A close-up photograph of a white boat's hull and a metal chain anchor in the water. The chain is rusted and attached to a metal plate on the hull. The water is dark blue.

# Blockchain Course

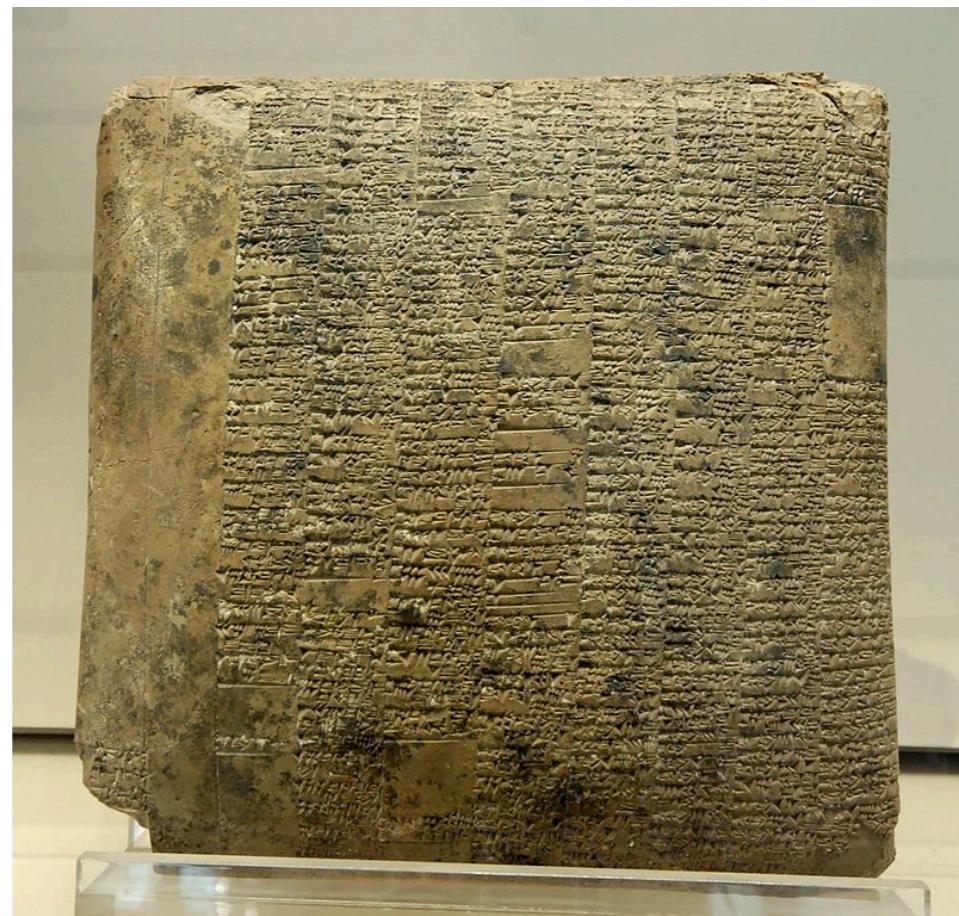
## Arthur Gervais

A close-up photograph of a white boat's bow and anchor chain against a dark blue sea. The boat's white hull is visible on the left, showing some wear and a metal plate with a chain attached. A thick, rusty anchor chain runs across the frame. The water is a deep, dark blue.

# Introduction

**2040 BC**

# 2040 BC



Balance sheet - Mesopotamia

Balance sheet



Mesopotamia

**2040 BC**

Centralized  
Banking

Balance sheet



Mesopotamia

**2040 BC**

Centralized  
Banking

Balance sheet



Mesopotamia

**2040 BC**

Centralized  
Banking

## Balance sheet



Mesopotamia

HSBC 

 BNP PARIBAS

2040 BC

1848



 Bank of America

Centralized  
Banking

## Balance sheet



Centralized  
Banking

## Balance sheet



Mesopotamia



HSBC



Turing

VISA

2040 BC

1848

1936

1958



Bank of America



Centralized  
Banking

Balance sheet



Mesopotamia



HSBC



Turing

VISA

ECash



Chaum

2040 BC

1848

1936

1958

1983



Bank of America



Centralized  
Banking

Privacy-Preserving  
Banking

Balance sheet



Mesopotamia



Turing



ECash



Chaum

Ethereum



CASH

2040 BC

1848

1936

1958

1983

2008

2015

2016

Centralized  
Banking

Privacy-Preserving  
Banking

Decentralized  
Banking



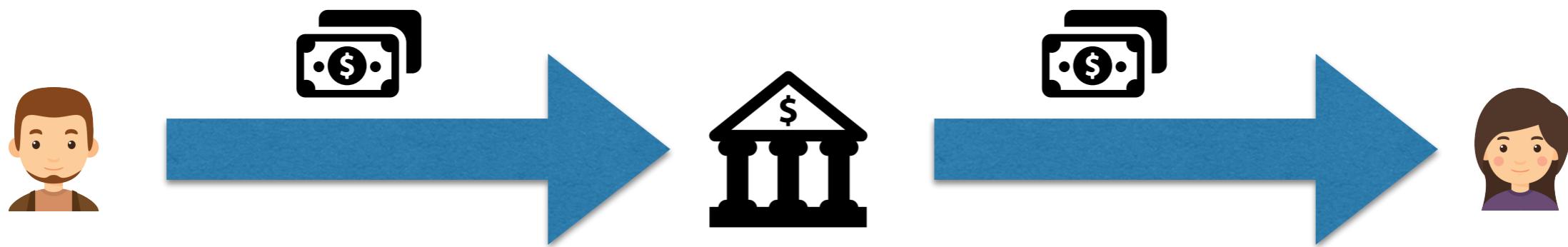
# From Centralized to Decentralized Payment Systems



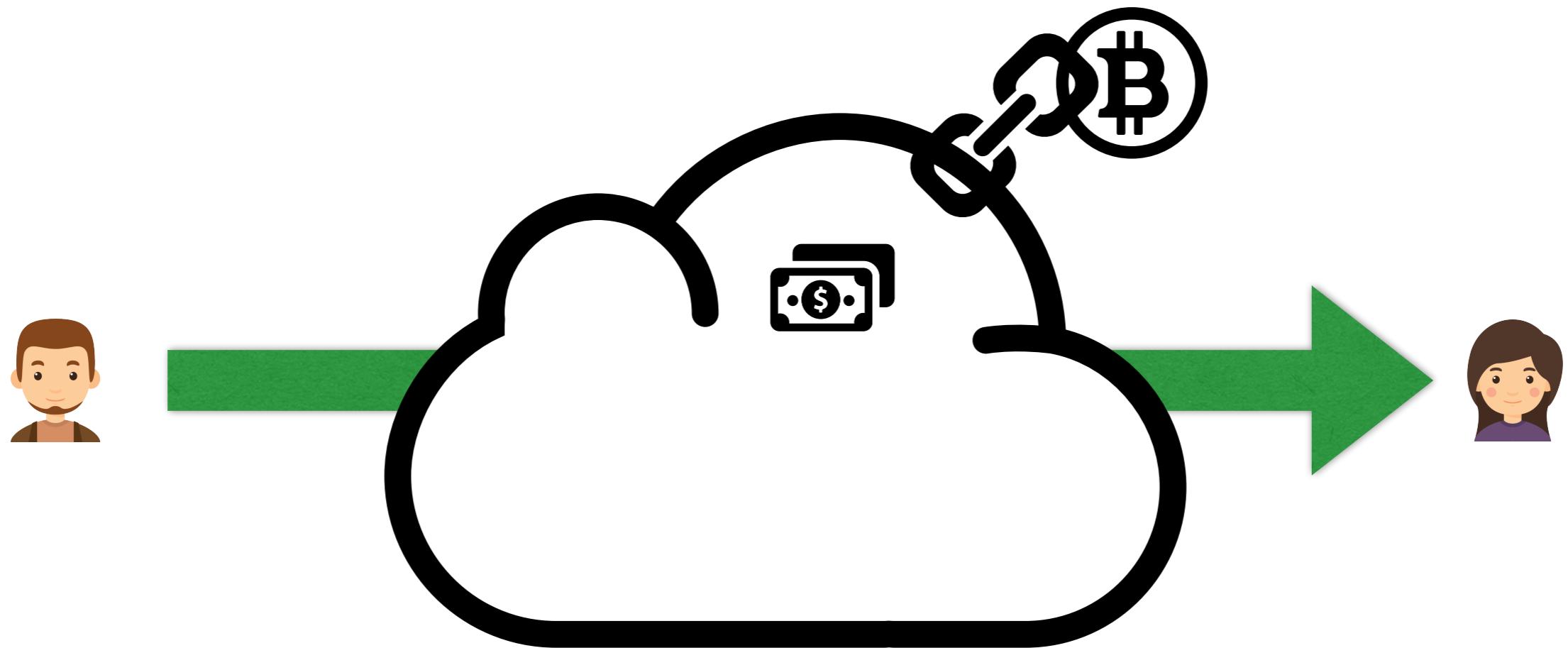
# From Centralized to Decentralized Payment Systems



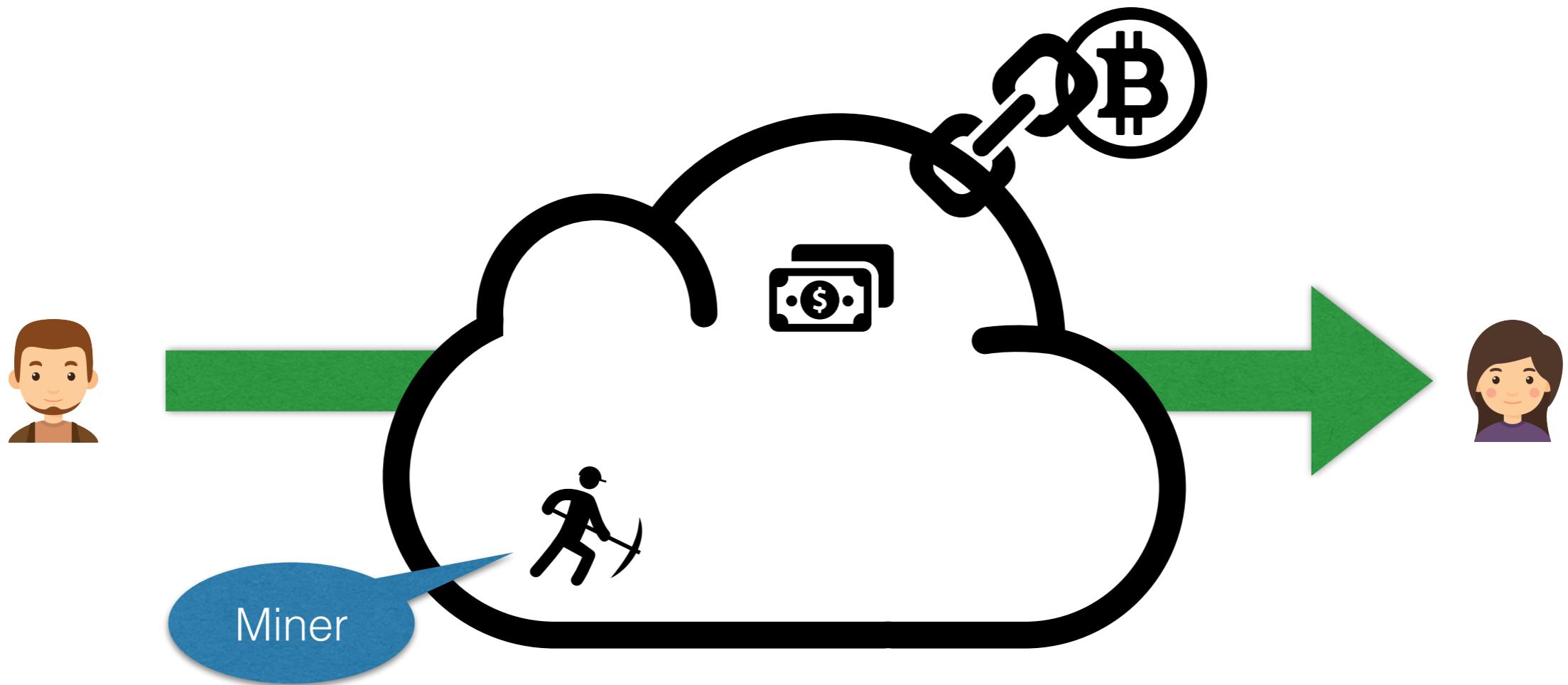
# From Centralized to Decentralized Payment Systems



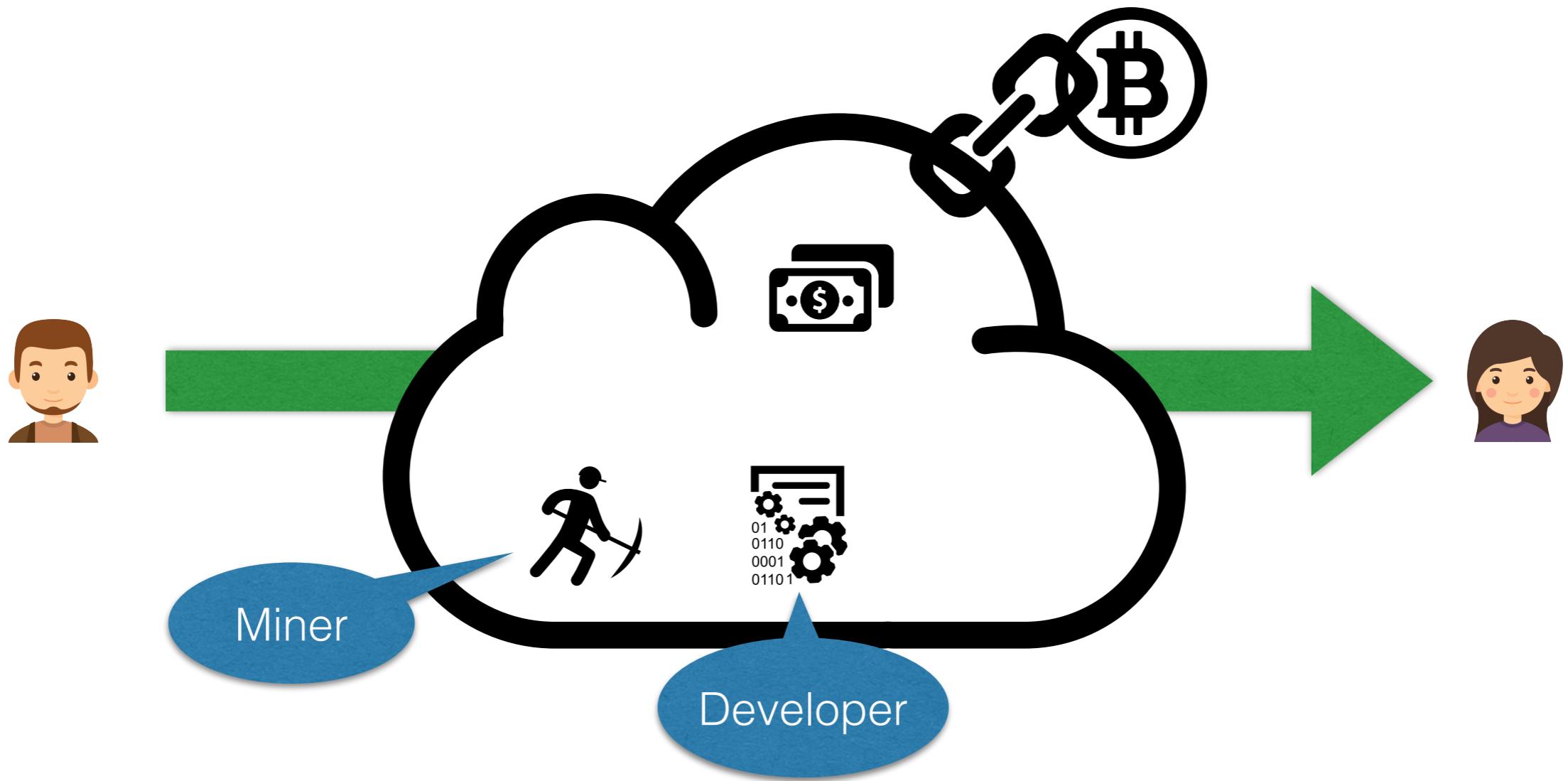
# From Centralized to Decentralized Payment Systems



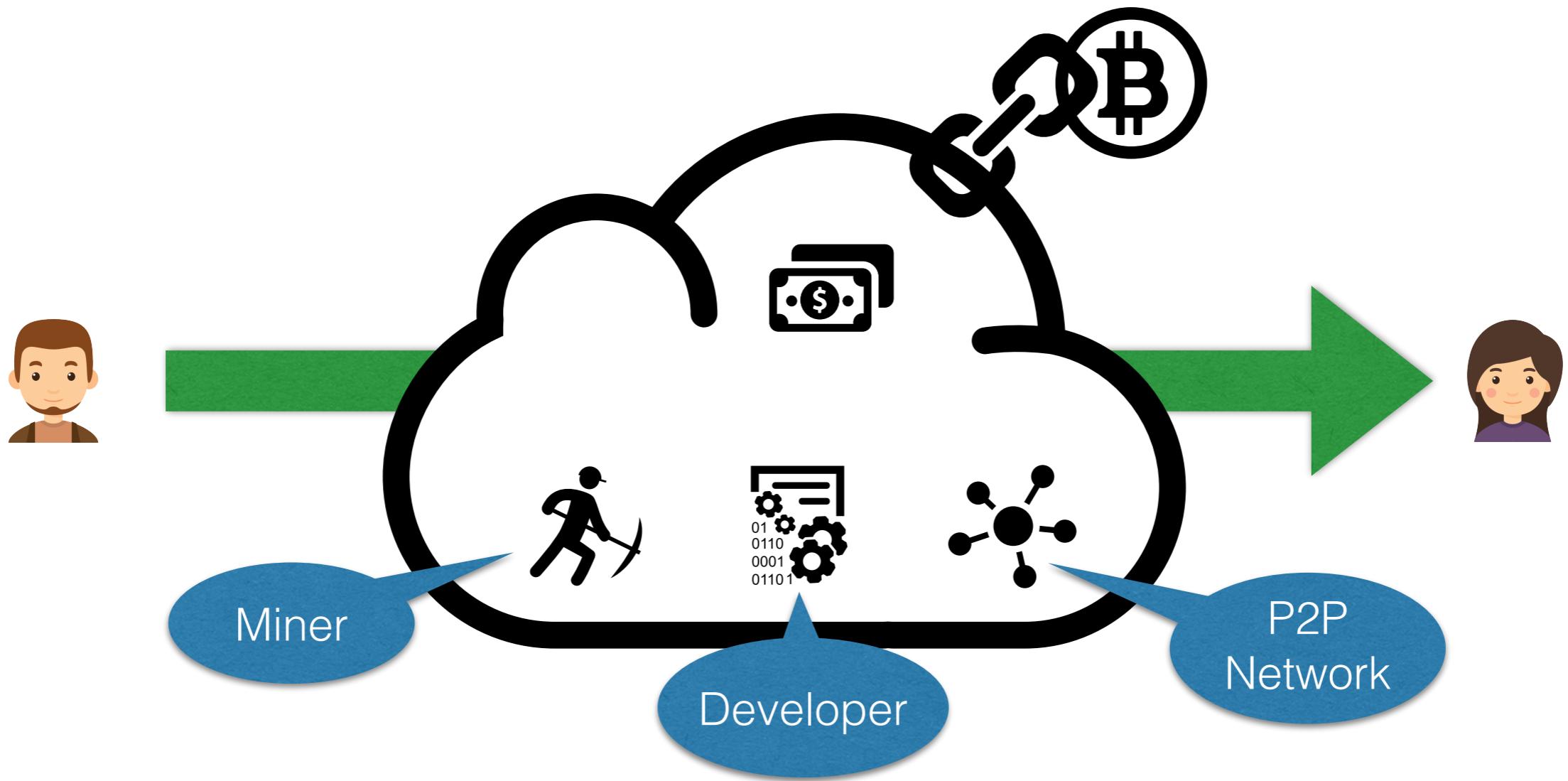
# From Centralized to Decentralized Payment Systems



