



OST

Eastern Switzerland
University of Applied Sciences

Blockchain (BlCh)

Repetition DSy – part 2

Thomas Bocek

25.09.2022

Lecture 10



Introduction

- Bitcoin is an experimental digital currency
 - Bitcoin is fully peer-2-peer (no central entity)
 - 1st Bitcoin issued on January 3, 2009
 - Smallest unit: 0.00000001 BTC (1 satoshi)
- Key characteristics
 - **Maximum** of **~21 million BTC**
 - Every transaction broadcast to all peers
 - Every peers knows all transactions (~**400 GByte** as of today)
 - Validation by proof-of-work (partial hash collision)
 - Difficult to fake proof-of-work
 - No double-spending
- The initiator is unknown so far

```

draft@home: /scratch/bitcoin/blocks
File Edit View Search Terminal Help
blk00000.dat blk00002.dat blk00004.dat blk00006.dat blk00008.dat
blk00001.dat blk00003.dat blk00005.dat blk00007.dat blk00009.dat
draft@home:/scratch/bitcoin/blocks$ head -c 300 blk00000.dat | hexdump -C
00000000 f9 be b4 d9 1d 01 00 00 01 00 00 00 00 00 00 00 | .....|
00000010 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | .....|
00000020 00 00 00 00 00 00 00 00 00 00 00 00 3b a3 ed fd | .....;...|
00000030 7a 7b 12 b2 7a c7 2c 3e 67 76 8f 61 7f c8 1b c3 | z{...z,>gv.a...|
00000040 88 8a 51 32 3a 9f b8 aa 4b 1e 5e 4a 29 ab 5f 49 | ..Q2:...K.^J)...I|
00000050 ff ff 00 1d 1d ac 2b 7c 01 01 00 00 00 01 00 00 | .....+|.....|
00000060 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | .....|
00000070 00 00 00 00 00 00 00 00 00 00 00 00 00 ff ff | .....|
00000080 ff ff 4d 04 ff ff 00 1d 01 04 45 54 68 65 20 54 | ..M.....EThe T|
00000090 69 6d 65 73 20 30 33 2f 4a 61 6e 2f 32 30 30 39 | imes 03/Jan/2009|
000000a0 20 43 68 61 6e 63 65 6c 6c 6f 72 20 6f 6e 20 62 | Chancellor on b|
000000b0 72 69 6e 6b 20 6f 66 20 73 65 63 6f 6e 64 20 62 | rink of second b|
000000c0 61 69 6c 6f 75 74 20 66 6f 72 20 62 61 6e 6b 73 | ailout for banks|
000000d0 ff ff ff ff 01 00 f2 05 2a 01 00 00 00 43 41 04 | .....*....CA.|
000000e0 67 8a fd b0 fe 55 48 27 19 67 f1 a6 71 30 b7 10 | g...UH'.g..q0..|
000000f0 5c d6 a8 28 e0 39 09 a6 79 62 e0 ea 1f 61 de b6 | \..(.9..yb...a..|
00000100 49 f6 bc 3f 4c ef 38 c4 f3 55 04 e5 1e c1 12 de | I..?L8..U.....|
00000110 5c 38 4d f7 ba 0b 8d 57 8a 4c 70 2b 6b f1 1d 5f | \8M...W.Lp+k...|
00000120 ac 00 00 00 00 f9 be b4 d9 d7 00 00 | .....|
0000012c
draft@home:/scratch/bitcoin/blocks$

```

Who is Satoshi Nakamoto?

- **The New Yorker** believes that Satoshi Nakamoto was Michael Clear.
 - Analyzed texts from Nakamoto and searching for linguistic clues
 - 2nd possible candidate Vili Lehdonvirta
- **Fast Company** argues its either Neal King, Vladimir Oksman, or Charles Bry.
- Other names suggested: **Martii Malmi** (involved in Bitcoins since the beginning), **Jed McCaleb** (founder of Ripple), **Donal O'Mahony**, **Michael Peirce**, **Hitesh Tewari** (authors of **Electronic Payment Systems for E-Commerce 2nd edition**), **Shinichi Mochizuki** (Math Prof. Kyoto University), Hal Finney, Michael Weber, Wei Dai, **Nick Szabo**, Craig Wright (**wired article**),
- **Dorian S Nakamoto** (a guy with the same name)
- Satoshi is probably rich, first miner, **may have ~1mio BTC**
- Craig Wright, May 2016: «**I'm Satoshi Nakamoto**», fails to **deliver proof**

