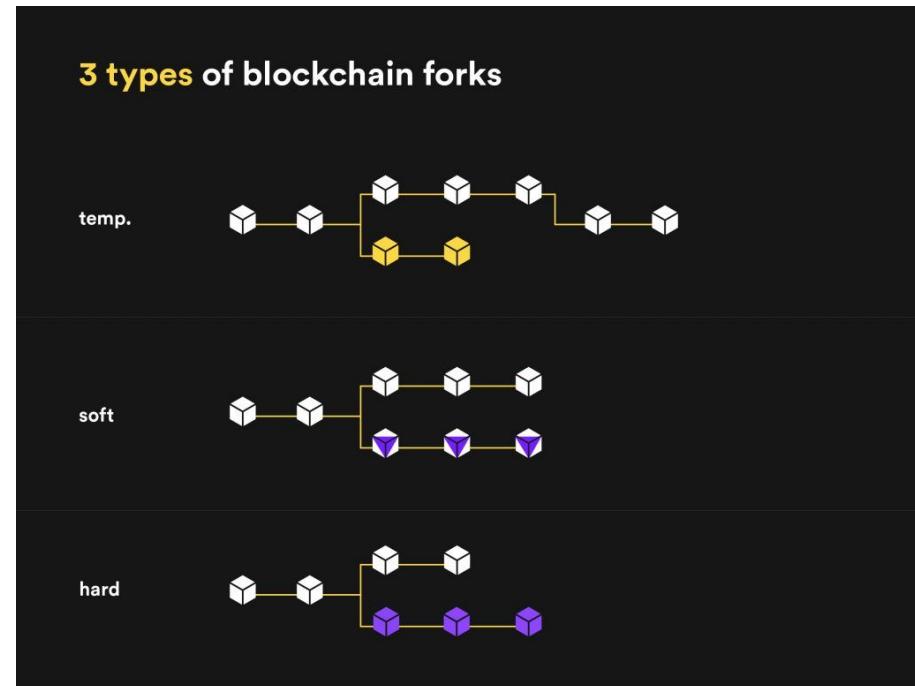
# Théorie des jeux

La théorie des jeux regroupe plusieurs dispositifs qui permettent de déterminer ce qu'un acteur, peut avoir comme influence sur un autre afin de retirer le meilleur de chaque situation

# Consensus, comment choisir un bloc valide

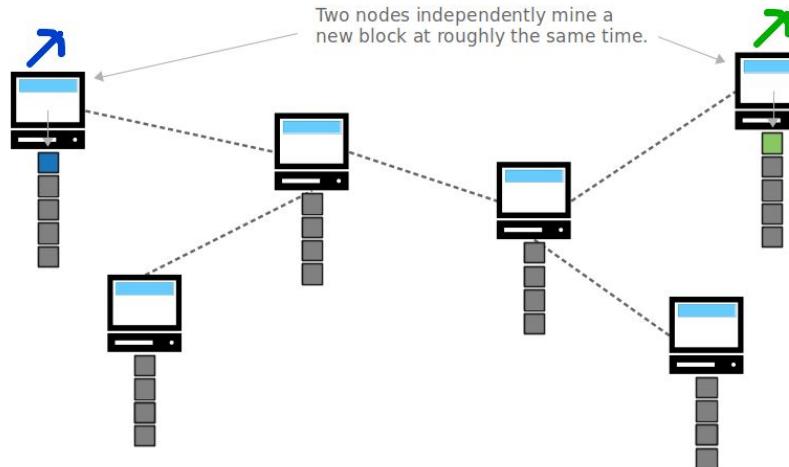
Le choix de la chaîne valide doit suivre deux composantes:

- Les règles (taille du bloc, transactions) doivent être valides (exclusif)
- On choisit toujours la chaîne la plus longue (non exclusif)



# Consensus, réorganisation de blocs

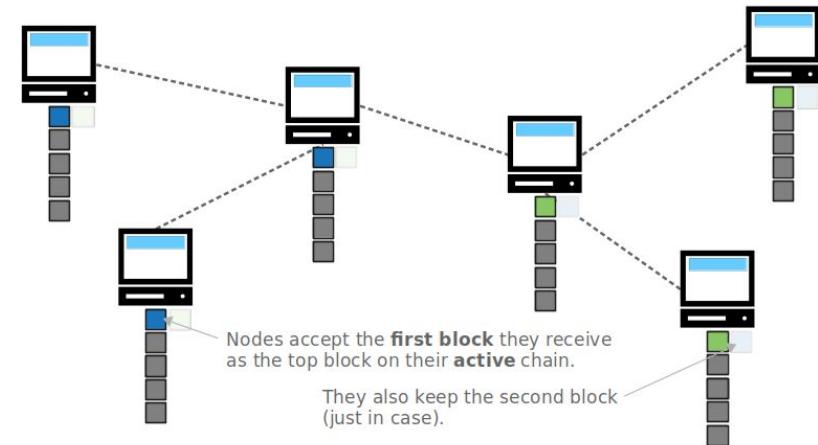
Que se passe-t-il lorsque deux blocs sont créés au même moment par deux mineurs différents ?



# Consensus, réorganisation de blocs

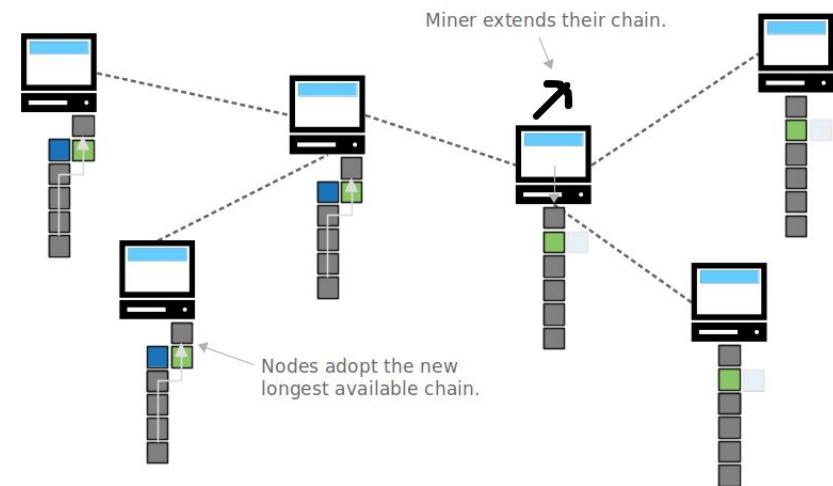
Dans un réseau distribué, chaque node ne connaît que l'état de ses voisins, il n'a pas forcément connaissance des autres blocs.

Si deux (ou plus) blocs sont vus au même moment, les deux règles sont respectées mais on ne sait pas qui choisir



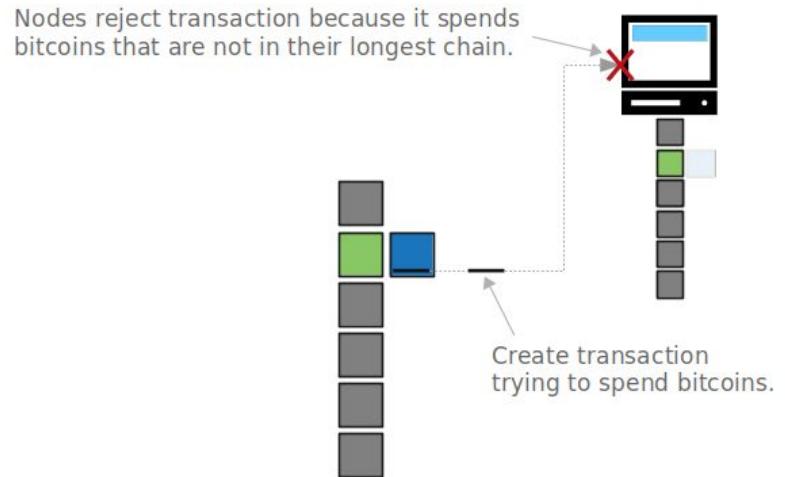
# Consensus, réorganisation de blocs

Les mineurs vont choisir un des deux blocs (généralement le premier vu) et essayer de construire un autre bloc par dessus



# Consensus, réorganisation de blocs

Les mineurs vont choisir un des deux blocs (généralement le premier vu) et essayer de construire un autre bloc par dessus



Block: # 2

Nonce: 178043

Tx:

\$ 100	From: Ripley	->	Lambert
\$	From:	->	

Prev: 0000c52990ee86de55ec4b9b32beef745d71675dc0eddfbc7b88336e2e296b

Hash: 0000d24136989b4a4f5fedb8622f9a760b7cc7514854d6ab755250cc2169e637

**Mine**

Block: # 3

Nonce: 24926

Tx:

\$ 10.00	From: Ripley	->	Jackson
\$	From:	->	
\$	From:	->	

Prev: 0000d24136989b4a4f5fedb8622f9a760b7cc7514854d6ab755250cc2169e637

Hash: 0000fa001da10dd8a621ba5ef5cb9f59d8d29908ae8c597bc483cb0bfeeb1c8c

**Mine**

Block: # 2

Nonce: 96771

Tx:

\$ 1	From: Ripley	->	Lambert
\$	From:	->	

Prev: 0000c52990ee86de55ec4b9b32beef745d71675dc0eddfbc7b88336e2e296b

Hash: 0000b8dbce1a8887dce2be02316963d7aa33ebb1ebf7accc55083d7b3073c39

**Mine**

Block: # 3

Nonce: 13804

Tx:

\$ 100	From: Ripley	->	Jackson
\$	From:	->	
\$	From:	->	

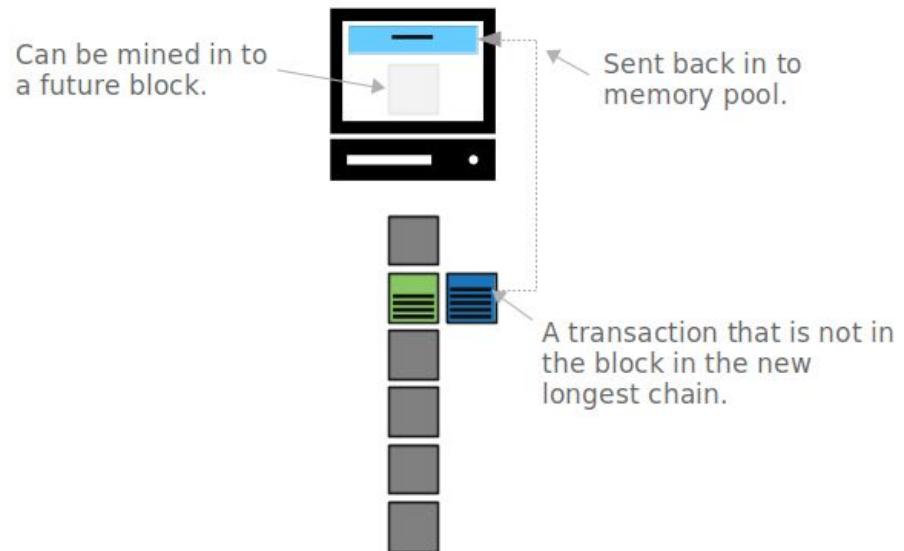
Prev: 0000b8dbce1a8887dce2be02316963d7aa33ebb1ebf7accc55083d7b3073c39