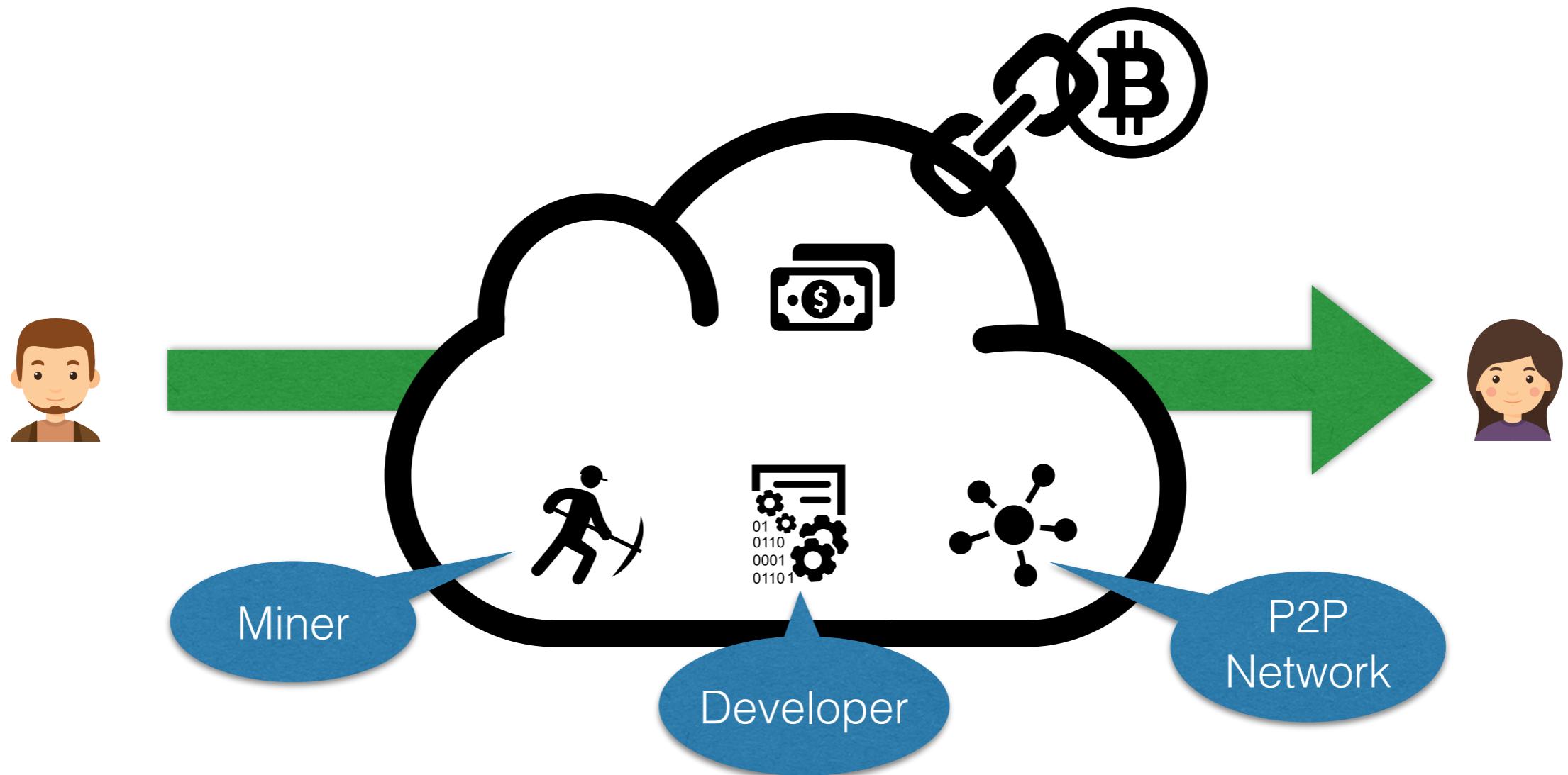
# From Centralized to Decentralized Payment Systems



# From Centralized to Decentralized Payment Systems



# From Centralized to Decentralized Payment Systems



How to perform secure **decentralised** payments?



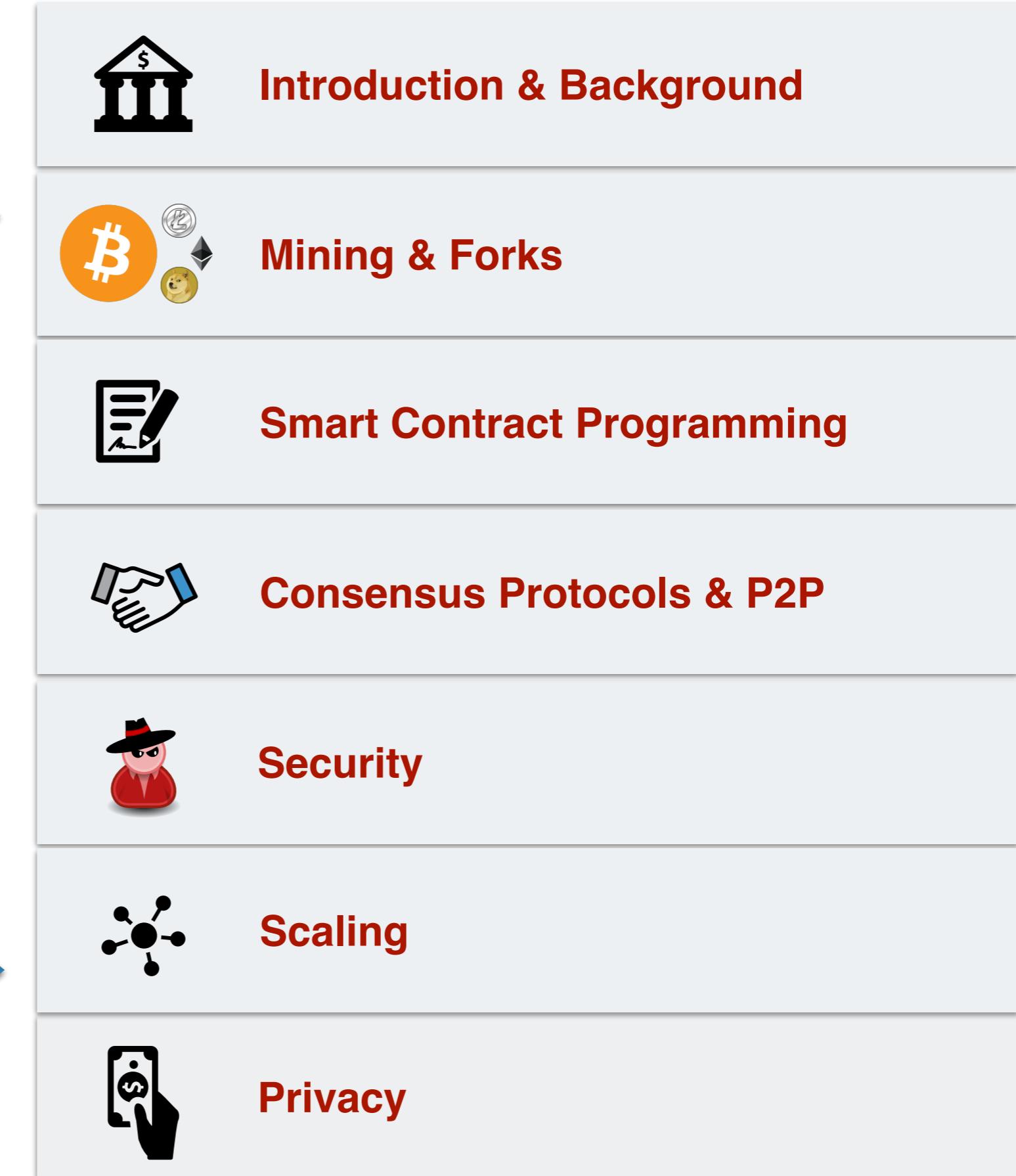
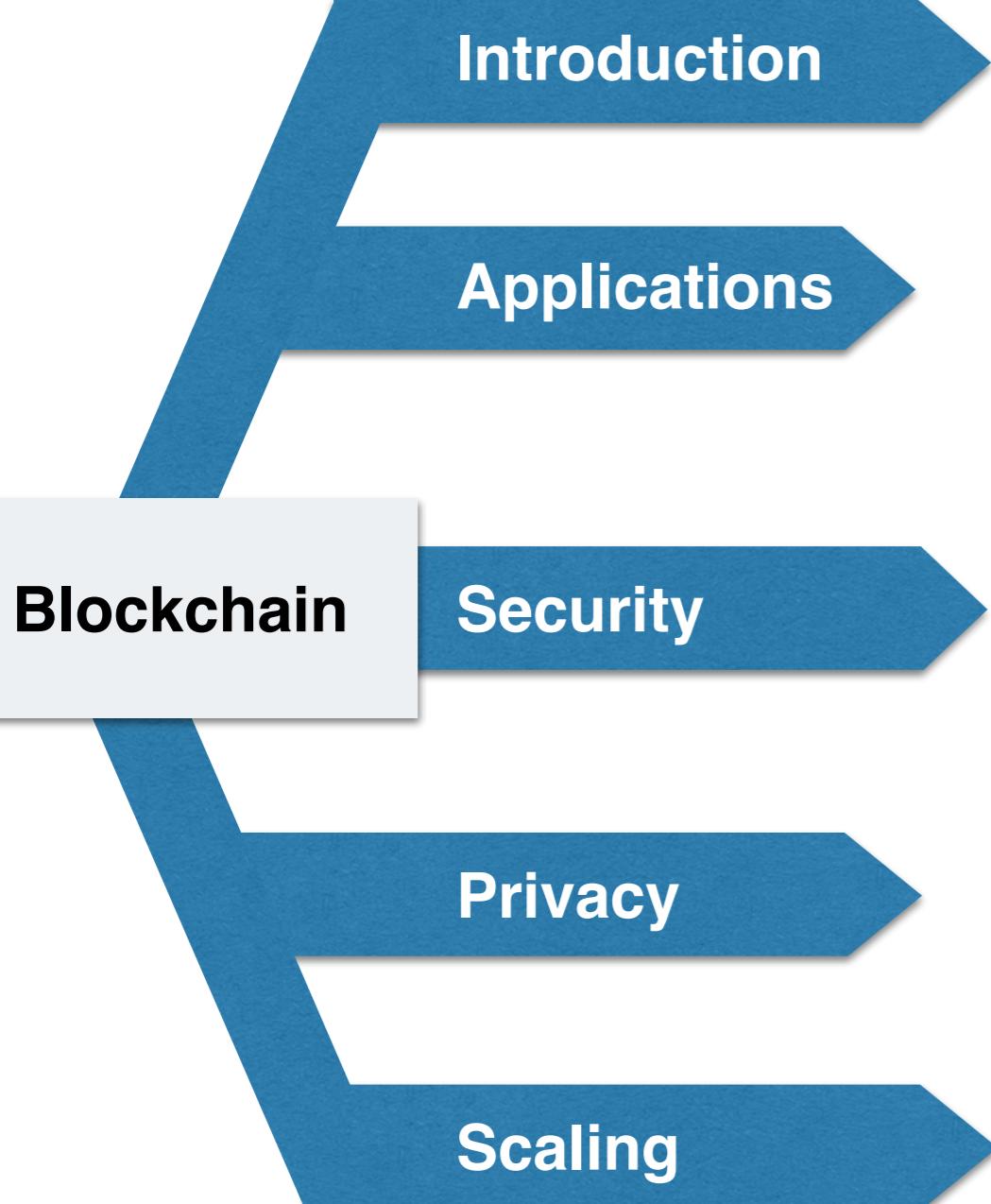
How to exchange **privacy-preserving** payments?



How to make decentralised systems **efficient**?



# Course Outline





# Blockchain Background

## Byzantine Generals' Problem

# Byzantine Generals' Problem



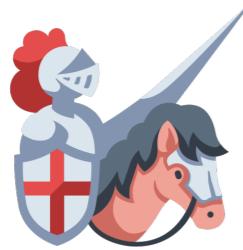
# Byzantine Generals' Problem



Attack?



# Byzantine Generals' Problem



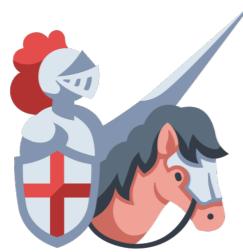
Attack?



Retreat?



# Byzantine Generals' Problem



Attack?



Retreat?



Attack?



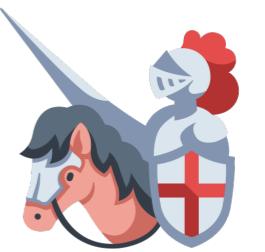
# Byzantine Generals' Problem



Attack?



Retreat?



Attack?



Surrender?

# Byzantine Generals' Problem

- Everyone has to know X
- Everyone knows that everyone knows X
- Everyone knows that everyone knows that everyone knows X



Attack?



Retreat?



Attack?



Surrender?

# Byzantine Generals' Problem



Who does this  
coin belong to?



Who does this  
coin belong to?

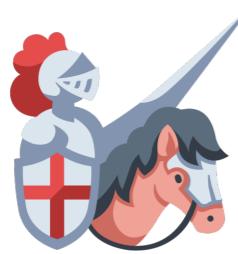


Who does this  
coin belong to?



Who does this  
coin belong to?

# Byzantine Generals' Problem



Who does this  
coin belong to?



Who does this  
coin belong to?



Who does this  
coin belong to?



Who does this  
coin belong to?

# Byzantine Generals' Problem



Who does this  
coin belong to?

Far away



Who does this  
coin belong to?



Who does this  
coin belong to?

How to agree, before  
taking an action?



Who does this  
coin belong to?

# (Centralised) Digital Payment Systems



Alice



The coin belongs to Alice.



Bob



- Examples
  - Pepper Micropayments [Rivest]
  - ECash [David Chaum] (privacy preserving)

# (Decentralised) Digital Payment Systems



The coin belongs  
to Alice.



The coin belongs  
to Alice.

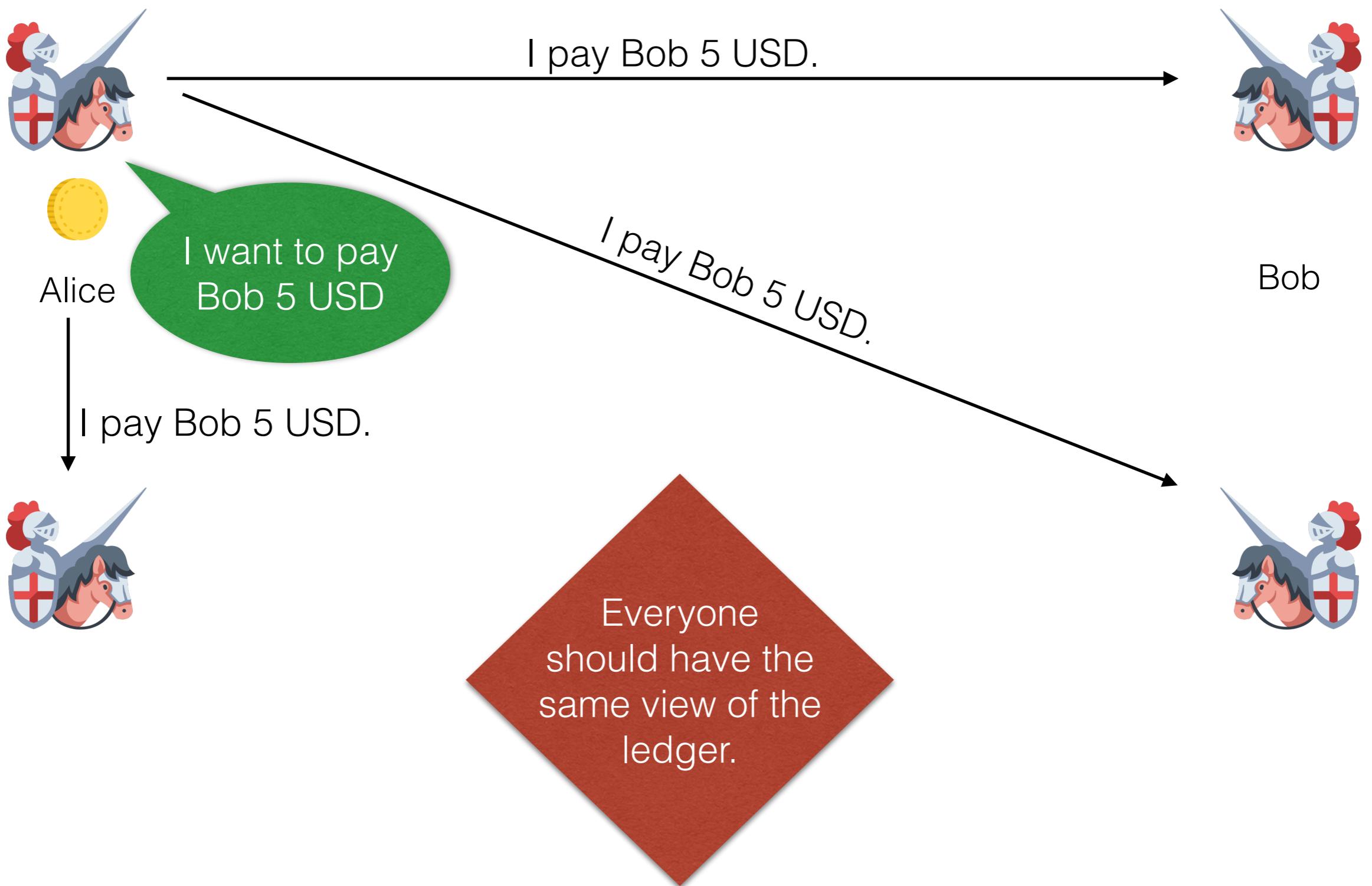


The coin belongs  
to Alice.



The coin belongs  
to Bob.

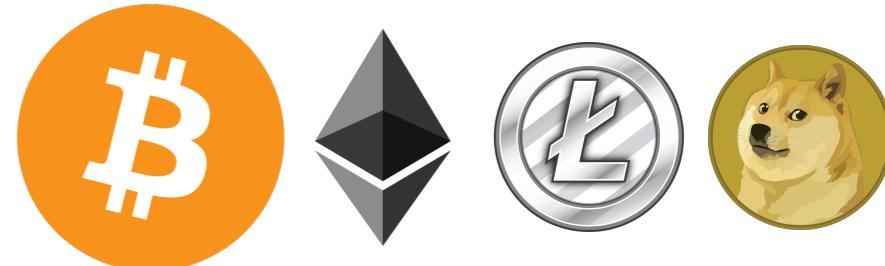
# (Decentralised) Digital Payment Systems



# Blockchain != Blockchain

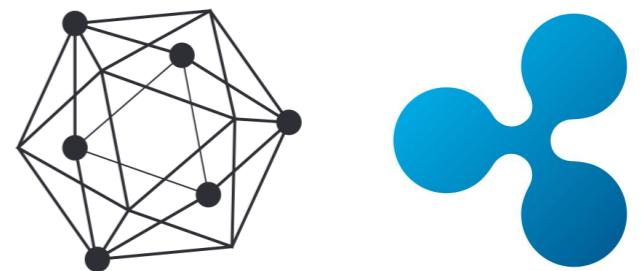
## Open and Decentralised Blockchains

- Ethereum
- Bitcoin



## Permission-based Blockchains

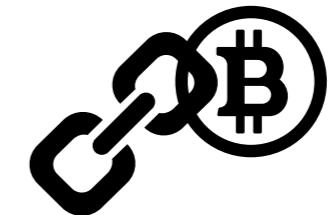
- Hyperledger
- Ripple
- Stellar



# Bitcoin



- First introduced in 2009
  - Pseudonym Satoshi Nakamoto
- Peer-to-peer decentralised currency
- No trusted third parties (???)
- Blockchain: distributed DB
  - ▶ Transactions
  - ▶ Blocks



# Bitcoin

- First introduced in 2009
- Pseudonymous
- Peer-to-peer
- No trusted third party
- Blockchain
- ▶ Transactions
- ▶ Blocks

The screenshot shows a PDF document titled "Is Bitcoin a Decentralized Currency?" by Arthur Gervais, Ghassan O. Karamé, Srdjan Capkun, and Vedran Capkun. The document is from an IACR eprint, specifically https://eprint.iacr.org/2013/829.pdf. The abstract discusses the decentralized nature of Bitcoin, noting that while it promises decentralization, recent incidents and observations reveal significant centralization. It highlights that a limited set of entities control vital operations like mining and incident resolution. The keywords listed are Bitcoin and Decentralized decision process.