Bitcoin - Introduction

- Not relying on trust, but on strong cryptography
- Weak anonymity (pseudonymity)
 - All peers know all transactions
 - **Clustering**: e.g. if a transaction has multiple input addresses, assume those addresses belong to the same wallet. (**example**)
- Not controlled by a single entity
 - Development community, no central bank – forks – Bitcoin Cash, SV
- **BIP**: Bitcoin Improvement Proposals
- Bitcoins can be exchange for real currencies
 - Several companies allow to exchange BTC for Dollar, Euro, ...
- US, CH considered Bitcoin friendly, **China** (**energy**), **Turkey** not that much

Mechanism

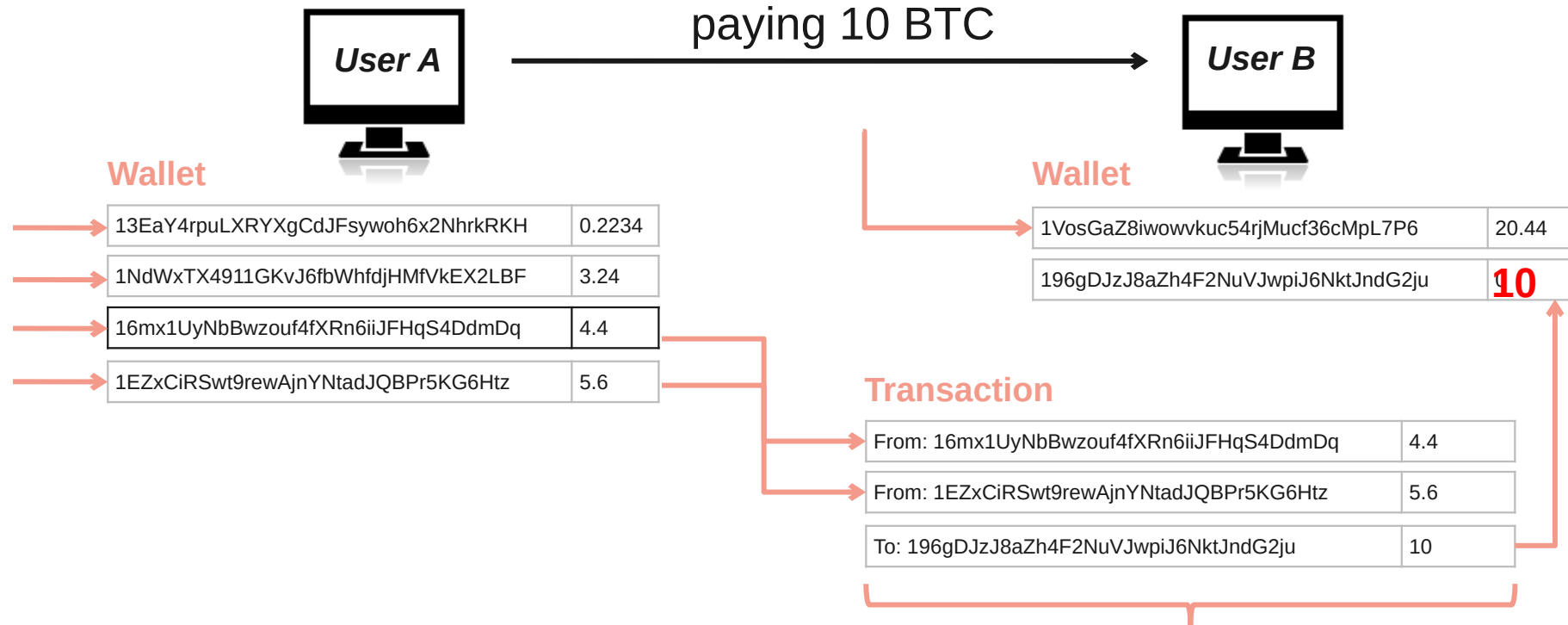
- A wallet has public-private keys (wallet.dat)
 - Public key, ECDSA 256 bit → Bitcoin address (can receive bitcoins)
 - Simple address ~ base58(RIPEM160(Sha256(ecdsa public key)))
 - E.g. 1GCeaKuhDYnNLNR6LGmBtKhPqEJD4KeEtF
 - Private key used for signing transactions

