Blockchain Course

Arthur Gervais

Introduction

2040 BC

Balance sheet - Mesopotamia

Balance sheet

Mesopotamia

2040 BC

HSBC

BNP PARIBAS

1848

AMERICAN EXPRESS

Bank of America

Turing

1936

VISA

1958

mastercard

Centralized Banking

ECash

Chaum

1983

Privacy-Preserving Banking

Ethereum

Bitcoin

2008

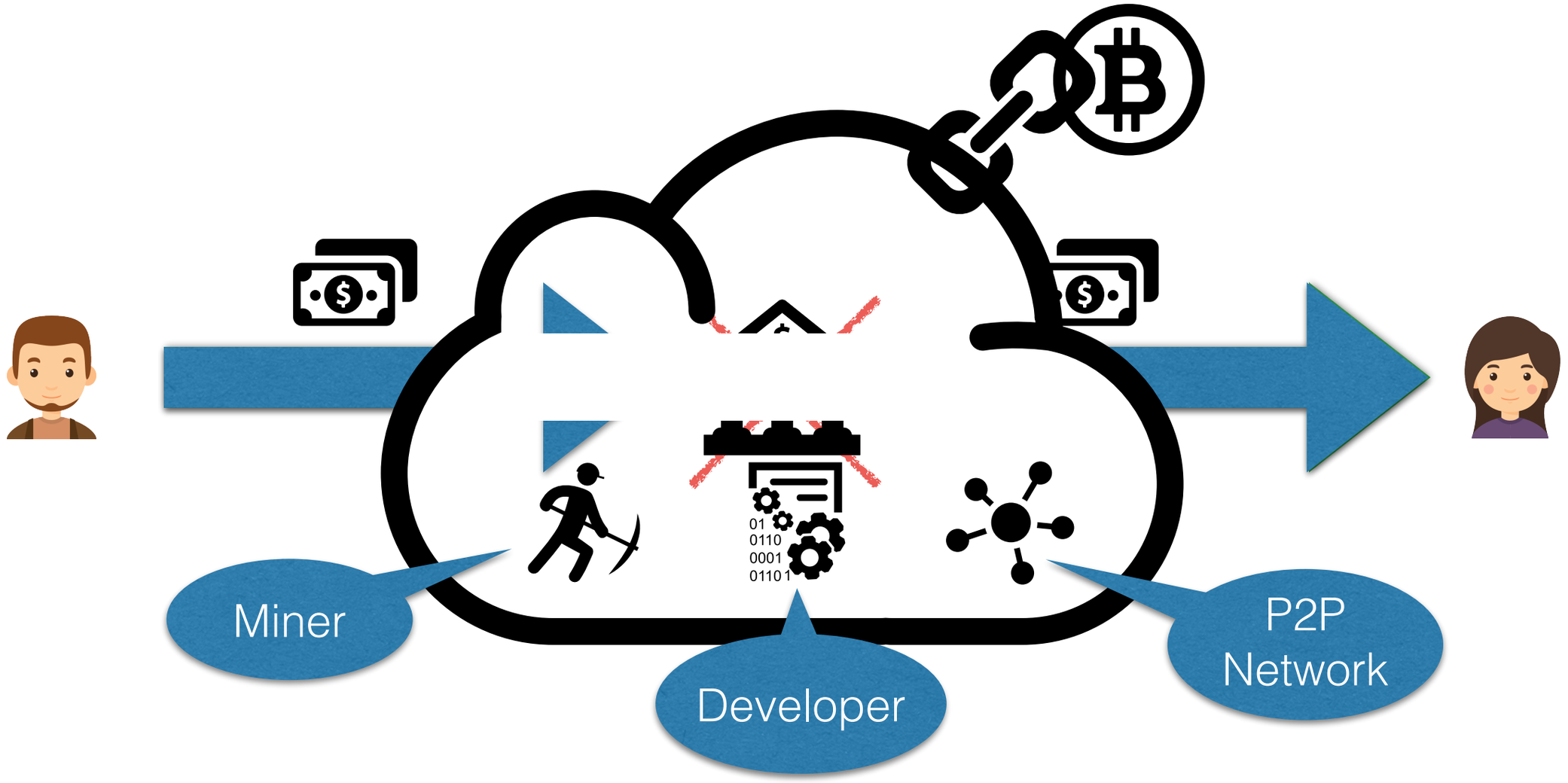
2015

CASH

2016

Decentralized Banking

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

- Introduction
- Applications
- Security
- Privacy
- Scaling

Introduction & Background

Mining & Forks

Smart Contract Programming

Consensus Protocols & P2P

Security

Scaling

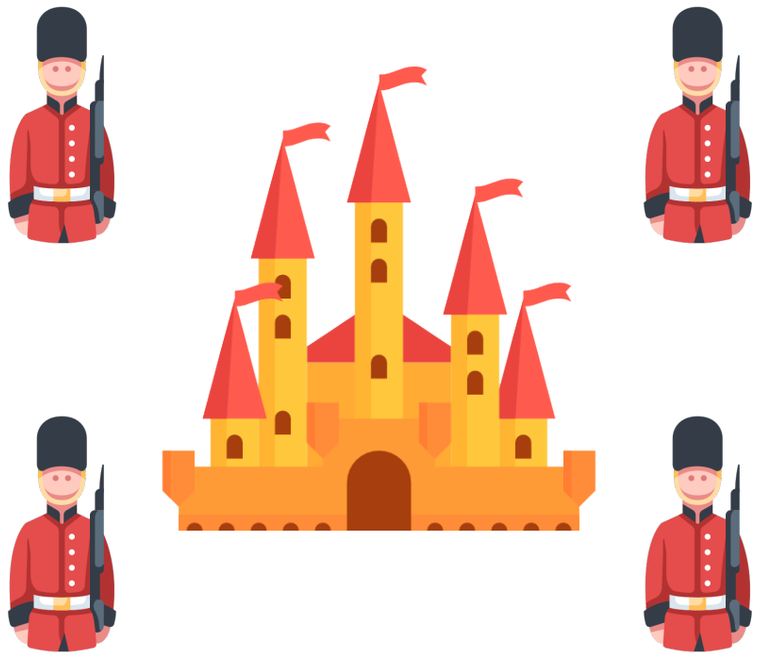
Privacy

Blockchain Background
Byzantine Generals' Problem

Byzantine Generals' Problem

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

Attack?



Attack?

Retreat?

Surrender?

Byzantine Generals' Problem

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?