Is Bitcoin a Decentralized Currency?

Arthur Gervais\*      Ghassan O. Karamé\*\*      Srdjan Capkun\*  
Vedran Capkun\*\*\*  
\*ETH Zurich, 8092 Zuerich, Switzerland.  
\*\*NEC Laboratories Europe, 69115 Heidelberg, Germany.  
\*\*\*HEC Paris, France.

**Abstract**

Bitcoin has achieved large-scale acceptance and popularity by promising its users a fully decentralized and low-cost virtual currency system. However, recent incidents and observations are revealing the true limits of decentralization in the Bitcoin system. In this article, we show that the vital operations and decisions that Bitcoin is currently undertaking are not decentralized. More specifically, we show that a limited set of entities currently control the services, decision making, mining, and the incident resolution processes in Bitcoin. We also show that third-party entities can unilaterally decide to “devalue” any specific set of Bitcoin addresses pertaining to any entity participating in the system. Finally, we explore possible avenues to enhance the decentralization in the Bitcoin system.

**Keywords:** Bitcoin, Decentralized decision process

## 1 Introduction

Bitcoin has witnessed a wider adoption and attention than any other digital currency proposed to date. One reason for such a broad adoption of Bitcoin has been a promise of a low-cost and decentralized currency that is inherently independent of governments and of any centralized authority [1]. In this work, we analyze the (de-)centralized nature of Bitcoin and show that—contrary to widespread belief—Bitcoin is not a truly decentralized system as it is deployed and implemented today.

Namely, in Bitcoin, the users “vote” with their computing power to prevent double-spending (i.e., by *power-voting*) which effectively limits the power of individual users and makes Sybil attacks difficult. Given the huge computing power harnessed in the Bitcoin system (currently around 30,000 Tera hashes per second), users believe that it is unlikely for any entity to acquire such power alone. However, even a quick look at the distribution of computing power in Bitcoin reveals that the power of dedicated “miners” far exceeds the power that individual users dedicate to mining, allowing few parties to effectively control the currency; currently the top-three (centrally managed) mining pools control more than 50% of the computing power in Bitcoin. Indeed, while mining and block generation in Bitcoin was originally designed to be decentralized, these processes are currently largely centralized.

On the other hand, other Bitcoin operations, like protocol updates and incident resolution are not designed to be decentralized, and are controlled by a small number of administrators whose



A close-up photograph of a white boat's hull and a metal chain anchor in the water. The chain is rusted and attached to a metal plate on the hull. The water is dark blue.

**Blockchain**  
**Simple Cryptography**

# Data Types



- **Cryptographic Hash Functions**
- Merkle Trees
- Elliptic Curve Signature Algorithm (ECDSA)

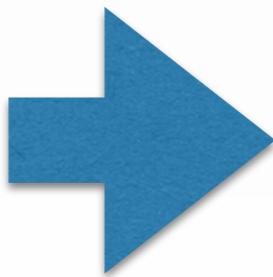
# Data Types



## • Cryptographic Hash Functions

- SHA256
- RIPEMD160

Arbitrarily long  
data



Fixed sized  
hash/digest



# Cryptographic Hash Function



- Cryptographic Hash Functions
  - Takes any byte sequence as input
  - Fixed size output
  - Efficiently computable
- Security Properties:
  - Collision-resistance
  - Second pre-image resistance
  - Pre-image resistance
  - Hiding
  - Puzzle-friendly

Example: <https://www.pelock.com/products/hash-calculator>



## Pre-image Resistance

- For any given  $h$  in the output space of the hash function, it is hard to find  $x$ , s.t.  $H(x)=h$

## Second Pre-image Resistance

- For a given message  $x$ , it is hard to find  $y$  s.t.  $x \neq y$  and  $H(x) = H(y)$

## Collision Resistance

- It is hard to find a pair of values,  $x \neq y$  and  $H(x) = H(y)$



## Hiding

- A hash function  $H$  is hiding when a secret value  $r$  is chosen from a high min-entropy probability distribution, then given  $H(r \parallel x)$ , it is hard to find  $x$ .

## Puzzle-friendly

- A hash function  $H$  is puzzle friendly if for every possible  $n$ -bit output value  $h$ , if  $k$  is chosen from a distribution with high min-entropy, then it is infeasible to find  $x$  such that  $H(k \parallel x) = h$  in time significantly less than  $2^n$ .



## Search puzzle

- A hash function  $H$
- A value,  $id$ , chosen from a high min-entropy distribution
- A target set  $Y$

A solution to the puzzle is a value  $x$ , s.t.

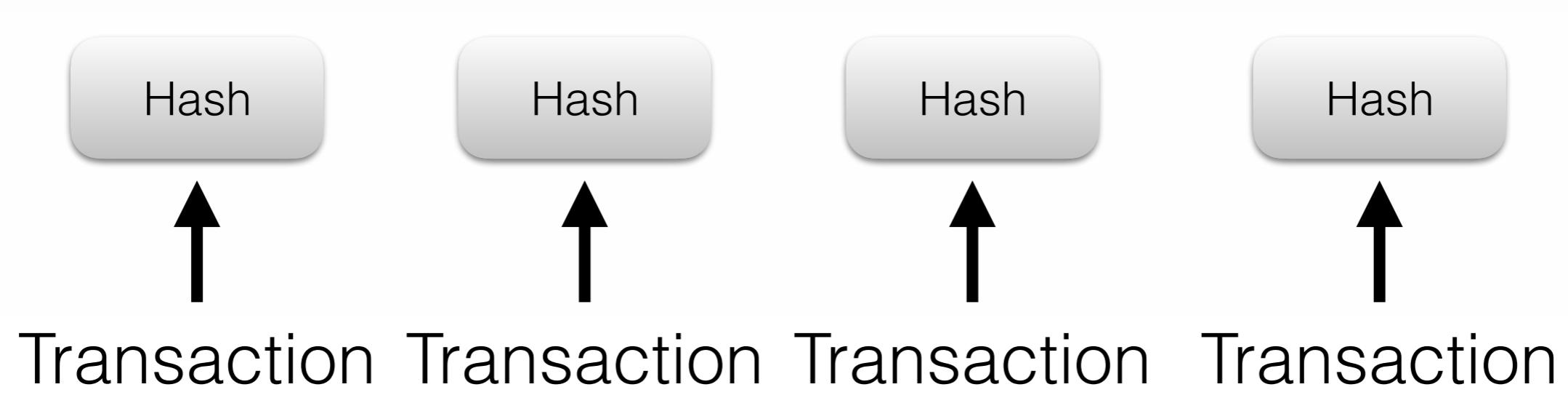
$$H(id||x) \in Y$$

# Data Types

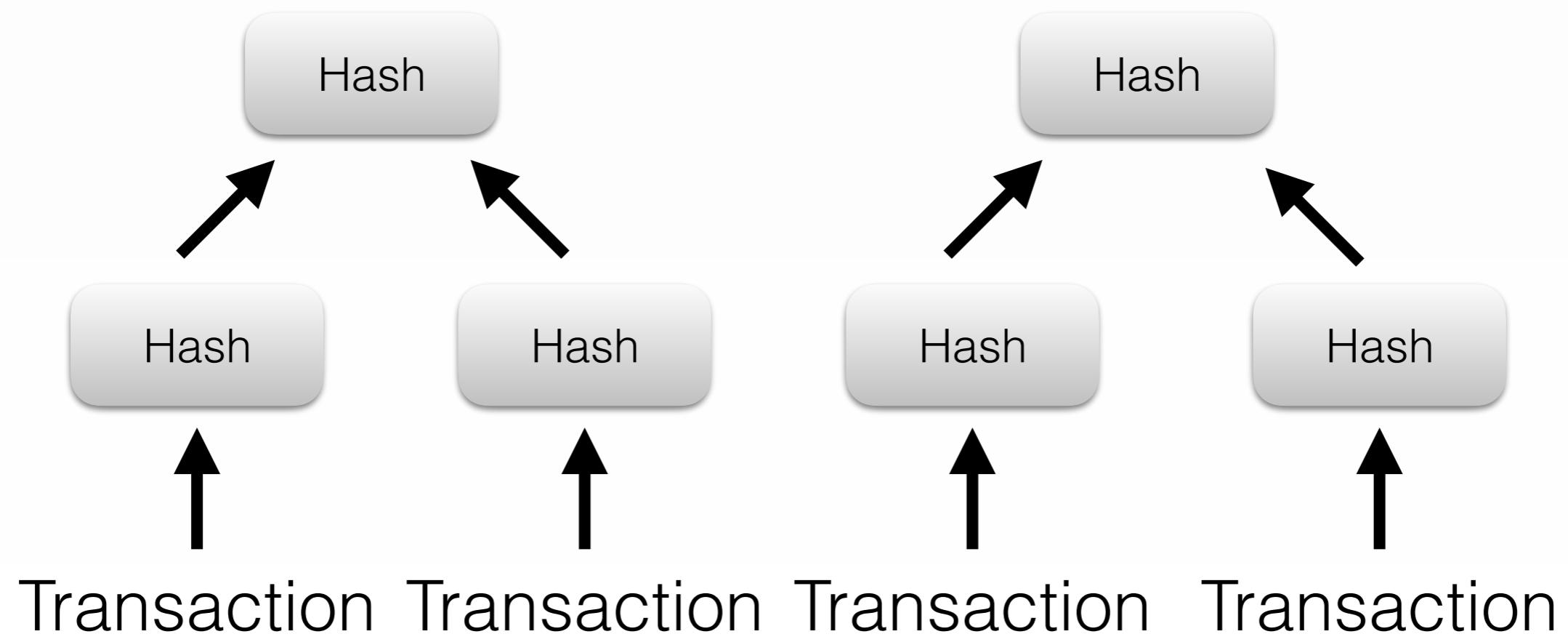


- Cryptographic Hash Functions
- **Merkle Trees**
- Elliptic Curve Signature Algorithm (ECDSA)

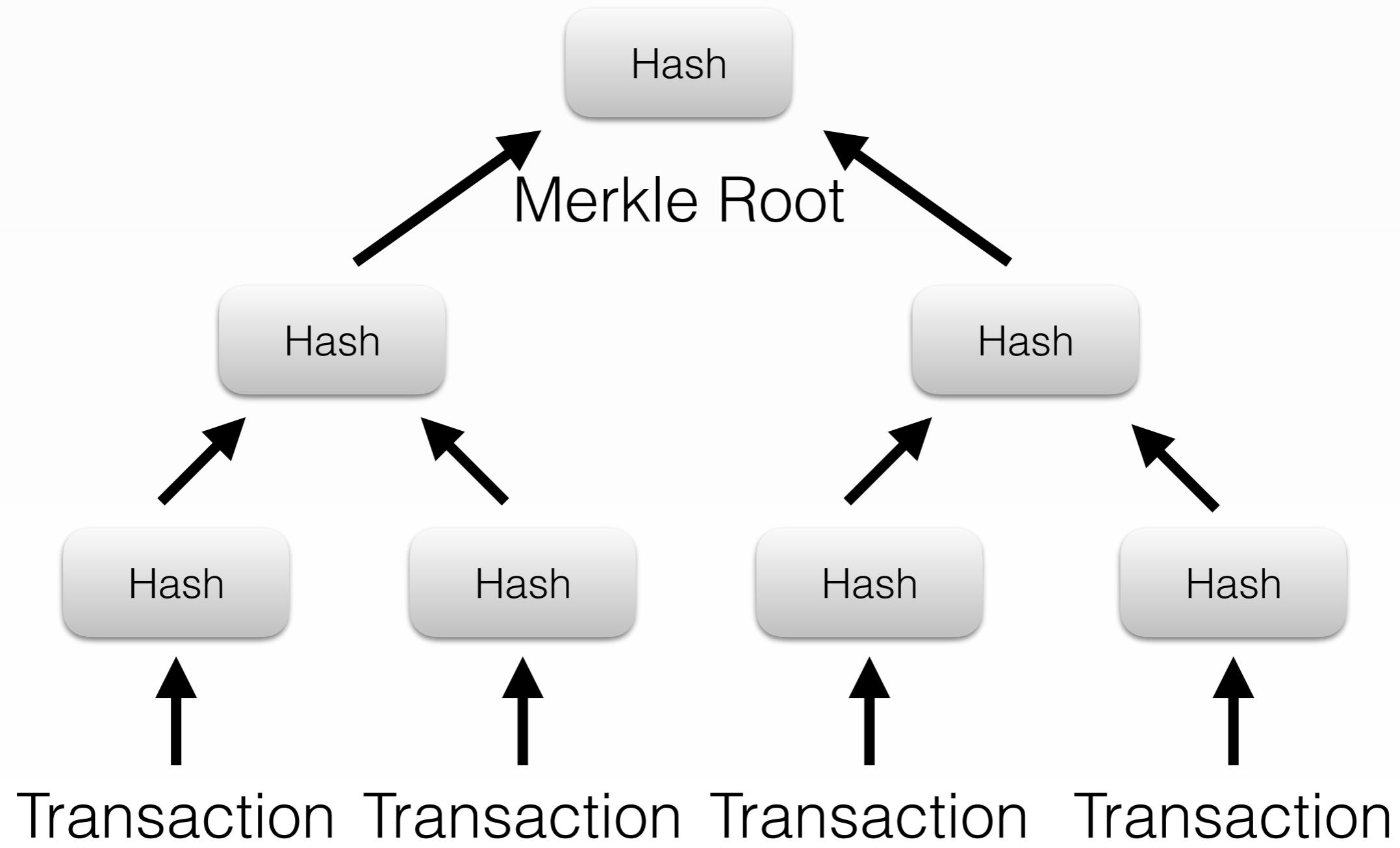
# Data Types



# Data Types



# Data Types



# Data Types



- Cryptographic Hash Functions
- Merkle Trees
- **Elliptic Curve Signature Algorithm (ECDSA)**

## Data Types



- ECDSA (secp256k1 curve) is used to
  - Sign transactions
  - Verify the signature of transactions
  - Nothing in Bitcoin is encrypted
- **Elliptic Curve Signature Algorithm (ECDSA)**



A close-up photograph of a white boat's hull and a metal chain anchor in the water. The chain is rusted and attached to a metal plate on the boat. The water is dark blue.

**Bitcoin  
Addresses**

## Addresses

- Unique identifier
- Hash of a public key
- $0 < \text{Balance} < 21 \text{ Million Bitcoin}$
- $1 \text{ Satoshi} == 10^{-8} \text{ Bitcoin}$

