Distributed Systems (DSy)

Ethereum Introduction, Smart Contracts
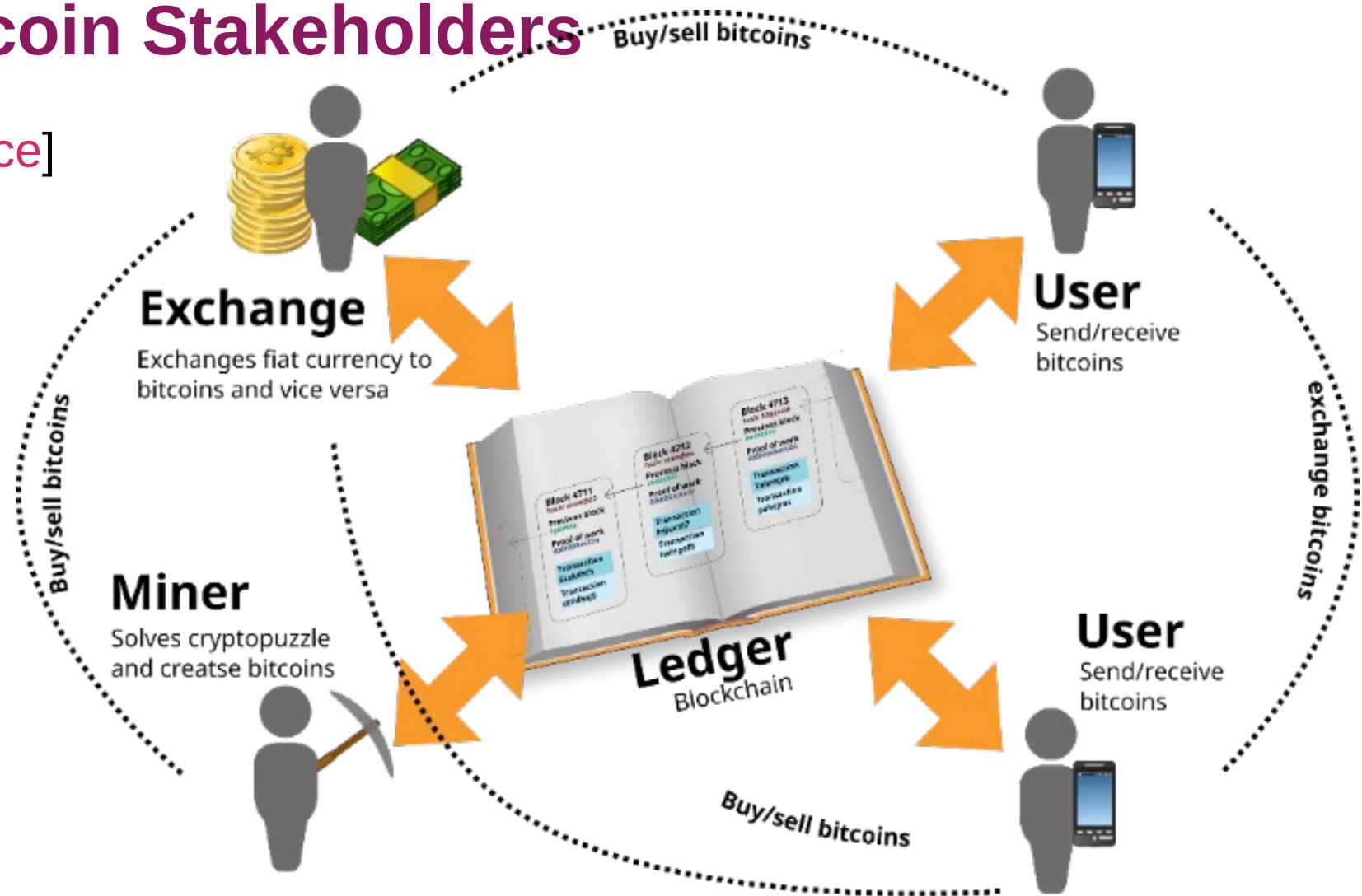
Thomas Bocek

