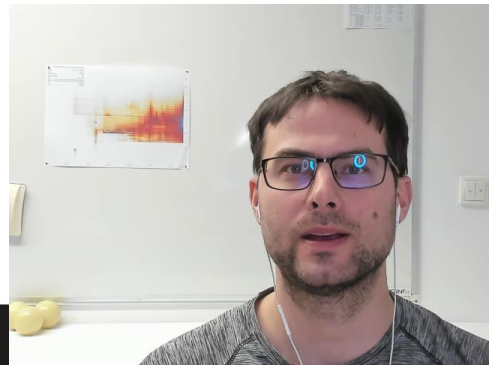
PV204 Security technologies

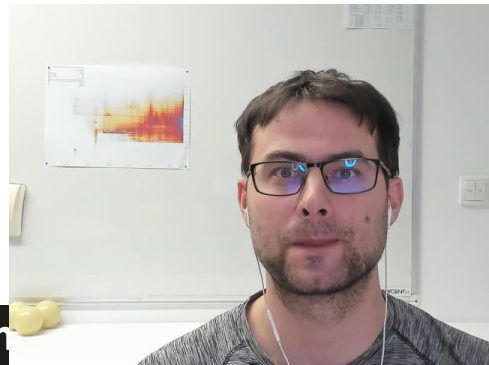
Cryptocurrencies II.

Petr Švenda ***svenda@fi.muni.cz*** ***@rngsec***
Centre for Research on Cryptography and Security, Masaryk University

Please provide any corrections and comments here (thank you!):
https://drive.google.com/file/d/1DH1rooFx6ZXNflaHRHqvfOAHXc_qikc3/view?usp=sharing