@Alice



@Bob

@Eve

@ETH

**1EGam2BeXd8sgphB44mEYqnDyDszw4YTEr**  
**3J98t1WpEZ73CNmQviecrnyiWrnqRhWNLy**

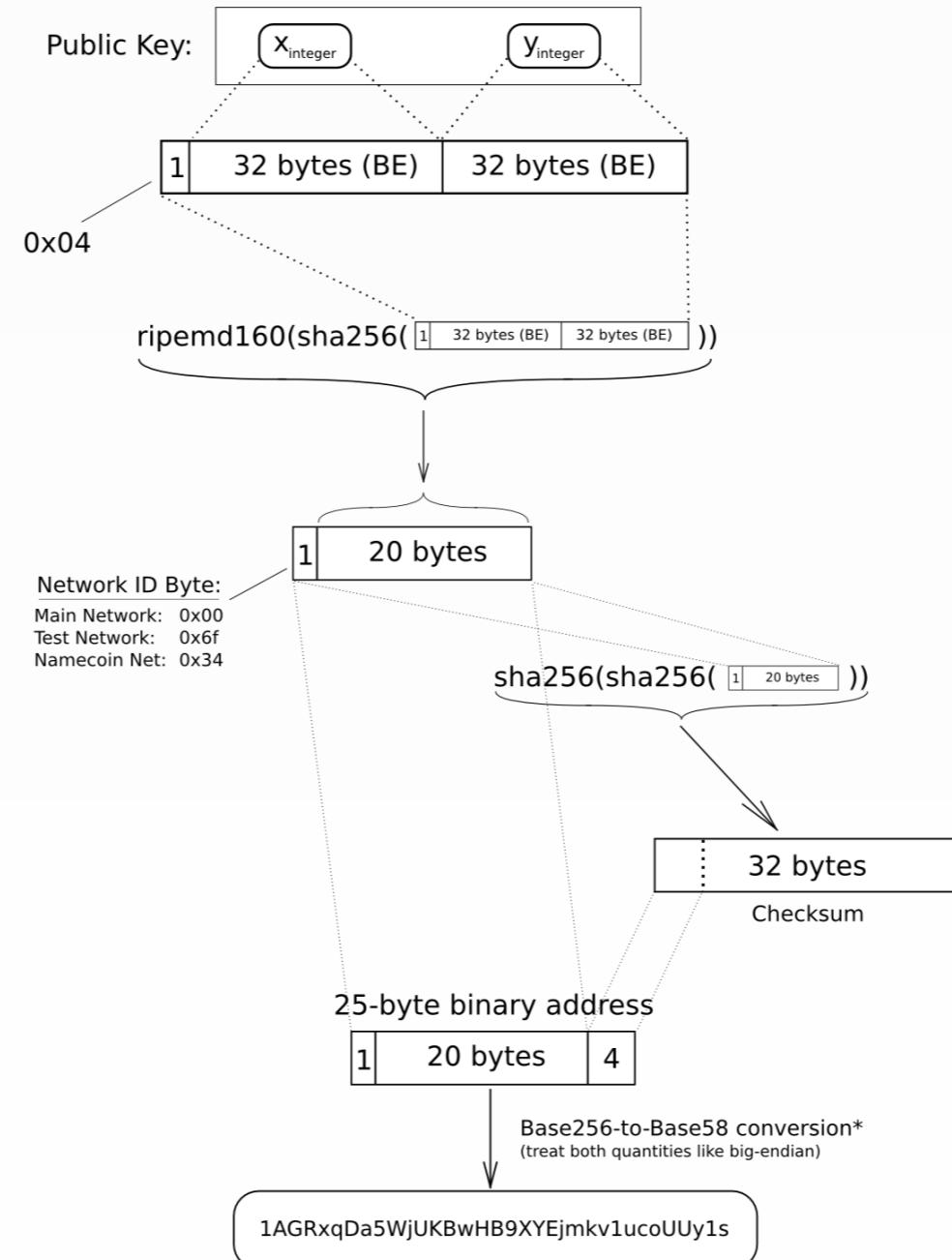
# Addresses

- Unique identifier
- Hash of a public key
- 0 < Balance
- 1 Satoshi

1EGa

3J98t

Elliptic-Curve Public Key to BTC Address conversion



etotheipi@gmail.com / 1Gffm7LKXcNPrtxy6yF4JBoe5rVka4sn1



@Eve

@ETH

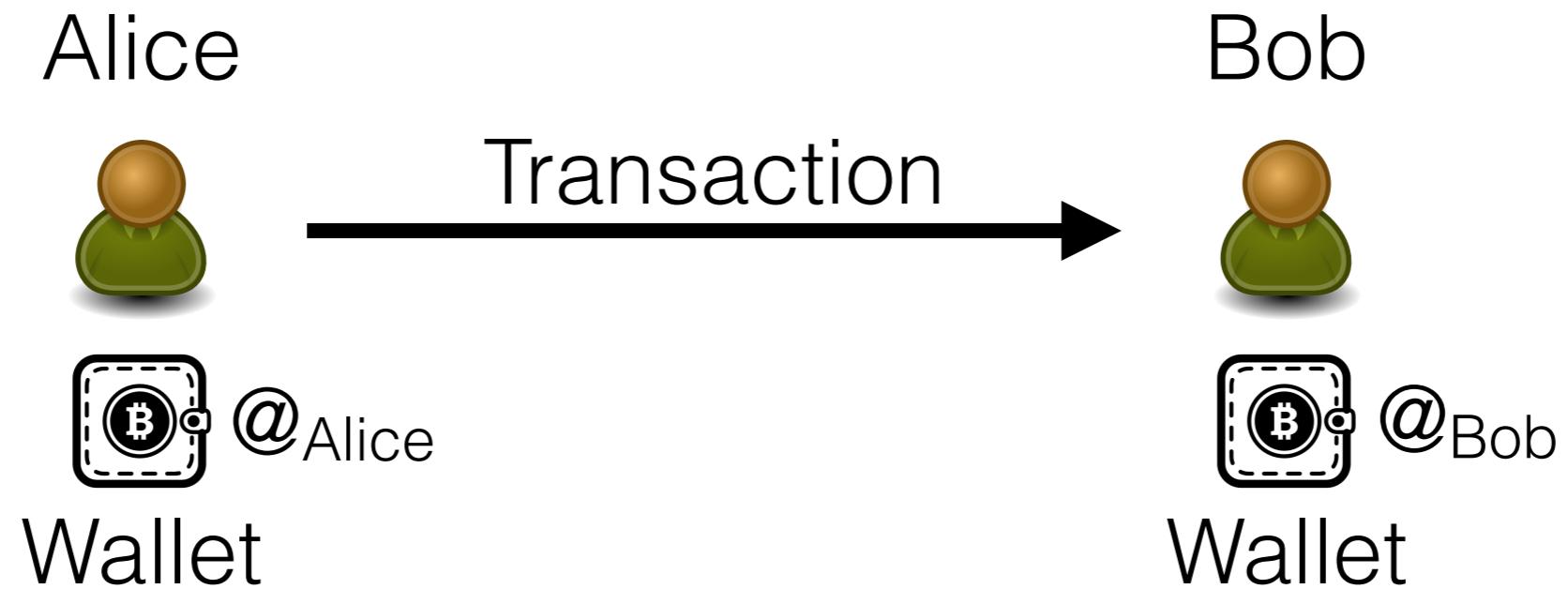
4YTEr

VNLy

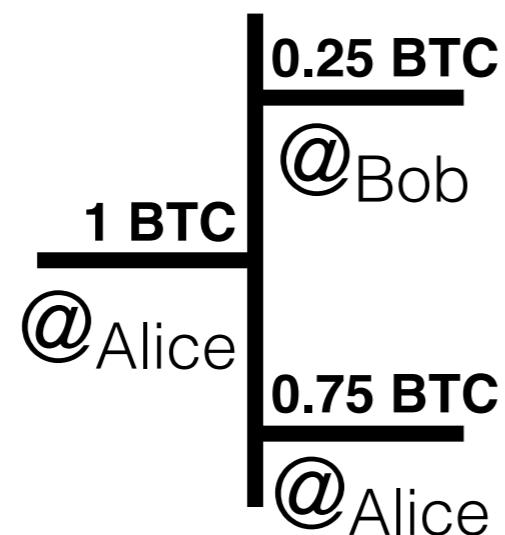
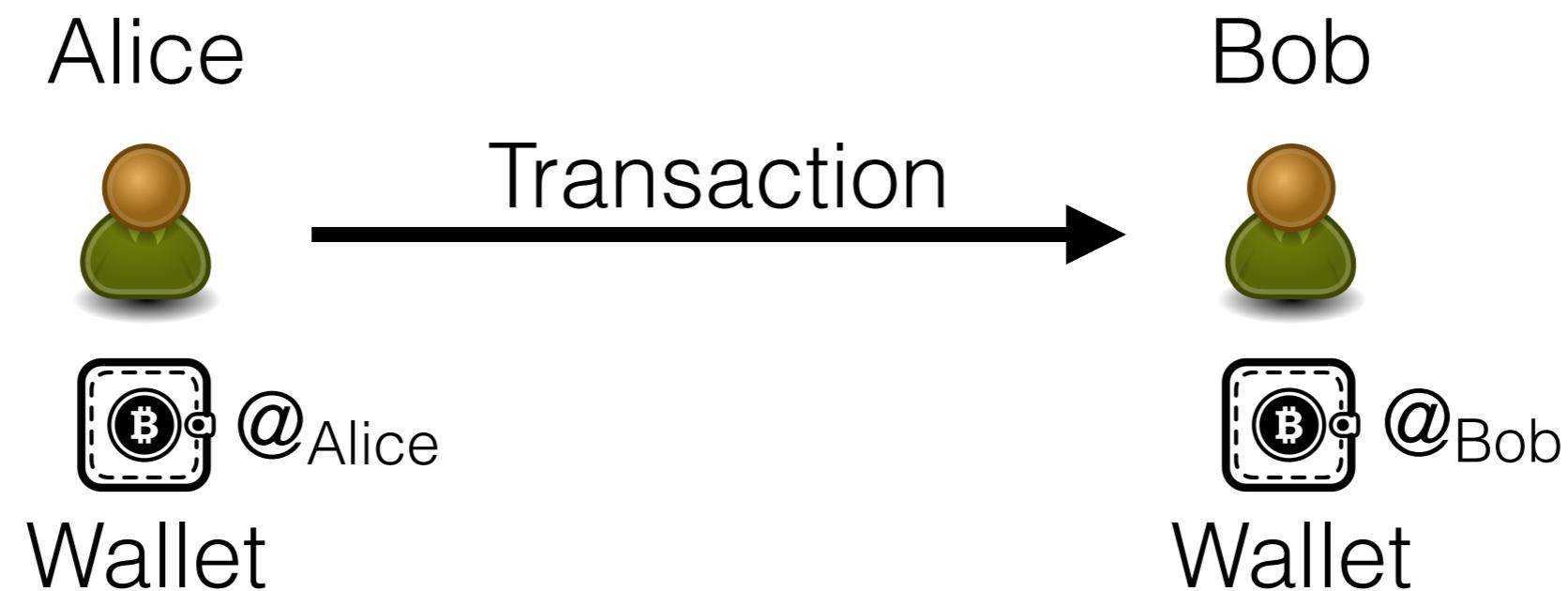
A close-up photograph of a white boat's hull and a metal chain anchor system against a dark blue sea background. The boat's hull is white with some weathering and a metal plate. A thick, rusty chain is attached to the hull and extends into the water. The water is a deep, dark blue.

**Bitcoin  
Transactions**

# Transactions in Bitcoin



# Transactions in Bitcoin



# Transactions in Bitcoin



Alice



Transaction

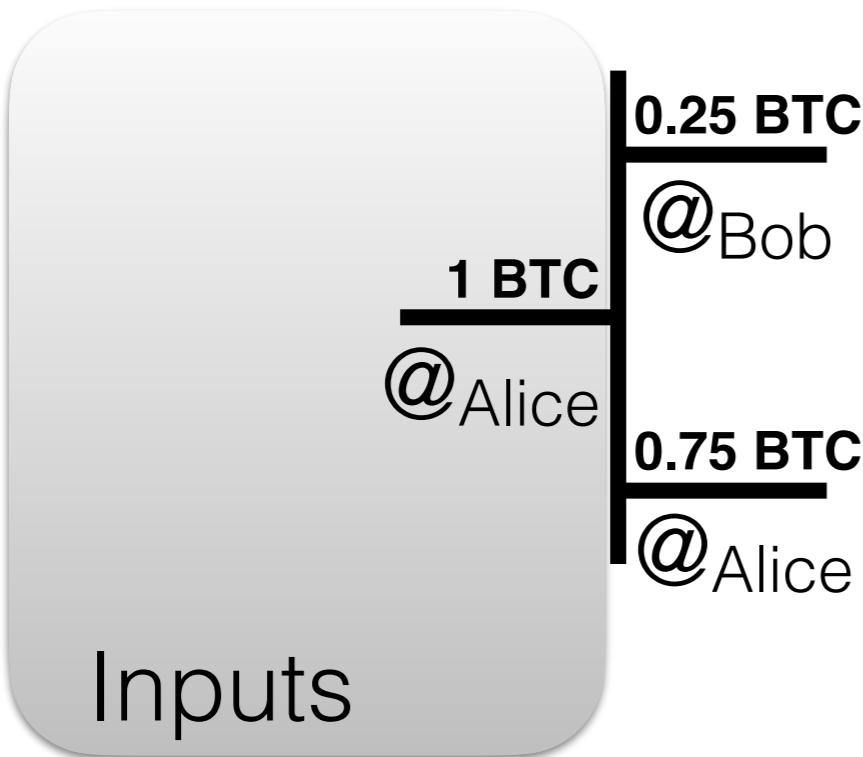
Bob



Wallet



Wallet



# Transactions in Bitcoin



Alice



Transaction

Bob



Wallet

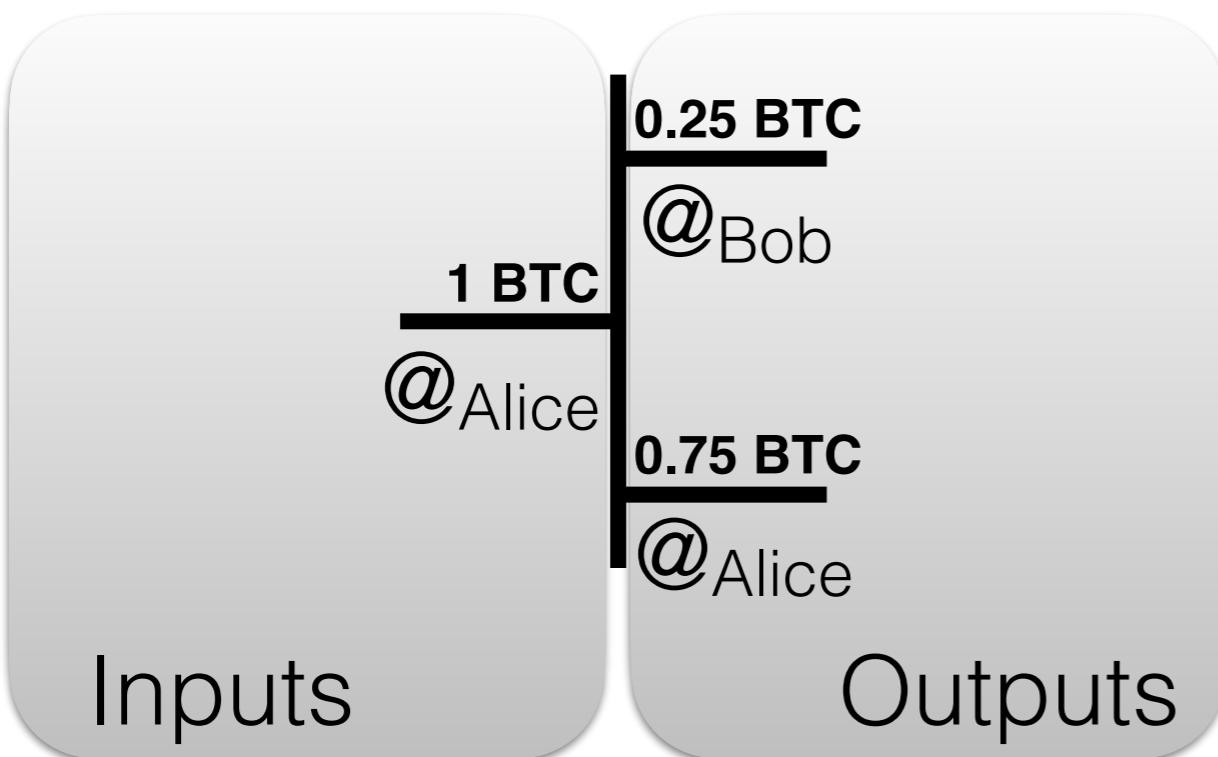


@Alice

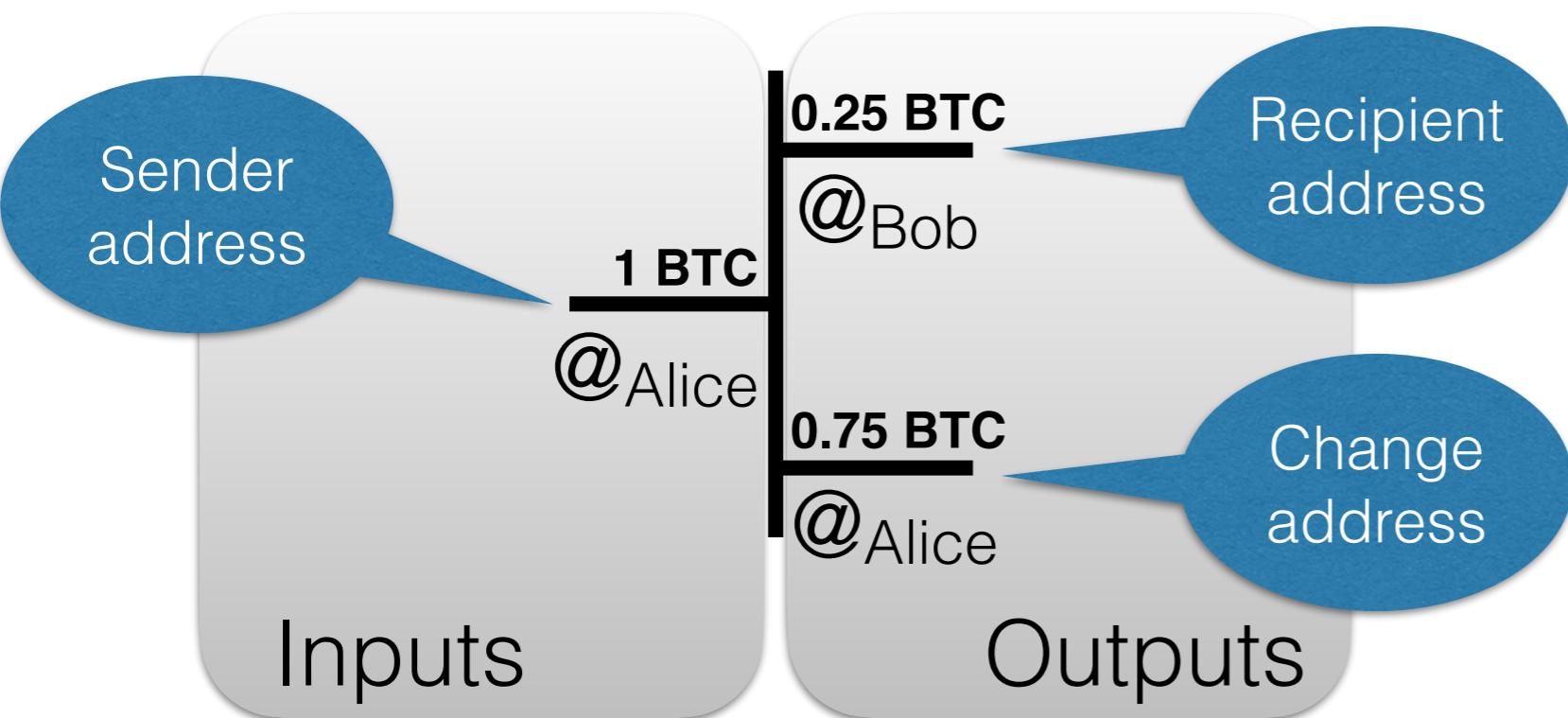
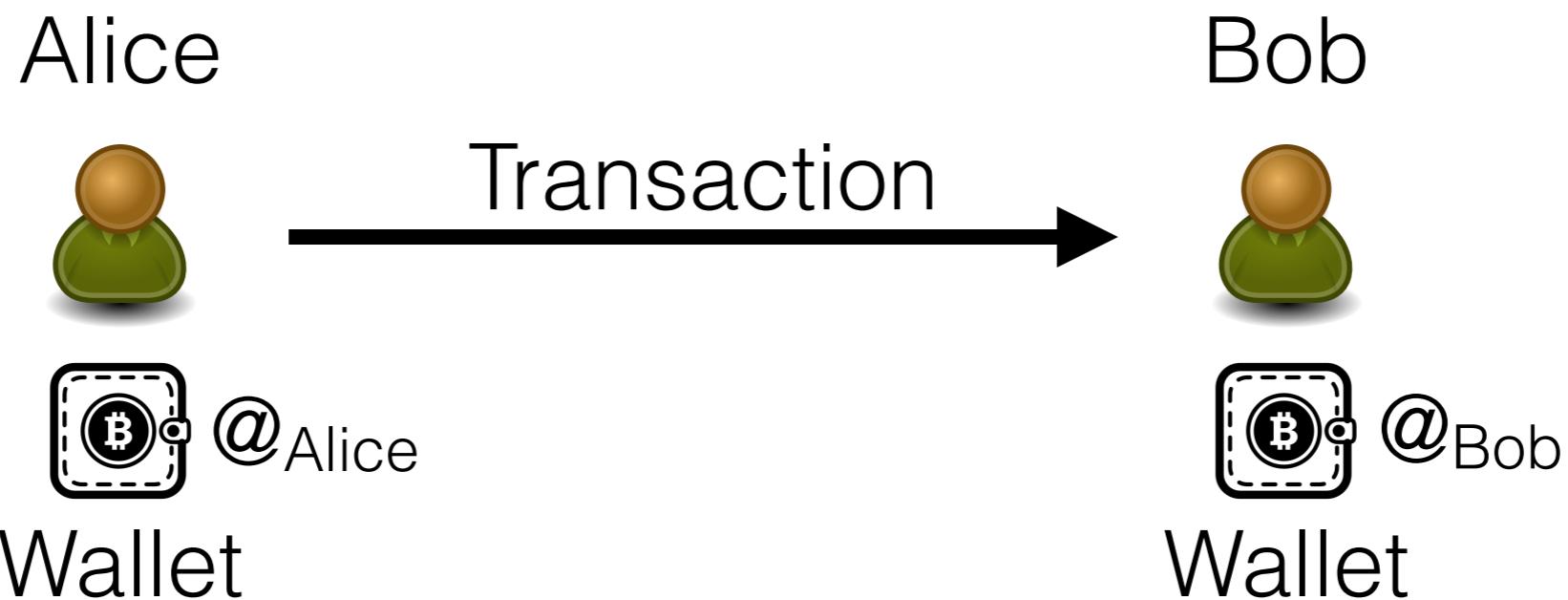
Wallet