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

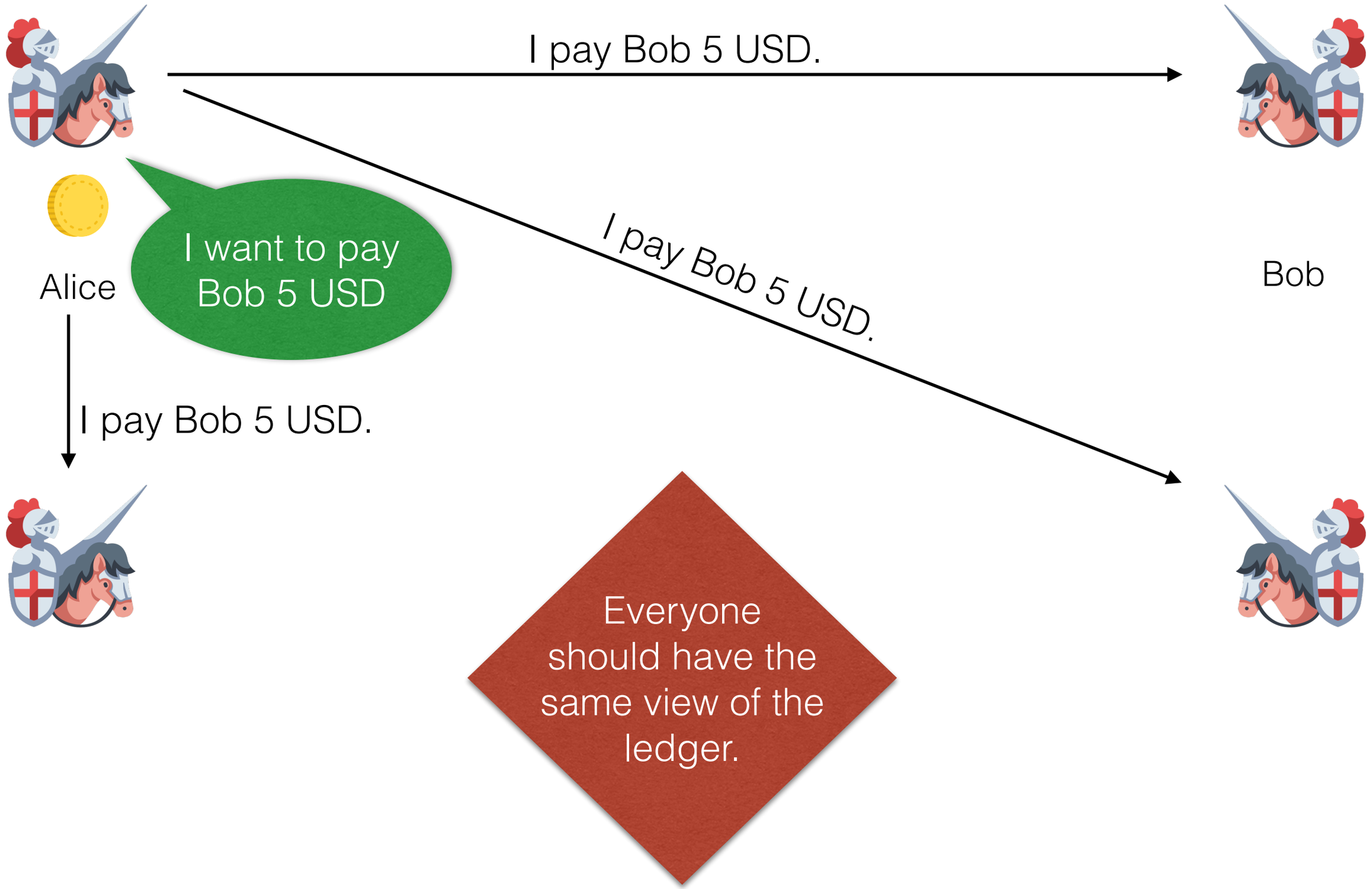
The coin belongs to Alice.

The coin belongs to Alice.

The coin belongs to Alice.

The coin belongs to Bob.

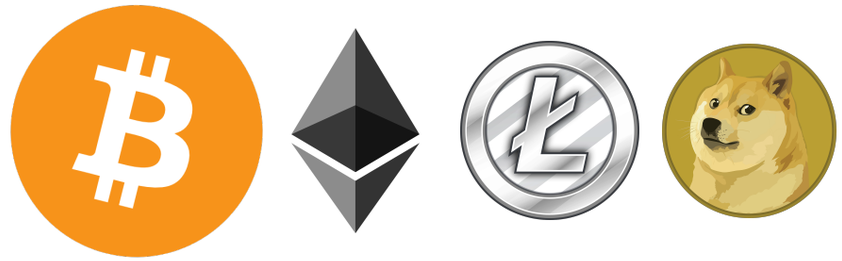
(Decentralised) Digital Payment Systems



Blockchain != Blockchain

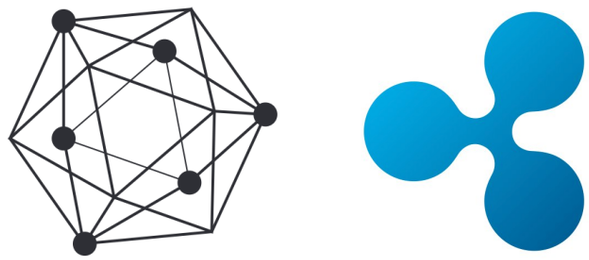
Open and Decentralised Blockchains

- Ethereum
- Bitcoin



Permission-based Blockchains

- Hyperledger
- Ripple
- Stellar



Bitcoin

- First introc
  - Pseudor
- Peer-to-pe
- No trusted
- Blockchai
  - Transac
  - Blocks

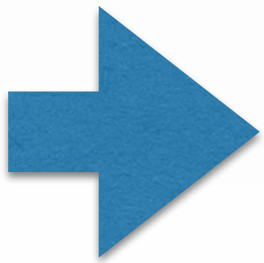
Blockchain

Simple Cryptography

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 || 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 || x) = h in time significantly less than 2^n.

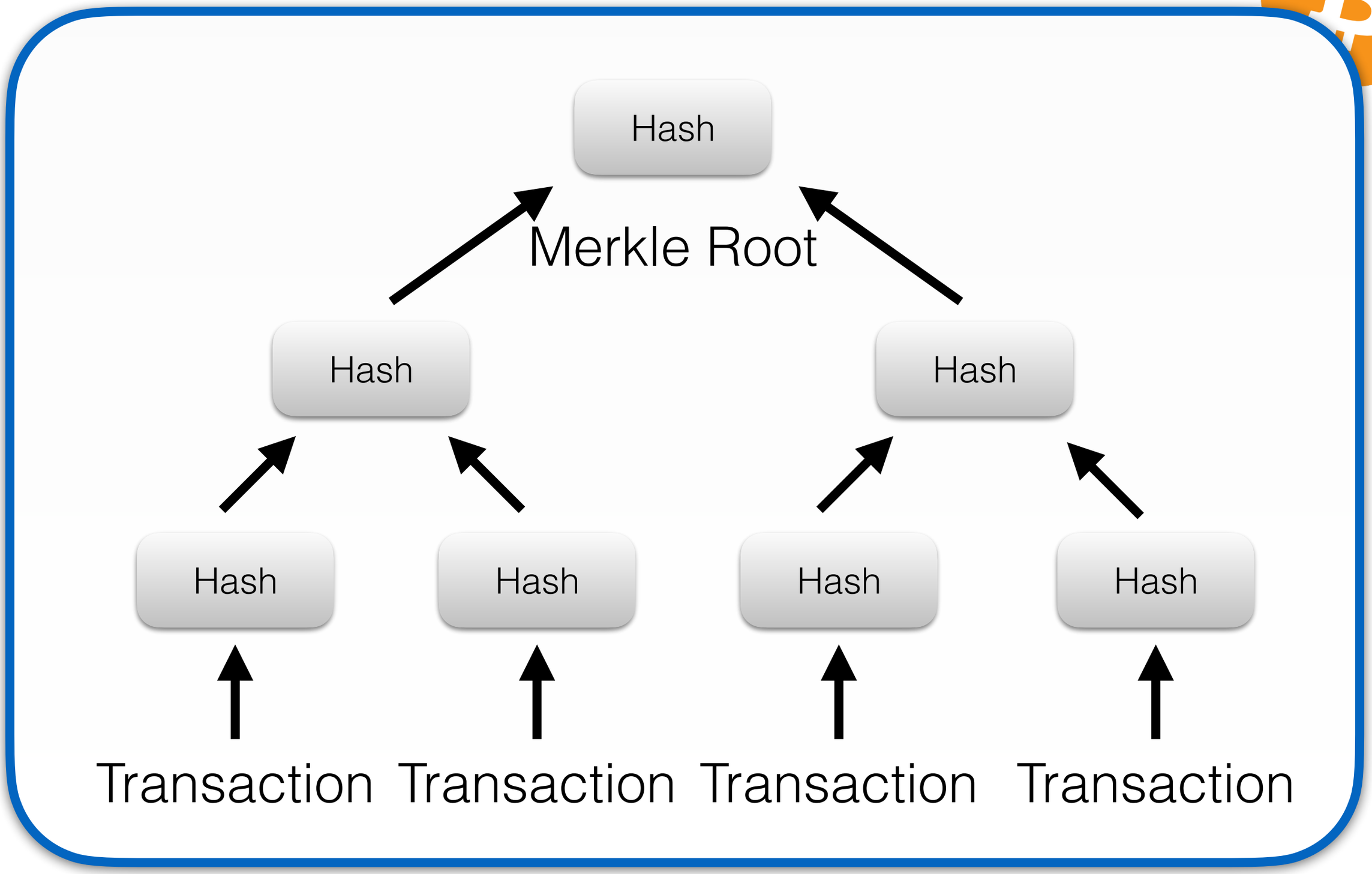
Search puzzle

- 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


Hash
Merkle Root
Hash
Hash
Hash
Hash
Hash
Hash
Transaction
Transaction
Transaction
Transaction