01.06.2022

Learning Goals

- Lecture 13

  - 51% Attacks

  - Ethereum basic concepts

    – Gas

    – Smart contracts

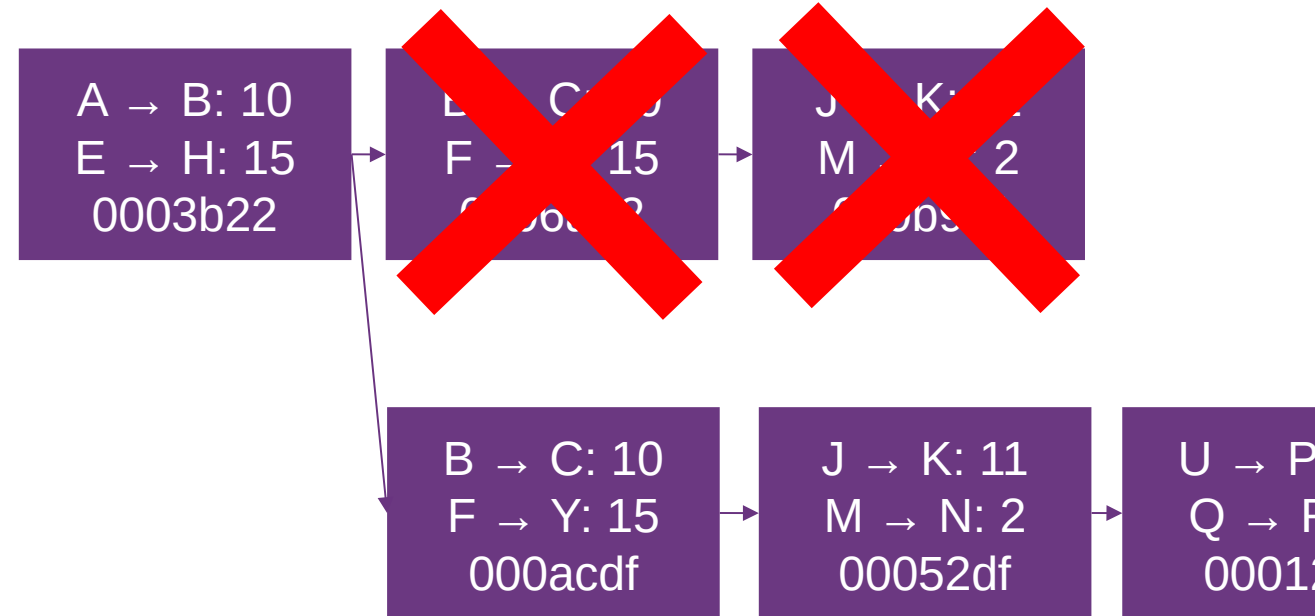   – Account / UTXO

    – Web3

OST

Summary: Bitcoin Stakeholders

- Building blocks [source]

OST

51% Attack

- "If a majority of CPU power is controlled by honest nodes, the honest chain will grow the fastest and outpace any competing chains."

  - https://bitcoin.org/bitcoin.pdf

- PoW: majority of hashing power, PoS: majority of coins

- How expensive is a 51% attack?

  - Buy an attack?

- Double spend, or rollback transactions

  - X is an exchange

  - Mine secretly, Y is your address

  - X arrived – payout (1 block conf.)