@Bob



# Transactions in Bitcoin



# Transactions in Bitcoin



Alice



Transaction

Bob



Wallet

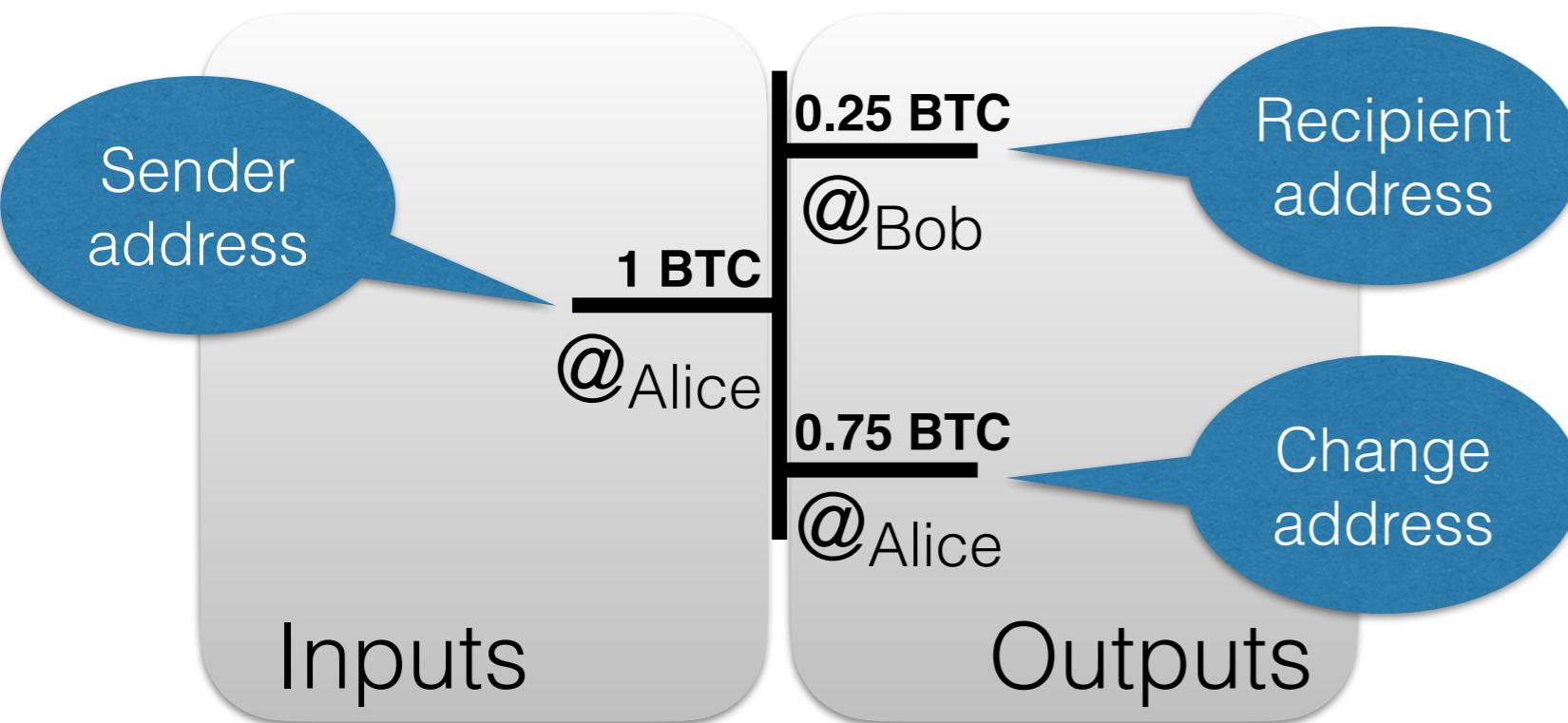


@Alice



@Bob

Wallet

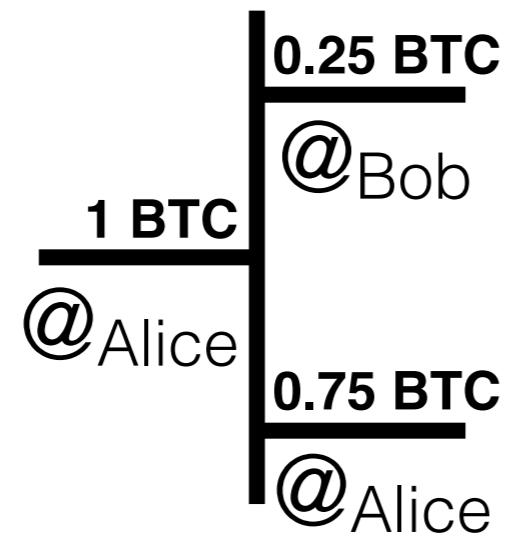


Transaction Fees

$$\sum \text{inputs} \geq \sum \text{outputs}$$

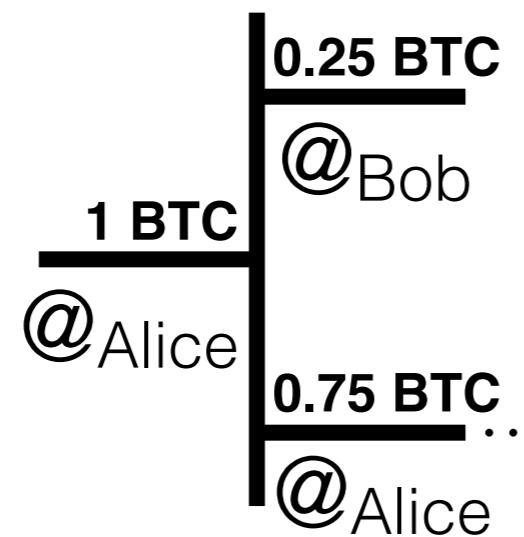
Difference are fees

# Transactions in Bitcoin

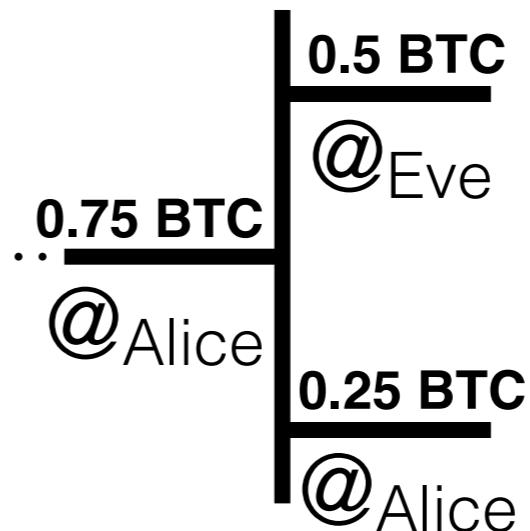


Transaction 1

# Transactions in Bitcoin



Transaction 1



Transaction 2

# Coinbase Transaction

## Coinbase Transaction

First transaction in a Block, defined by the miner.  
No inputs required.

# Coinbase Transaction

First transaction in a Block, defined by the miner.  
No inputs required.

The screenshot shows the Blockchain.info website interface. At the top, a navigation bar includes links for 'BLOCKCHAIN', 'WALLET', 'DATA', 'API', and 'ABOUT'. A search bar and a 'GET A FREE WALLET' button are also present. The main content area is titled 'Transactions' and displays a list of transactions from Bitcoin Block #505344. The first transaction is highlighted with a blue border:

Transaction ID	Date
627a9a7b6f3a9ef57305c0022034060ac403956180b7a7499689ad6dfc1d6a4c	2018-01-21 11:55:42
No Inputs (Newly Generated Coins)	
→ 147SwRQdpCfj5p8PnfsXV2SsVVpVcz3aPq Unable to decode output address	14.86613514 BTC 0 BTC
	14.86613514 BTC

Below this, other transactions are listed:

Transaction ID	Date
a01964c233290c4300e40a0703d6110b1082d20250748167bb0e619e33bard94	2018-01-21 11:33:00
155gmzUdJGpA4mBWXarSSb1e8ctTr62Yv8	
→ 1EdQmjXFRsESLEKd7C2uSBCveDbEyw6YQ4 155gmzUdJGpA4mBWXarSSb1e8ctTr62Yv8	200 BTC 50.92510325 BTC
	250.92510325 BTC

Transaction ID	Date
ff9dcb18e34ce26a0af2c97313e81041545a224aee9001b577dafa0e20b6ff0a	2018-01-21 11:53:11
1LFyMaPq3pAKD5pyPFwGF1TjBRvrie8mZv 1HiSevQbh87QH7dCzagE31ikWRAzsqQLtm 1WTtGb3W8uCBKWjwE4Bz5z9L2833offEe 12dmHXndnZXNci9LJFQgB8fTer8oLCvxoj 1L3FaRHQJbkVZFsE3KqfZdh4woSCZKerp	
→ 1GGRgpcCNWG8yA1awZpcoNTDk8xaJ5Se4v 13MH1iveHM7AGySUSoGfS6pf88H4MboLah 3FY9TMG7ehqjYb6MTfE6gkUjn7rmYhYwiQ 1IEKJCdfeGD9XFnKwtyu5ixLAq4aDxDd 1Hc3sch6WViymxi7jySYC2nDz8djkZjjDc 3BoUhP6THtGPEWNVaGAiDdDqLnD6Jy2MUN 1DDXUHKvKEqFo2wdE7Gf1VEEWbuyHuMs6S 36TDJNavxSvRmDveQw6dhCKzbAvWpAV1yU	0.00458507 BTC 0.00560442 BTC 0.027 BTC 0.00164797 BTC 0.10019393 BTC 0.00140299 BTC 0.0143409 BTC 0.00704523 BTC
	0.16182051 BTC

Transaction ID	Date
b0ae3be3eca7b4272aacb2b0143c48be849c101bd1bf36c34c0db7e2471a456e	2018-01-21 11:55:20
3CPhtdRxpZZv9RgVQ8HibhDUBXLSLU1eY	
→ 37eCSkoPEe4JawGqDWaZXpR1G2ty1ifgfN 3GSbRYCxpxZK1H1BLLDQ84bDCzYfYyDt3R	0.08628748 BTC 0.33275429 BTC
	0.41904177 BTC

A close-up photograph of a white boat's hull and a metal chain anchor in dark blue water. The boat's hull is visible on the left, showing some wear and a metal plate with a chain attached. A thick, rusty metal chain runs across the frame, secured to a metal plate on the hull. The water is a deep, dark blue.

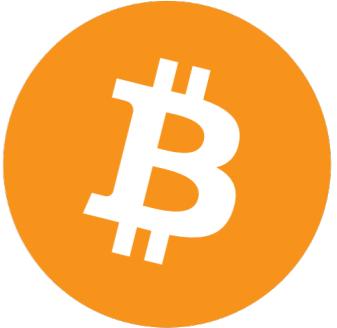
**Bitcoin  
Script**

# Script



- Stack based programming language
- If evals to *true* —> Bitcoin transaction is valid
- Many opcodes
- Execution time is critical to prevent DoS attacks

# Script



- Stack based programming language
- If evals to *true* —> Bitcoin transaction is valid
- Many opcodes
- Execution time is critical to prevent DoS attacks

## Example Script

```
<signature><publicKey> OP_CHECKSIG
```

# Script



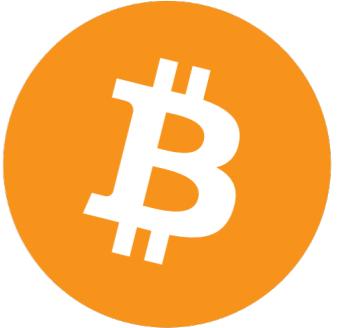
- Stack based programming language
- If evals to *true* —> Bitcoin transaction is valid
- Many opcodes
- Execution time is critical to prevent DoS attacks

## Example Script

```
<signature><publicKey> OP_CHECKSIG
```

Constants  
are pushed onto the  
stack

# Script



- Stack based programming language
- If evals to *true* —> Bitcoin transaction is valid
- Many opcodes
- Execution time is critical to prevent DoS attacks

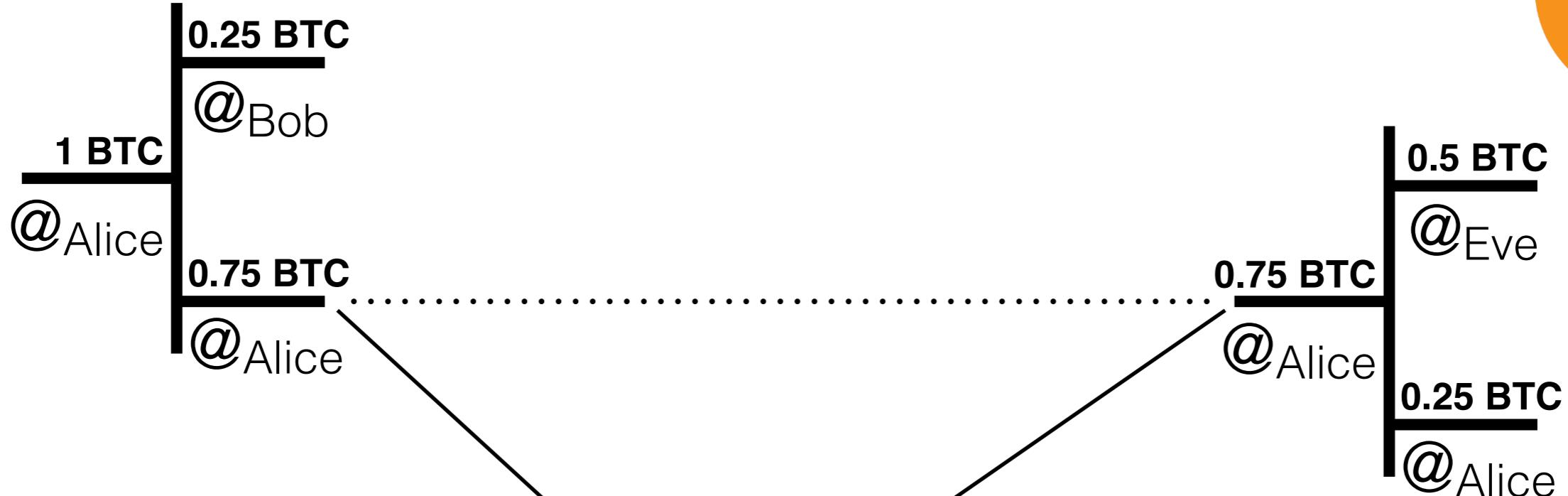
## Example Script

<signature><publicKey> OP\_CHECKSIG

Constants  
are pushed onto the  
stack

Operation  
executes on stack  
values

# Script Execution



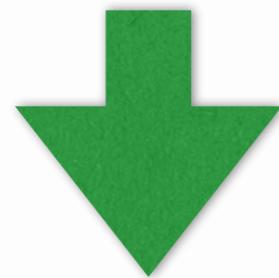
<Sig> <PubKey> OP\_DUP OP\_HASH160 <PubKeyHash> OP\_EQUALVERIFY OP\_CHECKSIG

<PubKey>	
<Sig>	OP_DUP
Execution Stack	
	Execution Code

Constants are pushed onto the stack

<Sig> <PubKey> OP\_DUP OP\_HASH160 <PubKeyHash> OP\_EQUALVERIFY OP\_CHECKSIG

<PubKey>	
<Sig>	OP_DUP
Execution Stack	
	Execution Code



<PubKey>	
<PubKey>	
<Sig>	OP_HASH160
Execution Stack	
	Execution Code

Constants are pushed onto the stack

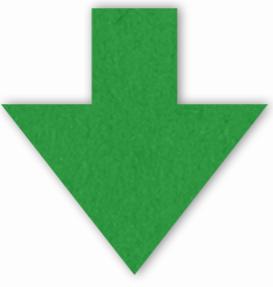
<Sig> <PubKey> OP\_DUP OP\_HASH160 <PubKeyHash> OP\_EQUALVERIFY OP\_CHECKSIG

<PubKey>	
<Sig>	
Execution Stack	

	OP_DUP
Execution Code	

<PubKeyHash>	
<PubKeyHash>	
<PubKey>	
<Sig>	
Execution Stack	

	OP_EQUALVERIFY
Execution Code	

A large green downward-pointing arrow is positioned between the first and second stack states.

<PubKey>	
<PubKey>	
<Sig>	
Execution Stack	

	OP_HASH160
Execution Code	

Constants are pushed onto the stack

<Sig> <PubKey>

OP\_DUP OP\_HASH160 <PubKeyHash> OP\_EQUALVERIFY OP\_CHECKSIG

<PubKey>	
<Sig>	
Execution Stack	OP_DUP
	Execution Code

<PubKeyHash>	
<PubKeyHash>	
<PubKey>	
<Sig>	OP_EQUALVERIFY
Execution Stack	
	Execution Code

<PubKey>	
<PubKey>	
<Sig>	OP_HASH160
Execution Stack	
	Execution Code

<PubKey>	
<Sig>	OP_CHECKSIG
Execution Stack	
	Execution Code

Constants are pushed onto the stack

<Sig> <PubKey> OP\_DUP OP\_HASH160 <PubKeyHash> OP\_EQUALVERIFY OP\_CHECKSIG

<PubKey>	
<Sig>	OP_DUP
Execution Stack	Execution Code

<PubKeyHash>	
<PubKeyHash>	
<PubKey>	
<Sig>	OP_EQUALVERIFY
Execution Stack	Execution Code

<PubKey>	
<PubKey>	
<Sig>	OP_HASH160
Execution Stack	Execution Code

<PubKey>	
<Sig>	OP_CHECKSIG
Execution Stack	Execution Code

Constants are pushed onto the stack

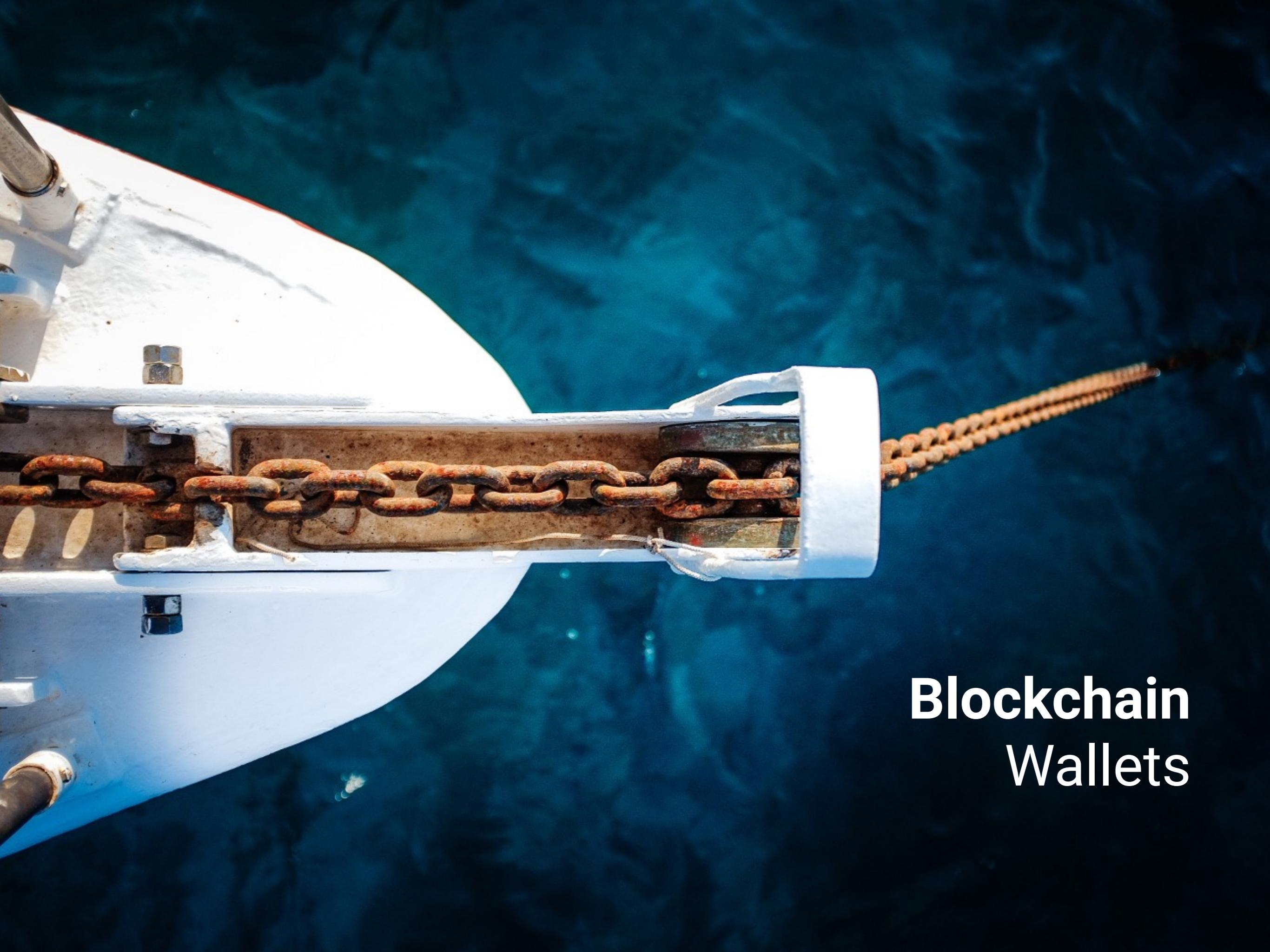
<Sig> <PubKey> OP\_DUP OP\_HASH160 <PubKeyHash> OP\_EQUALVERIFY OP\_CHECKSIG

