INTRODUCTION, BITCOIN

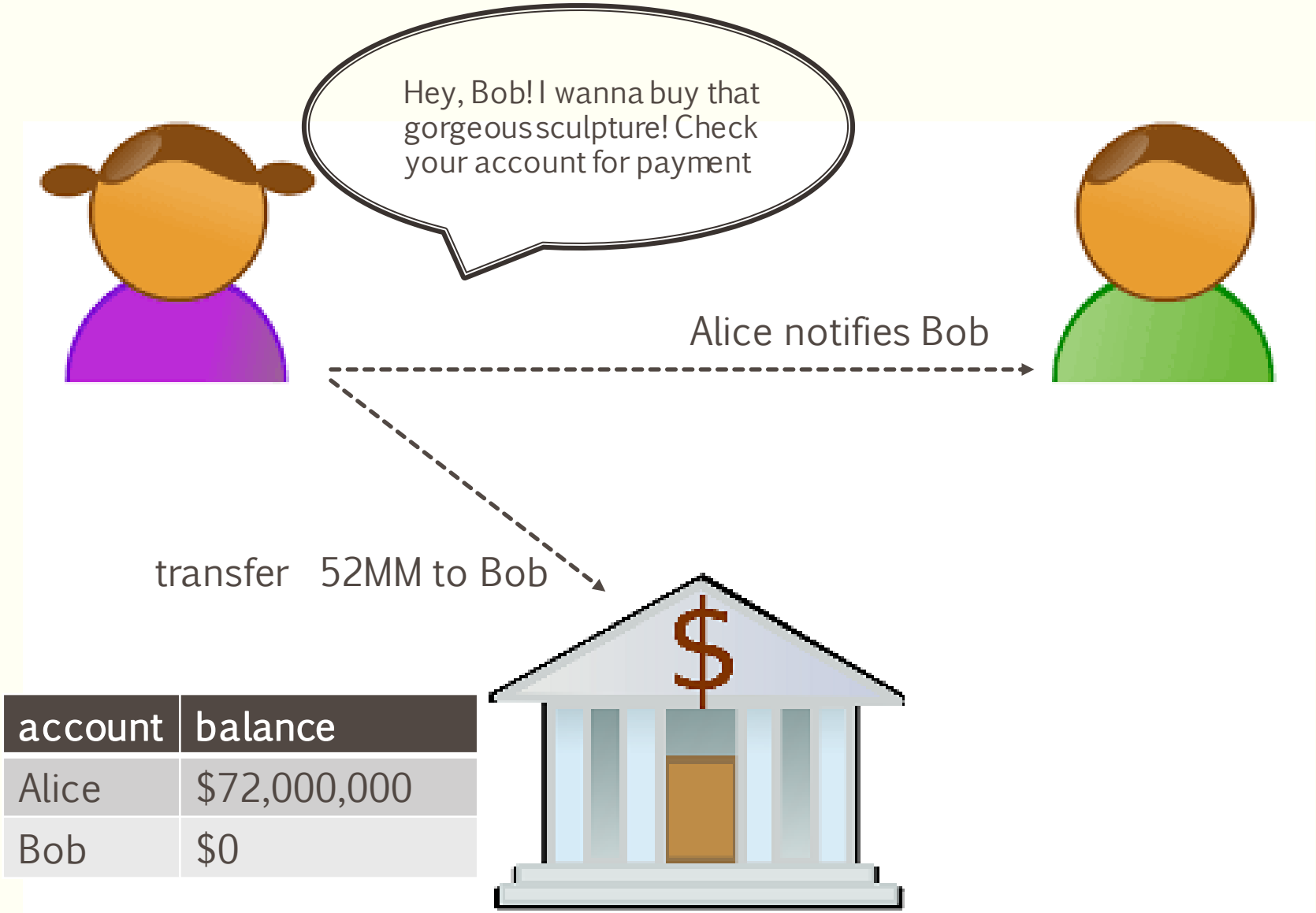
Blockchain technologies, **lecture 1**

Course overview

- Cryptocurrencies, history and motivation
- Blockchain characteristics and types of blockchains
- What is blockchain/how it works
- Bitcoin -- PoW
- Block difficulty
- Transactions, UTXO
- Applications of blockchain technologies

WHY BLOCKCHAIN

Cryptocurrencies, history and motivation



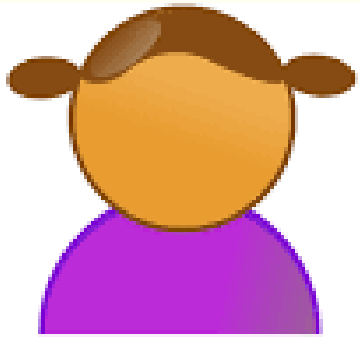
account	balance
Alice	$72,000,000
Bob	$0

traditional payments

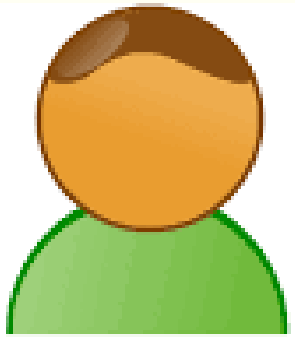
and

cryptographic hash function

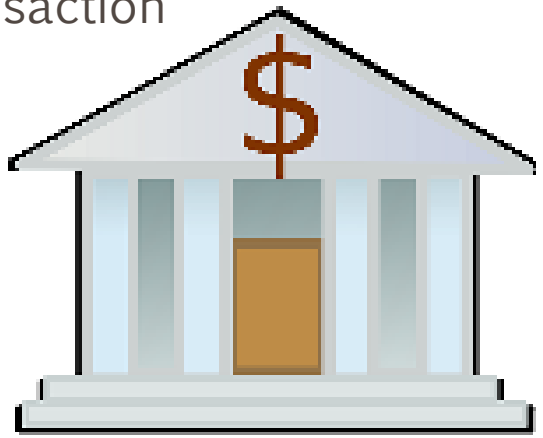
plain text is encrypted using a cipher to generate a hash value of fixed length



Bob check his balance ...



Bank validates transaction



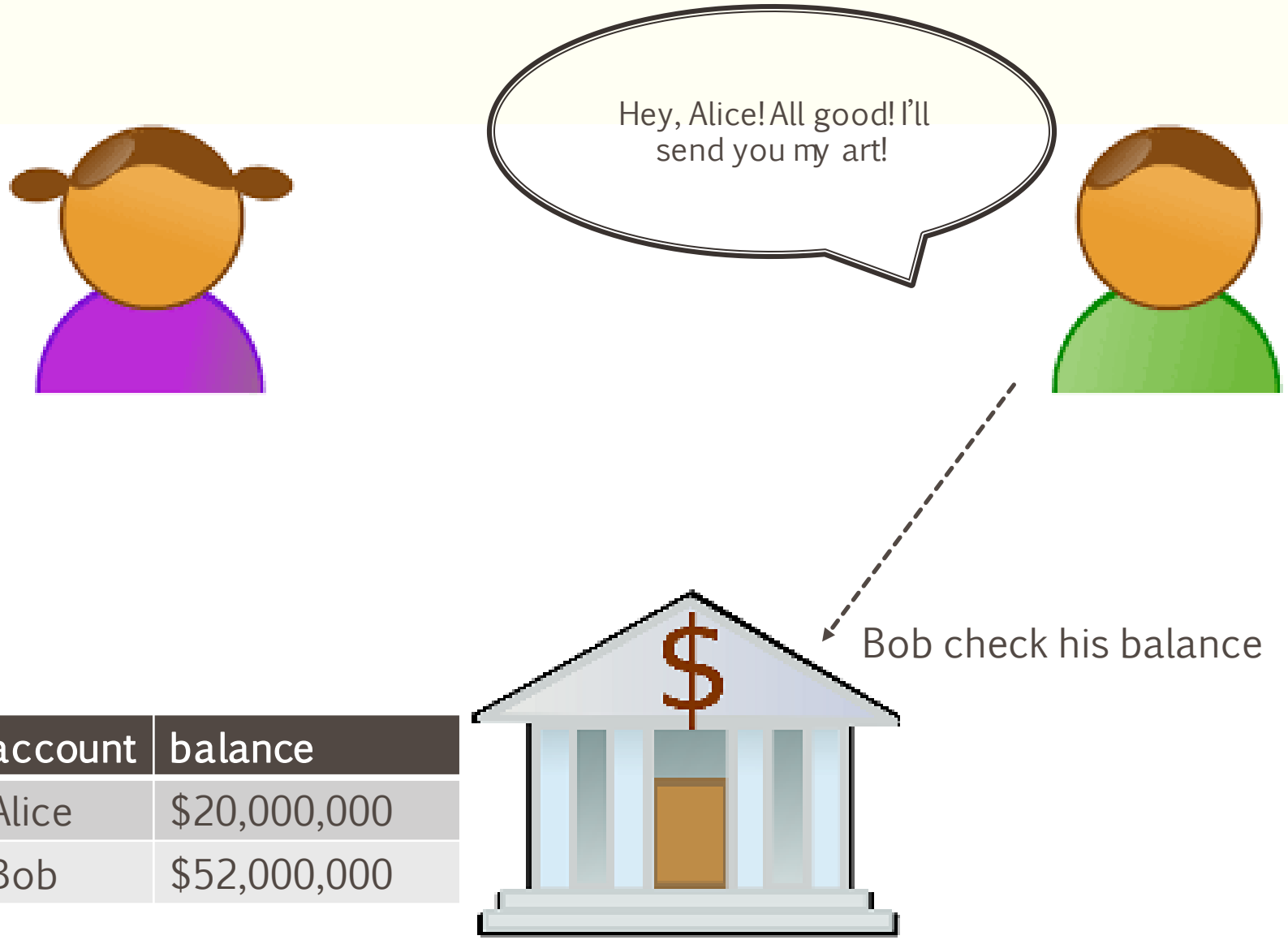
account	balance
Alice	$72,000,000
Bob	$0

traditional payments

and

cryptographic hash function

plain text is encrypted using a cipher to generate a hash value of fixed length



account	balance
Alice	$20,000,000
Bob	$52,000,000

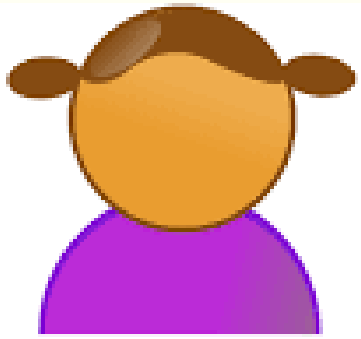
traditional payments

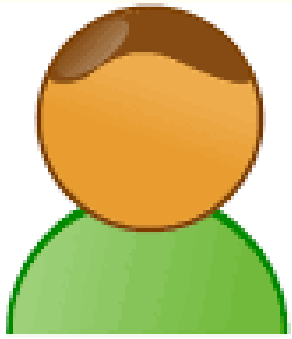
and

cryptographic hash function

plain text is encrypted using a cipher to generate a hash value of fixed length

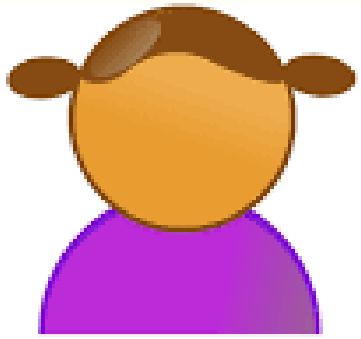
traditional payments

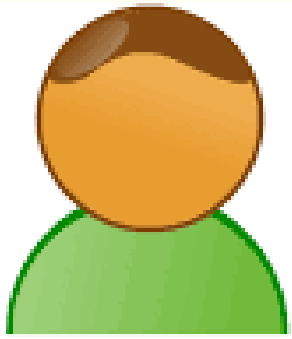




Bob sends “token”

account	balance
Alice	$20,000,000
Bob	$52,000,000



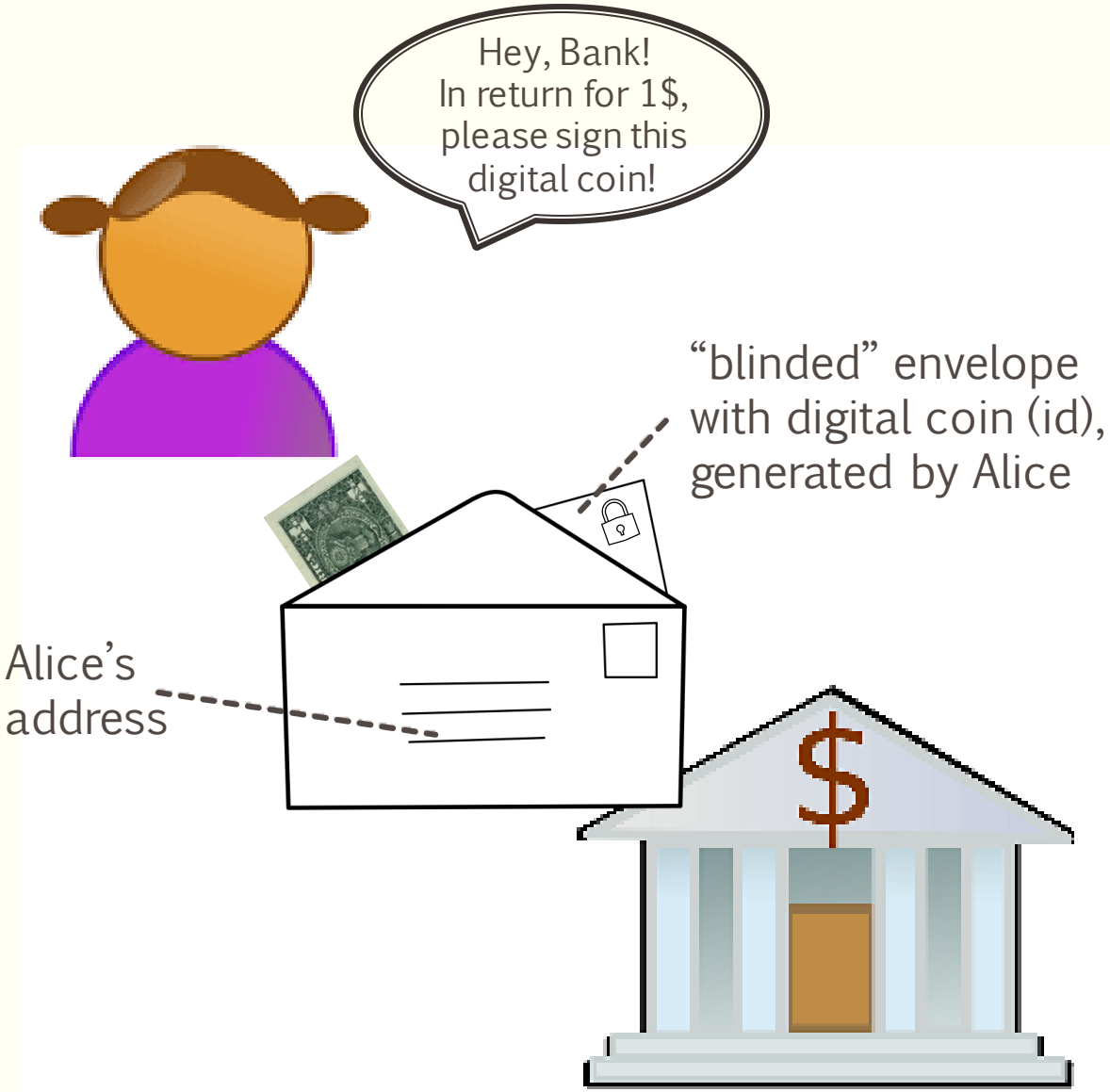


Bob sends “token”