  - You mine faster, broadcast secret chain

  - Tx F→X: 15 never happened, goes to Y

A → B: 10
E → H: 15
0003b22

B → C: 10
F → Y: 15
000acdf

J → K: 11
M → N: 2
00052df

U → P
Q → F
0001

OST

51% Attack

- Control over 50% of the scarce resources
  - Pools: cooperative puzzle solving
  - Solo: competitive puzzle solving

http://blockchain.info/pools
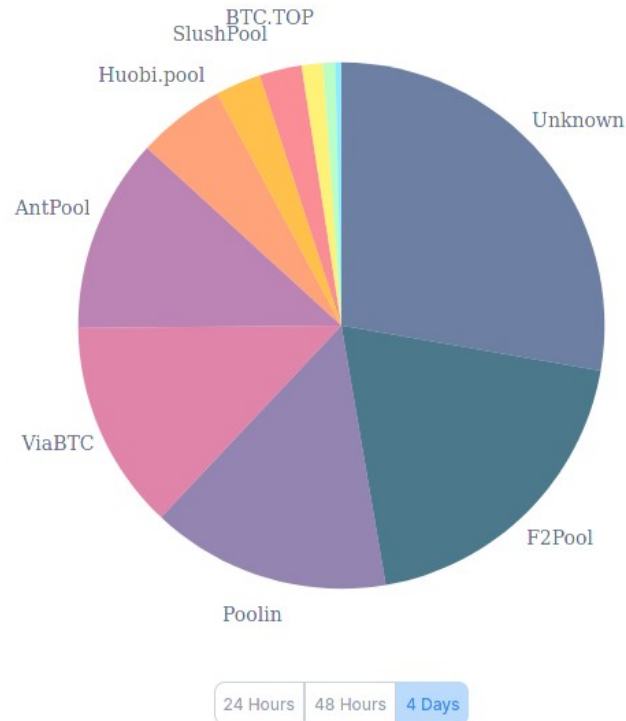
- 07.08.2021: Bitcoin SV rocked by three 51% attacks in as many months [link]
- 30.08.2020: Ethereum Classic suffers another 51% attack [link]
  - "The total value of the double spends that we have observed thus far is 219,500 ETC (~$1.1M)."
- 23.04.2020: DeFi Platform Suffers 51% Attack From Its Top Miners — or Does It? [link]
  - "resulted in $6.7 million worth of the USD-pegged stablecoin pUSD"
- 08.11.2020: Grin network hit with 51% attack while GRIN token remains resilient [link]

OST

Bitcoin / Ethereum

- Bitcoin vs. Ethereum

  - Implementing new features slow

    - Many Bitcoin hardforks (segregated witness vs. increasing block size voting) Cash vs. SV

  - Bitcoin Script limited

    - Lightning network

  - Pros and Cons – no silver bullet

- Ethereum (1 ETH ~1900$)

  - Generalized blockchain (loops, arithemitcs, etc.)

  - White paper released in December 2013

  - Protocols designed from scratch (not like Litecoin, Peercoin)

  - Ethereum foundation located in Zug (initiator known) - non-profit foundation

  - Mining reward ~ block every ~14s – ~2 ETH ("always", unlike Bitcoin)

Vitalik Buterin

Ethereum History

- Olympic (past) – released 09.05.2015
  - Last Ethereum Proof-of-Concept series
  - "Olympic will feature a total prize fund of up to 25,000 ether" (now 70m USD)
- Frontier (past) - released 30.07.2015
  - Main public network, "Beta"/use at your own risk
- Homestead (past) - released 14.03.2016
  - Public network considered "stable", integrate critical protocol changes

- Byzantium - released 16.10.2017
  - Initial supports for ZKP (private smart contracts), reducing mining reward to 3 ETH
- Constantinople – released 27.02.2019
  - Bitshifting, mining reward 2 ETH (delay difficulty bomb)
- Istanbul - released 08.12.2019, gas cost changes, DDoS resilience, Zcash interop
- Muir Glacier- released 01.01.2020 (delay difficulty bomb)
- Berlin – released 104.2021 (gas cost)
- London – released 05.08.2021 (dynamic block size)
- Ethereum 2.0 / Serenity (future)
  - Scalability, proof-of-stake, Sharding, ZKP

Ethereum Stats

- Basic Stats

  - 2nd in market cap ~ 235b USD

  - Daily transactions (highest ~18.2 TPS, 22.4.2021)
    now ~1150k per day

  - Node count (~3'000 nodes?)