True

## Transaction Types



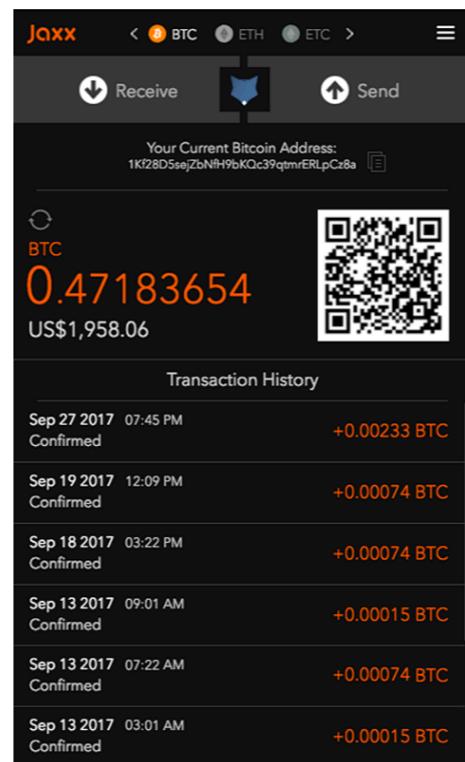
- P2PKH - Pay to Public Key Hash
  - Redeemer needs a public key and signature
- P2SH - Pay to Script Hash
  - Redeemer needs a script that matches a pre-defined hash
- Multisignature (m-n)
  - Requires multiples signatures to be redeemable
  - **m** out of **n** signatures required

A close-up photograph of a white boat's hull and a metal chain anchor in dark blue water. The chain is attached to a metal plate on the hull. The water is dark and reflects the light from the boat.

**Blockchain  
Wallets**

# Blockchain Wallets

- Bitcoin Core  
Full node, downloads all transactions
- Bitcoin Wallet  
Android
- Jaxx  
Multi-chain Wallet
- Ledger Nano S



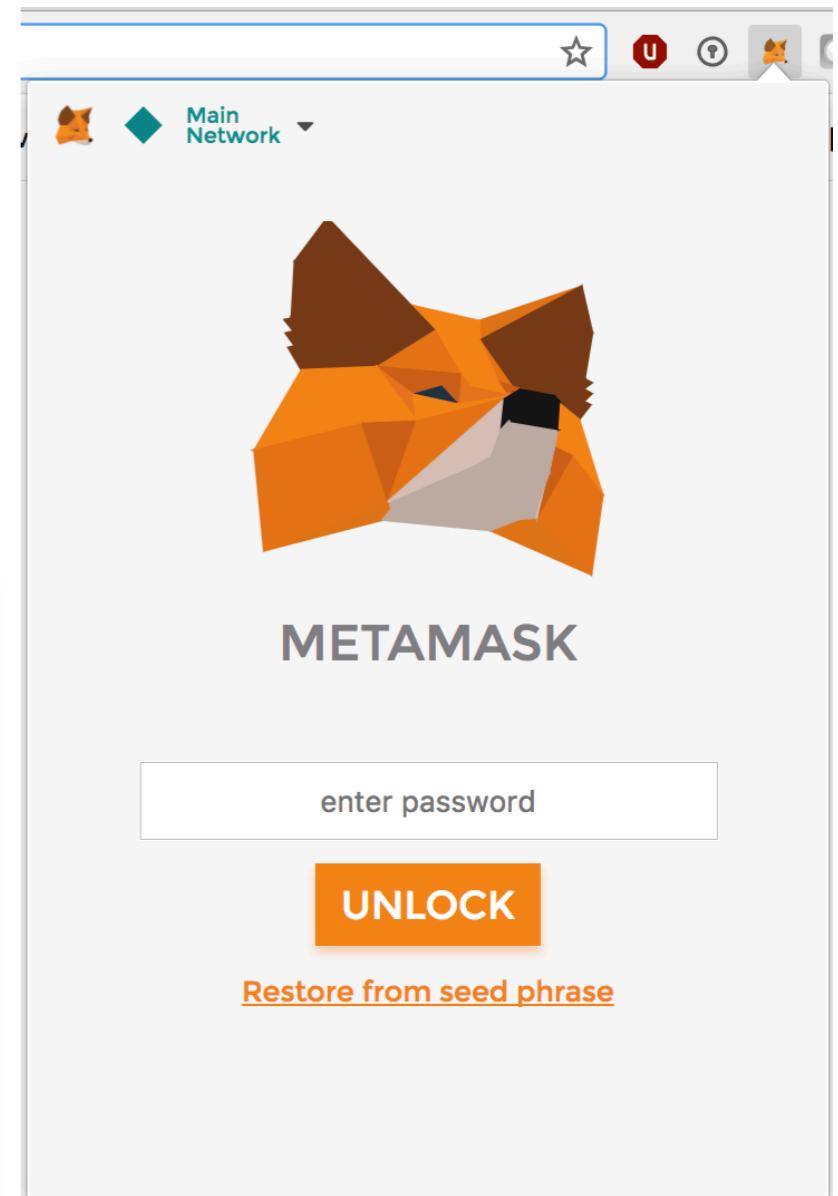
# Blockchain Wallets

- MetaMask
- MyEtherWallet

The screenshot shows the 'Create New Wallet' page of MyEtherWallet.com. At the top, there's a red banner with a warning: 'DON'T GET PHISHED, please! 🚫 Thank you! 🌟' followed by instructions: '1. BOOKMARK MYETHERWALLET.COM | 2. INSTALL EAL or MetaMask or Cryptonite'. Below the banner, the header includes the MyEtherWallet logo, version 3.11.2.3, language selection (English), gas price (41 Gwei), and network (ETH). A note says 'The network is really full right now. Check Eth Gas Station for gas price to use.' The main form has a placeholder 'Enter a password' and a note 'Do NOT forget to save this!' with a copy icon. A blue button says 'Create New Wallet'. Below the form, a note states: 'This password encrypts your private key. This does not act as a seed to generate your keys. You will need this password + your private key to unlock your wallet.' At the bottom, links for 'How to Create a Wallet' and 'Getting Started' are visible.

MyEtherWallet.com does not hold your keys for you. We cannot access accounts, recover keys, reset passwords, nor reverse transactions. Protect your keys & always check that you are on correct URL. [You are responsible for your security.](#)

You can support us by supporting our blockchain-family.  
Consider using our affiliate links to...  
[Swap ETH/BTC/EUR/CHF via Bitly.com](#)



# Ethereum Testnet Transaction

The screenshot shows the MyEtherWallet.com interface for sending a transaction. The top navigation bar includes links for New Wallet, Send Ether & Tokens (highlighted), Swap, Send Offline, Contracts, Check TX Status, View Wallet Info, and Help. The version is 3.11.2.3, the language is English, the gas price is 41 Gwei, and the network is set to Kovan (Etherscan.io). A red banner at the top urges users to bookmark the site and install EAL or MetaMask.

**To Address:** 0x0D0bE22D14C3C7f4b754DBd1838b36fdd0E23c72

**Amount to Send:** 0.001 KOVAN ETH

**Gas Limit:** 21084

**Data:** 0x000000001

**Generate Transaction**

**Account Address:** 0x0D0bE22D14C3C7f4b754DBd1838b36fdd0E23c72

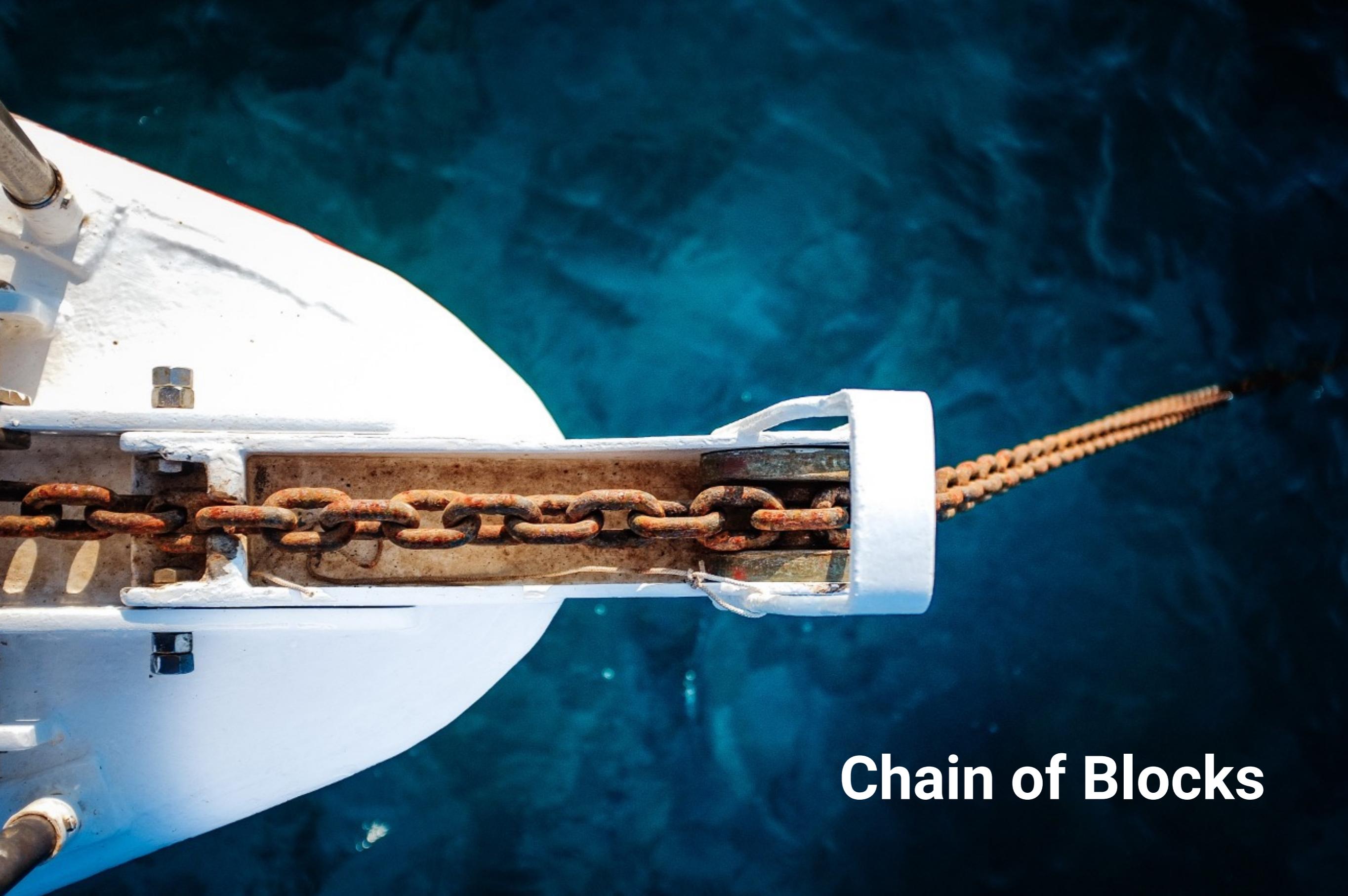
**Account Balance:** 0.57405 KOVAN ETH

**Transaction History:** KOVAN ETH (<https://kovan.etherscan.io>)

**Learn more about protecting your funds:** Ledger, TREZOR

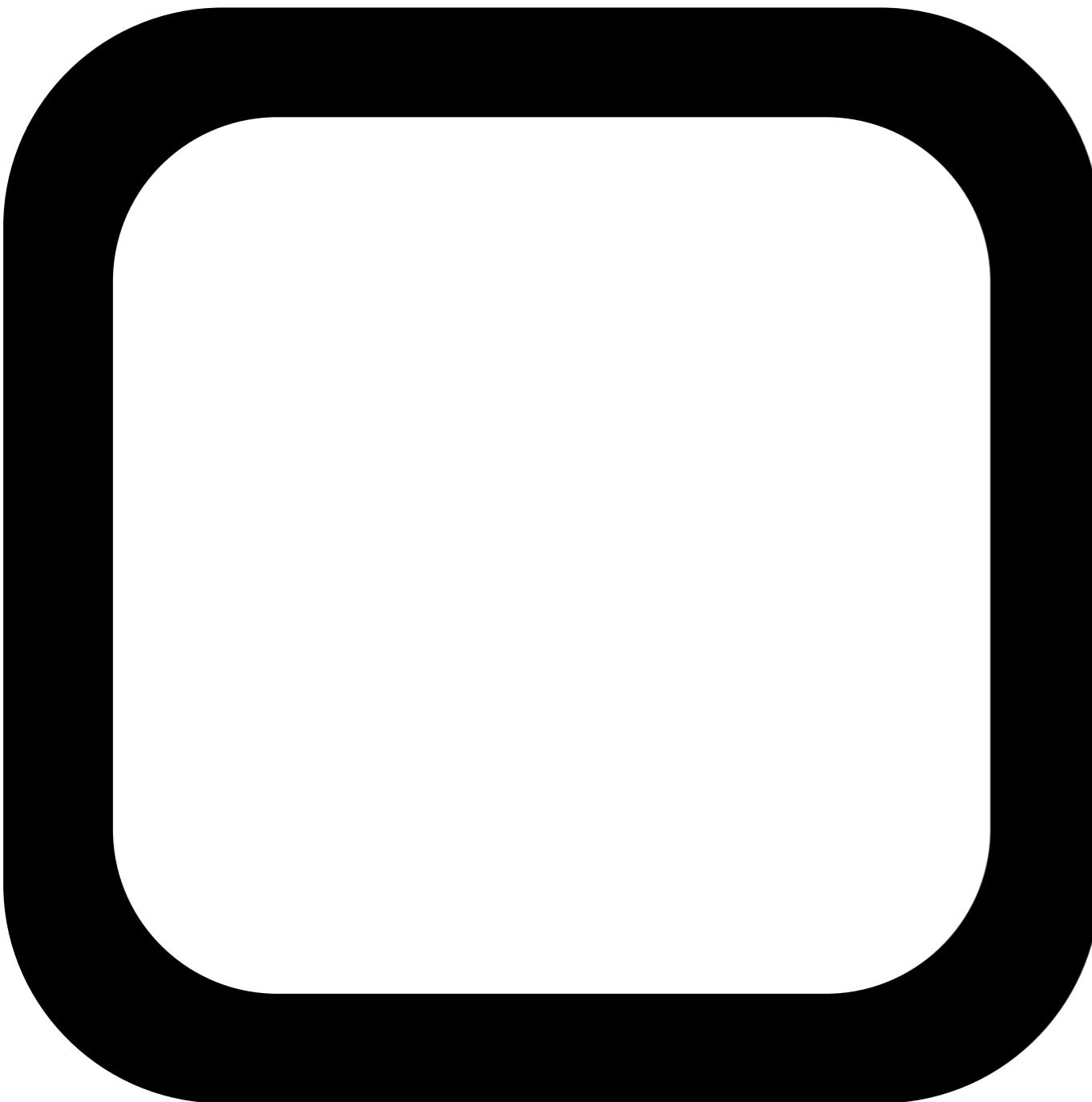
**Token Balances:** How to See Your Tokens. You can also view your balances on <https://kovan.etherscan.io>.

