Blockchain for Business

Understanding blockchain and how it creates business value

Marco Comuzzi, UNIST (Korea)
Paul Grefen, TU Eindhoven and Eviden (The Netherlands)

Tutorial at EDOC 2023 - October 31st, 2023

Your speakers today…

What blockchain is not?

A) A technology for implementing enterprise applications

B) A technology for implementing public and private cryptocurrency systems

C) A technology for improving the user trust in machine learning models

What is Bitcoin?

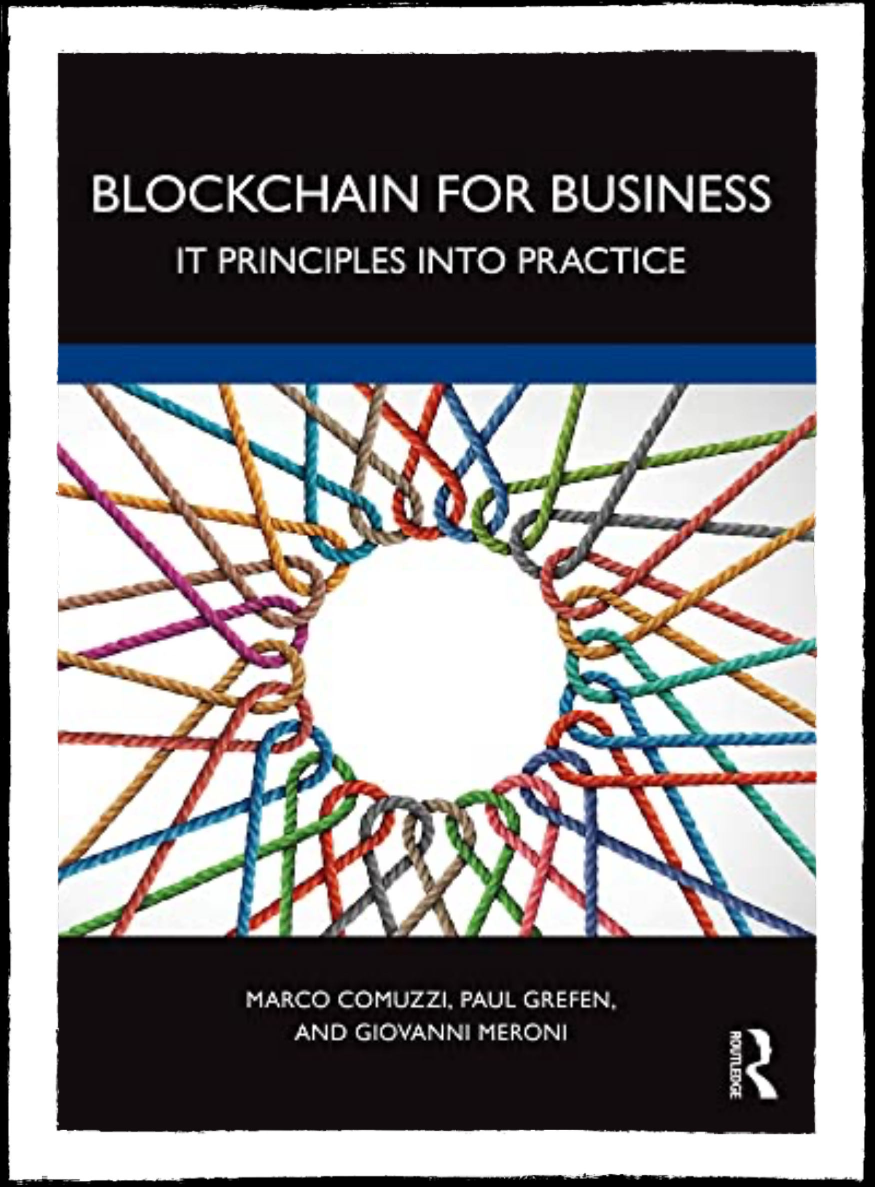
A) An asset for diversifying an investment portfolio

B) A blockchain system that can be extended with enterprise functionality

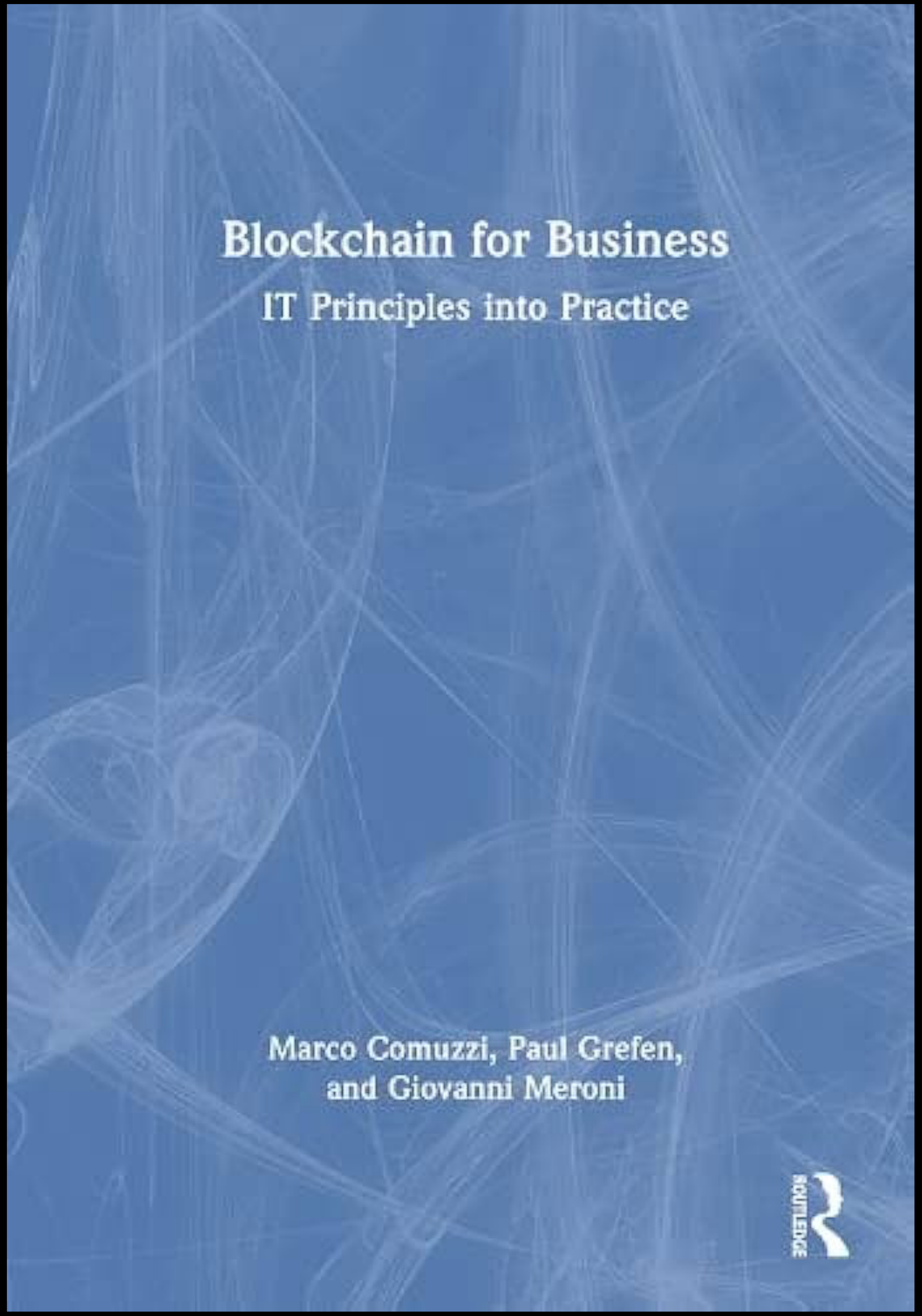
C) An open cryptocurrency system without financial intermediaries

Today’s plan

- Part 1
  - TBC TBC
  - TBC TBC
- Part 2
  - TBC TBC
  - TBC TBC


BLOCKCHAIN FOR BUSINESS
IT PRINCIPLES INTO PRACTICE
MARCO COMUZZI, PAUL GREFEN, AND GIOVANNI MERONI
ROUTLEDGE

Comuzzi, M., Grefen, P., & Meroni, G. (2023). Blockchain for Business: IT Principles into Practice. *Routledge*