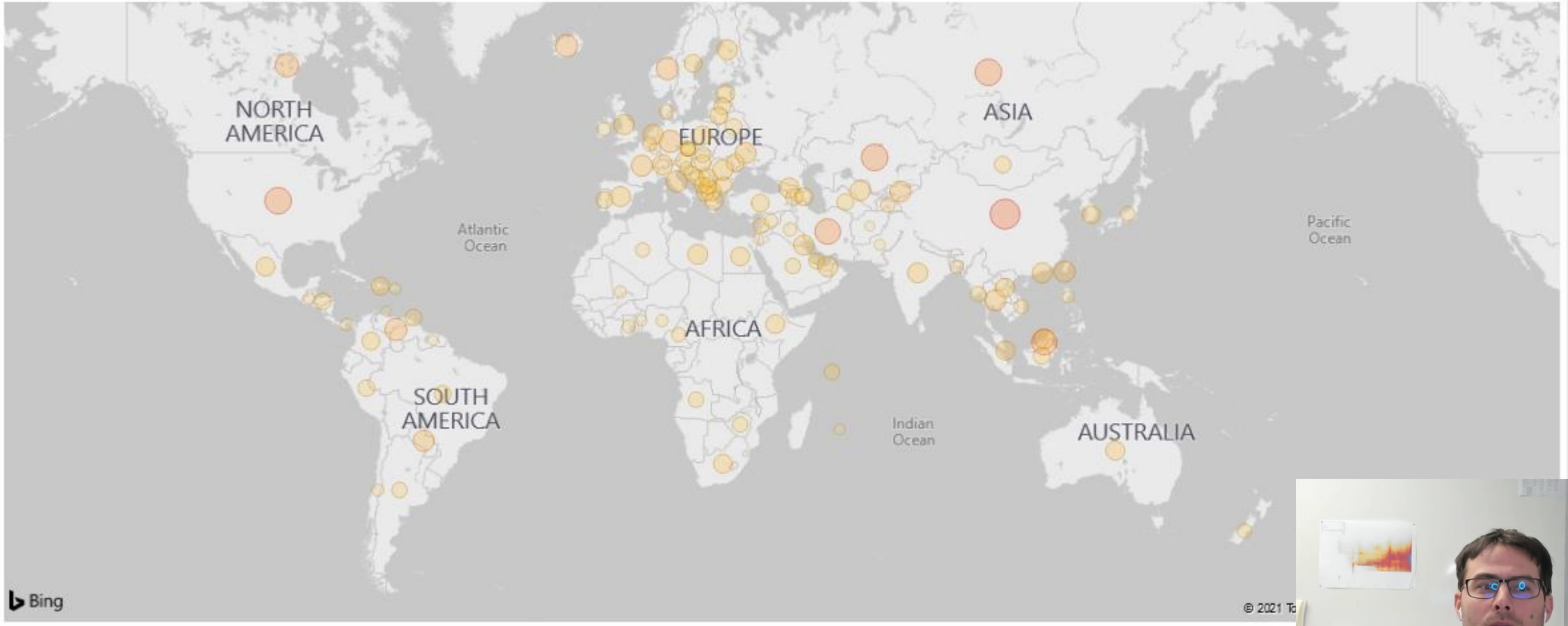
MINING

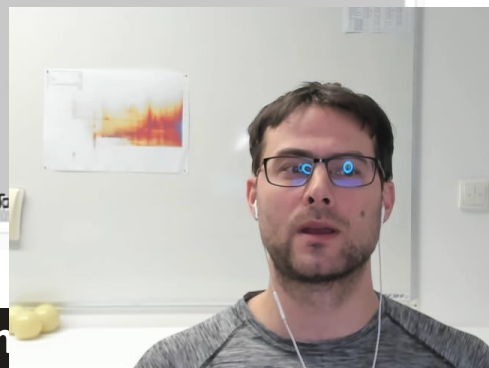


Mining

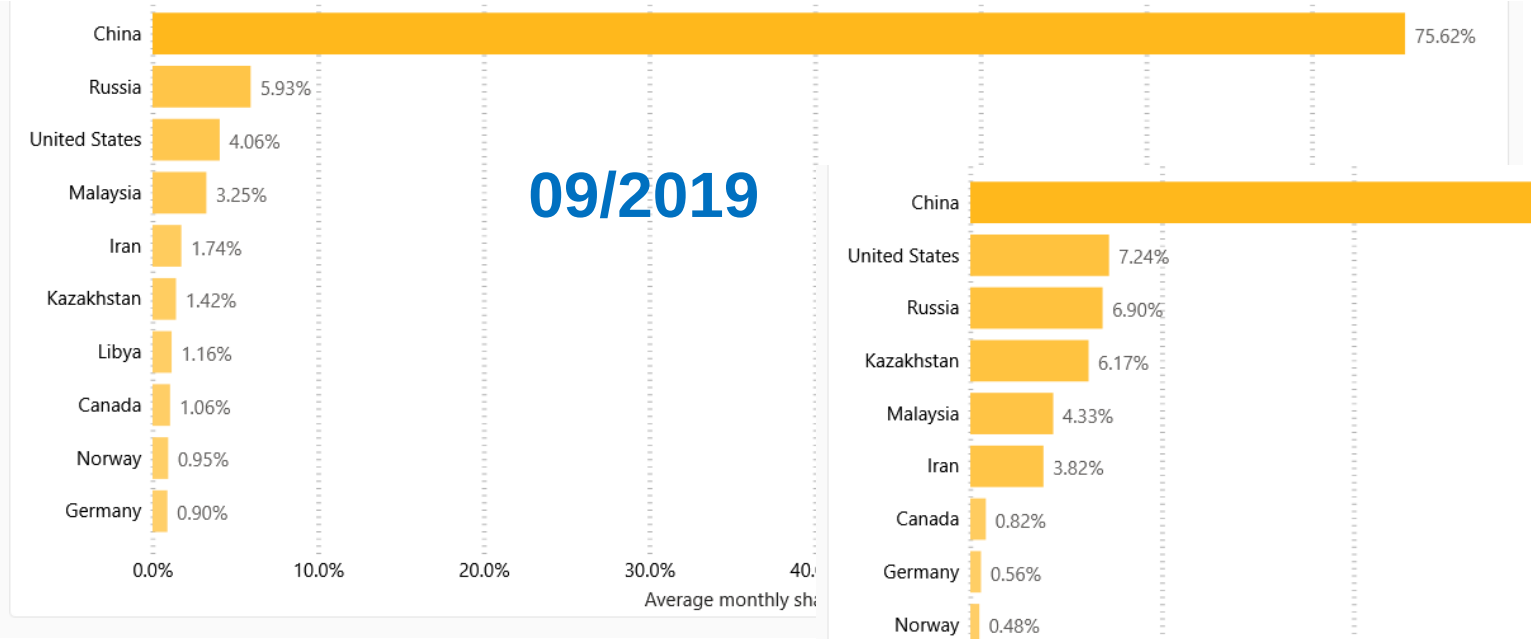
- Initially on CPU (Satoshi: everyone can participate 1 CPU 1 vote)
- Initially solo mining
- CPU→GPU →FPGA →ASIC
- First mining pool: SlushPool in Prague
  - Miners join hashrate, fraction of reward based on number of partial solutions
- Cambridge university centre for alternative finance (CBECI)
  - Where are miners? https://cbeci.org/mining_map/
  - More mining details: https://cbeci.org/cbeci/methodology

