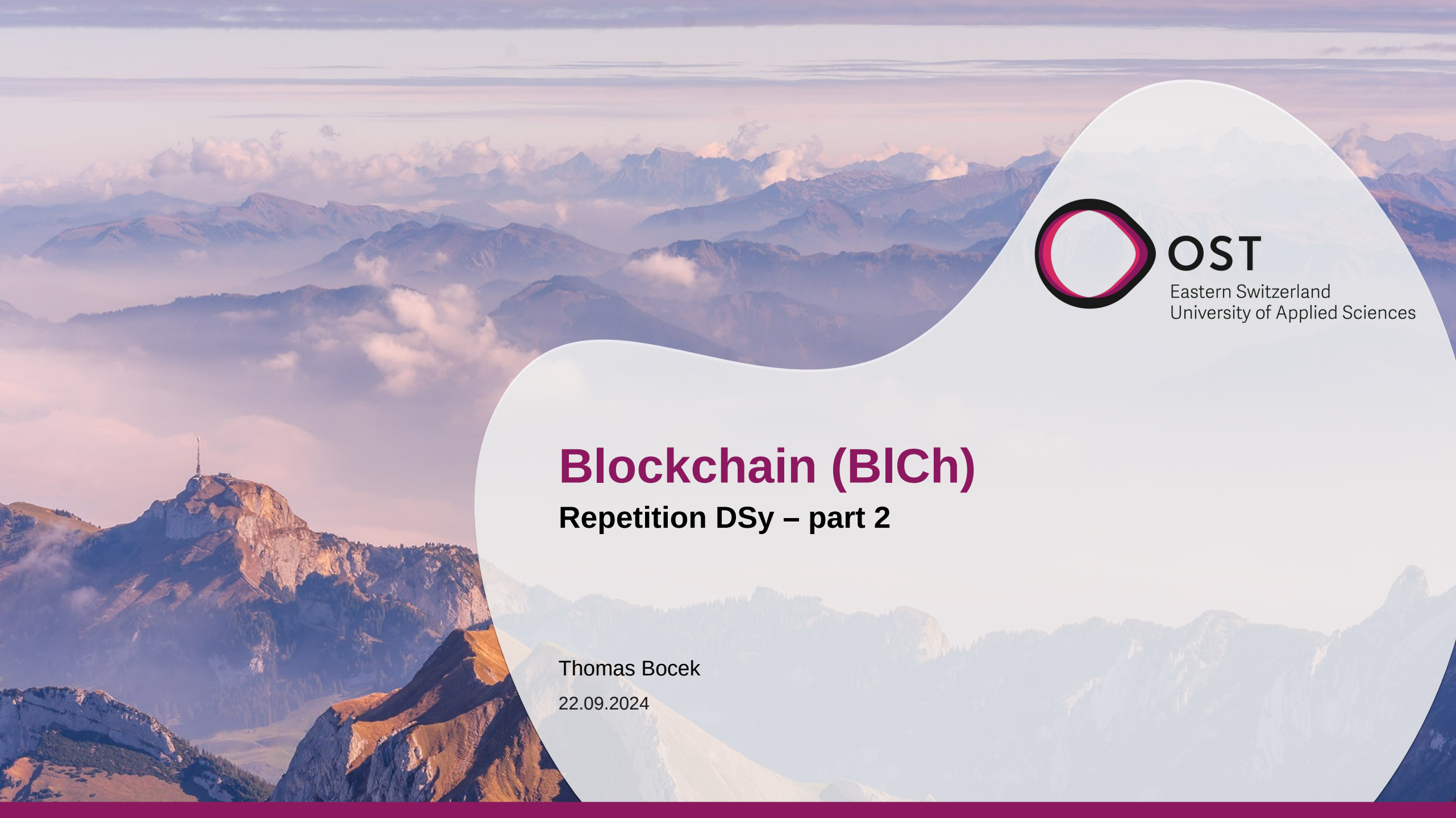


OST
Eastern Switzerland
University of Applied Sciences

Blockchain (BlCh)