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)**

Bitcoin Addresses

Addresses

- Unique ic
- Hash of a
- 0 < Balar
- 1 Satoshi

1EGa

3J98

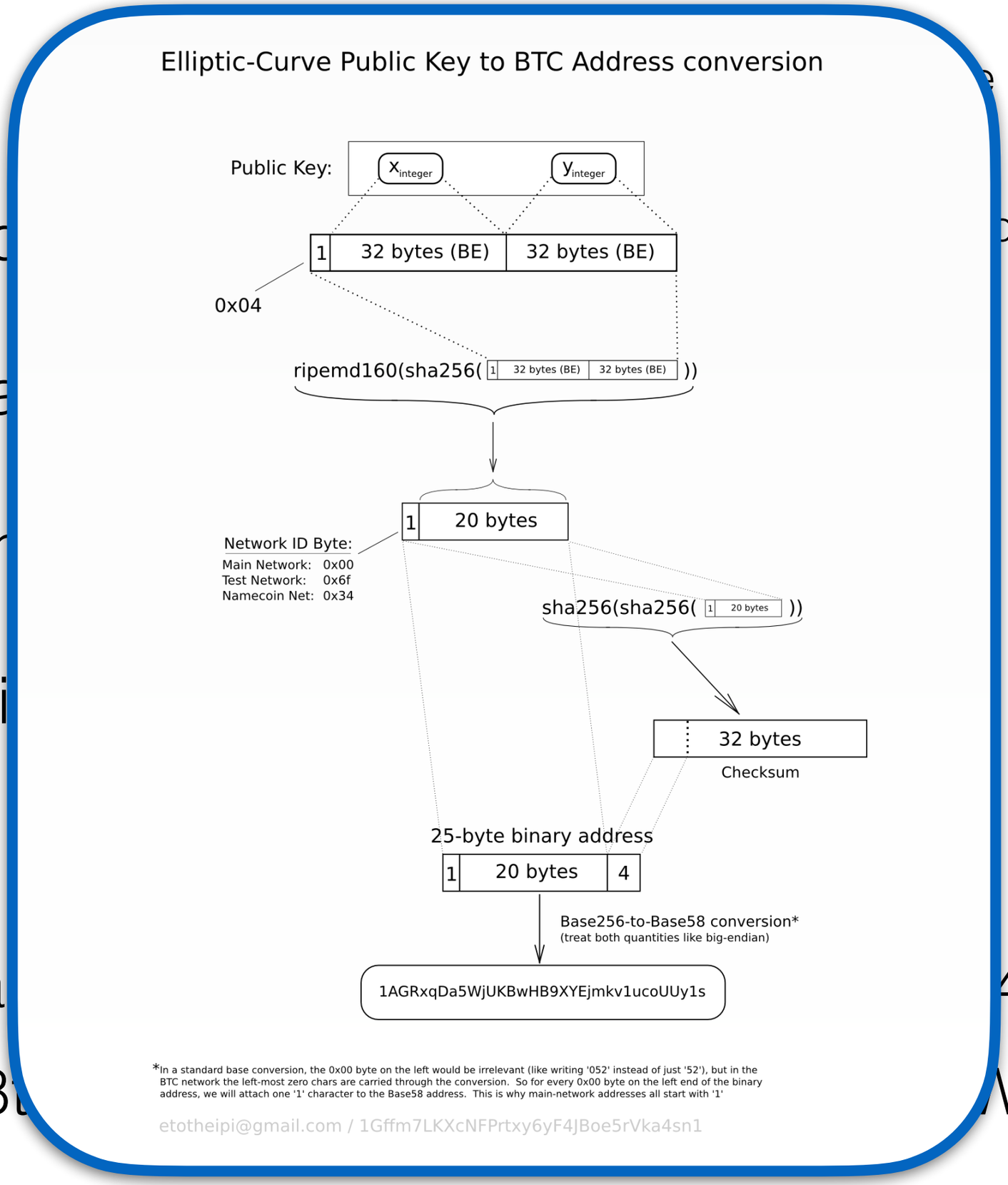
ob

@Eve

@ETH

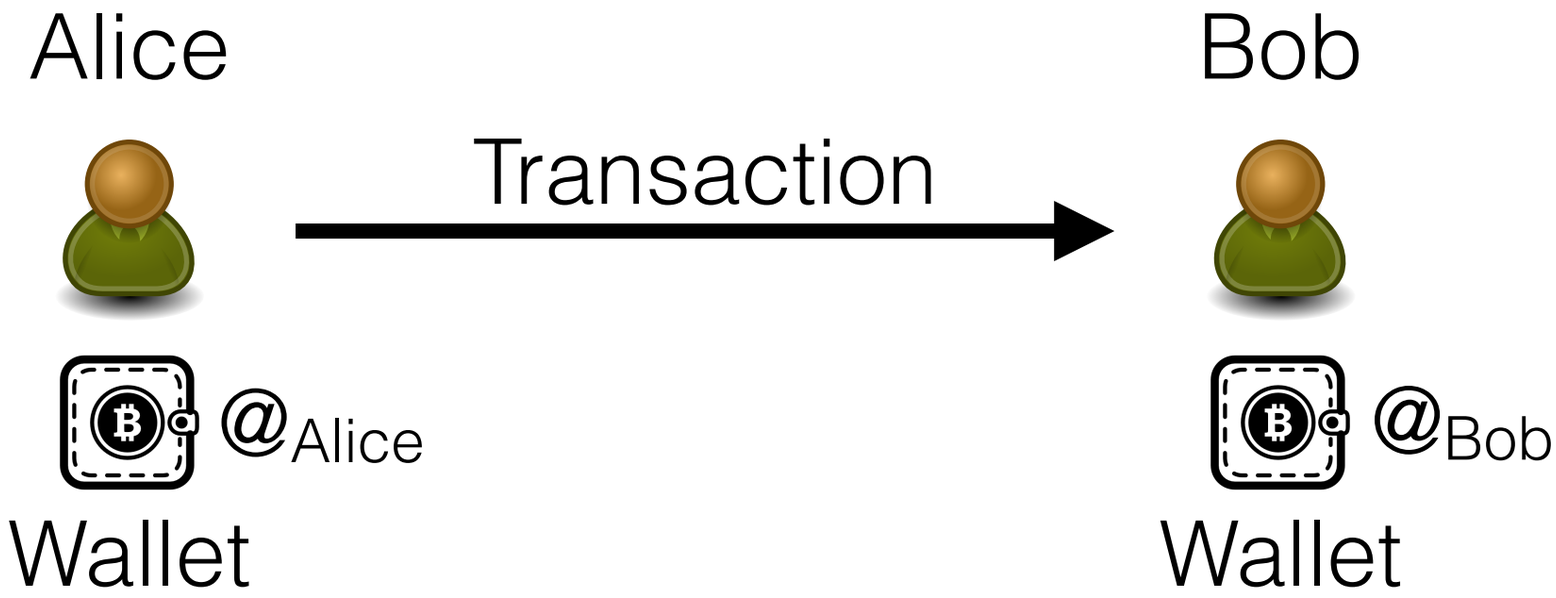
4YTEr

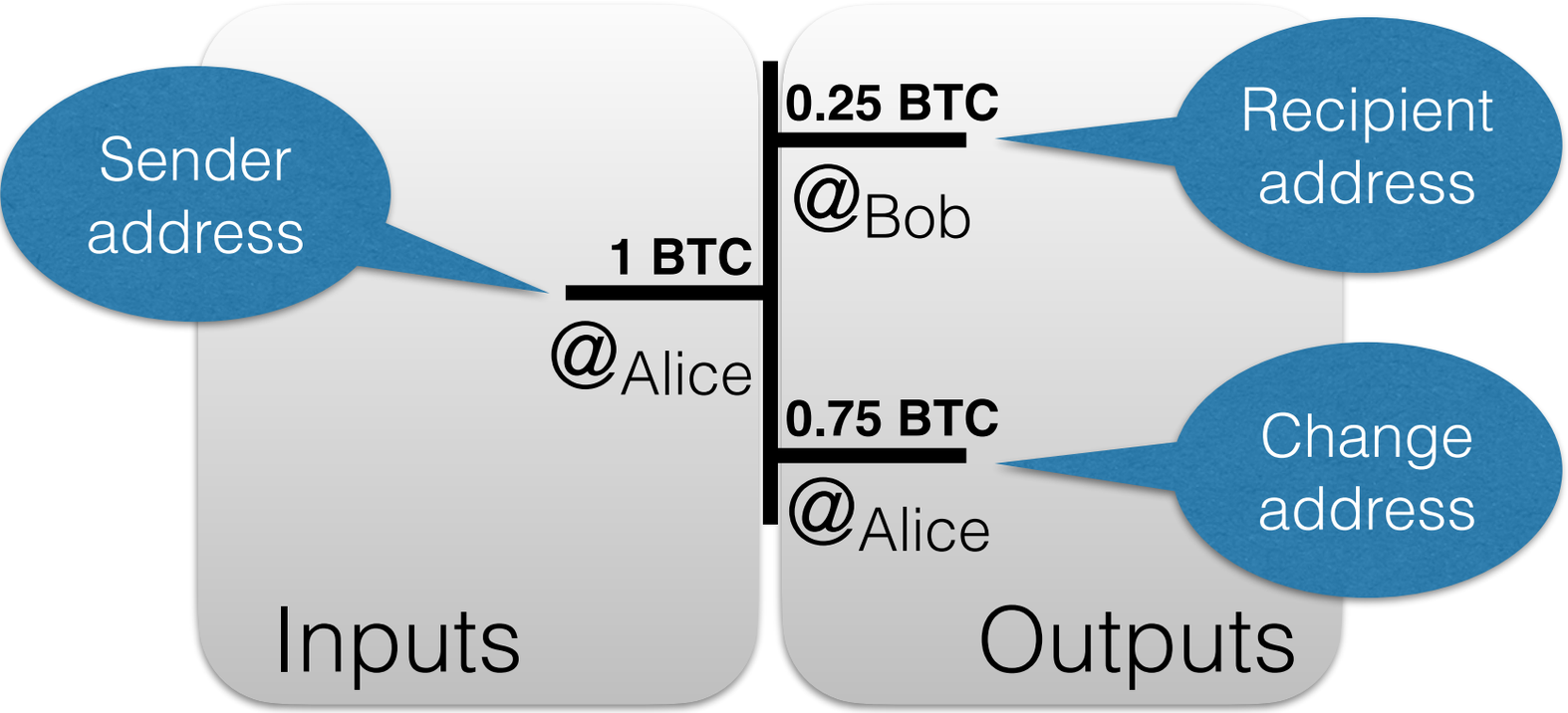
WNLy



Bitcoin Transactions

Transactions in Bitcoin



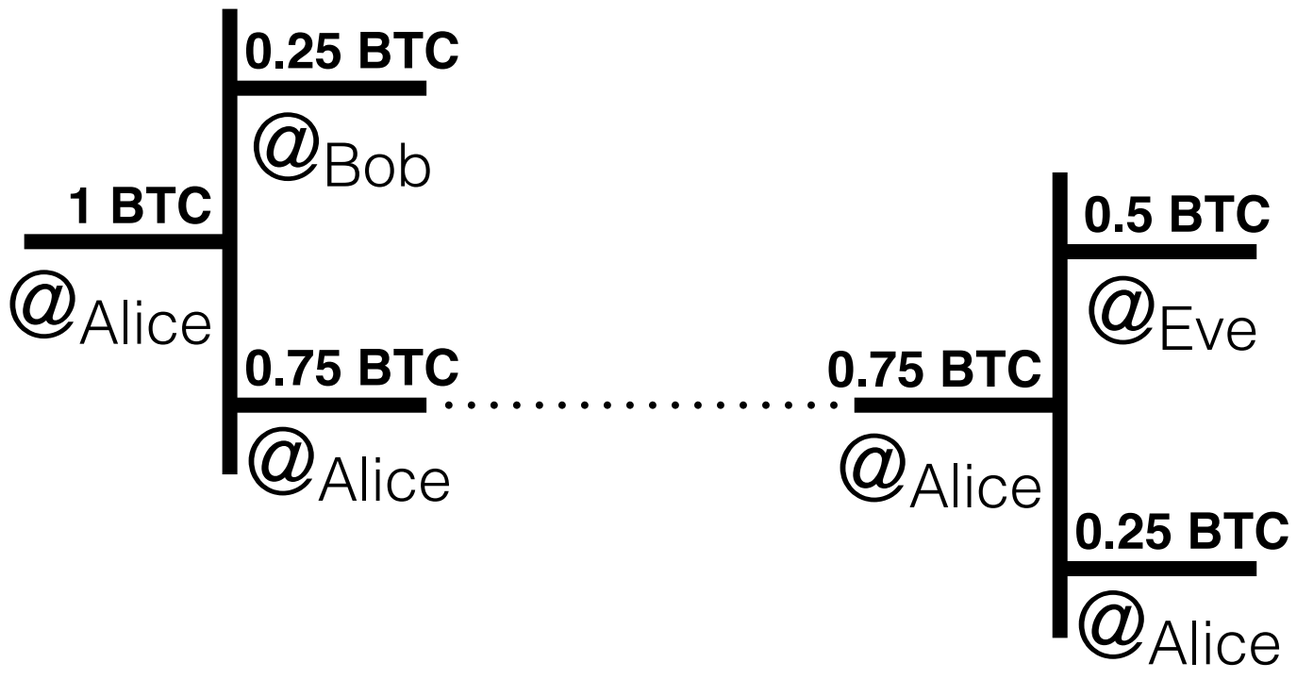


Transaction Fees

$$\sum inputs \geq \sum outputs$$

Difference are fees