  - Blocksize ~108KB

  - Accounts (200mio)

Price Evolution — 2016

Price Evolution — 2017
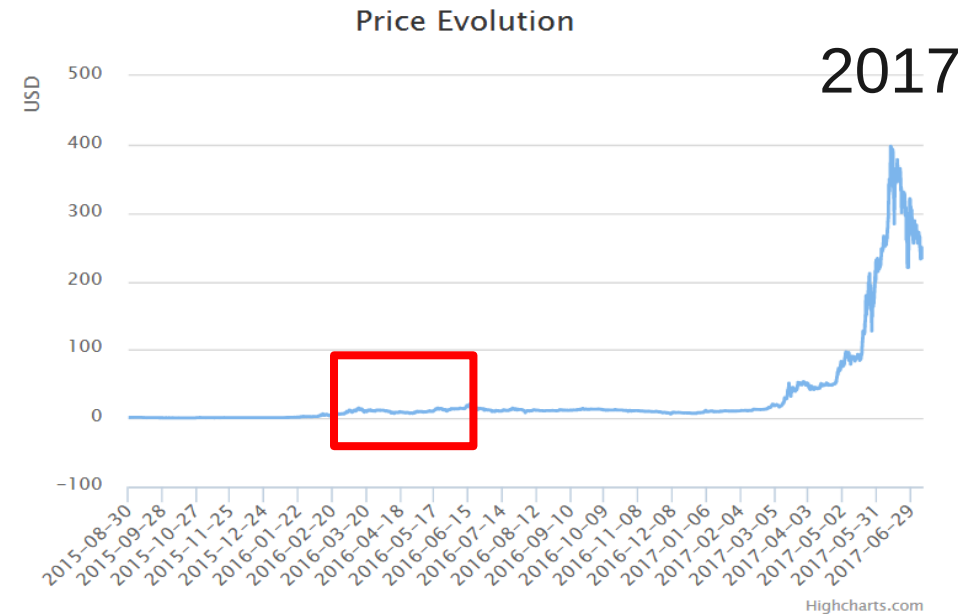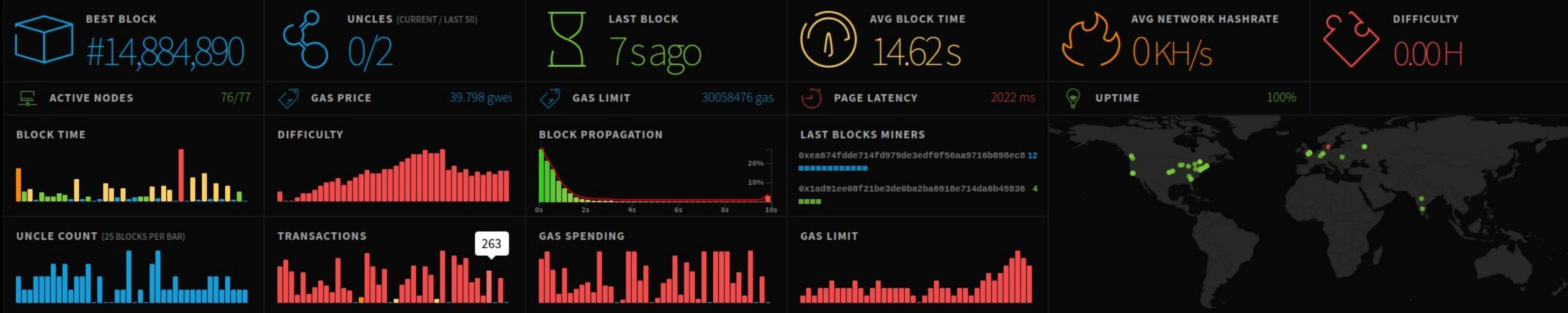
- 115mio ETH supply (stats)

