BNB CHAIN

WORKSHOP

La blockchain dans tous ses états

Principes, fonctionnement des blocs, transactions etc ....

Sommaire

- Qu'est-ce qu'une blockchain
- Qu'est-ce qu'une blockchain EVM
- Transactions & Bloc, Gas, comment ça marche dans le détail
- Principes du proof of Work / Proof of Stake
- Bonus : Les zk rollups ?

Qu'est-ce qu'une Blockchain ?

Qu'est-ce qu'une Blockchain ?

- Un peu d'histoire
  - 1995 : Premier concept de blockchain basée sur la cryptographie publiquement connu, l'objectif est de s'assurer que des documents horodatés ne peuvent pas être falsifiés
  - De 1991 à 2008, de nombreux travaux gravitant autour des concepts actuels de la blockchain sont entrepris par différentes personnes / équipes avec plus ou moins de succès (voir https://bitcoin.fr/Histoire/ pour ceux qui aiment l'histoire)
  - 2008: Satoshi Nakamoto conceptualise la blockchain Bitcoin et publie le white paper associé
  - janvier 2009: Création du bloc genesis de Bitcoin, premier minage d'un bloc Bitcoin
  - 2013 : Première implémentation du proof of stake dans une blockchain (PeerCoin)
  - Juillet 2015 : Lancement de la blockchain Ethereum
  - Septembre 2020: Lancement de la Binance Smart Chain (renommée en BNB Chain récemment)

Qu'est-ce qu'une Blockchain ?

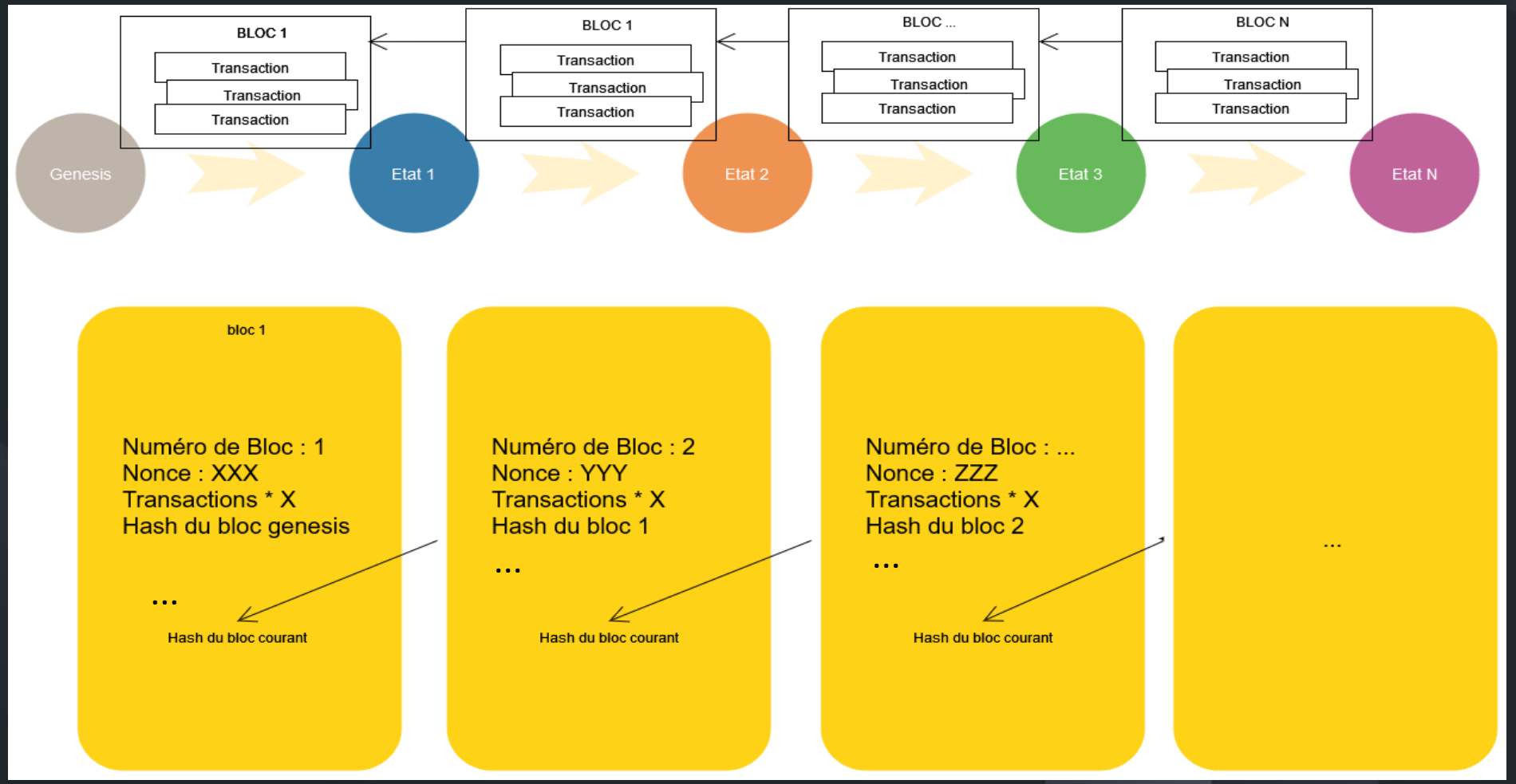
- Deux variantes principales de blockchain:
  - Monnaie digitale only (eg: Bitcoin, Litecoin) => Transfert de valeur
  - Monnaie digitale + smart contracts (eg : Ethereum, BnB Chain) => Transfert de valeur + exécution de code "métier"
- La blockchain est une machine à état dont l'état complet est distribué sur différents ordinateurs à travers le monde (les nodes)
- Un node est en charge de la réception de requêtes de mise à jour de la blockchain, de leur prise en compte (exécution) et de leur diffusion aux autres nodes du réseau (un node est appelé mineur dans une blockchain POW et validateur dans une blockchain POS).

Qu'est-ce qu'une Blockchain ?

- Une requête de mise à jour est appelée une transaction
  - Une transaction est émise par une ou plusieurs adresses (1 dans le cas de BNBChain et 1 ou plusieurs dans le cas de Bitcoin), voir le workshop Wallets pour plus de détails
  - Dans le cas des EVM, une adresse peut être external (adresse d'un account) ou contract (adresse d'un smart contract)
  - Une transaction contient des opérations à exécuter par la blockchain
    - Transfert de monnaie dans le cas de bitcoin par exemple
    - Transfert de monnaie et/ou exécution d'une fonction d'un smart contract sur BNBChain ou Ethereum
  - **Point important**, une adresse de contract ne peut jamais être à l'origine d'une transaction, seules les adresses external peuvent émettre des transactions

Qu'est-ce qu'une Blockchain ?

- Pourquoi parle-t-on d'une chaine de bloc ?



Qu'est-ce qu'une Blockchain ?

- Transfert de valeur
  - Les blockchains publique ont pour la plupart une "monnaie" native qui leur est associée et qui permet un transfert de valeur
  - 1 Bitcoin est subdivisé en sous unités, les satoshis et 1 bitcoin vaut 1^8 satoshis
  - 1 BNB est subdivisé en wei (1 BNB = 10 ^18 wei).
  - L'unité utilisée pour manipuler les monnaies virtuelles d'une blockchain est toujours la plus petite (wei dans le cadre de BNB par exemple).

