

BITCOIN L'OR NUMERIQUE

Consultant Blockchain

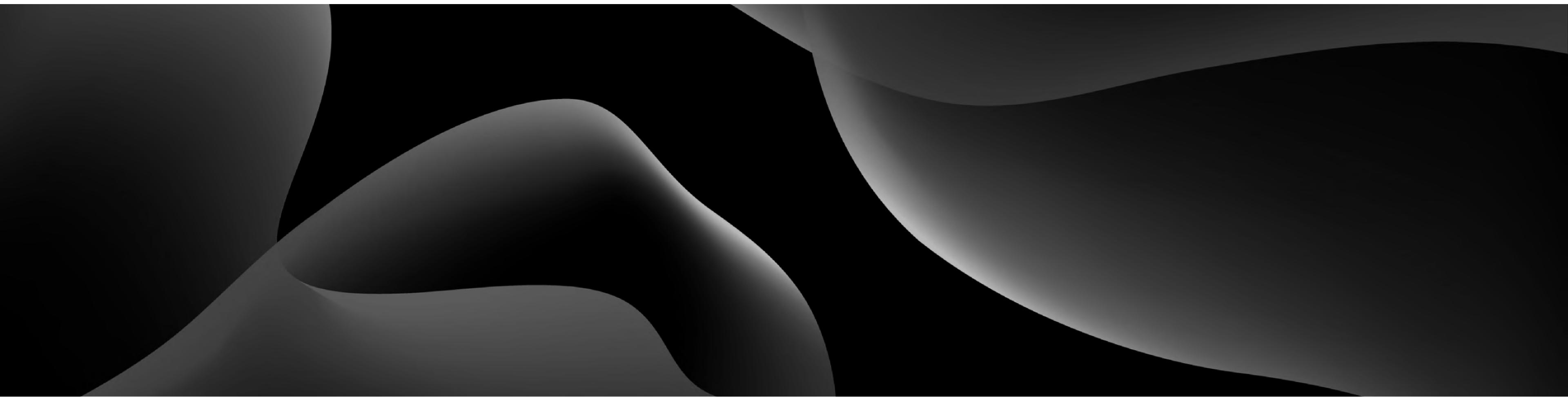
ARCHITECTURE & FONCTIONNEMENT D'UNE BLOCKCHAIN

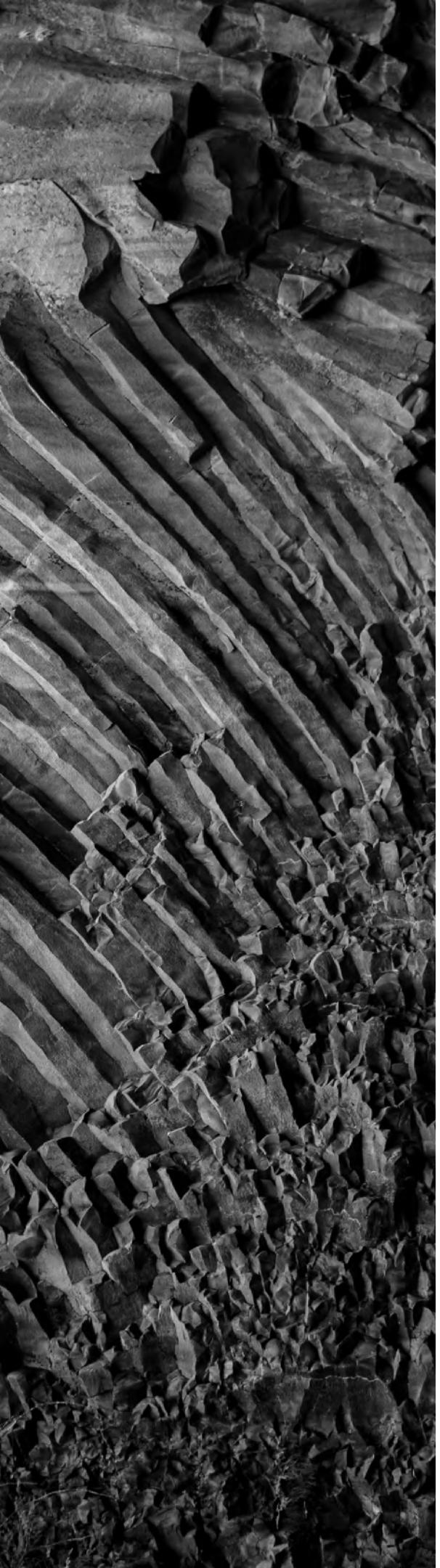
Objectifs

- Comprendre les spécificités et avantages d'un système de comptabilité triple
- Maîtriser le fonctionnement technique d'une blockchain, du hashrate à la difficulté
- Intégrer le circuit économique et le système incitatif de l'activité de minage
- Développer des outils d'expertise autour de l'architecture par blocs

01

Une révolution de la comptabilité





Comptabilité simple

- Système originel, consistant en un **simple enregistrement numérique des sorties et entrées** (ex : dépenses et recettes, pertes et profits, consommations et récoltes...),
- **Sans système compensatoire** permettant une meilleure précision, traçabilité et prévisibilité
- **Longue, fastidieuse, imprécise**, cette technique de comptabilité n'est pas adaptée aux grands ensembles de données
- Elle fut pendant plusieurs millénaires le système comptable utilisé par les empires et royaumes pour lever l'impôt, gérer leurs dépenses, ou administrer les récoltes

Comptabilité simple

Grande tablette d'inventaire
d'une institution sumérienne:
bilan annuel d'une exploitation
agricole, règne d'Amar-Sîn
(2046 à 2038 av. J.-C), musée
du Louvre



Comptabilité simple

Registre de la taille de Paris
(parchemin, 1296-1300)
conservé aux Archives
nationales

R obert le page	prie	xxv. f.	S ymon lussier	prie	xxv. f.
b aneguin de guier	prie	xxv. f.	E stiene le mareschal	prie	xxv. f.
J chame la matreschale begame	prie	xxv. f.	S erhart pointelne	prie	xxv. f.
E melme la nauere	prie	xxv. f.	O oule le becon	prie	xxv. f.
C e son neveu	prie	xxv. f.	J aquefene la dame	prie	xxv. f.
I analis la godeliche	prie	xxv. f.	D eant saint huitz vers le petit huitz	prie	xxv. f.
J chan le tenu	prie	xxv. f.	P ierre pointelne	prie	xxv. f.
L a rive des poyntoires le rive denee la rive			G erart le lev	prie	xxv. f.
de la rive			J chan hure	prie	xxv. f.
J chan pointelne	prie	xxv. f.	H elois la dame	prie	xxv. f.
E breuenot son frere	prie	xxv. f.	J a rive pierre chate prie le rive ou il demeure		
G uall le trompecur	prie	xxv. f.	J ebem lescot filz feu Goffre	prie	xxv. f.
S yndam dandim	prie	xxv. f.	O ichiel son frere	prie	xxv. f.
H uent le lamere	prie	xxv. f.	V soul de lourbon	prie	xxv. f.
O estre robert herbe	prie	xxv. f.	P ierre chagre rive	prie	xxv. f.
J aques de courbeul	prie	xxv. f.	P ierre de lome	prie	xxv. f.
L e gendre nicholas pointelne	prie	xxv. f.	P ierre lorbateur	prie	xxv. f.
N icholas pointelne	prie	xxv. f.	O estre pierre de nonsancte	prie	xxv. f.
L a buer la fume	prie	xxv. f.	L autre rive de sole rive		
O estre heude de lescole	prie	xxv. f.	V ichart poile haste	prie	xxv. f.
P ierre de maatoles	prie	xxv. f.	E enau de seulz	prie	xxv. f.
P asoil de crespi	prie	xxv. f.	G remba ala pie au lyon	prie	xxv. f.
N icholas le vogguigno	prie	xxv. f.	J chan lenglois le viel	prie	xxv. f.
L a lame feu vauant de mil	prie	xxv. f.	J ebam lghedeloue	prie	xxv. f.
G erart le chantier	prie	xxv. f.	D rive de vauveign	prie	xxv. f.
Z nicholas lenglois	prie	xxv. f.	L avocetene digne prie lamente		
J chan dieg aust	prie	xxv. f.	J chan de laudheuel	prie	xxv. f.
P ierre de maluz	prie	xxv. f.	B etian son gendre	prie	xxv. f.
T homas lenglois	prie	xxv. f.	D am le camus	prie	xxv. f.
J chan son fuiz	prie	xxv. f.	G ile daudueil	prie	xxv. f.
E melme la velue	prie	xxv. f.	L a lame feu endre daram	prie	xxv. f.
M ment le conce	prie	xxv. f.	P errot de la court neuve	prie	xxv. f.
G uill de boggi ferpier	prie	xxv. f.	J aques de laigni	prie	xxv. f.
P ierre de croif moulin	prie	xxv. f.	F inquis son fuiz	prie	xxv. f.
J aquer bouardon	prie	xxv. f.	J aques bernier son gendre	prie	xxv. f.
J chan encourt	prie	xxv. f.	J chan aie	prie	xxv. f.
A uben de s. julien	prie	xxv. f.	S et les ay suers tant liene que laute	prie	xxv. f.
G uall le petur	prie	xxv. f.	J oce le drapier	prie	xxv. f.
M oham bouardon	prie	xxv. f.	S ilebret dargenteul	prie	xxv. f.
P ichart de vin	prie	xxv. f.	N nicholas le grante	prie	xxv. f.
E rembar de tremblor	prie	xxv. f.	L a lame seban le piquant	prie	xxv. f.
A dresse rive de la rive aux poyntoires			J chan vonan	prie	xxv. f.
J chan de ville dieu	prie	xxv. f.	J chan de courbeul	prie	xxv. f.
W am jehanne de rimes	prie	xxv. f.	I gnat le chartron	prie	xxv. f.
E deline la guillere	prie	xxv. f.	J chan hoquec	prie	xxv. f.
J chan gobin tailleur de pierre	prie	xxv. f.	H eun de louuisan	prie	xxv. f.
P ierre le boursier	prie	xxv. f.	S erfri de noisel	prie	xxv. f.
P ierre de seulz	prie	xxv. f.	J aques de coupbeul	prie	xxv. f.
P asoil de campigne	prie	xxv. f.	P ierre nantier fruiter	prie	xxv. f.
D am jehanne la milloc	prie	xxv. f.	G uill de suar	prie	xxv. f.
L ouens le maure	prie	xxv. f.	C liment roissau	prie	xxv. f.
C ourant le Lombard	prie	xxv. f.	J chan le begue	prie	xxv. f.
O ay de Bourgnal	prie	xxv. f.			

Comptabilité double

Luca Pacioli (1445-1517),
père de la comptabilité double,
la méthode vénitienne de
tenue des comptes, adoptée
progressivement par tous les
pays du monde.

Tableau : *Luca Pacioli avec son élève Guidobaldo I^{er} de Montefeltro* (1495), attribué à Jacopo de' Barbari, musée Capodimonte de Naples



Comptabilité double

- Système actuel, dit **en partie double**, inventé par Lucas Pacioli, un mathématicien vénitien du 15^{ème} siècle
- C'est un **système compensatoire** où chaque débit « compense » un crédit, et inversement
- Cet équilibre est matérialisé par un système de **balance comptable**, dans lequel les crédits et les débits sont comptabilisés jusqu'à engendrer un solde positif ou négatif
- A l'époque moderne, ce système de comptabilité est employé en vue d'établir les **états financiers** tels que le **bilan** et le **compte de résultat**

Comptabilité double

Exemple de **bilan financier** utilisant une comptabilité double

Du côté des **actifs**, ce sont les biens immobiliers et mobiliers, ressources et possessions diverses qui sont listés

Du côté des **passifs**, ce sont les crédits, dettes, réserves et capitaux propres qui sont listés

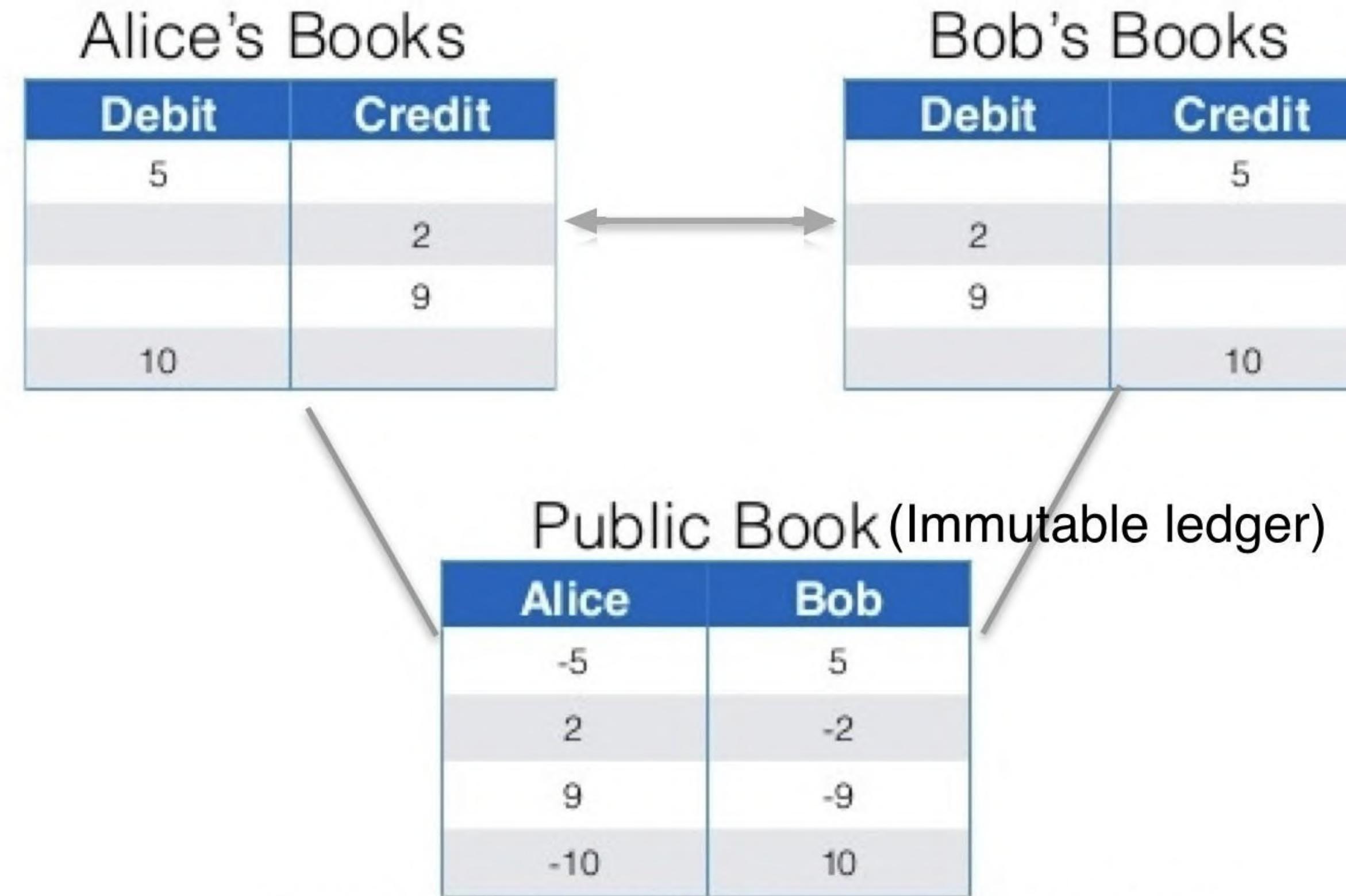
BILAN			
ACTIF (Composition du patrimoine - Emploi des ressources)		PASSIF (Origine du patrimoine - Origine des ressources)	
Actif immobilisé	<p>Entre autre :</p> <ul style="list-style-type: none"> • Les immobilisations corporelles (biens durables : terrains, bâtiments, matériels, mobiliers...) • Les immobilisations financières (Prêts accordés, dépôts et cautionnements versés...) 	Fonds propres	Apports Réserves Résultat de l'exercice
Actif circulant	<p>Entre autre :</p> <ul style="list-style-type: none"> • Les créances (ce que doivent les usagers...) • Les valeurs mobilières de placement (SICAV...) • Les disponibilités (en banque, en caisse...) 	Dettes	Dettes financières (Emprunts...) Dettes d'exploitation (auprès des fournisseurs non réglés...) Dettes diverses
Total de l'actif = Total du passif			

Comptabilité triple

- Système de la blockchain, à **trois entrées : l'actif, le passif et le registre (ledger) public**
- L'actif et le passif se compensent automatiquement, pas risque d'erreur
- Le bilan comptable s'opère sans intervention extérieure, il n'y a **pas de tiers de confiance** ou d'intermédiaire certificateur (expert comptable, commissaire au compte...)
- Toute partie tierce peut vérifier la validité d'une transaction entre deux autres parties

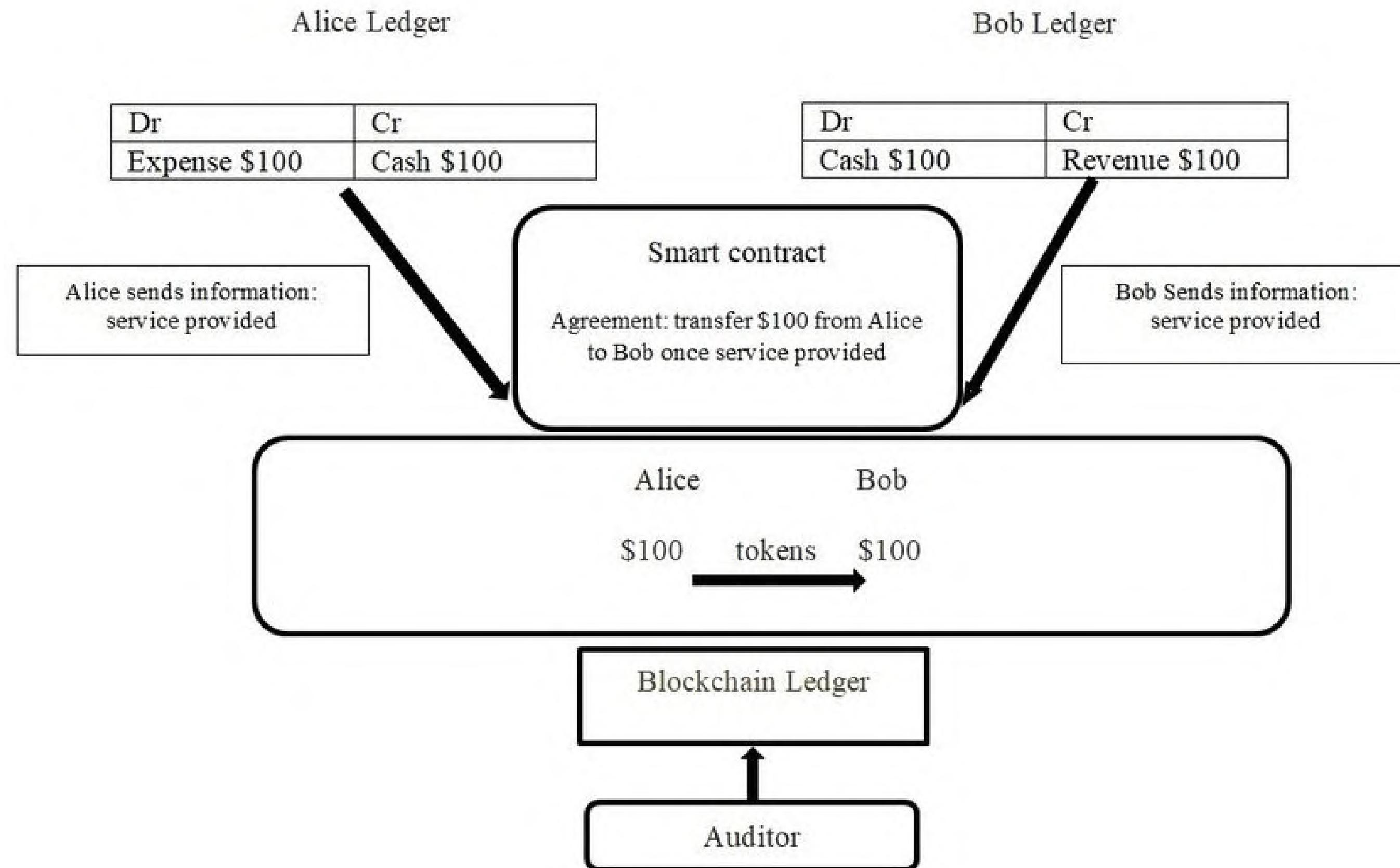
Un **audit simple** peut se faire via un **block explorer** qui contient toutes les informations minimales liées à un compte (adresse publique, solde, historique des transactions) ou à une transaction (bloc, montant,...)