Hash: 0a0a6159ec@1ae1b96b326f7735a6ad677d38252d45bca74441257e0608dd35

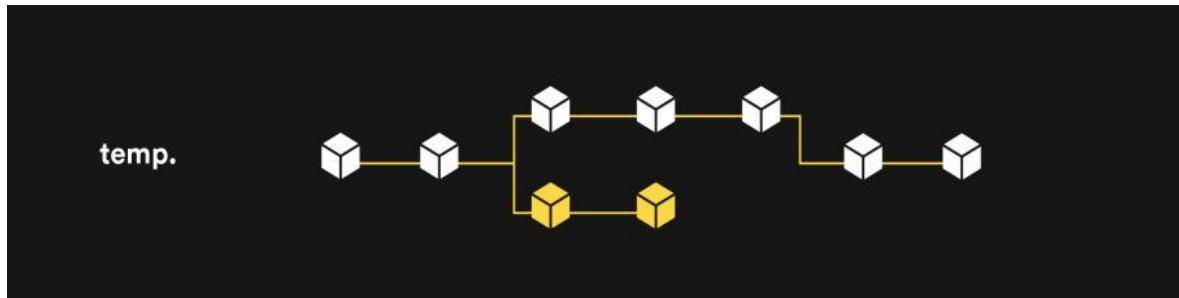
**Mine**

# Consensus, réorganisation de blocs

Lors d'une réorganisation, les transactions du bloc orphelin sont réinsérées dans la "mempool" et peuvent être minées dans les blocs suivants



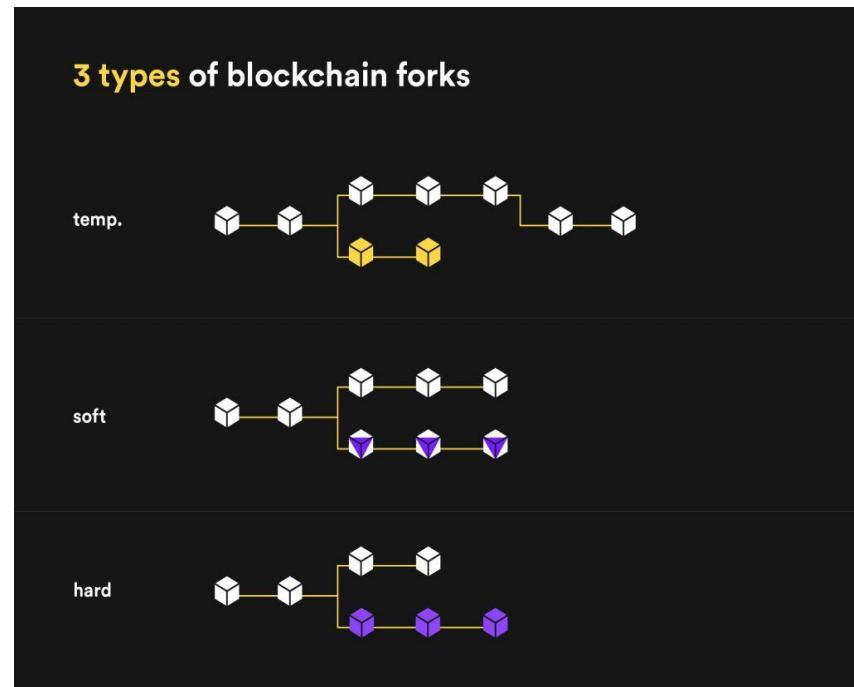
# Consensus, comment choisir un bloc valide



Les autres noeuds du réseau ont tout intérêt à choisir le chaîne la plus longue, ils vont donc choisir la chaine du haut et abandonner la chaine du bas qui deviendra alors **orpheline**

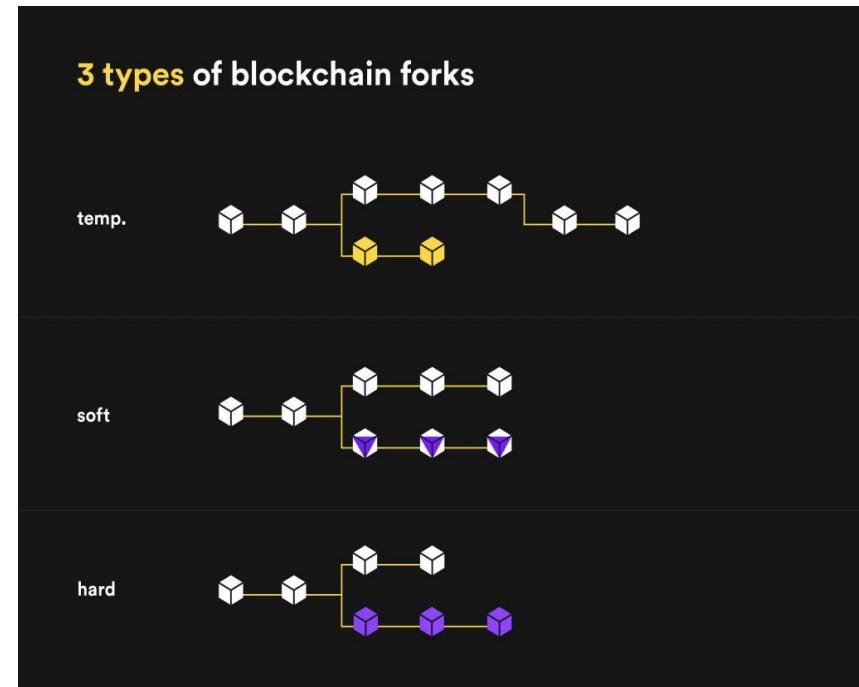
# Consensus, comment choisir un bloc valide

Un **hard fork** est un embranchement de la chaîne de blocs causé par une divergence des règles de consensus.



# Consensus, comment choisir un bloc valide

Un **soft fork** est une modification rétrocompatible (ou post compatible à proprement parler) des règles de consensus, dans le sens où les nœuds suivant les anciennes règles continuent de voir les blocs produits comme valides et restent donc connectés au réseau. Il s'agit essentiellement d'une restriction du protocole : des transactions et des blocs anciennement valides deviennent invalides. Un illustration typique de soft fork est la réduction de la taille des blocs (de 1 Mo à 300 ko par exemple) : les anciens nœuds voient les petits blocs comme valides alors même que la règle qu'ils appliquent (limite à 1 Mo) est plus large.



# Chiffrement clé publique/privée et signatures

<https://blockchaindemo2.westphal.fr>

Public / Private Key Pairs

Private Key

32051498339452478494859112526972400482152160531055882381917805645804794210600

Random

Public Key

043714e8beb71bcc24854561198a6723d7716e756ce600e906ef3d88ef0892ca9f4b3c3d5db0cbc945133c8f0b7a3372fd06097dea88{

Source: <https://github.com/anders94/public-private-key-demo>

# Chiffrement clé publique/privée et signatures

**Clé privée:** Très grand nombre aléatoire souvent représenté sous forme de caractères. À garder secret

**Clé publique:** Nombre dérivé de la clé publique, peut être partagé publiquement

**Signature:** Chaîne de caractères qui, joint à une clé publique, permet d'identifier un message

# Partie 2

## Plan

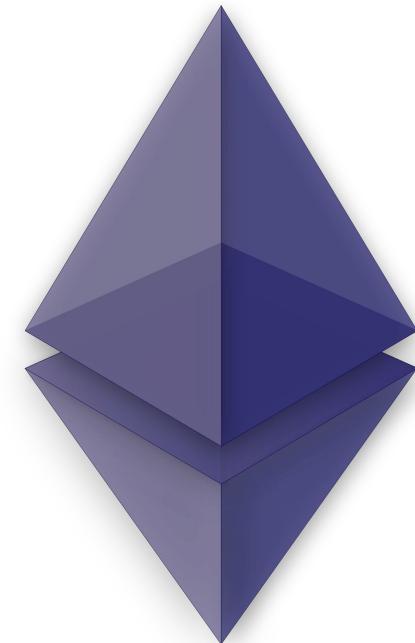
- Revue de l'histoire d'Ethereum
- Présentation de l'EVM
- Développement de votre propre token
- Déploiement du token sur le testnet BSC

# Histoire: Création et lancement d'Ethereum

En décembre 2013, **Vitalik Buterin (19 ans)**

publie une description de son projet Ethereum sur le forum BitcoinTalk sous la forme d'un livre blanc dans le but de lancer des applications décentralisées.

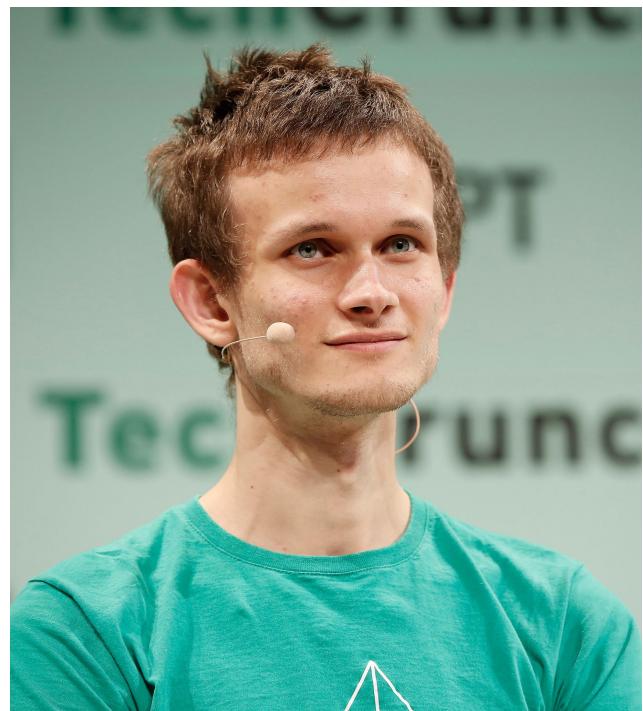
Début 2014, il met en vente les premiers Ethers pour financer le développement du projet.



# Histoire: Cr eation et lancement d'Ethereum

La vente lui permet de rassembler 31 591 bitcoins d'une valeur de plus de 18 millions de dollars  l' poque, pour 60 millions d'Ethers vendus.