account	balance
Alice	$20,000,000
Bob	$52,000,000

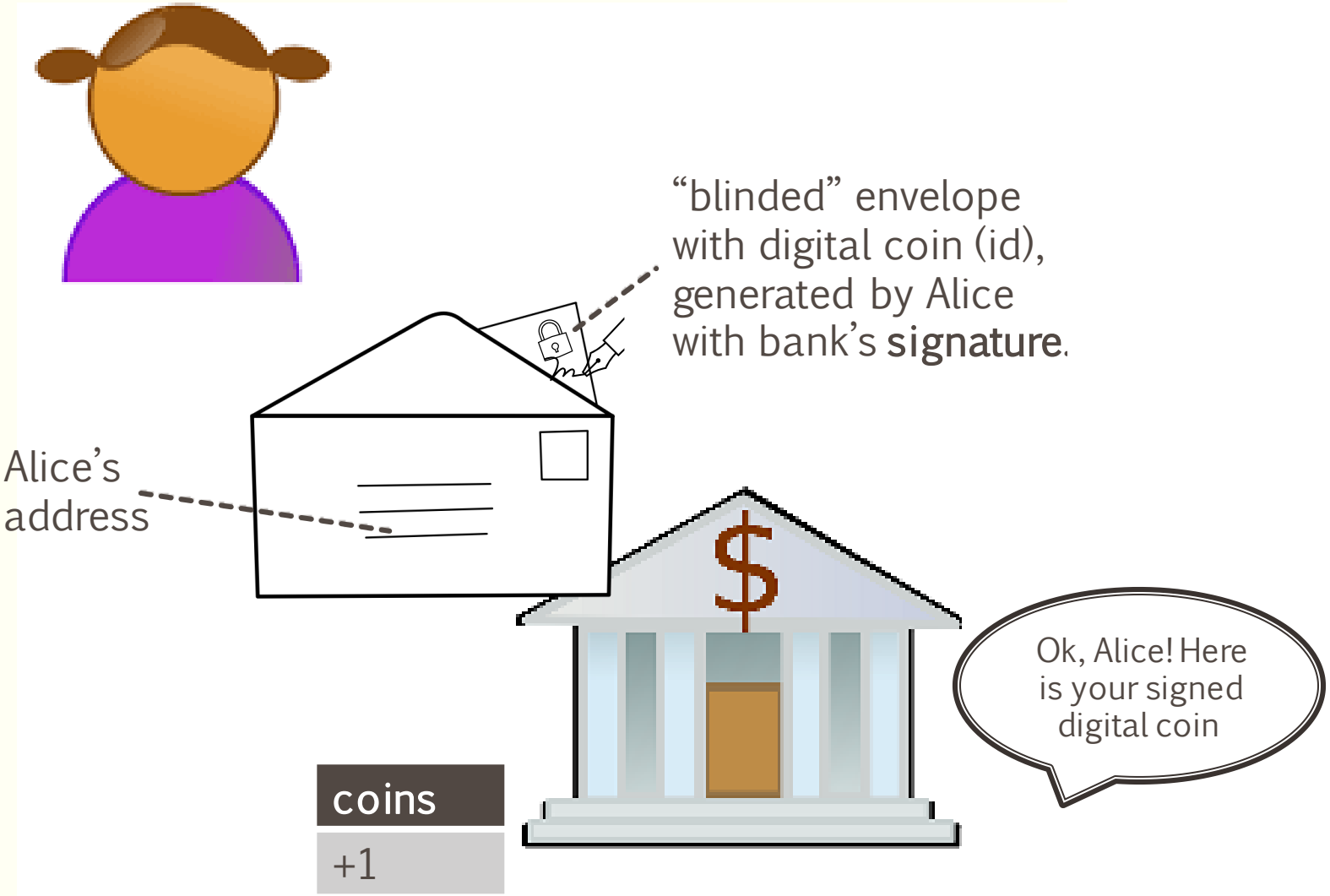
traditional payments

- Single point of failure, not peer-to-peer.
- Delayed or refused transactions.
- Security and privacy issues.
- Digital payments easy to implement.



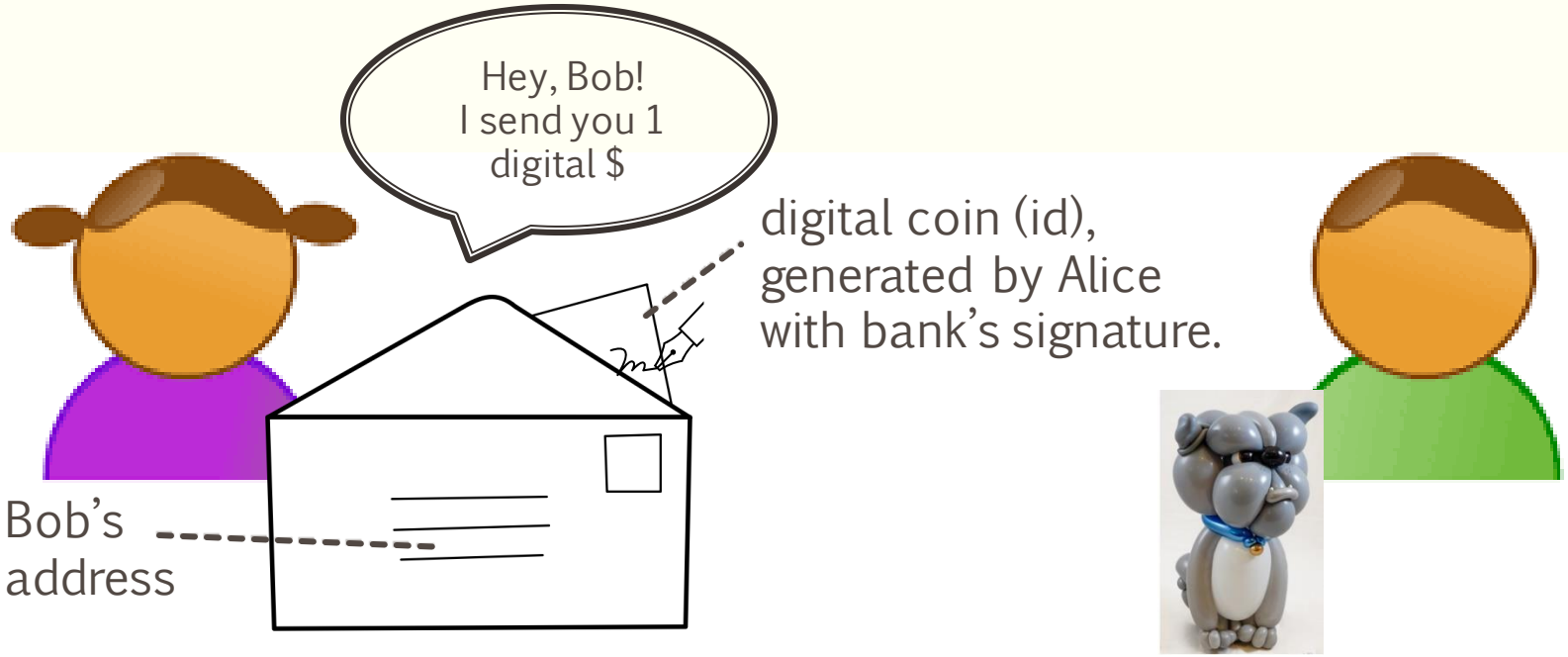
Chaum e-cash
Blind signatures

- Alice sends the bank a dollar and a blinded id, representing the digital coin she will be authorized to use in further transactions.
- Imagine an encrypted envelope or carbon paper nested in an envelope with Alice’s address on it.



Chaum e-cash (1983)
Blind signatures

- Bank is not able to find the id, it can only register how many coins are signed and sign Alice's blinded id.
- Bank has no record of Alices balance or transactions.
- Alice can recognize/recover the id she generated.
- Alice can recognize the signature of the bank.



Chaum e-cash
Blind signatures

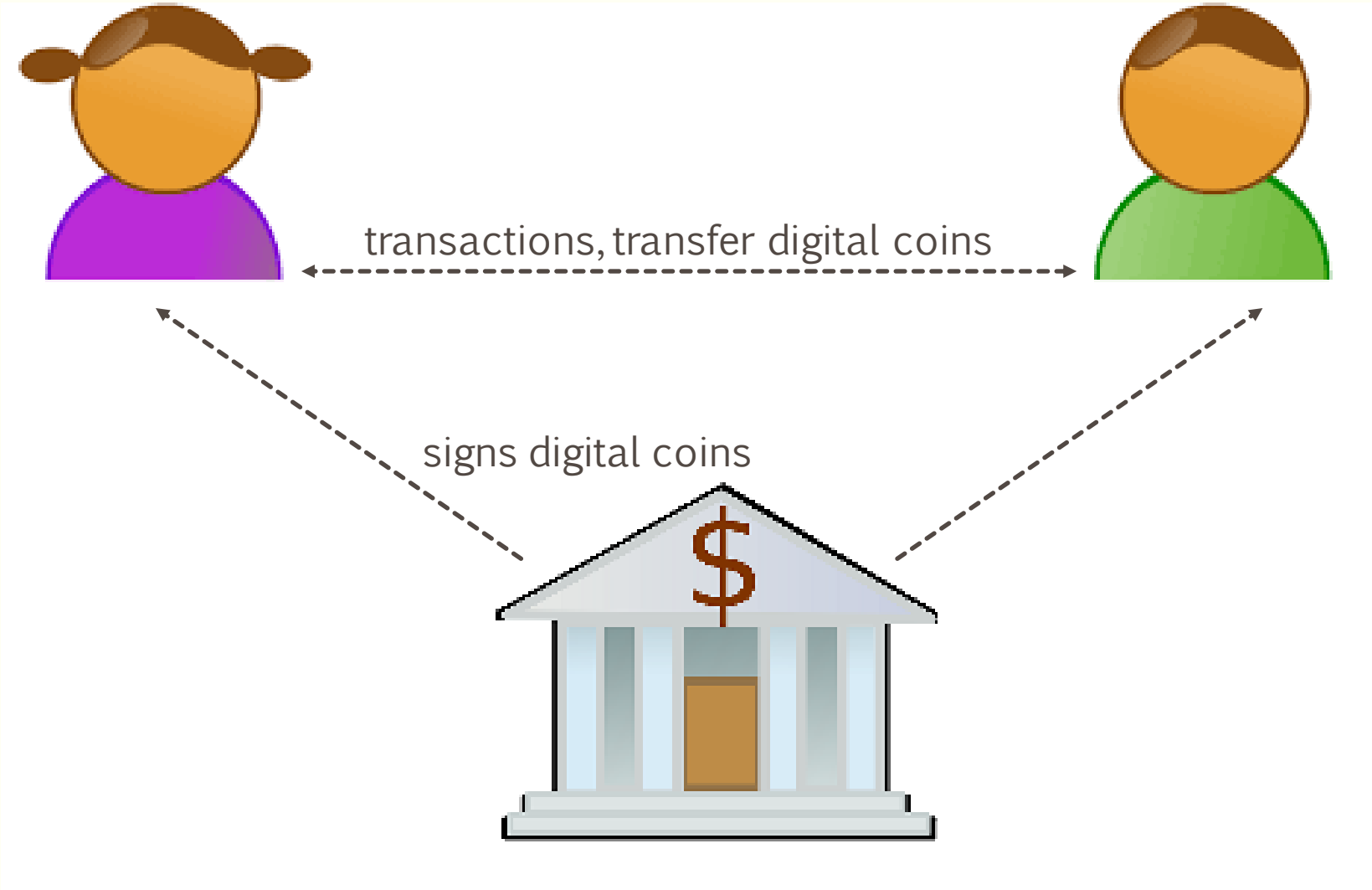
- Alice sends the signed, not blinded digital coin to Bob.

Chaum e-cash
Blind signatures

- Bob receives from Alice the id signed by the bank.
- Bob recognize the signature of the bank.

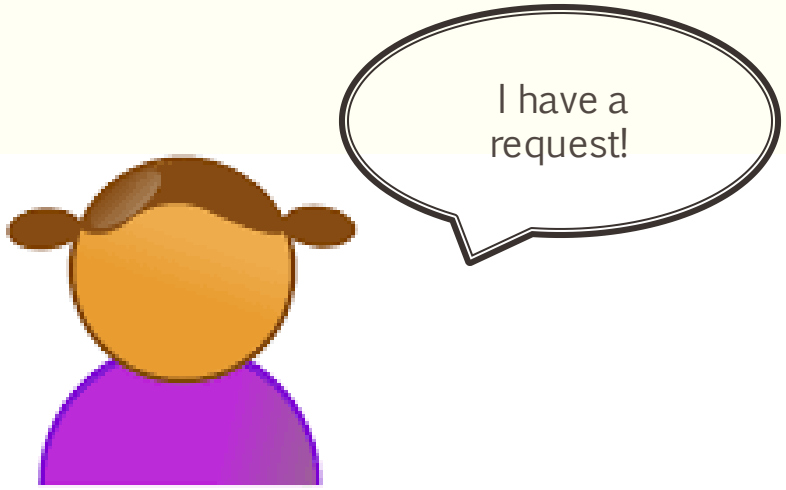
Chaum e-cash
Blind signatures

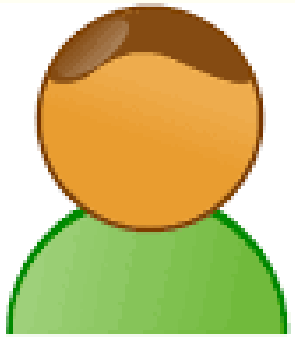
- Bobs claims 1$ in return for his digital coin.
- Bank registers the id.
- Bank doesn't know that the id came from Alice and it has no information about the transaction.
- Bank is able to detect **double spending**.