Show All Tokens | Add Custom Token



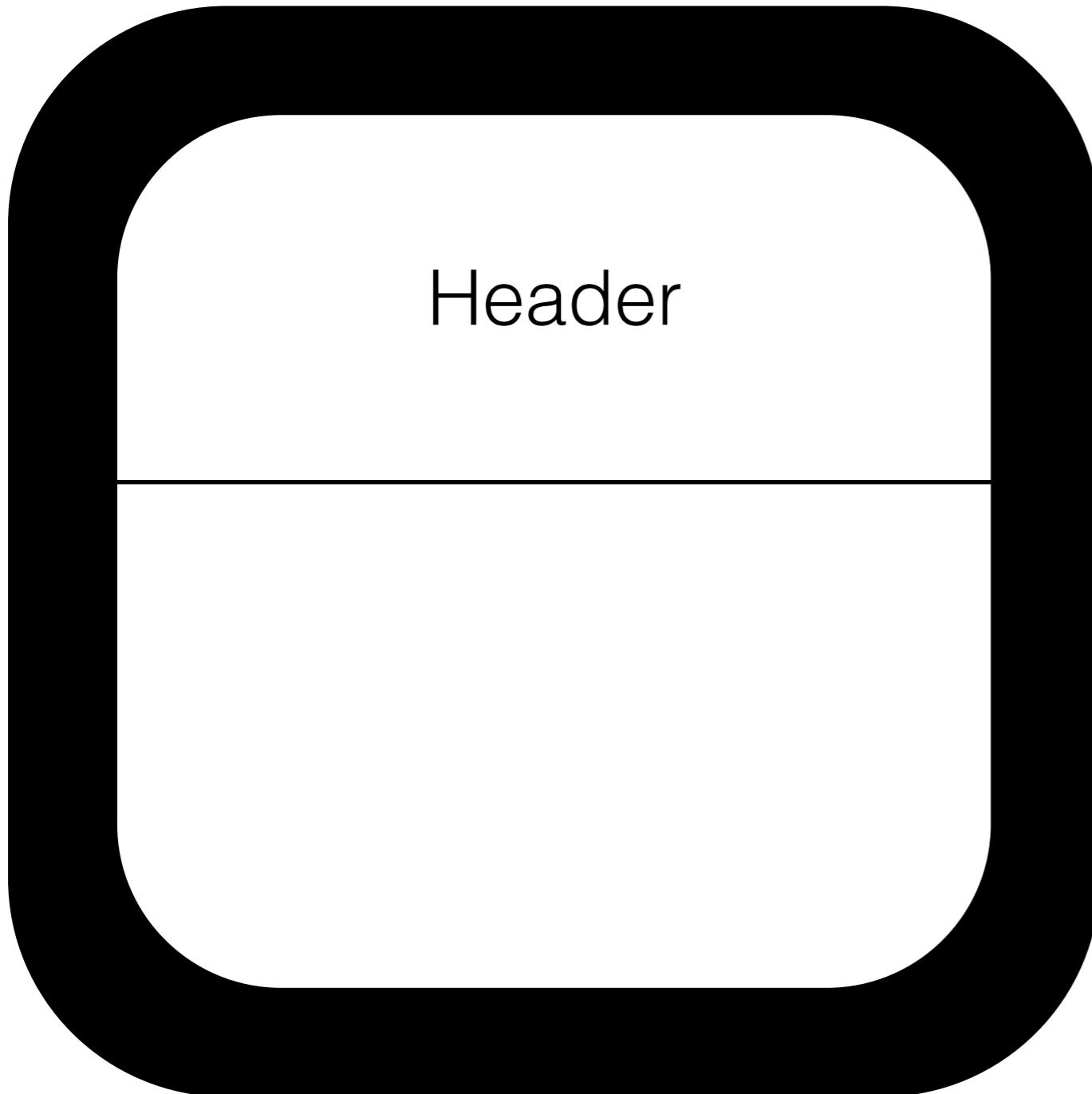
**Chain of Blocks**

# Block



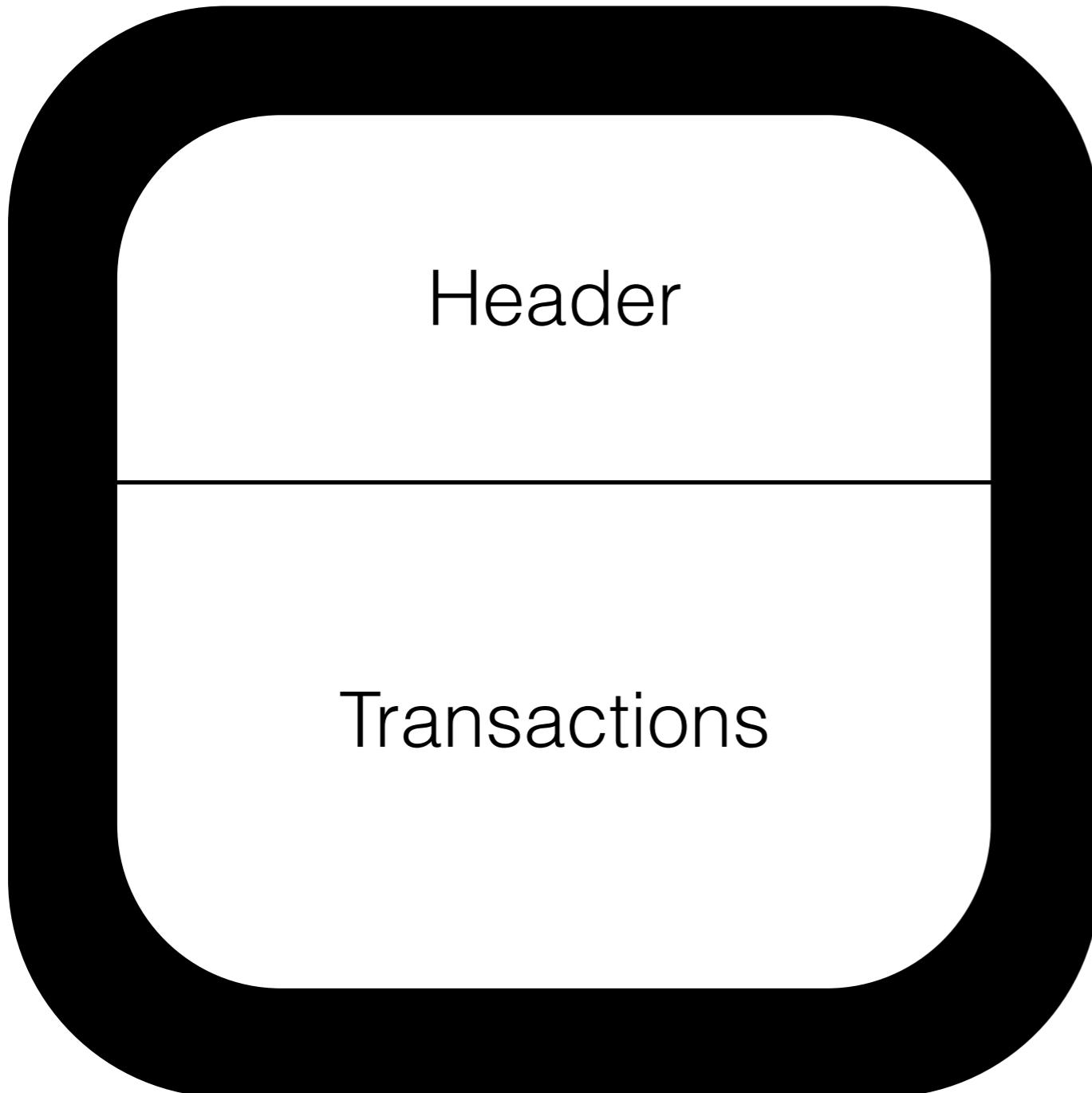
<https://bitcoin.org/en/developer-reference#block-headers>

# Block

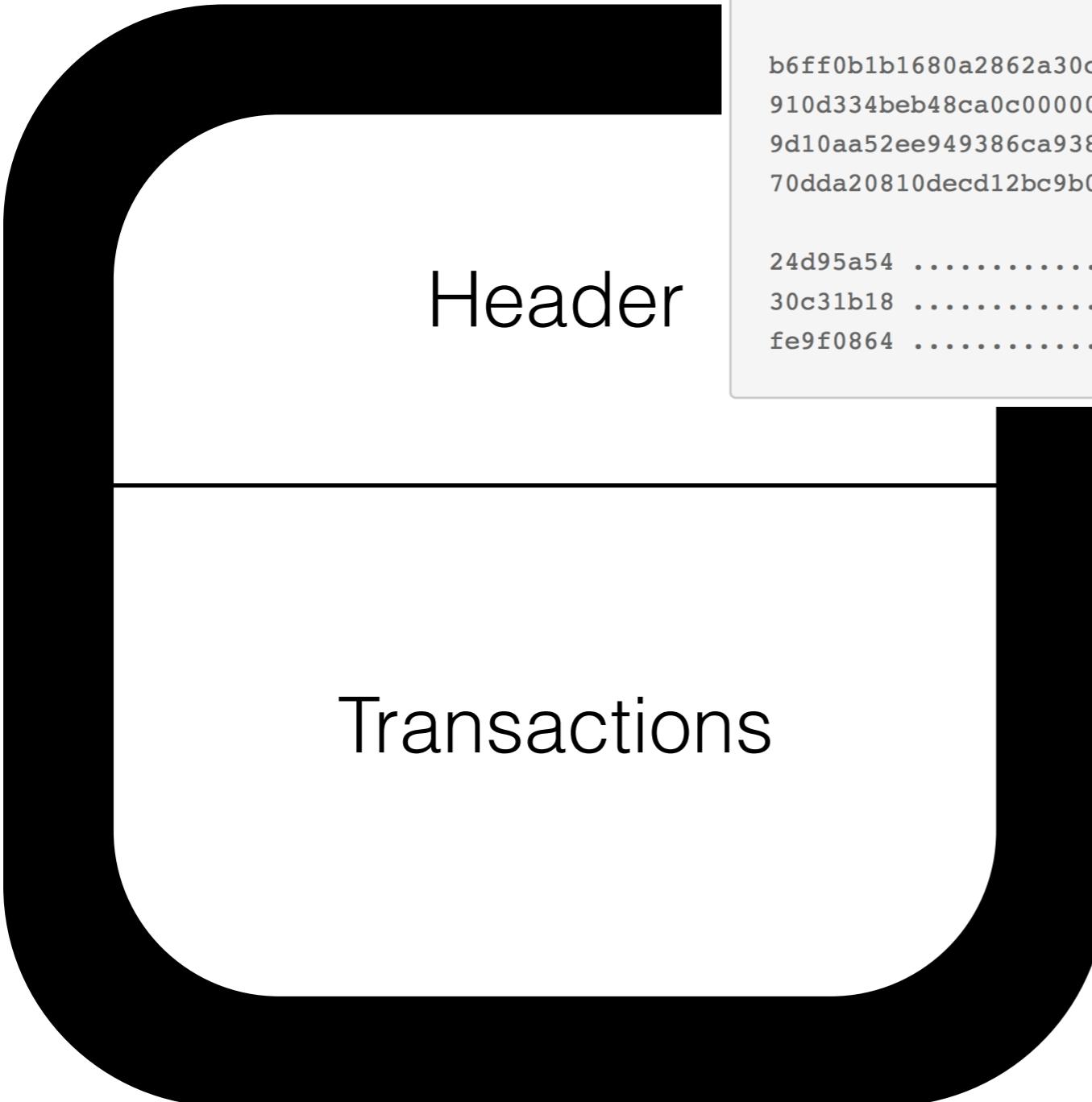


<https://bitcoin.org/en/developer-reference#block-headers>

# Block

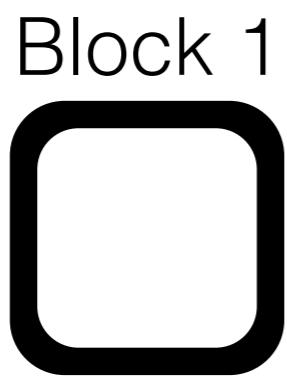


# Block



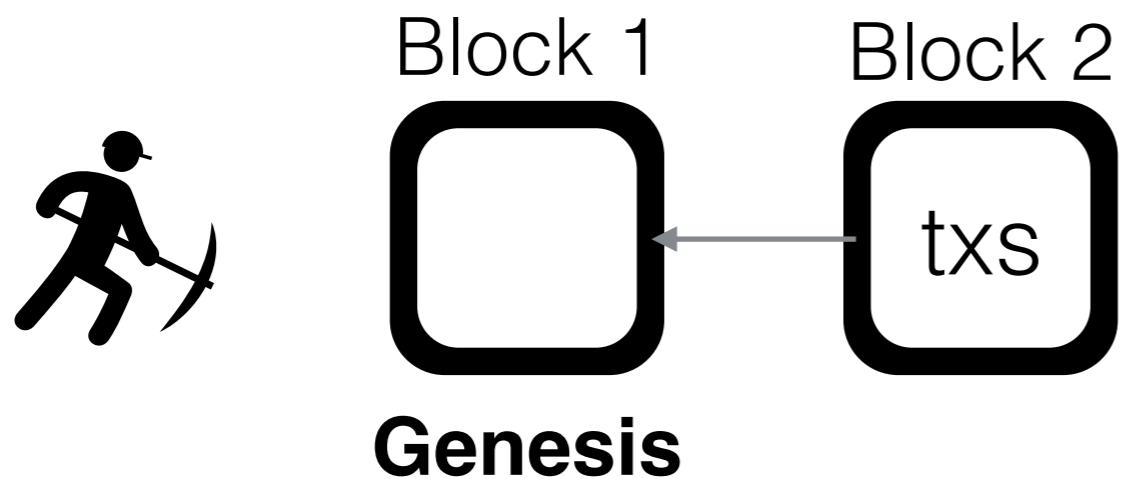
```
02000000 ..... Block version: 2  
b6ff0b1b1680a2862a30ca44d346d9e8  
910d334beb48ca0c0000000000000000 ... Hash of previous block's header  
9d10aa52ee949386ca9385695f04ede2  
70dda20810decd12bc9b048aaab31471 ... Merkle root  
24d95a54 ..... Unix time: 1415239972  
30c31b18 ..... Target: 0x1bc330 * 256**(0x18-3)  
fe9f0864 ..... Nonce
```

# Blockchain

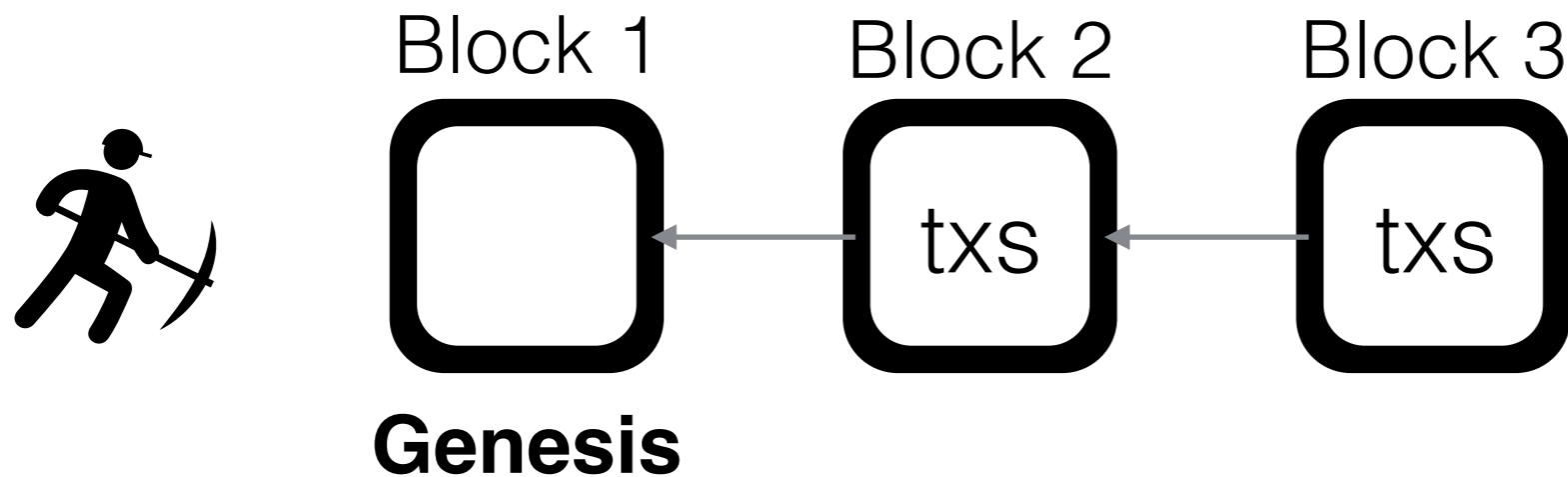


**Genesis**

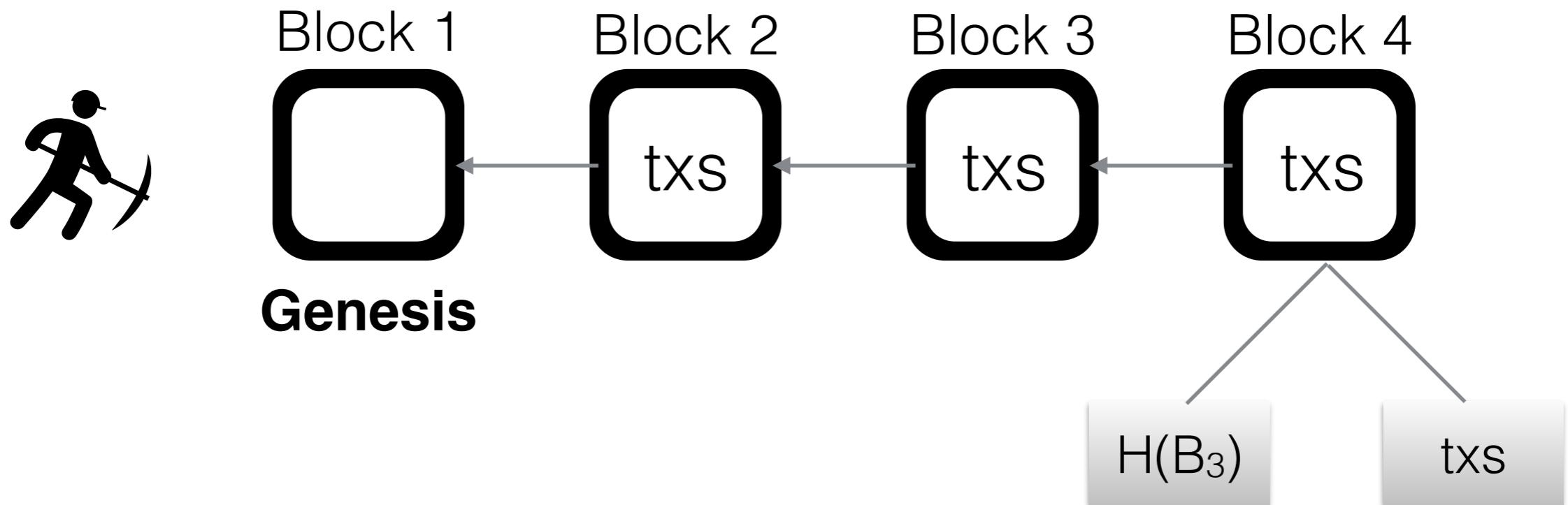
# Blockchain



# Blockchain



# Blockchain

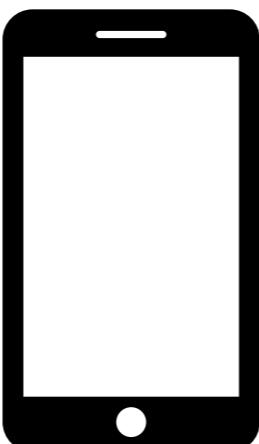


A close-up photograph of a white boat's hull and a metal chain anchor in dark blue water. The chain is attached to a metal plate on the hull. The water is dark and reflects the light.

# Bitcoin Lightweight Clients

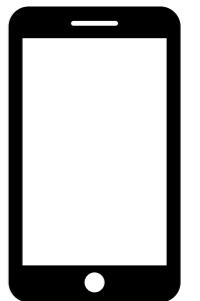
## Why Lightweight Clients?