Repetition DSy – part 2

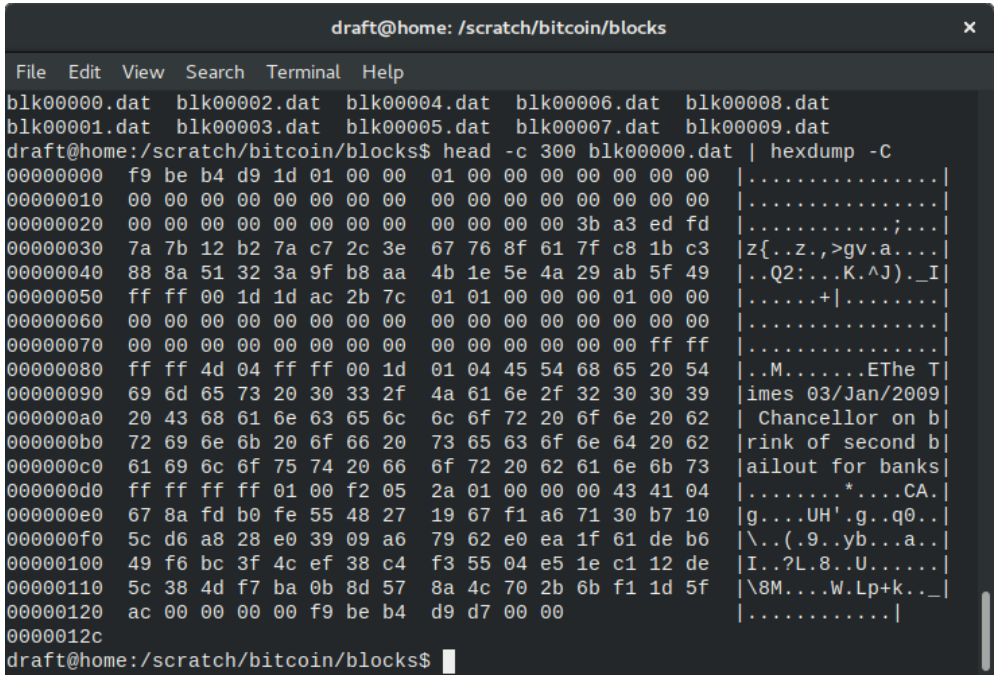
Thomas Bocek

22.09.2024

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



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

OST

Bitcoin - Introduction

- Not relying on trust, but on strong cryptography
- Weak anonymity (pseudonimity)
  - 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