Blockchain for Business

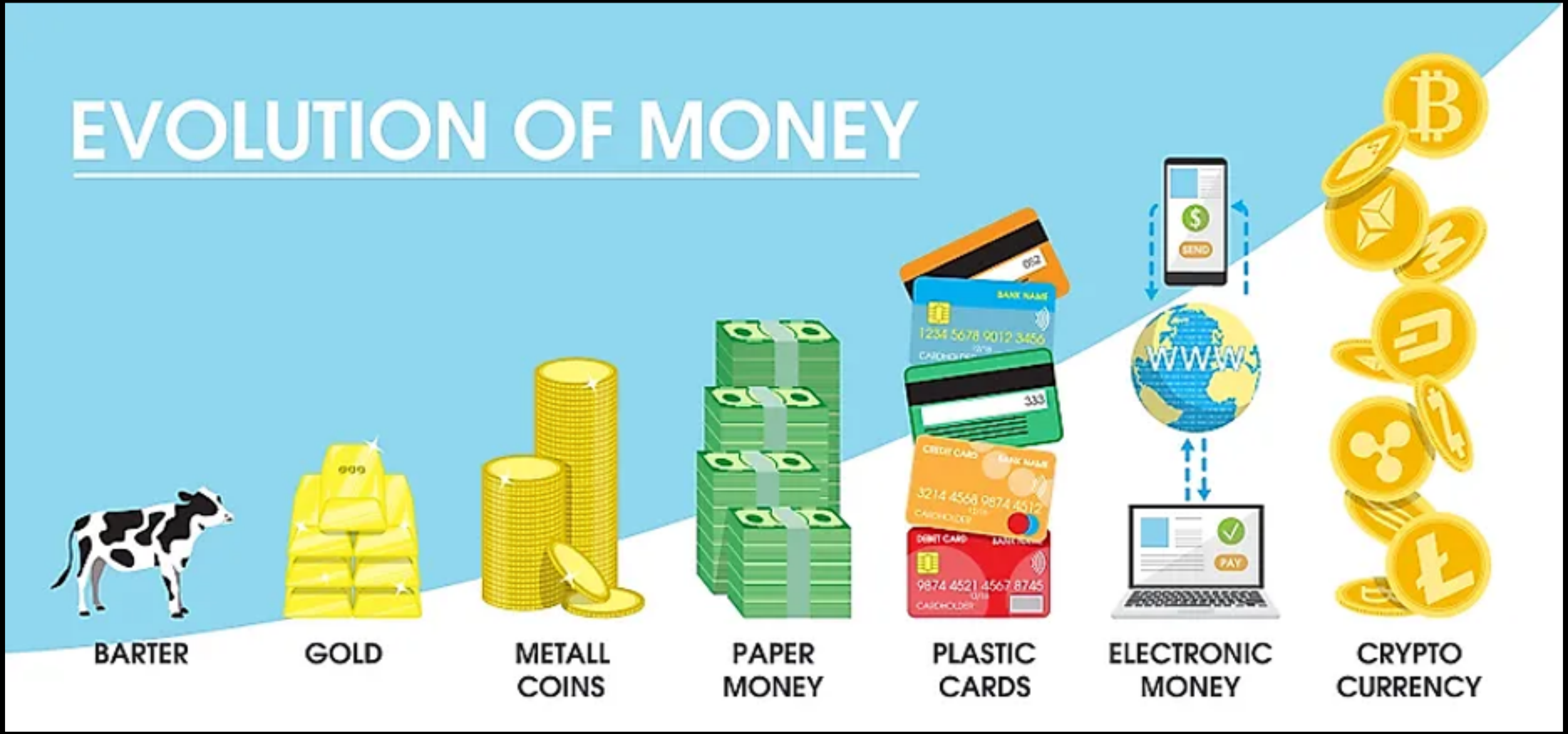
IT Principles into Practice

Marco Comuzzi, Paul Grefen, and Giovanni Meroni

ROUTLEDGE

What is blockchain?

A (very) brief history of money



There used to be no money at all!

The butcher and the fisherman trust that the meat and the fish that they are about to exchange have the same value

Commodity money

Value of coins given by the precious metal of which they are made

Everybody trusts the value of gold (scarce, hard to forge)

What gives value to ‘fiat’ currency?

A state, promising to always accept these banknotes from its citizens to pay for taxes

DESIGN

Money systems can be designed in different ways

TRUST

Value of money comes from the trust among agents in an economic systems

INTERMEDIARIES

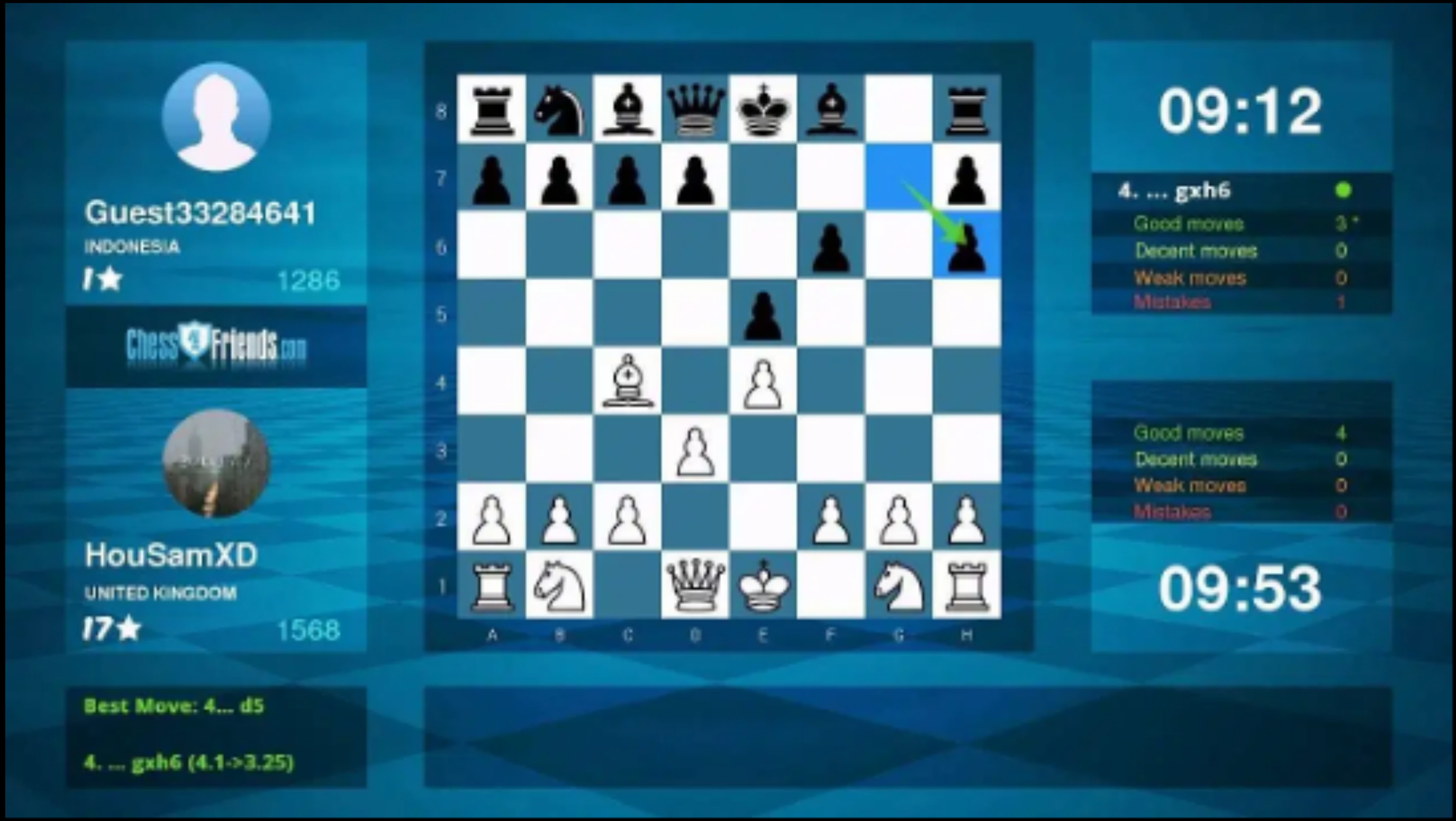
Can create the trust needed to give value to money

An online gaming platform

Users subscribe to platform

Platform stores the state of ongoing games

Platform authenticates players when needed



Best Move: 4... d5

4. ... gxh6 (4.1->3.25)

4. ... gxh6

Good moves	3 *
Decent moves	0
Weak moves	0
Mistakes	1

Good moves	4
Decent moves	0
Weak moves	0
Mistakes	0

TRUST

What if we could design a **technology** that

creates the **trust** needed by

a system and its users to operate safely,

without the need for any **intermediary**?

What is blockchain?

The technology behind Bitcoin and all cryptocurrencies

A technology that creates trust in a “trustless” world

Trustless world

A set of (economic) agents who do not trust each other and who need to exchange information…

… about money exchange, moves in a chess game … anything!

Thank you…but what is blockchain?

NETWORK

A set of computational nodes (peers) connected through the Internet

[P2P Network]

DATA STRUCTURE

A database replicated at each node of the network

[distributed ledger]

PROTOCOL

A set of rules for nodes to agree on the content of the database

[consensus mechanism]

Immutable Database

Data can only be appended to the database

Existing data cannot be modified

Existing data cannot be deleted

Let’s design two (!) blockhain systems

Blockchain for Chess (BC4C)

A new cryptocurrency: the EDOC Token (ETK)

BC4C: Blockchain for chess

P2P NETWORK

Each player is a node of the network

No other intermediary nodes

BC4C: Blockchain for chess

DISTRIBUTED LEDGER

Records the initial state of every game
(Always the same)

Records game creation and all the moves of both players in every game (transactions)

BC4C: Blockchain for chess

TRANSACTIONS

`CreateNewGame [gameId, counterPlayer]`: this type of transaction creates a new game between the originator of the transaction and another player, identified by the parameter counterPlayer.

`ConfirmGameCreation [gameId]`: this type of transaction confirms the creation of a game. It is issued by the counter player.

`Move[gameId, moveId, piece, new position, isCheckMate]`: This type of transaction specifies a move in a game. A move involves moving a piece to a new position on the board. A Boolean flag isCheckMate identifies whether the move leads to checkmate.

`ConfirmMove [gameId, moveId]`: This type of transaction is issued by a player to confirm the validity of a move issued by a counter player

BC4C: Blockchain for chess

CONSENSUS MECHANISM

Rule ID	Rule Specification
1	A game involves 2 players. A new game is proposed by one player and must be accepted by the opposite player to begin. When a game is accepted, an id is generated for it according to standard rules, e.g. “AliceBob3” for the 3rd game
2	A game starts with the standard configuration of a chess board (number/types of pieces and positions)
3	The players of a game take turns in making moves. The player who proposed a game moves first. A move is proposed by one player and must be accepted by the opposite player. A unique id for each move can be generated following simple
4	A game terminates when a checkmate move is proposed by one player and this is approved by the counterpart.