Qu'est-ce qu'une Blockchain EVM?

Qu'est-ce qu'une Blockchain EVM ?

- EVM signifie Ethereum Virtual Machine. BNB Chain & Ethereum sont des blockchains à base d'EVM.

- L'EVM est une machine virtuelle à état et chaque nœud de minage (en pow) ou de validation (en pos) maintient une copie de l'état de l'EVM (mise à jour à chaque nouveau bloc).

- Le concept d'EVM permet, en plus de la possibilité de transférer de la valeur, d'exécuter du code stocké dans la blockchain, (on appelle ces exécutions "On chain"), on appelle ces codes des **Smart Contracts**

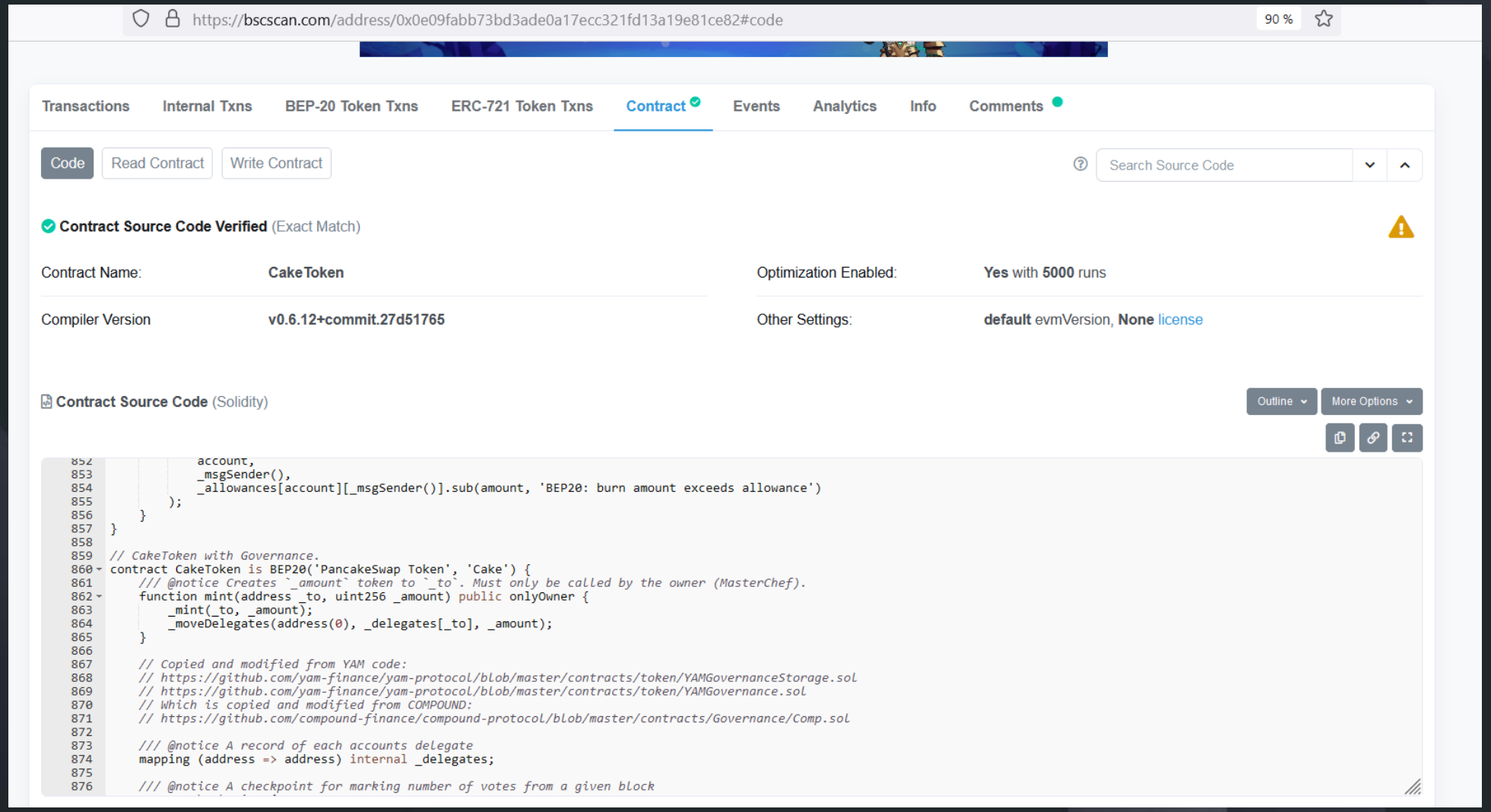
Qu'est-ce qu'une Blockchain EVM ?

- Le code de chaque smart contract est disponible et accessible publiquement (au format compilé ou au format Solidity) sur les explorateurs de chaque blockchain (par exemple, https://bscscan.com pour la BNBChain)

- Le développement de smart contracts se fait au travers du langage de programmation, **Solidity**, spécifiquement créé pour la blockchain EVM

- Chaque opération sur la blockchain (transfert de valeur ou exécution d'un smart contract) entraîne un passage de la blockchain d'un état X a un état X+1. Cet état X+1 est le même sur toutes les copies de l'EVM.

Qu'est-ce qu'une Blockchain EVM ?



Transactions & Bloc, Gas, ... comment ça marche dans le détail dans l'EVM

Transactions, Bloc, Gas ....

La blockchain, une machine à état

Source :https://ethereum.org/en/whitepaper/

Transactions, Bloc, Gas ....

La blockchain, une machine à état : Les merkle Tree au cœur de la blockchain

- En réalité au sein d'une blockchain de type EVM, il existe plusieurs états différents mais cohérents qui permettent d'assurer le fonctionnement & la sécurité

- L'état principal de la blockchain est répliqué sur la totalité des nœuds qui participent à la blockchain, **c'est cet état qui constitue réellement la blockchain** et il contient toutes les informations nécessaires à la sécurisation et la vérification
  - Il est constitué de blocs, composés de certaines données (vu en détail plus loin) et notamment 3 hash

Transactions, Bloc, Gas ....