- Transaction
 - Peer A wants to send BTC to peer B → creates transaction message
 - Transaction contains input / output
 - where the BTC came from and where it goes
 - Peer A broadcasts the transaction to all the peers in the network
 - Transaction stored in blocks → block is created / verified ~10min



Key Bitcoin Operations

- Private key authorizes the transaction (“access”)
 - If keys are stolen, thief may use “your” coins
 - If keys are lost, coins are lost
 - In UTXO (unspent transaction output) systems, complete output is spent



Sign with Private Key of User A

Bitcoin Scripting Language

- ScriptSig

PUSHDATA

signature data and SIGHASH_ALL

PUSHDATA

public key data

- ScriptPubKey

OP_DUP

OP_HASH160

PUSHDATA

Bitcoin address (public key hash)

OP_EQUALVERIFY

OP_CHECKSIG

- Non-turing complete (e.g. No loops)

- With scripts

- Multisig, n-of-m, escrow and dispute mediation
- Micropayment channel, refund tx in future

- Opcodes – [all codes](#)

- Data operations

- OP_PUSHDATA1, OP_PUSHDATA4,...

- Flow control

- OP_IF, OP_ELSE, ...

- Stack

- OP_DUP, OP_SWAP, ...

- Arithmetic

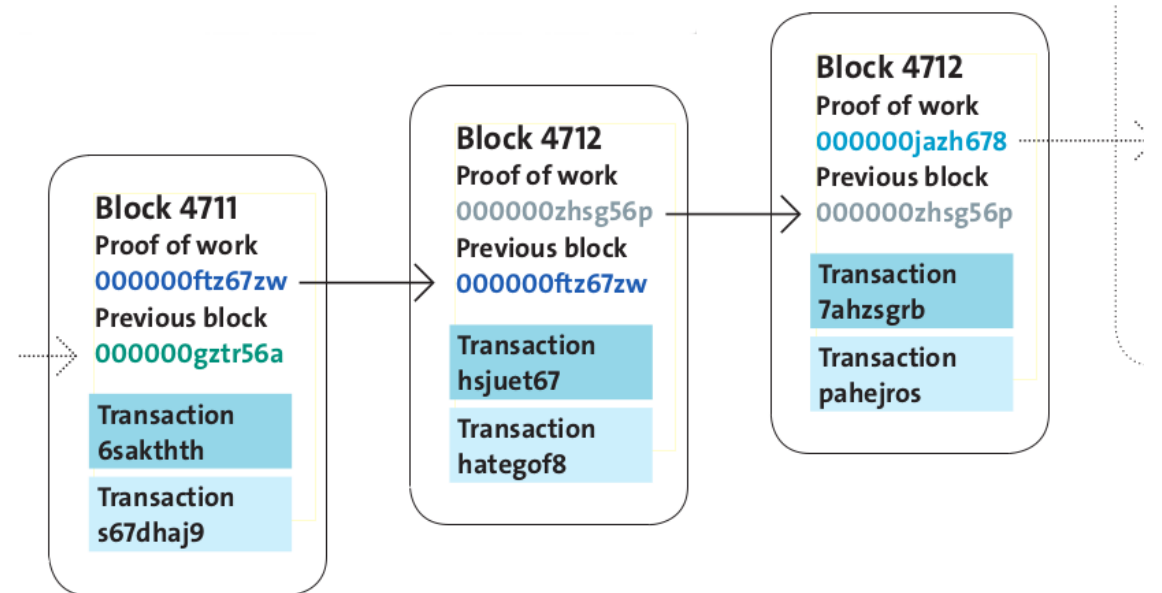
- OP_ADD, OP_ABS, ...