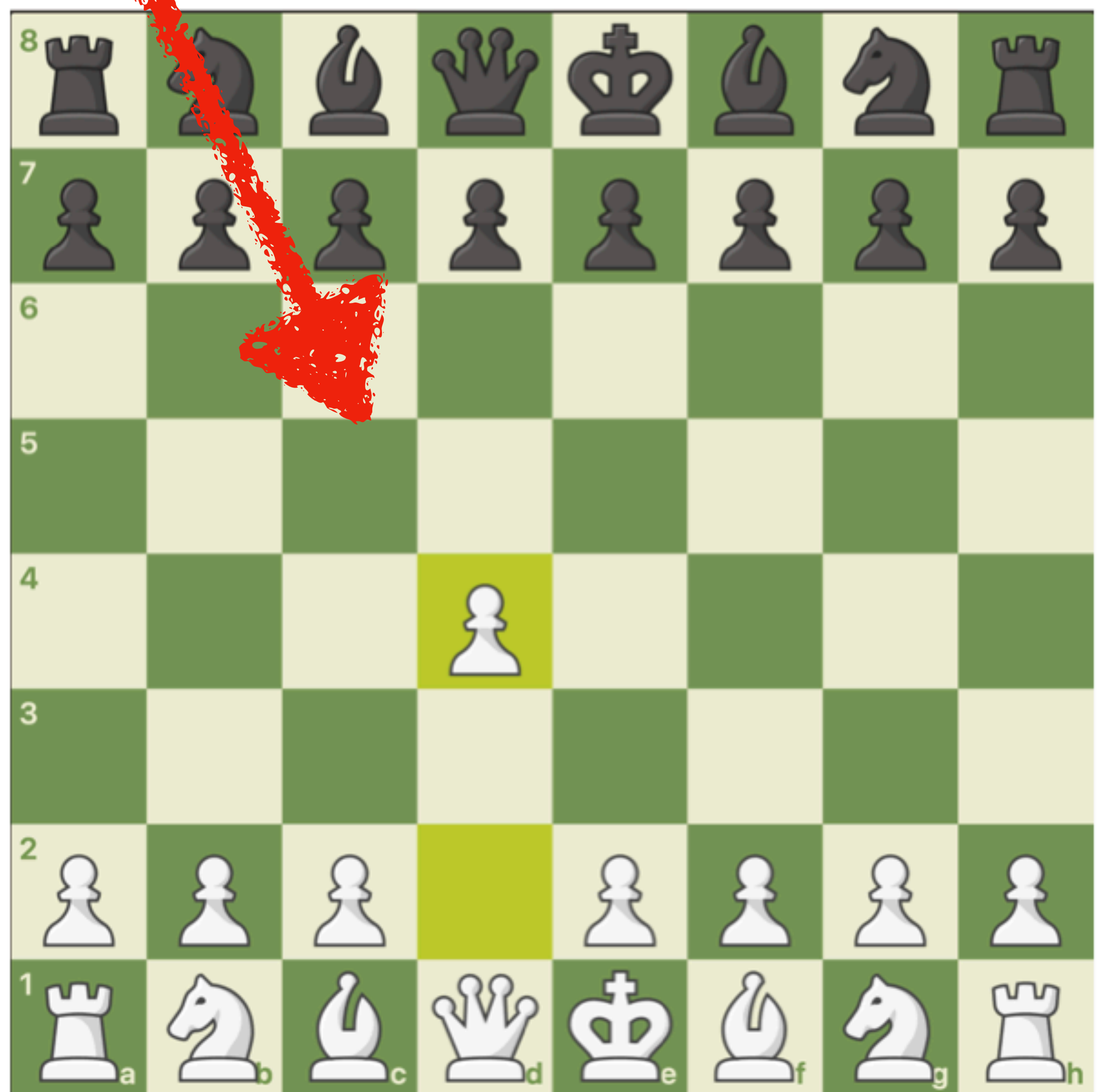
BC4C: Blockchain for chess

Content of the ledger (transactions)

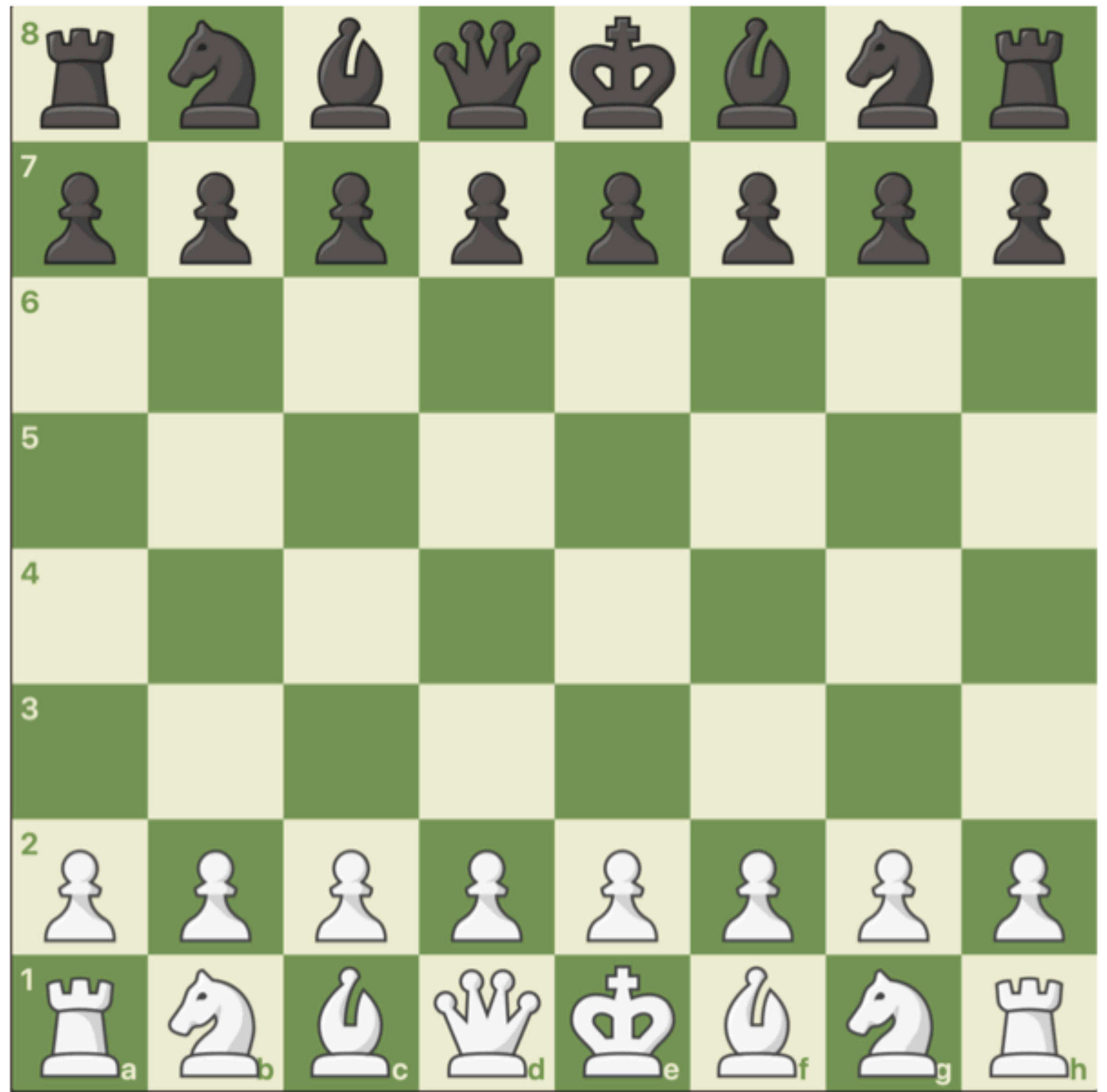
Transaction id	Transaction	Originator	Timestamp
0	Genesis of BC4C	BC4C	22-04-19 12:00:17
1	CreateNewGame[AliceCarol1, Carol]	Alice	22-04-20 09:00:03
2	CreateNewGame[BobCarol1, Carol]	Bob	22-04-20 09:00:45
3	ConfirmGameCreation[BobCarol1]	Carol	22-04-20 09:01:23
4	ConfirmGameCreation[AliceCarol1]	Carol	22-04-20 09:01:28
5	Move[BobCarol1, 1, pawn_h2, h3, false]	Bob	22-04-20 09:02:34
6	CreateNewGame[DaveAlice1, Alice]	Dave	22-04-20 09:02:48
7	Move[AliceCarol1, 1, pawn_d2, d4, false]	Alice	22-04-20 09:03:01
8	ConfirmMove[AliceCarol1, 1]	Carol	22-04-20 09:03:55
9	Move[AliceCarol, 2, pawn_a2, a3, false]	Carol	22-04-20 09:04:26

State of the System (games currently playing)

AliceCarol1



BobCarol1



There are currently two games being played: *AliceCarol1* and *BobCarol1*.

The game *DaveAlice1* has been proposed by Dave, but not confirmed by Alice, yet.