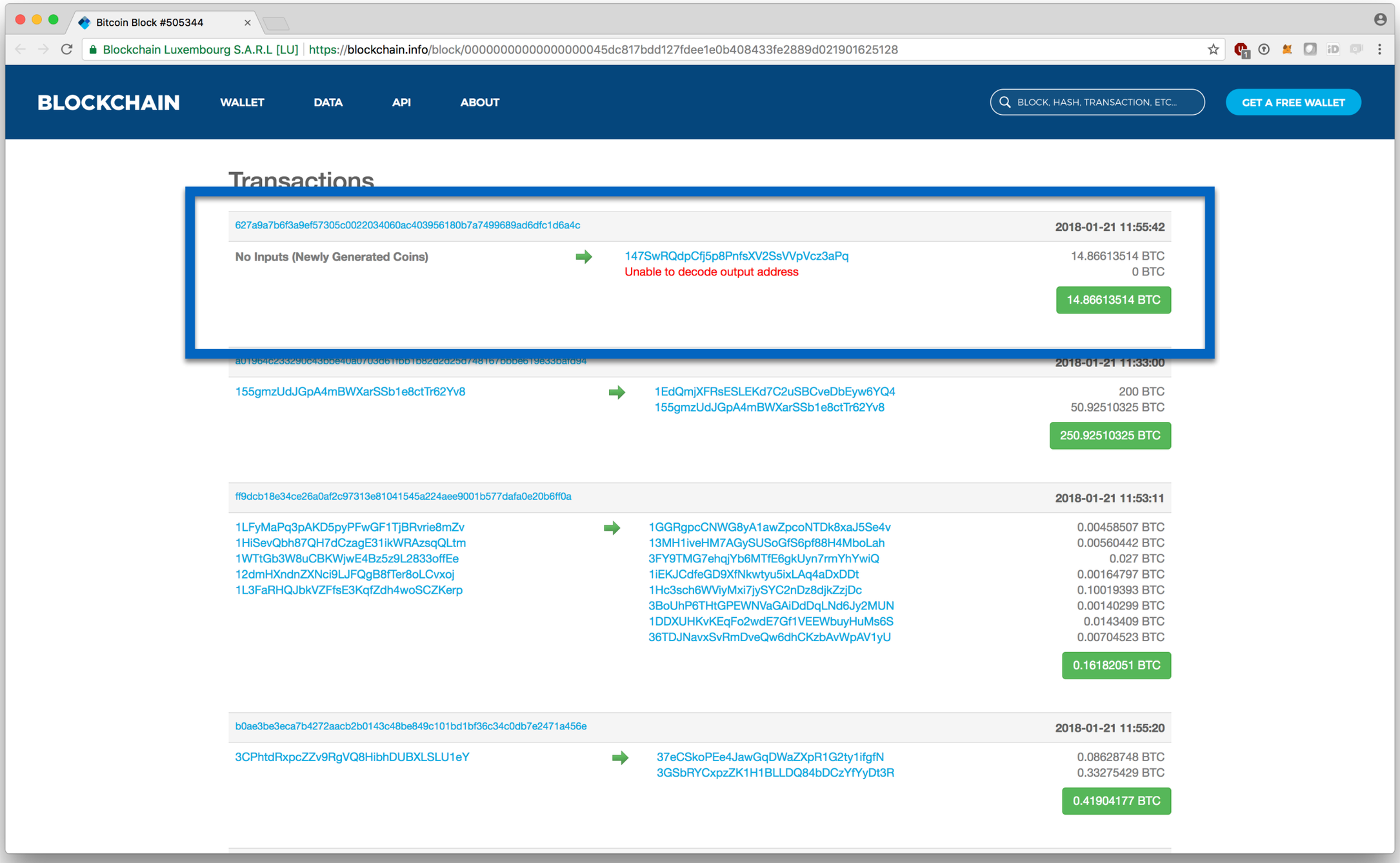
Transactions in Bitcoin



Transaction 1 Transaction 2

Coinbase Transaction

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



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

Example Script

<signature><publicKey> OP_CHECKSIG

Constants are pushed onto the stack

Operation executes on stack values

<PubKey> <Sig>	OP_DUP
Execution Stack	Execution Code

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