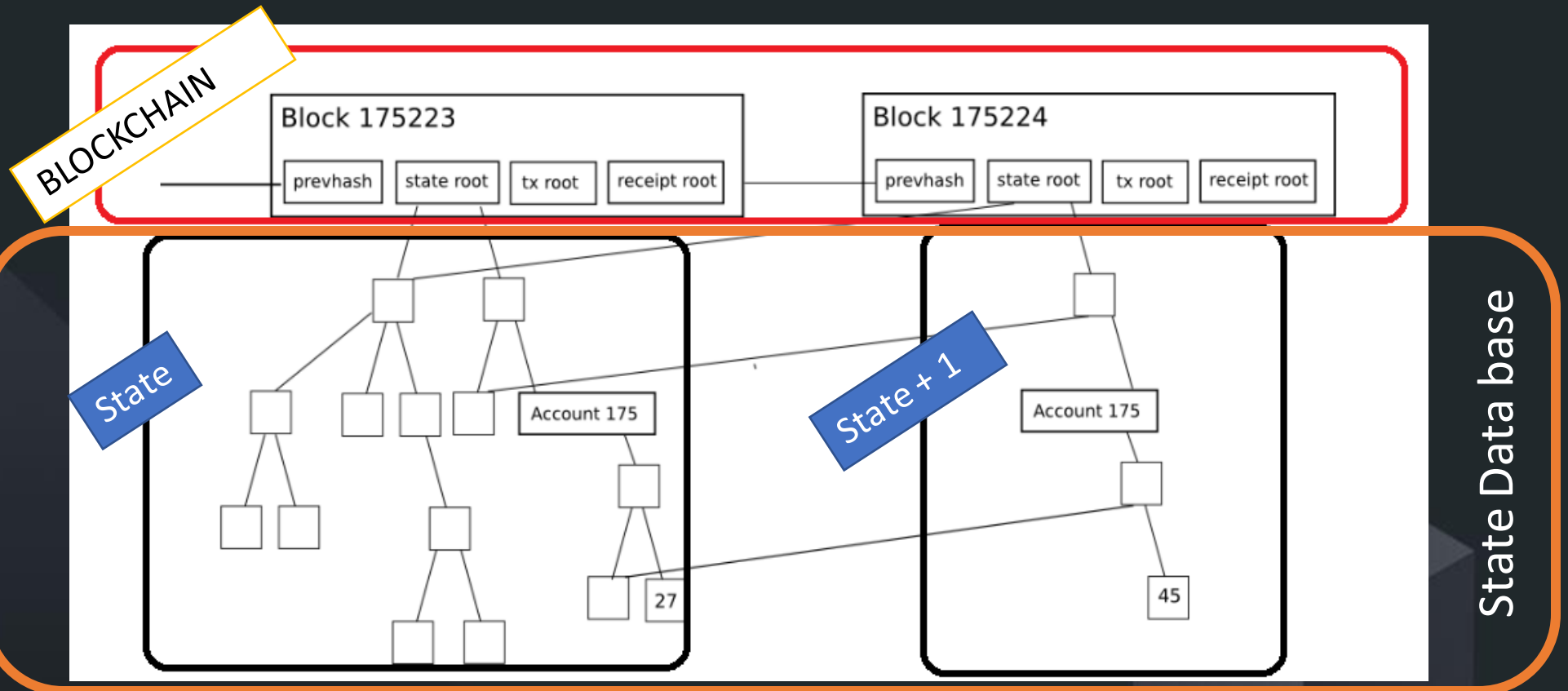
La blockchain, une machine à état : Les merkle Tree au cœur de la blockchain

- Ces hash correspondent aux nœuds parents de Merkle Patricia Tree (Des arbres dont le hash du nœud parent connu et validé publiquement par plusieurs parties et qui permet de vérifier cryptographiquement les valeurs dans les feuilles de l'arbre). Ces arbres sont stockés off chain par les nœuds qui participent à la création de blocs.
  - le state root : hash du nœud parent de l'arbre qui contient le dernier état connu des accounts (notamment la balance d'une adresse après une transaction X).
  - le tx root : hash du nœud parent de l'arbre qui contient les informations sur les transactions d'un bloc
  - Le receipt root : hash du nœud parent de l'arbre qui contient des informations sur l'exécution des transactions (gas utilisé, status code, logs émises)

Transactions, Bloc, Gas ....

La blockchain, une machine à état : Les merkle Tree au cœur de la blockchain

- Pour optimiser l'espace, à chaque bloc on ne stocke que les chemins de l'arbre ayant évolué on référence les nœuds 'participants ' du state précédent



Transactions, Bloc, Gas ….

La blockchain, une machine à état : Les merkle Tree au cœur de la blockchain

- Dans un Merkle Tree, chaque nœud de l'arbre stocke un hash Merkle (non réversible) correspondant au hash de la concaténation de ces deux enfants, les feuilles de l'arbre contiennent les objets finaux.

- Le nœud parent final est donc un hash qui est dépendant des hash de tous les descendants de l'arbre et dans la blockchain EVM, un merkle patricia trie est immutable (il ne peut pas être modifié)

- Le hash parent d'un stateRoot d'un bloc donné est stocké dans le bloc associé et permet ainsi de vérifier beaucoup de chose (une transaction T fait bien partie du bloc Bx, l'account X a bien envoyé 10BNB à l'account Y, l'adresse @1 est-elle connue de la blockchain ….)

Transactions, Bloc, Gas ....

La blockchain, une machine à état : Le state Root

- Le state tree est le seul merkle tree à évoluer à chaque bloc (pour les autres arbres, transaction tree et le receipt tree , on crée un nouvel arbre à chaque nouveau bloc)
- Le state tree stocke l'état des accounts et un account contient les valeurs suivantes

```
type Account struct {
    Nonce    uint64
    Balance  *big.Int
    Root     common.Hash // merkle root of the storage trie
    CodeHash []byte
}
```

Transactions, Bloc, Gas ....

La blockchain, une machine à état : Le state Root

- L'account stocké dans le state tree
  - Nonce : Identifiant unique incrémenté à chaque transaction envoyée par un account external ou à chaque création de contrat par un smart contract
  - Balance : Balance de l'account
  - Account trie root : hash du nœud parent d'un arbre spécifique à l'account permettant de stocker des données spécifiques à l'account (notamment le bytecode d'un smart contract), cet arbre est susceptible d'évoluer au fil du temps, lorsque des variables storages d'un contrat sont modifiées.
  - CodeHash : Vide pour un external account

Transactions, Bloc, Gas ....

Les transactions : Rappels

- Au sein d'une blockchain EVM, une transaction est une opération qui modifie l'état de la blockchain
- Une transaction a toujours pour origine un external account (action humaine ou automatisée en dehors du système de la blockchain).
- Une transaction est toujours signée par la clé privée associée à un account (voir le workshop Wallets pour plus de détails)
- Il existe deux types de transactions * :
  - Transfert de valeur ou exécution d'un smart contract
  - Création d'un nouvel account et de son code associé (création d'un smart contract)

** En réalité, la blockchain a évolué et on a maintenant d'autres types de transactions possibles qui ne seront pas évoquées ici*

Transactions, Bloc, Gas ....

*Les transactions : Le contenu d'une transaction **

- nonce: Nombre de transactions envoyées par l'initiateur ou nombre de création de contrats pour une adresse de contrat
- gasPrice: Le nombre de wei par unité de gas que l'initiateur souhaite payer pour que sa transaction soit exécutée
- gasLimit: La quantité de gas maximale qui doit être utilisée pour exécuter la transactions
- to: Adresse à laquelle le message est destiné ou vide dans le cas d'une création de contrat

** Les champs de la transaction ont évolué avec la version London publiée en 2021 pour Ethereum et avec la version Burno Upgrade pour la BNBChain depuis Novembre 2021*

Transactions, Bloc, Gas ....