Comptabilité triple



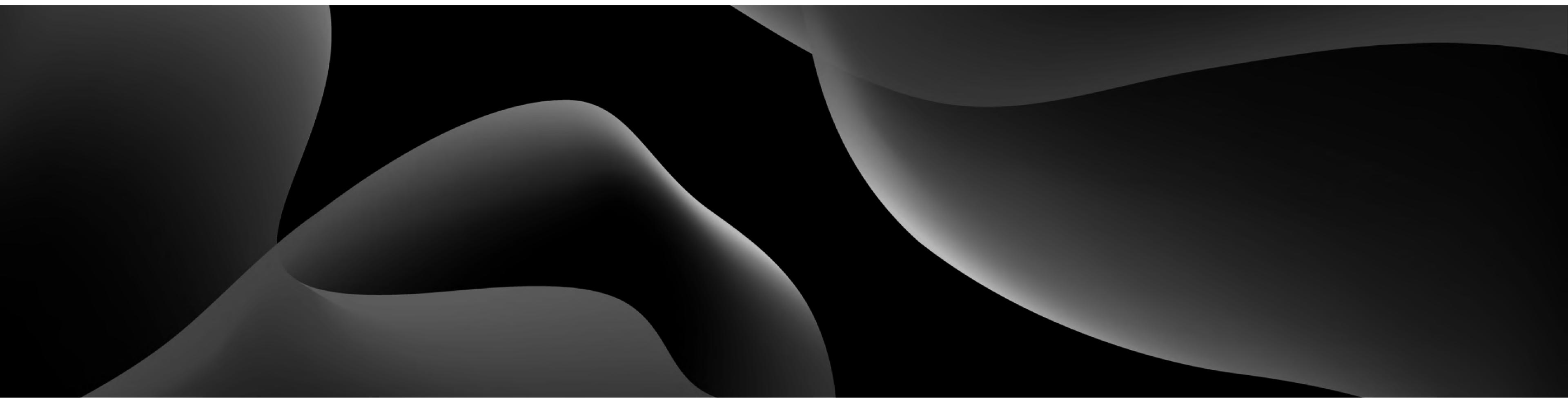
Government , Banks/Financial institution, Auditors

Comptabilité triple

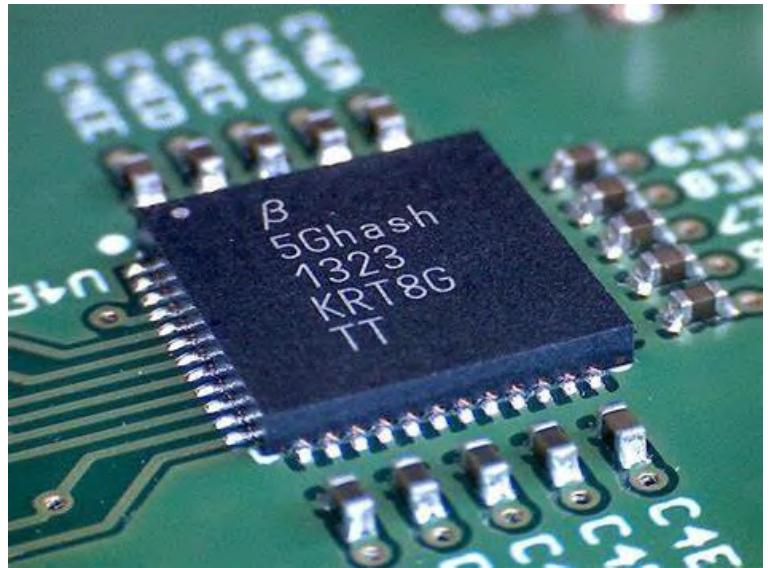


02

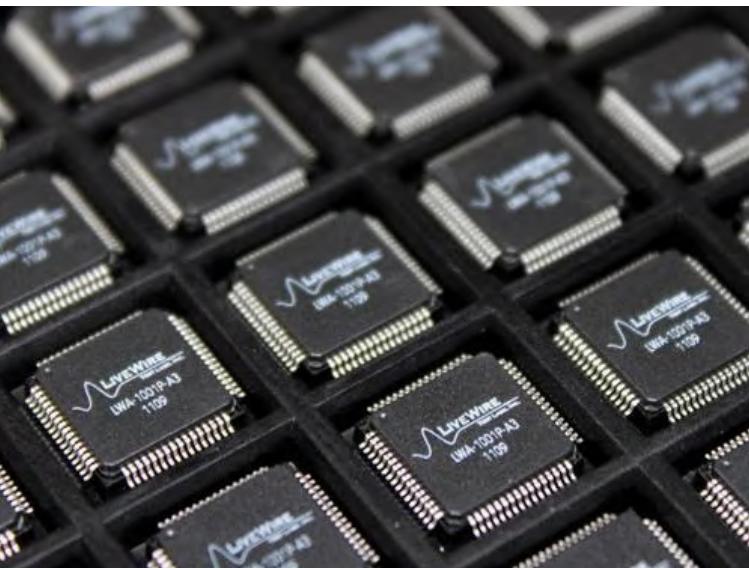
Un registre numérique sécurisé par cryptographie



De la puce électronique aux fermes de minage



ASIC



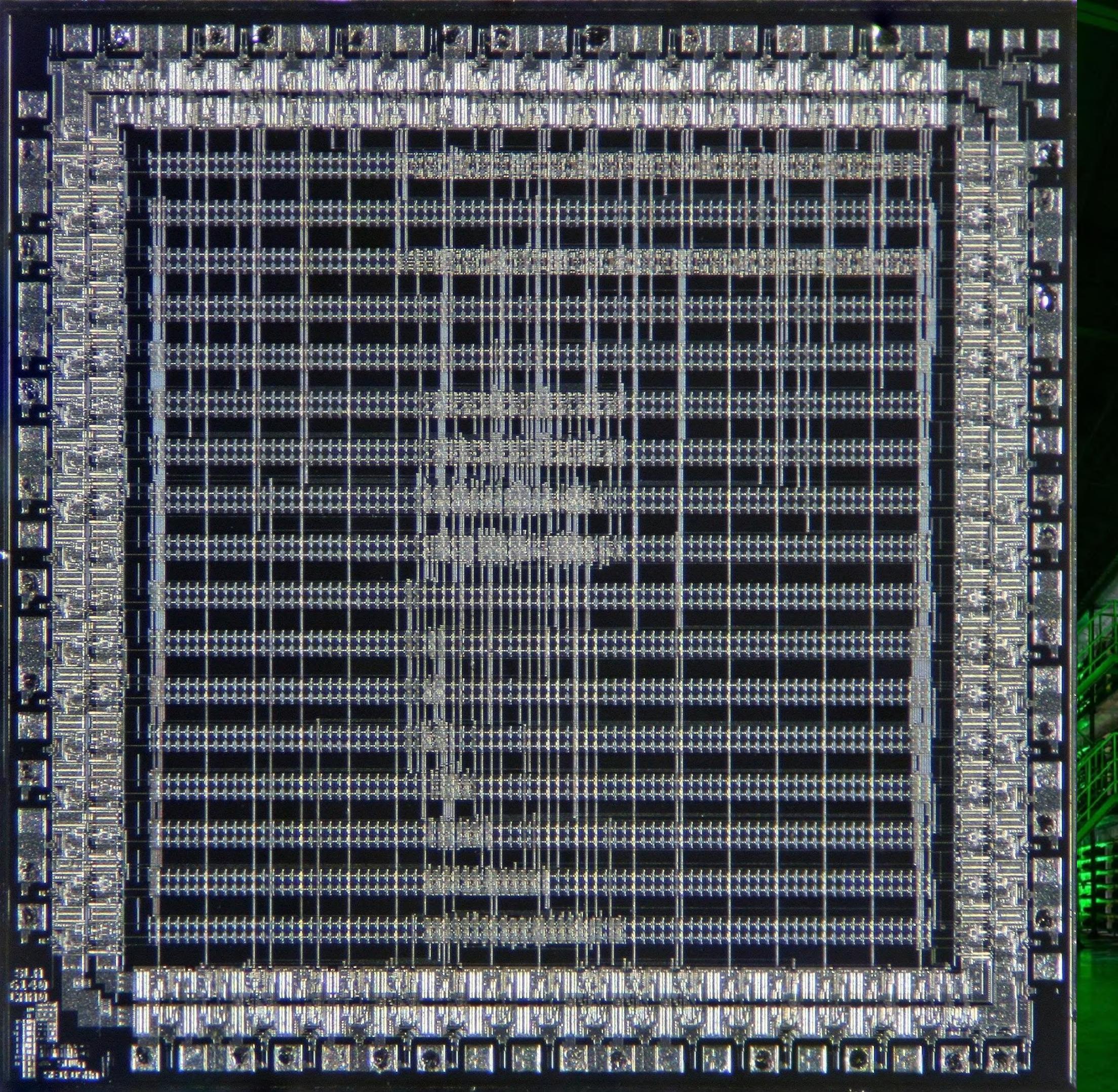
puces ASIC en série



Rig de minage



Racks de minage
en série

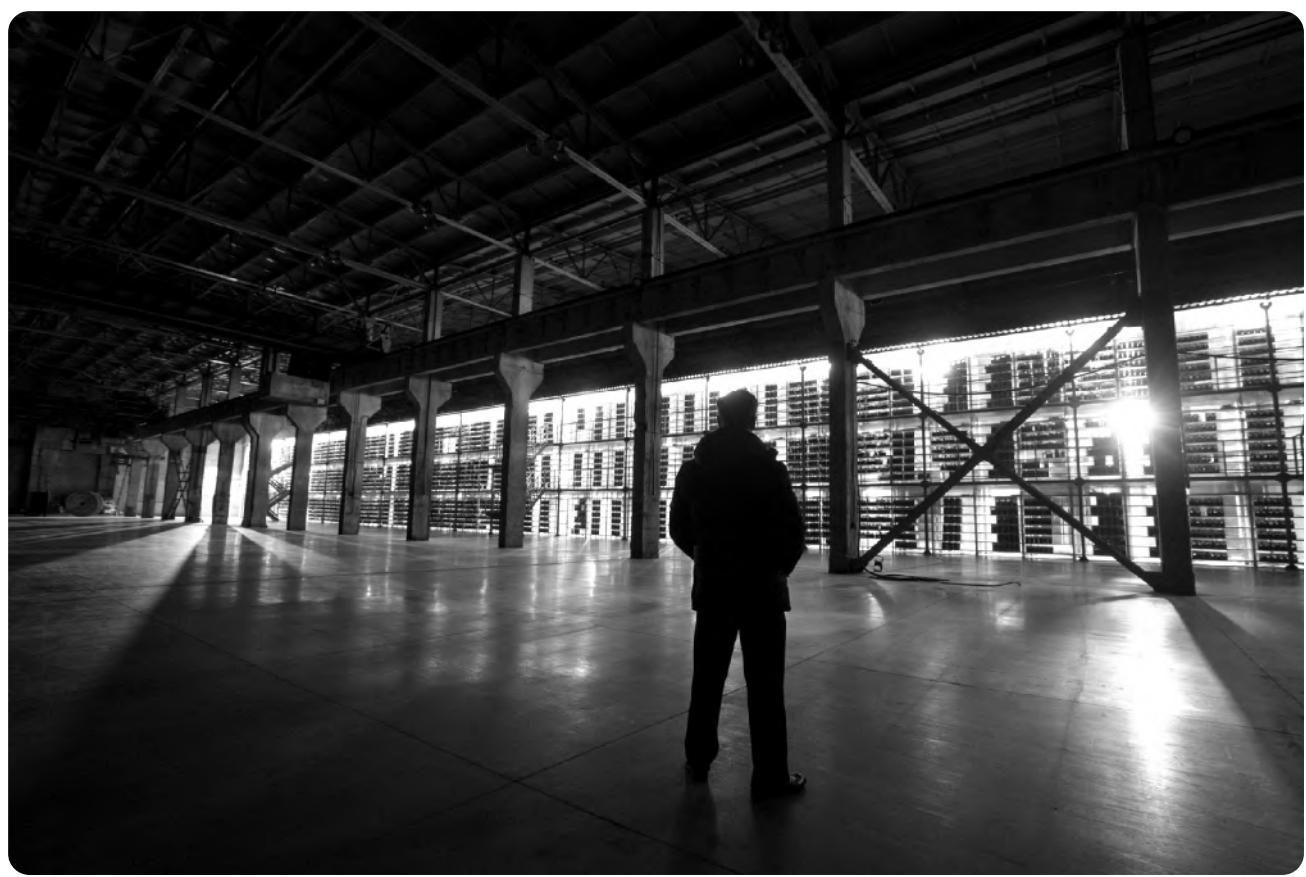




Il faut des salariés très compétents pour entretenir ces machines qui garantissent l'intégrité du registre



Ici, les techniciens de **Bitriver** (Russie) sont formés et certifiés par le géant du minage **Bitmain**, qui fabrique notamment les machines de minage les plus connus au monde, les **Antminer**

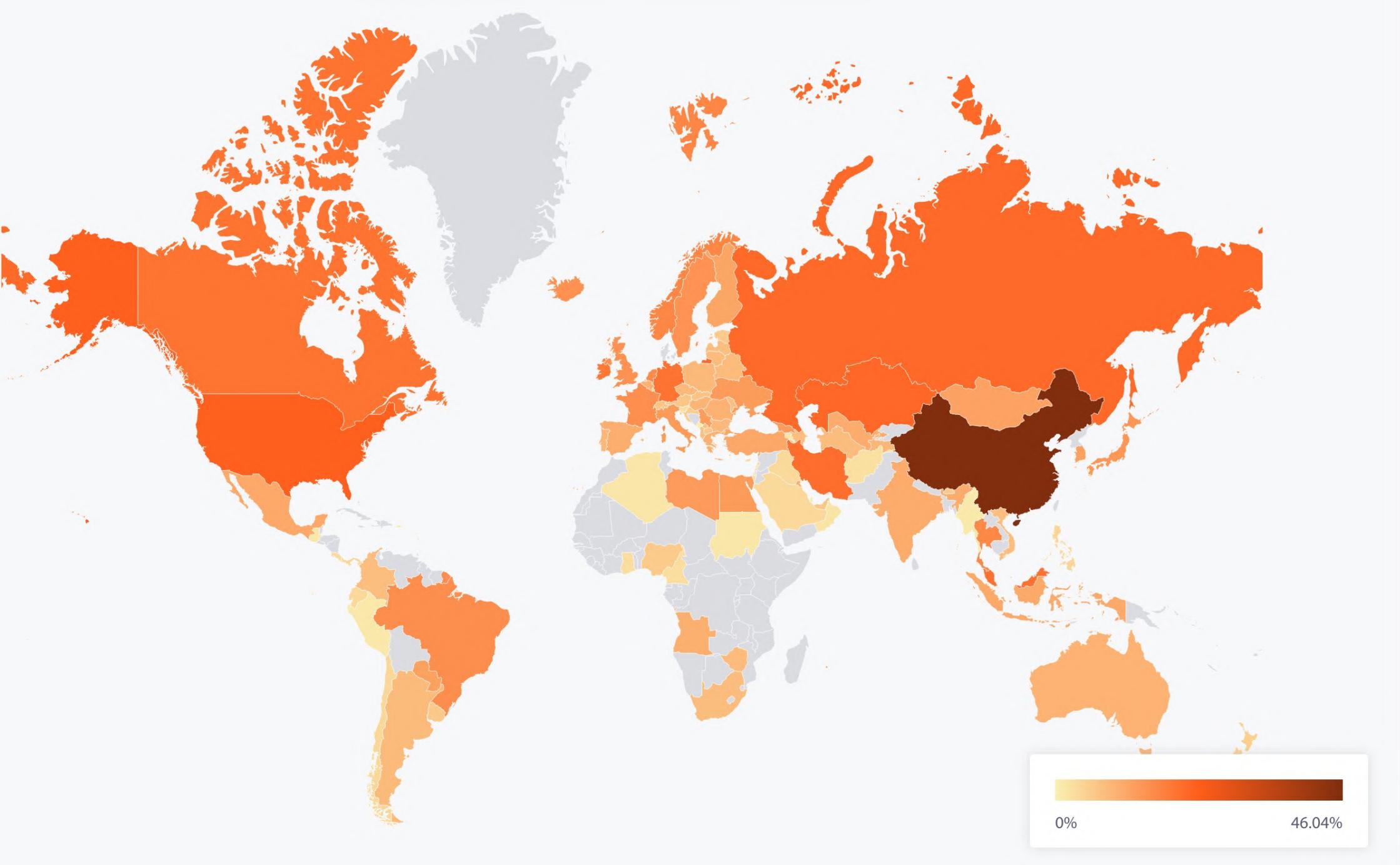


Une répartition du minage en évolution permanente

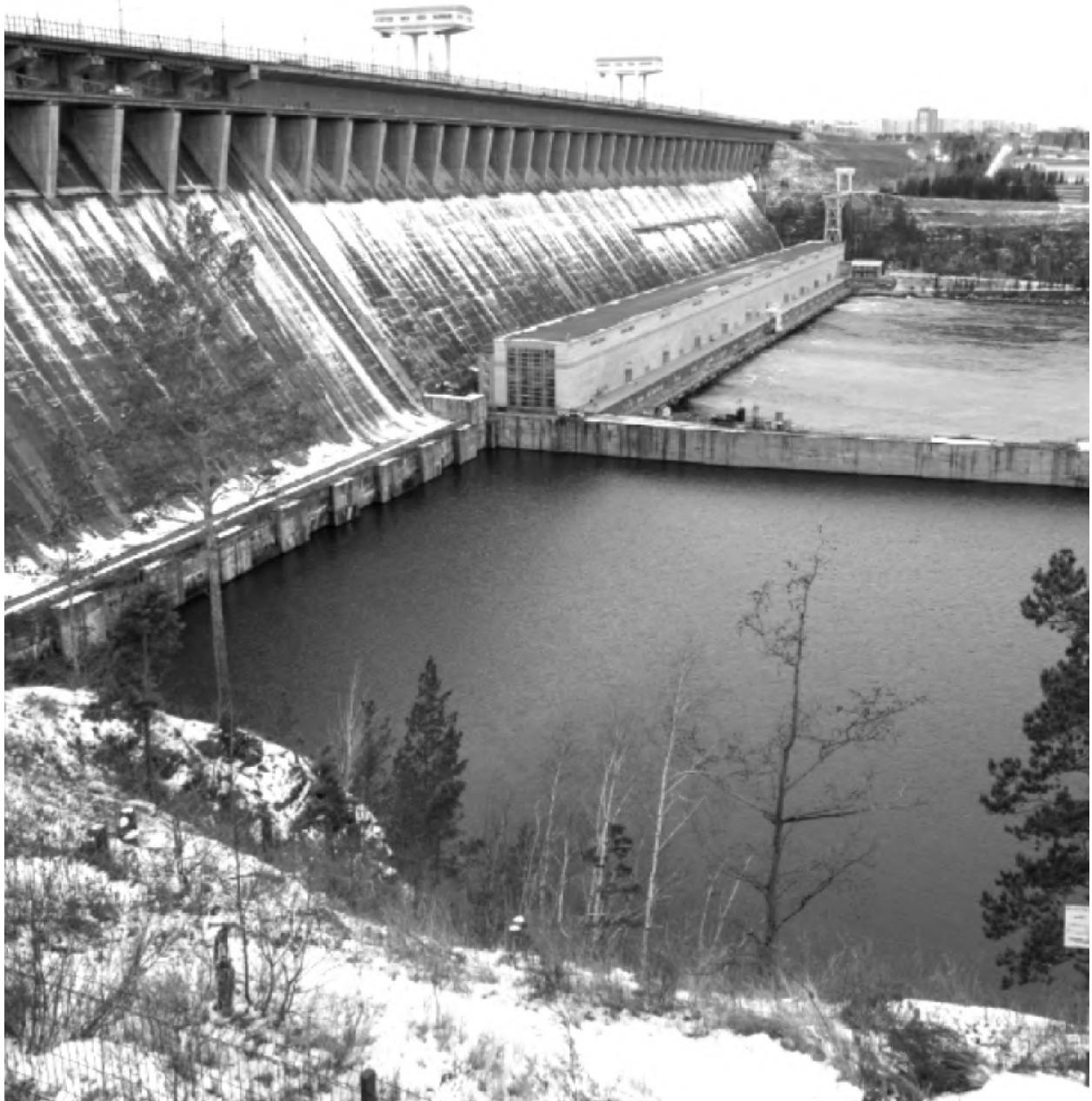
Le coût d'opportunité des mineurs est le résultat d'un jeu d'arbitrage entre 4 aspects :

- Puissance délivrée (hashrate)
- Consommation énergétique
- Efficiency énergétique
- Prix de l'énergie

[Bitcoin mining map](#) de l'université de Cambridge



Un arbitrage constant entre nature et machines



Le niveau de rendement espéré est dépendant d'une imbrication subtile de 4 éléments :

- Machine de minage
- Source d'énergie
- Emplacement géographique
- Saisonnalité

Demande journalière d'énergie pour Bitcoin (GW, données CBECI)

Historical Bitcoin network power demand

Select an area by dragging across the lower chart



Consommation mensuelle d'électricité pour Bitcoin (TWh, données CBECI)

Total Bitcoin electricity consumption

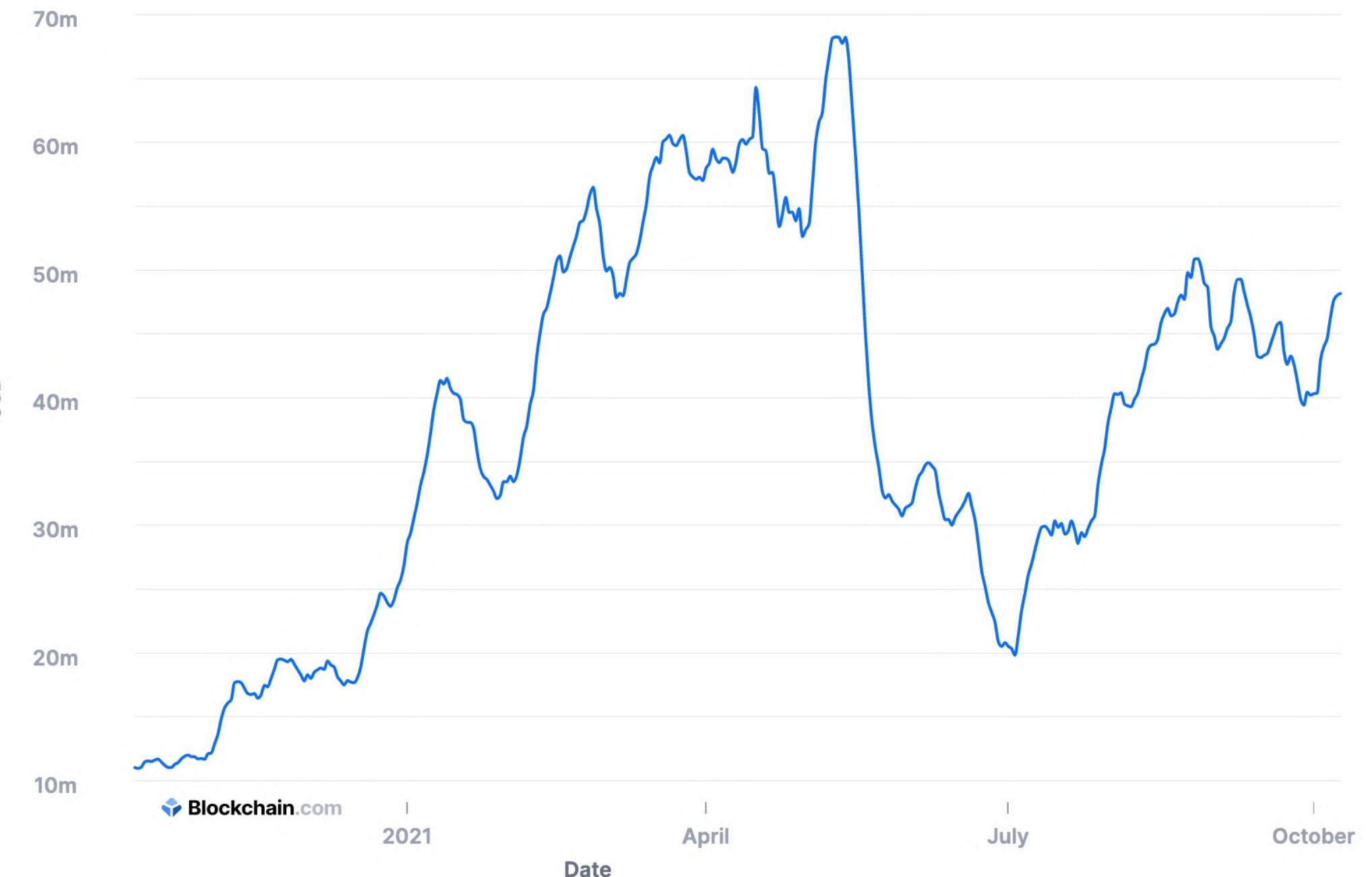
Select an area by dragging across the lower chart



Revenu moyen des mineurs sur une année (échelle linéaire)

Le modèle économique de l'activité de minage est fonction de 3 variables dynamiques :

- Les facteurs exogènes environnementaux vus précédemment
- Les facteurs endogènes techniques vus précédemment
- Le modèle « tokenomics » de la blockchain étudiée, soit les facteurs endogènes économiques



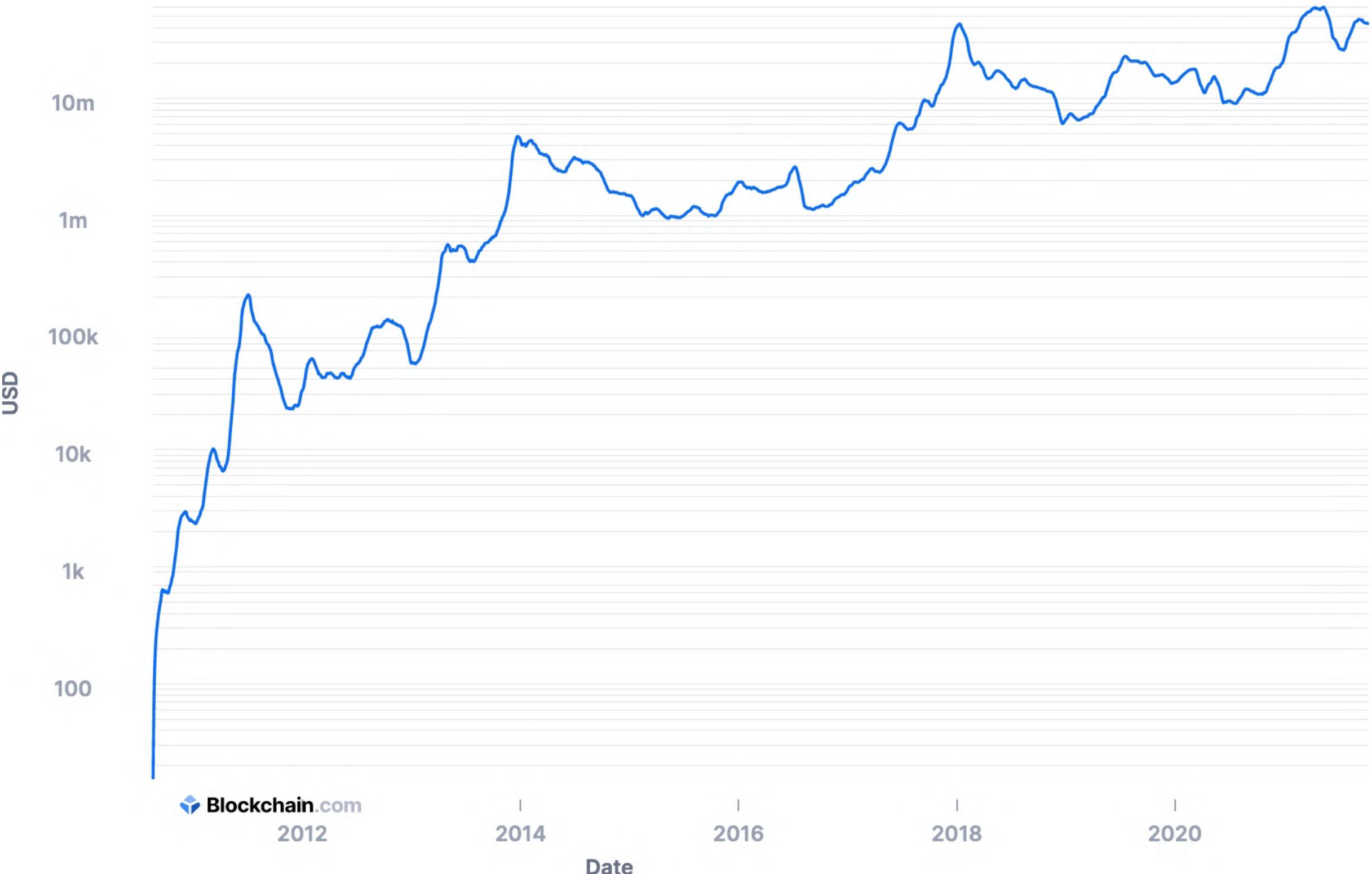
Revenu moyen des mineurs historique (échelle logarithmique)