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

<PubKey> <Sig>	OP_CHECKSIG
Execution Stack	Execution Code

True

Constants are pushed onto the stack

<Sig> <PubKey> OP_DUP OP_HASH160 <PubKeyHash> OP_EQUALVERIFY OP_CHECKSIG

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

Blockchain Wallets

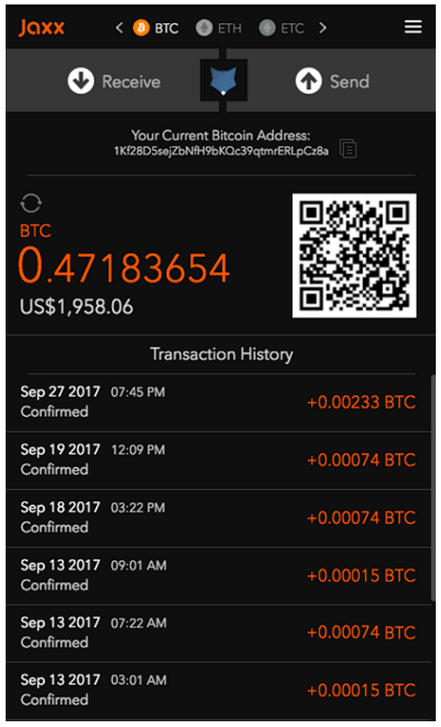
Blockchain Wallets

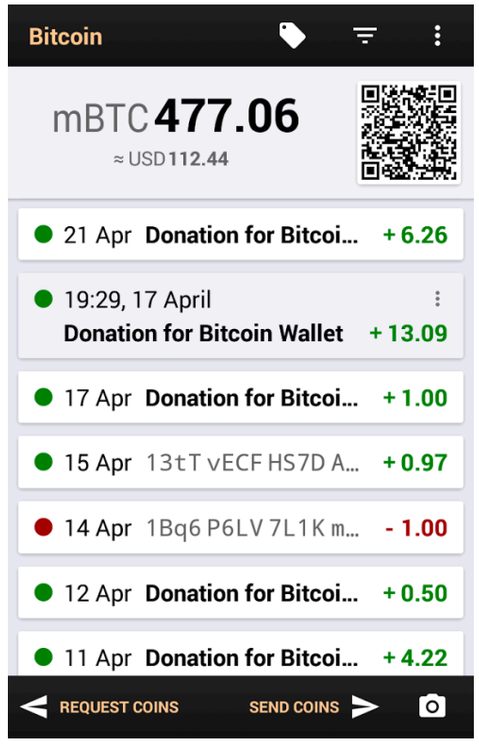
- Bitcoin Core
  Full node, downloads all transactions