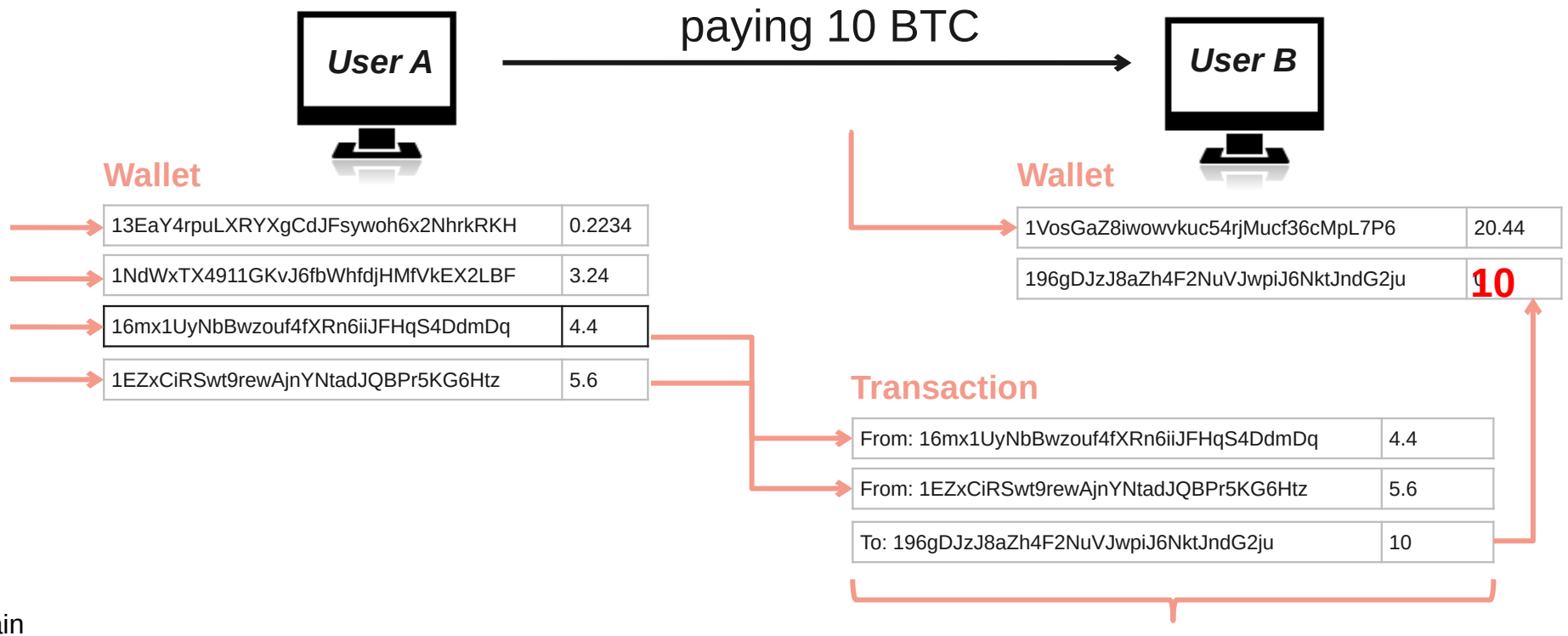
- A wallet has public-private keys (wallet.dat)
  - Public key, ECDSA 256 bit → Bitcoin address (can receive bitcoins)
  - Simple address ~ base58(RIPEM160(Sha256(ecdsa public key)))
    - E.g. 1GCeaKuhDYnNLNR6LGmBtKhPqEJD4KeEtF
  - Private key used for signing transactions