Chaum e-cash
Blind signatures

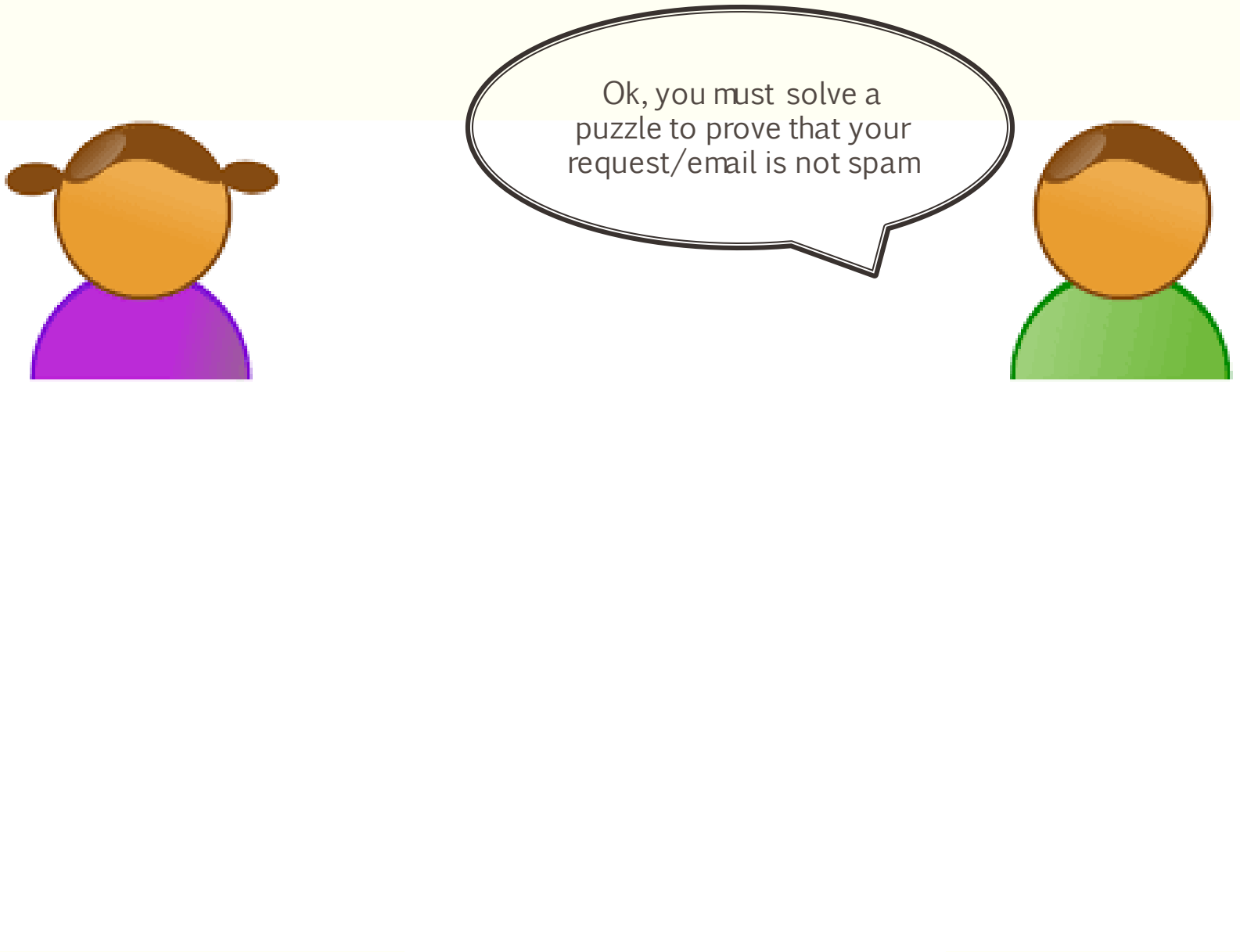
- Peer-to-peer, but bank signs all digital coins.
- Privacy.
- Digital payments.
- Double spending detection.





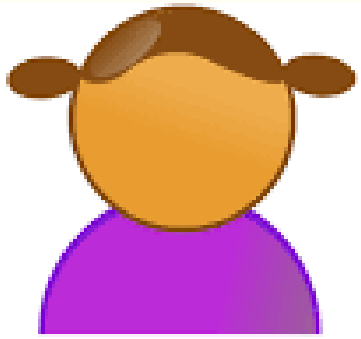
Hashcash (1997)
PoW and reusable PoW (2005)

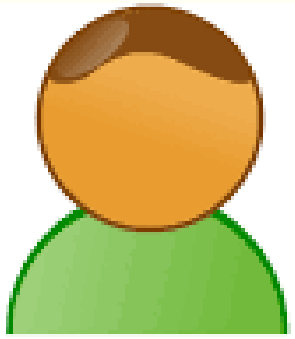
- Initially used to prevent **denial-of-service** attacks and spam emails.



Hashcash (1997)
PoW and reusable PoW (2005)

- Initially used to prevent **denial-of-service** attacks and spam emails.

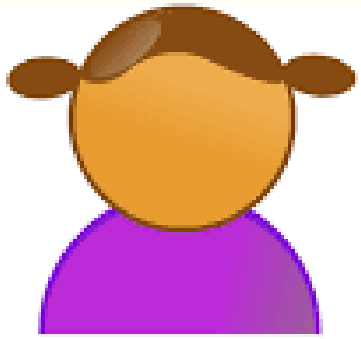




header
timestamp data_string …… …… …… counter

Hashcash (1997)
PoW and reusable PoW (2005)

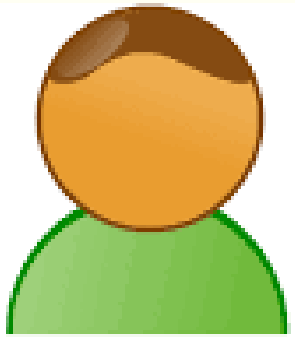
- Sender must fill a counter value, initialized to a random number.



SHA(header)=
00090030000100000500 4578…..5634

20 bits

160 bits



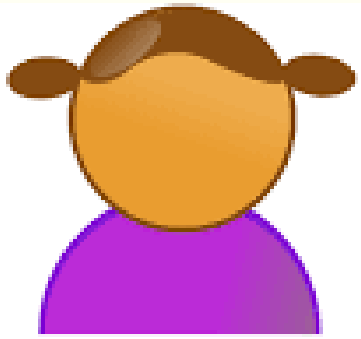
header

timestamp
data_string
……
……
……

counter

Hashcash (1997) PoW and reusable PoW (2005)

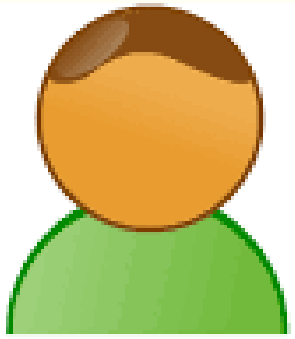
- Sender must fill a counter value, initialized to a random number.
- If SHA-1(header) has the first 20 bits set to 0, the header is valid
- If header is not valid (first 20 bits are not set to 0) increment counter.



SHA(header)=
00000000000000000000 4578…..5634

20 bits

160 bits



header

timestamp
data_string
……
……
……

counter+1

Hashcash (1997) PoW and reusable PoW (2005)

- Sender must fill a counter value, initialized to a random number.
- If SHA-1(header) has the first 20 bits set to 0, the header is valid
- If header is not valid (first 20 bits are not set to 0) increment counter.

BLOCKCHAIN CHARACTERISTICS

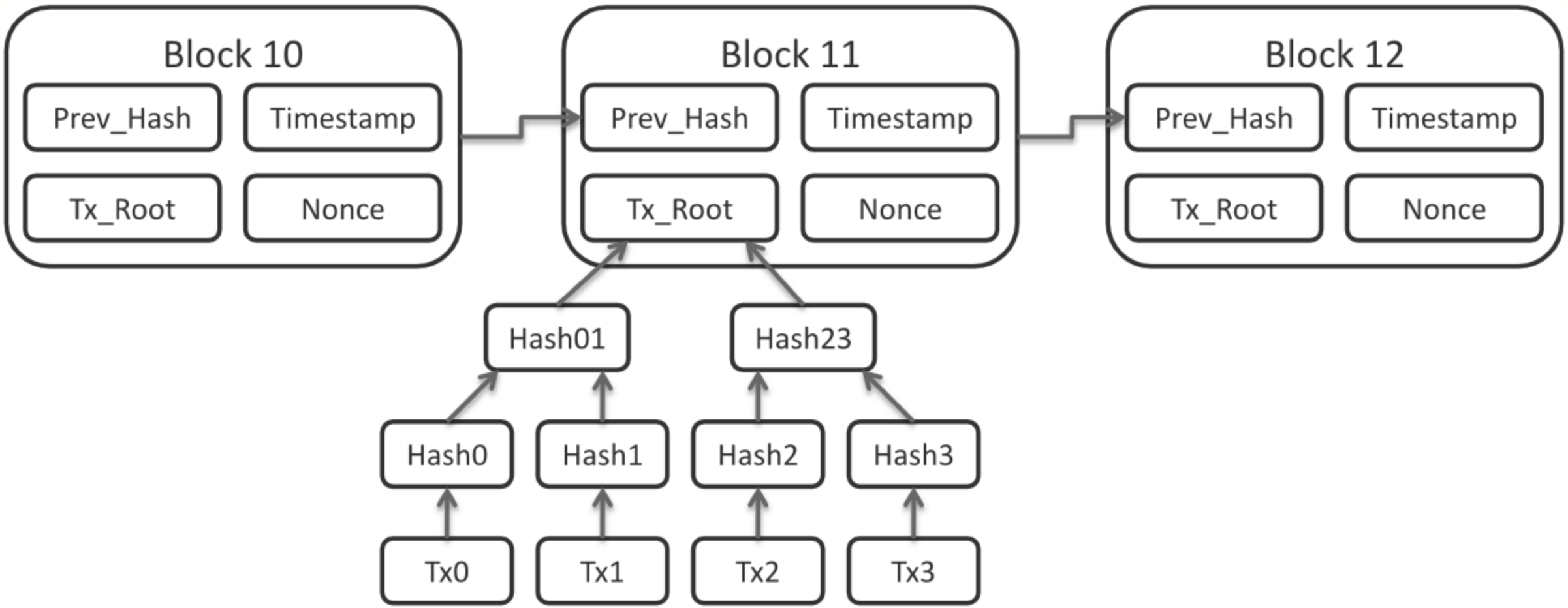
Blockchain characteristics

- **Public ledger:** A public database, all nodes share the same information about transaction and accounts (UTXO model or state-machine model).
- Records added in the ledger are immutable, only new transactions are continuously appended.
- All nodes must reach consensus, deciding the validity of transactions.
- **Auditable**: Transactions are timestamp and signed.

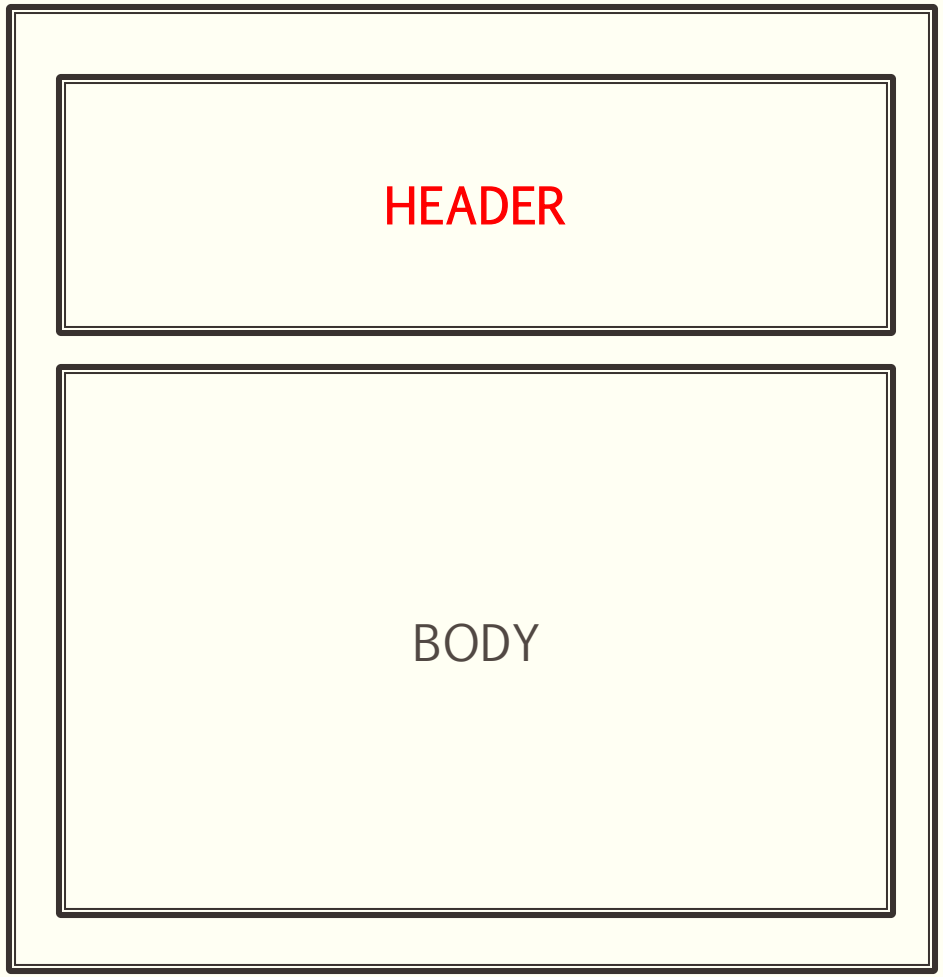
Blockchain characteristics

- **Immutable:** A public blockchain is a series of immutable record of data. Data is time-stamped
- **Decentralized**, **peer-to-peer**: Information is stored in a cluster of computers, there is no central authority. Everyone is accountable. Everyone keeps a copy of the database.
- **Transparent**: Everyone has access to all information.
- **Secure**: use asymmetric cryptography, data blocks are linked via hashes (block-chain) and protected via cryptographic functions.
- **Anonymity (pseudonimity)**: each participant may store several pairs of public-private keys to sign transactions or to prove ownership of his assets (UTXOs, ETHs, NFTs etc.) Identity is not revealed.

Transactions are gathered in blocks.
Each block has a header and a body.
Block is identified by its hash value.
Block header contains the hash of the previous block.