Le modèle économique de l'activité de minage est fonction de 3 variables dynamiques :

Les facteurs exogènes environnementaux vus précédemment

Les facteurs endogènes techniques vus précédemment

Le modèle « tokenomics » de la blockchain étudiée, soit les facteurs endogènes économiques



Profitabilité dynamique des mineurs par type d'ASIC

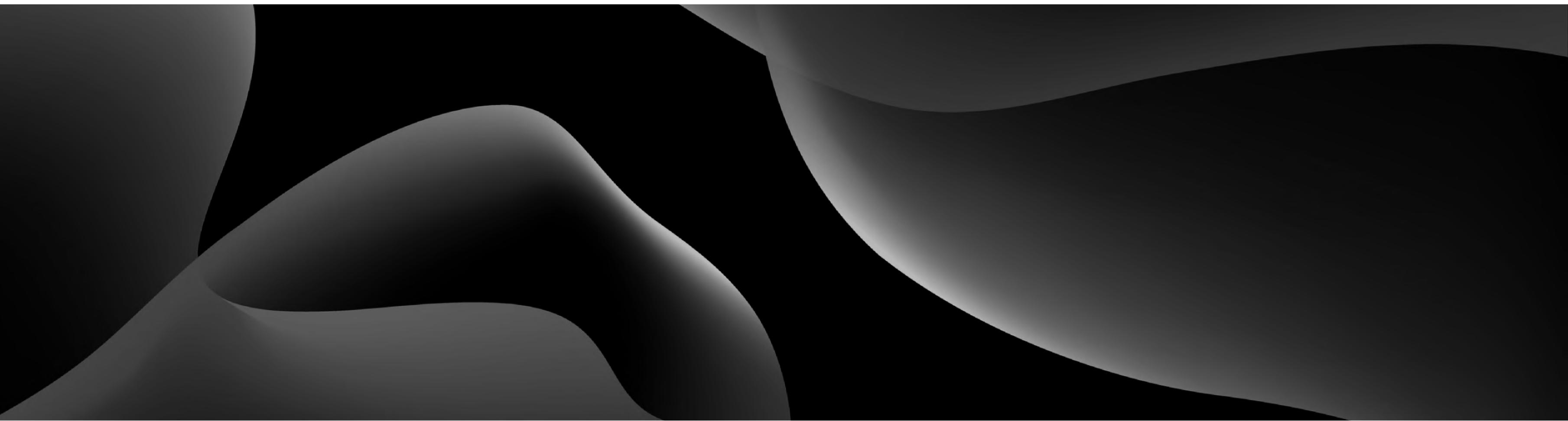
Search: sha-256

Model	Release	Hashrate	Power	Noise	Algo	Profitability
 Bitmain Antminer S19 XP (140Th)	Jul 2022	140 Th/s	3010W	75 db	SHA-256	\$44.16 /day 
 MicroBT Whatsminer M30S++	Oct 2020	112 Th/s	3472W	75 db	SHA-256	\$32.26 /day 
 iPollo B2	Oct 2021	110 Th/s	3250W	75 db	SHA-256	\$32.15 /day 
 Bitmain Antminer S19 Pro (110Th)	May 2020	110 Th/s	3250W	75 db	SHA-256	\$32.15 /day 
 Bitmain Antminer S19j Pro (104Th)	Jul 2021	104 Th/s	3068W	75 db	SHA-256	\$30.41 /day 
 Bitmain Antminer S19j Pro (100Th)	Jun 2021	100 Th/s	3050W	75 db	SHA-256	\$28.95 /day 
 Bitmain Antminer S19j Pro (96Th)	Aug 2021	96 Th/s	2832W	75 db	SHA-256	\$28.07 /day 
 MicroBT Whatsminer M30S+	Oct 2020	100 Th/s	3400W	75 db	SHA-256	\$27.94 /day 
 Bitmain Antminer S19 (95Th)	May 2020	95 Th/s	3250W	75 db	SHA-256	\$26.49 /day 
 Bitmain Antminer S19j (90Th)	Jun 2021	90 Th/s	3250W	75 db	SHA-256	\$24.60 /day 
 Canaan AvalonMiner 1246	Jan 2021	90 Th/s	3420W	75 db	SHA-256	\$24.11 /day 
 Bitmain Antminer T19 (88Th)	Aug 2021	88 Th/s	3344W	75 db	SHA-256	\$23.58 /day 

Une estimation de la profitabilité peut être faite en temps réel [ici](#)

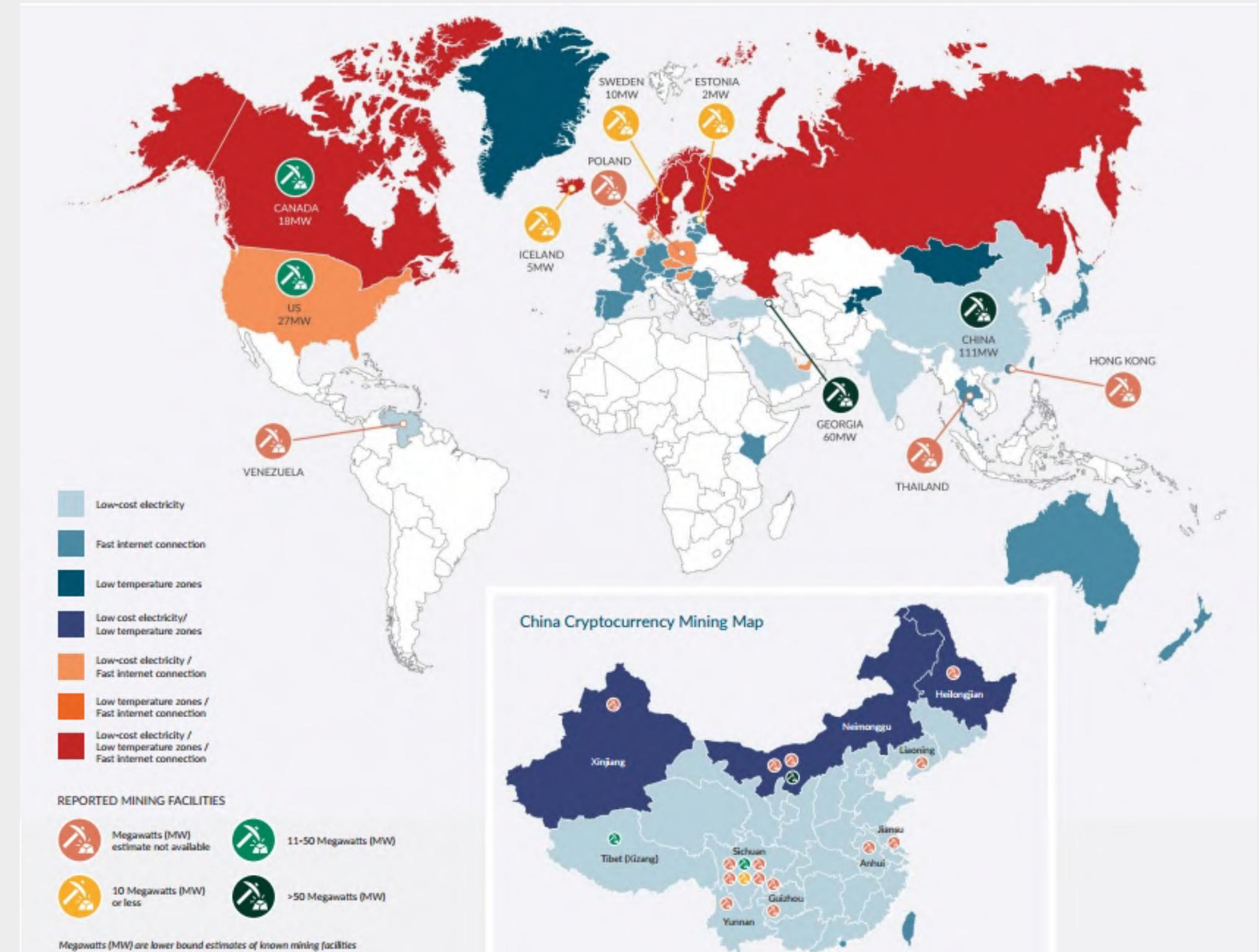
03

De la difficulté du minage (et du hashrate)



Le taux de hachage (hashrate)

- Le taux de hachage, en Terahash/seconde, représente la puissance cumulée de hachage de tous les mineurs à un instant T
- Le hashrate est relativement volatile, car il est dépendant de nombreux facteurs exogènes (environnement politique, infrastructures énergétiques, coût de l'énergie, du hardware, disponibilité des composants électroniques, climat, météo, connexion internet...)



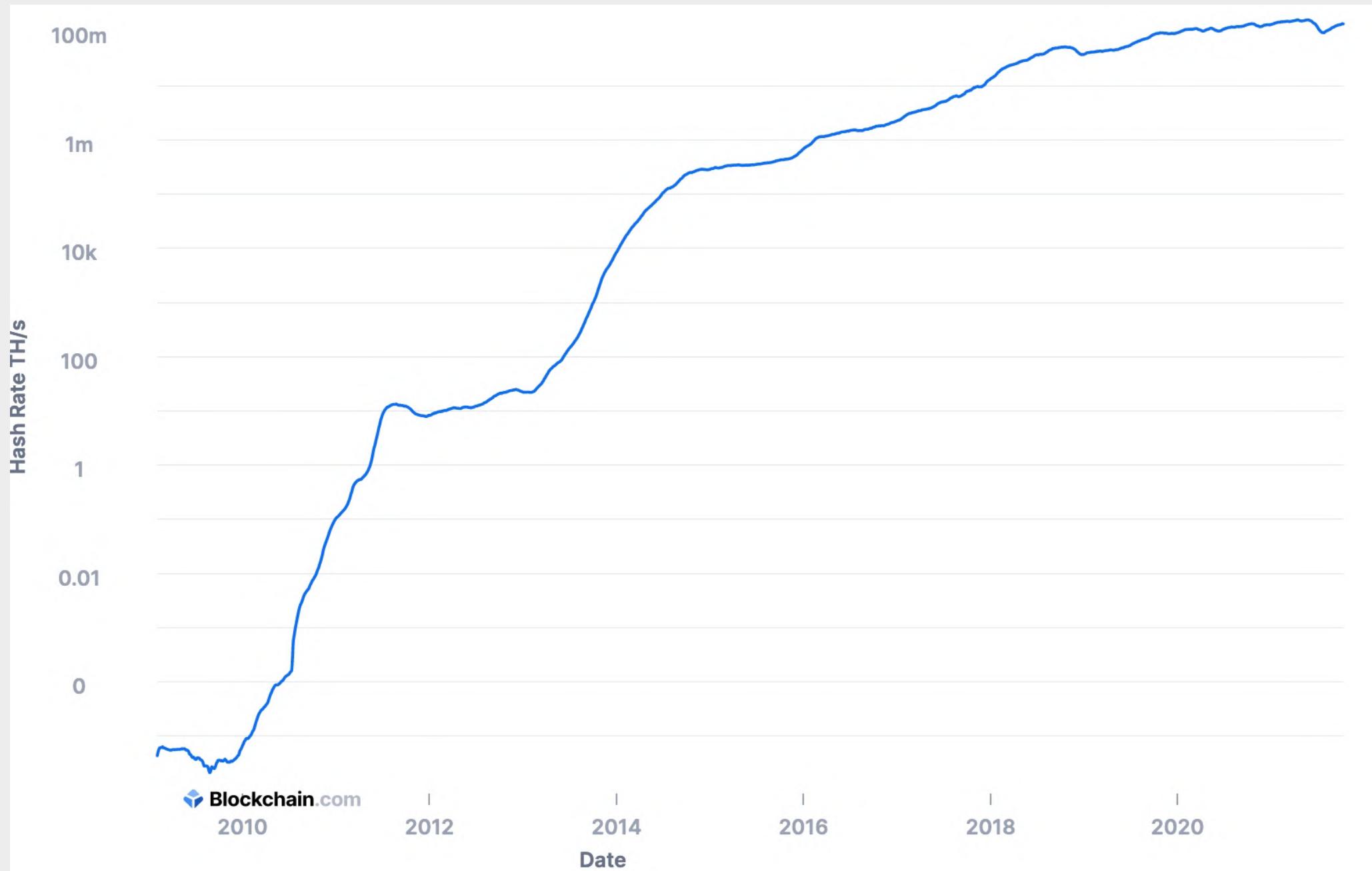
Taux de hachage moyen sur une année (échelle linéaire)

- Le hashrate fluctue du simple au double et inversement au cours d'une année
- En 2021, le taux de hachage atteint environ 180m TH/s juste avant l'été, au moment où les cours boursiers atteignent leurs « plus hauts »
- Lors d'un « pic de demande », les mineurs vont anticiper des fees plus nombreuses et plus élevées, donc renforcer leurs capacités de minage



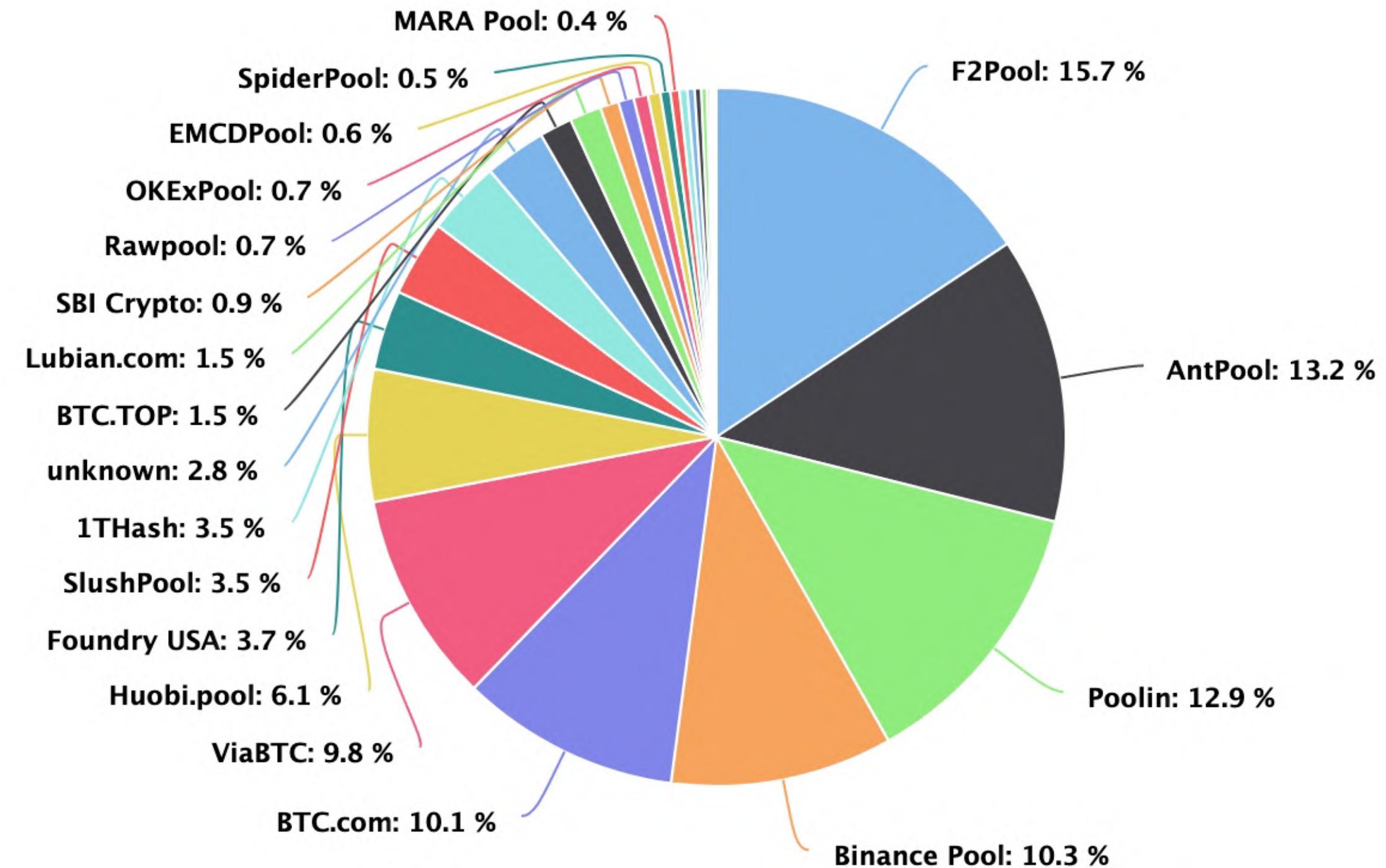
Taux de hachage historique (échelle logarithmique)

- Structurellement, le taux de hachage est en très forte augmentation (courbe exponentielle)
- Il a explosé à partir de 2017, passant d'environ 2,2m TH/s à plus de 150 mi 2021
- Il est l'un des moyens privilégiés de connaissance du niveau de sécurité dynamique d'un réseau blockchain, car il représente le coût d'opportunité de l'attaque d'un réseau, que l'on peut retrouver [ici](#)



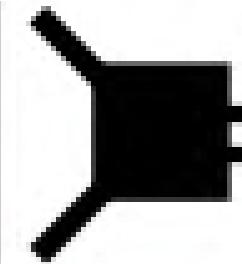
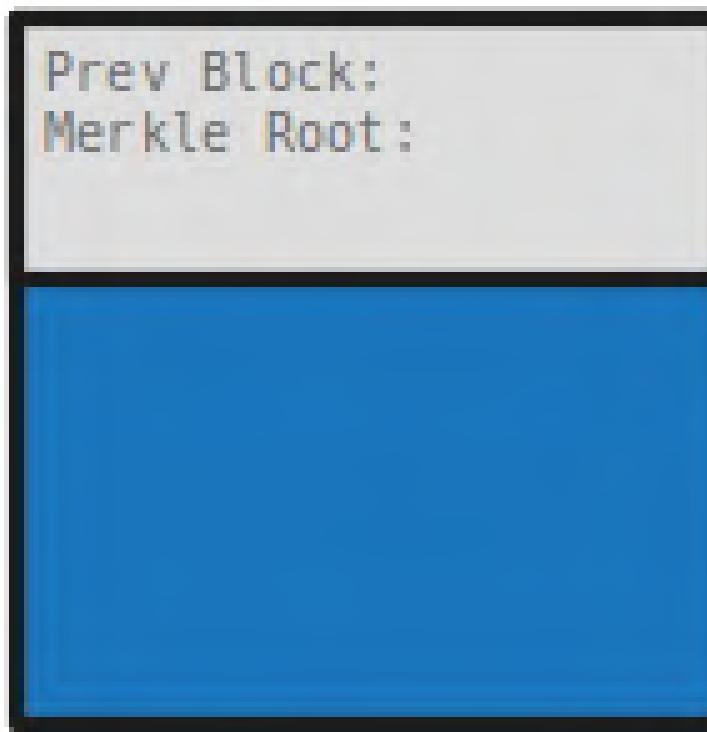
Répartition moyenne du hashrate entre pools de minage (2021)

- L'identité des mineurs est très largement connue (97%)
- La plupart des pools appartiennent à de grands groupes spécialisés, voire à des multinationales de la blockchain (Binance, Huobi...)
- Les groupes chinois fournissent plus de 75% du hashrate
- Une part négligeable du hashrate demeure d'origine inconnue et potentiellement indépendante (3%)

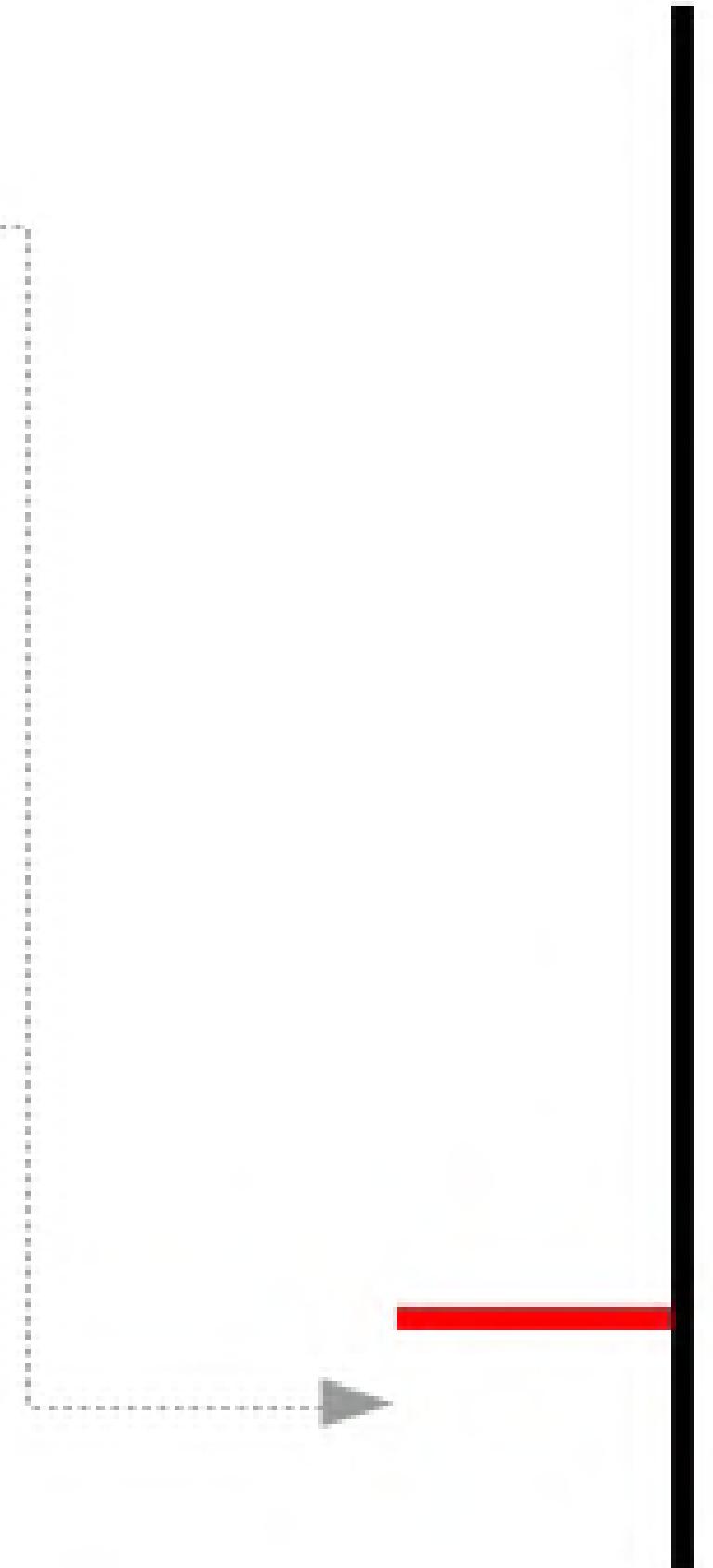


« Pour compenser l'augmentation de la vitesse (de calcul) du matériel (informatique) et l'intérêt variable pour le maintien des nœuds dans le temps, la difficulté de la preuve de travail est déterminée par une moyenne mobile ciblant un nombre moyen de blocs par heure. S'ils sont générés trop vite, la difficulté augmente. »