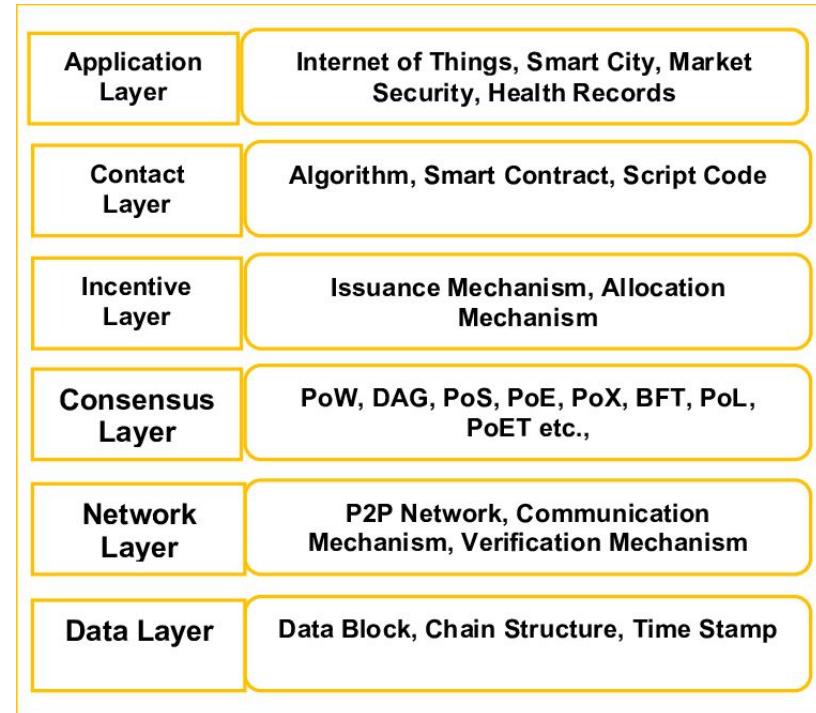
Avec cet argent il fonde Ethereum Switzerland GmbH (d veloppement de la cha ne) et The Ethereum Foundation (association  but non lucratif) pour promouvoir le d veloppement de cette nouvelle monnaie



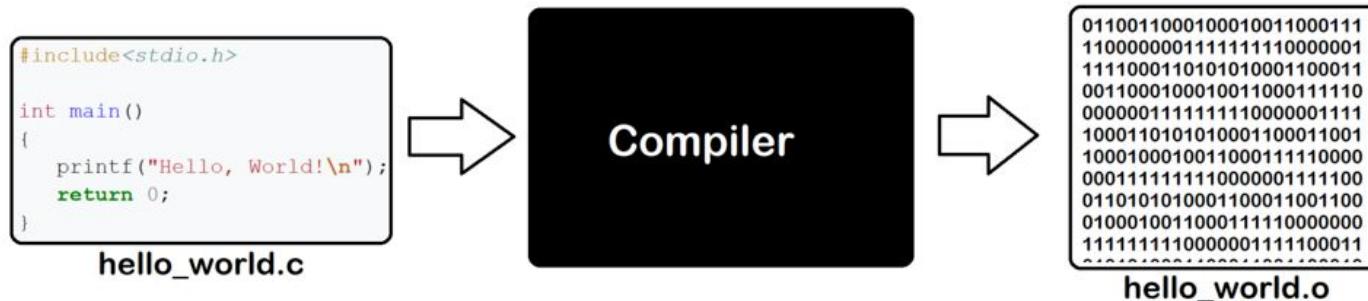
# Histoire: Création et lancement d'Ethereum

Bitcoin: Utilise uniquement les quatres premières couches

Ethereum: rajoute deux couches applicatives



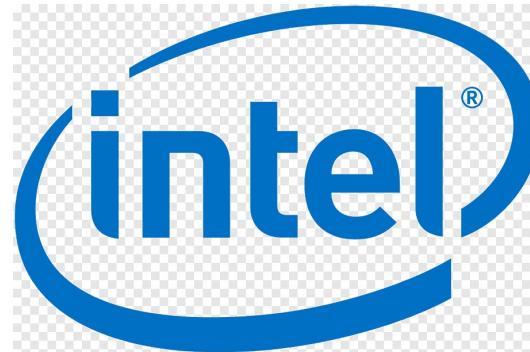
# EVM: qu'est-ce qu'une machine virtuelle ?



Un compilateur est un programme informatique qui traduit le code source écrit dans un langage de programmation en une forme lisible par la machine et pouvant être exécutée par un ordinateur (**assembleur**).

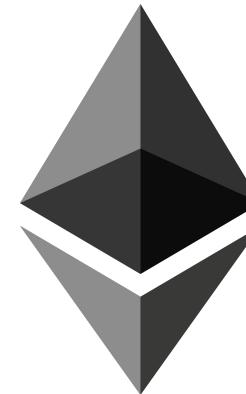
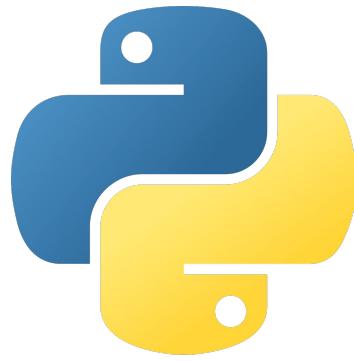
# Différents types d'assembleur: natif

ARM



L'assembleur natif est destiné à être lu par un processeur physique.

# Différents types d'assembleur: bytecode



Le bytecode est un code intermédiaire entre les instructions machines et le code source, qui n'est pas directement exécutable.

# EVM: comment faire ?

Créer une machine virtuelle décentralisée pose plusieurs problèmes:

- Où sont stockés les programmes ?
- Qui exécute les programmes ?
- Comment éviter les abus ? (ex: trop de calculs/stockage)
- Comment être sûr que les programmes soient déterministes ? (ex: Jeux de hasard)

# EVM: comment créer un contrat ?

Un programme stocké sur une blockchain s'appelle un **smart contract**

Pour créer un contrat, une transaction est émise contenant tout le code stocké. Une fois validée, le code est **immutable**



② Transaction Hash: 0x9b28f893c322350f9425bfaf6841dce7d1e50c45676cba78cc2a7b7fde3f2405 [🔗](#)

② Status: ✓ Success

② Block: ✓ 16363398 423 Block Confirmations

② Timestamp: ① 1 hr 25 mins ago (Jan-08-2023 04:55:35 PM +UTC) | ① Confirmed within 4 secs

② Sponsored:

② From:  [yannickcrypto.eth](#) [🔗](#)

② To: [Contract [0x2843dd740e1f2e0662fcde14e7203485da34f4ad](#) Created] ✓ [🔗](#)  
└ TRANSFER 0.000000000001 Ether From [0x2843dd740e1f2e0662fcde14...](#) To → [0xfc69f5418d69b5699115e516...](#)

② Value: 0.000000000001 Ether (< \$0.000001)

② Transaction Fee: 0.00945949925761497 Ether (\$11.97)

② Gas Price: 0.000000017015630129 Ether (17.015630129 Gwei)

# EVM: qui exécute les contrats ?

- Les appels aux contrats (exécution) sont effectués une première fois par le mineur/validateur
- Pour vérifier un bloc, les autres noeuds du réseau ré-exécutent la transaction (l'appel au contrat) dans leur propre EVM et vérifient si le résultat est identique
- Le résultat (et les modifications) ne sont pas stockés directement dans les blocs, uniquement les transactions

## Block:

# 1

1

## Nonce:

139358

## Block:

# 2

8

**Tx:**

\$	25.00	From:	Darcy	->	Bingley
\$	4.27	From:	Elizabe	->	Jane
\$	19.22	From:	Wickham	->	Lydia
\$	106.44	From:	Lady Ca	->	Collins
\$	6.42	From:	Charlot	->	Elizabeth

**Tx:**

\$	97.67	From:	Ripley	->	Lambert
\$	48.61	From:	Kane	->	Ash
\$	6.15	From:	Parker	->	Dallas
\$	10.44	From:	Hicks	->	Newt
\$	88.32	From:	Bishop	->	Burke
\$	45.00	From:	Hudson	->	Gorman
\$	92.00	From:	Vasquez	->	Apone

Prev:

Prev:

00000c52990ee86de55ec4b9b32beef745d71675dc

## Hash:

00000c52990ee86de55ec4b9b32beefd745d71675dc

## Hash:

000078be183417844c14a9251ca246fb15df107401c

Mine

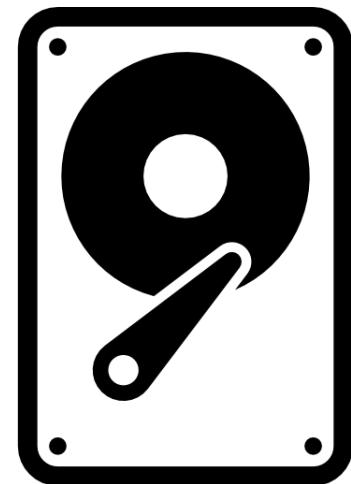
Mine

# EVM: Qu'est ce que le “state” ?

Bloc 0	Bloc 1		Bloc 2	
Coinbase:	Alice	Alice -> Martin	10	Bob -> Alex
		Coinbase	Bob	Coinbase
State		State		State
Alice	25	Alice	15	Bob
		Martin	10	Alex
		Bob	25	Alice
				Thomas

# EVM: Qu'est ce que le “state” ?

- Le state est comme le disque dur de l'EVM
- Il stocke:
  - La balance de chaque adresse (contrat ou EOA)
  - Le code d'exécution des smart contracts
  - Les variables modifiées par les contrats



# EVM: Comment éviter les abus ?

Que se passe-t-il si quelqu'un s'amuse à:

- stocker tout son disque dur dans le state ?
- l'utiliser pour calculer tout et n'importe quoi ?

Solution: faire payer chaque opération de calcul ou de stockage via le "gas" (ou gaz)



# EVM: Le gaz

Sur Ethereum (et les chaînes compatibles EVM), chaque transaction est associée à une "limite de gaz" (gas limit)

Le gaz est consommé au fur et à mesure que la transaction effectue des calculs ou des opérations de stockage



# EVM: Le gas

Pour éviter les abus, chaque transaction doit payer le gas consommé

② Transaction Fee:	0.0012629 BNB (\$0.34)
② Gas Limit:	277,044
② Gas Used by Transaction:	252,580 (91.17%)
② Gas Price:	0.000000005 BNB (5 Gwei)

Transaction Fee = Gas Price \* Gas used by Transaction

# EVM: Le gas

Que se passe-t-il si une transaction n'a pas assez de gas ?

② Transaction Fee:	0.00011 BNB (\$0.03)
② Gas Limit:	22,000
② Gas Used by Transaction:	22,000 (100%)
② Gas Price:	0.000000005 BNB (5 Gwei)
② Status:	<span style="color: red;">✖ Fail</span>

# EVM: Le revert

Lorsqu'une transaction "revert", toutes les modifications du state par la transaction sont annulées

Seul les frais de transactions sont effectivement payés

ⓘ Transaction Fee:	0.00011 BNB (\$0.03)
ⓘ Gas Limit:	22,000
ⓘ Gas Used by Transaction:	22,000 (100%)
ⓘ Gas Price:	0.000000005 BNB (5 Gwei)
ⓘ Status:	<span style="color: red;">✖ Fail</span>

# Projet: programmation de smart contracts

Étapes du projet:

- Programmation d'un premier smart contract simple
- Création d'un ERC-20 (token)
- Déploiement d'un token sur le testnet BSC
- Ajout du token sur Uniswap
- ?