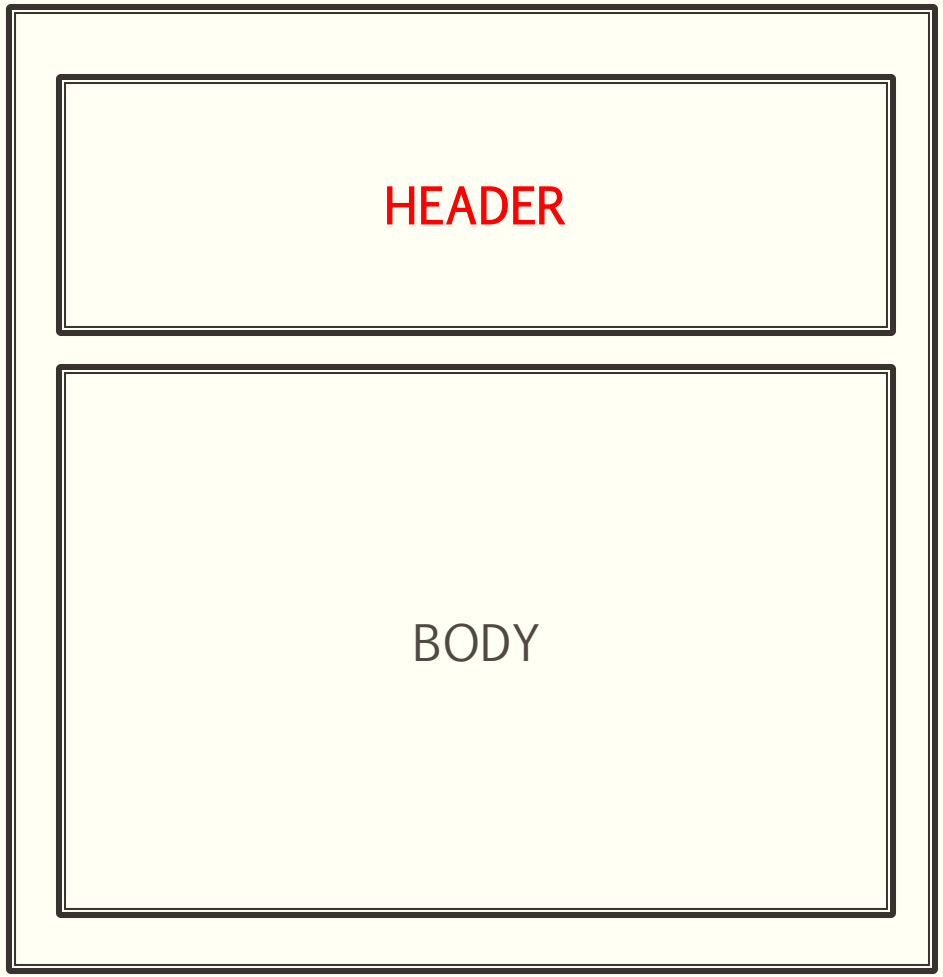


BITCOIN BLOCK STRUCTURE



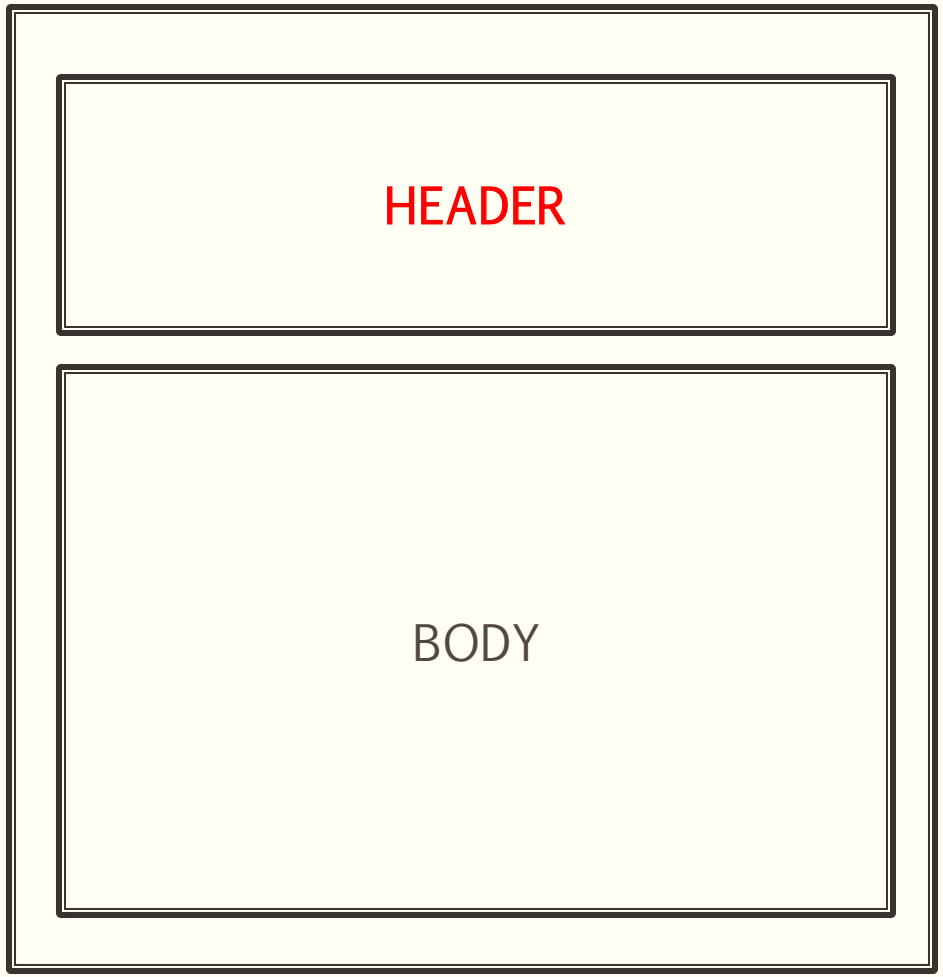
- 4-byte version.
- 4-byte timestamp.
- 4-byte difficulty target.
- 4-byte nonce
- 32-byte previous block hash
- 32-byte merkle root

BITCOIN BLOCK STRUCTURE



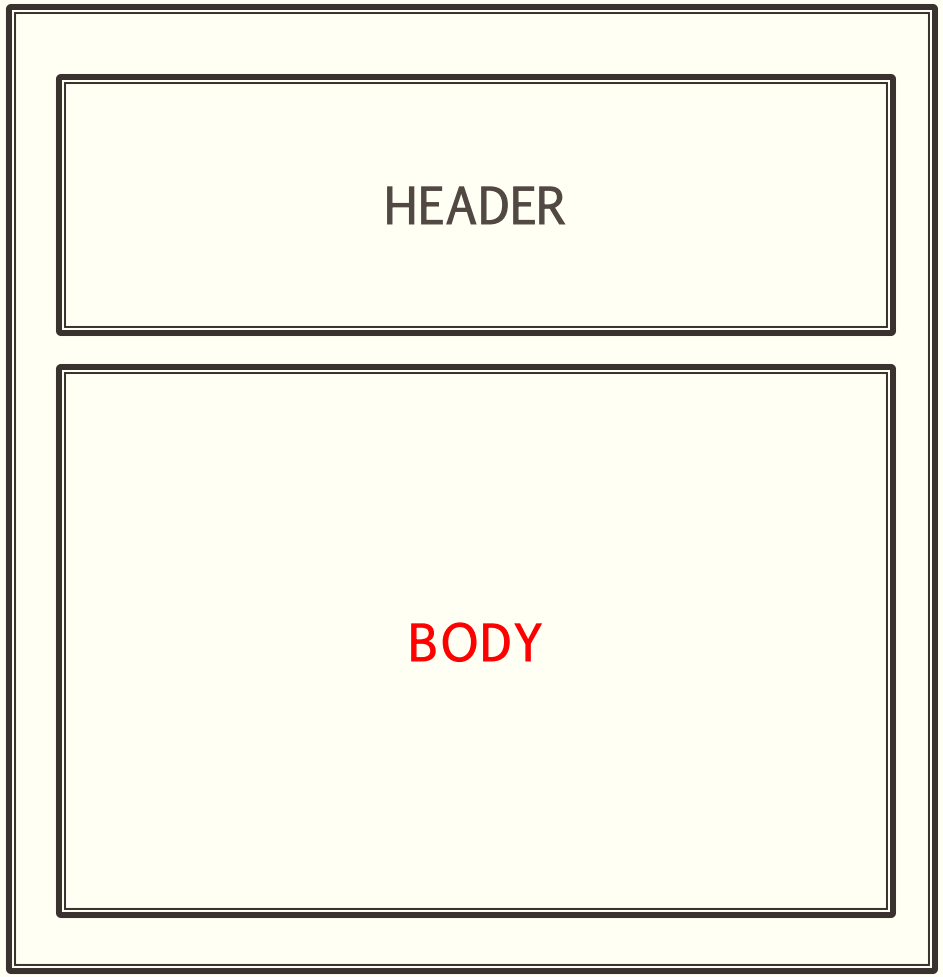
- 4-byte version.
- 4-byte timestamp.
- 4-byte difficulty target. PoW
- 4-byte nonce
- 32-byte previous block hash
- 32-byte merkle root

BITCOIN BLOCK STRUCTURE



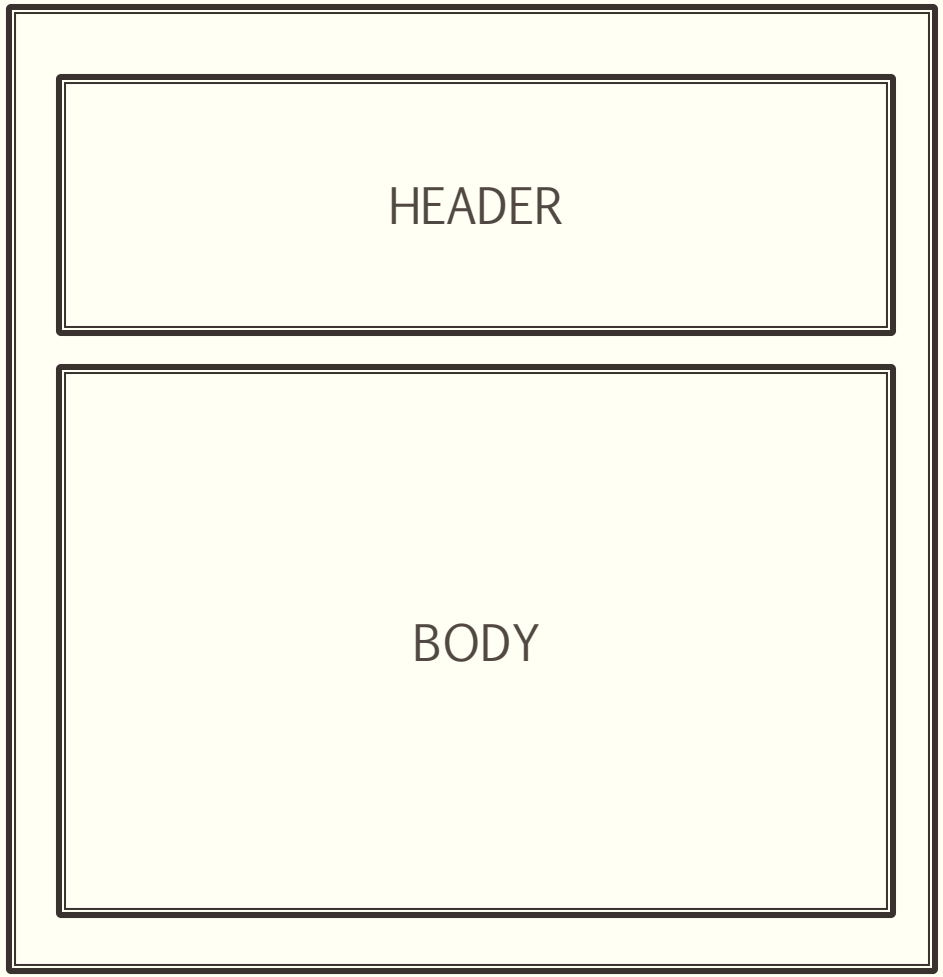
- 4-byte version.
- 4-byte timestamp.
- 4-byte difficulty target.
- 4-byte nonce
- 32-byte previous block hash
- 32-byte merkle root transactions

BITCOIN BLOCK STRUCTURE



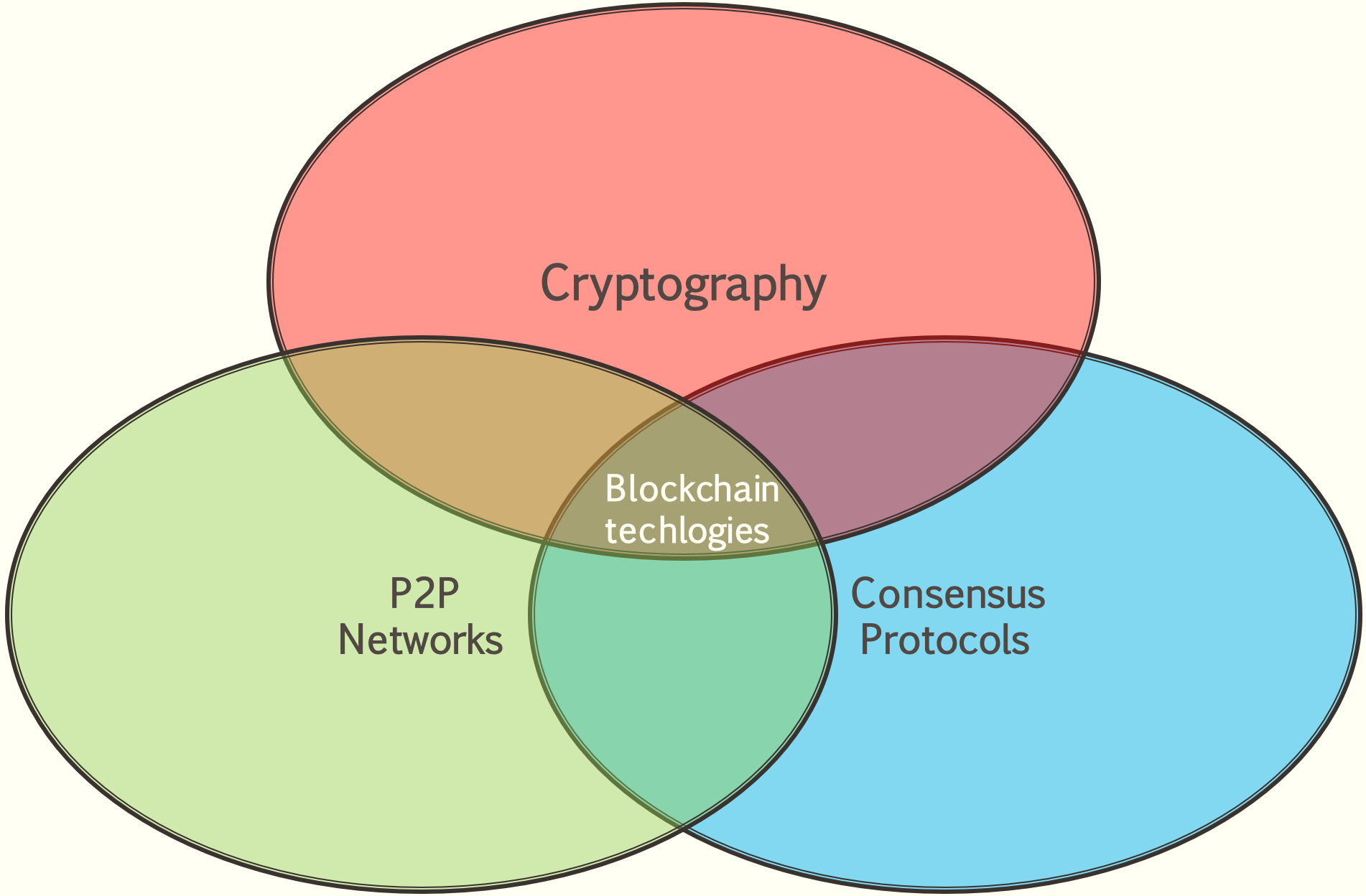
- 4-byte block size.
- 1-9 bytes transaction count.
- variable transactions.