Satoshi Nakamoto

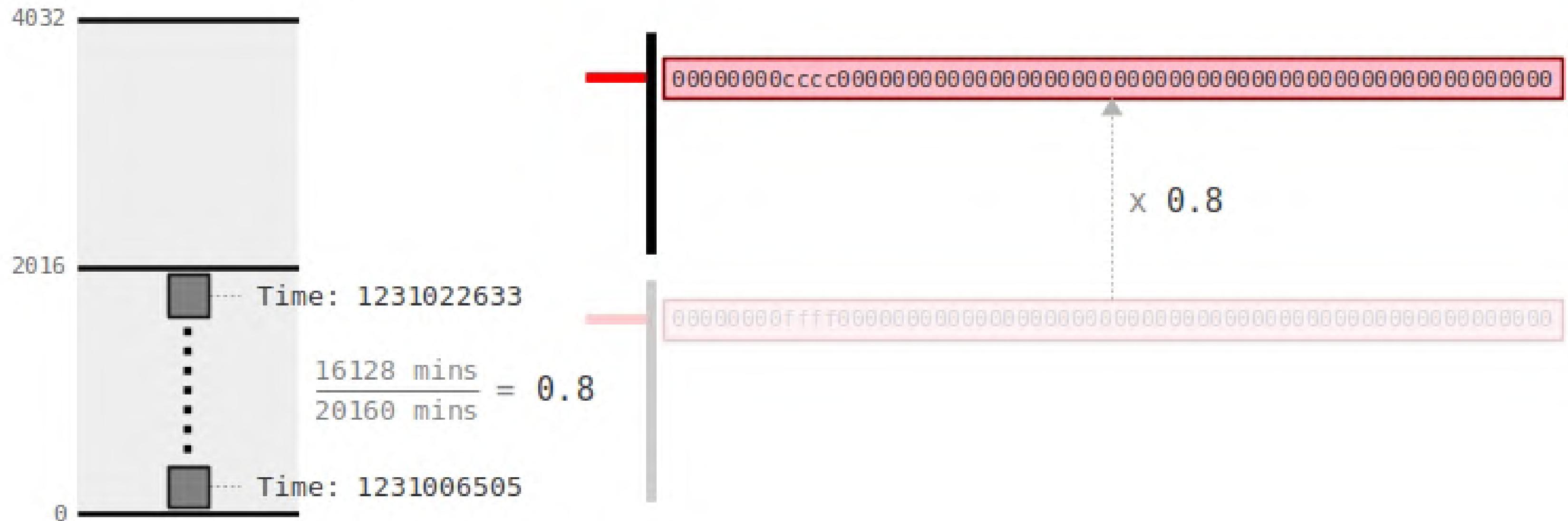


Block Hash



La difficulté du réseau

- La difficulté du réseau représente la difficulté à miner un bloc, à savoir trouver un nonce qui permette au hash du bloc d'être sous la cible de difficulté
- Cette cible est matérialisée par un certain nombre de 0 par lesquels doit commencer le hash du bloc en hexadécimal (à multiplier par 4 pour l'écriture linéaire)



La difficulté du réseau

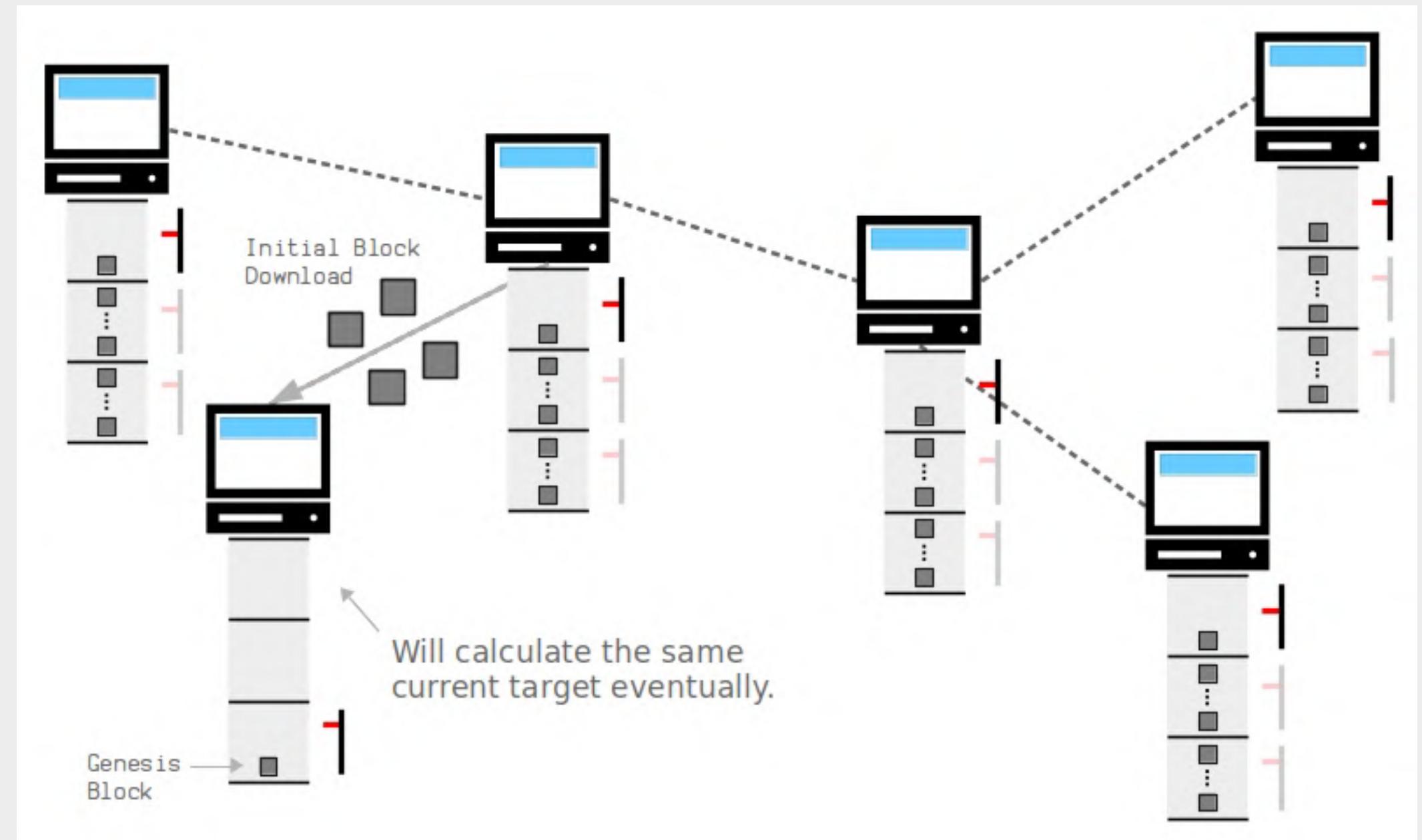
- Tous les 2016 blocs, chaque nœud divise le temps écoulé depuis les 2016 derniers blocs, et le divise par 20160 (temps cible = $10\text{mn} \times 2016 \text{ blocs} = 20160\text{mn}$)
- Si le résultat n'est pas égal à 1, alors le nœud va automatiquement modifier sa cible de difficulté : la cible est réduite et la difficulté augmentée si le résultat est inférieur à 1, et inversement.

« Tout le monde fait le même calcul avec les mêmes données de chaîne, donc tous obtiennent le même résultat au même maillon de la chaîne »

Satoshi Nakamoto

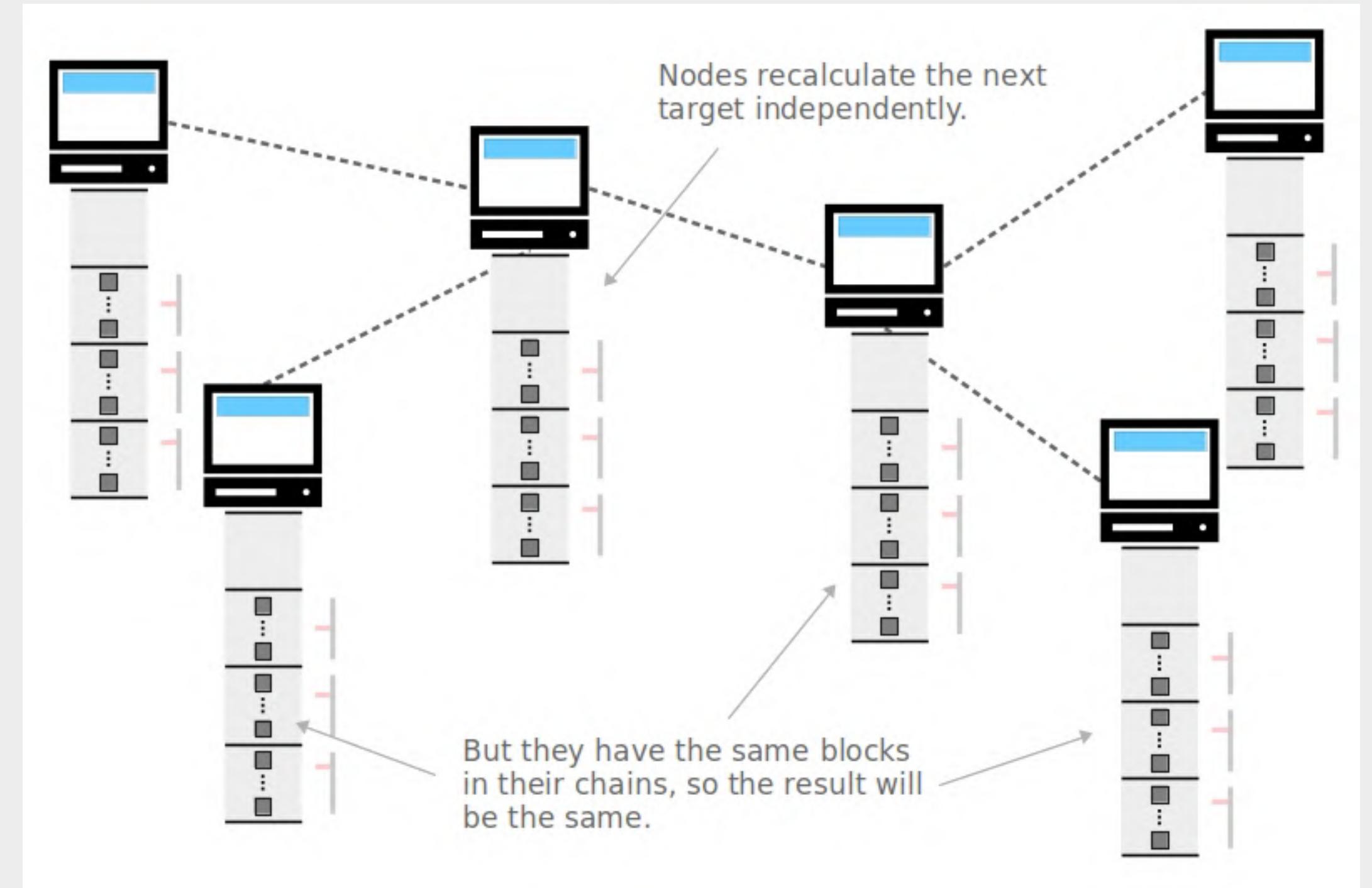
La difficulté du réseau

- Chaque nœud calcule de manière indépendante sa cible de difficulté
- Mais chaque nœud possède une copie locale de la chaîne de blocs depuis le bloc génésis, et va donc opérer exactement les mêmes calculs que tous les autres
- En vertu de la règle « la chaîne la plus longue prévaut », tous les nœuds du réseau partagent la même cible, même si il peut exister des anomalies pendant quelques blocs



La difficulté du réseau

- Chaque nœud calcule de manière indépendante sa cible de difficulté
- Mais chaque nœud possède une copie locale de la chaîne de blocs depuis le bloc génésis, et va donc opérer exactement les mêmes calculs que tous les autres
- En vertu de la règle « la chaîne la plus longue prévaut », tous les nœuds du réseau partagent la même cible, même si il peut exister des anomalies pendant quelques blocs



De la difficulté à ... atteindre la cible

- Il faut imaginer la cible de difficulté comme une cible de tir à l'arc
- Plus on ajoute de 0, plus on réduit le diamètre de la cible, plus le tir est difficile
- Moins on ajoute de 0, plus on augmente le diamètre de la cible, plus le tir est facile
- A cible constante, un hashrate plus important correspond à une distance de tir plus courte : le tir est facilité, et inversement.
- La variable d'ajustement du hashrate, c'est la difficulté du réseau



Mais pourquoi tant de difficulté..?

Le système d'ajustement dynamique entre hashrate et difficulté offre plusieurs avantages au niveau technique :

- Cela laisse au réseau le temps de diffuser les blocs, les nœuds ayant le temps de se mettre à jour durant ces 10 min
- Le risque de forks temporaires est diminué puisque la cible de 10 min est une option conservatrice
- Puisque le risque de forks temporaires est diminué, le risque de réorganisation de chaîne diminue également
- Il n'y a pas de gâchis d'énergie, les mineurs dirigent leur hashrate vers la bonne chaîne et améliore ainsi sa sécurité

Mais pourquoi tant de difficulté..?

Le système d'ajustement dynamique entre hashrate et difficulté offre plusieurs avantages au niveau économique :

- Le schéma de la création monétaire de Bitcoin est le suivant : à chaque nouveau bloc miné, de nouveaux bitcoins sont créés, regroupés au sein de la transaction « coinbase » du bloc en question. Assurer une latence de 10 min entre chaque bloc assure un rythme de création monétaire stable. Or, la stabilité d'une monnaie est le premier gage de la confiance qu'elle inspire. N'hésitez pas à comparer ce point avec le fonctionnement des monnaies FIAT.

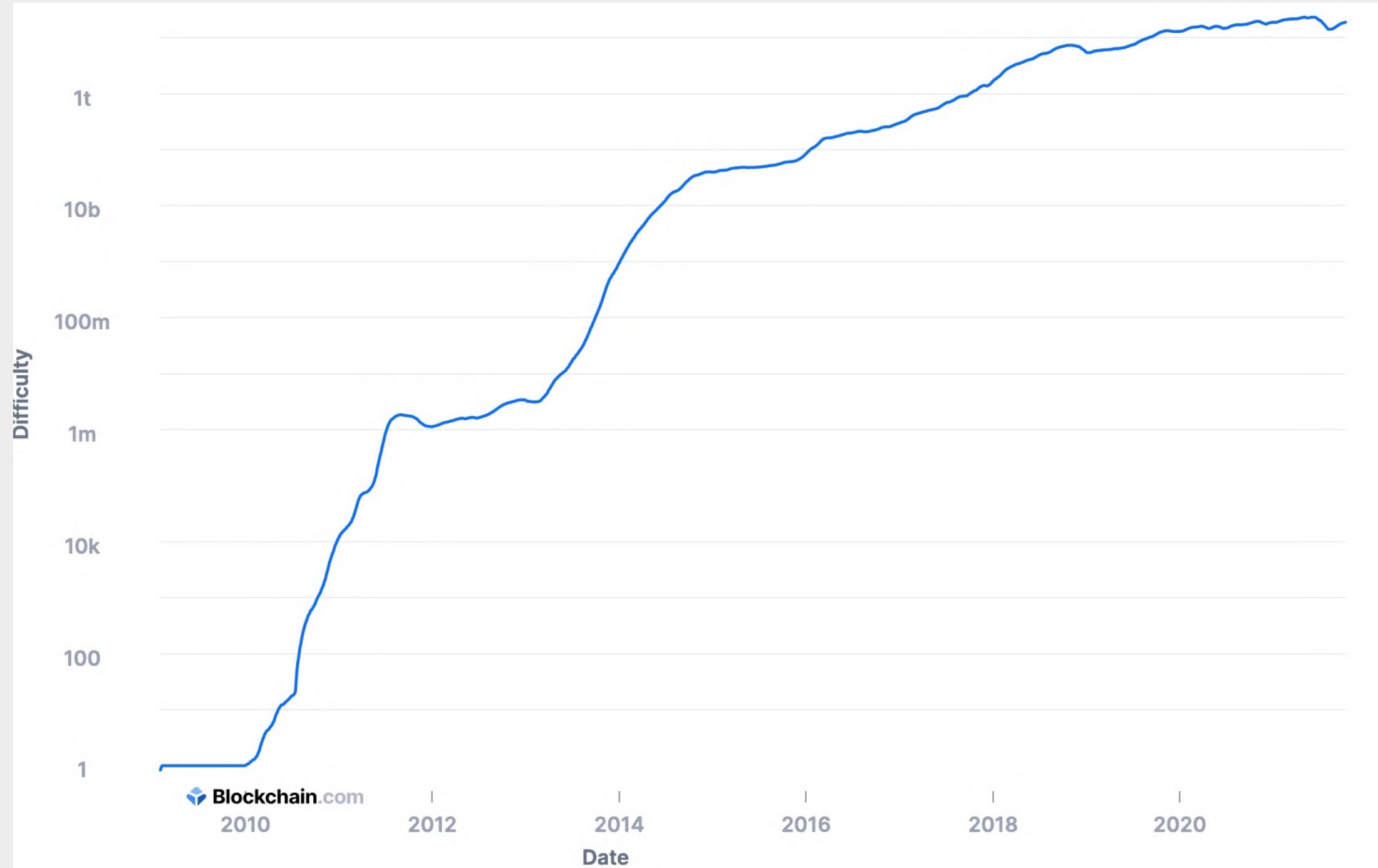
Mais pourquoi tant de difficulté..?

Le système d'ajustement dynamique entre hashrate et difficulté offre plusieurs avantages au niveau économique :

- Le système technique s'adapte au modèle économique des mineurs, et non l'inverse. C'est brillant, car en l'absence de tiers chargé d'assurer la confiance, les mineurs sont les acteurs principaux de la sécurité comme de l'effectivité d'une blockchain. Il est donc essentiel de leur proposer le système le plus incitatif possible, sur le plan économique comme technique.

Difficulté historique (échelle logarithmique)

- La difficulté du réseau étant directement fonction du hashrate (au sens propre!), elle en est en partie le reflet
- En partie seulement, car l'ajustement se fait *a posteriori* et non en temps réel
- Logiquement, la difficulté a suivi une courbe exponentielle et a « explosé » à partir de 2017, « courant » après celle du hashrate



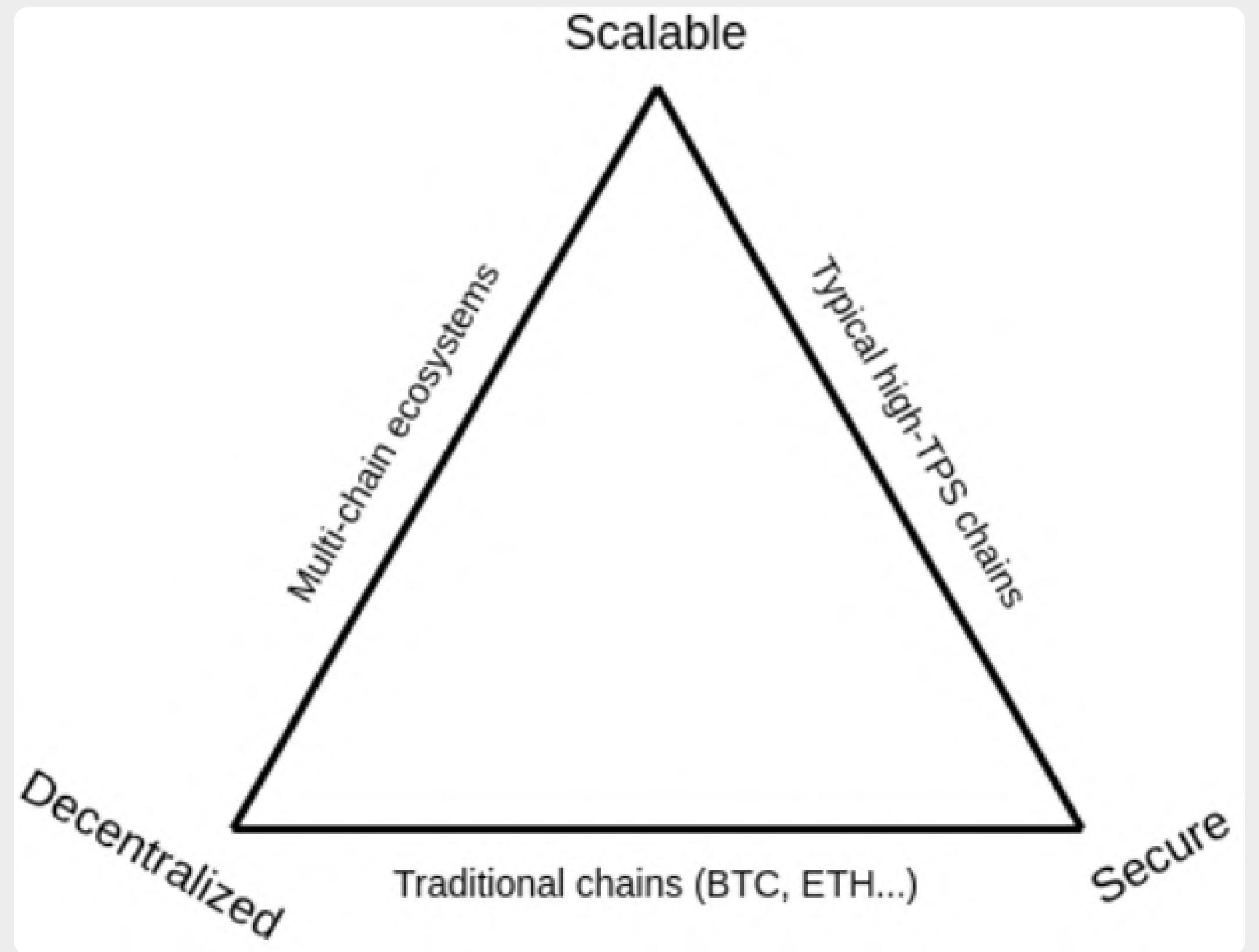
Sécurité

Décentralisation

Scalabilité

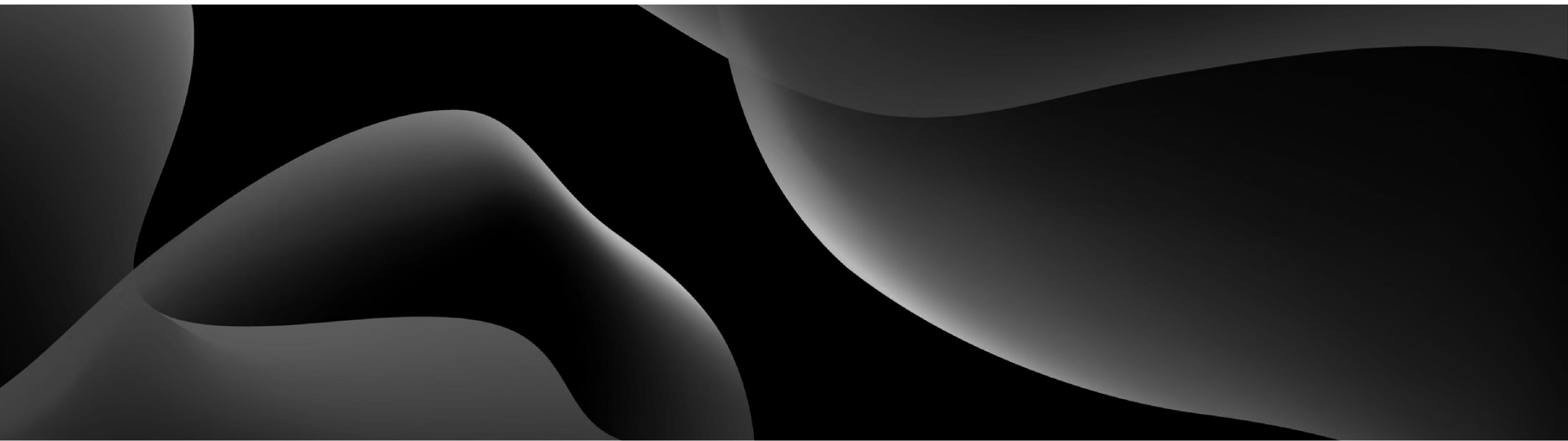
Conclusion - Le trilemme de la blockchain