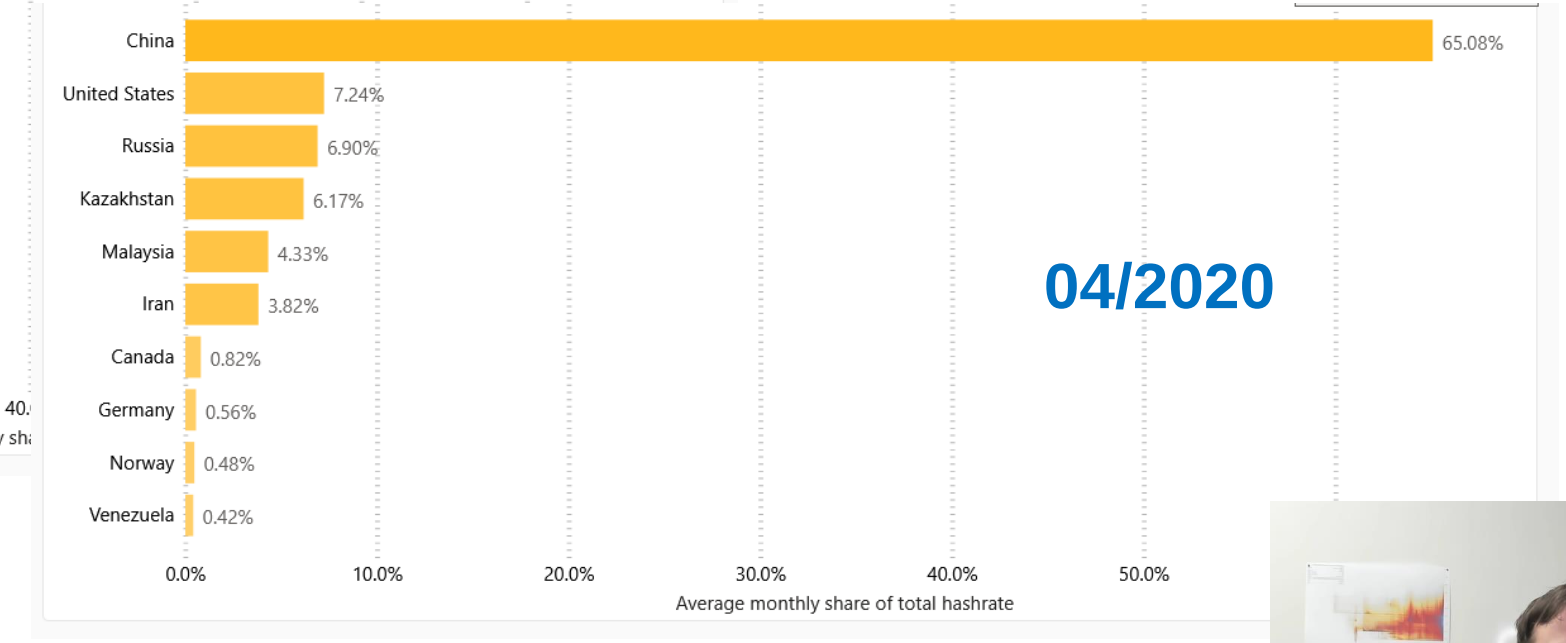
Bitcoin mining map in April 2020

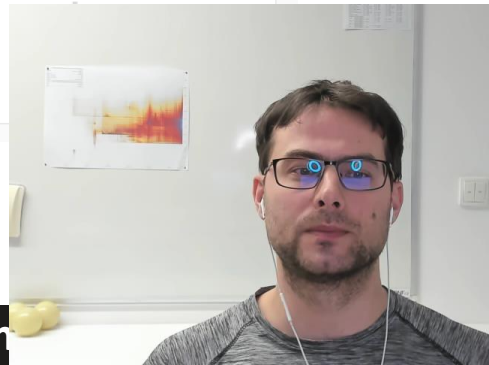