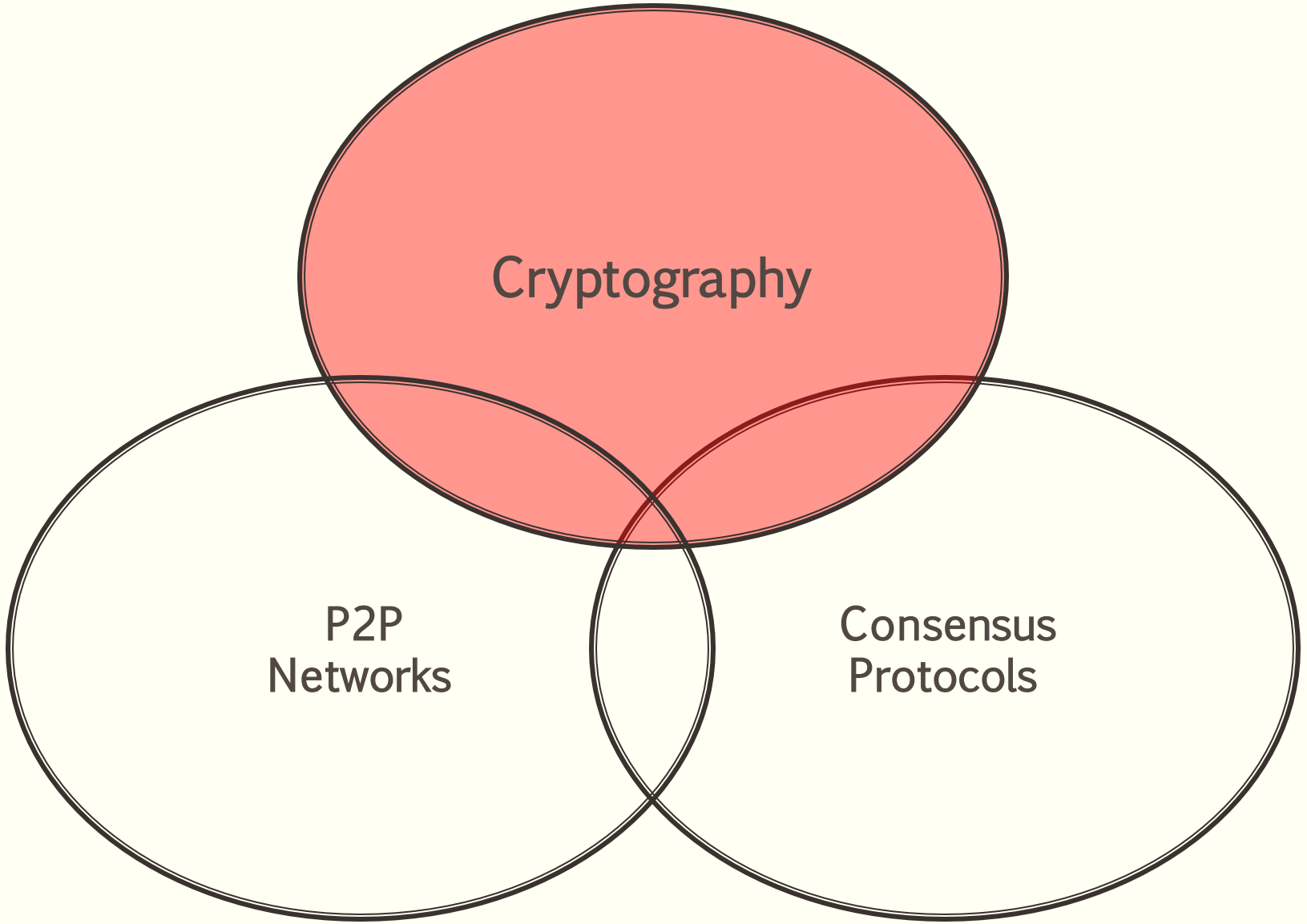
BITCOIN GENESIS BLOCK



- First block in the blockchain.
- **Coinbase** – transactions without input
- Hidden message: *The Times **03/Jan/2009** Chancellor on brink of second bailout for banks.*
- Height 0.
- Reward 50BTC to Nakamoto, unusable.

WHAT IS BLOCKCHAIN


Cryptography
Blockchain
techlogies
P2P
Networks
Consensus
Protocols

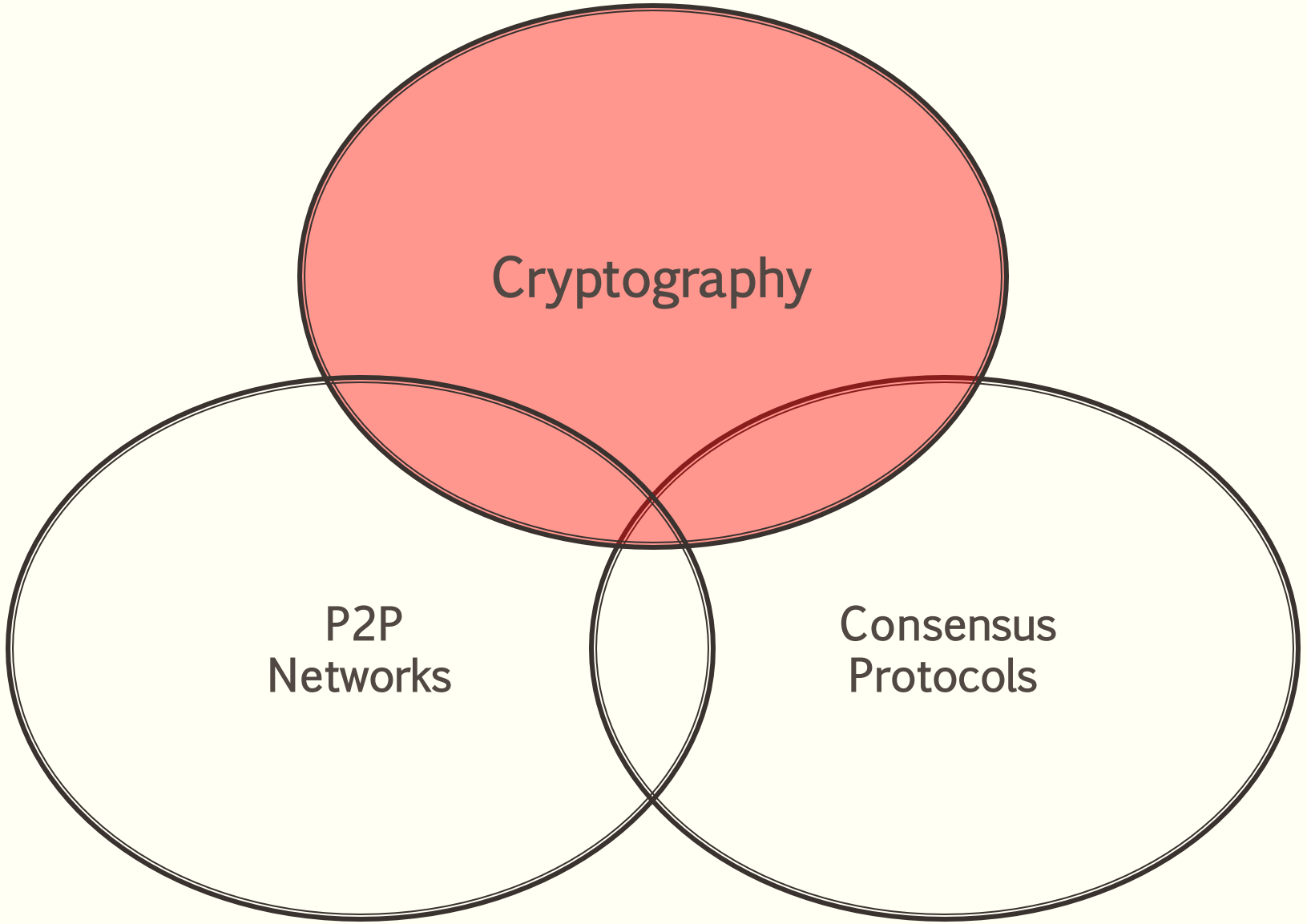


public key cryptography
and
cryptographic hash function.

users encode/deocde transactions using a pair of keys: private key/public key.

signature sig=**sign**(private_key, message)

boolean ok=**verify**(public_key, signature, message)



public key cryptography
and
cryptographic hash function
and
cryptographic hash function.

plain text is encrypted using a cipher to generate a hash value of fixed length.

hash(message)

preimage resistance,
collision resistance

stored in hash-trees
used as commit-reveal scheme

Bitcoin UTXO model

- Unspent transaction output
- Bitcoin Balance = sum of the unspent transaction outputs “owned”

  pay with previous unspent transaction outputs

  “change” is considered unspent transaction output locked to payer himself

- no double spending:

  total input of a transaction = equal total output + fees

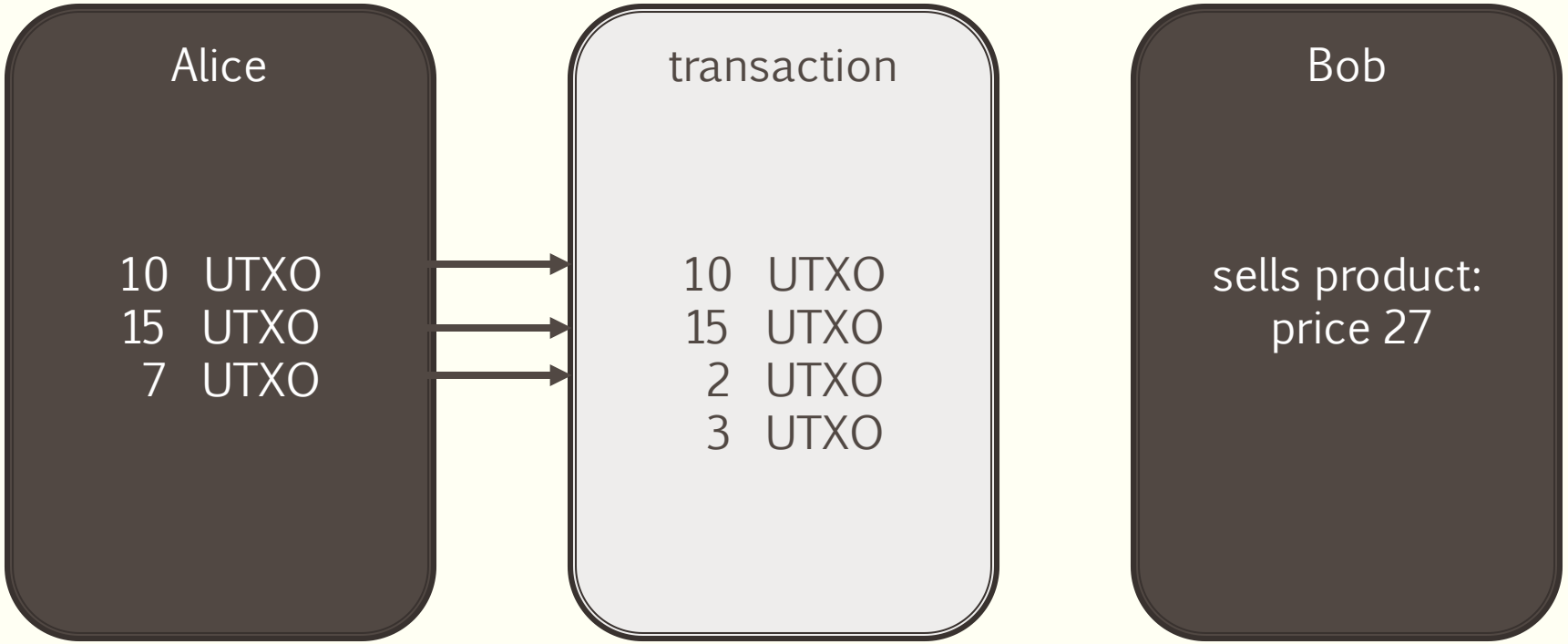
  in a transaction coins are consumed and replaced with new ones.

  new coins are also created by mining

  transactions without input: **Coinbase transaction** -- miners reward!