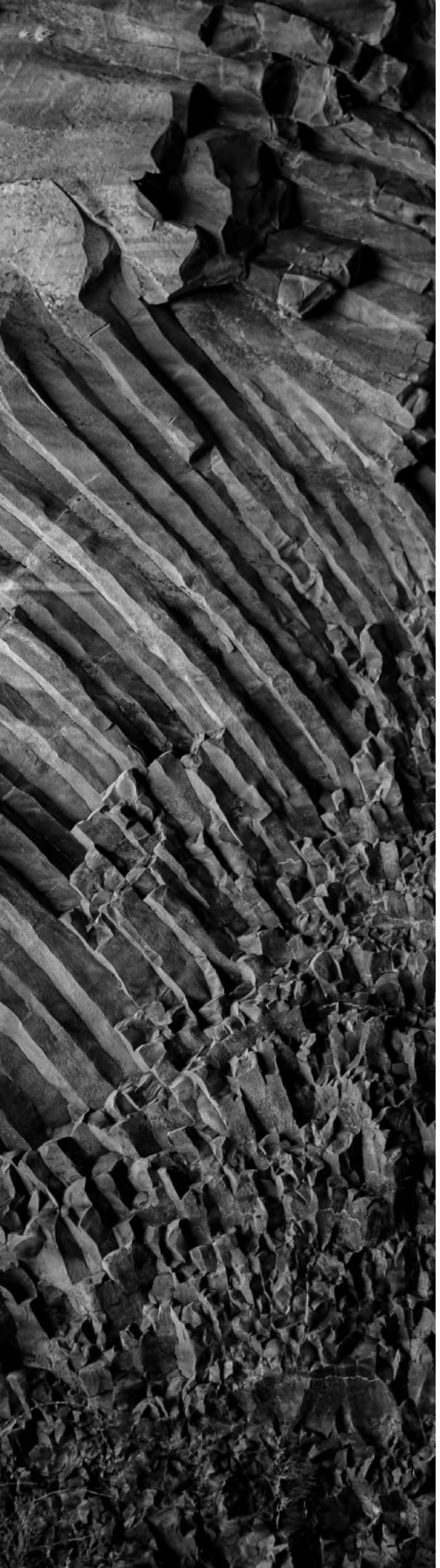
- Un conseil, pensez au trilemme de la blockchain entre les 3 idéaux réputés inconciliables que sont :
 - La sécurité
 - La décentralisation
 - La scalabilité
- Ce trilemme fut clairement énoncé par Vitalik Buterin, principal initiateur de la blockchain Ethereum. Plus de background sur le sujet [ici](#)



05

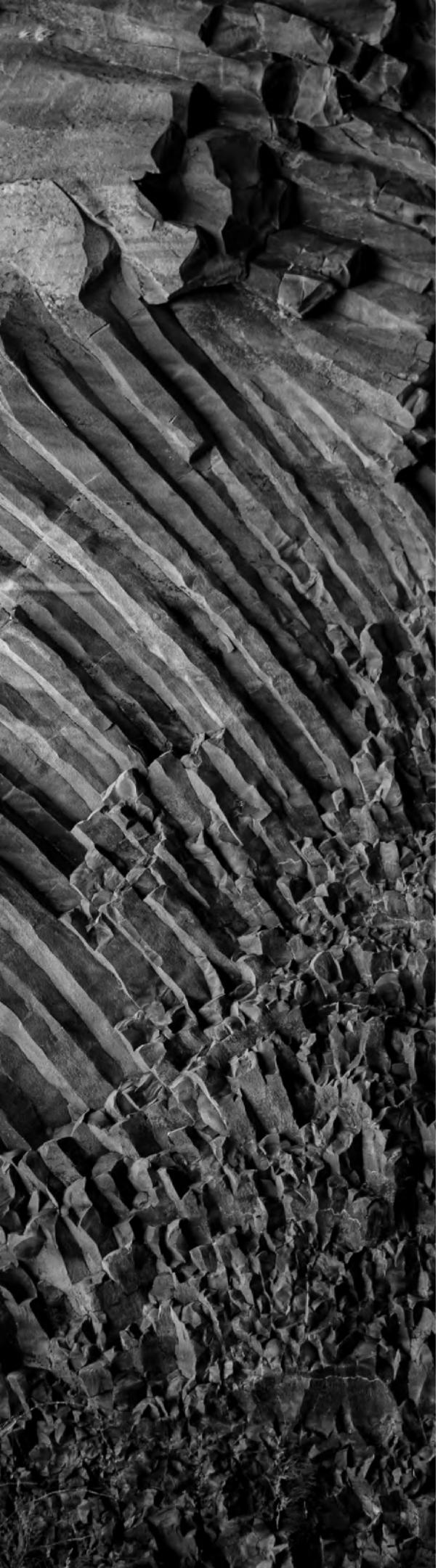
Plonger dans le mempool





Le « mempool »

- Le **memory pool**, ou « mempool », représente **l'ensemble des transactions en attente**, c'est-à-dire non confirmées par au moins un bloc
- Le **BIP35** porté par Jeff Garzick et adopté en 2012 permet à tout nœud d'accéder aux mempools propres à chaque nœud complet, d'une simple commande
- Grâce à cette maj du protocole, les utilisateurs ont une idée du niveau de commissions à payer aux mineurs



Le « mempool »

- Meilleur efficience économique **théorique du système grâce à arbitrage par les prix des mineurs**
- Rapidité de traitement **pour les transactions qui ne nécessitent pas d'attendre la confirmation d'un bloc, dites à vérification simple**
- **Audit permanent** du réseau quant à son activité et son niveau de congestion

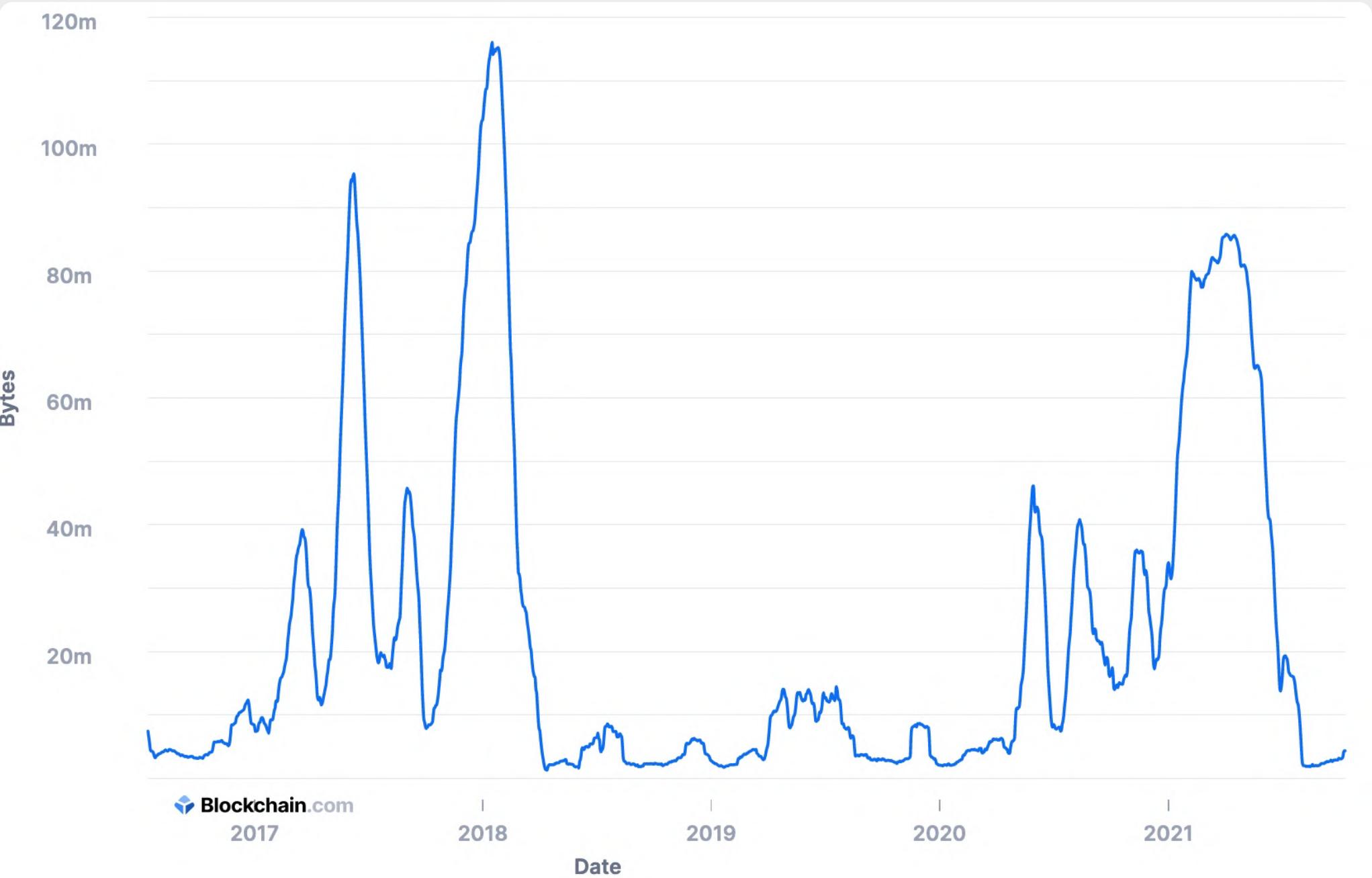
Taille moyenne du « mempool » (megabytes)

La taille du mempool varie fortement au cours du temps

Entre 2017 et 2021, elle atteint des seuils à moins de 5 MB et des plafonds à plus de 100 MB

La taille du mempool est le reflet de l'activité sur la chaîne

La taille du mempool augmente lorsque les demandes de transactions des utilisateurs augmentent, et inversement

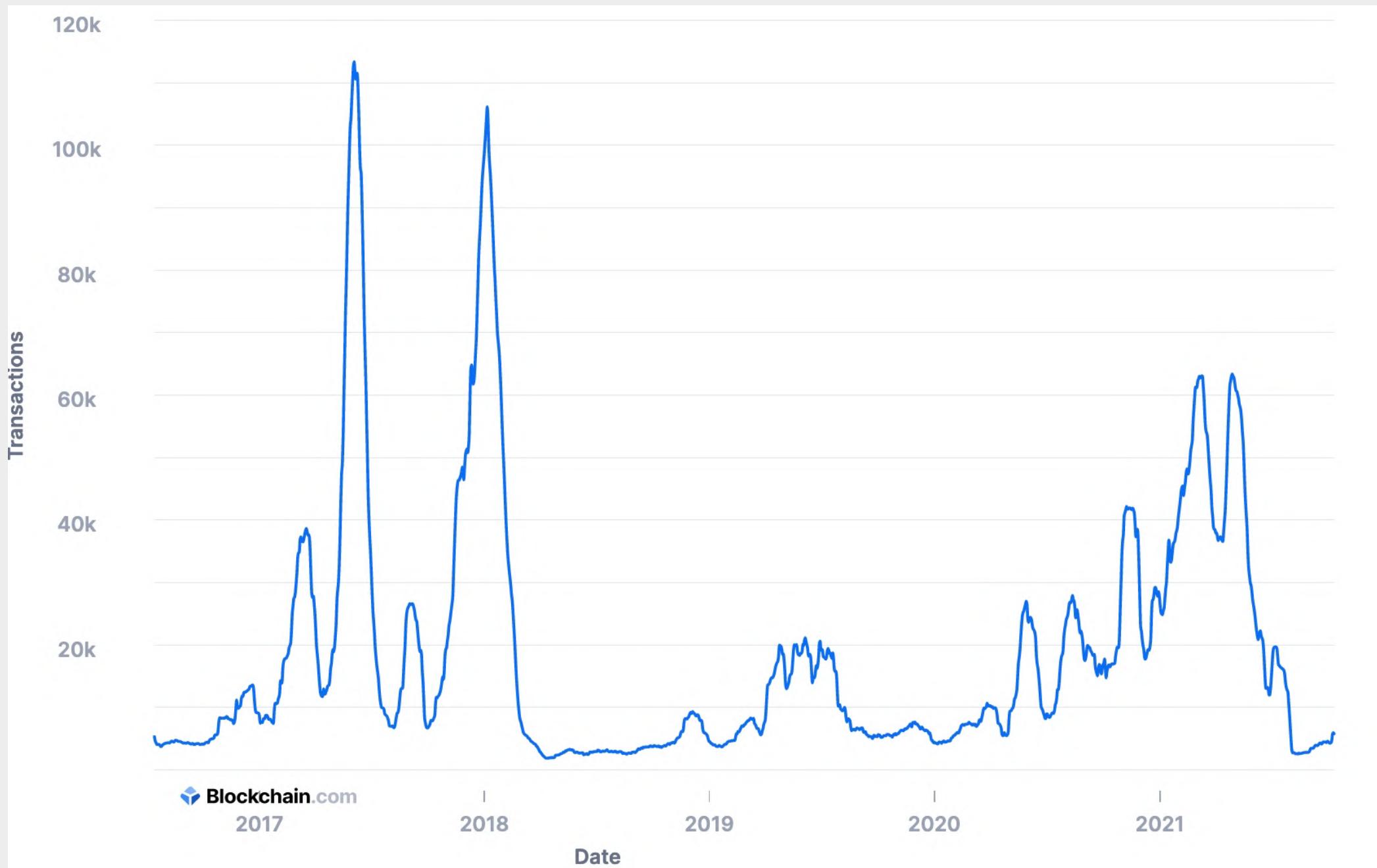


Taille moyenne du « mempool » (nombre de transactions)

Le nombre de transactions en attente varie également fortement au cours du temps

Les trajectoires de ces variations ne sont pas tout à fait superposables à celles de la taille du mempool en MB car ici la nature ou le contenu des transactions en attente n'importe pas

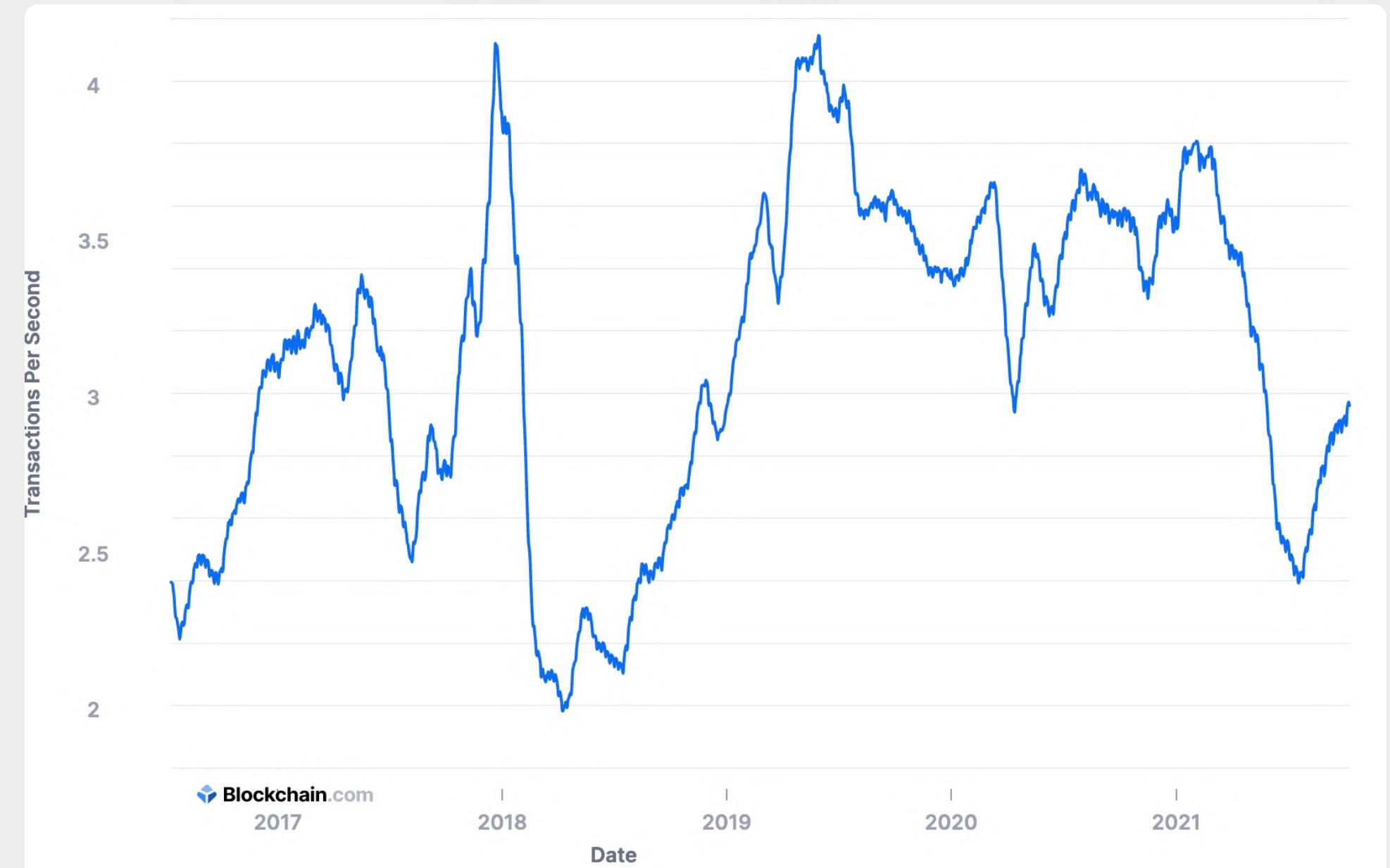
La taille du mempool en nombre de transactions augmente lorsque le réseau est très demandé i.e. lors des pics de prix



Nombre moyen de transactions entrantes / seconde

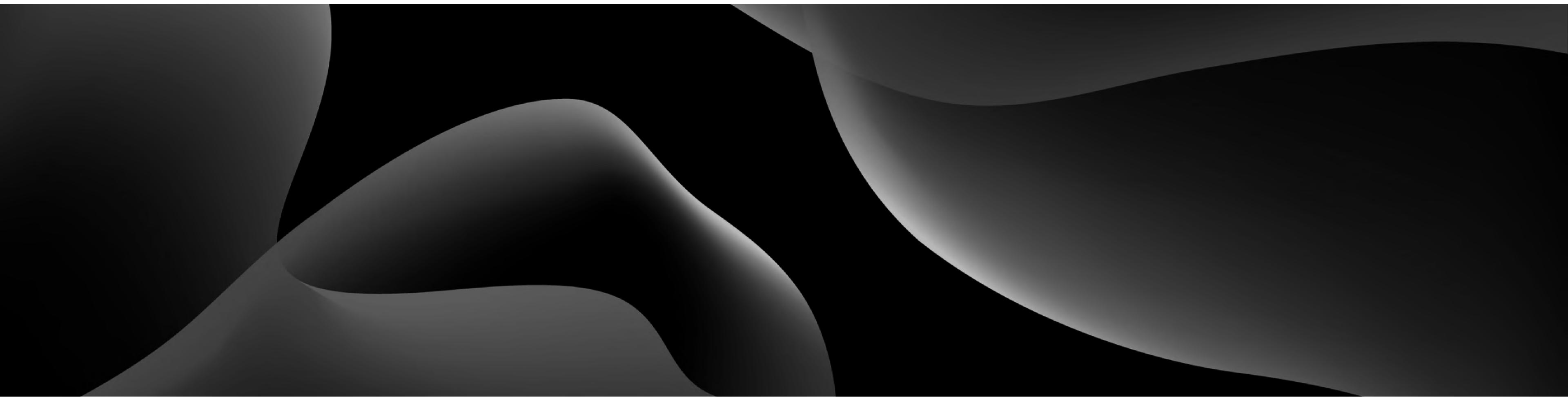
Cet indicateur est un, digne reflet de l'attraction pour Bitcoin en tant que « valeur mobilière », beaucoup plus que les « bids » et « asks » des market makers

Les transactions entrantes représentent l'activité réelle de la chaîne : c'est ce qu'on appelle un indicateur « on-chain », par opposition aux indicateurs off-chain (dépôts Github, activité de la communauté sur les réseaux, discussions et arguments forums...) Savoir qui alimente qui dans la dialectique prix/activité revient à déterminer l'origine de l'œuf et de la poule...



06

Explorer les blocs



Explorer les blocs

- Un audit simple peut se faire via un explorateur de blocs (block explorer ou blockchain browser)
- Il n'est pas connecté à Internet (par sécurité) mais agit comme un navigateur web (tri des données et visualisation)
- Il contient toutes les informations d'identification liées à un compte : adresse publique, solde, historique des transactions
- Il contient toutes les informations d'identification liées à une transaction : bloc, montant, émetteur, récepteur

En voici quelques-uns : Bitinfocharts (le meilleur) blockchain.com, bitmain, blockcypher ou Etherscan (le meilleur pour Ethereum)

Historique auditable

\$55,225.32

[Price →](#)

156.244 EH/s

[Estimated Hash Rate →](#)

214,963

[Transactions \(24hrs\) →](#)

3.153m BTC

[Transaction Volume →](#)

40,535 BTC

[Transaction Volume \(Est\) →](#)



ETHER PRICE

\$3,564.15 @ 0.06462 BTC (-0.94%)



MARKET CAP

\$419,355,137,857.00



TRANSACTIONS

1,313.62 M (14.1 TPS)



DIFFICULTY

9,487.21 TH

MED GAS PRICE

41 Gwei (\$3.07)

HASH RATE

728,971.30 GH/s

ETHEREUM TRANSACTION HISTORY IN 14 DAYS

1 360k

1 040k

Sep 25

Oct 2



Oct 9

Bitcoin

Blockchain information for Bitcoin (BTC) including historical prices, the most recently mined blocks, the mempool size of unconfirmed transactions, and data for the latest transactions.

\$55,225.32[Price →](#)**156.244 EH/s**[Estimated Hash Rate →](#)**214,963**[Transactions \(24hrs\) →](#)**3.153m BTC**[Transaction Volume →](#)**40,535 BTC**[Transaction Volume \(Est\) →](#)

Price

The price of Bitcoin over the last day

1 Day ▾

[View All Prices →](#)

Mempool Size (Bytes)

The aggregate size of unconfirmed transactions in bytes

1 Day ▾

[View All Charts →](#)

The Ethereum Blockchain Explorer

All Filters ▾

Search by Address / Txn Hash / Block / Token / Ens



Featured: Bridging tokens between Ethereum, Layer 2 and other chains? Browse through the Blockscan [bridges list](#).