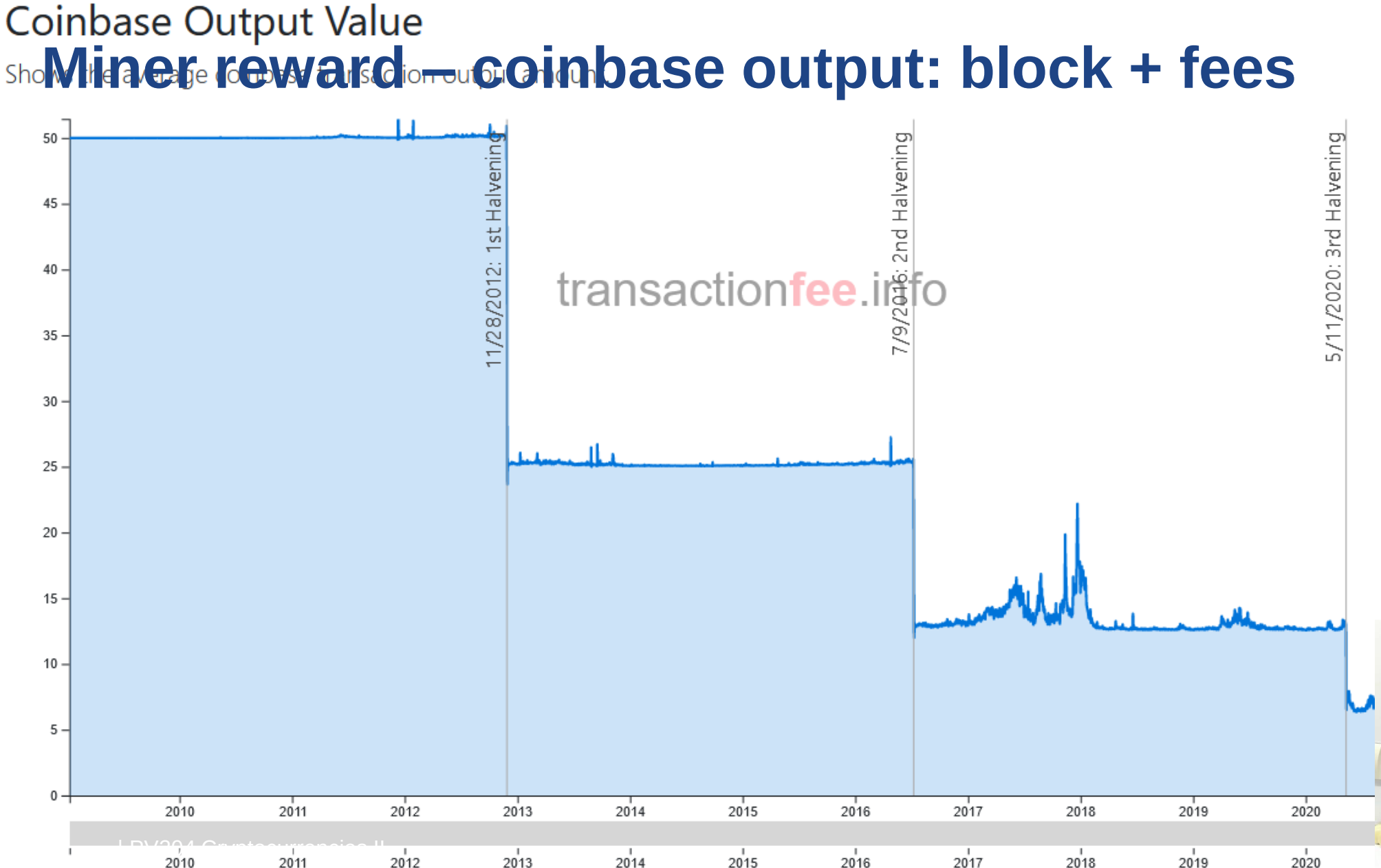
China mining dominance (09/2019→04/2020: 75.6%->65%)