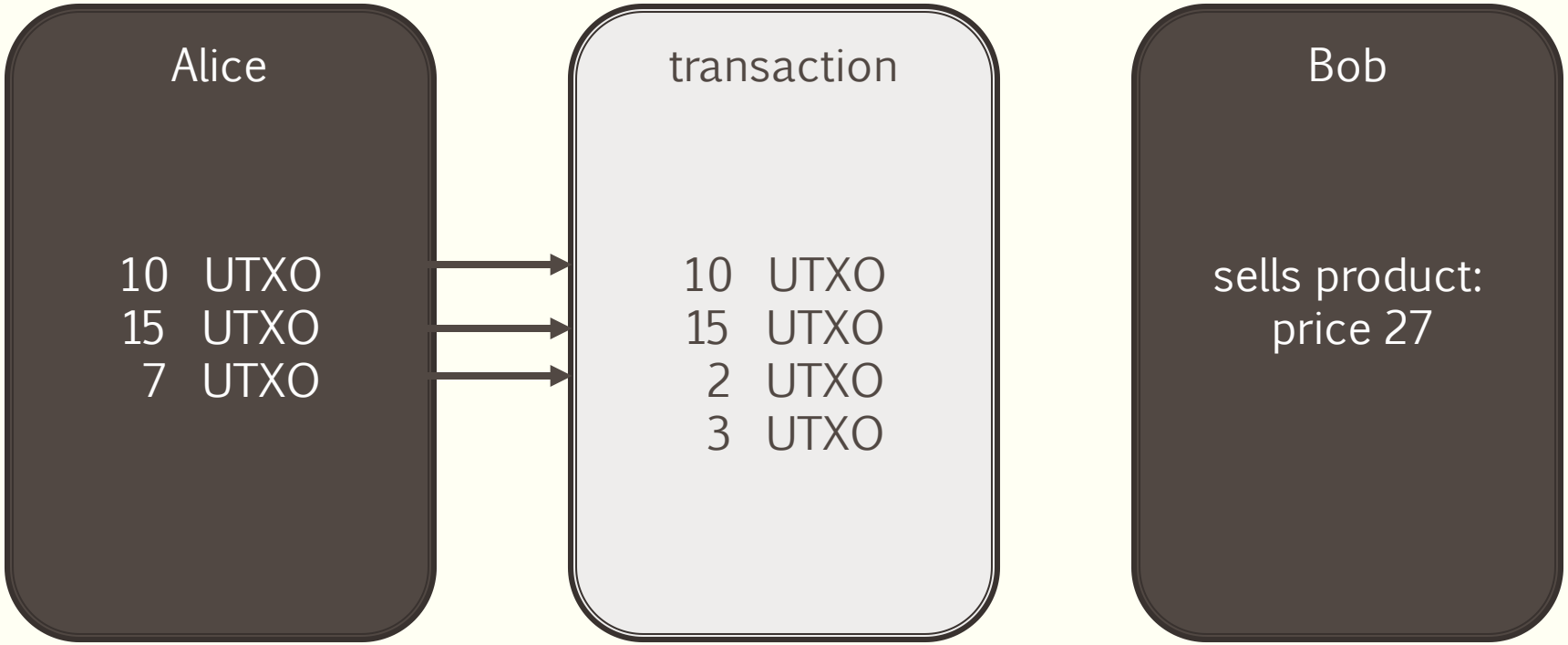
Bitcoin UTXO model

Bitcoin UTXO model

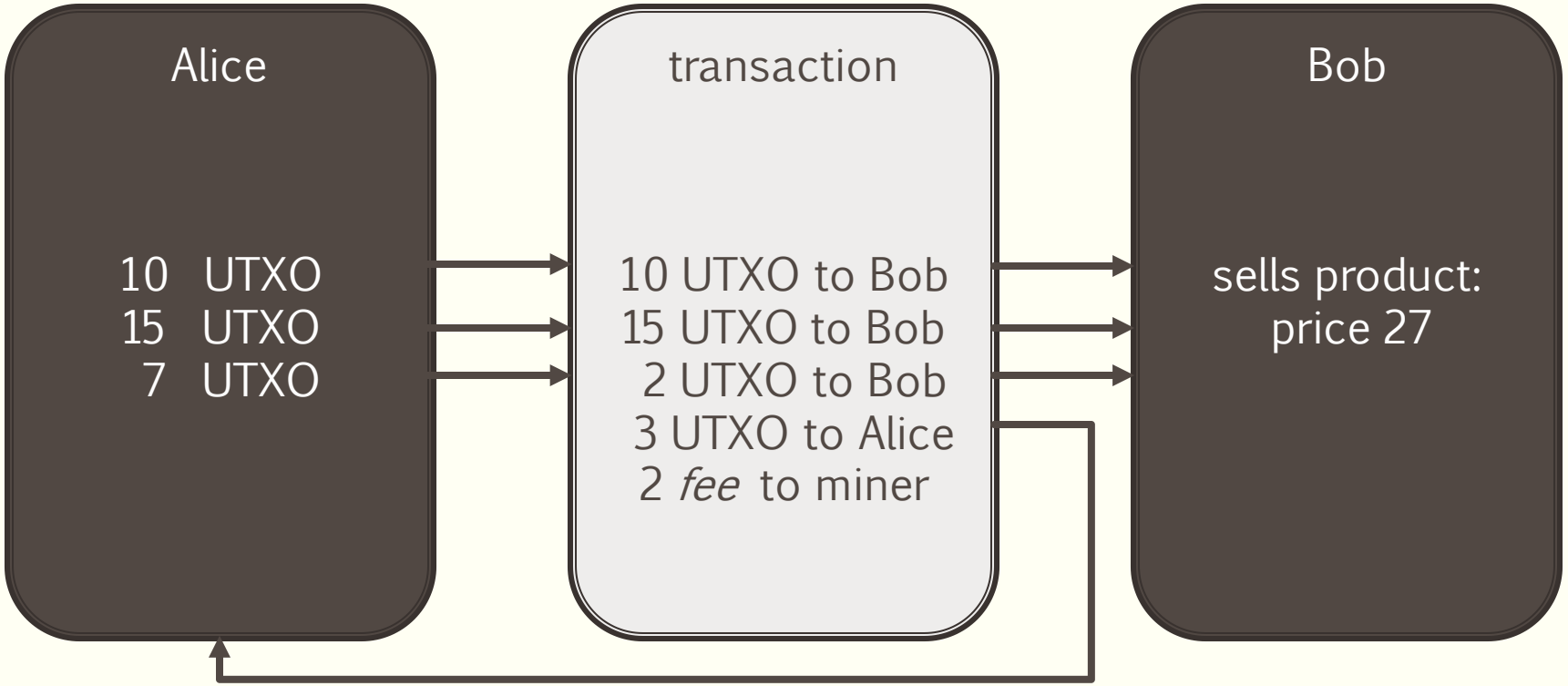


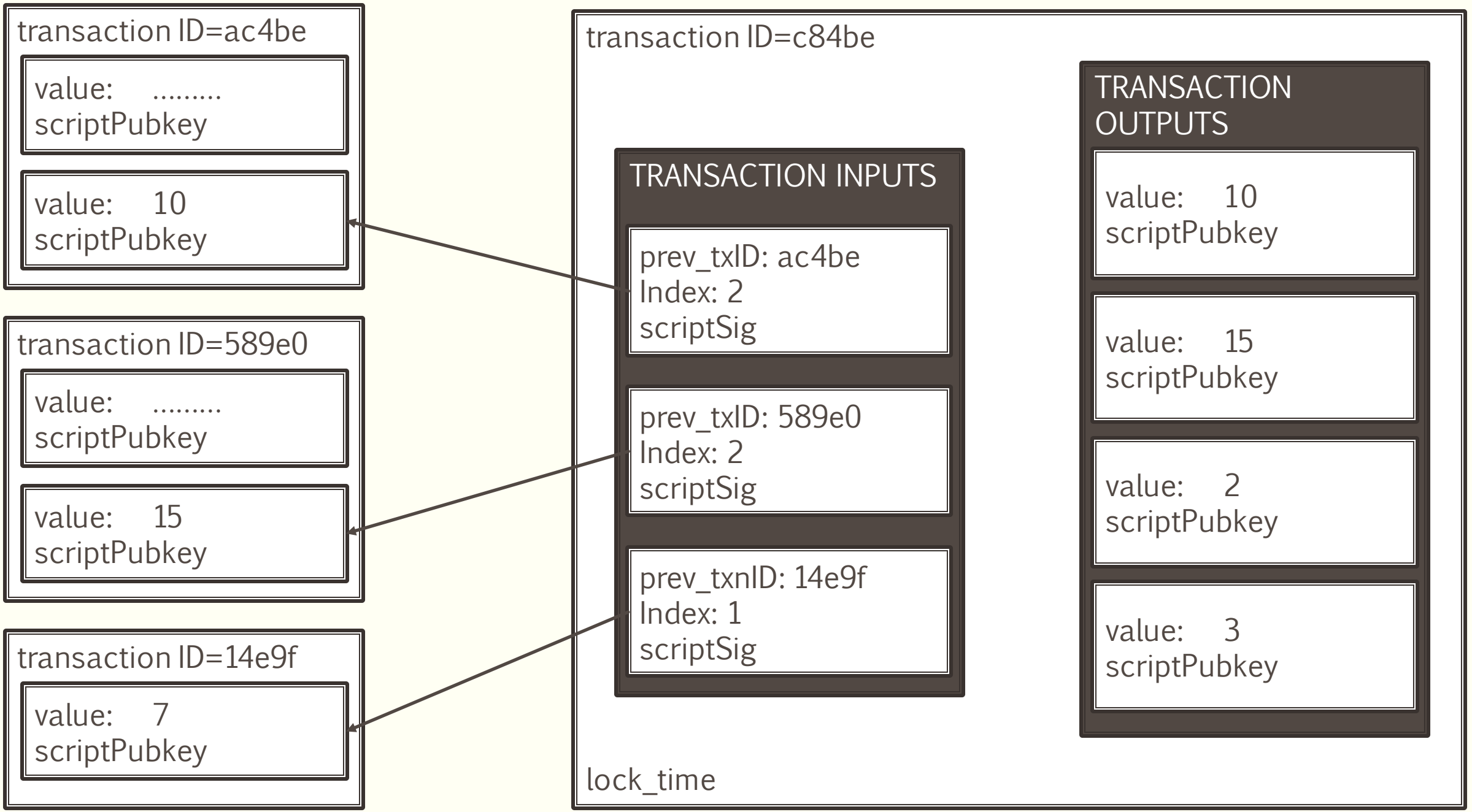
Bitcoin UTXO model

multiple inputs
coins have different values
a coin may be spent only once



Bitcoin UTXO model




transaction ID=ac4be
value: .....
scriptPubkey
value: 10
scriptPubkey
transaction ID=589e0
value: .....
scriptPubkey
value: 15
scriptPubkey
transaction ID=14e9f
value: 7
scriptPubkey
transaction ID=c84be
TRANSACTION INPUTS
prev_txID: ac4be
Index: 2
scriptSig
prev_txID: 589e0
Index: 2
scriptSig
prev_txnID: 14e9f
Index: 1
scriptSig
TRANSACTION OUTPUTS
value: 10
scriptPubkey
value: 15
scriptPubkey
value: 2
scriptPubkey
value: 3
scriptPubkey
lock_time

Bitcoin UTXO model

- Unlocking Script: **scriptSig**
  - include receiver public key hash
  - instructions (op_codes) to verify public key

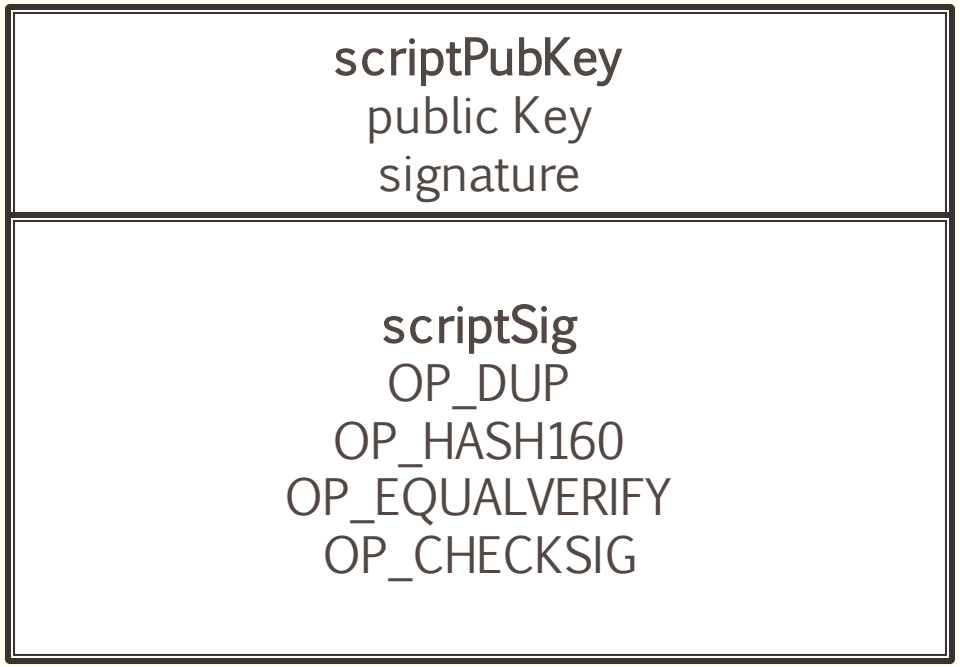
- Locking Script: **scriptPubKey**
  - Include signature and public key

- P2PKH pay to public key hash, example of simple smart contract, code that is executed on blockchain

- ECDSA signature

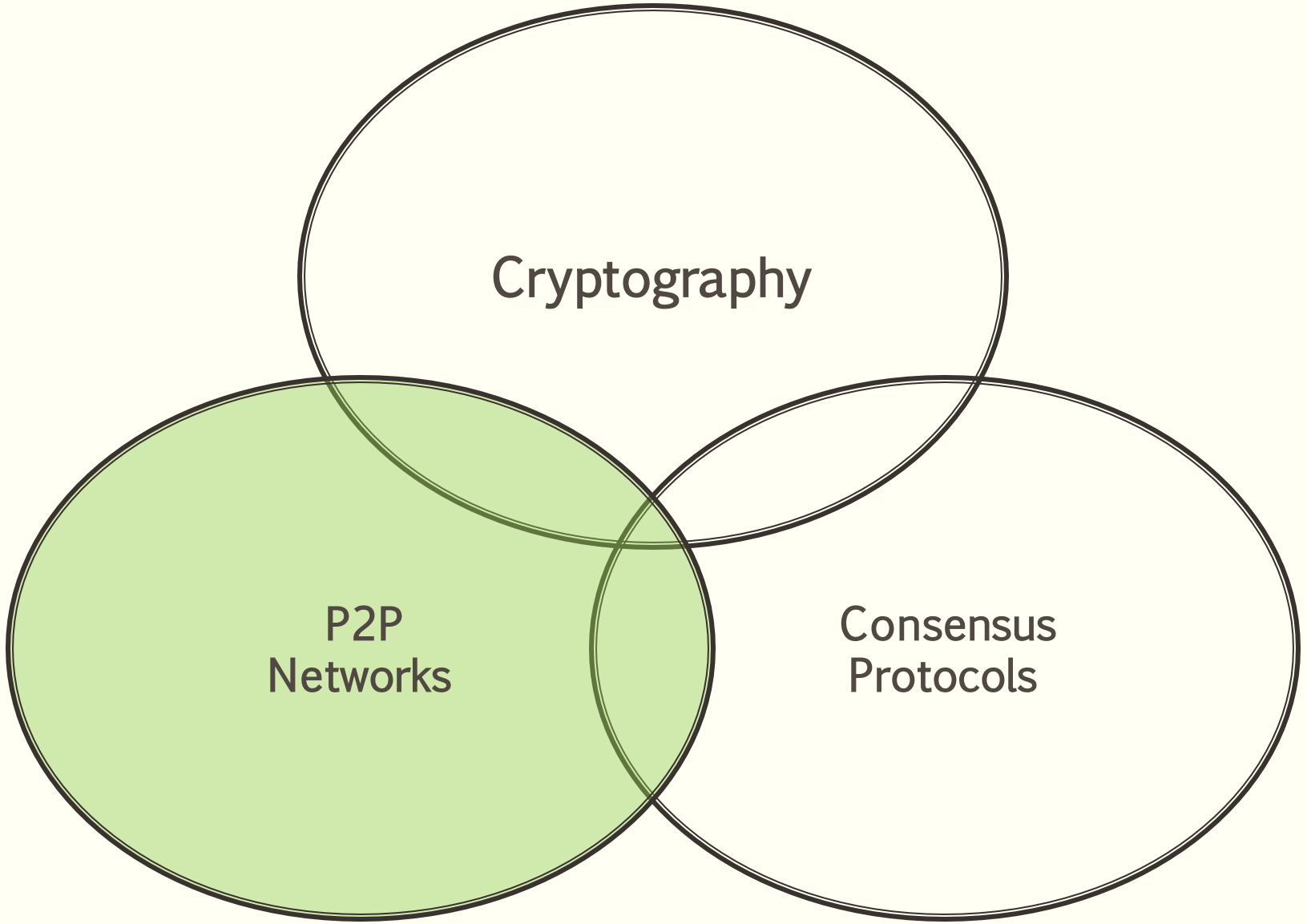
Bitcoin UTXO model

- P2PKH pay to public key hash



UTXO model

- Complex
- Verify transactions in parallel,
  - a UTXO affected by only one transaction
- Privacy (wallets)
- Limited smart contract capabilities



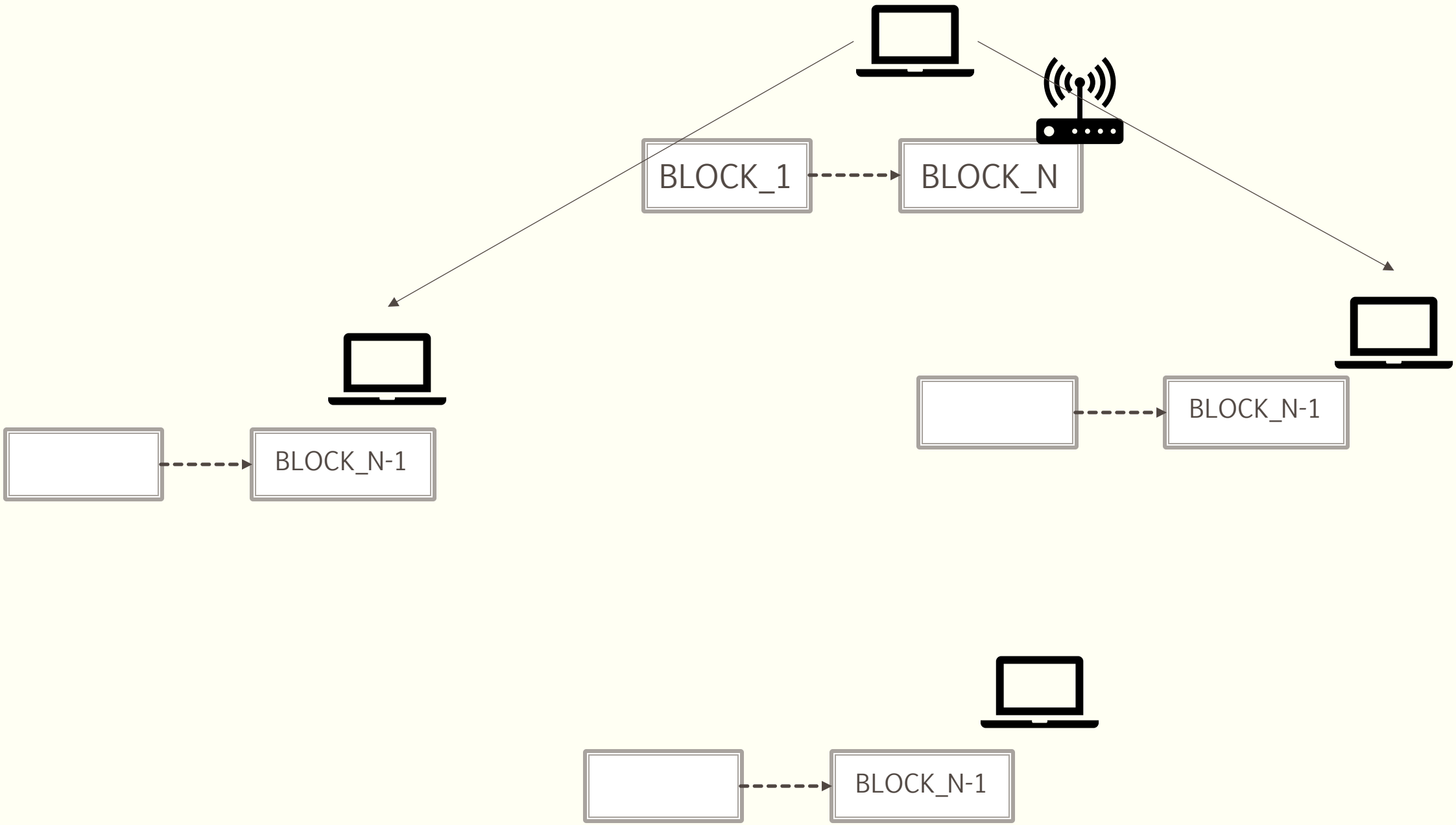
P2P architecture
and
cryptographic hash function.