# src/Counter.sol

```
1 // SPDX-License-Identifier: UNLICENSED
2 pragma solidity ^0.8.13;
3
4 contract Counter {
5     uint256 public number;
6
7     function setNumber(uint256 newNumber) public {
8         number = newNumber;
9     }
10
11    function increment() public {
12        number++;
13    }
14 }
```

# test/Counter.t.sol

```
1 /* SPDX-License-Identifier: UNLICENSED
2 pragma solidity ^0.8.13;
3
4 import "forge-std/Test.sol";
5 import "../src/Counter.sol";
6
7 contract CounterTest is Test {
8     Counter public counter;
9
10    function setUp() public {
11        counter = new Counter();
12        counter.setNumber(0);
13    }
14
15    function testIncrement() public {
16        counter.increment();
17        assertEq(counter.number(), 1);
18    }
19
20    function testSetNumber(uint256 x) public {
21        counter.setNumber(x);
22        assertEq(counter.number(), x);
23    }
24 }
```

<https://github.com/wesraph/td-blockchain-esgi>

# Le standard ERC-20

Un token est un sous ensemble des crypto monnaies qui vit directement sur une chaîne

Chaque token doit respecter la norme ERC-20 définie par les standards Ethereum.

La norme est définie au lien suivant:

<https://eips.ethereum.org/EIPS/eip-20>



# Le standard ERC-20

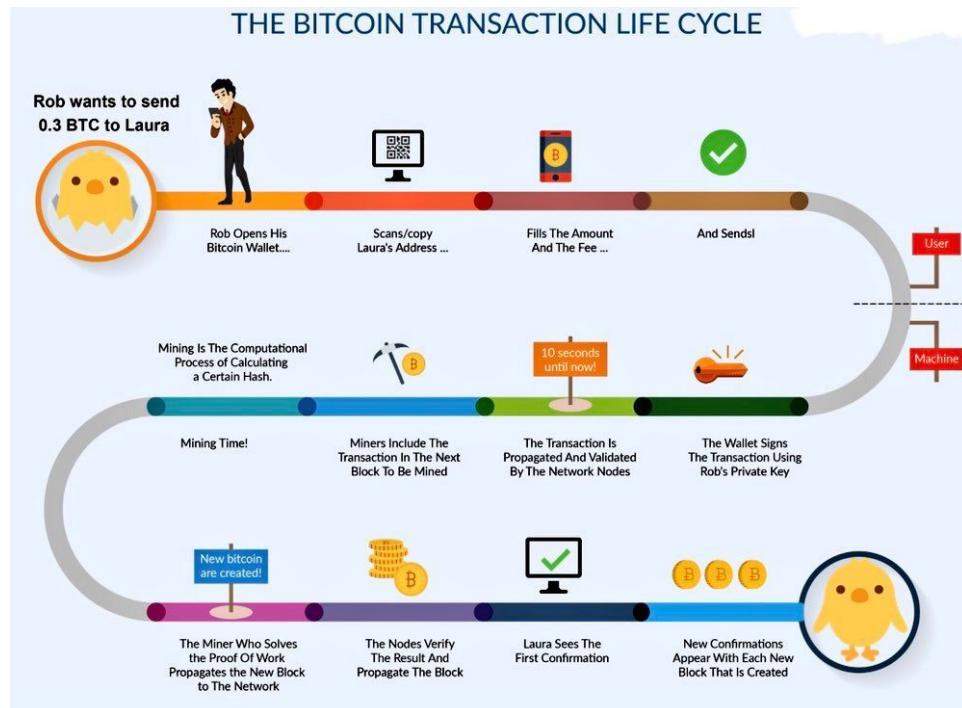
Le standard ERC-20 demande à ce que le contrat implémente **au moins** les fonctions suivantes:

```
1 function name() public view returns (string)
2 function symbol() public view returns (string)
3 function decimals() public view returns (uint8)
4 function totalSupply() public view returns (uint256)
5 function balanceOf(address owner) public view returns (uint256 balance)
6 function transfer(address _to, uint256 _value) public returns (bool success)
7 function transferFrom(address _from, address _to, uint256 _value) public returns (bool success)
8 function approve(address _spender, uint256 _value) public returns (bool success)
9 function allowance(address owner, address spender) public view returns (uint256 remaining)
```

# Plan

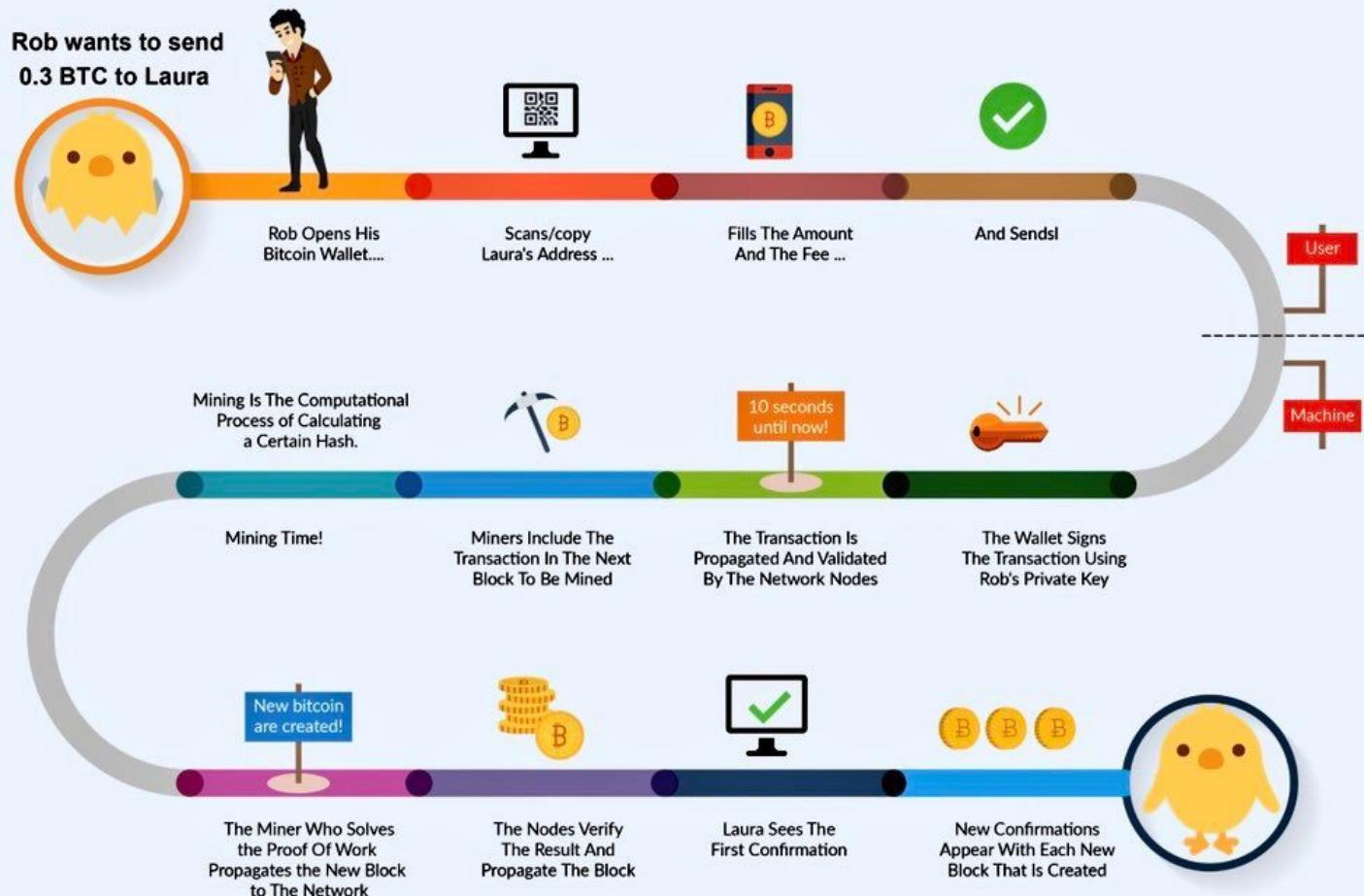
- Cycle de vie d'une transaction
- Exemple de projets utilisant la cryptomonnaie/blockchain
- TD/Projet

# Cycle de vie d'une transaction



Source: BTCTime

# THE BITCOIN TRANSACTION LIFE CYCLE



## Transaction

Sign

Verify

### Message

\$ 20.00

From:

04f4dd3198959c6b163ad6f9ddd16€

->

04cc955bf8e359cc7ebbb66f4c2dc€

### Private Key

107710091672706254366077546328575893100830454061966065764127839545094475120341

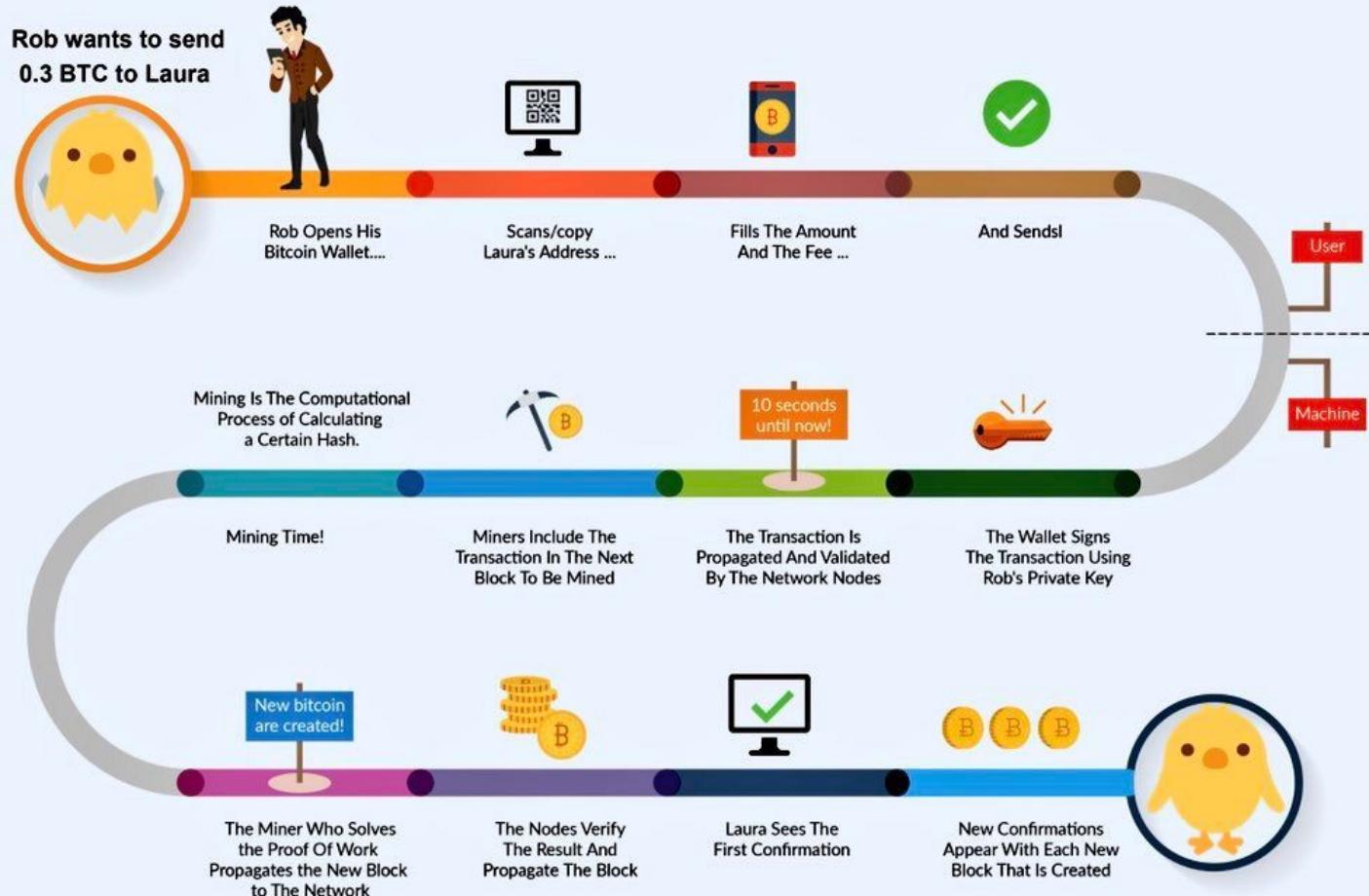


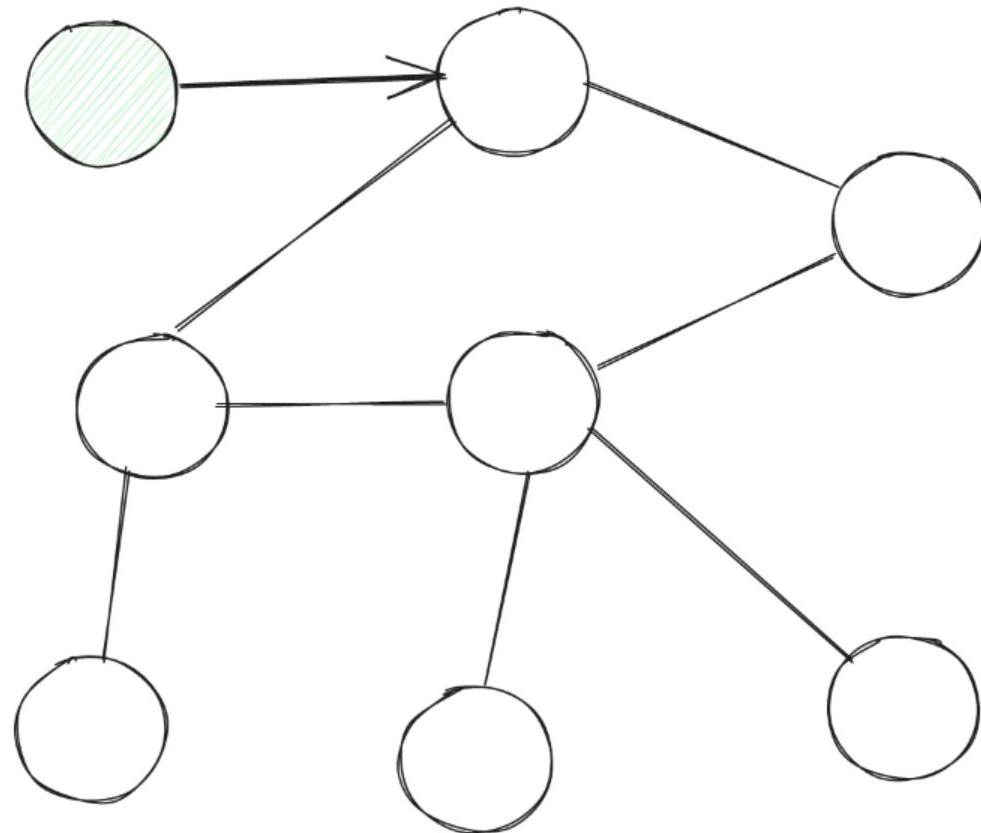
Sign

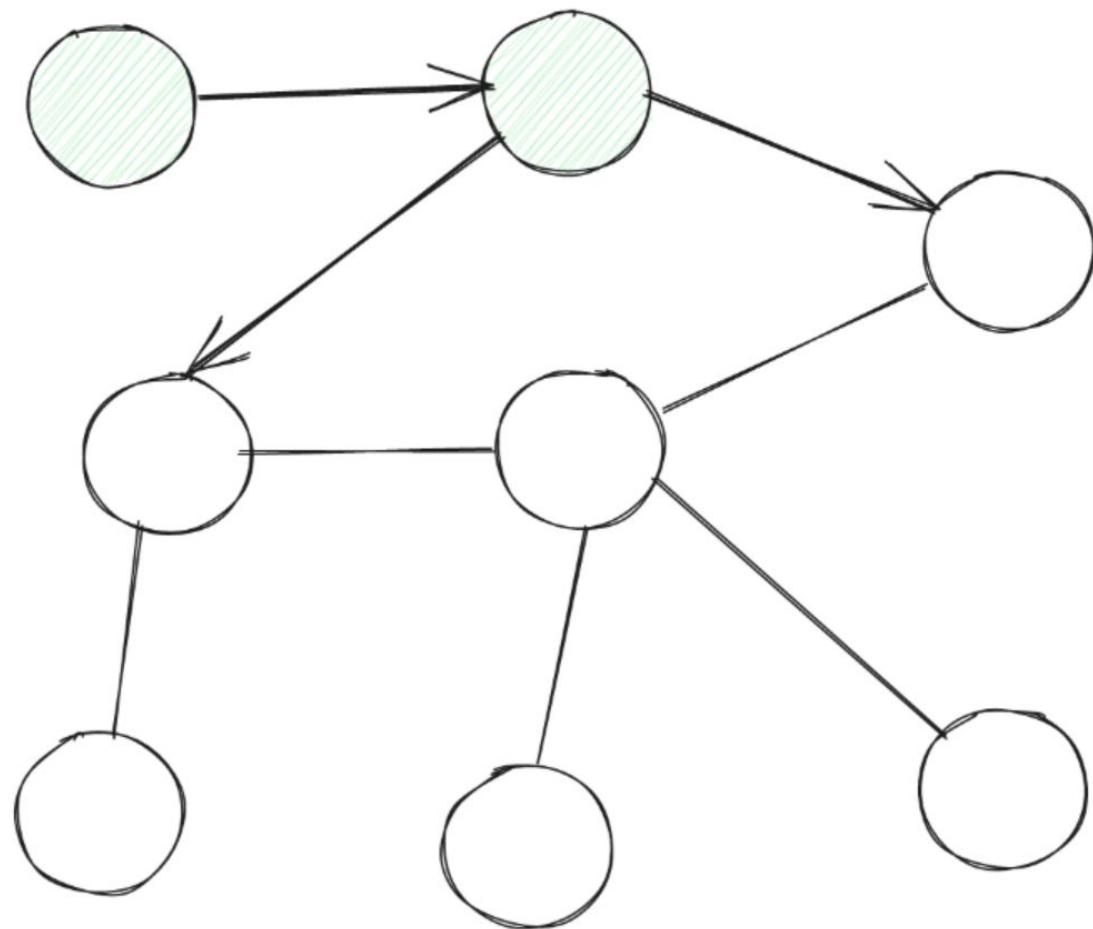
### Message Signature

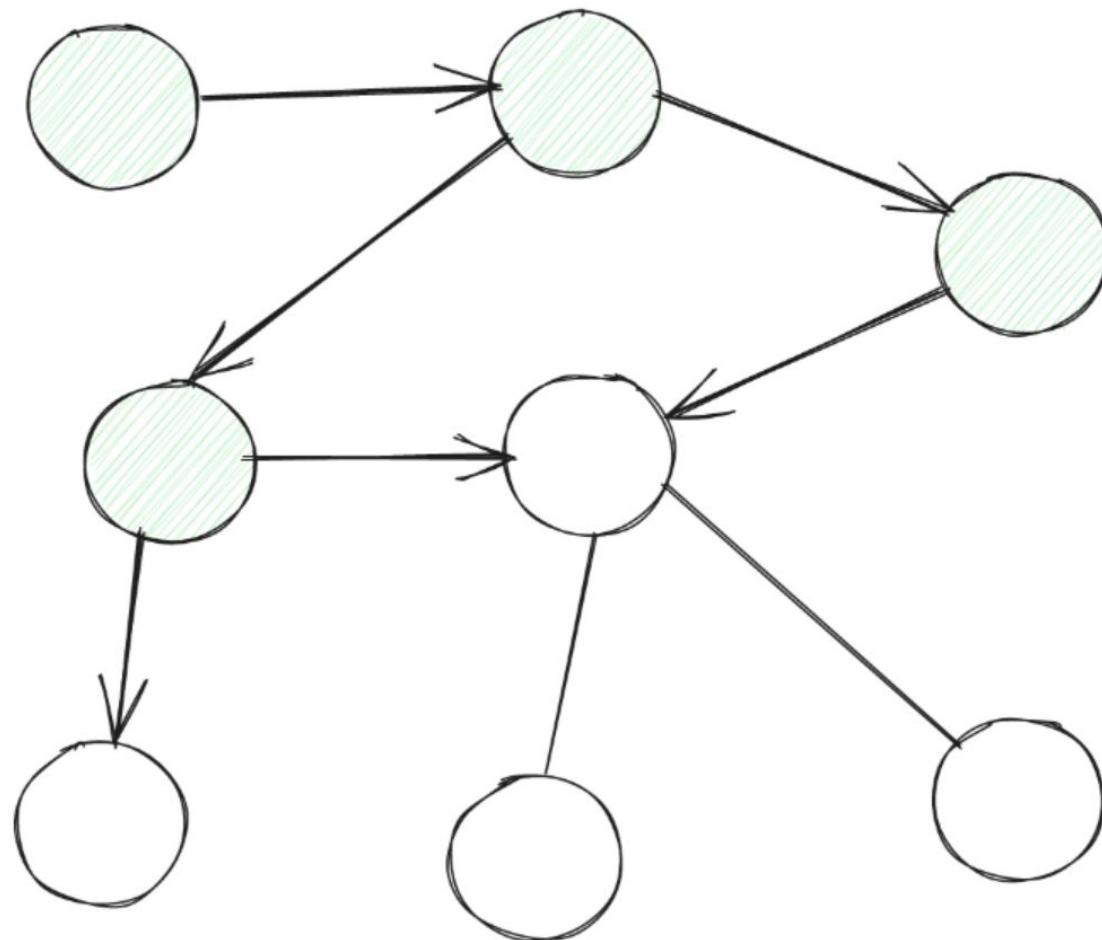
3046022100950019095c3cbb4d103089469ac14ab65dd0698191cb9852ccfdcb4d8c115e90022100b9dce46fa07050021fd8b92195cb€