- 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

OST

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



OST

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

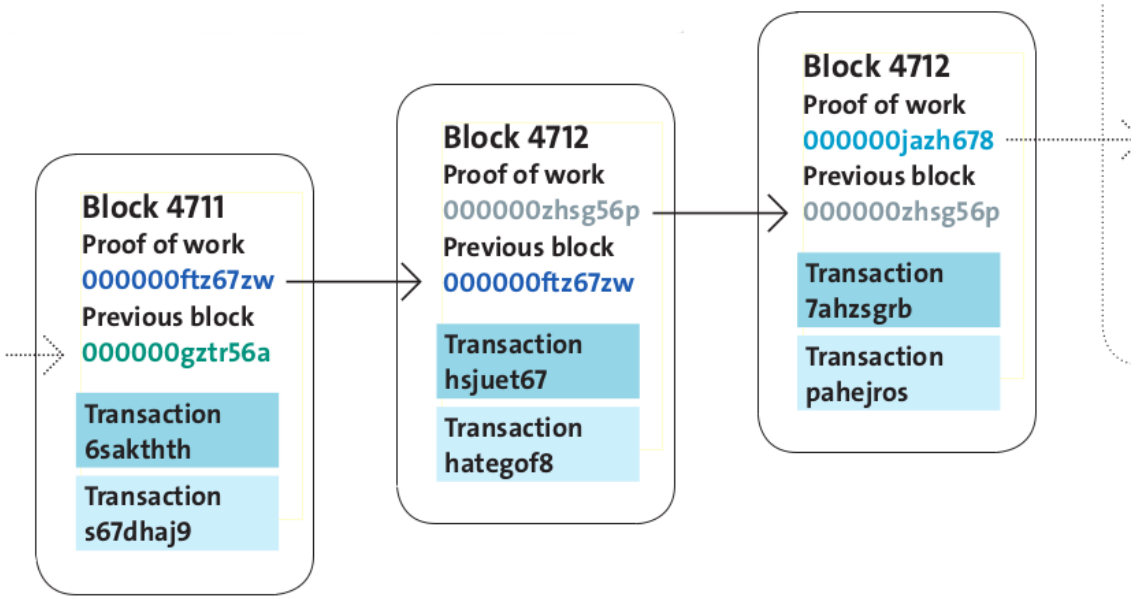
  OP_CHECKSIG

- 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

OST

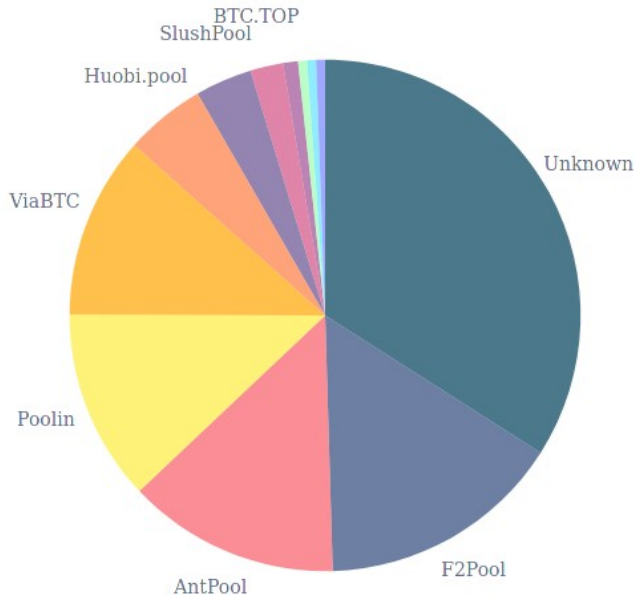
Blockchain

- 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

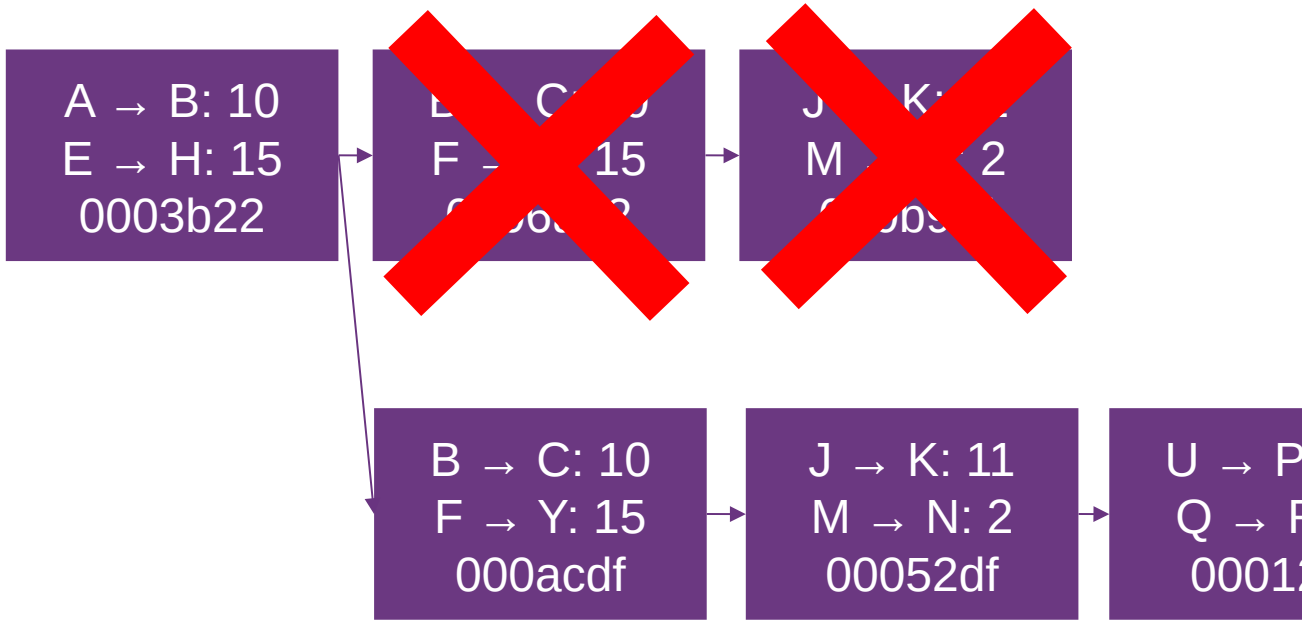
OST

Lecture 11

OST

51% Attack

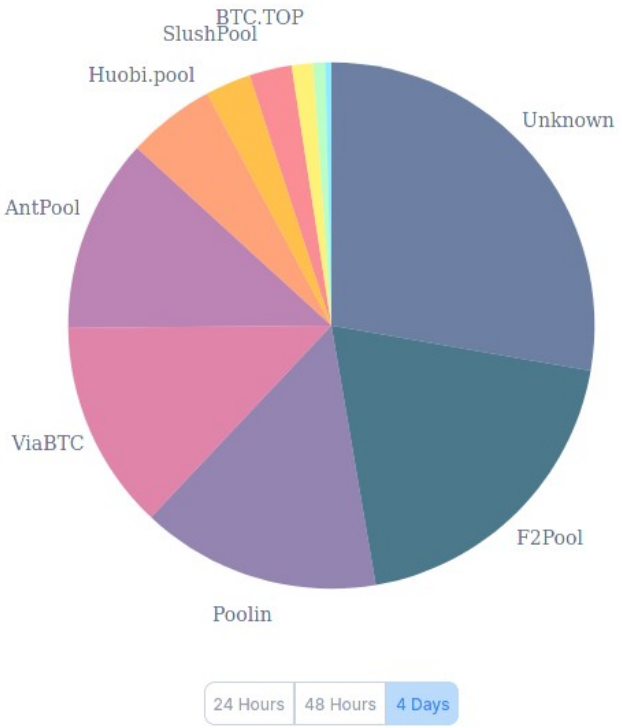
- “If a majority of CPU power is controlled by honest nodes, the honest chain will grow the fastest and outpace any competing chains.”
  - https://bitcoin.org/bitcoin.pdf
- PoW: majority of hashing power, PoS: majority of coins
- How expensive is a 51% attack?
  - Buy an attack?