- Increasing difficulty

  - Mining in pools

  - Ethermine + F2Pool ~42%
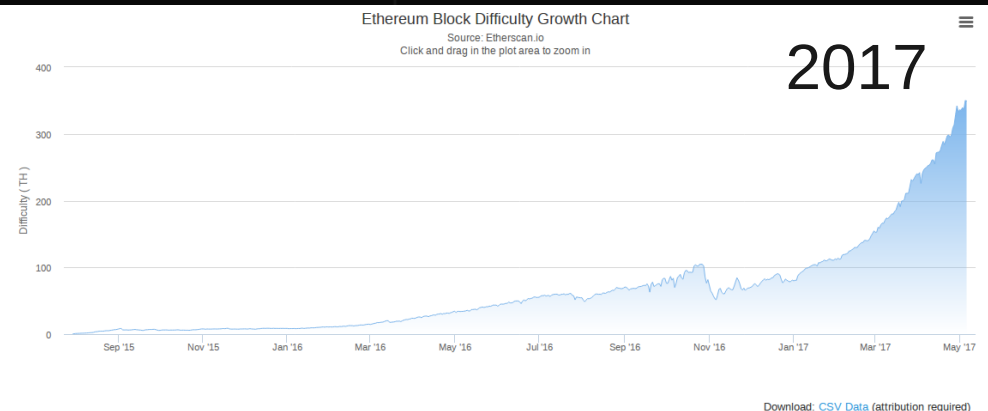
Top 25 Miners by Blocks
In the last 7 days
Source: Etherscan.io

2017

2022

OST
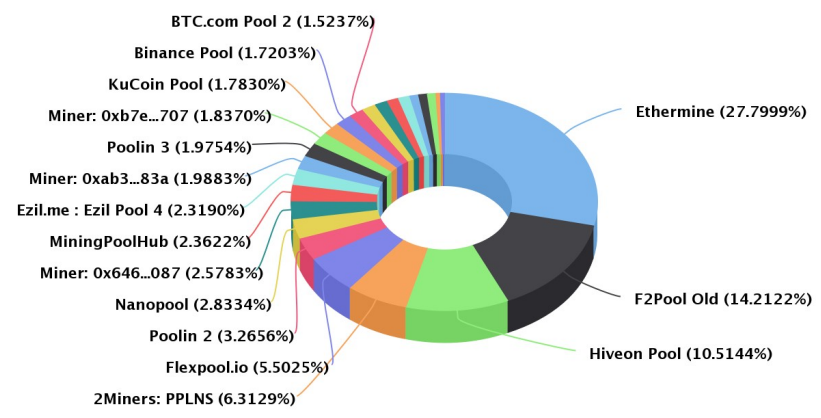
Blocktime and Gas

- Gas Price set by Miner

  - Gas price ~36+2 gwei

- Miner decides which transaction at which gas price to include

  - Market for TX

- Gas price too low, longer waiting time until TX will be included

- Units:

1 ether =	
1000000000000000000	wei
1000000000000000	Kwei
1000000000000	Mwei
1000000000	Gwei
1000000	szabo
1000	finney
1	ether
0.001	Kether
0.000001	Mether
0.000000001	Gether
0.000000000001	Tether

OST

Blocktime and Gas

- Block time: ~14-15s

  - Ice age

- Smart Contracts are turing complete

  - Every instruction needs to be paid for (example)

- Gas price

  - If you run out of gas, state is reverted, ETH gone

source

Ethereum smart contract

- For now, proof of work (PoW)

  - But difficult with ASIC (memory-hard), starts at 1GB RAM for full node and miner (2GB RAM should be enough)

- Computation and storage on EVM is "very expensive": every contract is run on every full Ethereum node