ETK: The EDOC Token

P2P NETWORK

Anybody needs cash, so anybody should be able to join (“public blockchain”)

No intermediary nodes (obviously)

ETK: The EDOC Token

A distributed database to store the balance of every node …

…or a list of all the transactions among the nodes from the inception of the system…

… or both

DISTRIBUTED LEDGER

ETK: The EDOC Token

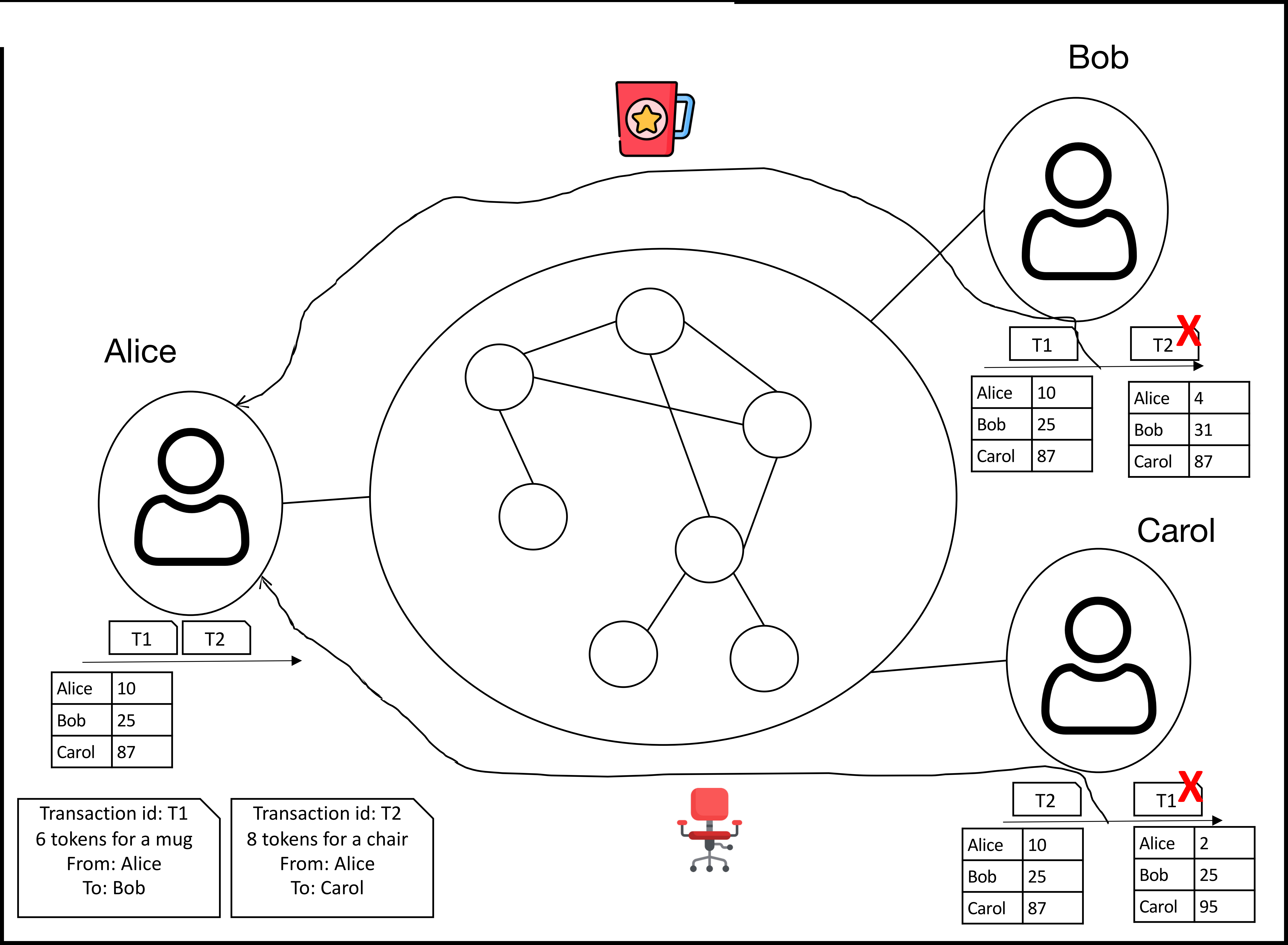
Nodes can only transfer to other nodes currency (tokens) that they own

For each transaction, the balance of the sender (recipient) is decreased (increased) of a certain quantity

… is it that easy?

CONSENSUS MECHANISM

ETK: double spending



Alice	10
Bob	25
Carol	87

Transaction id: T1
6 tokens for a mug
From: Alice
To: Bob

Transaction id: T2
8 tokens for a chair
From: Alice
To: Carol

Alice	10
Bob	25
Carol	87

Alice	4
Bob	31
Carol	87

Alice	10
Bob	25
Carol	87

Alice	2
Bob	25
Carol	95

In a P2P network we cannot guarantee that all nodes receive the transactions in the same order

Alice has only 10 tokens, but she can issue T1 and T2 very close in time…

… if Bob receives T1 before T2 (and Carol T2 before T1), then Alice may “double spend” her tokens

Consensus mechanism

Domain-specific rules —> “Transaction Validation rules”

BC4C: validation of moves
ETK: validation of currency transfers

General rules to avoid “double spending” (= double use of transactions)

Actual “consensus mechanism”: how to create an order of transactiosn agreed upon by all nodes
Example: Proof-of-Work in Bitcoin

Exercise
(Part A)

Exercise

(Check the leaflet)

- Work in groups of 5~6
- Read the business scenario; think how we can design a blockchain-based information system to support it; answer the following questions:
    - Who are the blockchain nodes?
    - What are the transaction types?
    - What are the consensus (transaction validation) rules?
    - List 2~3 concerns related to blockchain usage in this scenario

How can we build an immutable database?

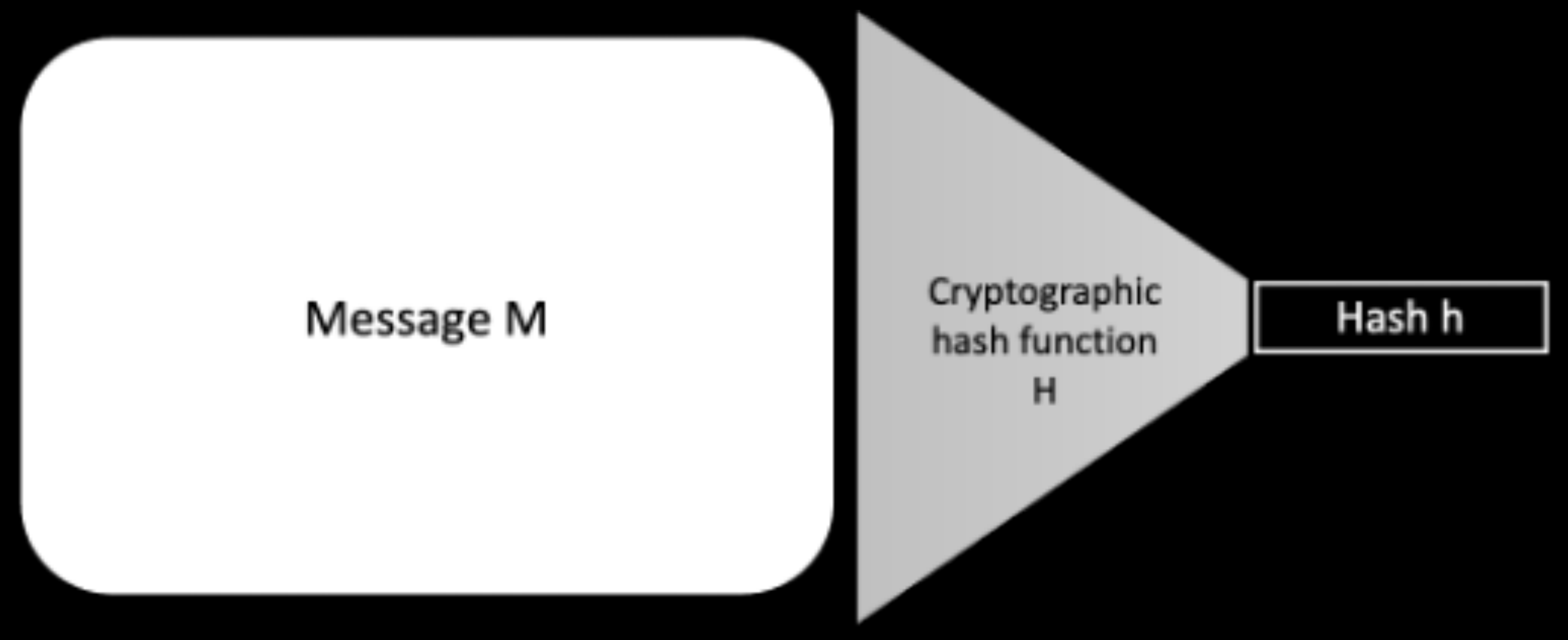
How can we identify the blockchain nodes?

(Blockchain seems very simple…)

Cryptographic Hashing: A Mathematical Function

FIXED OUTPUT SIZE

Size of hash is fixed, no matter how big or small is the input.