- Double spend, or rollback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



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

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

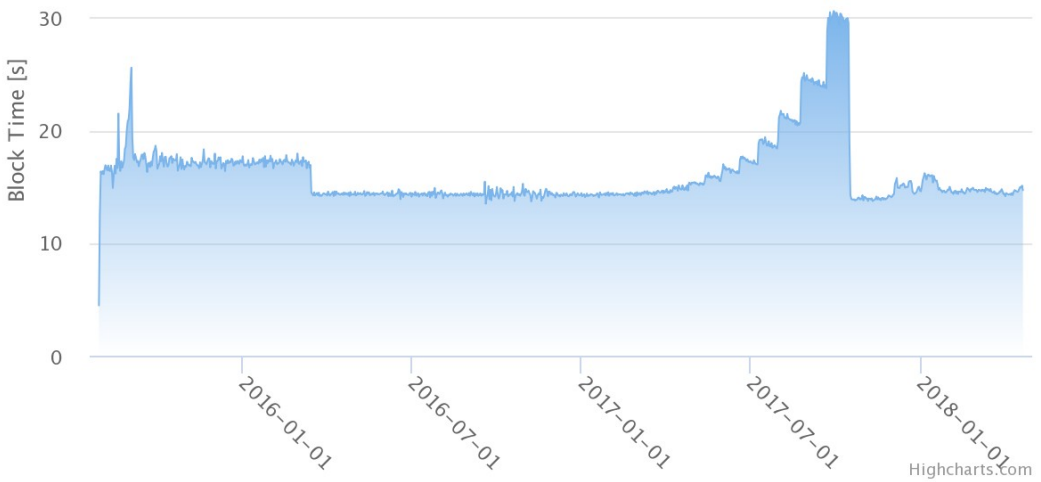
$W_{zero} = \{\text{STOP, RETURN}\}$
$W_{base} = \{\text{ADDRESS, ORIGIN, CALLER, CALLVALUE, CALLDATASIZE, CODESIZE, GASPRICE, COINBASE, TIMESTAMP, NUMBER, DIFFICULTY, GASLIMIT, POP, PC, MSIZE, GAS}\}$
$W_{verylow} = \{\text{ADD, SUB, NOT, LT, GT, SLT, SGT, EQ, ISZERO, AND, OR, XOR, BYTE, CALLDATALOAD, MLOAD, MSTORE, MSTORE8, PUSH*, DUP*, SWAP*}\}$
$W_{low} = \{\text{MUL, DIV, SDIV, MOD, SMOD, SIGNEXTEND}\}$
$W_{mid} = \{\text{ADDMOD, MULMOD, JUMP}\}$
$W_{high} = \{\text{JUMPI}\}$
$W_{extcode} = \{\text{EXTCODESIZE}\}$

source

Appendix G. Fee Schedule

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.