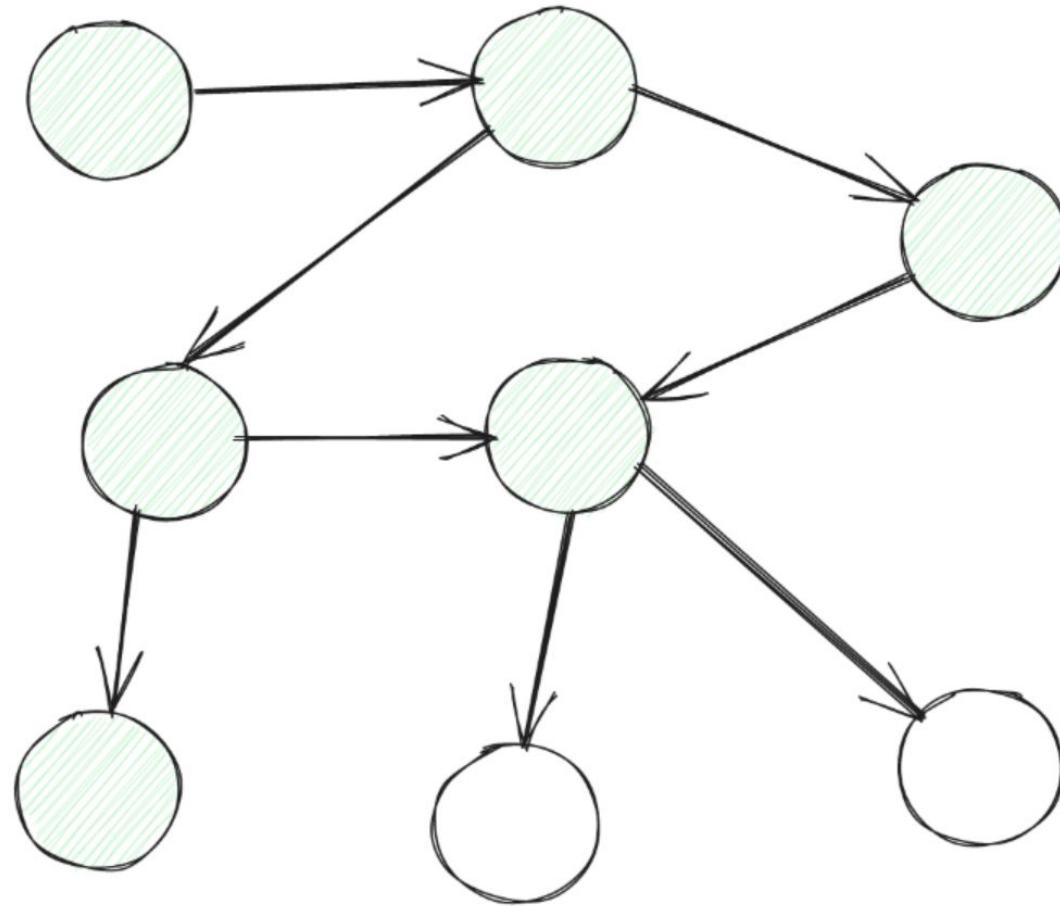
# THE BITCOIN TRANSACTION LIFE CYCLE

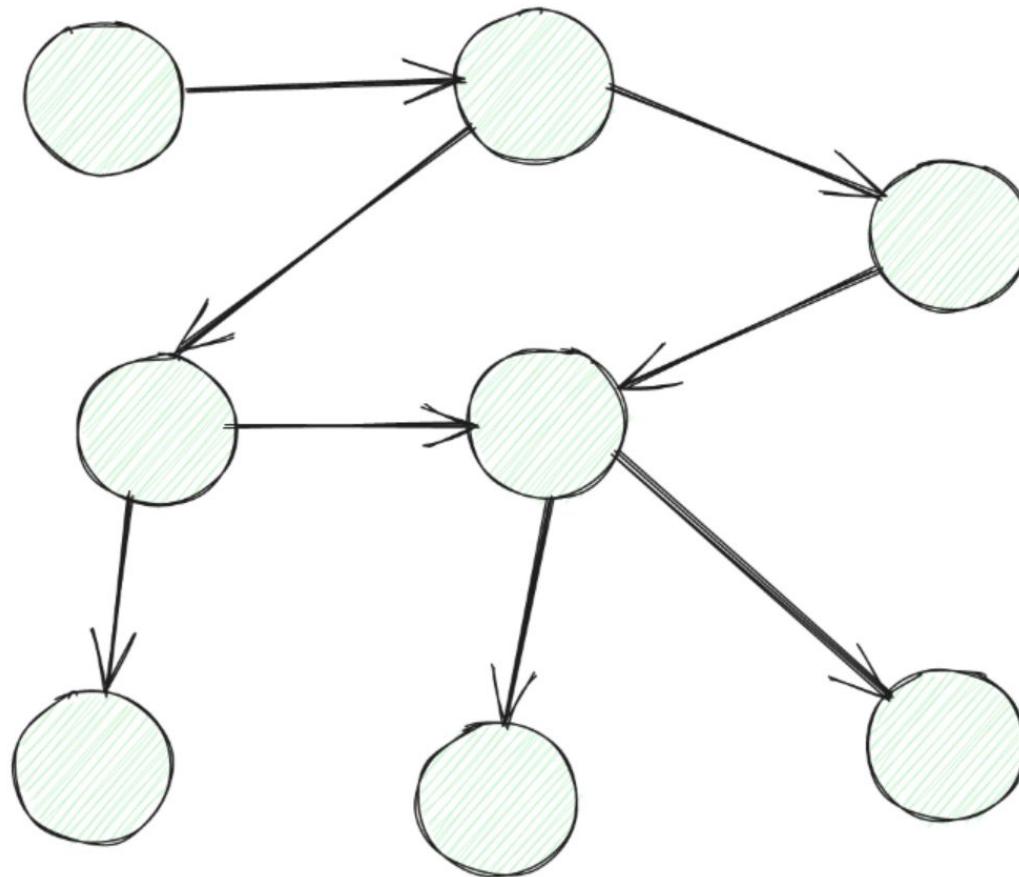




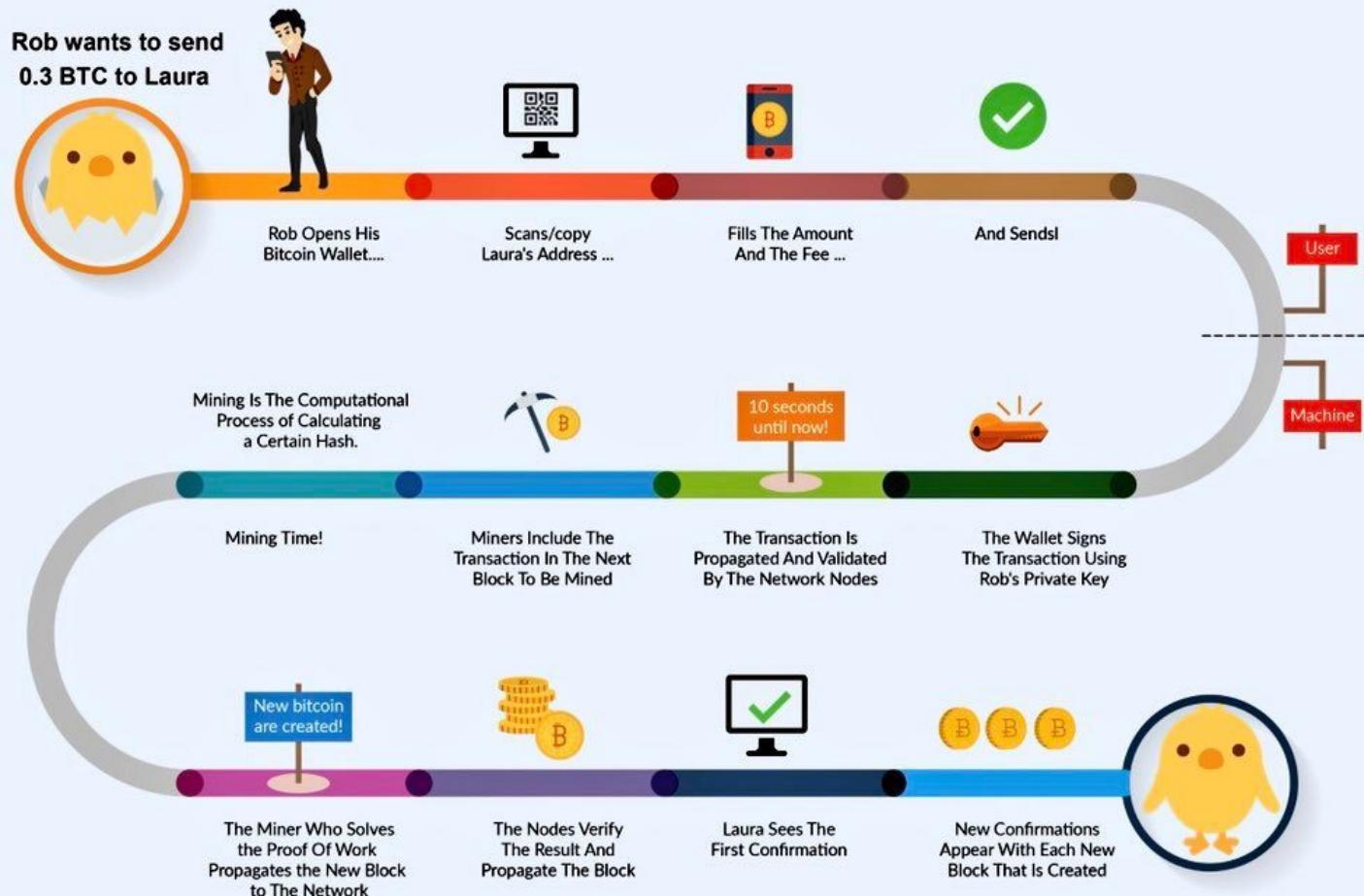








# THE BITCOIN TRANSACTION LIFE CYCLE



## Block:

# 1

1

## Nonce:

139358

## Block:

# 2

5

**Tx:**

\$	25.00	From:	Darcy	->	Bingley
\$	4.27	From:	Elizabe	->	Jane
\$	19.22	From:	Wickham	->	Lydia
\$	106.44	From:	Lady Ca	->	Collins
\$	6.42	From:	Charlot	->	Elizabeth

**Tx:**

\$	97.67	From:	Ripley	->	Lambert
\$	48.61	From:	Kane	->	Ash
\$	6.15	From:	Parker	->	Dallas
\$	10.44	From:	Hicks	->	Newt
\$	88.32	From:	Bishop	->	Burke
\$	45.00	From:	Hudson	->	Gorman
\$	92.00	From:	Vasquez	->	Apone