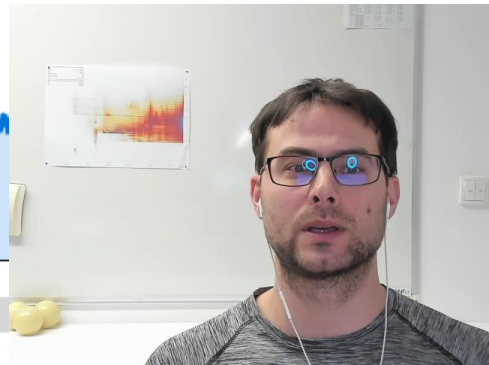




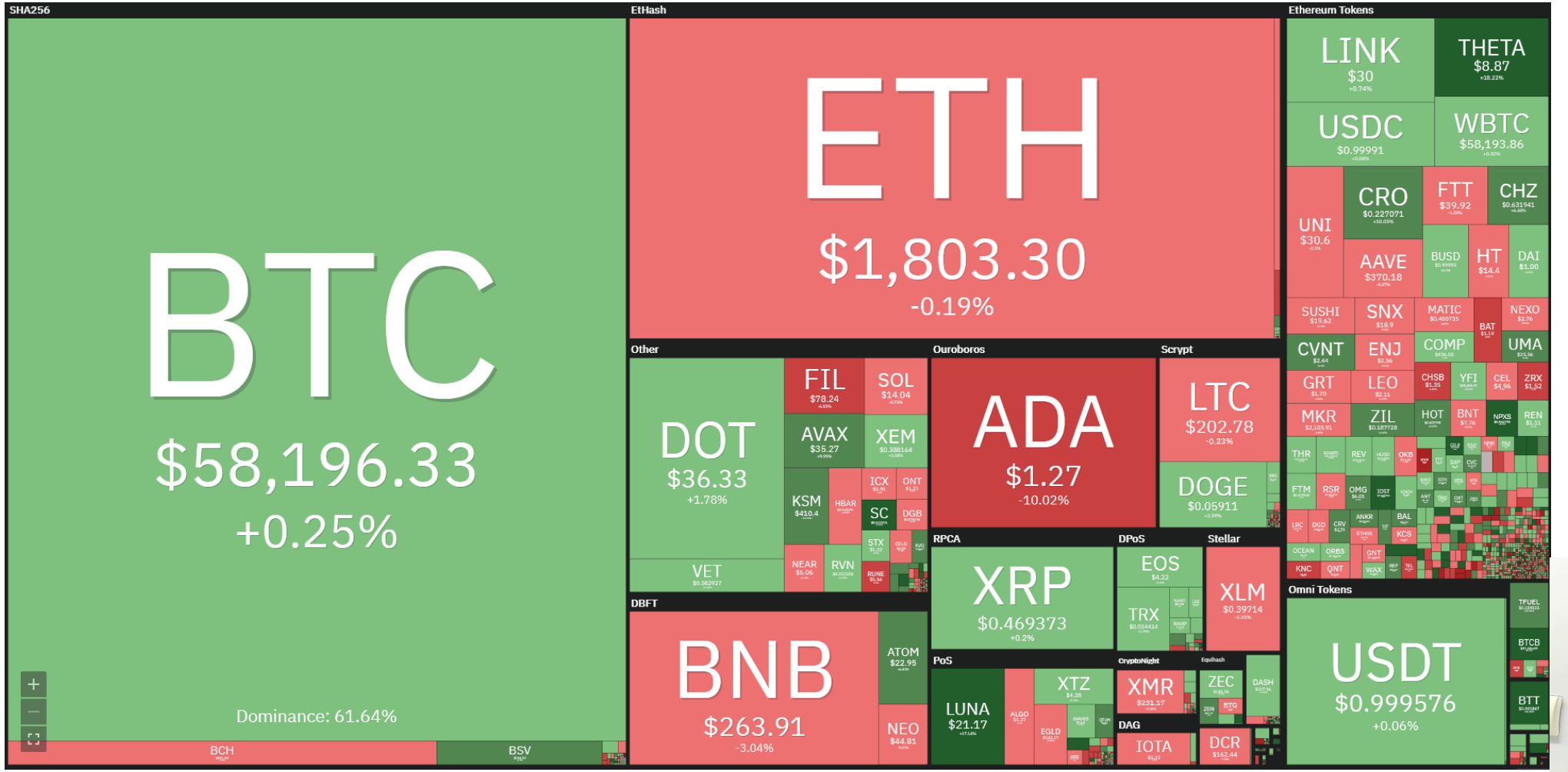


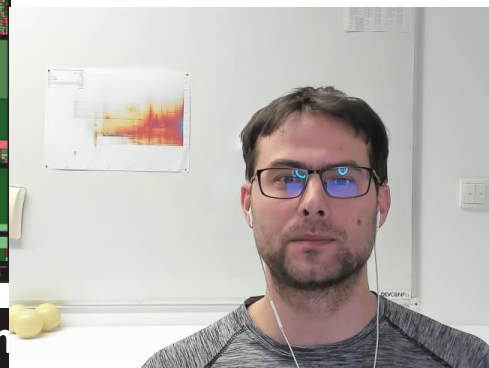
Miner reward – coinbase output: block + fees