AX

Crypto savings with
60% APY
Earn while you sleep



ETHER PRICE
 \$3,564.15 @ 0.06462 BTC (-0.94%)

MARKET CAP
 \$419,355,137,857.00

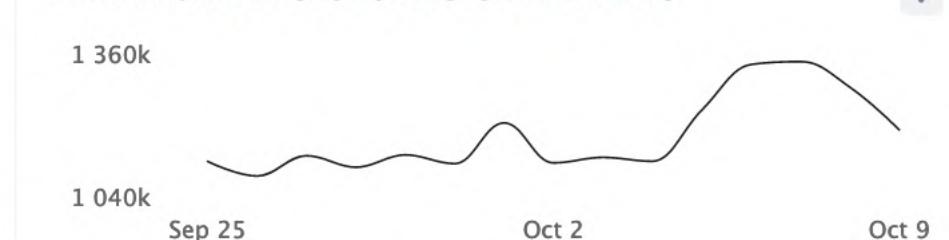
TRANSACTIONS
 1,313.62 M (14.1 TPS)

DIFFICULTY
 9,487.21 TH

MED GAS PRICE
41 Gwei (\$3.07)

HASH RATE
728,971.30 GH/s

ETHEREUM TRANSACTION HISTORY IN 14 DAYS



Latest Blocks

Bk	13390766 23 secs ago	Miner Ethermine 130 txns in 16 secs	2.03062 Eth
Bk	13390765 39 secs ago	Miner BeePool 311 txns in 5 secs	2.26539 Eth
Bk	13390764 44 secs ago	Miner 0x4069e799da927c06b4... 16 txns in 8 secs	2.00836 Eth
Bk	13390763 52 secs ago	Miner Ethermine 196 txns in 10 secs	2.0971 Eth
Bk	13390762 1 min ago	Miner Nanopool 73 txns in 2 secs	2.04533 Eth
Bk	13390761 1 min ago	Miner 2Miners: PPLNS 94 txns in 18 secs	2.03718 Eth

[View all blocks](#)

Latest Transactions

Tx	0xbbd9f581c66d... 23 secs ago	From 0xe5e2216ed8d9296015... To 0x0f4b28d46cab209bc5f...	0 Eth
Tx	0xb1c2343f8d4e... 23 secs ago	From 0xf162e56189cbeefe99c... To 0x5ebdec217daaa912fb...	0.1 Eth
Tx	0x5dd382af6c9c6... 23 secs ago	From 0x222154a054cc377ff98f... To 0x495f947276749ce646f...	0 Eth
Tx	0x65b8152a108f... 23 secs ago	From 0x84cf5911a10e7a3c9c8... To 0x7be8076f4ea4a4ad08...	0.24 Eth
Tx	0x68e951633ff5a... 23 secs ago	From 0x06edd8a506623c6b6b... To 0x837704ec8dfec19878...	0 Eth
Tx	0xe67c412c2cccf... 23 secs ago	From 0x06edd8a506623c6b6b... To 0xa5409ec958c83c3f309...	0 Eth

[View all transactions](#)

Latest Blocks

Bk	13390766 23 secs ago	Miner Ethermine 130 txns in 16 secs	2.03062 Eth
Bk	13390765 39 secs ago	Miner BeePool 311 txns in 5 secs	2.26539 Eth
Bk	13390764 44 secs ago	Miner 0x4069e799da927c06b4... 16 txns in 8 secs	2.00836 Eth
Bk	13390763 52 secs ago	Miner Ethermine 196 txns in 10 secs	2.0971 Eth
Bk	13390762 1 min ago	Miner Nanopool 73 txns in 2 secs	2.04533 Eth
Bk	13390761 1 min ago	Miner 2Miners: PPLNS 94 txns in 18 secs	2.03718 Eth

[View all blocks](#)

Latest Blocks

The most recently mined blocks

Height	Mined	Miner	Size
704371	1 minute	Unknown	1,549,250 bytes
704370	5 minutes	Unknown	1,202,906 bytes
704369	34 minutes	Unknown	83,317 bytes
704368	35 minutes	AntPool	590,380 bytes
704367	35 minutes	ViaBTC	1,396,266 bytes
704366	56 minutes	Unknown	1,458,391 bytes

[View All Blocks →](#)

Latest Transactions

The most recently published unconfirmed transactions

Hash	Time	Amount (BTC)	Amount (USD)
0fb769ee2a407817dd515...	13:57	0.03500804 BTC	\$1,933.33
1583a6b7127947f5be3b9...	13:57	0.15974536 BTC	\$8,821.99
155e1a74d2736361c1d70...	13:57	0.00521424 BTC	\$287.96
100f195c1a6f7f19de9e28...	13:57	0.00127215 BTC	\$70.25
00779f36d2938ef47af4d...	13:57	0.01792202 BTC	\$989.75
4b0962723496ca2b1622...	13:57	0.00224511 BTC	\$123.99

[View All Transactions →](#)

Latest Transactions

Tx	0xbbd9f581c66d... 23 secs ago	From 0xe5e2216ed8d9296015... To 0x0f4b28d46cab209bc5f...	0 Eth
Tx	0xb1c2343f8d4e... 23 secs ago	From 0xf162e56189cbeefe99c... To 0x5ebdec217daaa912fb...	0.1 Eth
Tx	0x5dd382af6c9c6... 23 secs ago	From 0x222154a054cc377ff98f... To 0x495f947276749ce646f...	0 Eth
Tx	0x65b8152a108f... 23 secs ago	From 0x84cf5911a10e7a3c9c8... To 0x7be8076f4ea4a4ad08...	0.24 Eth
Tx	0x68e951633ff5a... 23 secs ago	From 0x06edd8a506623c6b6b... To 0x837704ec8dfec19878...	0 Eth
Tx	0xe67c412c2cccf... 23 secs ago	From 0x06edd8a506623c6b6b... To 0xa5409a9a58a83a2f30a	0 Eth

[View all transactions](#)

Block 704374 ⓘ

USD BTC

This block was mined on October 10, 2021 at 2:01 PM GMT+2 by [Poolin](#). It currently has 1 confirmations on the Bitcoin blockchain.

The miner(s) of this block earned a total reward of 6.25000000 BTC (\$345,158.25). The reward consisted of a base reward of 6.25000000 BTC (\$345,158.25) with an additional 0.05010005 BTC (\$2,766.79) reward paid as fees of the 907 transactions which were included in the block. The Block rewards, also known as the Coinbase reward, were sent to this [address](#).

A total of 1,880.91784931 BTC (\$103,874,290.12) were sent in the block with the average transaction being 2.07377933 BTC (\$114,525.13). Learn more about [how blocks work](#).

Hash	00000000000000000000000000000000e72b17f844c159764000d102e97b18c05ff056d19ebf
Confirmations	1
Timestamp	2021-10-10 14:01
Height	704374
Miner	Poolin
Number of Transactions	907
Difficulty	19,893,045,048,575.13
Merkle root	f78683bdeee882037523634ff6d2d491c6f19f1b09748fcfc79795cd16483256
Version	0x20000004
Bits	386,803,250
Weight	1,844,197 WU
Size	724,168 bytes
Nonce	1,264,397,938
Transaction Volume	1880.91784931 BTC
Block Reward	6.25000000 BTC
Fee Reward	0.05010005 BTC

Block #13390766

Featured: Review and revoke dApp access to your tokens with our [Token Approvals tool!](#)

Overview Comments

② Block Height: **13390766**  

② Timestamp:  10 mins ago (Oct-10-2021 11:58:03 AM +UTC)

② Transactions:  and  in this block

② Mined by:  (**Ethermine**) in 16 secs

② Block Reward: 2.030624261098272172 Ether (2 + 0.312892772838858284 - 0.282268511740586112)

② Uncles Reward: 0

② Difficulty: 9,342,247,687,920,115

② Total Difficulty: 32,152,403,370,433,632,688,304

② Size: 30,381 bytes

② Gas Used: 6,399,466 (21.33%)  **-57% Gas Target**

② Gas Limit: 30,000,000

② Base Fee Per Gas: 0.000000044108135232 Ether (44.108135232 Gwei)

② Burnt Fees:  0.282268511740586112 Ether

② Extra Data: ethermine-asia-east2 (Hex:0x65746865726d696e652d617369612d6561737432)

② Hash: 0x0f0608d967d9070cc91efa460d9c012e76f1140177581706ba49323c4fab26ac

② Parent Hash:  

② Sha3Uncles: 0x1dcc4de8dec75d7aab85b567b6cccd41ad312451b948a7413f0a142fd40d49347

② StateRoot: 0x9b1caa301f9e39e52dc8b25667e8deea7d75b20e339a7a6d0bb5fb06ded4aa4f

② Nonce: 0x01a377cca88cbf42

Summary

USD BTC

Fee	0.00100000 BTC (392.157 sat/B - 98.039 sat/WU - 255 bytes)	0.13167615 BTC
Hash	7bc15724f52fe25d3a3ff50141d83007e59b203004c843f098b85477... 	2021-10-10 14:00
	12cgpFdJViXbwHbhrA3TuW1EGnL25Zqc3P	0.13267615 BTC  
		356y8oSNPCT8mwKkFoVDvCfvGECe71rncJ 
		33C1DweA2hdqD5ofR5zX98itLAiU3PnnkT 
		17A16QmavnUfcW11DAApiJxp7ARnxN5pGX 

This transaction was first broadcast to the Bitcoin network on October 10, 2021 at 2:00 PM GMT+2. The transaction currently has 15 confirmations on the network. At the time of this transaction, 0.13167615 BTC was sent with a value of \$7,271.86. The current value of this transaction is now \$7,271.86. Learn more about [how transactions work](#).

Details

Hash	7bc15724f52fe25d3a3ff50141d83007e59b203004c843f098b85477a9776d94
Status	Confirmed
Received Time	2021-10-10 14:00
Size	255 bytes
Weight	1,020
Included in Block	704374
Confirmations	15
Total Input	0.13267615 BTC
Total Output	0.13167615 BTC
Fees	0.00100000 BTC
Fee per byte	392.157 sat/B
Fee per vbyte	N/A
Fee per weight unit	98.039 sat/WU
Value when transacted	\$7,271.86

Inputs i

HEX **ASM**

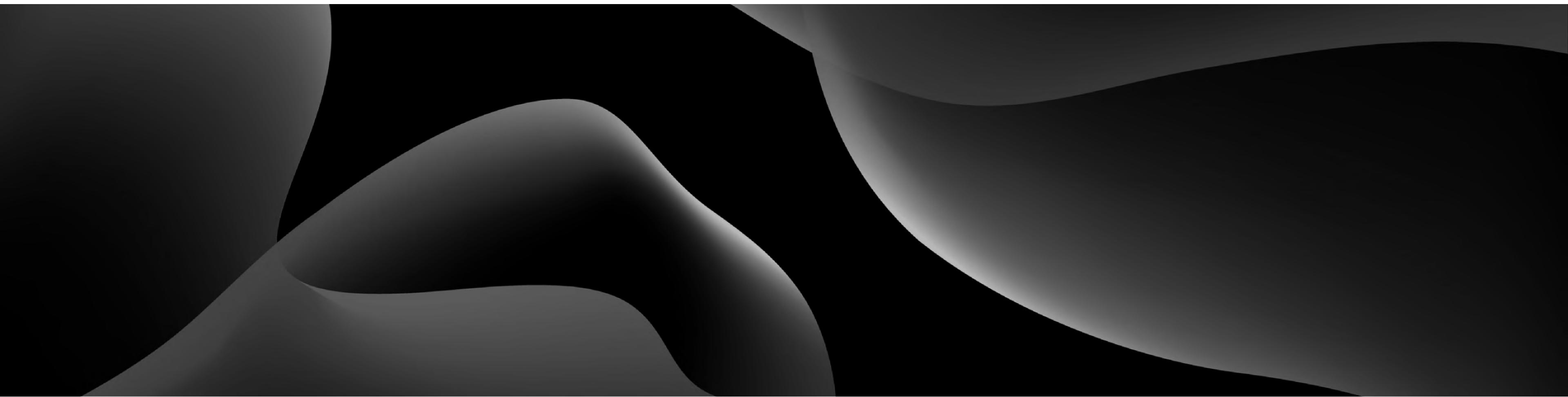
Index	0	Details	Output
Address	12cgpFdJViXbwHbhrA3TuW1EGnL25Zqc3P	Value	0.13267615 BTC
Pkscript	OP_DUP OP_HASH160 11b7eb8a3c1cc8a2a076c8ce916a4f0da3a18ab6 OP_EQUALVERIFY OP_CHECKSIG		
Sigscript	304402201210f2db63d9d8350dec7c0da8de8b7c0b3a245d9eab8df4bf183f068db0e0cc02201dc9c3c6a6157a587b253d91f966dc0554105ccd8520751b28835d0 75341da2201 03a0c53fcc4704ba78331a896c3bd684328b44890b25f91cfb853ab0bb301c7875		
Witness			

Outputs i

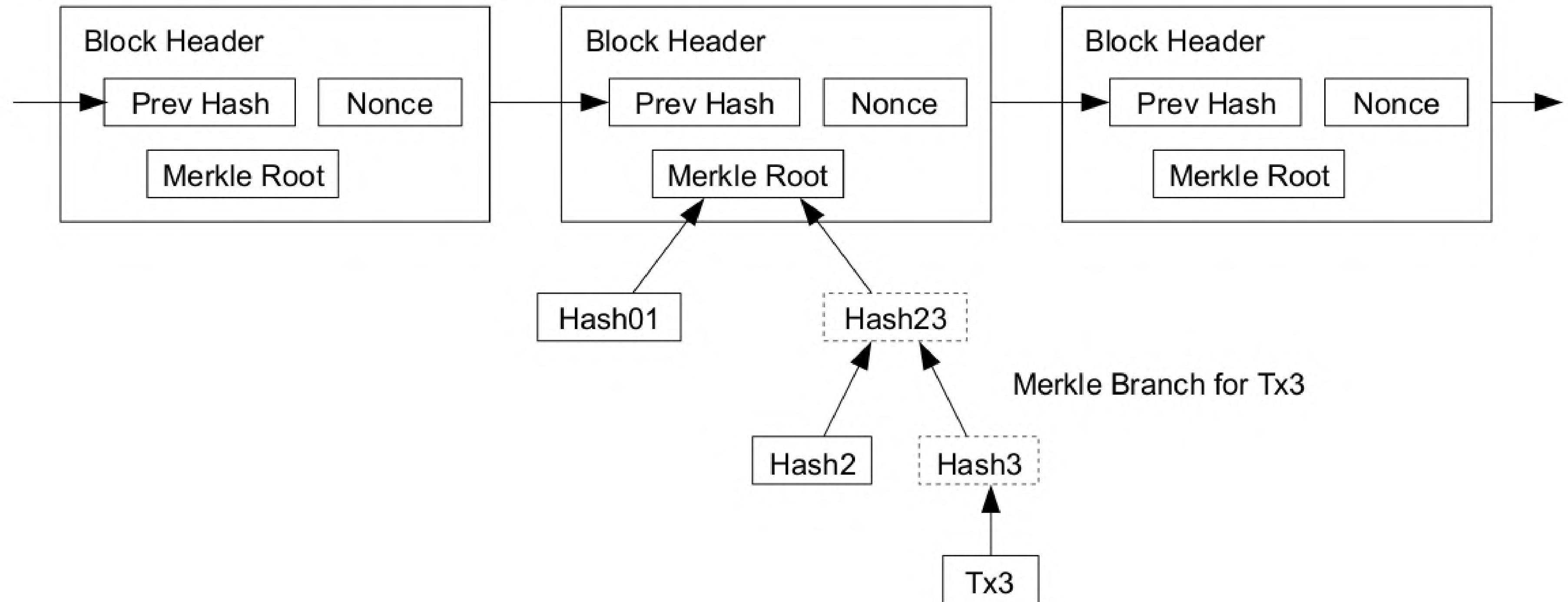
Index	0	Details	Unspent
Address	356y8oSNPCT8mwKkFoVDvCfvGECe71rncJ	Value	0.00606575 BTC
Pkscript	OP_HASH160 25700857076febddcf49c76ad5837561d0ad95d1 OP_EQUAL		
Index	1	Details	Unspent
Address	33C1DweA2hdqD5ofR5zX98itLAiU3PnnkT	Value	0.05450000 BTC
Pkscript	OP_HASH160 10739ab58d185c1bc8098e1f9c21c3f46e8e74f7 OP_EQUAL		
Index	2	Details	Spent
Address	17A16QmavnUfCW11DAApiJxp7ARnxN5pGX	Value	0.07111040 BTC
Pkscript	OP_DUP OP_HASH160 43849383122ebb8a28268a89700c9f723663b5b8 OP_EQUALVERIFY OP_CHECKSIG		

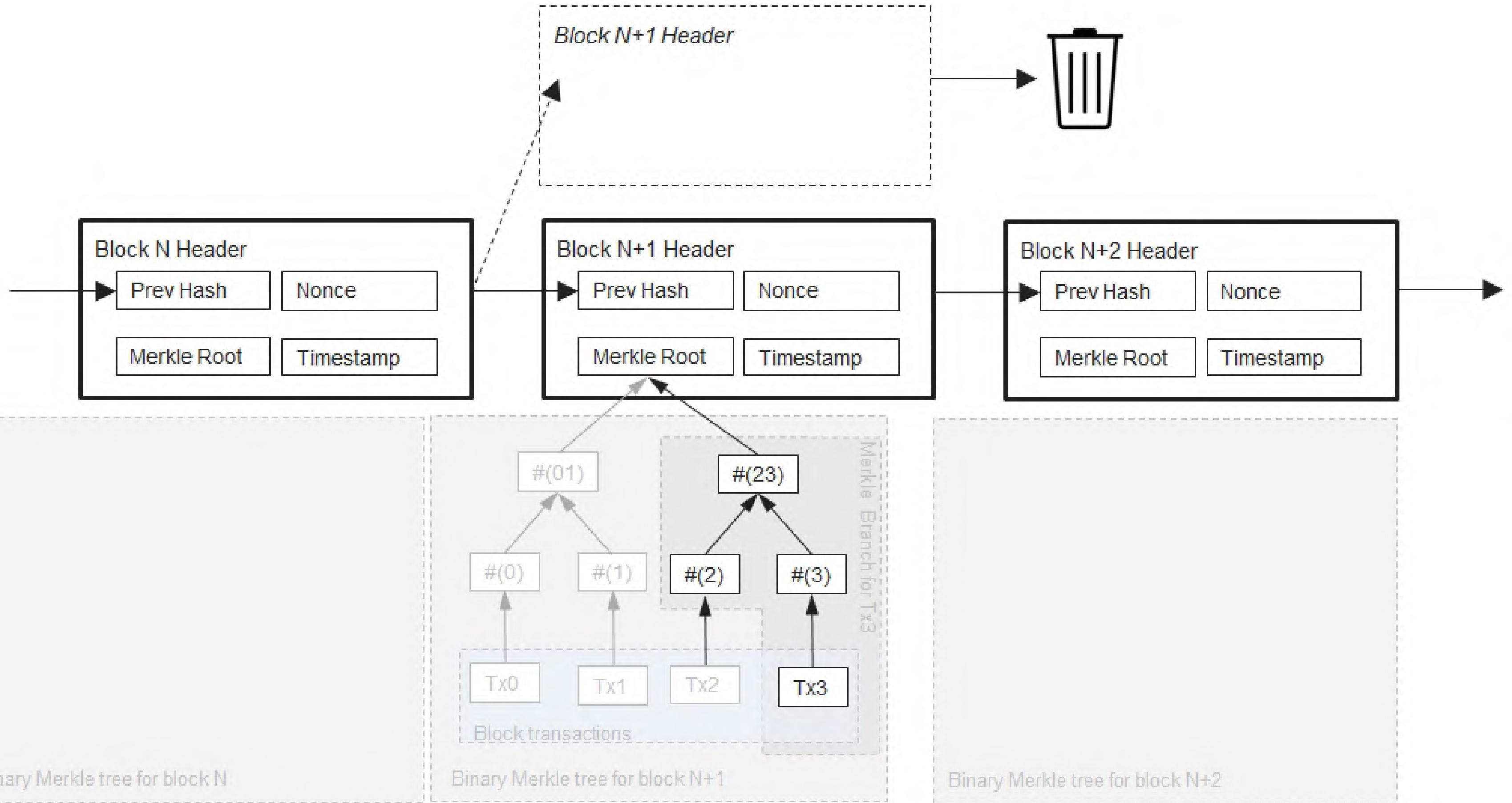
07

La chaine la plus longue prévaut



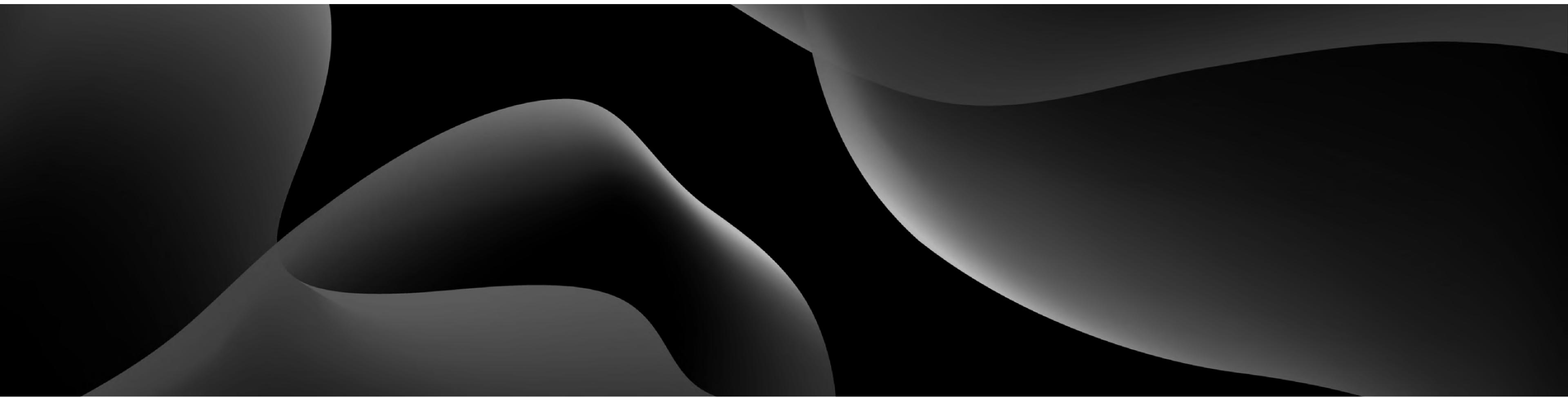
Longest Proof-of-Work Chain





08

Block infos



Les différentes versions des (règles de consensus) des blocs suivies par les clients

Indispensable à connaître pour :

- Processer les transactions
- Assurer les compatibilités
- Sécuriser les blocs
- Auditer la chaîne

Block Versions

- **Version 1** was introduced in the genesis block (January 2009).
- [Version 2](#) was introduced in [Bitcoin Core 0.7.0](#) (September 2012) as a soft fork. As described in [BIP34](#), valid [version 2 blocks](#) require a [block height parameter in the coinbase](#). Also described in [BIP34](#) are rules for rejecting certain blocks; based on those rules, [Bitcoin Core 0.7.0](#) and later versions began to reject [version 2 blocks](#) without the block height in coinbase at block height 224,412 (March 2013) and began to reject new version 1 blocks three weeks later at block height 227,930.
- **Version 3** blocks were introduced in [Bitcoin Core 0.10.0](#) (February 2015) as a soft fork. When the fork reached full enforcement (July 2015), it required strict [DER](#) encoding of all [ECDSA](#) signatures in new blocks as described in [BIP66](#). Transactions that do not use strict [DER](#) encoding had previously been non-standard since [Bitcoin Core 0.8.0](#) (February 2012).
- **Version 4** blocks specified in BIP65 and introduced in [Bitcoin Core 0.11.2](#) (November 2015) as a soft fork became active in December 2015. These blocks now support the new [OP_CHECKLOCKTIMEVERIFY](#) opcode described in that BIP.