- Bitcoin Wallet
  Android

- Jaxx
  Multi-chain Wallet

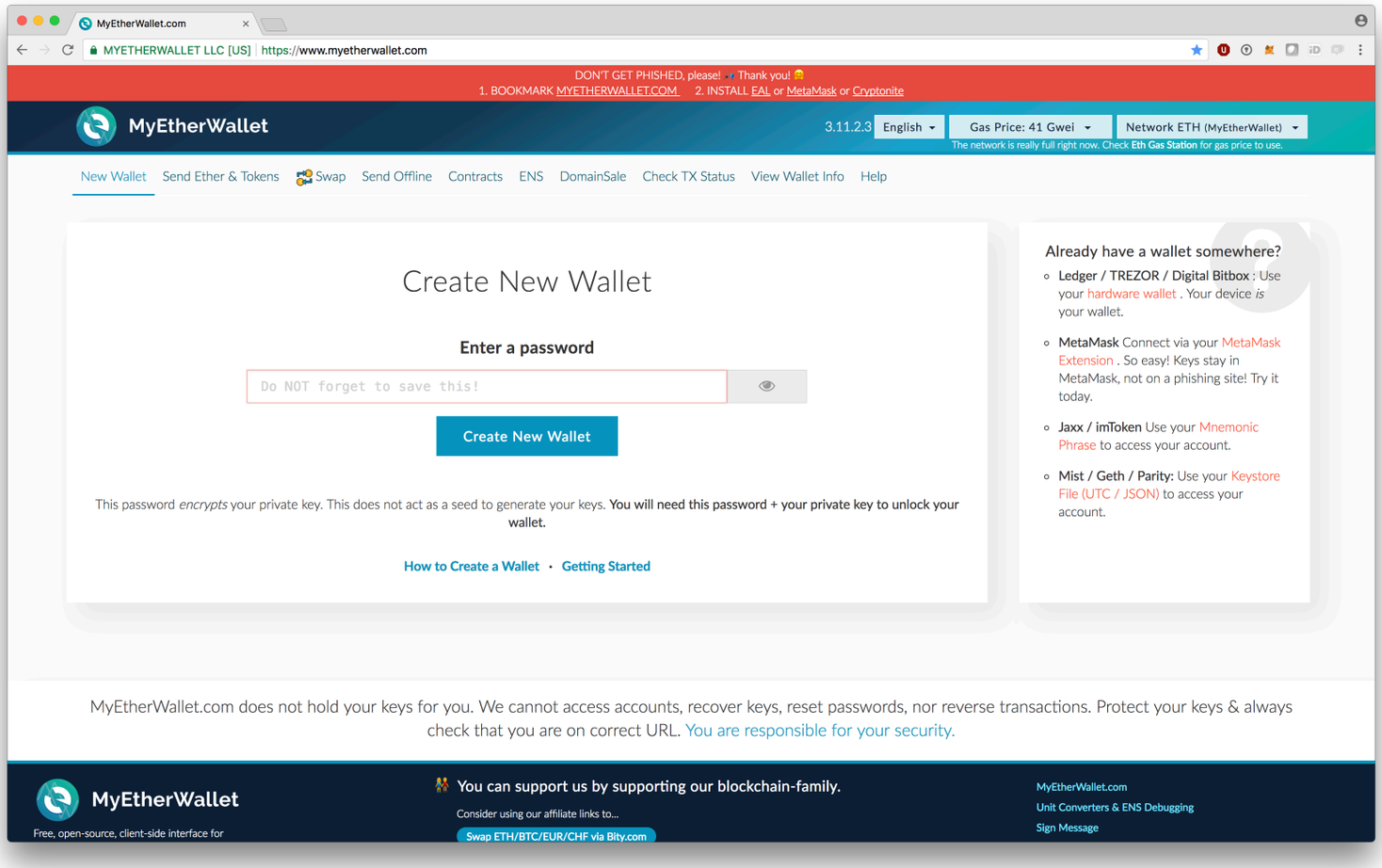
- Ledger Nano S

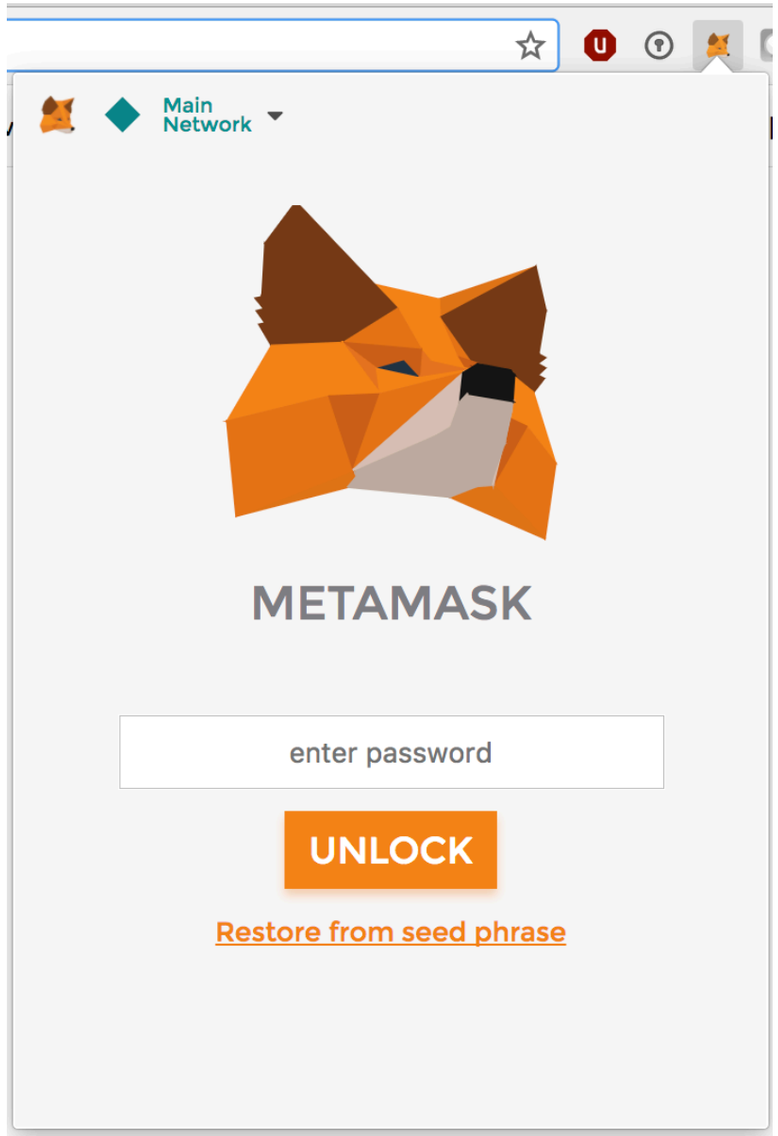




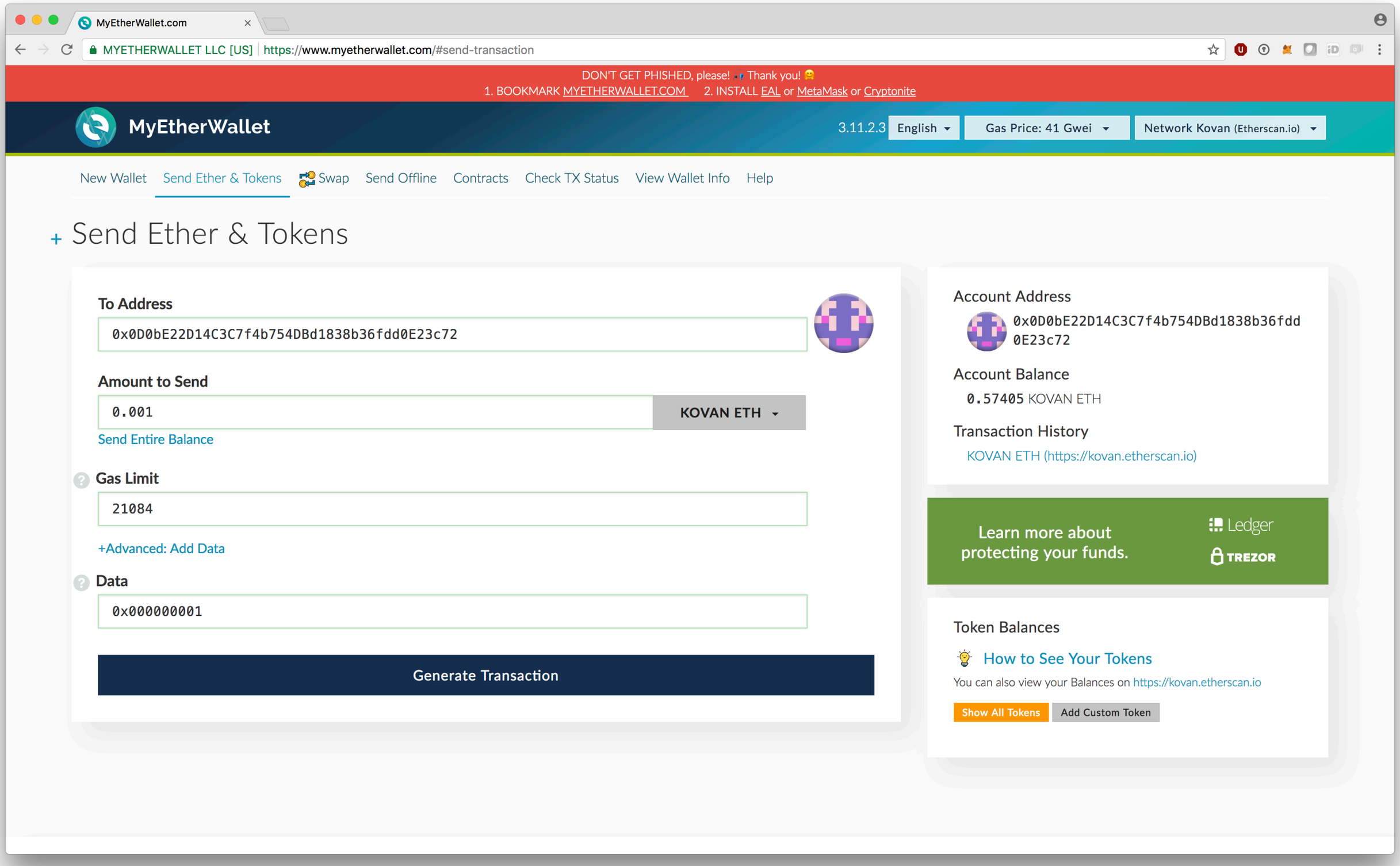
Blockchain Wallets

- MetaMask
- MyEtherWallet



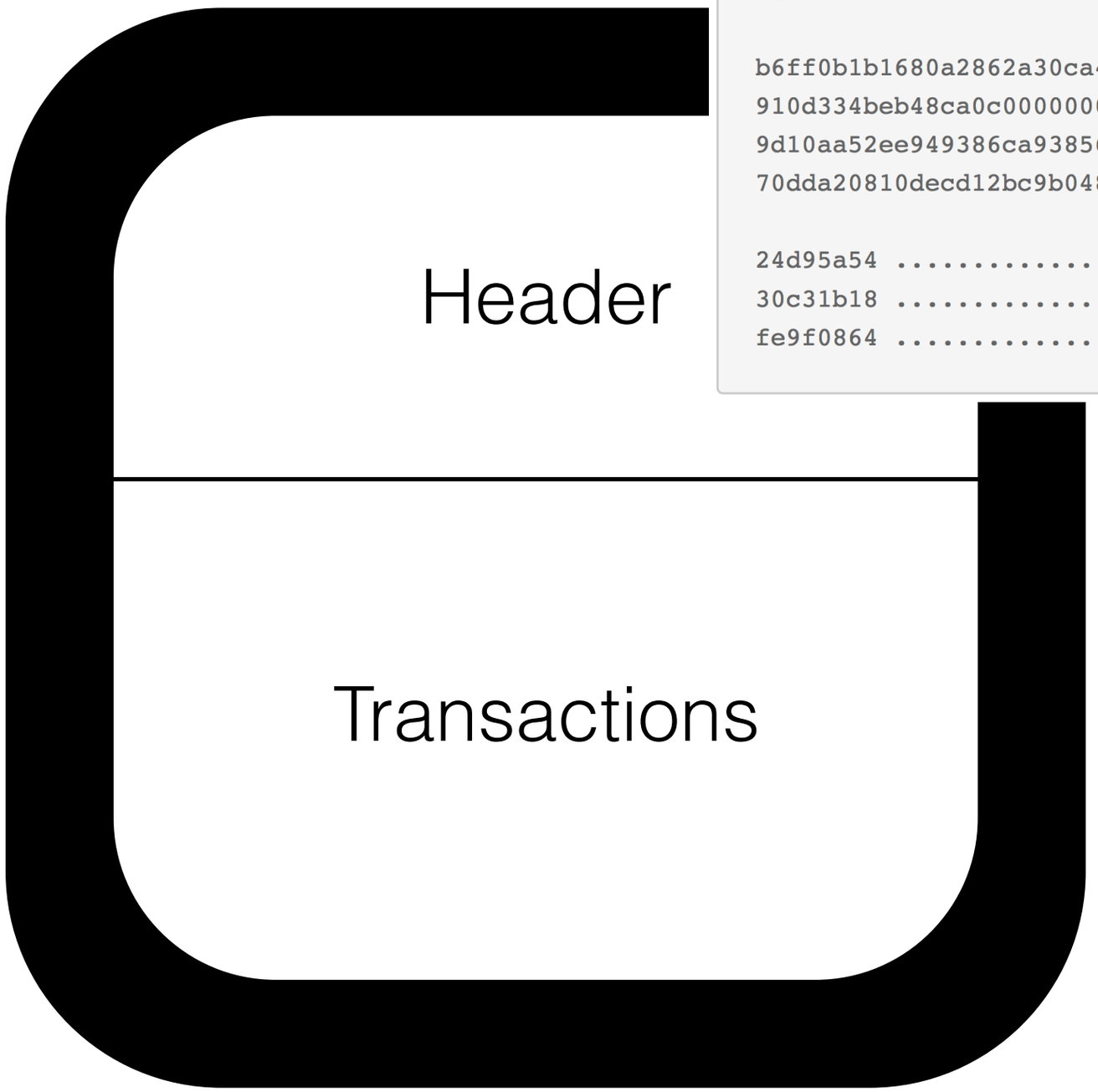


Ethereum Testnet Transaction



Chain of Blocks

Block



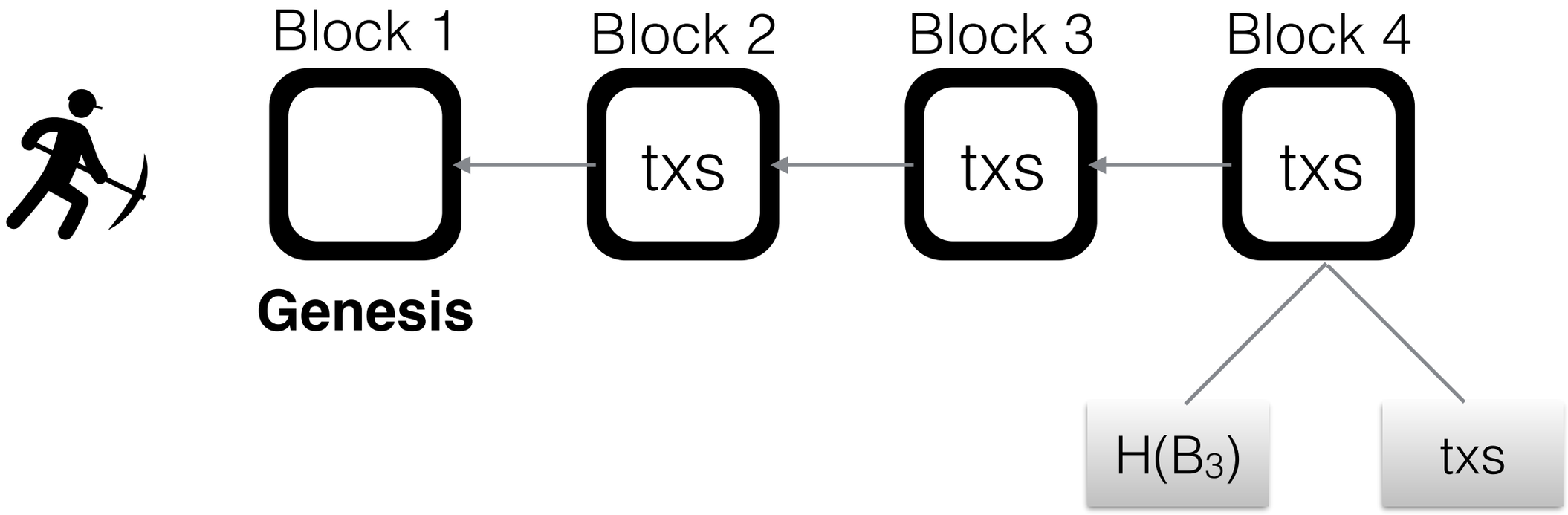

```
02000000 ........................... Block version: 2

b6ff0b1b1680a2862a30ca44d346d9e8
910d334beb48ca0c0000000000000000 ... Hash of previous block's header
9d10aa52ee949386ca9385695f04ede2
70dda20810decd12bc9b048aaab31471 ... Merkle root

24d95a54 ........................... Unix time: 1415239972
30c31b18 ........................... Target: 0x1bc330 * 256**(0x18-3)
fe9f0864 ........................... Nonce
```