  - Result on every node is the same

  - Global computer, always running, always correct

- Account-based

  - 2 types: externally controlled, contract

    - Both can have and send ether

    - External accounts: controlled by private keys

  - Contract accounts never executed on their own

    - Contract accounts: controlled by code

    - All action fired from externally controlled accounts
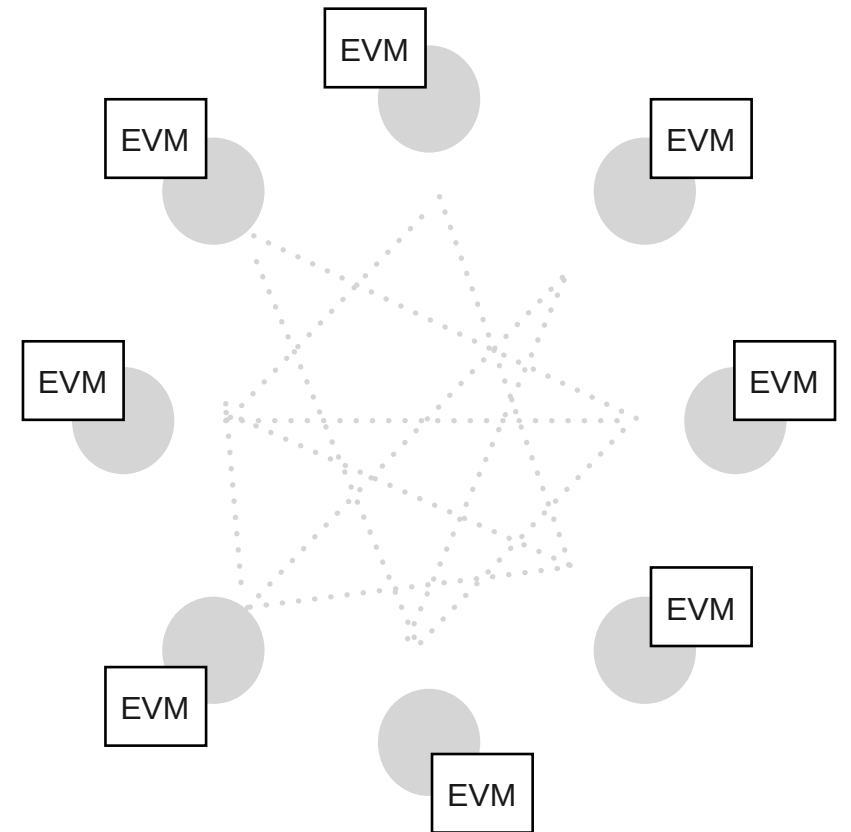
Account vs UTXO - Introduction

Account-based

- Global state stores a list of accounts with balances and code

- Transaction is valid if the sending account has enough balance

  - Balance on sender is deducted, new balance

- If the receiving account has code, the code runs, and state may be changed
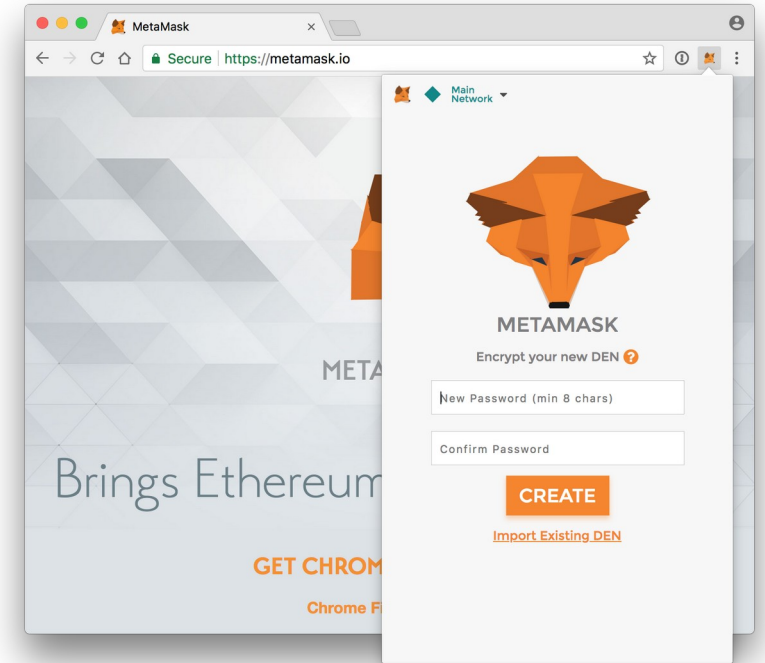
  - Signature must match sending account
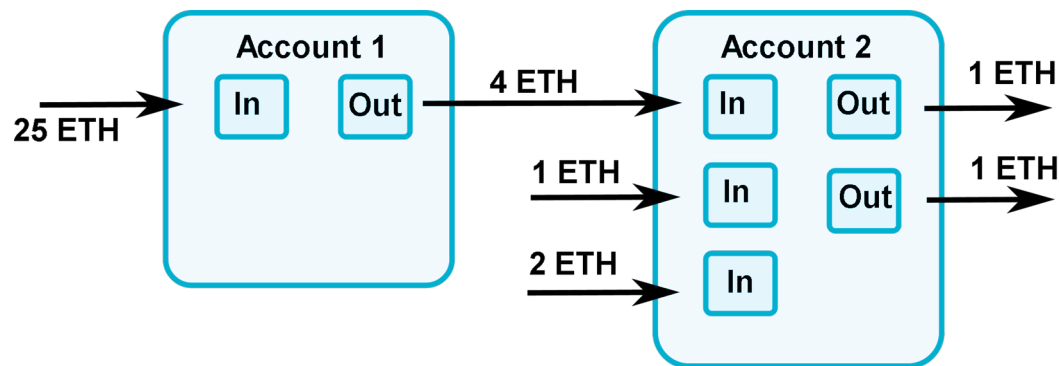
UTXO-based