*Les transactions : Le contenu d'une transaction **

- value: Nombre de weis à transférer (peut être zéro s'il n'y a pas lieu de transférer des weis)
- v, r, s: Correspond à la signature de la transaction opérée par l'account appelant (Permet de vérifier que l'account qui prétend être à l'origine de la transaction est bien le bon)
- Init : Champ présent uniquement pour les transactions créant un contrat, et qui contient le code du contract
- Le transactions peuvent être fabriquées via un wallet ou via du code

** Les champs de la transaction ont évolué avec la version London publiée en 2021 pour Ethereum et avec la version Burno Upgrade pour la BNBChain depuis Novembre 2021*

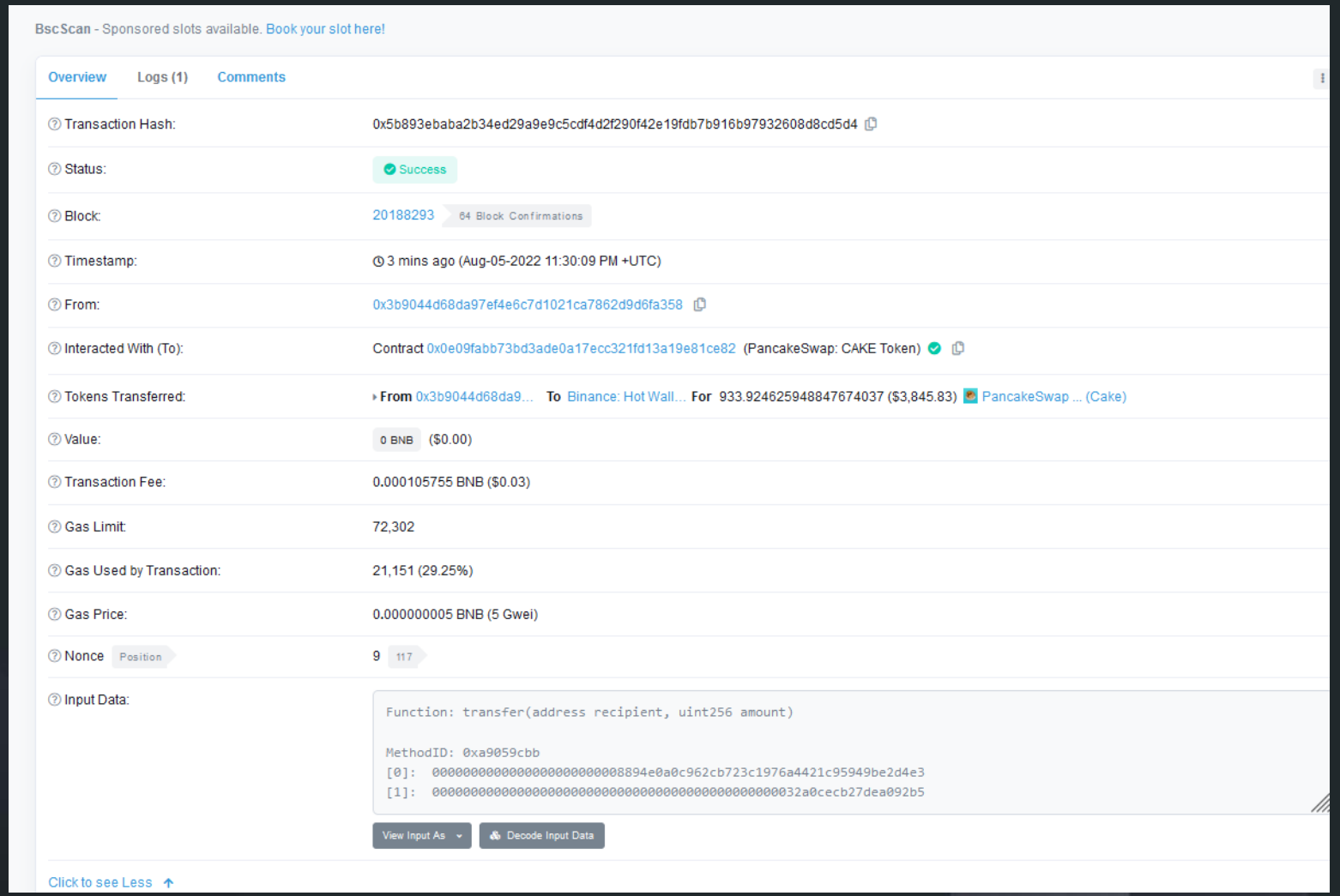
Transactions, Bloc, Gas ....

Les transactions : Envoyer une transaction avec Python

```python
from web3 import Web3

bsc_test_rpc = 'https://data-seed-prebsc-2-s1.binance.org:8545/'
web3 = Web3(Web3.HTTPProvider(bsc_test_rpc))
account_1 = '0x710fD0E2CC92dbD3a15f1AB316Be3a6FC479461A'
private_key1 = '3de00fe27c36ff253abb5a4a05b9c57ac389650f261929c376f999a5f712e09c'
account_2 = '0xfEB9Cf42Cd4E9778273152e761dF9471A1767708'

#get the nonce.  Prevents one from sending the transaction twice
nonce = web3.eth.getTransactionCount(account_1)

#build a transaction in a dictionary
tx = {
    'nonce': nonce,
    'to': account_2,
    'value': web3.toWei(1, 'ether'),
    'gas': 2000000,
    'gasPrice': web3.toWei('50', 'gwei') # Un gwei est un giga wei soit 1 000 000 000 wei

}

#sign the transaction
signed_tx = web3.eth.account.sign_transaction(tx, private_key1)

#send transaction
tx_hash = web3.eth.sendRawTransaction(signed_tx.rawTransaction)

#get transaction hash
print(web3.toHex(tx_hash))
```

Transactions, Bloc, Gas ….

Les transactions : Lire une transaction



Transactions, Bloc, Gas ....

Les blocs : Le contenu

- Header
  - Contient des informations importantes pour la sécurisation et la validation du bloc par les autres mineurs ou validateurs
- Liste des headers des blocs orphelins (non utilisé sur la BNBChain)
  - Un bloc orphelin dans le cadre d'une blockchain EVM POW est un bloc qui a été miné au même moment qu'un autre bloc mais qui n'est pas intégré dans le chemin de la blockchain, le mineur associé reçoit quand même une fraction de la reward associée au minage d'un bloc
- Liste des transactions
  - Liste des transactions embarquées dans le bloc

Transactions, Bloc, Gas ....

Les blocs : Le Header

- **parentHash**: Hash Keccak-256 du block header du bloc parent
- **ommersHash**: Hash Keccak-256 de la liste des blocs orphelins de ce bloc
- **beneficiary**: L'adresse qui reçoit les frais de transactions associés au bloc
- **stateRoot**: Hash Keccak-256 du nœud parent du state trie finalisé
- **transactionsRoot**: Hash Keccak-256 du nœud parent du transaction trie contenant toutes les transactions du bloc
- **receiptsRoot**: Hash Keccak-256 du nœud parent du receipt tree contenant tous les receipts associés aux transactions du bloc
- **logsBloom**: Filtre de bloom permettant d'indexer les receipts et de vérifier si une adresse émettrice ou un identifiant de receipt est présent dans la liste de manière rapide