- Mainstream will not store >300 GB of data for a wallet
- Mainstream will not use substantial CPU power to validate transactions of other users
- Transaction security should still be high



# Scaling Problems for Lightweight Clients

1. Log of transactions (>120 GB)



2. Mobile phones receive irrelevant transactions

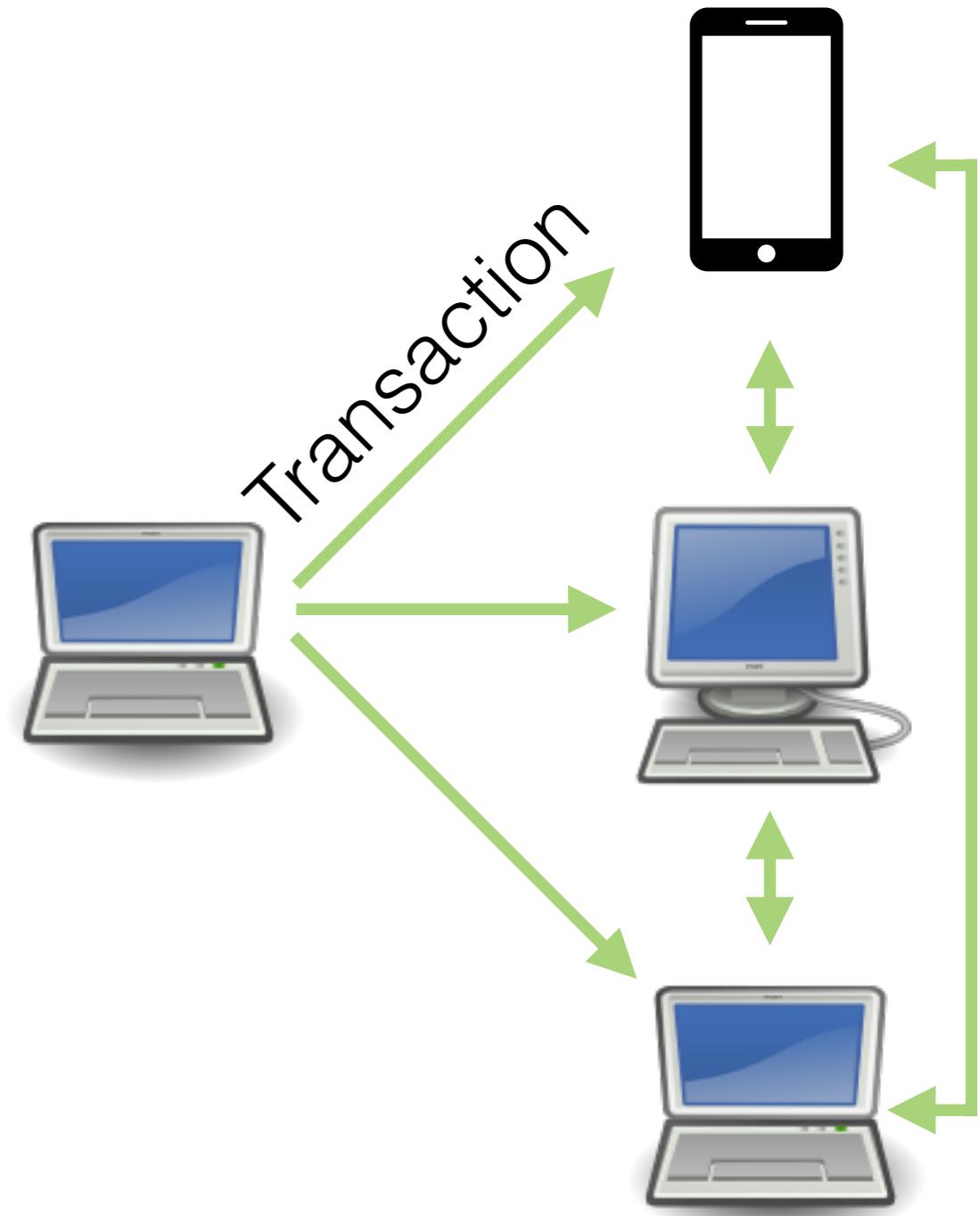


3. Limited data traffic over 3G/4G

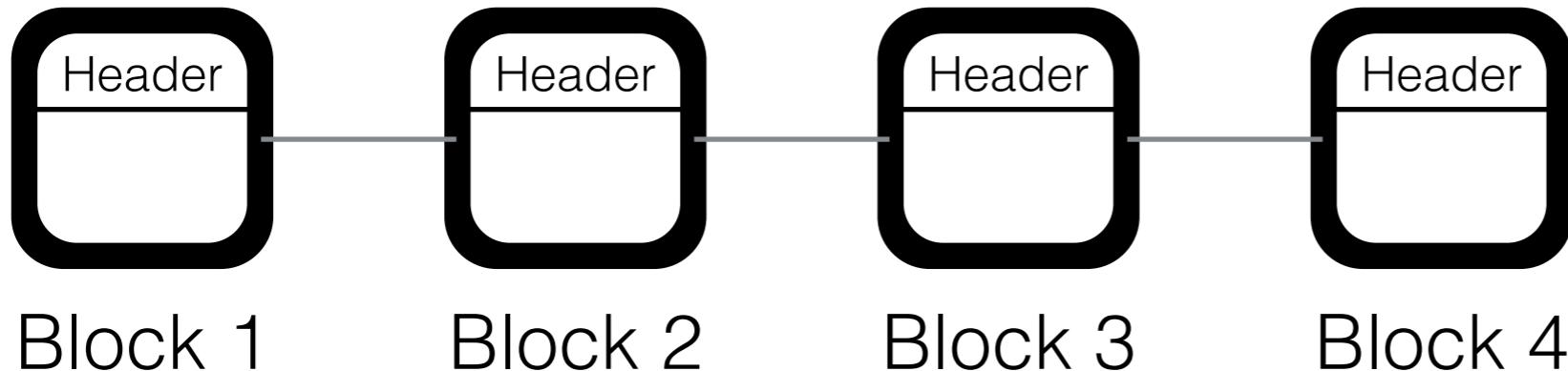


# Scaling Problems for Lightweight Clients

1. Log of transactions (>120 GB)
2. Mobile phones receive irrelevant transactions
3. Limited data traffic over 3G/4G



## SPV Transaction verification



Blockchain  
with headers  
**only**

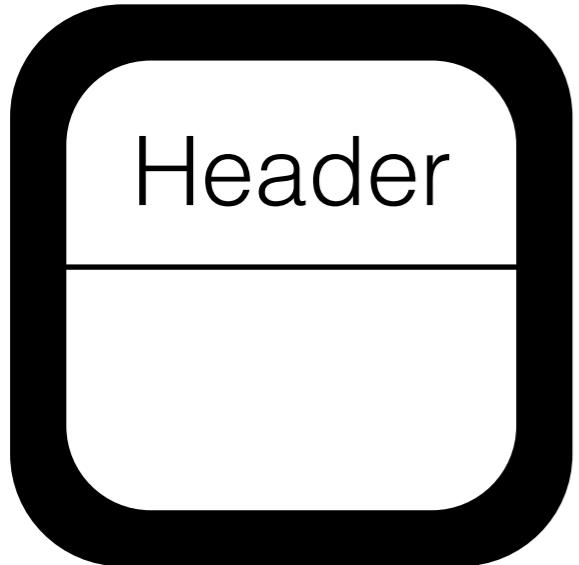
- Much smaller
- Can verify the Proof of Work
- Can find the longest chain with most work!

## SPV Transaction verification



Blockchain  
with headers  
**only**

## SPV Transaction verification



Blockchain  
with headers  
**only**

Transaction

## SPV Transaction verification



Blockchain  
with headers  
**only**

Transaction

Is this transaction in the  
blockchain?

# SPV Transaction verification



Blockchain  
with headers  
**only**

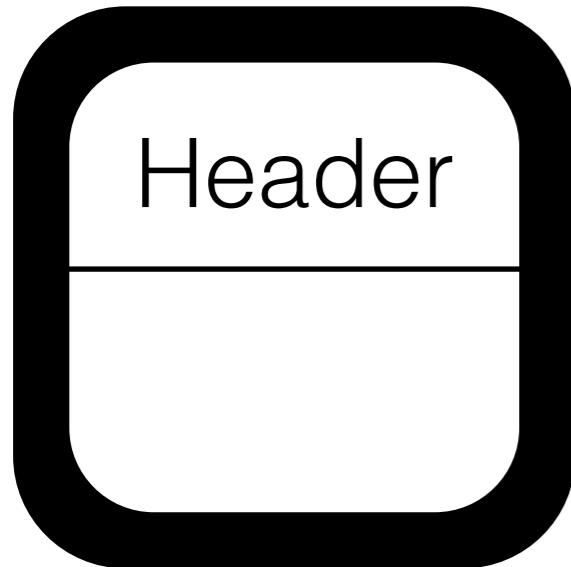


Transaction **Transaction**

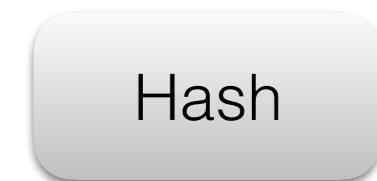
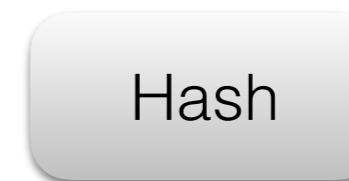
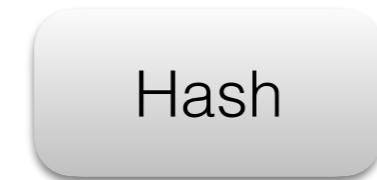


Is this transaction in the  
blockchain?

## SPV Transaction verification



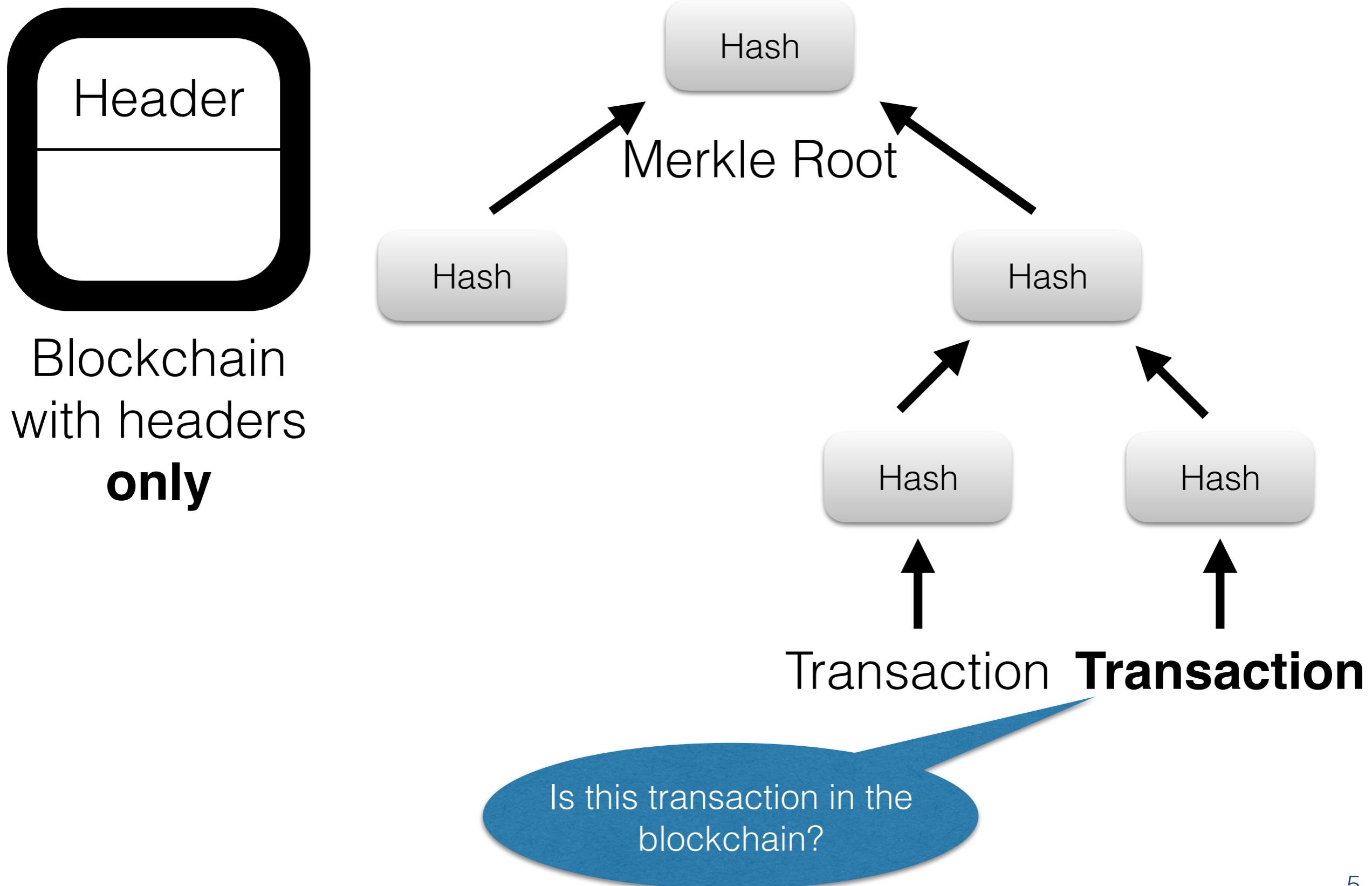
Blockchain  
with headers  
**only**



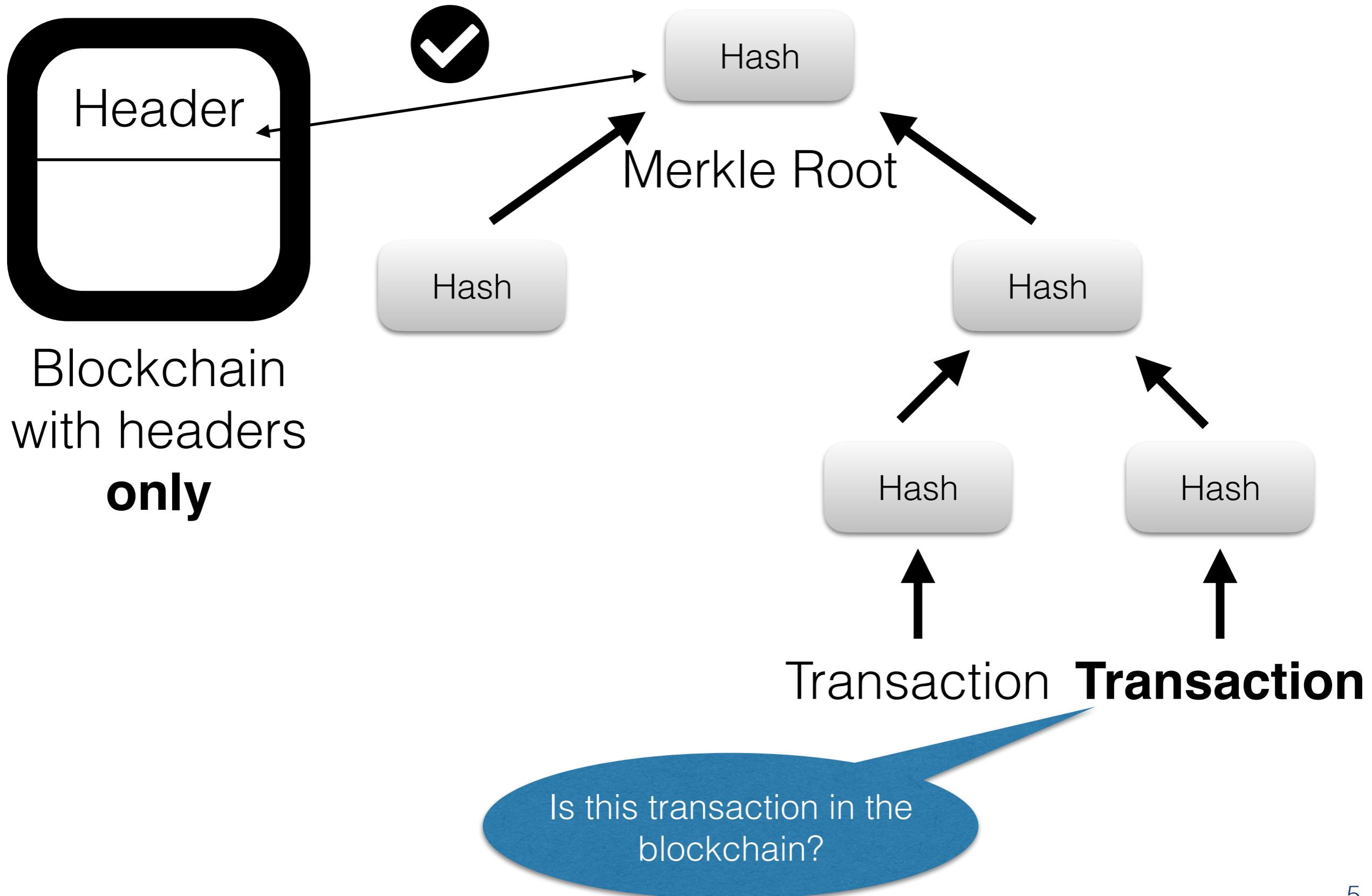
Transaction **Transaction**

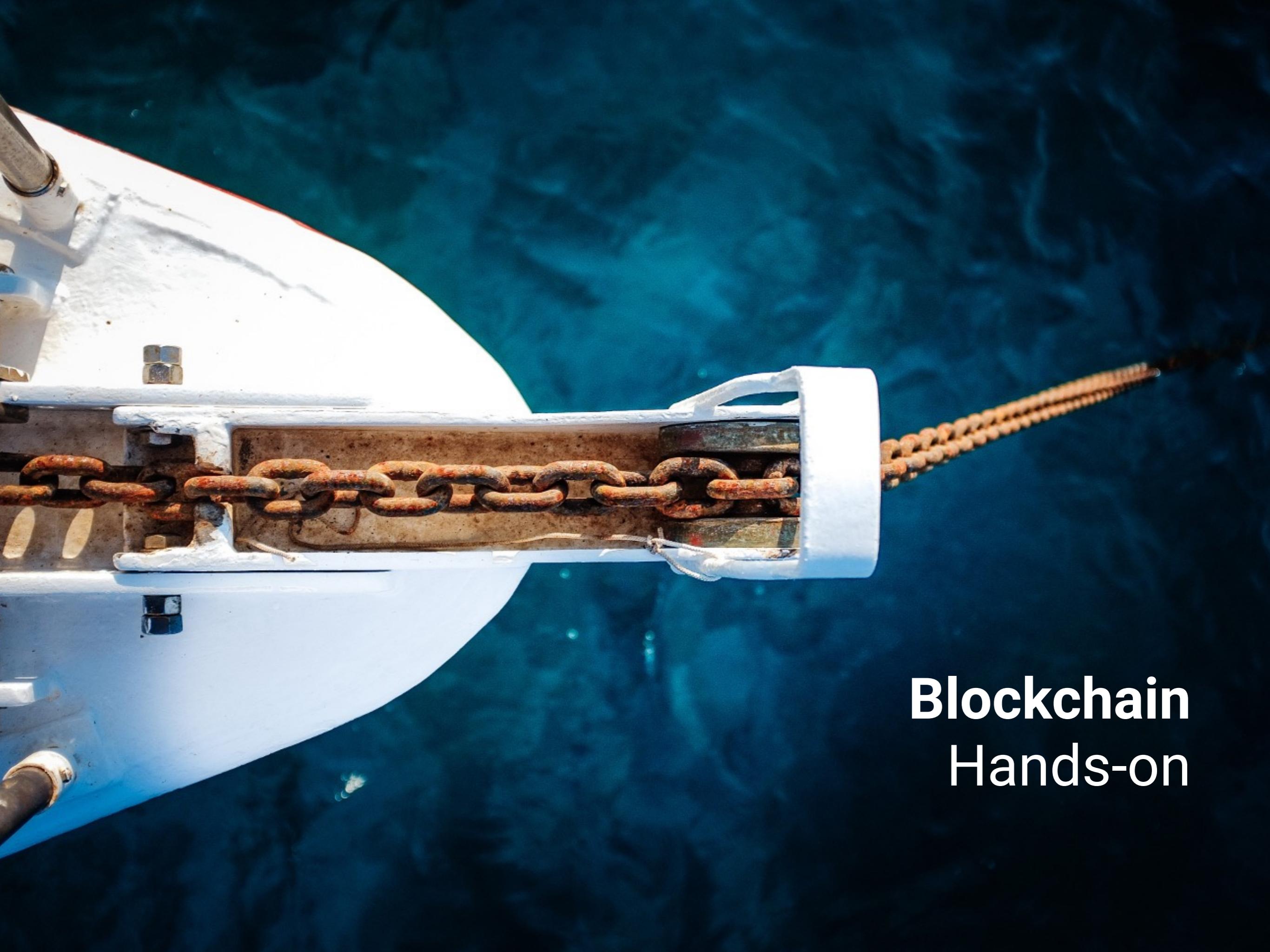
Is this transaction in the  
blockchain?

## SPV Transaction verification



## SPV Transaction verification



A close-up photograph of a white boat's hull and a metal anchor chain against a dark blue sea. The chain is attached to a metal plate on the hull. The water is slightly choppy.

# Blockchain Hands-on

## For next class..

- Create one wallet on MyEtherWallet.com
- Derive two addresses A and B.
- Get test Ether for the Rinkeby network on A.
- Send 1 Rinkeby Test Ether and your lucky number as data attached from address A to B.

## Checklist

- Generate a keypair
- Send your “address” to someone
- Get someone else’s “address”
- Lookup the history of the Ethereum Address  
0x97d4b02ce33c399ffec618bfd2d5bf7108e556ac
  - > Try [etherscan.io](https://etherscan.io)
  - > Try [debank.com](https://debank.com)
  - > Try [zapper.fi](https://zapper.fi)
- How much USD value does this address hold?

## **What can you do with Crypto?**

- Trade with other people
- Exchange to other crypto/FIAT
- Buy something online
- Donate
- Build applications/games

... and many more things!