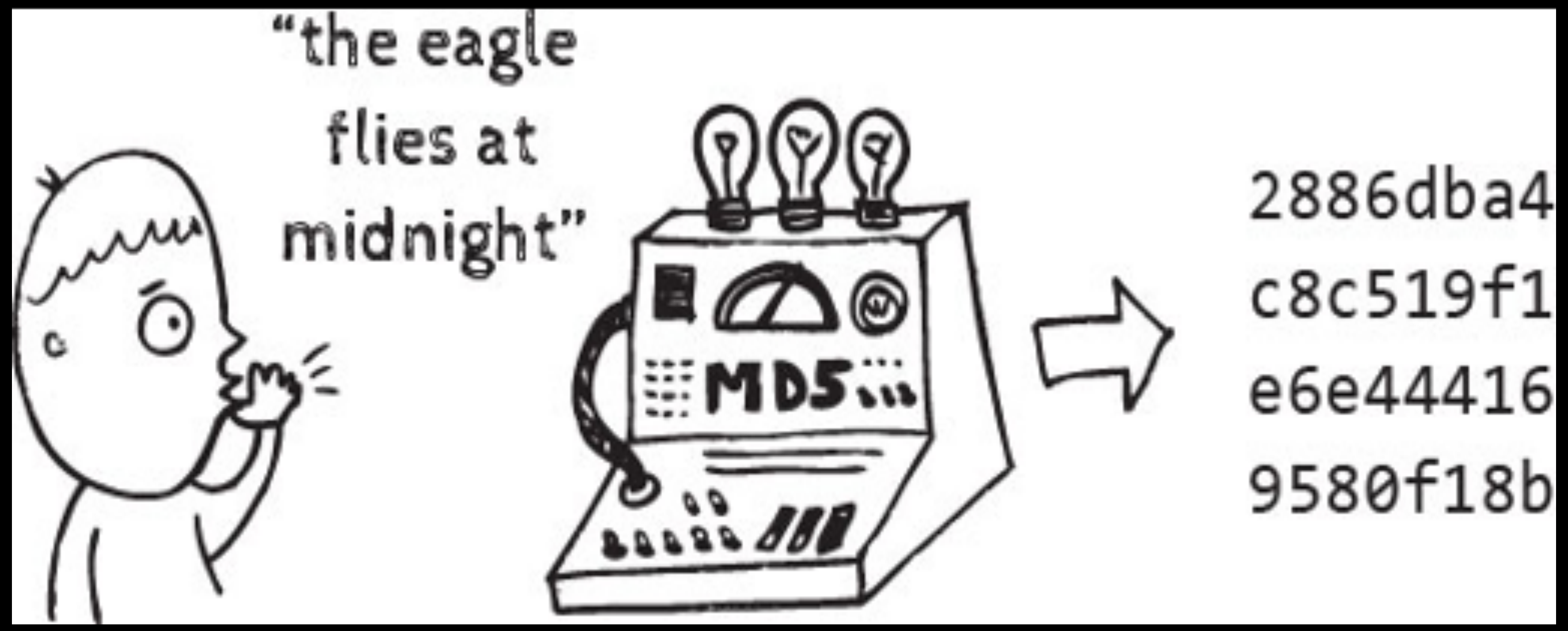


UNIQUE OUTPUT

Two different inputs never map to the same hash value

IMPOSSIBLE TO INVERT

Given the hash, it is impossible to reconstruct the input



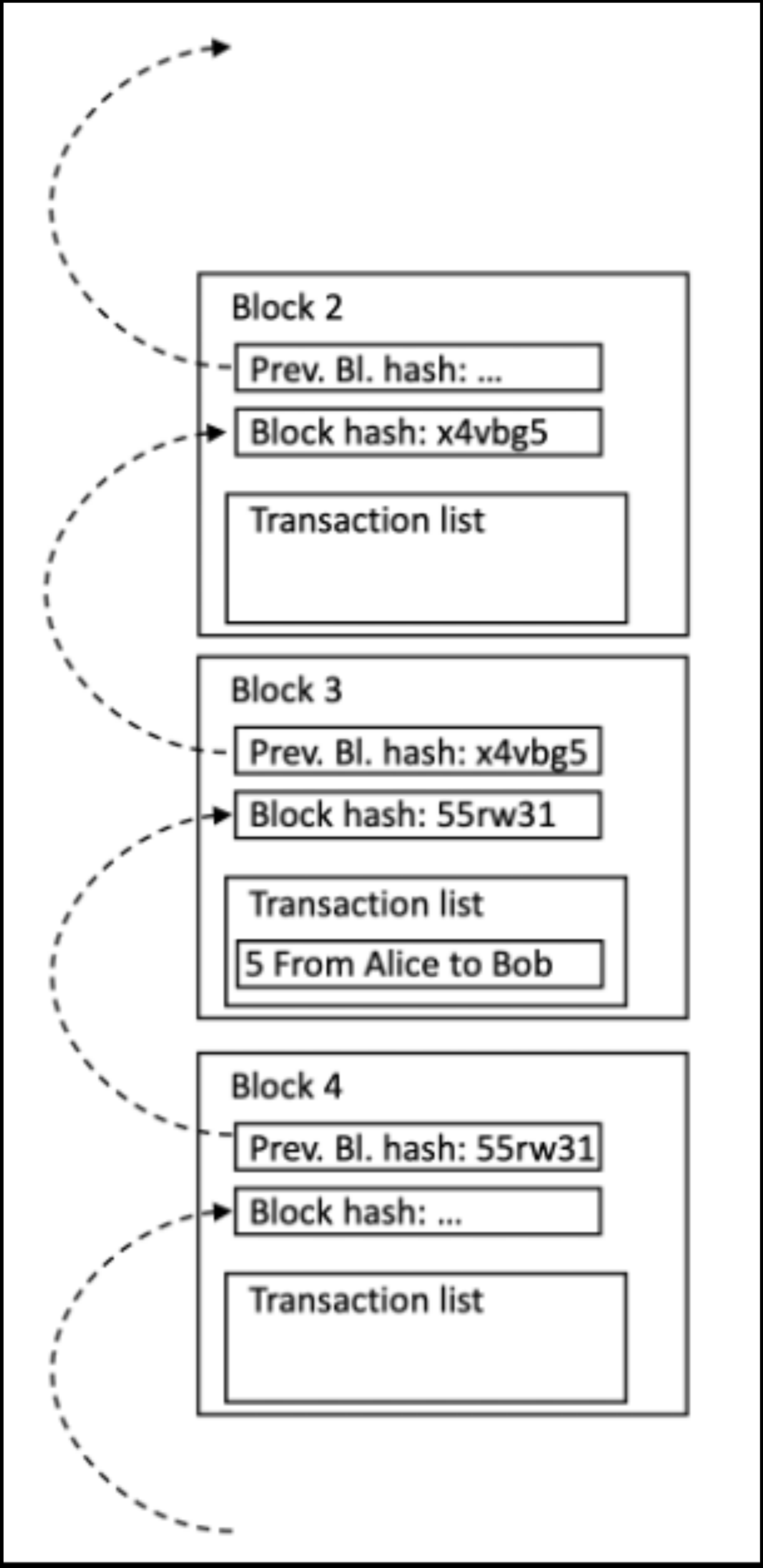
Example: MD5, SHA256

A chain of blocks

The distributed ledger grows with new “blocks”

Blocks contain transactions

Blocks linked through a cryptographic “chain”: each block contains the value of the hash of the previous one



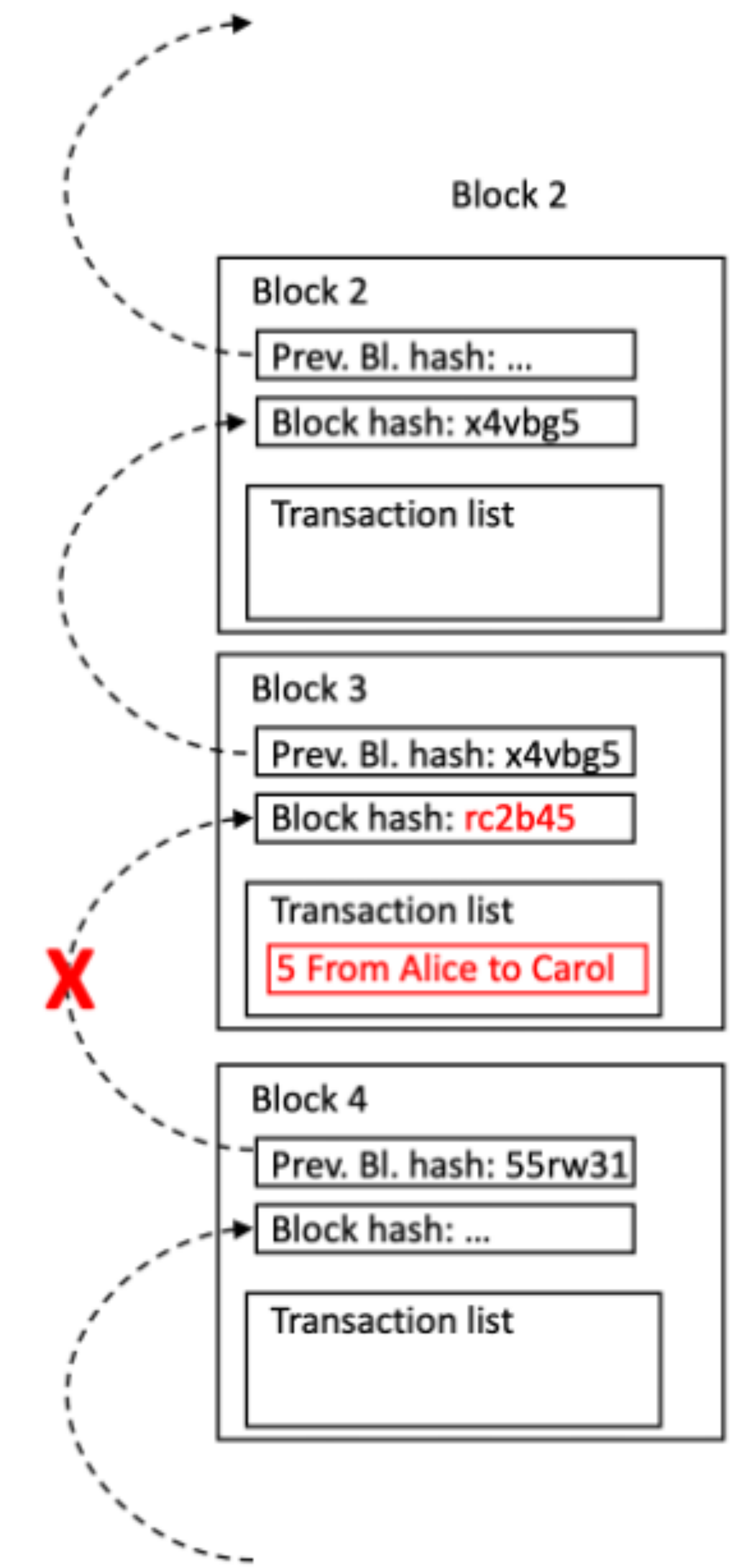
Immutability through the chain of hashes