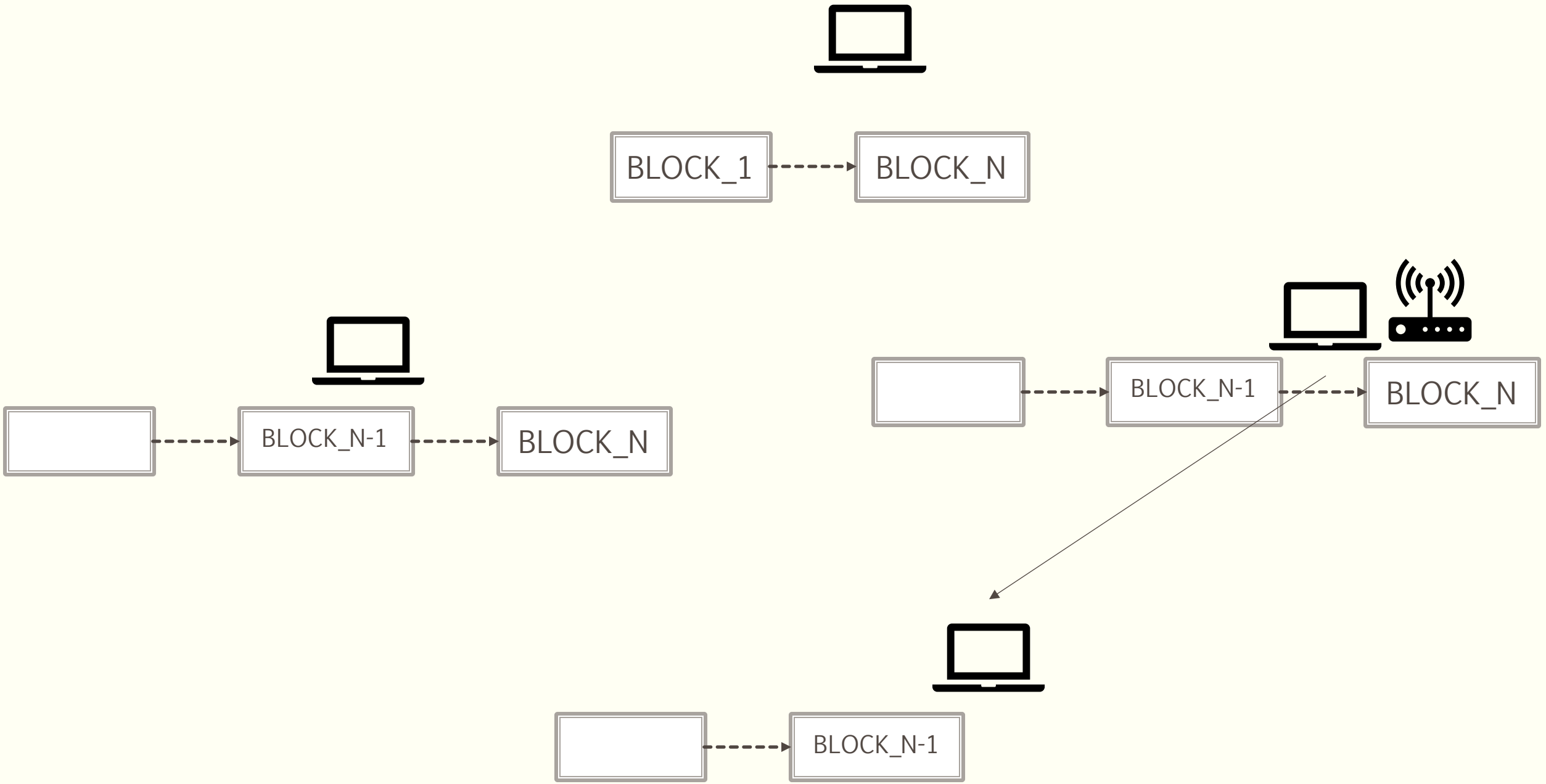
Full nodes download and verify every block.

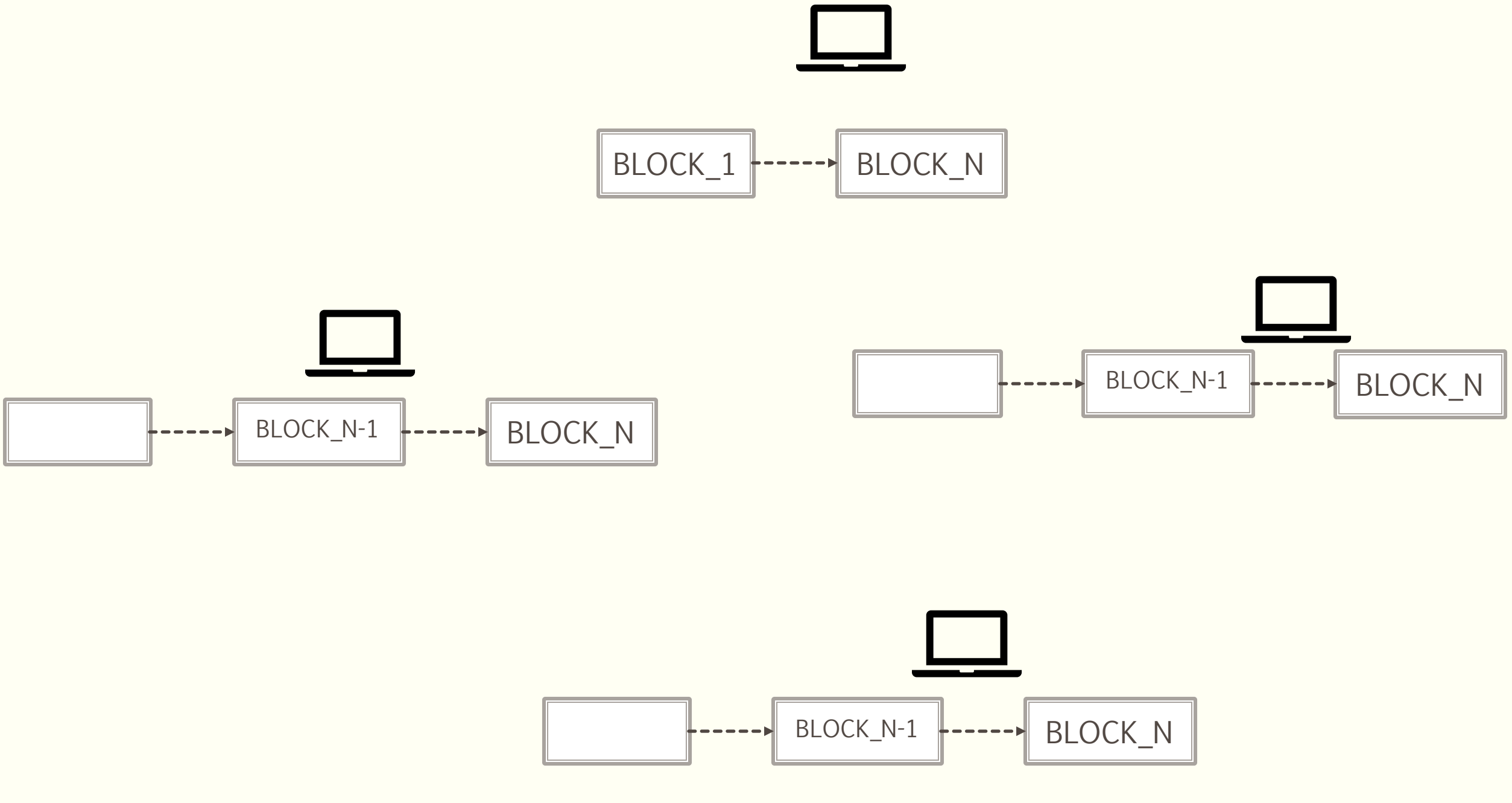
Newly joined nodes query **DNS seeds** to discover full nodes that accept connections.

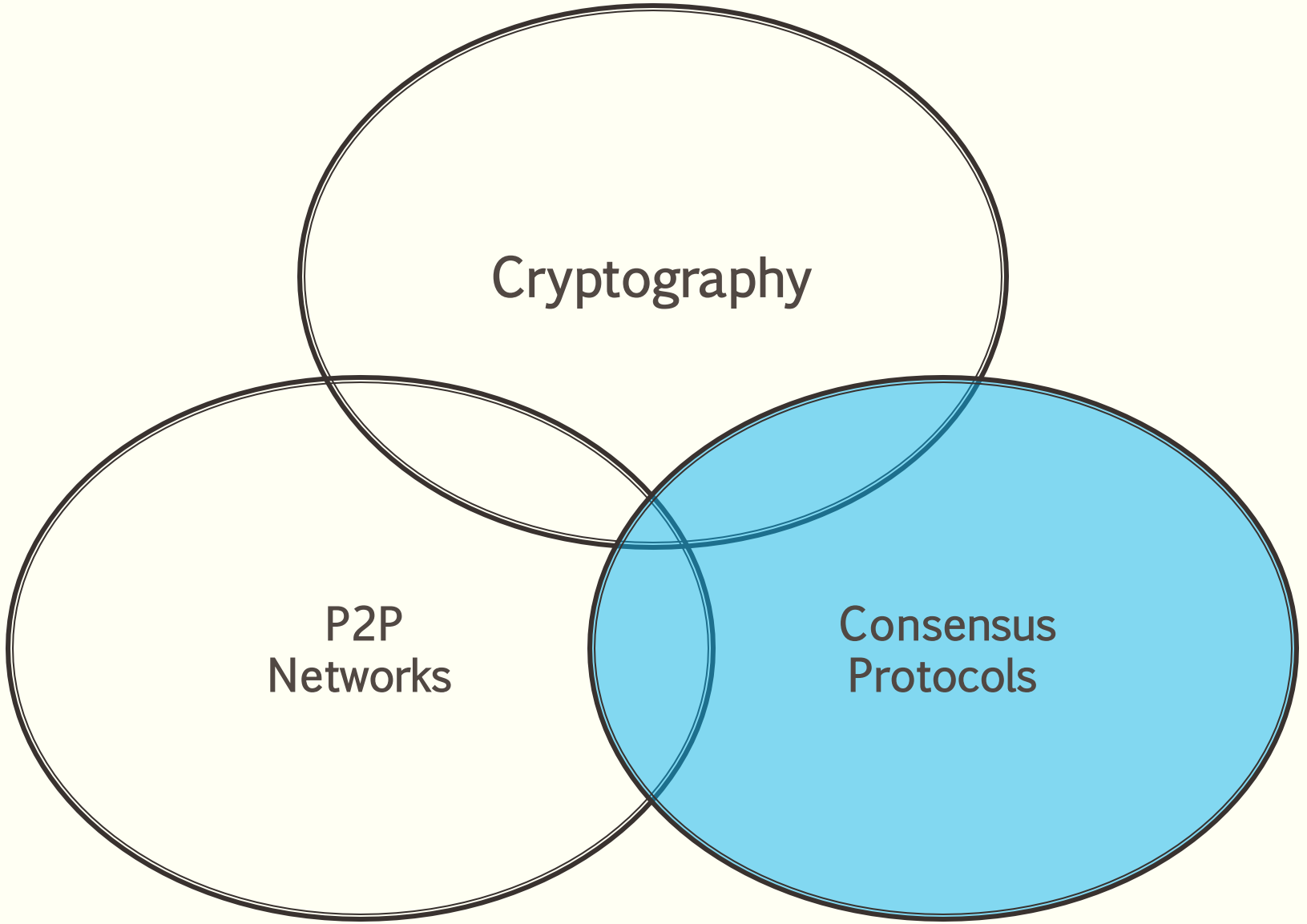
Initial Block Download: Before a full node can validate transactions, it must download and validate all blocks from block 1.

Block Broadcasting when a miner discovers a new block, it broadcasts the new block to its peers


BLOCK_1
BLOCK_N
BLOCK_N-1
BLOCK_N-1
BLOCK_N-1


BLOCK_1
BLOCK_N
BLOCK_N-1
BLOCK_N
BLOCK_N-1
BLOCK_N
BLOCK_N-1


BLOCK_1
BLOCK_N
BLOCK_N-1
BLOCK_N
BLOCK_N-1
BLOCK_N
BLOCK_N-1
BLOCK_N

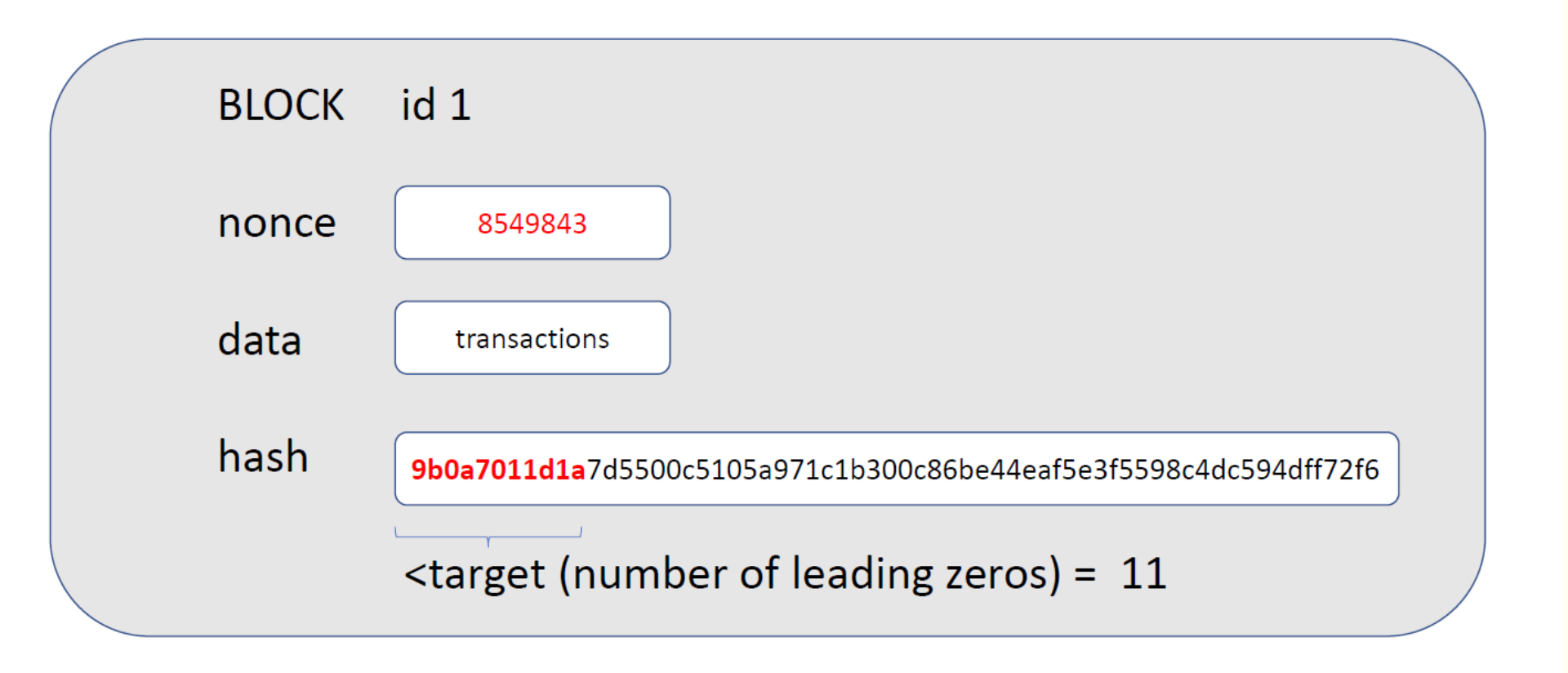


Consensus protocol
and
cryptographic hash function.