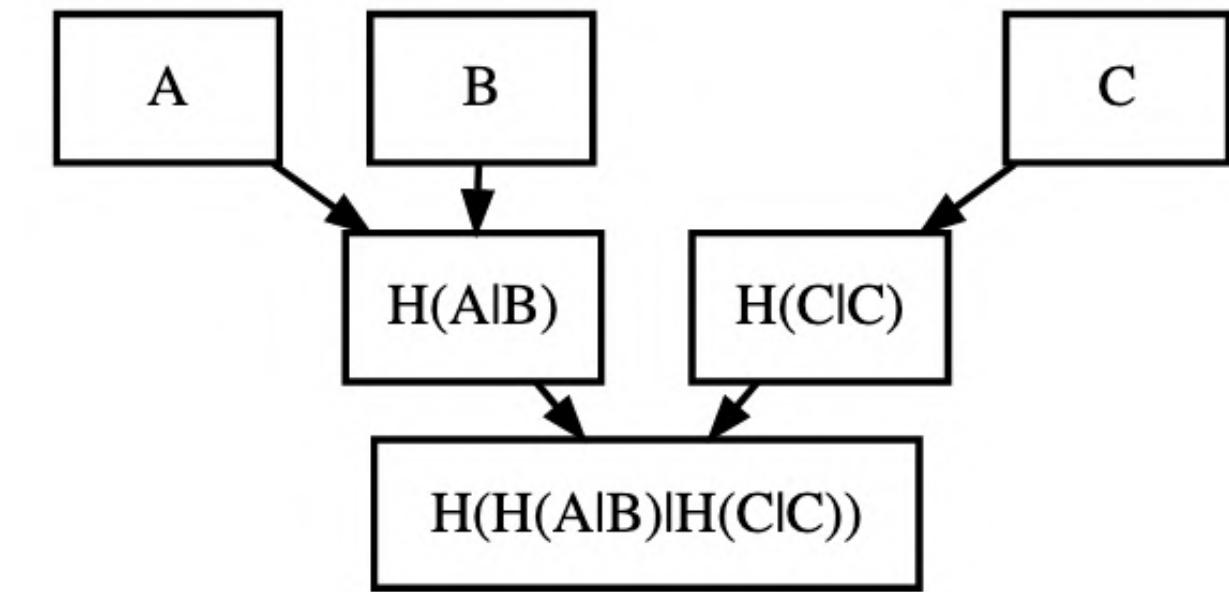
Arbre de Merkle des transactions

- Les transactions sont d'abord hachées une à une en SHA256 : le résultat est appelé transaction ID (TXID)
- Les hashs individuels (TXID) sont ensuite hachés deux à deux par concaténation (mis bout à bout) fixe pour permettre la traçabilité
- Ce processus est répété jusqu'à obtenir un hash « parent » : c'est la racine de Merkle (de l'arbre) des transactions

Row 1: Transaction hashes (TXIDs)
(A is coinbase; C can spend output from B)

Row 2: Hashes of paired TXIDs

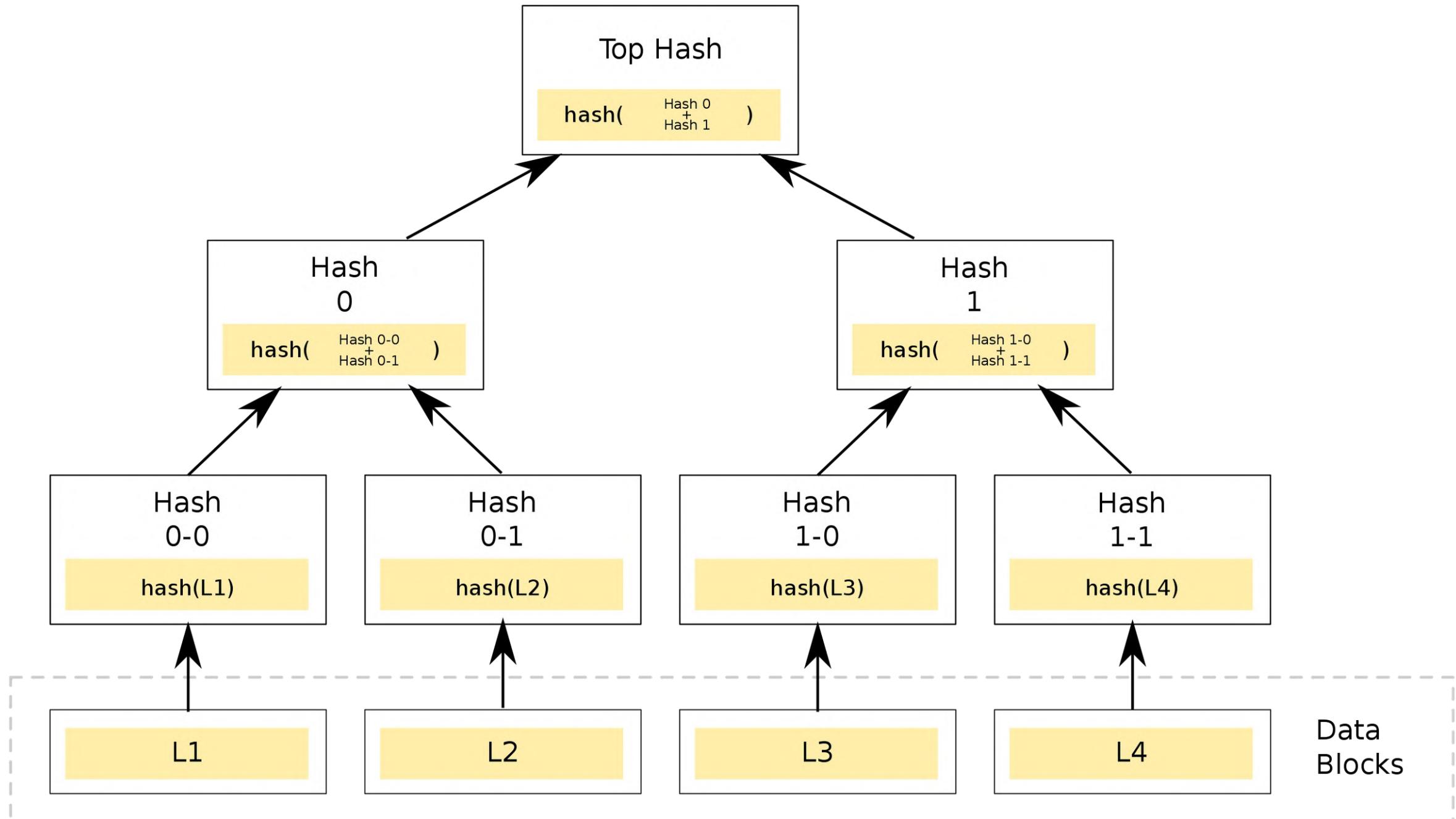
Merkle root



Example Merkle Tree Construction [Hash function $H()$ = SHA256(SHA256())]

Arbre de Merkle des transactions

- Les transactions sont d'abord hachées une à une en SHA256 : le résultat est appelé transaction ID (TXID)
- Les hashes individuels (TXID) sont ensuite hachés deux à deux par concaténation (mis bout à bout) fixe pour permettre la traçabilité
- Ce processus est répété jusqu'à obtenir un hash « parent » : c'est la racine de Merkle (de l'arbre) des transactions



Arbre de Merkle des transactions

- La cible de difficulté (target) est un nombre de 256 bits auquel doit être inférieur le hash du bloc courant
 - Le champ correspondant au sein du bloc header ($nBits$) ne peut contenir que 32 bits, donc la difficulté est encodée en format compacte
 - Il est aisément de retrouver la cible en 256 bits depuis son encodage en 32 bits

$0x181bc330 \rightarrow 0x1bc330 * 256 ^ {(0x18 - 3)}$

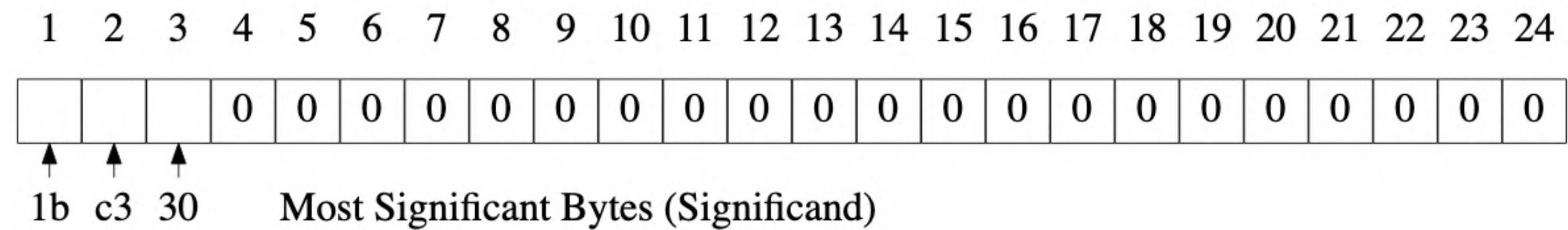
nBits In Big-Endian Order	Significand (Mantissa)	Base	Exponent	Bytes In Significand
---------------------------------	---------------------------	------	----------	----------------------------

Converting nBits Into A Target Threshold

Arbre de Merkle des transactions

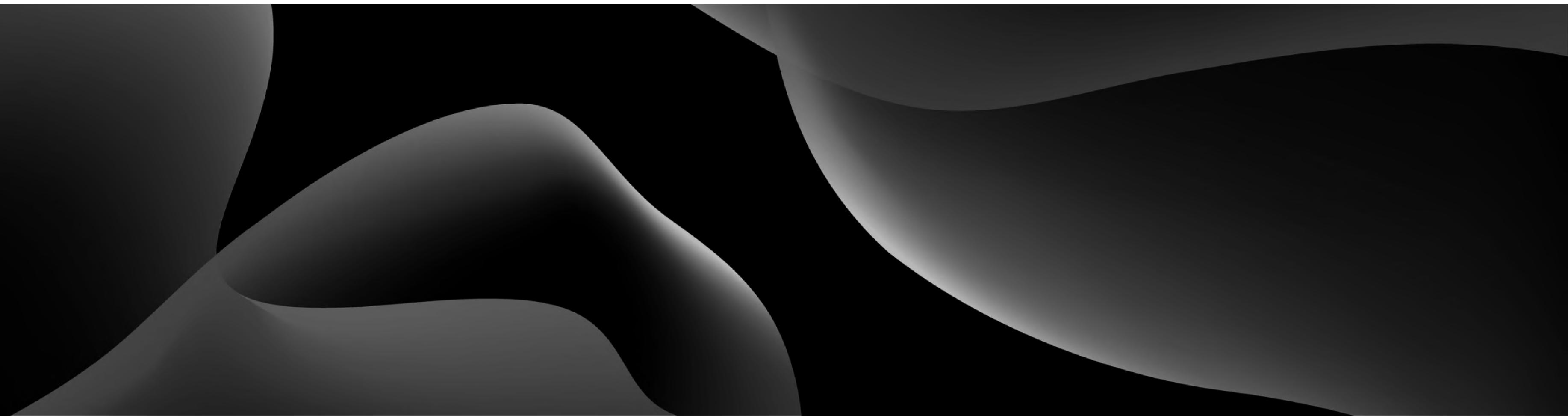
- La cible de difficulté (target) est un nombre de 256 bits auquel doit être inférieur le hash du bloc courant
 - Le champ correspondant au sein du bloc header (nBits) ne peut contenir que 32 bits, donc la difficulté est encodée en format compacte
 - Il est ais  de retrouver la cible en 256 bits depuis son encodage en 32 bits

Byte Length: 0x18 (Decimal 24)



09

Transcrire le script



Transcrire le script

- Le langage script de Bitcoin a besoin, pour s'exécuter, d'une liste de fonctions, à la manière de n'importe quel programme. Ce sont les codes d'opération, les OpCodes
- Bitcoin est fermé et son script est volontairement Turing incomplete, en conséquence les OpCodes sont limités à une liste finie de
- Bitcoin script est un langage procédural qui doit servir de base à une monnaie programmable. Il est inspiré par Forth et fonctionnant comme lui par superposition de piles de données (stacks)
- Retrouvez la liste, l'historique et les explications associées aux codes d'opérations du script de Bitcoin

```
/** Script opcodes */
enum opcodetype
{
    // push value
    OP_0 = 0x00,
    OP_FALSE = OP_0,
    OP_PUSHDATA1 = 0x4c,
    OP_PUSHDATA2 = 0x4d,
    OP_PUSHDATA4 = 0x4e,
    OP_1NEGATE = 0x4f,
    OP_RESERVED = 0x50,
    OP_1 = 0x51,
    OP_TRUE=OP_1,
    OP_2 = 0x52,
    OP_3 = 0x53,
    OP_4 = 0x54,
    OP_5 = 0x55,
    OP_6 = 0x56,
    OP_7 = 0x57,
    OP_8 = 0x58,
    OP_9 = 0x59,
    OP_10 = 0x5a,
    OP_11 = 0x5b,
    OP_12 = 0x5c,
    OP_13 = 0x5d,
    OP_14 = 0x5e,
    OP_15 = 0x5f,
    OP_16 = 0x60,
    // control
    OP_NOP = 0x61,
    OP_VER = 0x62,
    OP_IF = 0x63,
    OP_NOTIF = 0x64,
    OP_VERIF = 0x65,
    OP_VERNOTIF = 0x66,
    OP_ELSE = 0x67,
    OP_ENDIF = 0x68,
    OP_VERIFY = 0x69,
    OP_RETURN = 0x6a,
    // stack ops
    OP_TOALTSTACK = 0x6b,
    OP_FROMALTSTACK = 0x6c,
    OP_2DROP = 0x6d,
    OP_2DUP = 0x6e,
    OP_3DUP = 0x6f,
    OP_2OVER = 0x70,
    OP_2ROT = 0x71,
    OP_2SWAP = 0x72,
    OP_IFDUP = 0x73,
    OP_DEPTH = 0x74,
    OP_DROP = 0x75,
    OP_DUP = 0x76,
    OP_NIP = 0x77,
    OP_OVER = 0x78,
    OP_PICK = 0x79,
    OP_ROLL = 0x7a,
    OP_ROT = 0x7b,
    OP_SWAP = 0x7c,
    OP_TUCK = 0x7d,
    // splice ops
    OP_CAT = 0x7e,
    OP_SUBSTR = 0x7f,
    OP_LEFT = 0x80,
    OP_RIGHT = 0x81,
    OP_SIZE = 0x82,
    // bit logic
    OP_INVERT = 0x83,
    OP_AND = 0x84,
    OP_OR = 0x85,
    OP_XOR = 0x86,
    OP_EQUAL = 0x87,
    OP_EQUALVERIFY = 0x88,
    OP_RESERVED1 = 0x89,
    OP_RESERVED2 = 0x8a,
    // numeric
    OP_1ADD = 0x8b,
    OP_1SUB = 0x8c,
    OP_2MUL = 0x8d,
    OP_2DIV = 0x8e,
    OP_NEGATE = 0x8f,
    OP_ABS = 0x90,
    OP_NOT = 0x91,
    OP_NOTEQUAL = 0x92,
```

Transcrire le script

- Le langage script de Bitcoin a besoin, pour s'exécuter, d'une liste de fonctions, à la manière de n'importe quel programme. Ce sont les codes d'opération, les OpCodes
- Bitcoin est fermé et son script est volontairement Turing incomplete, en conséquence les OpCodes sont limités à une liste finie de
- Bitcoin script est un langage procédural qui doit servir de base à une monnaie programmable. Il est inspiré par Forth et fonctionnant comme lui par superposition de piles de données (stacks)
- Retrouvez la liste, l'historique et les explications associées aux codes d'opérations du script de Bitcoin

```
OP_ADD = 0x93,  
OP_SUB = 0x94,  
OP_MUL = 0x95,  
OP_DIV = 0x96,  
OP_MOD = 0x97,  
OP_LSHIFT = 0x98,  
OP_RSHIFT = 0x99.  
  
OP_BOOLAND = 0x9a,  
OP_BOOLOR = 0x9b,  
OP_NUMEQUAL = 0x9c,  
OP_NUMEQUALVERIFY = 0x9d,  
OP_NUMNOTEQUAL = 0x9e,  
OP_LESSTHAN = 0x9f,  
OP_GREATERTHAN = 0xa0,  
OP_LESSTHANOREQUAL = 0xa1,  
OP_GREATERTHANOREQUAL = 0xa2,  
OP_MIN = 0xa3,  
OP_MAX = 0xa4,  
  
OP_WITHIN = 0xa5.  
  
// crypto  
OP_RIPEMD160 = 0xa6,  
OP_SHA1 = 0xa7,  
OP_SHA256 = 0xa8,  
OP_HASH160 = 0xa9,  
OP_HASH256 = 0xaa,  
OP_CODESEPARATOR = 0xab,  
OP_CHECKSIG = 0xac,  
OP_CHECKSIGVERIFY = 0xad,  
OP_CHECKMULTISIG = 0xae,  
OP_CHECKMULTISIGVERIFY = 0xaf,  
  
// expansion  
OP_NOP1 = 0xb0,  
OP_CHECKLOCKTIMEVERIFY = 0xb1,  
OP_NOP2 = OP_CHECKLOCKTIMEVERIFY,  
OP_CHECKSEQUENCEVERIFY = 0xb2,  
OP_NOP3 = OP_CHECKSEQUENCEVERIFY,  
OP_NOP4 = 0xb3,  
OP_NOP5 = 0xb4,  
OP_NOP6 = 0xb5,  
OP_NOP7 = 0xb6,  
OP_NOP8 = 0xb7,  
OP_NOP9 = 0xb8,  
OP_NOP10 = 0xb9,  
  
OP_INVALIDOPCODE = 0xff.
```

Crypto

Word	Opcode	Hex	Input	Output	Description
OP_RIPEMD160	166	0xa6	in	hash	The input is hashed using RIPEMD-160.
OP_SHA1	167	0xa7	in	hash	The input is hashed using SHA-1.
OP_SHA256	168	0xa8	in	hash	The input is hashed using SHA-256.
OP_HASH160	169	0xa9	in	hash	The input is hashed twice: first with SHA-256 and then with RIPEMD-160.
OP_HASH256	170	0xaa	in	hash	The input is hashed two times with SHA-256.
OP_CODESEPARATOR	171	0xab	Nothing	Nothing	All of the signature checking words will only match signatures to the data after the most recently-executed OP_CODESEPARATOR.
OP_CHECKSIG	172	0xac	sig pubkey	True / false	The entire transaction's outputs, inputs, and script (from the most recently-executed OP_CODESEPARATOR to the end) are hashed. The signature used by OP_CHECKSIG must be a valid signature for this hash and public key. If it is, 1 is returned, 0 otherwise.
OP_CHECKSIGVERIFY	173	0xad	sig pubkey	Nothing / fail	Same as OP_CHECKSIG, but OP_VERIFY is executed afterward.
OP_CHECKMULTISIG	174	0xae	x sig1 sig2 ... <number of signatures> pub1 pub2 <number of public keys>	True / False	<p>Compares the first signature against each public key until it finds an ECDSA match. Starting with the subsequent public key, it compares the second signature against each remaining public key until it finds an ECDSA match. The process is repeated until all signatures have been checked or not enough public keys remain to produce a successful result. All signatures need to match a public key. Because public keys are not checked again if they fail any signature comparison, signatures must be placed in the scriptSig using the same order as their corresponding public keys were placed in the scriptPubKey or redeemScript. If all signatures are valid, 1 is returned, 0 otherwise.</p> <p>Due to a bug, one extra unused value is removed from the stack.</p>
OP_CHECKMULTISIGVERIFY	175	0xaf	x sig1 sig2 ... <number of signatures> pub1 pub2 ... <number of public keys>	Nothing / fail	Same as OP_CHECKMULTISIG, but OP_VERIFY is executed afterward.

RPC API Reference

Remote Procedure Calls (background RPC)

- Les OpCodes du Script ont également un intérêt pour l'utilisateur final, que ce soit à visée personnelle ou professionnelle
- Des lignes de commandes gérées en API depuis la console RPC d'un client comme Bitcoin Core peuvent être utilisées
- Ils sont utilisés comme source de données on-chain pour de la recherche ou collecte d'informations, dans le cadre d'un audit, par exemple

Blockchain RPCs

- [getbestblockhash](#)
- [getblock](#)
- [getblockchaininfo](#)
- [getblockcount](#)
- [getblockfilter](#)
- [getblockhash](#)
- [getblockheader](#)
- [getblockstats](#)
- [getchaintips](#)
- [getchaintxstats](#)
- [getdifficulty](#)
- [getmempoolancestors](#)
- [getmempooldescendants](#)
- [getmempoolentry](#)
- [getmempoolinfo](#)
- [getrawmempool](#)
- [gettxout](#)
- [gettxoutproof](#)
- [gettxoutsetinfo](#)
- [preciousblock](#)
- [pruneblockchain](#)
- [savemempool](#)
- [scantxoutset](#)
- [verifychain](#)
- [verifytxoutproof](#)

getmempoolinfo

getmempoolinfo

Returns details on the active state of the TX memory pool.

Result

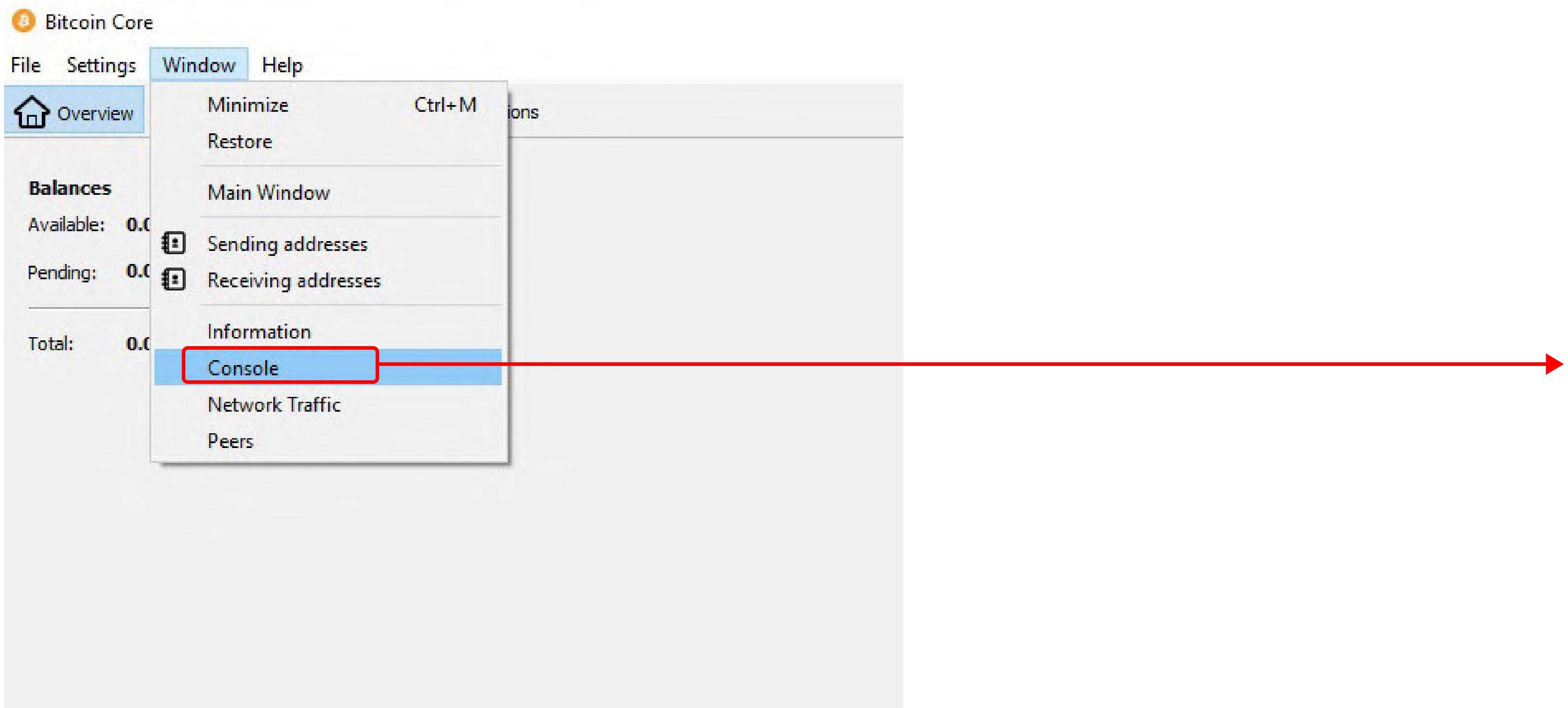
```
{  
    "loaded" : true|false,          (json object)  
    "size" : n,                   (boolean) True if the mempool is fully loaded  
    "bytes" : n,                  (numeric) Current tx count  
    "usage" : n,                  (numeric) Sum of all virtual transaction sizes as defined in  
    "maxmempool" : n,             (numeric) Total memory usage for the mempool  
    "mempoolminfee" : n,          (numeric) Maximum memory usage for the mempool  
    "minrelaytxfee" : n,          (numeric) Minimum fee rate in BTC/kB for tx to be accepted.  
    "unbroadcastcount" : n        (numeric) Current minimum relay fee for transactions  
}
```