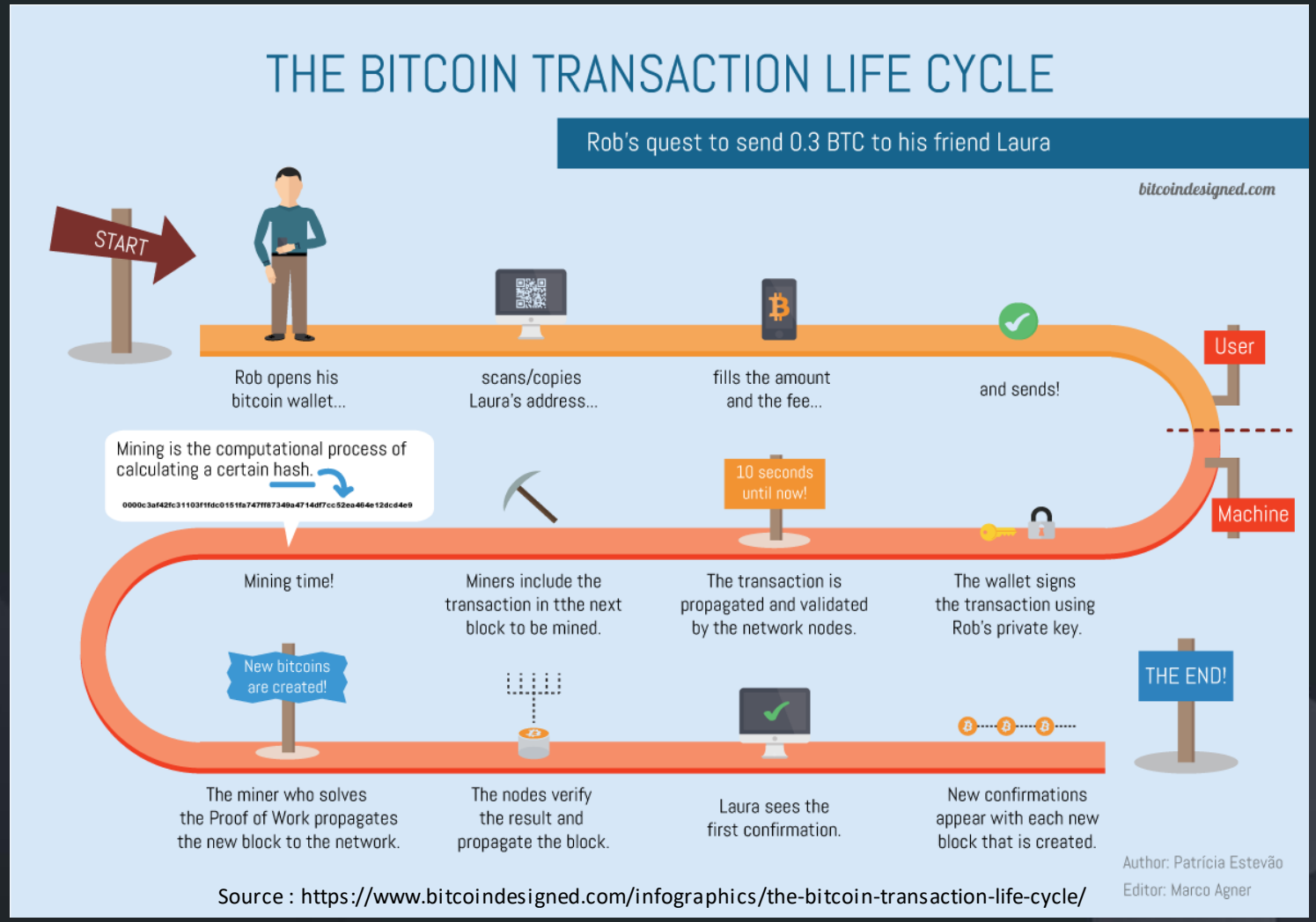
Transactions, Bloc, Gas ....

Les blocs : Le Header

- **difficulty**: Difficulté de minage (la difficulté de minage est réévaluée par le système à chaque nouveau bloc)
- **number**: Numéro du bloc depuis genesis (0)
- **gasLimit**: Limite de gas autorisée pour le bloc courant (le bloc ne peut pas consommer plus que cette limite)
- **gasUsed**: Totalité du gas utilisé pour exécuter toutes les transactions du bloc
- **timestamp**: Timestamp Unix du bloc
- **extraData**: Espace dédié à stocker des informations supplémentaires pour ce bloc
- **mixHash**: Hash de 256-bit hash participant au consensus pour le POW et permet de sécuriser cryptographiquement un bloc dans le cas du POS
- **nonce**: Nombre de 64 bits participant au consensus pour le POW (vaut 0 dans le cas de la BNB Chain)

Transactions, Bloc, Gas ....

La valse des transactions & des blocs: comment une transaction est validée



Source : https://www.bitcoindesigned.com/infographics/the-bitcoin-transaction-life-cycle/

Principes du proof of Work / Proof of Stake

Le proof of Work

Principes du proof of Work / Proof of Stake

Proof of Work : informations générales

- **Le proof of Work, couplé au reward distribué au mineur qui résout le problème est le mécanisme qui permet de sécuriser la blockchain et de résoudre le problème des généraux byzantins**

- C'est un mécanisme qui permet de s'assurer que la personne qui mine le prochain bloc de la blockchain a fourni une quantité de travail suffisante via la résolution d'un problème mathématique, Bitcoin a été la première blockchain à proposer un algorithme proof of work pour la sécurisation de son réseau

- A chaque fois qu'un bloc est miné, le mineur gagne le droit de récupérer les frais de transactions (gas) associés au bloc qu'il a miné, et déclenche la création de monnaie qui lui sera destinée (reward).

Principes du proof of Work / Proof of Stake

Proof of Work : informations générales

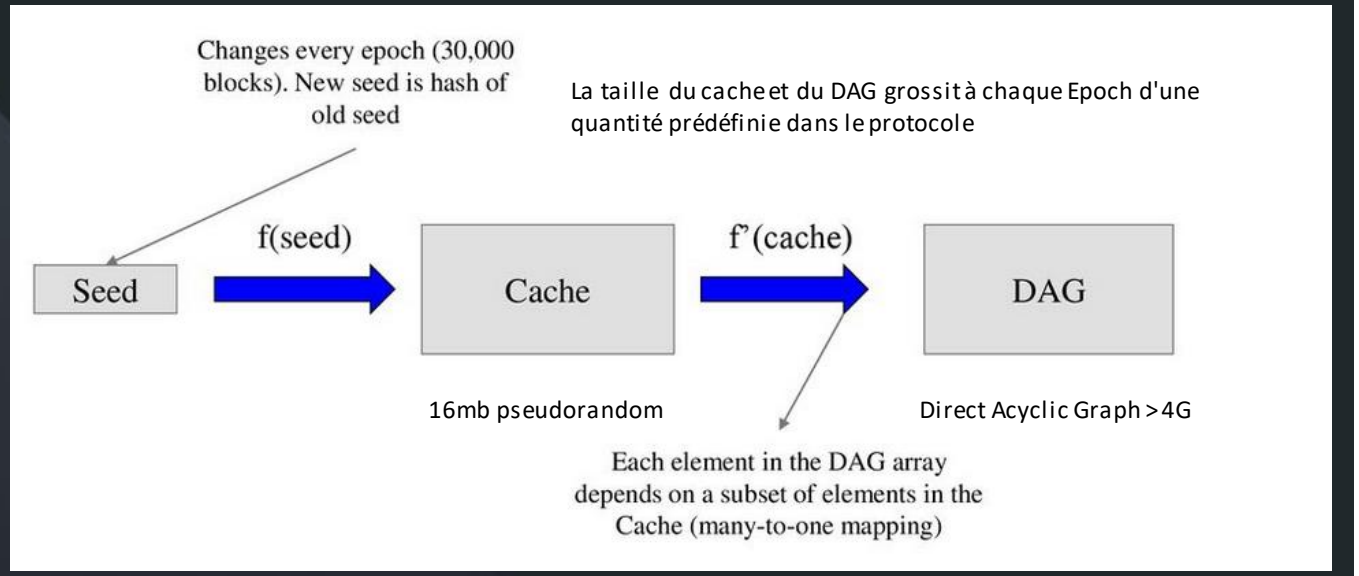
- Dans le cas d'Ethereum, l'algorithme qui permet de créer la preuve de travail pour le minage d'un bloc s'appelle Ethash, c'est une version modifiée de l'algorithme Dagger-Hashimoto.