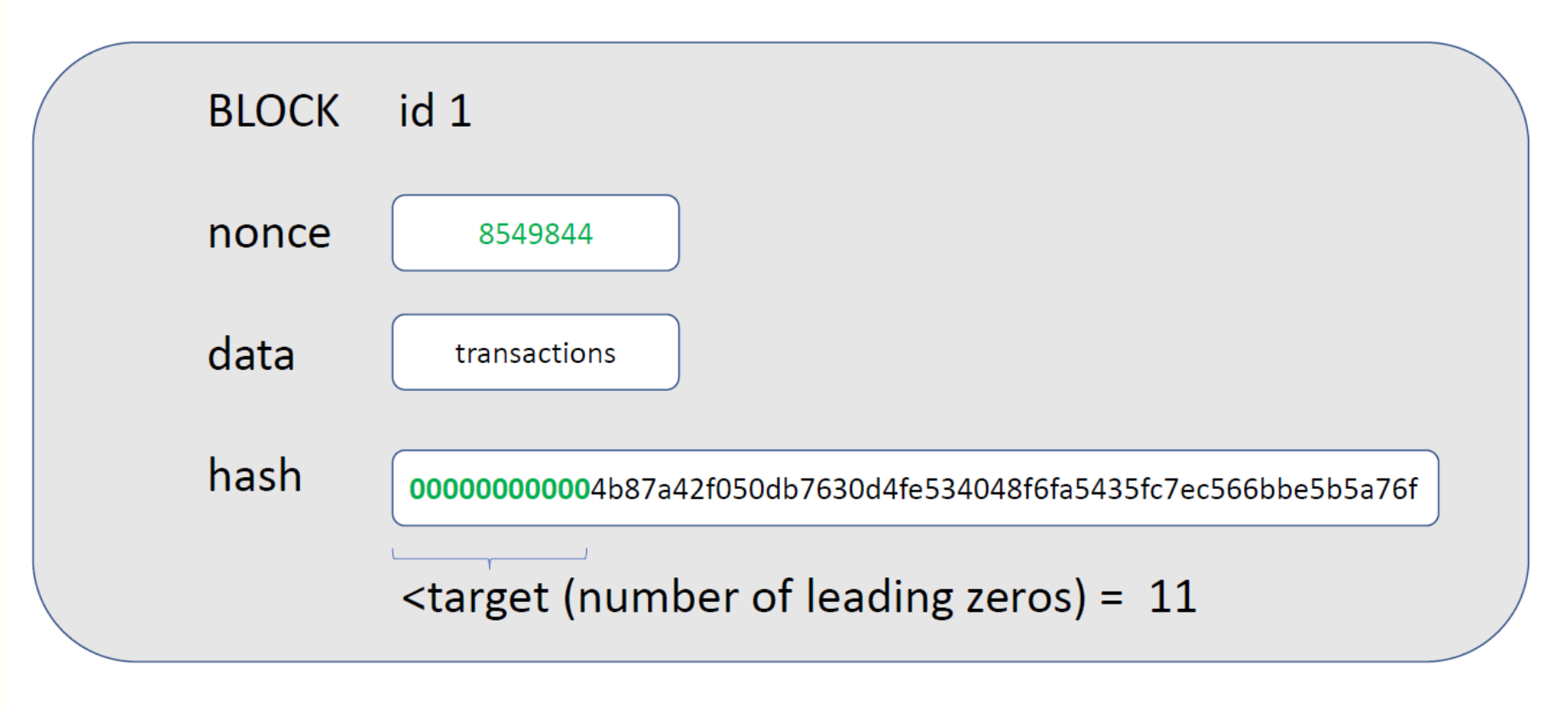
agree on some value,
leader election,
agree on transactions order ...

Ensures all participants agree on a unified transaction ledger without a central authority.

Block generation -- PoW Consensus

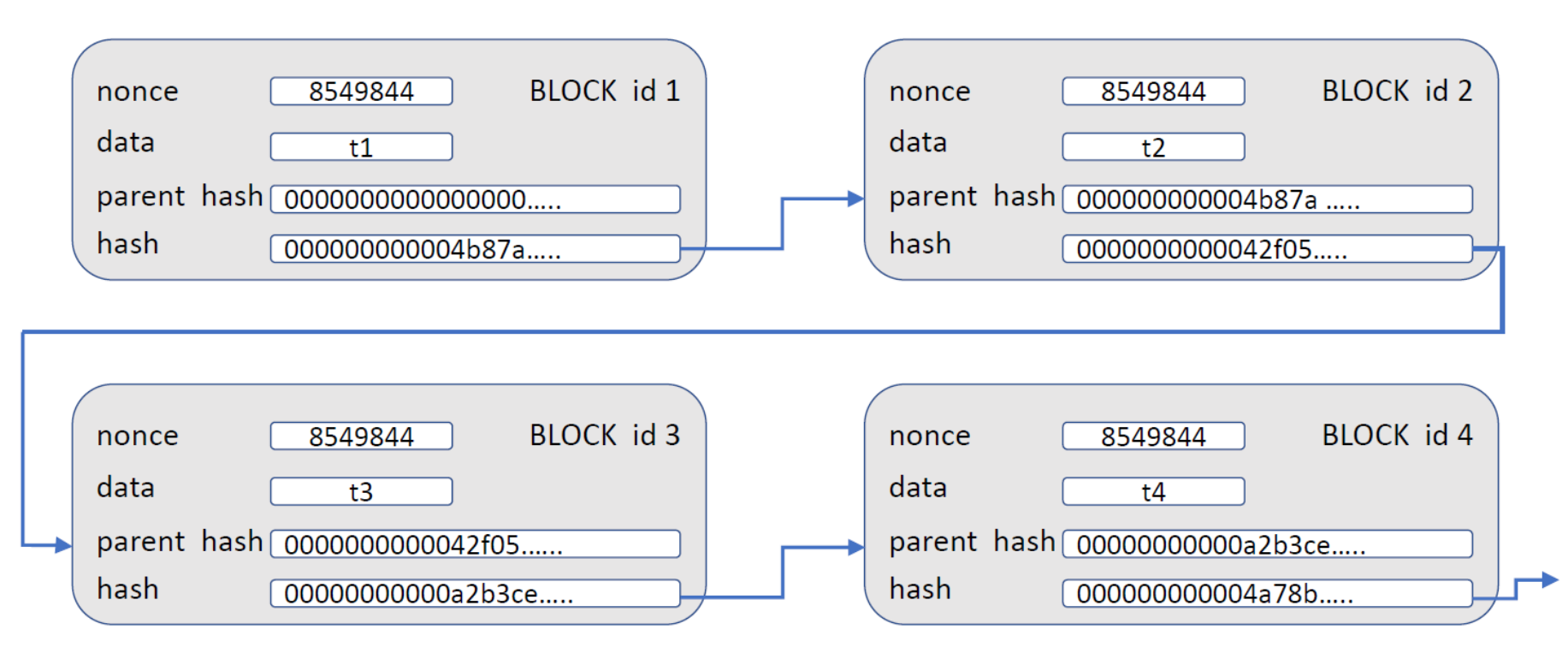


Block generation -- PoW Consensus



Block generation -- PoW Consensus

- hard to revert transactions, all hashes must be recalculated.



PoW keywords

- **nonce**: value which miners adjust so that the hash value of the block is less than target value.
- **block time**: expected/average time spent to mine a block.
- **difficulty**: measures how difficult it is to mine a block. (i.e find a nonce such that the block hash is valid)
- **target**: number of leading 0 in hash.

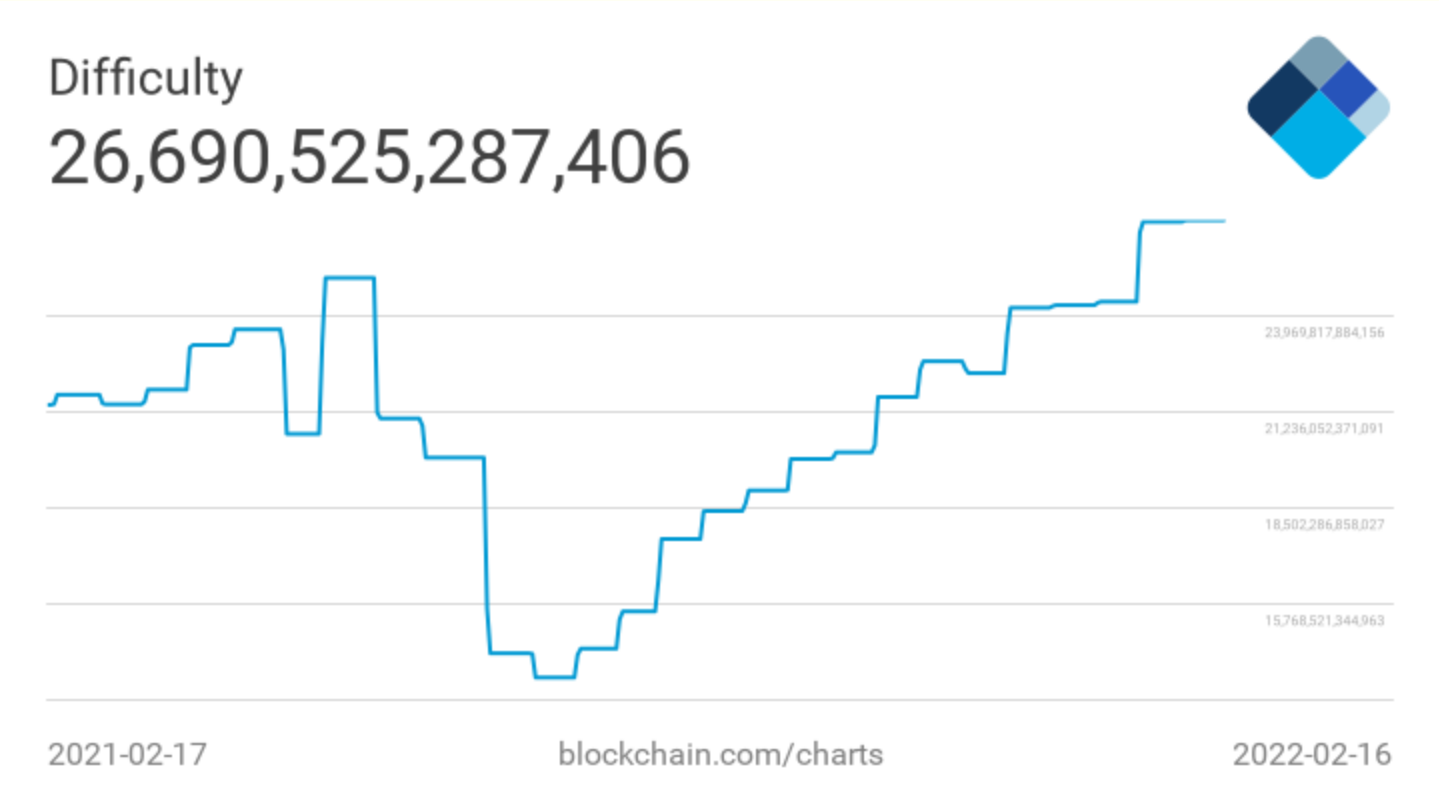
- If [average block time] > [expected block time] ➔ reduce difficulty
- If [average block time] < [expected block time] ➔ increase difficulty

PoW keywords

- If [average block time] > [expected block time] ➔ reduce difficulty
- If [average block time] < [expected block time] ➔ increase difficulty

Bitcoin	Ethereum
new_target = old_target / new_diff	new_target = old_target / new_diff
new_diff = old_diff x 10 min / AGV_TIME(2016 blocks)	bl_time = crt_bl_timestamp - parent_bl_timestamp
	new_diff = parent_block_diff + (parent_block_diff // 2048) * max(1, bl_time // 10), 99) + int(2**((current_block_number // 100000) 2))
10 min	30sec

Block difficulty



https://www.blockchain.com/charts/difficulty
https://etherscan.io/chart/difficulty

Bitcoin consensus protocol

- Block generation: PoW find nonce, hash satisfies a difficulty target
- Block propagation: gossiping any block (received or locally generated) should be *advertised* to peers and *broadcast*
- Block validation: check block header and transactions
- Longest-chain rule: Blocks should always extend the longest chain.
- Incentives/Rewards: coinbaise transactions.

Bitcoin consensus transaction rules

- sum(inputs) >= sum(outputs)
- [scriptSig scriptPubKey] true
- output has not already been spent
- lock_time minimal time before which the transaction cannot be accepted into block.

BLOCKCHAIN APPLICATIONS

Token Systems

- Company stock assets, coupons, incentives etc.
- Easy to implement, example of transaction: A sends x unit to B, provided that A has at least x unit in its balance before the transaction.
- Ethereum standards ERC-20, ERC-721 (NFTs)

Identity Systems

- DNS system, Namecoin, email authentication.
- Implement as a key(name)-value(data) database stored on blockchain network. Owner may change *data* associated with *name* or transfer ownership.
- Ethereum standard ERC-721 (NFTs)

Decentralized Autonomous Organizations

- Transparent rules, not influenced by a central authority
- Members have the right to spend funds.
- Members collectively decide to add or remove members.
- Controlled by smart contracts

Supply management

- Tracking environmental conditions
- Detect unethical suppliers and counterfeit products
- Endorsement of the Forestry Certification

https://fsc.org/en/innovation/blockchain “permissioned” private blockchain ledger platform designed to verify materials trade compliance across FSC supply chains.

Taxonomy

	Permissionless	Permissioned
anonymity	yes	no
number of nodes	large number of nodes	fewer nodes
security	high level of security	vulnerable
processing times	long	short

	PUBLIC	PRIVATE	CONSORTIUM
ownership	public	Controlled by a single organization	Group of organizations
centralization	decentralized	Partially decentralized	Partially decentralized
examples	Ethereum	Hyperledger	supply chain sector

Bibliography

- Ethereum yellow paper: https://ethereum.github.io/yellowpaper/paper.pdf
- Ethereum white paper: https://ethereum.org/en/whitepaper/
- Blind signatures for untraceable payments, David Chaum http://www.hit.bme.hu/~buttyan/courses/BMEVIHIM219/2009/Chaum.BlindSigForPayment.1982.PDF
- Hashcash A denial of service counter measure, Adam Back http://www.hashcash.org/papers/hashcash.pdf
- Mastering Bitcoin, Andreas M. Antonopoulos, O'Reilly Media, Inc. https://www.oreilly.com/library/view/mastering-bitcoin/9781491902639/ch07.html

- https://commons.wikimedia.org/wiki/File:Bitcoin_Block_Data.png
- https://www.balloonhq.com/photos/ts2019/