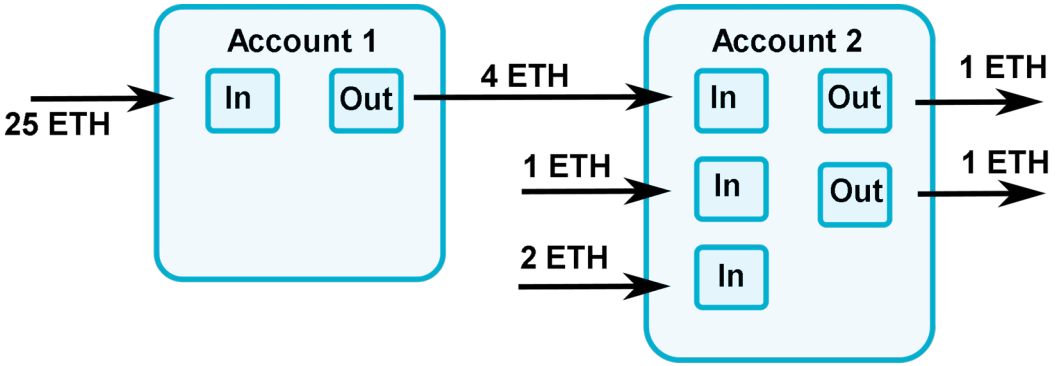
Name	Value	Description*
G_{zero}	0	Nothing paid for operations of the set W_{zero}.
G_{base}	2	Amount of gas to pay for operations of the set W_{base}.
$G_{verylow}$	3	Amount of gas to pay for operations of the set $W_{verylow}$.
G_{low}	5	Amount of gas to pay for operations of the set W_{low}.
G_{mid}	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 *Homestead transition*.
$G_{txdatazero}$	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