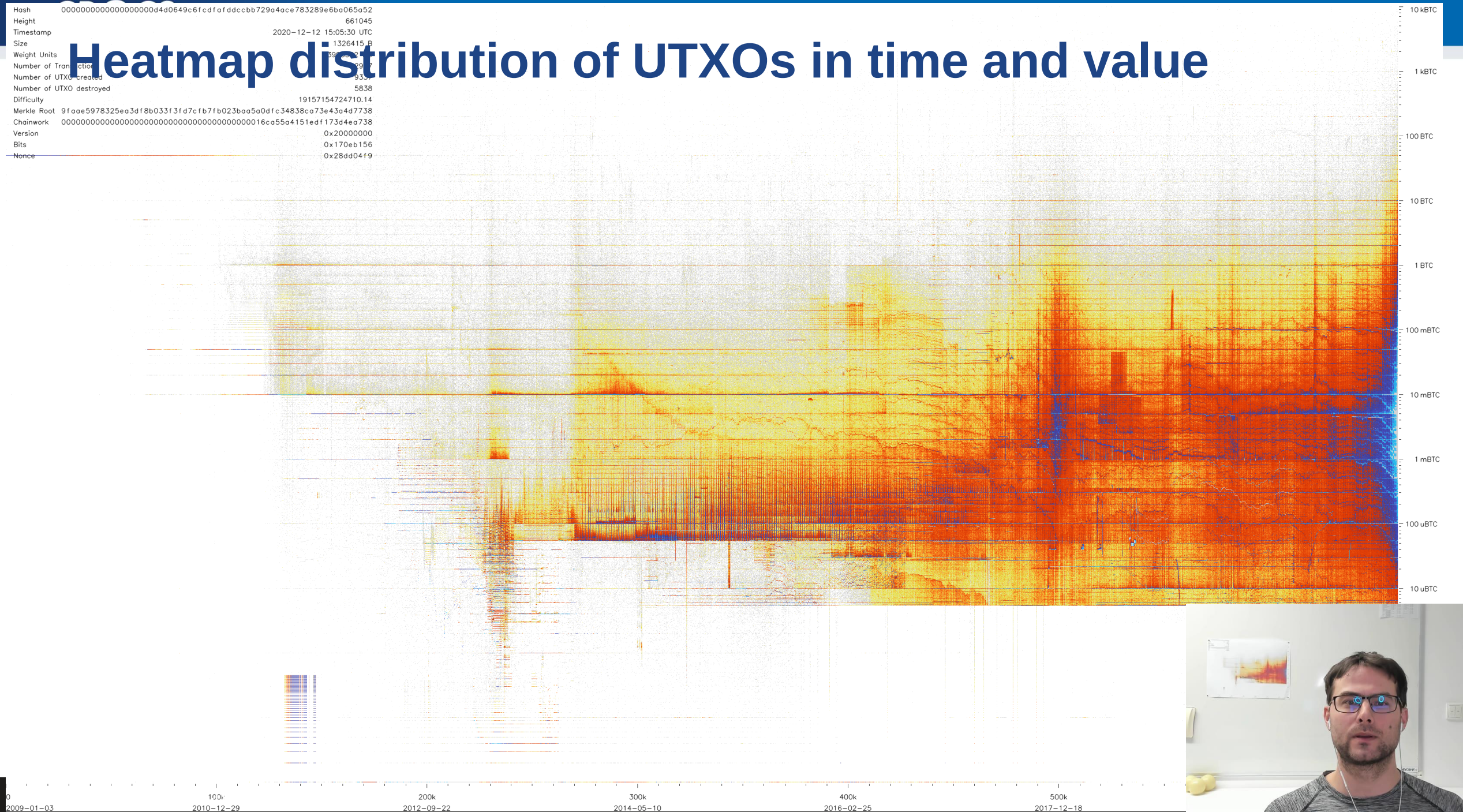


Coin mining algorithm https://coin360.com/





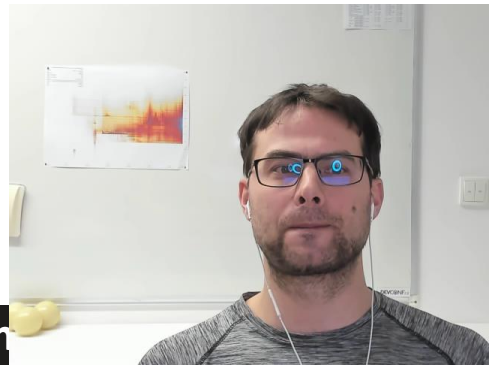
Heatmap distribution of UTXOs in time and value