An attacker wants to modify an existing transaction in block 3

Changing even one bit of Block 3 will change its hash and “break” the chain

The attacker must modify the content of the next blocks to match the new hash of block 3…

…which is computationally impossible because of the properties of cryptographic hashing



Identifying nodes on the blockchain: the principle

Identification on the Web usually done by central entities: service providers, identity providers

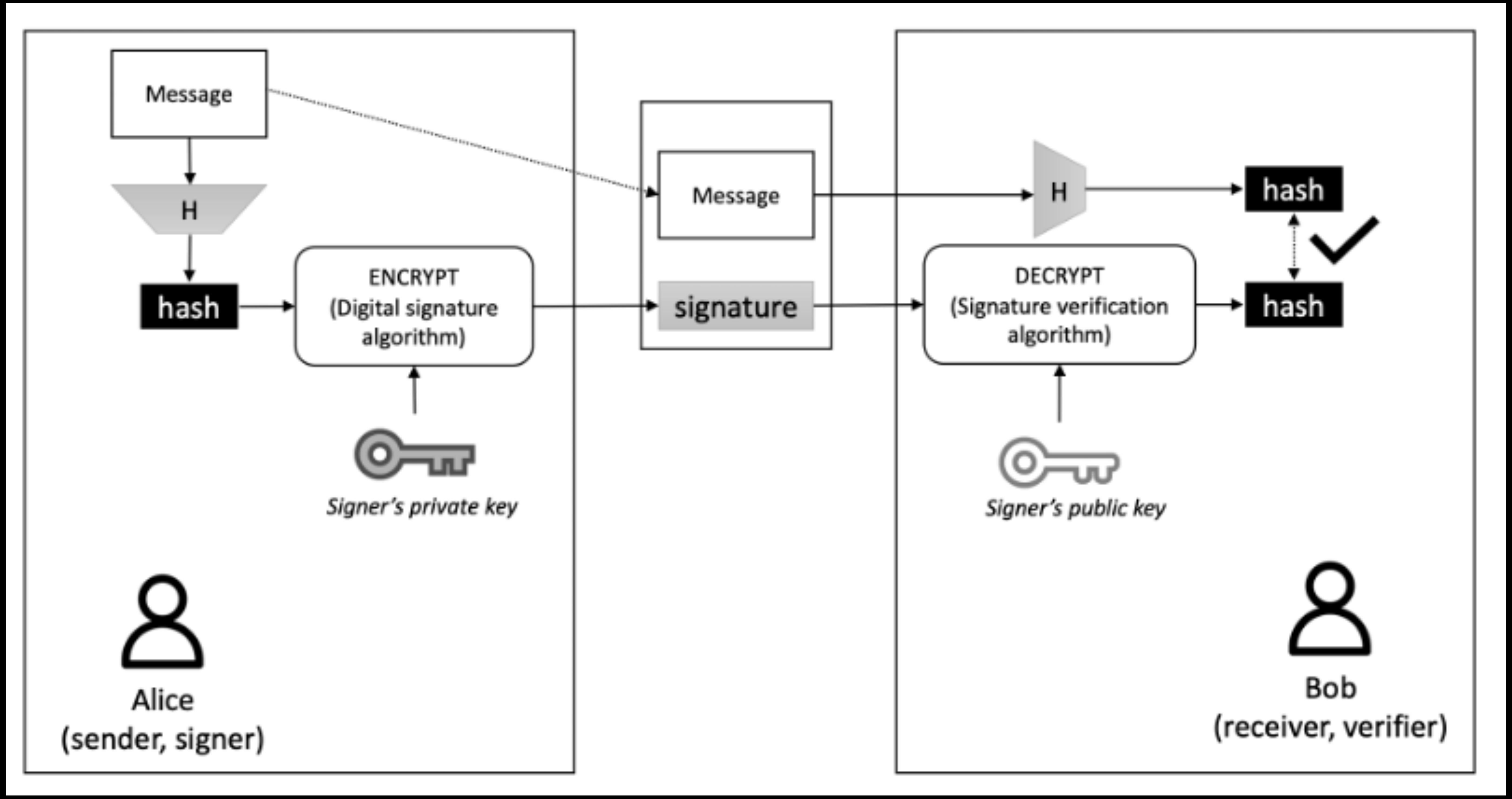
Central entities do not exist on the blockchain!

Nodes of a blockchain digitally sign every transaction that they submit to it

Guarantees non-repudiation (and message integrity)

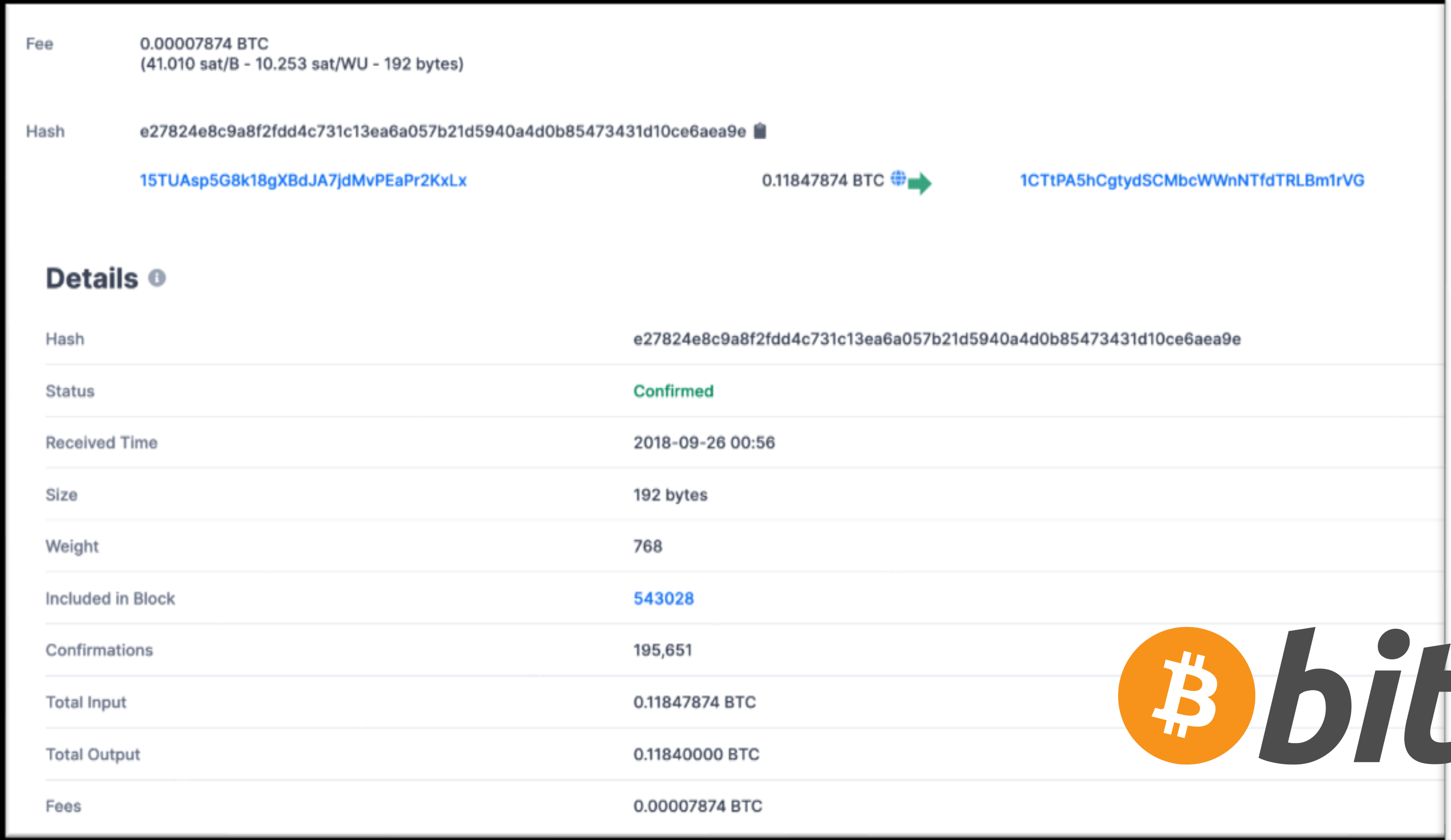
Digital signatures

Combine cryptographic hashing with Asymmetric Encryption: one (private) key to encrypt, one (public) key to decrypt messages



Example: RSA, ECDSA

Alice (signer) needs to send a message to Bob (verifier)
Bob wants to be sure that the message M he receives was sent by Alice
Alice wants to be sure that Bob cannot repudiate her as the source of M