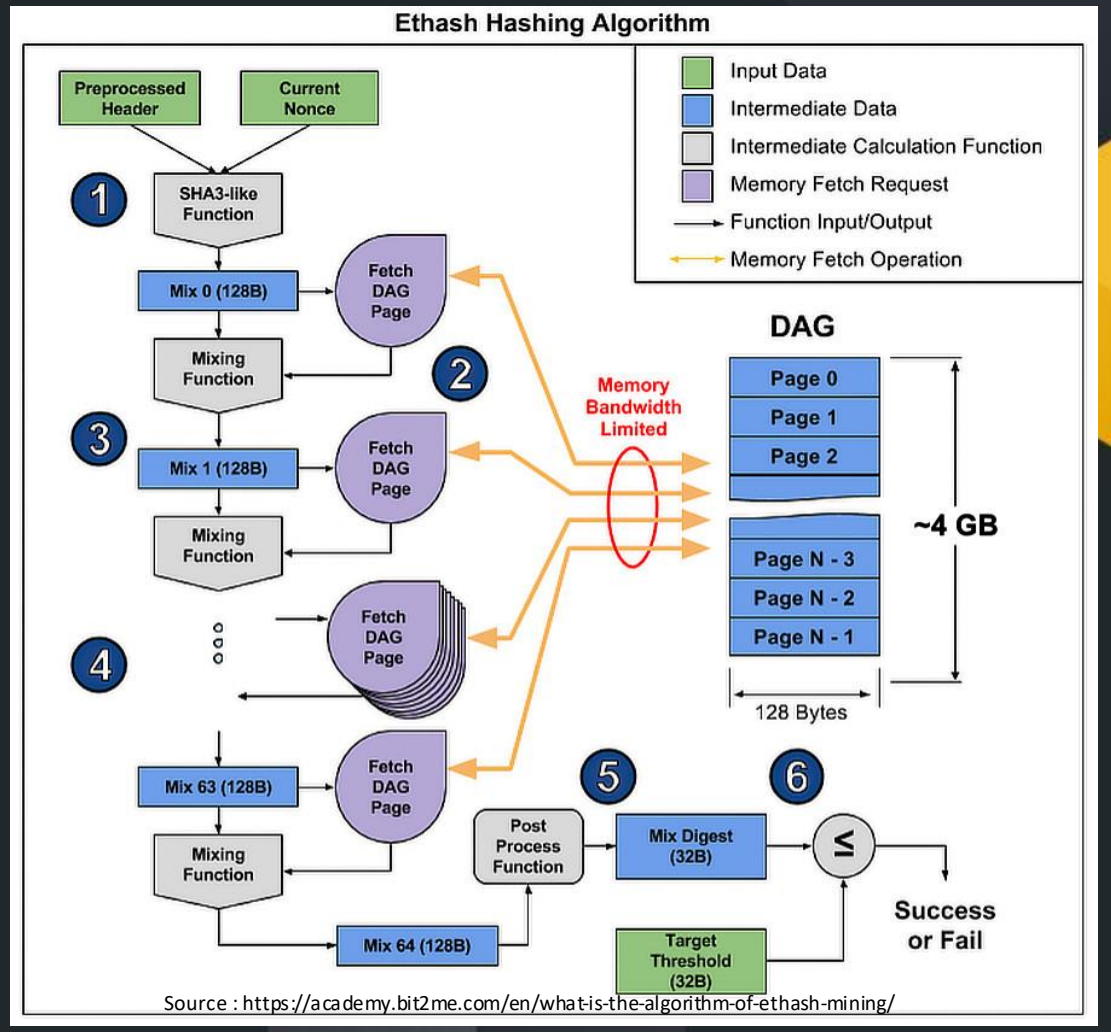
- Cet algorithme a été pensé pour être "ASIC Resistant" de part la nécessité des accès répétés à la RAM pour la résolution du puzzle mathématique (même si depuis 2018, la démonstration a été faite qu'on peut miner de l'Ethereum avec des ASICS).

- Le concept de proof of work est peu à peu abandonné au profit du proof of Stake car ce dernier est beaucoup plus écologique

Principes du proof of Work / Proof of Stake

Proof of Work : Ethash, comment ça marche





Principes du proof of Work / Proof of Stake

Proof of Work : Ethash, comment ça marche

- Une fois qu'un mineur a réussi à trouver une solution qui satisfasse à la difficulté courante du bloc, il fabrique son bloc et diffuse le bloc avec la preuve sur le réseau

- Chaque nœud du réseau dès lors qu'il vérifie que la preuve fournie est valide va
  - Récupérer des transactions du bloc
  - Réexécuter chaque transaction
  - Vérifier que le state root / tx root / receipts root sont corrects
  - Vérifier que le nonce et le mixhash produisent un résultat inférieure ou égal à la difficulté
  - Si 51% des nœuds valident le bloc, celui-ci est ajouté à la blockchain, et les nœuds arrêtent de chercher la solution pour passer au bloc suivant

Le proof of Stake

Principes du proof of Work / Proof of Stake

Proof of Stake BNB Chain: informations générales

- Le proof of Stake ne repose pas sur une preuve de travail mais plutôt sur la quantité de jetons possédé par un validateur

- Les blocs ne sont pas minés mais forgés, et ne nécessitent pas d'investissement couteux dans du matériel de minage (GPU ou ASIC) mais nécessitent quand même de posséder un certain montant en BNB à staker pour pouvoir postuler en tant que validateur (minimum de 10 000 BNB, soit 3 200 000 $ à l'heure actuelle)

- La sélection des validateurs pour la blockchain dépend de la blockchain, sur BNBChain, il y a un total de 41 validateurs, 21 actifs et 20 candidats.

Principes du proof of Work / Proof of Stake

Proof of Stake BNB Chain: informations générales

- La quantité de BNB étant limitée (200m, avec un objectif de 100m burn à terme), les validateurs sont rémunérés avec les frais de gas des transactions, il n'y a pas de création de monnaie comme dans les POW (à noter qu'une petite quantité du gas, 1/16 ème, limité à 100 BNB, est ajoutée à un contrat tiers qui permet de rémunérer les relayers, voir plus loin pour le détail).