- Crypto

- OP_SHA256, OP_CHECKSIGVERIFY

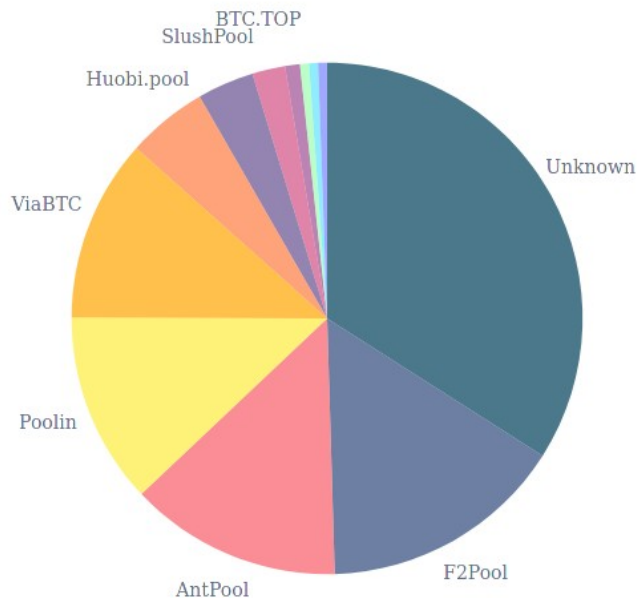
Blockchain

- Transactions are collected in blocks
 - New block created approximately every 10 min
- Blocks contain solved crypto puzzles
 - In the form of partial hash collisions (SHA256)
- A block has a pointer to previous block → Blockchain
- Creation of blocks is called mining (reward)
 - Miners use highly specialized hardware!



Mechanism - Mining

- Couple of big miners
- Miners specialized, AMD GPUs, FPGA, ASIC (application-specific integrated circuit) [1][2][3]



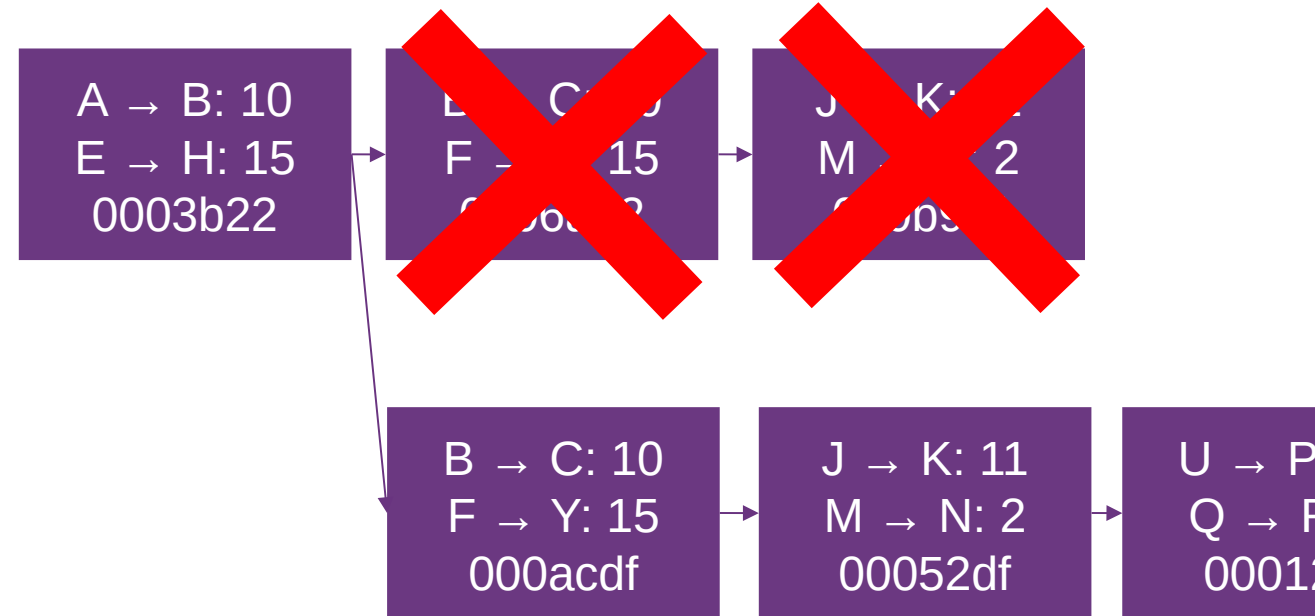
<http://blockchain.info/pools>

- Mining = creating valid blocks
- Blocks are linked to previous blocks
 - Longest block survive (most difficult)
- Different level of confirmations
 - 3-6 block conf. is considered secure
- Dangerous if someone has more than 50% computing power
 - Can exclude and modify the ordering of transactions