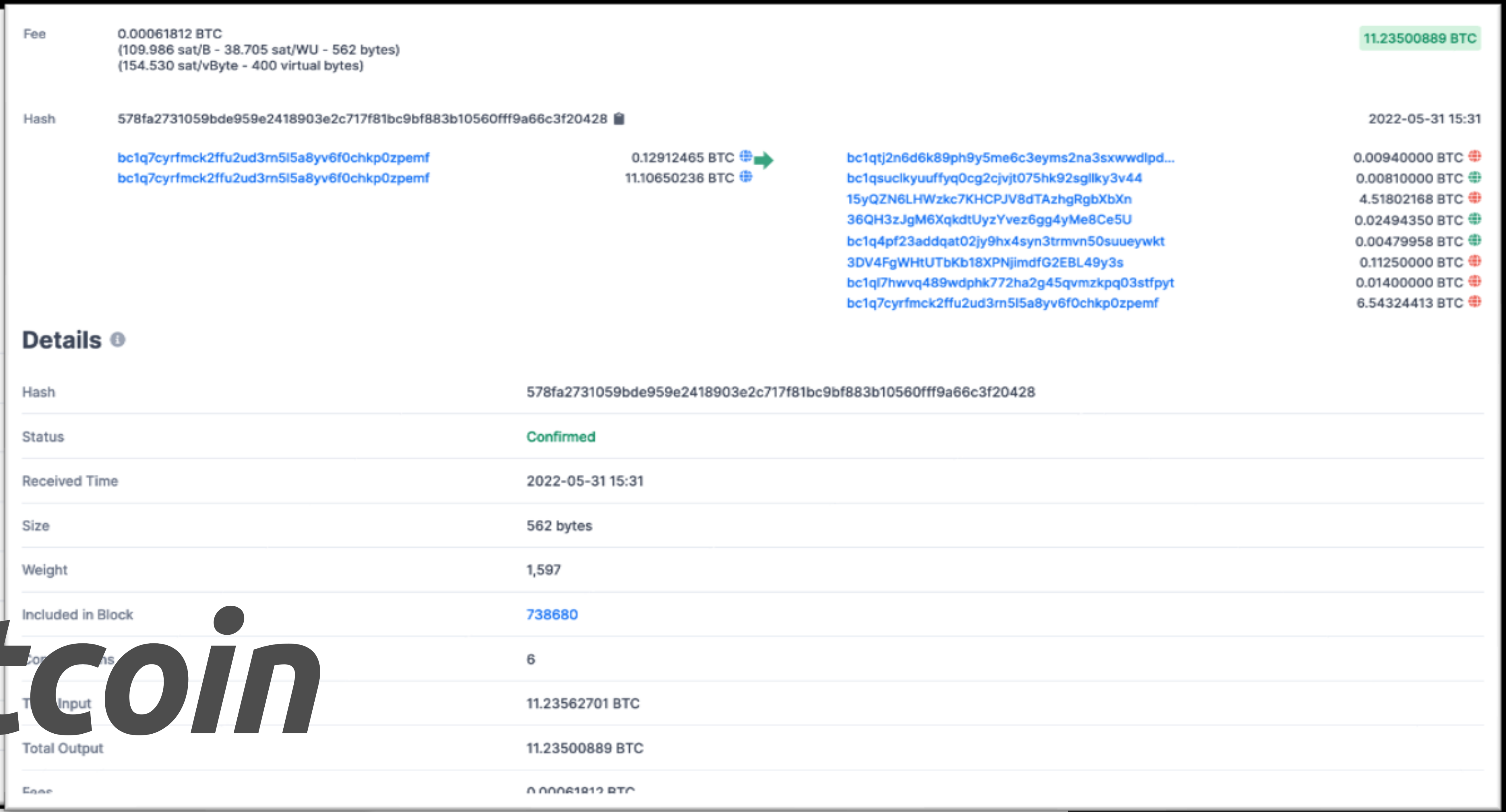
Fee 0.00007874 BTC
(41.010 sat/B - 10.253 sat/WU - 192 bytes)

Hash e27824e8c9a8f2fdd4c731c13ea6a057b21d5940a4d0b85473431d10ce6aea9e

15TUAsp5G8k18gXBdJA7jdMvPEaPr2KxLx 0.11847874 BTC → 1CTtPA5hCgtydSCMbcWWnNTfdTRLBm1rVG

Details

Hash	e27824e8c9a8f2fdd4c731c13ea6a057b21d5940a4d0b85473431d10ce6aea9e
Status	Confirmed
Received Time	2018-09-26 00:56
Size	192 bytes
Weight	768
Included in Block	543028
Confirmations	195,651
Total Input	0.11847874 BTC
Total Output	0.11840000 BTC
Fees	0.00007874 BTC



Fee 0.00061812 BTC
(109.986 sat/B - 38.705 sat/WU - 562 bytes)
(154.530 sat/vByte - 400 virtual bytes)

11.23500889 BTC

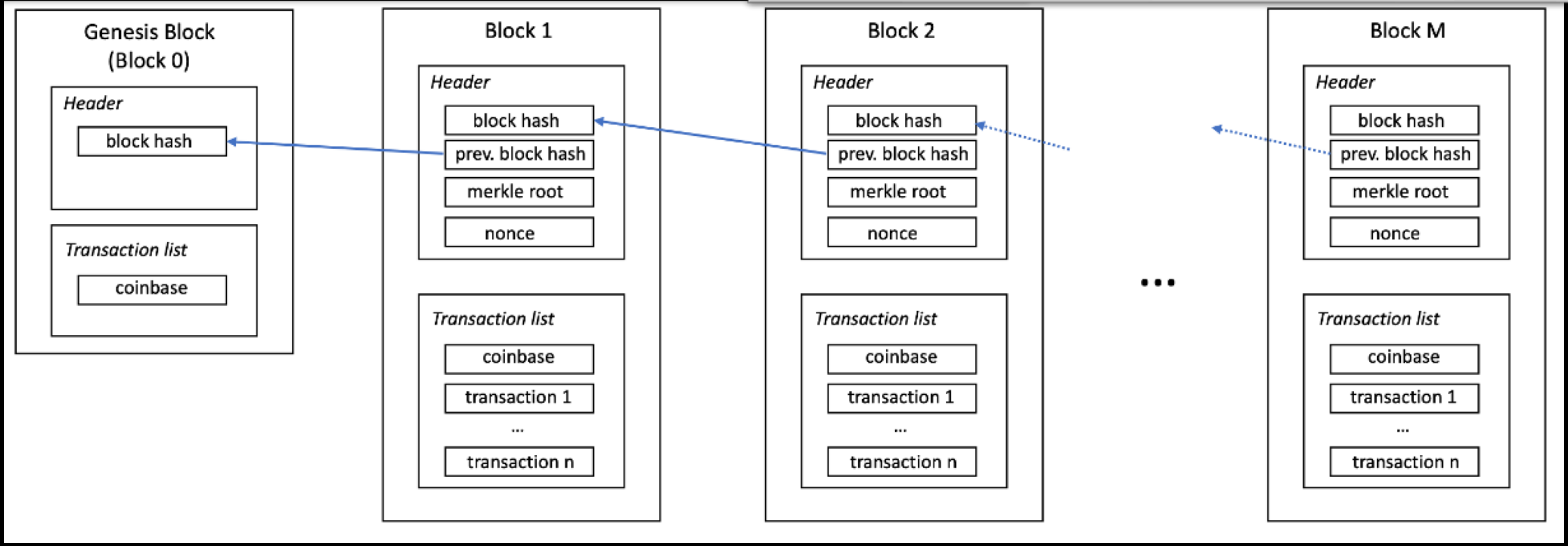
Hash 578fa2731059bde959e2418903e2c717f81bc9bf883b10560fff9a66c3f20428

2022-05-31 15:31

bc1q7cyrfmck2ffu2ud3rn5l5a8yv6f0chkp0zpemf	0.12912465 BTC →	bc1qtj2n6d6k89ph9y5me6c3eyms2na3sxwwdlpd...	0.00940000 BTC
bc1q7cyrfmck2ffu2ud3rn5l5a8yv6f0chkp0zpemf	11.10650236 BTC	bc1qsuclkyuuffyq0cg2cjvjt075hk92sgllky3v44	0.00810000 BTC
		15yQZN6LHWzkc7KHCPJV8dTAzhgRgbXbXn	4.51802168 BTC
		36QH3zJgM6XqkdtUyzYvez6gg4yMe8Ce5U	0.02494350 BTC
		bc1q4pf23addqat02jy9hx4syn3trmvn50suueywkt	0.00479958 BTC
		3DV4FgWHtUTbKb18XPNjimdfG2EBL49y3s	0.11250000 BTC
		bc1ql7hwvq489wdphk772ha2g45qvmzkpq03stfpyt	0.01400000 BTC
		bc1q7cyrfmck2ffu2ud3rn5l5a8yv6f0chkp0zpemf	6.54324413 BTC

Details

Hash	578fa2731059bde959e2418903e2c717f81bc9bf883b10560fff9a66c3f20428
Status	Confirmed
Received Time	2022-05-31 15:31
Size	562 bytes
Weight	1,597
Included in Block	738680
Co ns	6
T Input	11.23562701 BTC
Total Output	11.23500889 BTC
Fees	0.00061812 BTC



Smart Contracts

Blockchain Recipe

P2P NETWORK

DISTRIBUTED LEDGER

CONSENSUS MECHANISM

CRYPTO HASHING

DIGITAL SIGNATURES

What is blockchain? (Encore…)

A “distributed state machine”

A system allowing the nodes of a network to agree on the value of a set of variables describing the “state” of a system, without the need for a centralized intermediary

State in BC4C: active games and their moves

State in ETK: balance of all users and/or list of transactions from beginning of system history

Is that all?