Prev:

Prev:

00000c52990ee86de55ec4b9b32beef745d71675dc

## Hash:

00000c52990ee86de55ec4b9b32beef745d71675dc

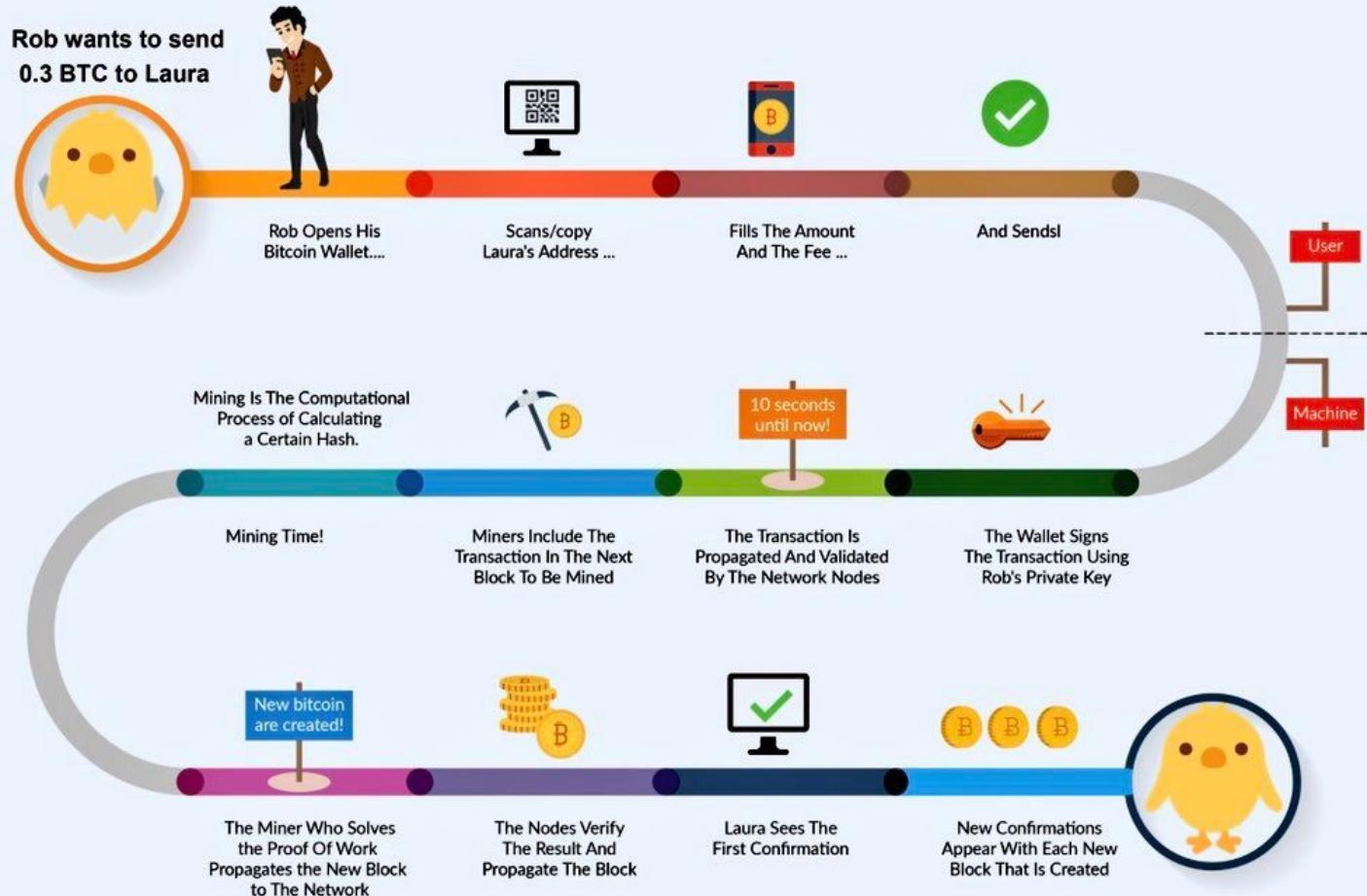
## Hash:

000078be183417844c14a9251ca246fb15df107401c

## Mine

Mine

# THE BITCOIN TRANSACTION LIFE CYCLE



# Exemple d'applications de la blockchain



Bitcoin



Ethereum



Cardano



Binance



Tether



Uniswap



Polkadot



Dogecoin



XRP



Chainlink

# Cas 1: Détenir de l'argent liquide virtuel

- La crypto monnaie doit être considérée comme de l'argent physique qui peut s'échanger virtuellement
- Not your keys not your coins
- Vous en êtes seul responsable de sa sécurité
- On ne peut pas vous le retirer



# Cas 2: Détenir de l'argent liquide virtuel de manière anonyme

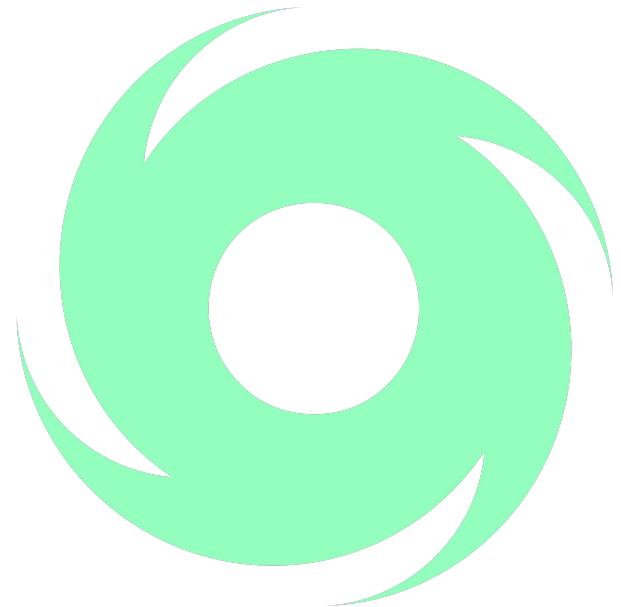
Sur la majorité des chaînes, lorsque l'on connaît l'adresse d'une personne, on peut voir l'intégralité de son solde et observer toutes les transactions qui sont effectuées.

Certaines chaînes comme Monero garantissent un anonymat complet, même en cas de divulgation de votre adresse.



# Cas 2.5: Anonymisation

Des protocoles comme Tornado cash permettent de transformer de l'argent "doxxé" en argent anonyme



# Cas 3.1: Real World Assets (RWA)

- Création d'un token lié à un objet physique. Exemple: 1 DGLD = 3.3 grammes d'or
- "Virtualisation" d'objets réels sur la blockchain
- Permet des transactions instantanées
- Utilisation "active" des assets



# Cas 3: Real World Assets (RWA)

	DGLD	Physical Gold	GOLD ETF
Allocated Ownership	◎	◎	Varies by Product
24/7 Verifiable Ownership	◎	Varies by Custodian	Impossible
Physically Redeemable	◎	◎	Varies by Product
Settlement Time	Instant	T+2 -> T+5	T+2
Swiss Custody	◎	Available	Varies by Product
Gold Quality	LBMA Good Delivery Gold	Varies by Broker	Varies by Product
Regulator	FINMA, VQF	Xne	Securities regulation

# Cas 3.2: Real World Assets (RWA)

- Tokenisation d'un contrat de location (SCPI)
- Possibilité de détenir un part dans une SCPI de manière décentralisé
- Toucher les loyers directement en cryptomonnaie
- Usage dans d'autres protocoles (composabilité)



# Cas 3.3: Stable Coins (USDT, USDC, EUROCO...)

- Cryptomonnaie conçue pour maintenir une valeur stable par rapport à un actif spécifique.
- Deux types de stable coins:
  - Algorithmique, soutenu par un système de smart contrats lui assurant une valeur proche. Exemple: DAI
  - Adossé à une devise fiduciaire. Exemple: USDC
- Fonctionnement expliqué plus tard dans le cours,  
à retenir ici: 1 Token = 1 USD



# Composabilité

Définition: La composabilité est la capacité des différents protocoles ou systèmes à interagir et à s'intégrer efficacement. Cette interaction peut permettre aux utilisateurs de tirer parti de fonctionnalités uniques et d'optimiser l'efficacité de leurs transactions ou de leurs investissements.

# Cas 4: Prêt d'argent (Aave, Compound)

Des protocoles permettent de prêter ou d'emprunter de l'argent entre utilisateurs de manière décentralisée.

Ils fonctionnent dans la majorité des cas comme des prêts sur gage (c'est à dire qu'il faut déposer une somme **supérieur** à la somme que l'on veut emprunter)



# Cas 4: Prêt d'argent (Aave, Compound)

Exemple: Je veux acheter de l'or et en même temps utiliser cet argent pour faire autre chose.

- J'achète 1000\$ d'or via le DGLD
- Je "prête" mes DGLD d'une valeur de 1000\$ contre 800\$ en USDC
- J'ai maintenant 1000\$ de DGLD ainsi que 800\$ en USDC

Pour disposer à nouveau de mes DGLD, je dois rembourser l'intégralité de mon prêt (800 USDC) ainsi que des frais.

La valeur de mes actifs prêtés doit **toujours** être supérieur à la somme empruntée

# Cas 4: Prêt d'argent (Aave, Compound)

Que se passe-t-il si le prix de l'or baisse ?

La valeur de mes actifs prêtés doit **toujours** être supérieur à la somme empruntée

Trois options s'offre à moi:

- J'augmente la valeur de mon collatéral
- Je rembourse mon prêt
- Je ne fais rien

Collatéral: un actif déposé par un emprunteur pour sécuriser un prêt. Dans ce cas là la somme équivalente en DGLD

# Cas 4: Prêt d'argent (Aave, Compound)

Que se passe-t-il, en dehors du contexte de la cryptomonnaie, lorsque l'emprunteur ne rembourse pas son prêt ?

Le prêteur sur gage vend le bien

En crypto c'est identique, lorsque la valeur du collatéral passe en dessous de la valeur de l'argent emprunté, le prêt est **liquidé**.

# Cas 4: Prêt d'argent (Aave, Compound)

## Liquidation:

Une liquidation se produit lorsque la valeur de la garantie d'un emprunteur tombe en dessous d'un certain seuil, appelé "**seuil de liquidation**". Lorsque cela se produit, une partie ou la totalité de la garantie de l'emprunteur peut être vendue automatiquement pour rembourser le prêt. Cette vente est réalisée par des liquidateurs, qui sont récompensés par une "prime de liquidation" pour leur action.

# Cas 4: Prêt d'argent (Aave, Compound)

## Exemple:

J'ai acheté du DGLD mais au lieu de laisser dormir, je souhaite le faire "travailler". Je peux le prêter contre gage à d'autres utilisateurs contre des intérêts.