Lecture 13

51% Attack

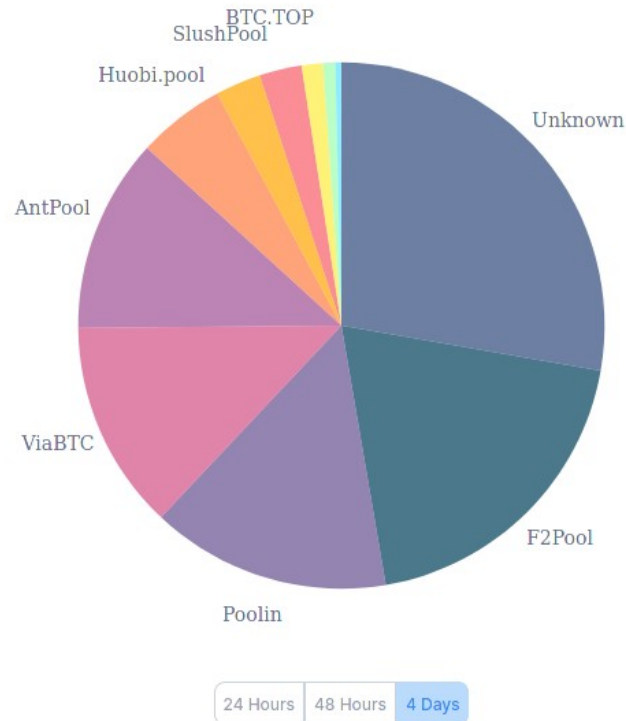
- “If a majority of CPU power is controlled by honest nodes, the honest chain will grow the fastest and outpace any competing chains.”
 - <https://bitcoin.org/bitcoin.pdf>
- PoW: majority of hashing power, PoS: majority of coins
- How expensive is a 51% attack?
 - Buy an attack?
- Double spend, or rollback 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



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 [\[link\]](#)

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, arithemits, 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

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

```

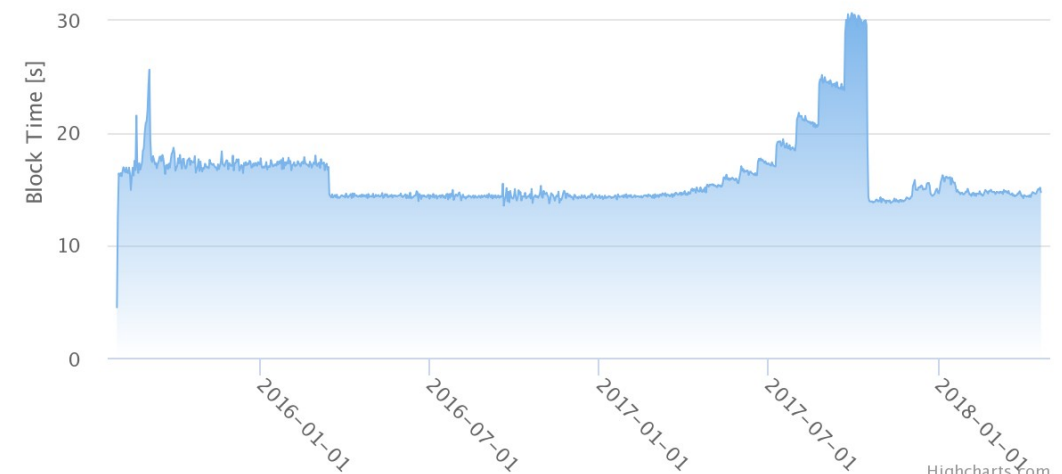
Wzero = {STOP, RETURN}
Wbase = {ADDRESS, ORIGIN, CALLER, CALVALUE, CALLDATASIZE, CODESIZE, GASPRICE, COINBASE,
TIMESTAMP, NUMBER, DIFFICULTY, GASLIMIT, POP, PC, MSIZE, GAS}
Wverylow = {ADD, SUB, NOT, LT, GT, SLT, SGT, EQ, ISZERO, AND, OR, XOR, BYTE, CALLDATALOAD,
MLOAD, MSTORE, MSTORE8, PUSH*, DUP*, SWAP*}
Wlow = {MUL, DIV, SDIV, MOD, SMOD, SIGNEXTEND}
Wmid = {ADDMOD, MULMOD, JUMP}
Whigh = {JUMPI}
Wextcode = {EXTCODESIZE}