Examples

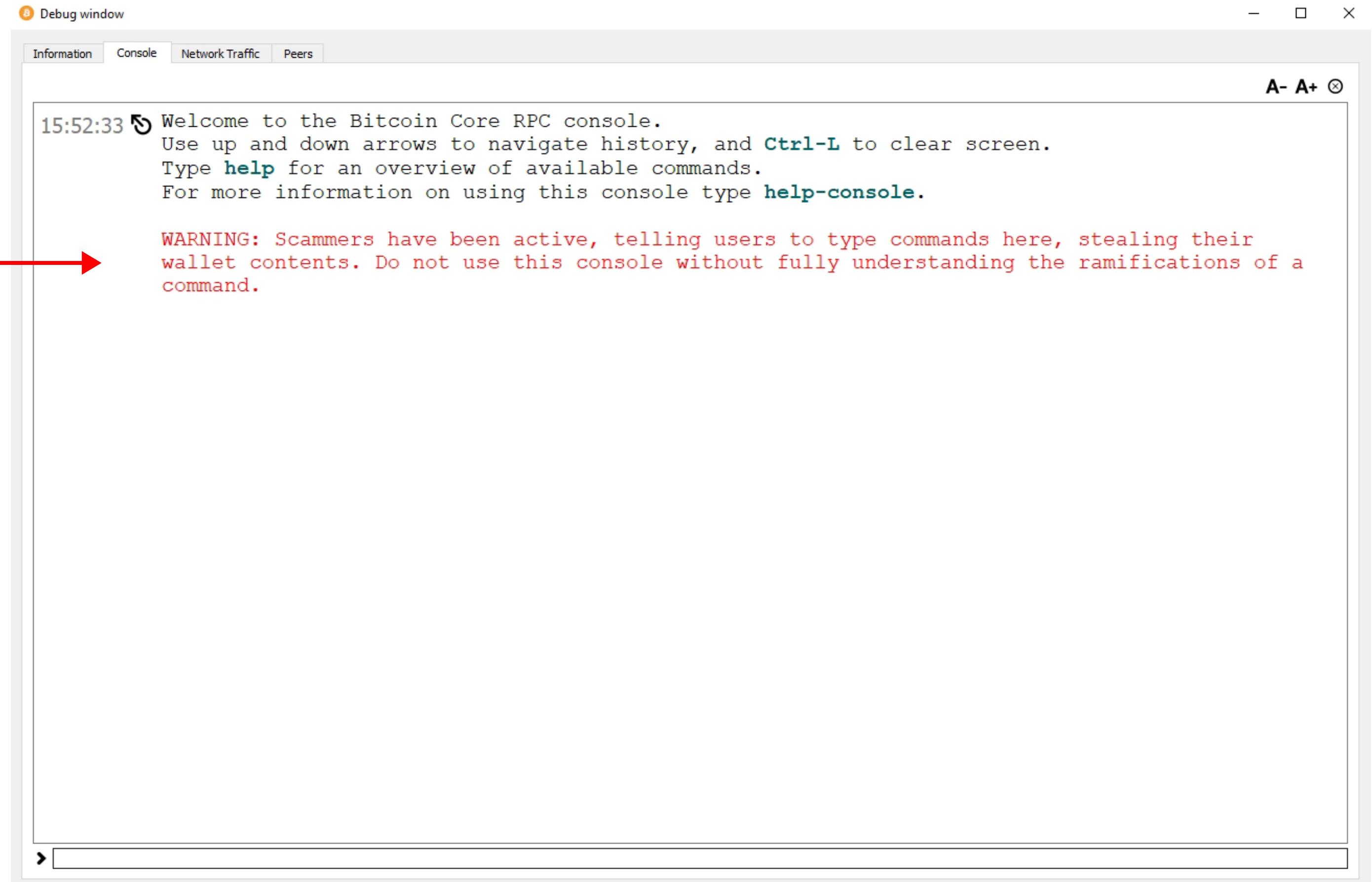
```
bitcoin-cli getmempoolinfo
```

```
curl --user myusername --data-binary '{"jsonrpc": "1.0", "id": "curltest", "method": "get
```

La console RCP



La console RCP

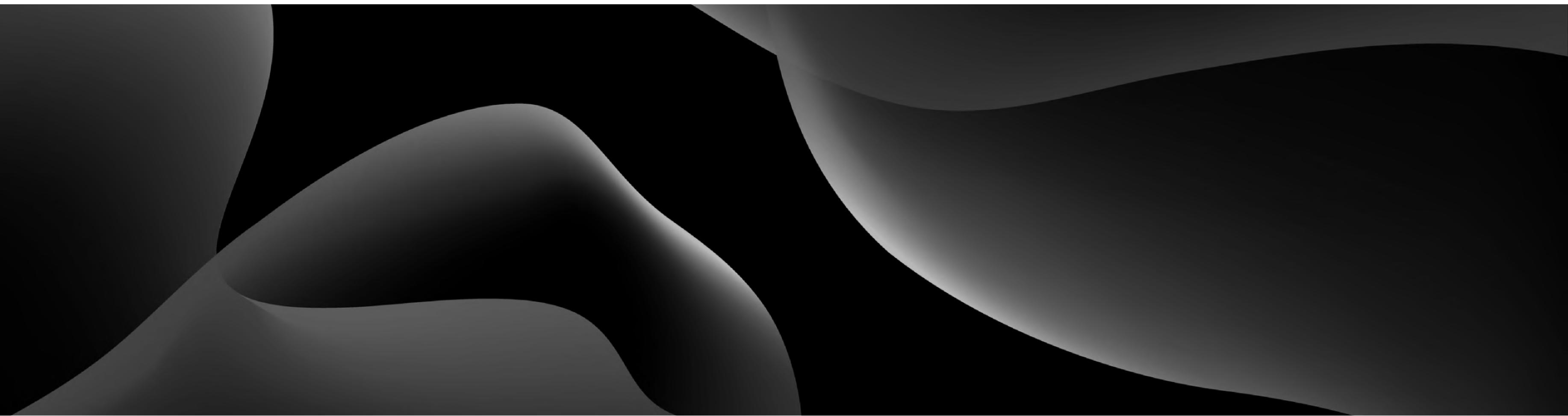


Exemple de commande RCP « getblockchaininfo »

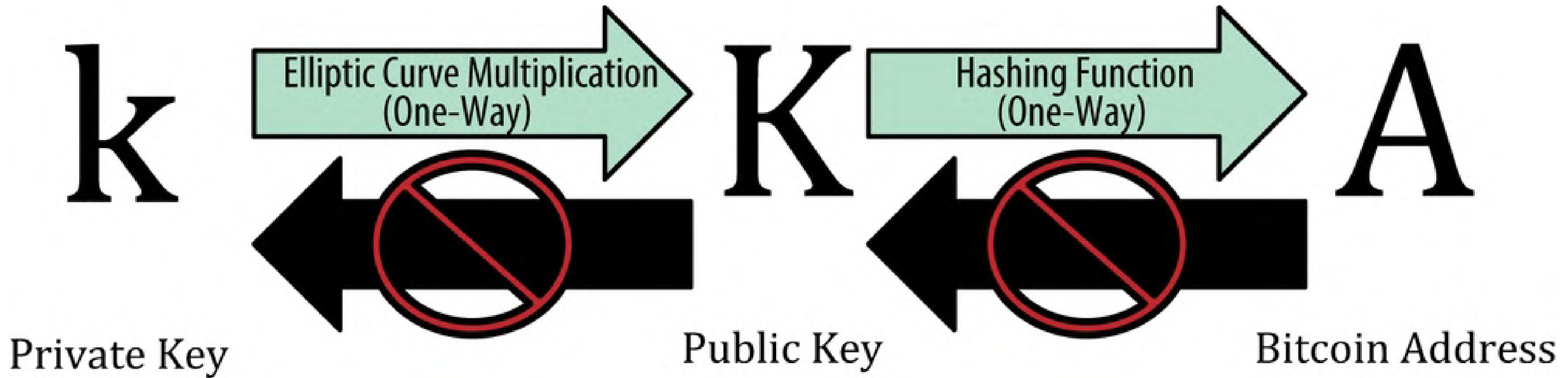
```
{  
    "chain": "main",  
    "blocks": 582101,  
    "headers": 582101,  
    "bestblockhash":  
        "0000000000000000165bf4a8eaa460df4752002840009c29ec0adfd9376406",  
        "difficulty": 7409399249090.253,  
        "mediantime": 1561321207,  
        "verificationprogress": 0.9999922936147396,  
        "initialblockdownload": false,  
        "chainwork":  
            "000000000000000000000000000000000000000000000000000000000000000d23718c9e22adc4275b706",  
            "size_on_disk": 257091100656,  
            "pruned": false,  
            "softforks": [  
                {  
                    "id": "bip34",  
                    "version": 2,  
                    "reject": {  
                        "status": true  
                    }  
                },  
                {  
                    "id": "bip66",  
                    "version": 3,  
                    "reject": {  
                        "status": true  
                    }  
                },  
                {  
                    "id": "bip65",  
                    "version": 4,  
                    "reject": {  
                        "status": true  
                    }  
                }  
            ],  
            "bip9_softforks": {  
                "csv": {  
                    "status": "active",  
                    "startTime": 1462060800,  
                    "timeout": 1493596800,  
                    "since": 419328  
                },  
                "segwit": {  
                    "status": "active",  
                    "startTime": 1479168000,  
                    "timeout": 1510704000,  
                    "since": 481824  
                }  
            },  
            "warnings": ""  
        }  
}
```

10

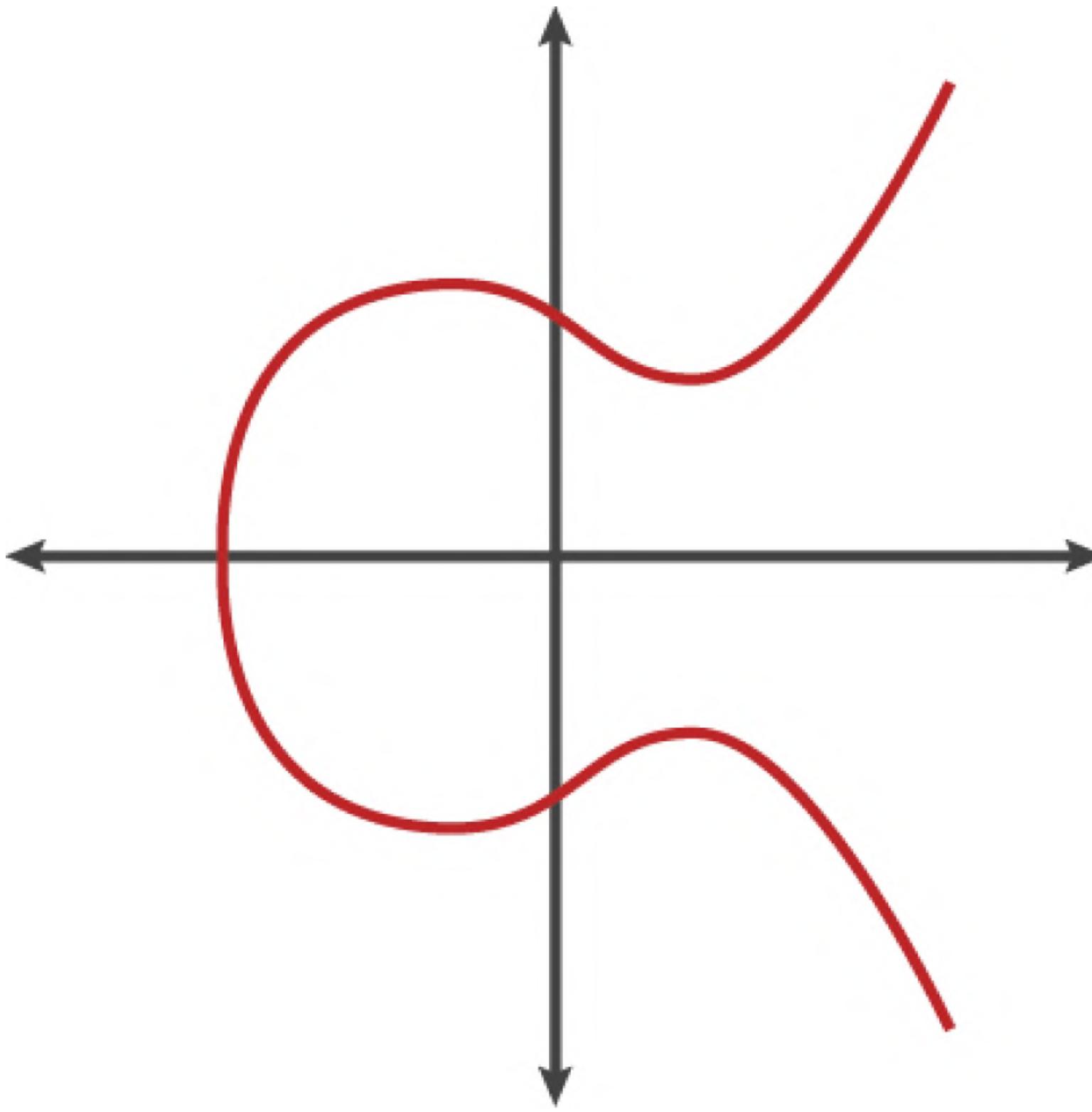
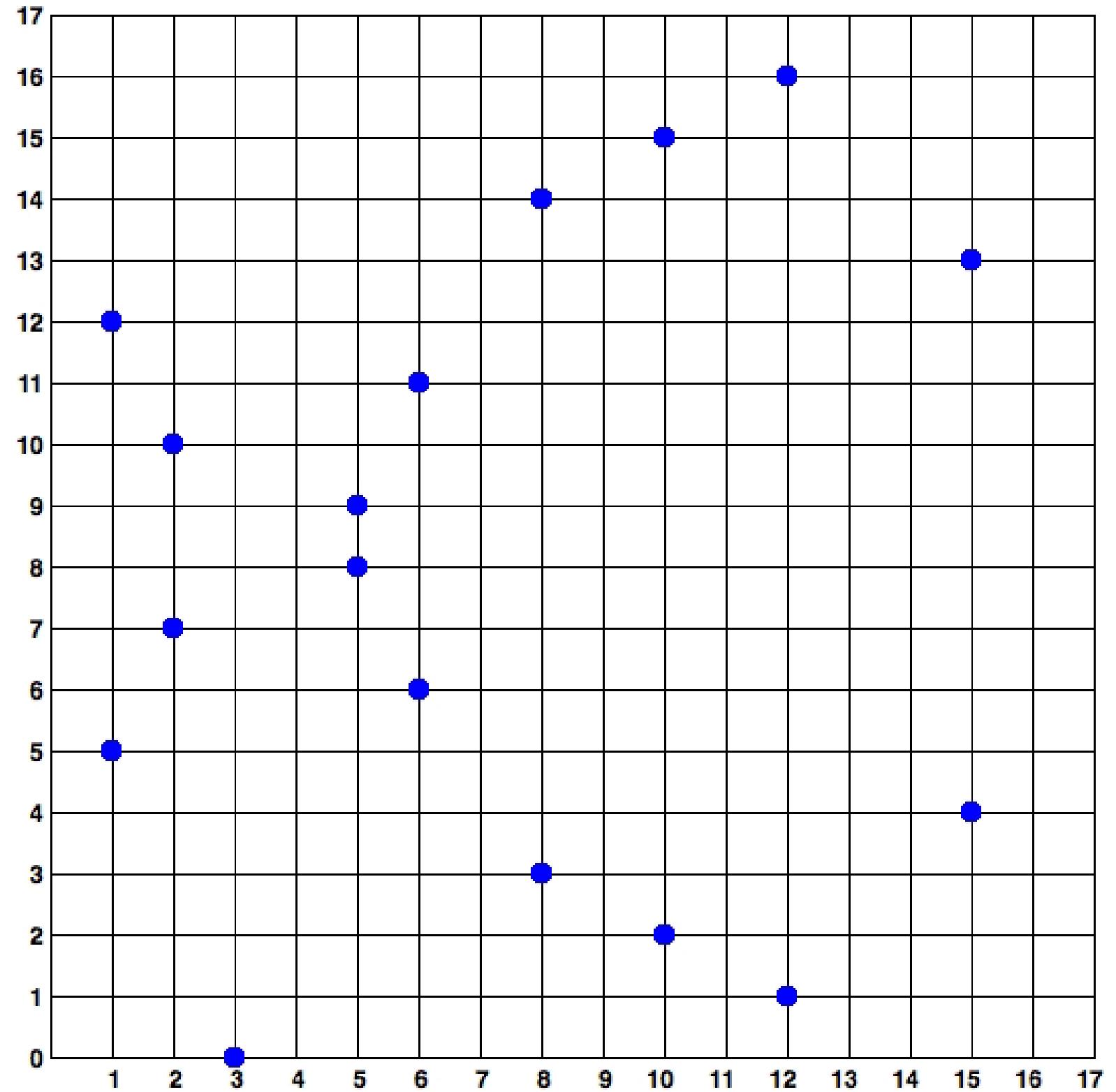
De la clef privée à l'adresse publique



Des fonctions à sens unique

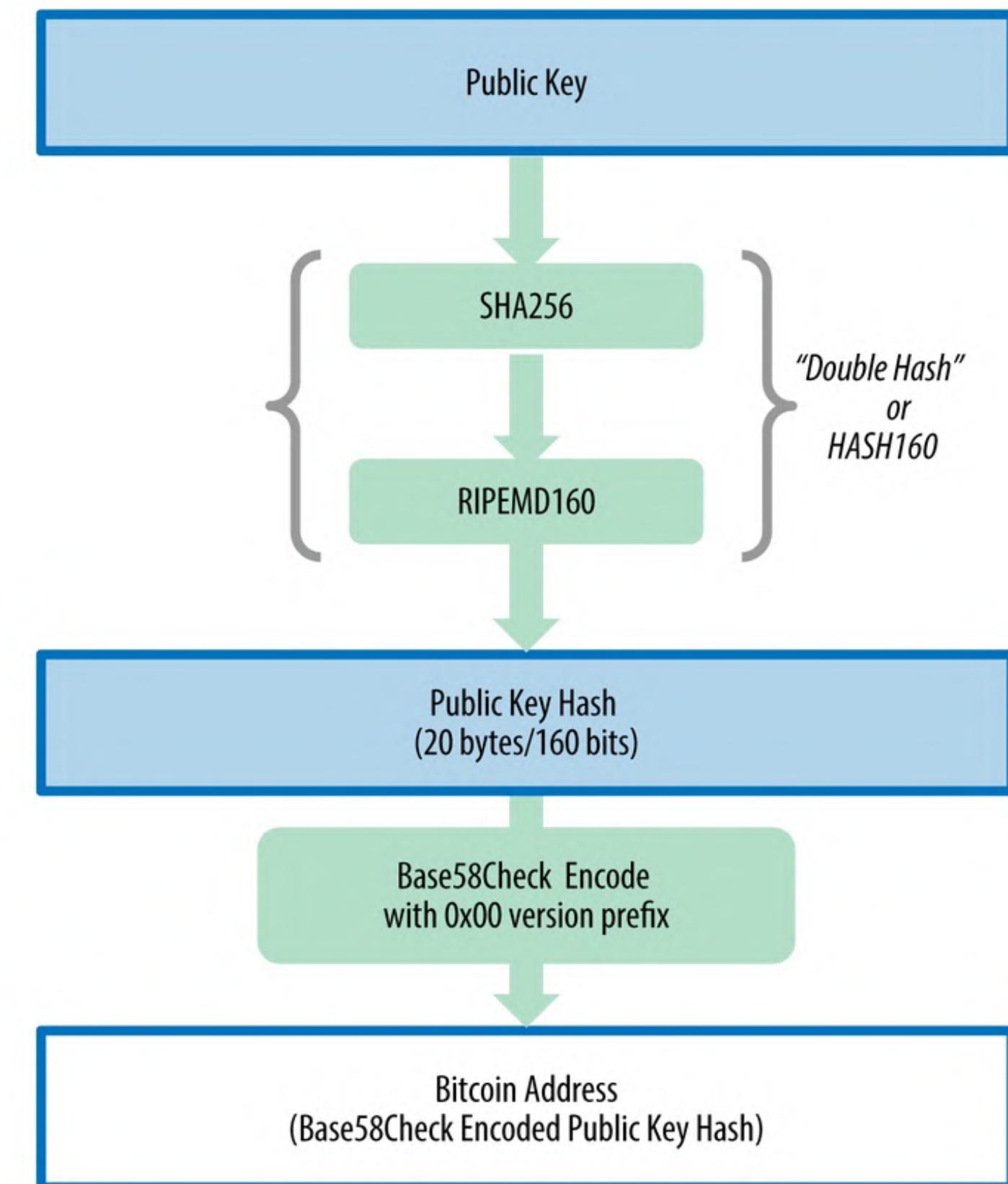


Courbe elliptique & dispersion



De la clef publique à la
génération d'une adresse

Public Key to Bitcoin Address



Différents formats d'encodage

Type	Prefix	Description
Hex	None	64 hexadecimal digits
WIF	5	Base58Check encoding: Base58 with version prefix of 128 and 32-bit checksum
WIF-compressed	K or L	As above, with added suffix 0x01 before encoding
Format	Private Key	
Hex	1E99423A4ED27608A15A2616A2B0E9E52CED330AC530EDCC32C8FFC6A526AEDD	
WIF	5J3mBbAH58CpQ3Y5RNJpUKPE62SQ5tfcvU2JpbnkeyhfsYB1Jcn	
WIF-compressed	KxFC1jmwwCoACiCAWZ3eXa96mBM6tb3TYzGmf6YwgdGWZgawrtJ	

Différents formats d'adresse

Type	Version prefix (hex)	Base58 result prefix
Bitcoin Address	0x00	1
Pay-to-Script-Hash Address	0x05	3
Bitcoin Testnet Address	0x6F	m or n
Private Key WIF	0x80	5, K or L
BIP38 Encrypted Private Key	0x0142	6P
BIP32 Extended Public Key	0x0488B21E	xpub

Exemple de création d'adresse publique à partir d'une clef privée

```
#include <bitcoin/bitcoin.hpp>

int main()
{
    // Private secret key.
    bc::ec_secret secret = bc::decode_hex(
        "038109007313a5807b2ecc082c8c3fbb988a973cacf1a7df9ce725c31b14776");
    // Get public key.
    bc::ec_point public_key = bc::secret_to_public_key(secret);
    std::cout << "Public key: " << bc::encode_hex(public_key) << std::endl;

    // Create Bitcoin address.
    // Normally you can use:
    //   bc::payment_address payaddr;
    //   bc::set_public_key(payaddr, public_key);
    //   const std::string address = payaddr.encoded();

    // Compute hash of public key for P2PKH address.
    const bc::short_hash hash = bc::bitcoin_short_hash(public_key);

    bc::data_chunk unencoded_address;
    // Reserve 25 bytes
    // [ version:1 ]
    // [ hash:20 ]
    // [ checksum:4 ]
    unencoded_address.reserve(25);
    // Version byte, 0 is normal BTC address (P2PKH).
    unencoded_address.push_back(0);
    // Hash data
    bc::extend_data(unencoded_address, hash);
    // Checksum is computed by hashing data, and adding 4 bytes from hash.
    bc::append_checksum(unencoded_address);
    // Finally we must encode the result in Bitcoin's base58 encoding
    assert(unencoded_address.size() == 25);
    const std::string address = bc::encode_base58(unencoded_address);

    std::cout << "Address: " << address << std::endl;
    return 0;
}
```

Le code mnémonique à 12 mots

Entropy input (128 bits)	0c1e24e5917779d297e14d45f14e1a1a
Mnemonic (12 words)	army van defense carry jealous true garbage claim echo media make crunch
Seed (512 bits)	3338a6d2ee71c7f28eb5b882159634cd46a898463e9d2d0980f8e80dfbba5b0fa0291e5fb88 8a599b44b93187be6ee3ab5fd3ead7dd646341b2cdb8d08d13bf7

Le code mnémone à 24 mots

Entropy input (256 bits)	2041546864449caff939d32d574753fe684d3c947c3346713dd8423e74abcf8c
Mnemonic (24 words)	cake apple borrow silk endorse fitness top denial coil riot stay wolf luggage oxygen faint major edit measure invite love trap field dilemma oblige
Seed (512 bits)	3972e432e99040f75ebe13a660110c3e29d131a2c808c7ee5f1631d0a977fcf473bee22 fce540af281bf7cdeade0dd2c1c795bd02f1e4049e205a0158906c343

Mnémone et entropie

Entropy (bits)	Checksum (bits)	Entropy+checksum	Word length
128	4	132	12
160	5	165	15
192	6	198	18
224	7	231	21
256	8	264	24