- Every referenced input must be valid and not yet spent

- Total value of the inputs must equal or exceed the total value of the outputs

  - You always spend all outputs

- Transaction must have a signature matching the owner of the input for every input

  - Script determines if input is valid
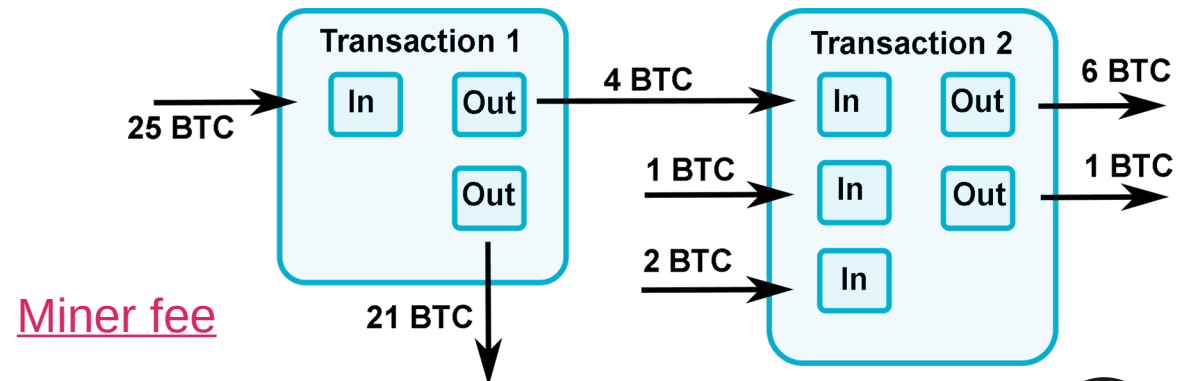
MetaMask

- MetaMask

  - Web3 browser plugin to make Ethereum transactions in browsers

  - Manage your key pairs and sign blockchain transactions

  - MetaMask injects javascript library - ethers.js

  - Uses infura

- Remix IDE: https://remix.ethereum.org

- Testnet: rinkeby/goerli/ropsten – merge on 02.06.2022)

  - https://rinkeby.etherscan.io/ (blockchain explorer)

  - No mining (use twitter with https://www.rinkeby.io)

Example

- Installation

  - npm install

  - ./node_modules/.bin/webpack

  - ./node_modules/.bin/webpack serve

- Open Browser: http://localhost:8080/