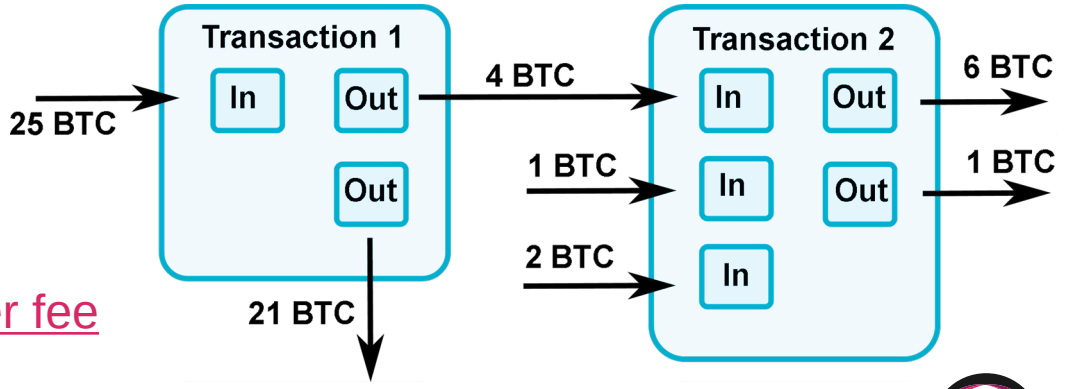
Account-based

- 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

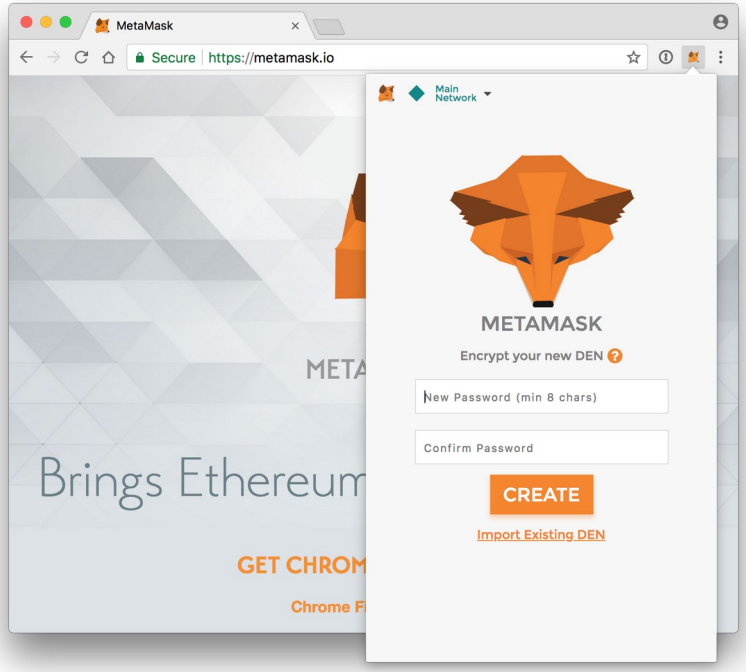
- Every referenced input must be valid and not yet spent
- Total value of the inputs must equal or exceed the total value of the outputs
  - You always spend all outputs
- Transaction must have a signature matching the owner of the input for every input
  - Script determines if input is valid



OST

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)



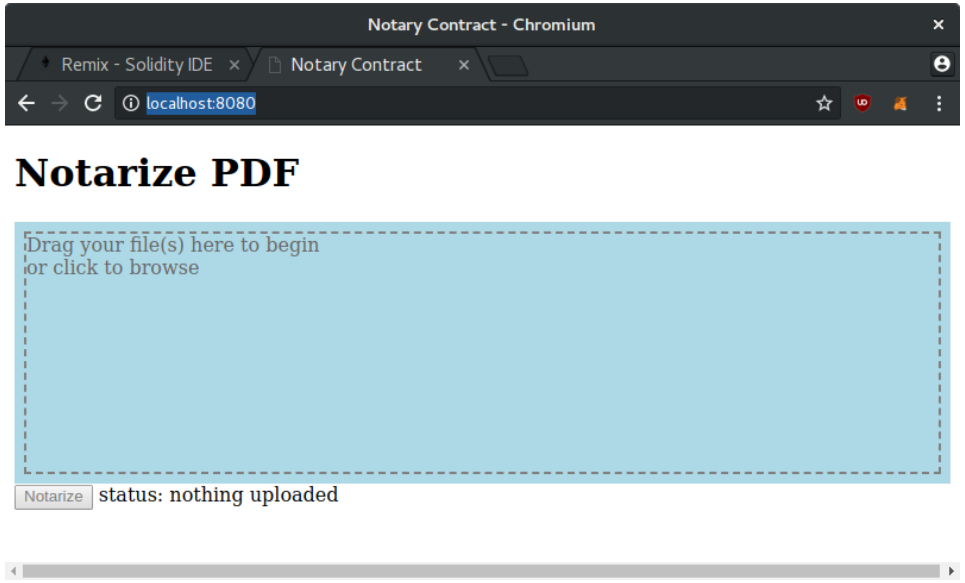

```
INFO [05-08|17:14:43] Commit new mining work                   number=891910 txs=0  uncles=0 elapsed=392.257µs
INFO [05-08|17:15:06] Imported new chain segment               blocks=1 txs=0  mgas=0.000 elapsed=17.225ms  mgasps=0.000   number=891910 hash=812c12…de3c2e
INFO [05-08|17:15:06] Commit new 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)

OST

Example

- Installation
  - npm install
  - ./node_modules/.bin/webpack
  - ./node_modules/.bin/webpack serve
- Open Browser: http://localhost:8080/




```
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/c
lient?http://localhost:8080 7.75 KiB {main} [built]
[./node_modules/webpack-dev-server/client/overlay.js] (webpack)-dev-server/client/overlay.js 3.5
8 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.
```


OST