```

[source](#)

The fee schedule G is a tuple of 31 scalar values corresponding to the relative costs, in gas, of a number of abstract operations that a transaction may effect.

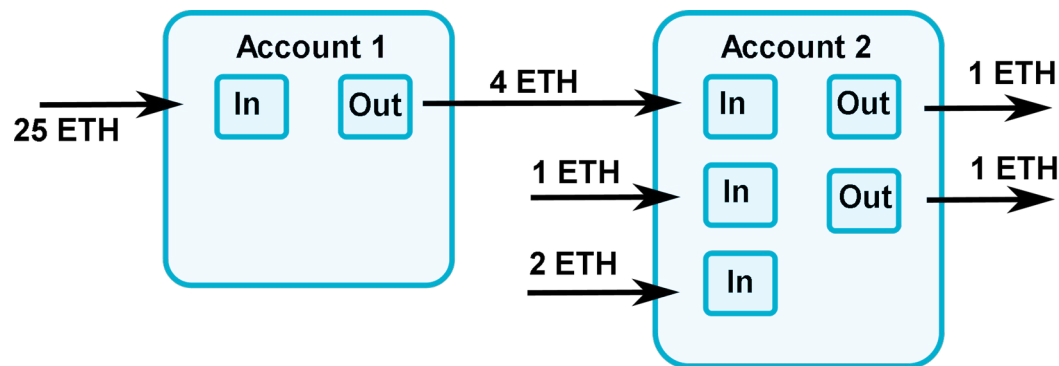
Name	Value	Description*
G_{zero}	0	Nothing paid for operations of the set W_{zero} .
G_{base}	2	Amount of gas to pay for operations of the set W_{base} .
$G_{verylow}$	3	Amount of gas to pay for operations of the set $W_{verylow}$.
G_{low}	5	Amount of gas to pay for operations of the set W_{low} .
G_{mid}	8	Amount of gas to pay for operations of the set W_{mid} .
G_{high}	10	Amount of gas to pay for operations of the set W_{high} .
$G_{extcode}$	700	Amount of gas to pay for operations of the set $W_{extcode}$.
$G_{balance}$	400	Amount of gas to pay for a BALANCE operation.
G_{sload}	200	Paid for a SLOAD operation.
$G_{jumpdest}$	1	Paid for a JUMPDEST operation.
G_{sset}	20000	Paid for an SSTORE operation when the storage value is set to non-zero from zero.
G_{sreset}	5000	Paid for an SSTORE operation when the storage value's zeroness remains unchanged or is set to zero.
R_{sclear}	15000	Refund given (added into refund counter) when the storage value is set to zero from non-zero.
$R_{suicide}$	24000	Refund given (added into refund counter) for suiciding an account.
$G_{suicide}$	5000	Amount of gas to pay for a SUICIDE operation.
G_{create}	32000	Paid for a CREATE operation.
$G_{codedeposit}$	200	Paid per byte for a CREATE operation to succeed in placing code into state.
G_{call}	700	Paid for a CALL operation.
$G_{callvalue}$	9000	Paid for a non-zero value transfer as part of the CALL operation.
$G_{callstipend}$	2300	A stipend for the called contract subtracted from $G_{callvalue}$ for a non-zero value transfer.
$G_{newaccount}$	25000	Paid for a CALL or SUICIDE operation which creates an account.
G_{exp}	10	Partial payment for an EXP operation.
$G_{expbyte}$	10	Partial payment when multiplied by $\lceil \log_{256}(exponent) \rceil$ for the EXP operation.
G_{memory}	3	Paid for every additional word when expanding memory.
$G_{txcreate}$	32000	Paid by all contract-creating transactions after the <i>Homestead transition</i> .
$G_{txdatazero}$	4	Paid for every zero byte of data or code for a transaction.
$G_{txdatanonzero}$	68	Paid for every non-zero byte of data or code for a transaction.
$G_{transaction}$	21000	Paid for every transaction.
G_{log}	375	Partial payment for a LOG operation.
$G_{logdata}$	8	Paid for each byte in a LOG operation's data.
$G_{logtopic}$	375	Paid for each topic of a LOG operation.
G_{sha3}	30	Paid for each SHA3 operation.
$G_{sha3word}$	6	Paid for each word (rounded up) for input data to a SHA3 operation.
G_{copy}	3	Partial payment for *COPY operations, multiplied by words copied, rounded up.
$G_{blockhash}$	20	Payment for BLOCKHASH operation.