- Les 41 validateurs sont sélectionnés tous les jours à 00h00 UTC par un module intégré à la Beacon Chain et communiqué au sein de la BNB Chain via une communication cross chain. A noter que la Beacon Chain gère aussi le staking des BNB des validateurs.

- Si le réseau détecte une tentative de fraude de la part d'un validateur, une opération de slashing est actionnée, le slashing est une pénalité sur les BNB stakés par le validateur

- Si un validateur échoue à produire un bloc, il est aussi slashé

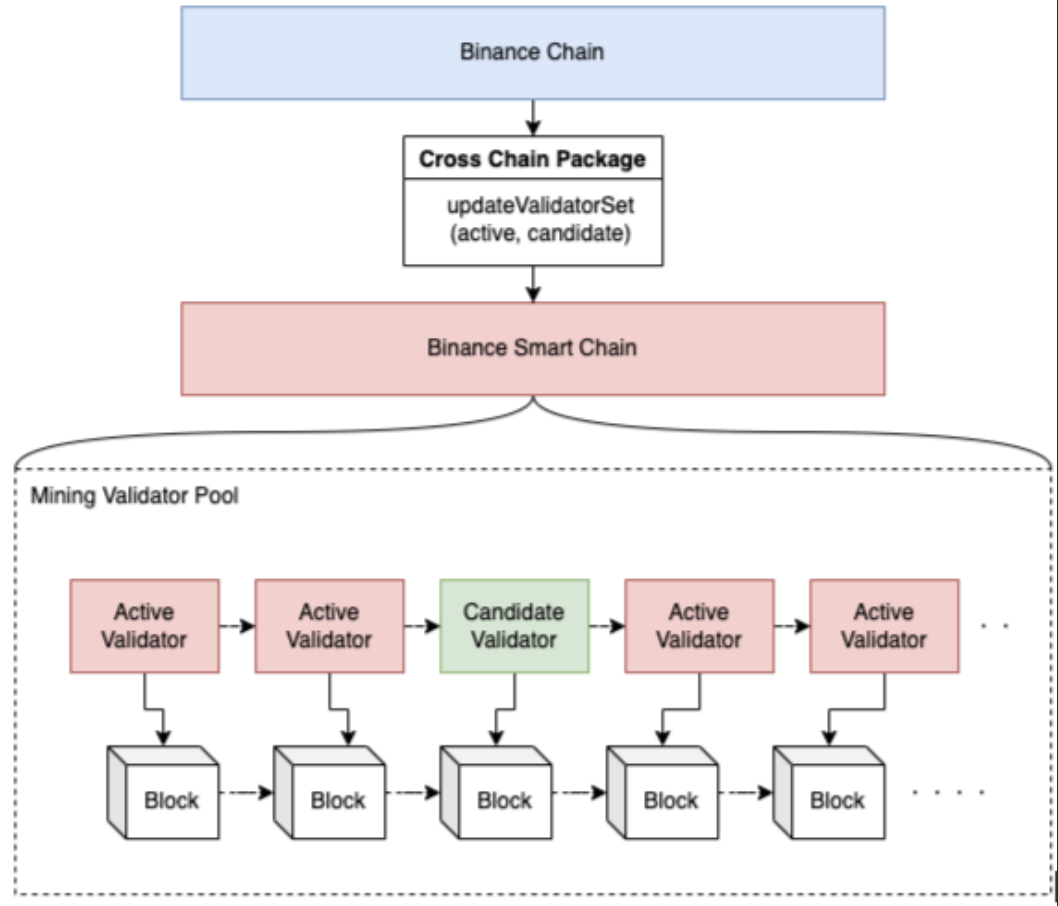
Principes du proof of Work / Proof of Stake

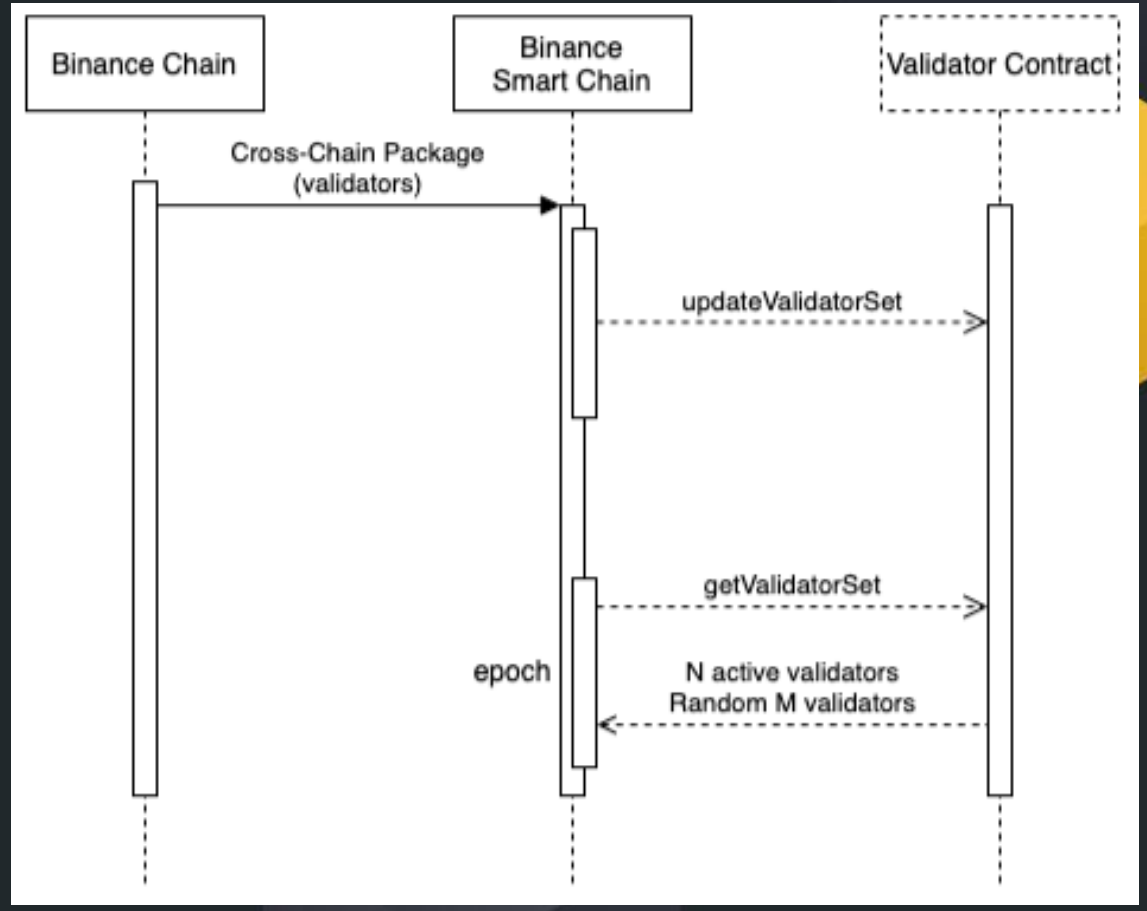
Proof of Stake BNB Chain: Fondamentaux

- La BNB Chain repose sur plusieurs briques
  - La beacon chain
    - Elle est responsable du staking des BNB des validateurs et elle porte le processus de sélection des validateurs ainsi que leur renouvellement
    - La communication entre la BC et la BNBChain se fait un protocole cross chain qui repose sur des relayers
  - Les validateurs de la BNBChain
  - Des contrats BNBChain genesis