Interesting stats about mined transactions

- https://forkmonitor.info/nodes/btc
- https://transactionfee.info/
- https://cryptobriefing.com/unpacking-bitcoins-recent-double-spend-event

BITCOIN PRIVACY

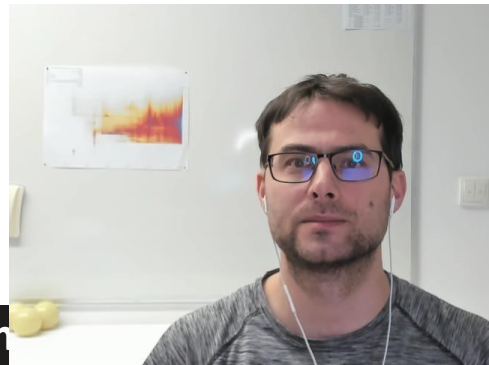


Risks

- Risk of lost coins
  - Lost wallet keys, forgotten access credentials
- Risk of stolen coins
  - Malware on computer (wallet keys), phishing/scam (recovery phrase)
  - Compromised trusted third party (exchange, web wallet…)
  - Random burglary (don’t know you have btc)
  - Targeted burglary (know you have btc), with(-out) you present
- Risk of traced coins
  - blockchain analysis, additional metadata correlation analysis (KYC/AML, scans, tx propagation, wallet peeling…)
  - Crooks, governments, wife…

Attacker models

- Blockchain-only analysis
- Malware, phishing
- Active network analysis, metadata
- Cryptographic analysis of used algorithms
- Side-channel analysis