### Ethereum assets

Search asset name, symbol, or address

Asset	Total supplied	Supply APY	Total borrowed	Borrow APY, variable	Borrow APY, stable	
 Balancer BAL	<b>161.11K</b> \$ 845.63K	<b>8.35 %</b>	<b>83.71K</b> \$ 439.39K	<b>21.27 %</b>	—	<button>Details</button>
 USD Coin USDC	<b>163.82M</b> \$ 163.78M	<b>2.72 %</b>	<b>143.48M</b> \$ 143.44M	<b>3.46 %</b>	—	<button>Details</button>

# Cas 4: Prêt d'argent (Aave, Compound)

## **Exemple (à ne pas faire):**

Je suis certain que le prix de l'or va augmenter durant les prochaines semaines et je souhaite en acquérir plus que ma somme de départ de 1000\$. Je peux alors:

- Déposer 1000\$ de DGLD, emprunter 800\$ en USDC
- Avec les 800\$ en USDC, acheter pour 800 \$ de DGLD
- Déposer les 800\$ de DGLD, faire un emprunt de 500\$ en USDC
- Acheter 500\$ de DGLD

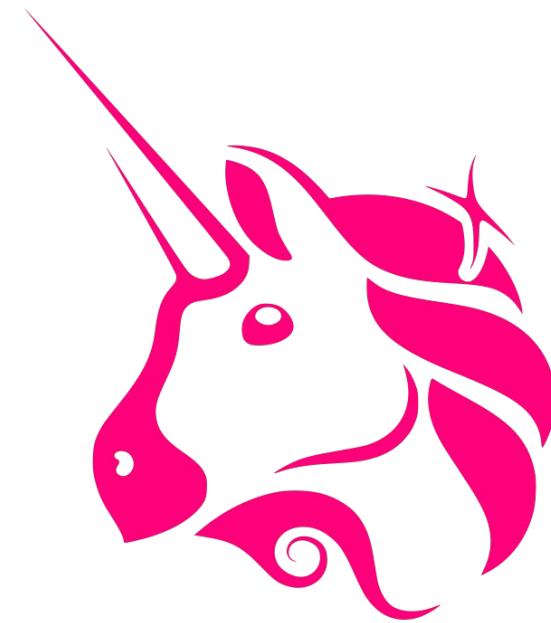
À la fin de cette opération je "possède" l'équivalent de 1000\$ + 800\$ + 500\$ d'or. Cependant, attention à la liquidation

# Cas 5: Échanges décentralisés (Uniswap)

Uniswap est un protocole open-source pour échanger des cryptomonnaies de manière décentralisée sur la blockchain Ethereum.

C'est un exemple de ce qu'on appelle un échange décentralisé (DEX), qui permet aux utilisateurs d'échanger directement des tokens entre eux, sans intermédiaire.

Uniswap se distingue par son utilisation de "pools de liquidités" plutôt que de carnets d'ordres traditionnels pour faciliter les échanges.



# Cas 5: Échanges décentralisés (Uniswap)

Les utilisateurs déposent des paires de tokens dans des pools de liquidités. Par exemple, une pool pourrait contenir des tokens ETH et DAI.

En retour, ils reçoivent des tokens de liquidité qui représentent leur part dans la pool. Ces tokens peuvent être échangés contre la part de l'utilisateur dans la pool à tout moment.

Les échanges entre tokens sont facilités par ces pools de liquidités. Par exemple, si quelqu'un veut échanger de l'ETH contre du DAI, il peut le faire en utilisant le pool de liquidités ETH/DAI.



# Cas 5: Échanges décentralisés (Uniswap)

Uniswap utilise un modèle de tarification automatique appelé "Automated Market Maker" (AMM).

Le ratio de tokens dans un pool de liquidités détermine le prix de chaque token. Par exemple, si un pool contient 50 ETH et 10 000 DAI, le prix de l'ETH est de 200 DAI.

Lorsqu'un échange est effectué, le ratio change, ce qui entraîne une modification du prix. Les frais de transaction sont ajoutés au pool de liquidités, récompensant ainsi les fournisseurs de liquidités.

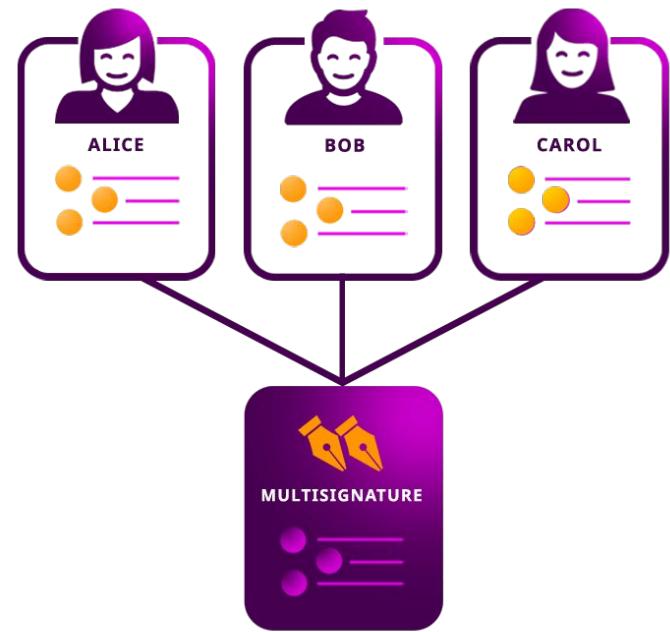


# Cas 5: Multi sig

Un wallet multi-signature (ou multi-sig) est un type de portefeuille de cryptomonnaie qui nécessite plusieurs clés privées pour signer et valider une transaction.

Le principe de base est "M sur N", où M est le nombre minimum de signatures requises pour effectuer une transaction et N est le nombre total de clés privées.

Les wallets multi-signatures offrent un niveau de sécurité supplémentaire en exigeant un consensus parmi plusieurs parties avant qu'une transaction puisse être approuvée.



2-of-3 Signature Needed

# Cas 5: Multi sig

Imaginons un wallet multi-sig configuré pour un "2 sur 3". Cela signifie que trois clés privées sont générées, mais qu'au moins deux d'entre elles sont nécessaires pour signer une transaction.

Ce type de configuration pourrait être utilisé par une entreprise pour gérer ses fonds. Par exemple, trois directeurs pourraient détenir une clé chacun. Ainsi, aucune dépense ne peut être effectuée sans l'accord d'au moins deux directeurs.

De cette manière, le wallet multi-sig augmente la sécurité et la responsabilité, tout en empêchant le vol ou la mauvaise utilisation des fonds par une seule personne.



# Cas 6: Gestion de portfolio automatique (Balancer)

Balancer est un protocole DeFi sur Ethereum offrant un échange automatique de tokens.

Il se distingue par sa capacité à créer des pools de liquidités contenant jusqu'à 8 tokens avec des pondérations personnalisables.

Il agit comme un "portefeuille indexé automatique", ajustant les proportions de tokens pour maintenir les pondérations définies.



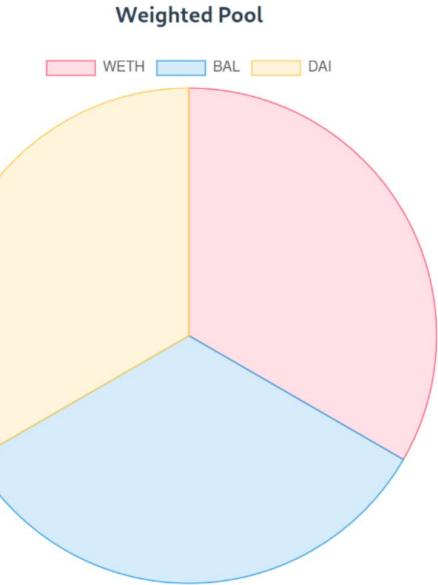
# Cas 6: Gestion de portfolio automatique (Balancer)

Exemple:

Un utilisateur souhaite disposer des trois tokens suivant: 33% de WETH, 33 % de BAL, 33% de DAI.

Le pourcentage représente **la valeur** du token, pas son montant (exemple: 33\$ de WETH, 33\$ de BAL et 33\$ de DAI)

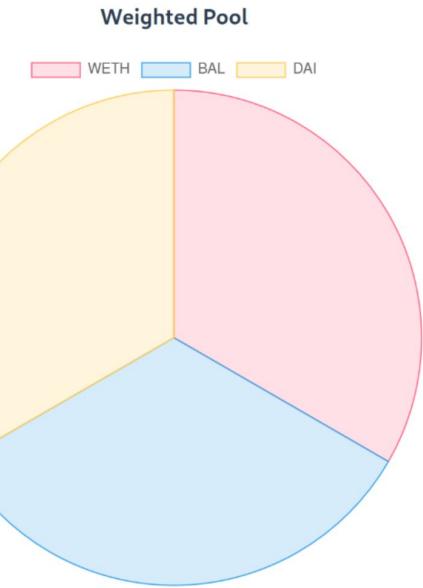
Lorsque la valeur d'un de ces assets change, le protocole permet d'échanger automatiquement les assets entre eux.



# Cas 6: Gestion de portfolio automatique (Balancer)

Exemple:

- La valeur initiale des assets



# Cas 7: NFT

Un NFT (Non-Fungible Token) est un type de token unique et non interchangeable.

Contrairement aux crypto-monnaies comme Bitcoin ou Ether, qui sont fungibles et peuvent être échangées sur une base 1:1, chaque NFT a des informations uniques qui le rendent irremplaçable.

Cela permet de prouver la propriété et l'authenticité d'un objet numérique.



# Cas 7: NFT

Les NFT ont révolutionné le monde de l'art numérique en permettant aux artistes de monétiser leurs œuvres d'une manière nouvelle et innovante.

Les œuvres d'art numériques peuvent être "tokenisées" en NFT, donnant à l'acheteur la propriété vérifiable de l'œuvre d'art originale.

Des œuvres d'art numérique ont été vendues pour des millions de dollars sous forme de NFT, mettant en évidence leur potentiel de valeur.



# Cas 7: NFT

Comme pour toute innovation, le marché des NFT est sujet aux arnaques. Il est essentiel de faire preuve de prudence et de diligence raisonnable avant d'acheter des NFT.

Certains escrocs peuvent essayer de vendre des NFT qui ne leur appartiennent pas, ou de fausses représentations d'œuvres d'art numériques connues. D'autres peuvent essayer de manipuler le marché en créant de faux buzz autour d'un NFT.

Ces arnaques peuvent nuire à la réputation des cryptomonnaies et de la blockchain en général. Il est important de rappeler que la technologie blockchain elle-même n'est pas en faute, mais plutôt l'utilisation abusive ou malhonnête de cette technologie.





# Cas 7: Provision de liquidité (Uniswap)

# Cas 9: Oracles décentralisés (Chainlink)

# Cas 10: Stockage de fichiers décentralisé (Filecoin)

# Cas 11: DAO

# Cas 12: Rémunération d'artistes (Audius)

# Cas 14: Traçabilité