The state is simple: transfers of currency, player moves in a game

State updates are static: they only happen when transactions are submitted by nodes

No business logic associated with transaction execution

Example
BC4C (+ ETK)

(Players have an ETK balance and bet on themselves to win a game)

When playing a game, players pay a deposit*

The winner of a game takes all the deposits paid for it

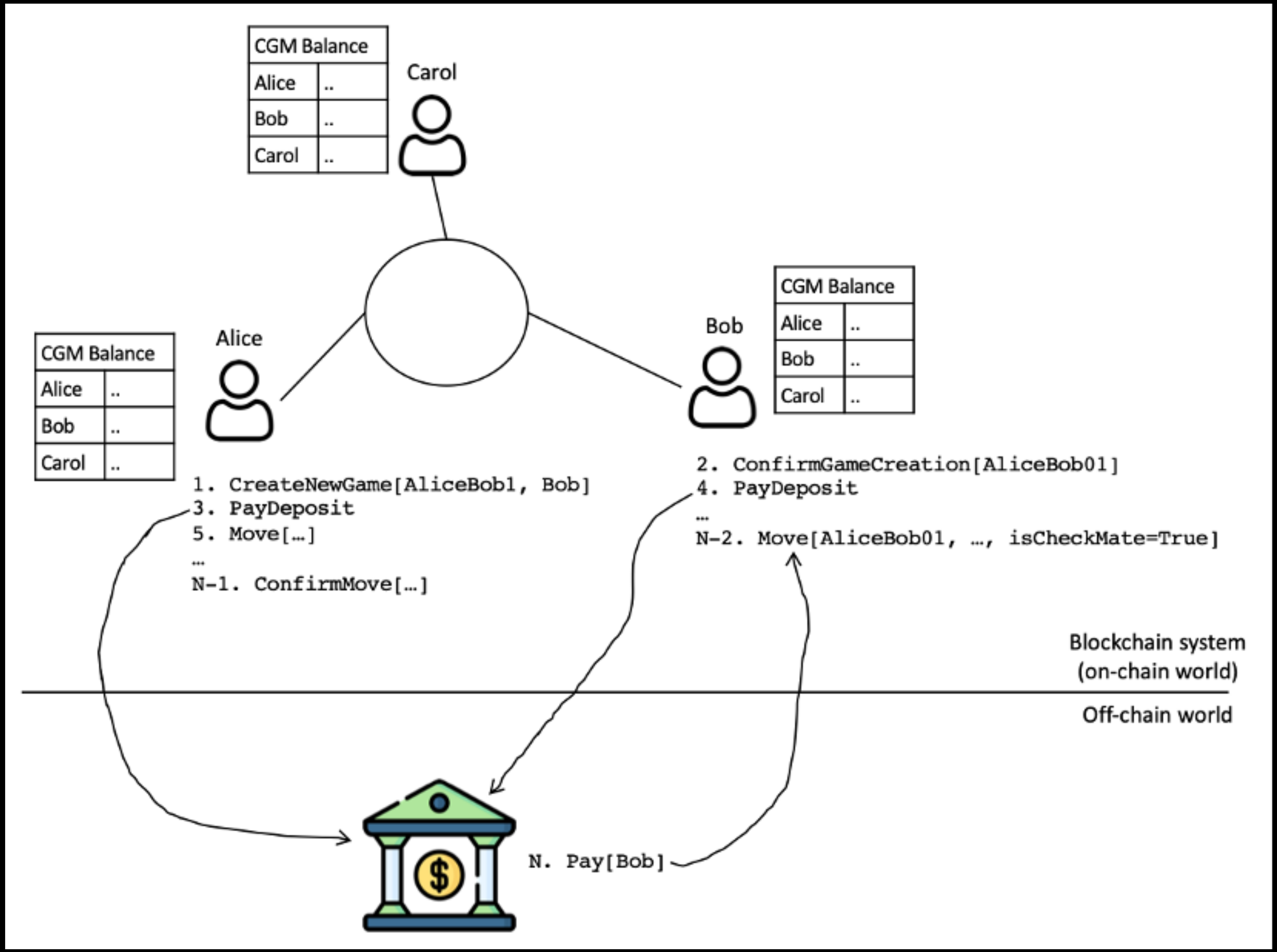
* pay a deposit = modify the value of blockchain state variable (balance of the player)

Everything works out with an intermediary…

Force players to pay deposits to a bank

Bank monitors the games and pays out the deposits to the winner

It works …
but it's not blockchain :(



CGM Balance	
Alice	..
Bob	..
Carol	..

CGM Balance	
Alice	..
Bob	..
Carol	..

CGM Balance	
Alice	..
Bob	..
Carol	..

```
1. CreateNewGame[AliceBob1, Bob]
3. PayDeposit
5. Move[…]
…
N-1. ConfirmMove[…]
```

```
2. ConfirmGameCreation[AliceBob01]
4. PayDeposit
…
N-2. Move[AliceBob01, …, isCheckMate=True]
```

```
N. Pay[Bob]
```

What if…

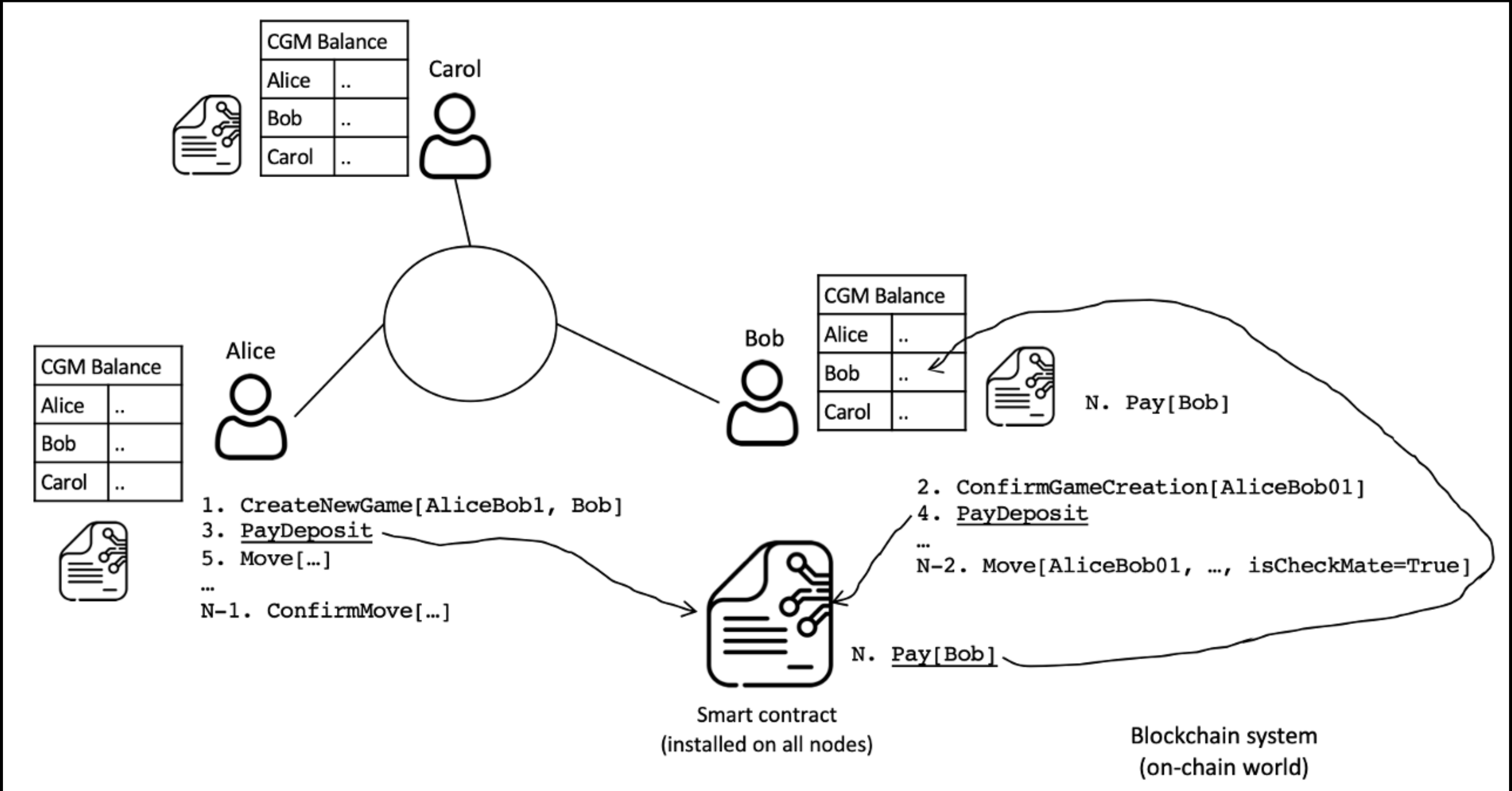
… we wrote a simple computer program that players must call only once before a game starts to pay the deposits

A game cannot start if both players have not paid their deposit

Every node installs this program locally, transactions can trigger the execution of this program

The program monitors the state of the blockchain and, when a game ends, pays the deposits back to the winner

(Let’s call this program a “smart contract”)



CGM Balance	
Alice	..
Bob	..
Carol	..

CGM Balance	
Alice	..
Bob	..
Carol	..

CGM Balance	
Alice	..
Bob	..
Carol	..

N. Pay[Bob]

1. CreateNewGame[AliceBob1, Bob]
3. PayDeposit
5. Move[…]
…
N-1. ConfirmMove[…]

2. ConfirmGameCreation[AliceBob01]
4. PayDeposit
…
N-2. Move[AliceBob01, …, isCheckMate=True]

N. Pay[Bob]

Smart contract
(installed on all nodes)

Blockchain system
(on-chain world)

Welcome Smart Contracts

Simply stated, a smart contract is a computer program that can manipulate the state of a blockchain

Smart contract code is deployed at every node and it is also immutable by design (how?)

Smart contracts allow consistent state updates among peers, controlled by (complex) business logic

Smart contracts are not very “smart”

Smart contracts do not have to be legally binding “contracts”

Smart contracts are not very “smart” (actually, they have a lot of limitations!)

Random variables?

Off-chain data?

Ethereum Tokens

- Business
  - Tokens are native on-chain assets
  - Private cryptocurrencies, asset/resource identifiers
- IT:
  - Tokens are smart contracts implementing a standard interface
  - Their usage and behaviour can be programmed
  - They are Ethereum nodes, that transactions can address

Exercise
(Part B)