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

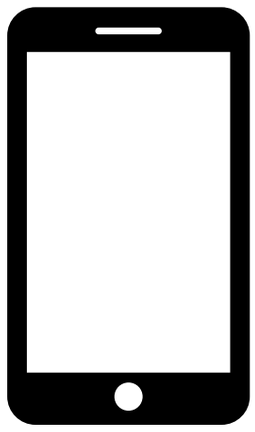
Blockchain



Bitcoin
Lightweight Clients

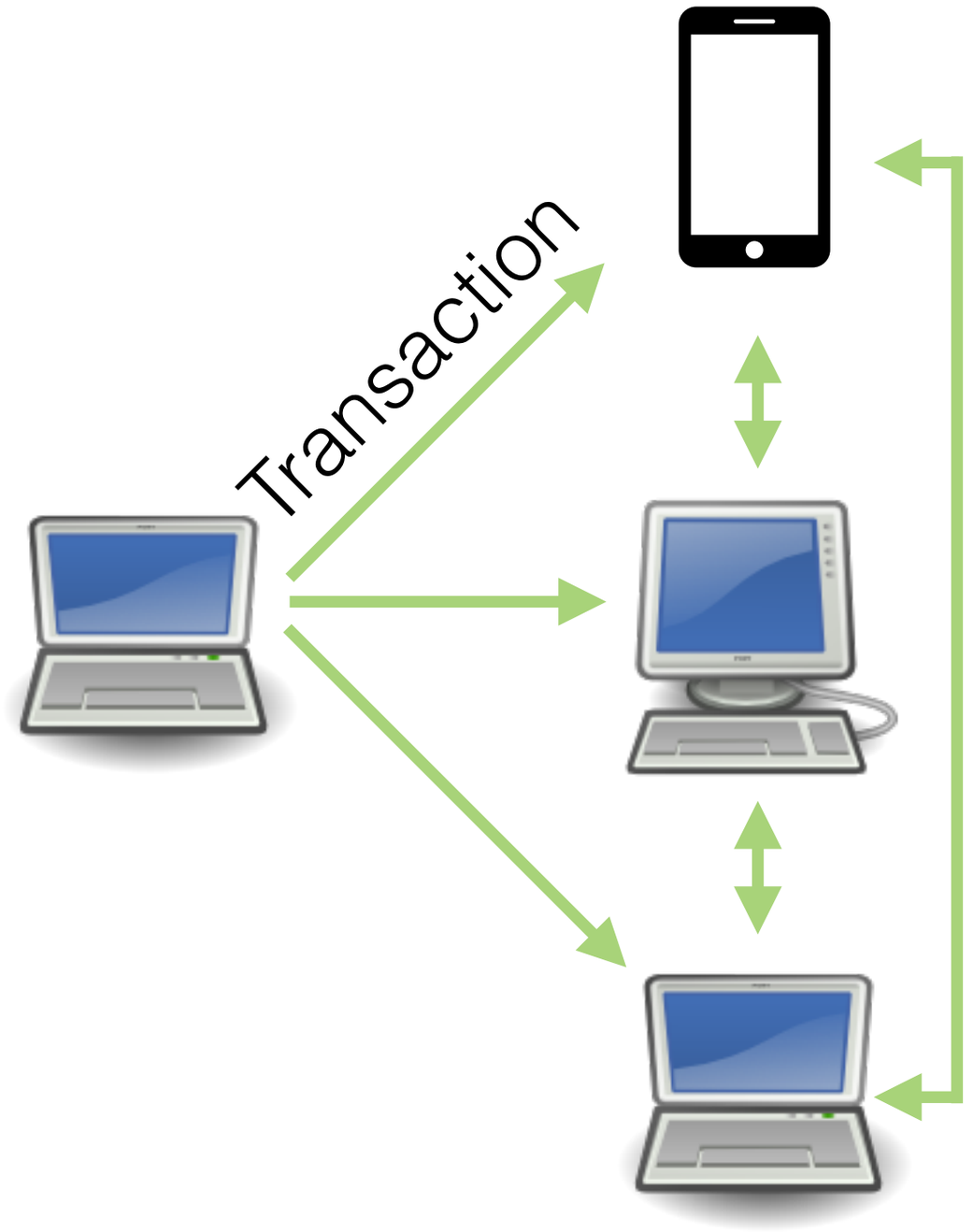
Why Lightweight Clients?

- 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

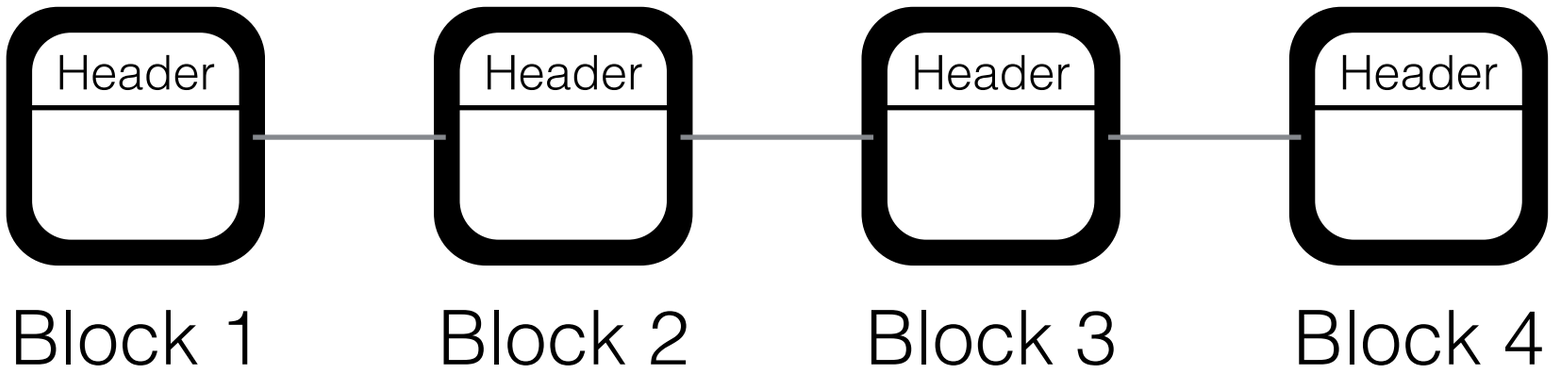


Scaling Problems for Lightweight Clients

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



SPV Transaction verification

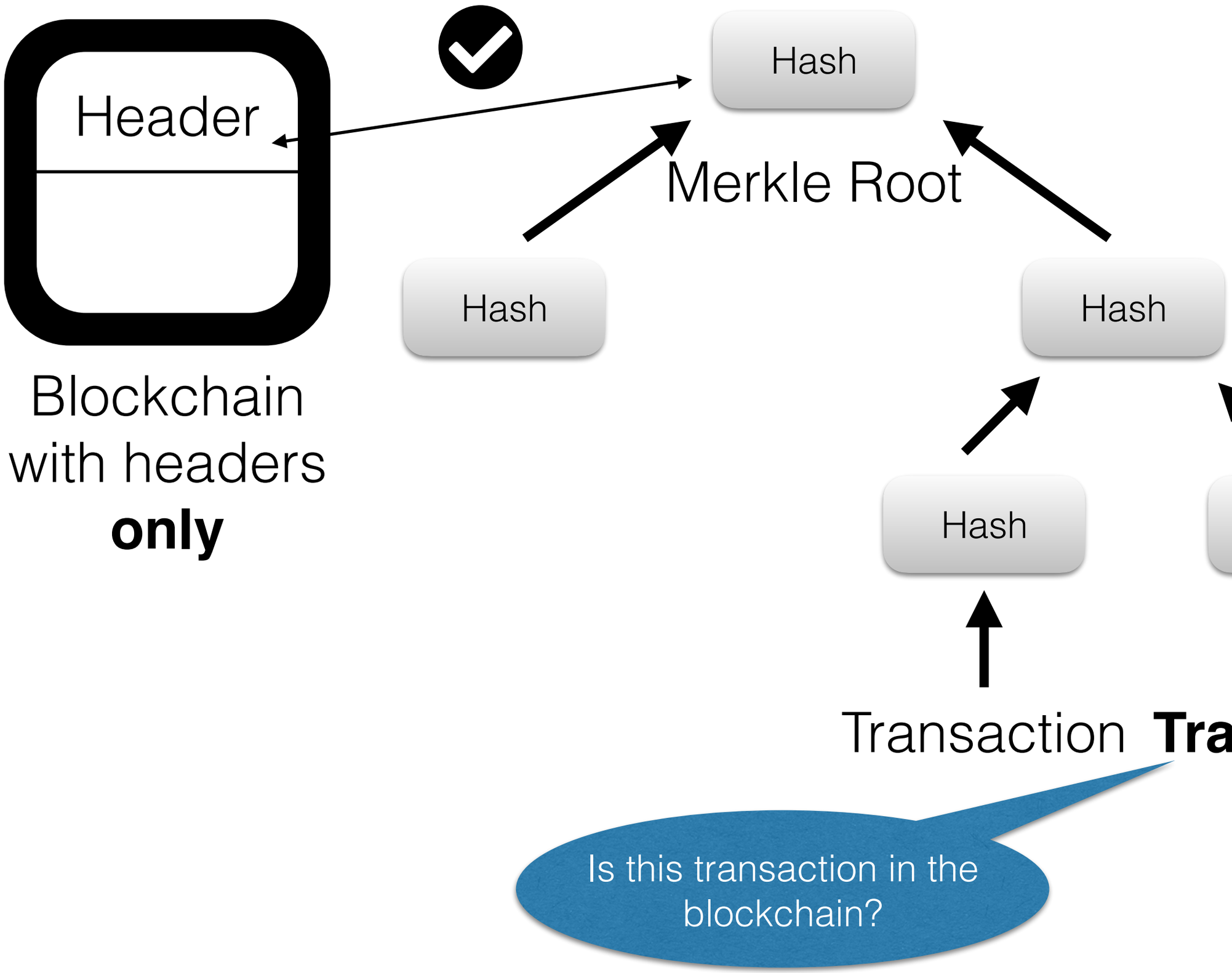


Blockchain with headers **only**

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

SPV Transaction verification

Blockchain with headers **only**



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
  -> Try debank.com
  -> Try 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!