Account vs UTXO - Introduction

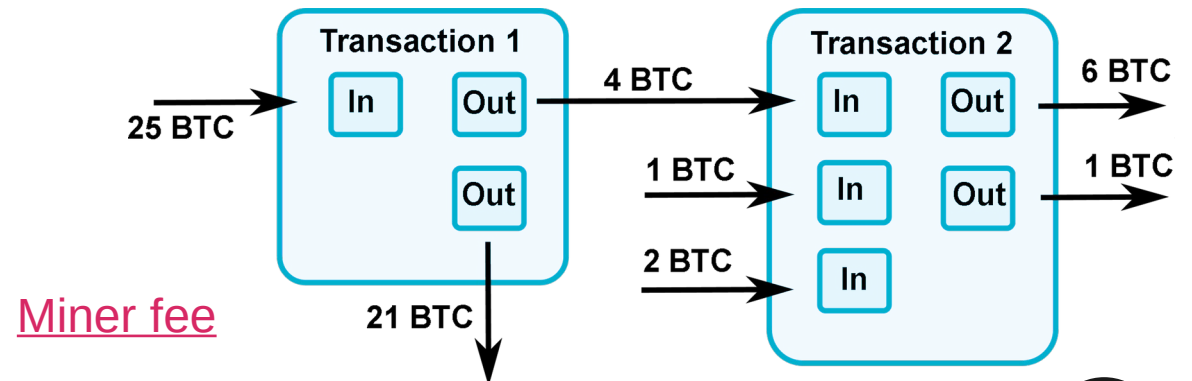
Account-based

- 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



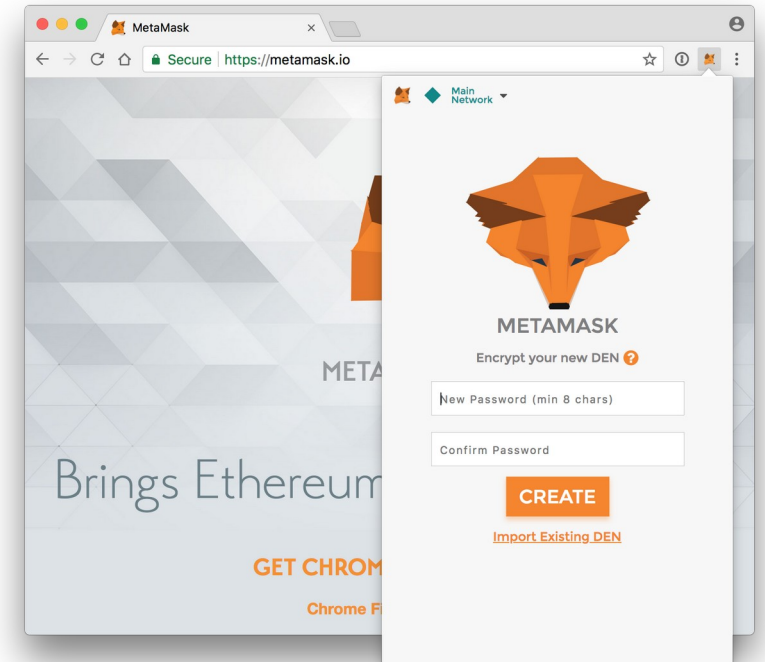
UTXO-based

- 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)