Principes du proof of Work / Proof of Stake

Proof of Stake BNB Chain: Sélection des validateurs (BEP131)





Principes du proof of Work / Proof of Stake

Proof of Stake BNB Chain: Sélection des validateurs (BEP131)

- Les 41 validateurs sont sélectionnés tous les jours à 00h00 UTC par un module dédié au niveau de la Beacon Chain (21 prime et 20 candidats)

- Les nœuds Relayer récupèrent l'information depuis la BC et mettent à jour la BNBChain au travers d'un smart contract dédié

- Tous les 200 blocks, les nœuds appellent le smart contract pour récupérer les 21 nœuds validateurs actifs (le smart contract sélectionne 19 validateurs parmi les 21 et deux parmi les candidats de manière aléatoire)

Principes du proof of Work / Proof of Stake

Proof of Stake BNB Chain: Sélection des validateurs (BEP131)

- Durant une epoch, la sélection du validateur qui validera le prochain bloc se fait dans l'ordre (nœud 1, puis nœud 2, puis ….)

- Avant la BEP 131, les 21 validateurs étaient sélectionnés et figés pour 24 h

Principes du proof of Work / Proof of Stake

Proof of Stake BNB Chain: Emission des blocs

- Le principe de la diffusion des transactions et de leur prise en compte reste le même que pour les blockchains à base de POW (stateRoot, transactionRoot ….)

- Lorsque le tour d'un validateur arrive, il réalise le processus de fabrication du bloc en exécutant les transactions (impacte les différents trees) et signe le header du bloc puis ajoute la signature au champ extraData du bloc puis il broadcaste le bloc aux autres validateurs

Principes du proof of Work / Proof of Stake

Proof of Stake BNB Chain: Emission des blocs

- Les validateurs témoins récupèrent le bloc et entament le processus de vérification en réexécutant toutes les transactions et en vérifiant que les différents hashs des trees correspondent comme dans le cadre du POW

- A noter qu'il y a un temps minimal à respecter pour valider un bloc, les nœuds témoins vérifient aussi que le temps est respecté

- Si au moment de la forge du bloc, on entre dans une nouvelle Epoch, le validateur ajoute aussi la liste des nouveaux validateurs dans le champ extraData

Principes du proof of Work / Proof of Stake

Proof of Stake BNB Chain: Les contrats BNB genesis

- La BNB Chain vient avec un certain nombre de contrats natifs
- Ces contrats portent une partie de la logique utilisée par la BNB Chain pour la gestion des rewards pour les Validateurs et les relayers
- Ils gèrent aussi la communication cross chain avec la beacon chain et la gestion du slashing
- Les relayers utilisent ces contrats pour communiquer des informations de la Beacon Chain vers la BNB Chain et inversement et son payés avec les 1/16 des fees prélevés lors de la validation d'un bloc

Principes du proof of Work / Proof of Stake

Proof of Stake BNB Chain: Sécurité

- La sécurité dans le POS de BNBChain repose sur le nombre de validateurs et sur le slashing

- Lorsqu'un nœud validateur a un comportement inadéquat (temps de validation trop long, absence de validation, tentative de double dépense) il peut être exclu ou mis en "jail" et sera aussi amputé d'une partie de ces jetons stakés, lesquels seront redistribués (10 000 BNB pour une tentative de double dépense)

- N'importe quel user de la BNBChain peut faire une requête de slashing

Les ZK Rollups

Les ZK Rollups

Principes

- Un ZK Rollup est une blockchain EVM de niveau 2 (Layer 2) qui permet d'augmenter la scalabilité d'une chaine principale (L1) et de diminuer les frais de gas

- Les zk rollups peuvent avoir un seul nœud en charge de l'exécution des transactions et du report dans la L1 ou peuvent aussi fonctionner sur le principe du POS avec plusieurs validateurs qui doivent staker pour participer

- La blockchain reste fortement liée à la blockchain principale puisque toutes les transactions ayant eu lieu sur le L2 sont répercutés dans la blockchain L1, cependant les transactions sont exécutées off chain

Les ZK Rollups

Principes

- Le lien entre le Layer 2 et la chaine Layer 1 se fait au travers de smart contracts déployé sur le Layer 1 (rollup contracts)

- Les transactions sont compressées et reportées du L2 vers le L1 dans des champs "calldata" (stockés dans le receipts root et moins cher que le storage) via les rollup contracts

- Le L2 publie aussi sur la L1, les différents hashs de tree à chaque report de transactions

Les ZK Rollups

Principes

- Les fonds gérés dans la blockchain zkRollup sont des assets peggués sur les assets de la L1, par exemple si Bob dépose 10 BNB sur le rollup contract, 10 zkBNB seront mintés sur le L2 (via une communication cross chain initiée par le validateur coté L2)

- Contrairement à l'Optimistic Rollup, le zkRollup fournit à chaque report une preuve cryptographique (zero knowledge) qui est vérifié par le rollup contract de la L1. Cela permet aussi de s'assurer qu'une transaction est valide dés lorsque la preuve est vérifiée côté L1

- Cette preuve permet de s'assurer que les modifications des trees de la L2 sont légitimes, on appelle cette preuve "Validity proof" ou "Zk proof"

Merci !

Questions ?

Annexes


```
"difficulty":"0x2",
"extraData":"0xd88301010b846765746888676f312e31332e34856c696e75780000005d43d2fd24501664b50b6a34b645d654ba6ee8546ad6892382e727b971f665cf11e086985d51d82c669ca70
"gasLimit":"0x541f94e",
"gasUsed":"0xede688",
"hash":"0xa13d028f2a4369a26b06b33b27886fb1418fdb7f96f2b49ad77759e7542c6785",
"logsBloom":"0xa1a2329838a560587ab8c6c09519e975ec05a26bbdb4878f75509ff2291b2d85ba5d417bc91db0a443bb10308a260e40260fe79e083a356c675c0280823f6bf5c06cc725c536153
"miner":"0xea0a6e3c511bbd10f4519ece37dc24887e11b55d",
"mixHash":"0x0000000000000000000000000000000000000000000000000000000000000000",
"nonce":"0x0000000000000000",
"number":"0x1352937",
"parentHash":"0x241102140f06cc28b1782e1f836e14497617089403619ba7d3efade3eccd61e9",
"receiptsRoot":"0x150265094c3b184a392b539d47b7b79093ba479e6cd0326b134fec4ec2a3b842",
"sha3Uncles":"0x1dcc4de8dec75d7aab85b567b6ccd41ad312451b948a7413f0a142fd40d49347",
"size":"0x8e24",
"stateRoot":"0x4e6b283bf8a301ee9a1918d6e8c698b395848700e73dfaad06a00b75abc03c61",
"timestamp":"0x62f0fedd",
"totalDifficulty":"0x266c37b",
"transactions":[ ⊞ ],
"transactionsRoot":"0x3066955af60a112b9b7330ebc724985e190d32a4d1ca5e1968e178adbc52cc6c",
"uncles":[ ⊟

]
```