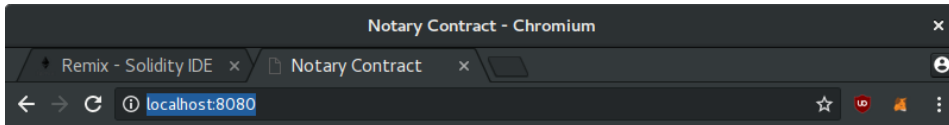


```
INFO [05-08|17:14:43] Commit new mining work          number=891910 txs=0 uncles=0 elapsed=392.257µs
INFO [05-08|17:15:06] Imported new chain segment          blocks=1 txs=0 mgas=0.000 elapsed=17.225ms mgasps=0.000 number=891910 hash=812c12...de3c2e
INFO [05-08|17:15:06] Commit new mining work          number=891911 txs=2 uncles=0 elapsed=6.039ms
INFO [05-08|17:15:16] Successfully sealed new block      number=891911 hash=9efde0...7642c5
INFO [05-08|17:15:16] ^ mined potential block           number=891911 hash=9efde0...7642c5
INFO [05-08|17:15:16] Commit new mining work          number=891912 txs=0 uncles=0 elapsed=507.117µs
INFO [05-08|17:15:23] Imported new chain segment          blocks=1 txs=0 mgas=0.000 elapsed=6.596ms mgasps=0.000 number=891912 hash=c80dc0...5fbfde
```

- 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/>



Notarize PDF



```
draft@home: ~/git/VSS-web3js
File Edit View Search Terminal Help
draft@home:~/git/VSS-web3js$ ./node_modules/.bin/webpack-dev-server
i [wds]: Project is running at http://localhost:8080/
i [wds]: webpack output is served from /
i [wdm]: Hash: c4c7c0d3279286de6649
Version: webpack 4.7.0
Time: 1139ms
Built at: 2018-05-06 12:57:52
    Asset      Size  Chunks             Chunk Names
main.c4c7c0d3279286de6649.js  947 KiB    main    [emitted]  main
    index.html  395 bytes                [emitted]
Entrypoint main = main.c4c7c0d3279286de6649.js
[./node_modules/ansi-html/index.js] 4.16 KiB {main} [built]
[./node_modules/loglevel/lib/loglevel.js] 7.68 KiB {main} [built]
[./node_modules/strip-ansi/index.js] 161 bytes {main} [built]
[./node_modules/url/url.js] 22.8 KiB {main} [built]
[./node_modules/vue/dist/vue.esm.js] 286 KiB {main} [built]
[./node_modules/webpack-dev-server/client/index.js?http://localhost:8080] (webpack)-dev-server/client?http://localhost:8080 7.75 KiB {main} [built]
[./node_modules/webpack-dev-server/client/overlay.js] (webpack)-dev-server/client/overlay.js 3.58 KiB {main} [built]
[./node_modules/webpack-dev-server/client/socket.js] (webpack)-dev-server/client/socket.js 1.05 KiB {main} [built]
[./node_modules/webpack/hot sync ^\\.\\.log$] (webpack)/hot sync nonrecursive ^\\.\\.log$ 170 bytes {main} [built]
[./node_modules/webpack/hot/emitter.js] (webpack)/hot/emitter.js 77 bytes {main} [built]
[./node_modules/webpack/hot/log.js] (webpack)/hot/log.js 1010 bytes {main} [optional] [built]
[./src/App.vue] 908 bytes {main} [built]
[./src/App.vue?vue&type=template&id=7ba5bd90] 194 bytes {main} [built]
[0] multi (webpack)-dev-server/client?http://localhost:8080 ./src 40 bytes {main} [built]
[./src/index.js] 129 bytes {main} [built]
+ 63 hidden modules
Child html-webpack-plugin for "index.html":
  1 asset
  Entrypoint undefined = index.html
  [./node_modules/html-webpack-plugin/lib/loader.js!./index.html] 527 bytes {0} [built]
  [./node_modules/lodash/lodash.js] 527 KiB {0} [built]
  [./node_modules/webpack/buildin/global.js] (webpack)/buildin/global.js 489 bytes {0} [built]
  [./node_modules/webpack/buildin/module.js] (webpack)/buildin/module.js 497 bytes {0} [built]
i [wdm]: Compiled successfully.
```