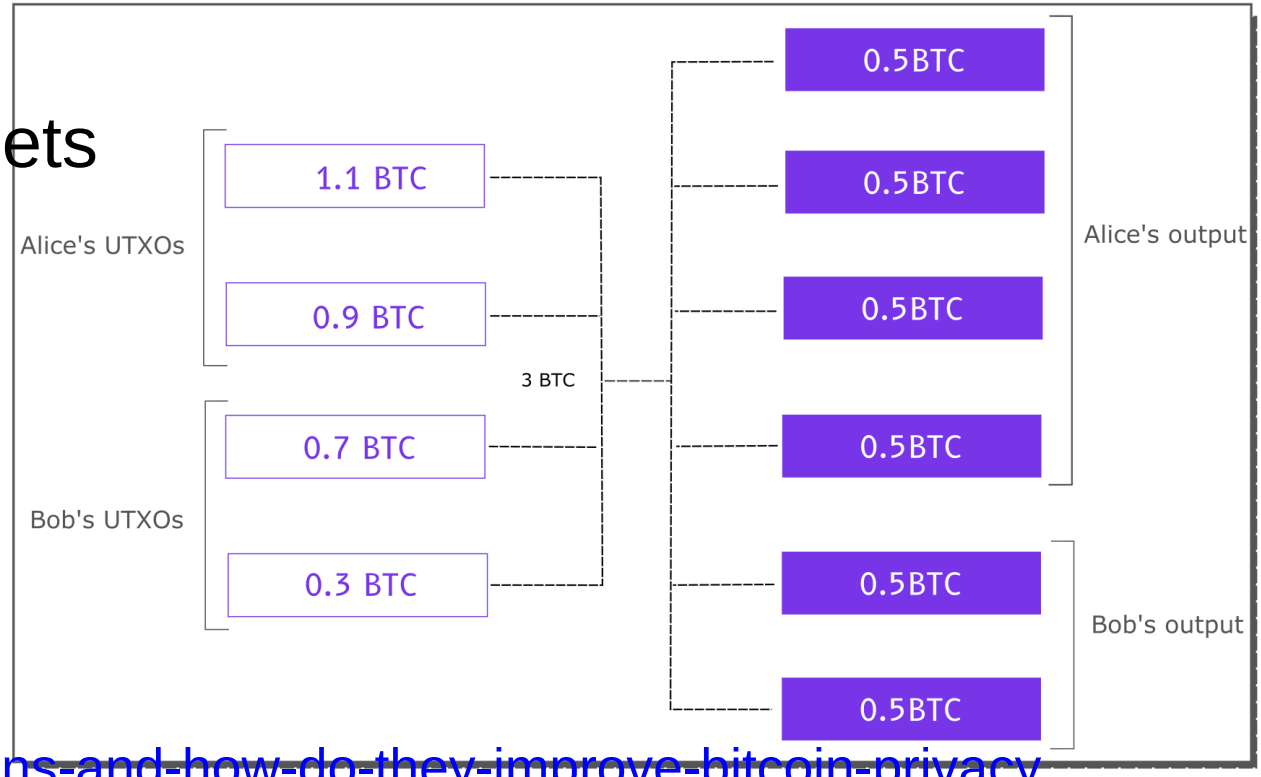


Improving privacy

- Hold your private keys (no custodial service like exchange…)
  - Cannot steal, cannot observe, cannot “vote” on your behalf
- Store private key in hardware wallet (Trezor, ColdCard, Ledger…)
  - Keys in “hot” software wallets are prone to malware attack
- Run own full node over Tor and connect your wallet to it
- Make on-chain analysis harder: https://en.bitcoin.it/wiki/Privacy
- Use manual coin selection, label coins by its origin
- Use CoinJoin, PayJoin (multiple users mix their inputs in single transaction)
- Have good opsec (no posting of own btc addresses, use Tor to broadcast tx, delink via CoinJoin after KYC…)

CoinJoin

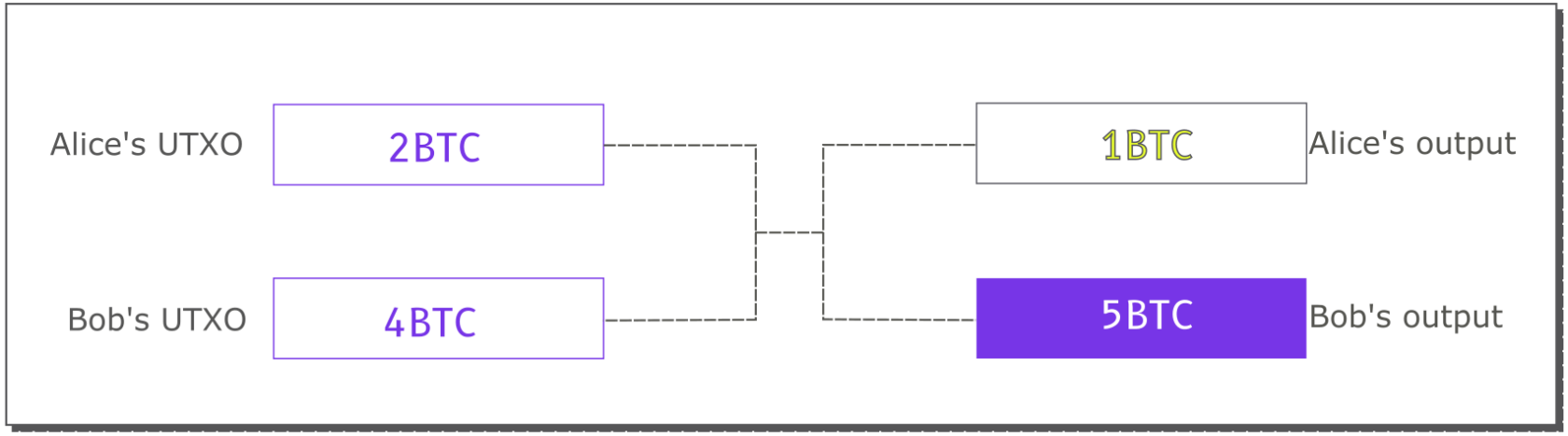
- Multiple users collaborates trustlessly in creating large transaction
- Outputs are all the same value => cannot be attributed to one of senders based on the value
- Supported by more advanced wallets
  - Wasabi wallet
  - Samurai wallet



- https://en.bitcoinwiki.org/wiki/CoinJoin
- https://cryptotesters.com/blog/what-are-coinjoins-and-how-do-they-improve-bitcoin-privacy

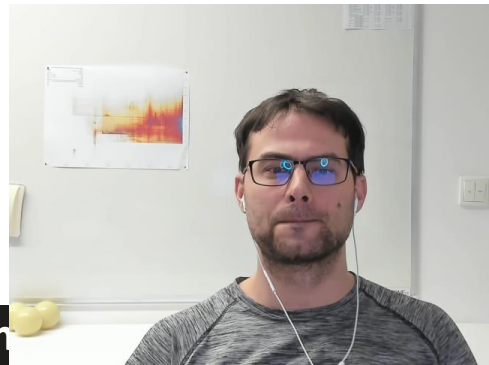
PayJoin

- PayJoin is special case of CoinJoin, but with less participants (sender, receiver) and without equal UTXO sizes
- Faster than CoinJoin, done during a normal payment



- https://cryptotesters.com/blog/what-are-coinjoins-and-how-do-they-improve-bitcoin-privacy

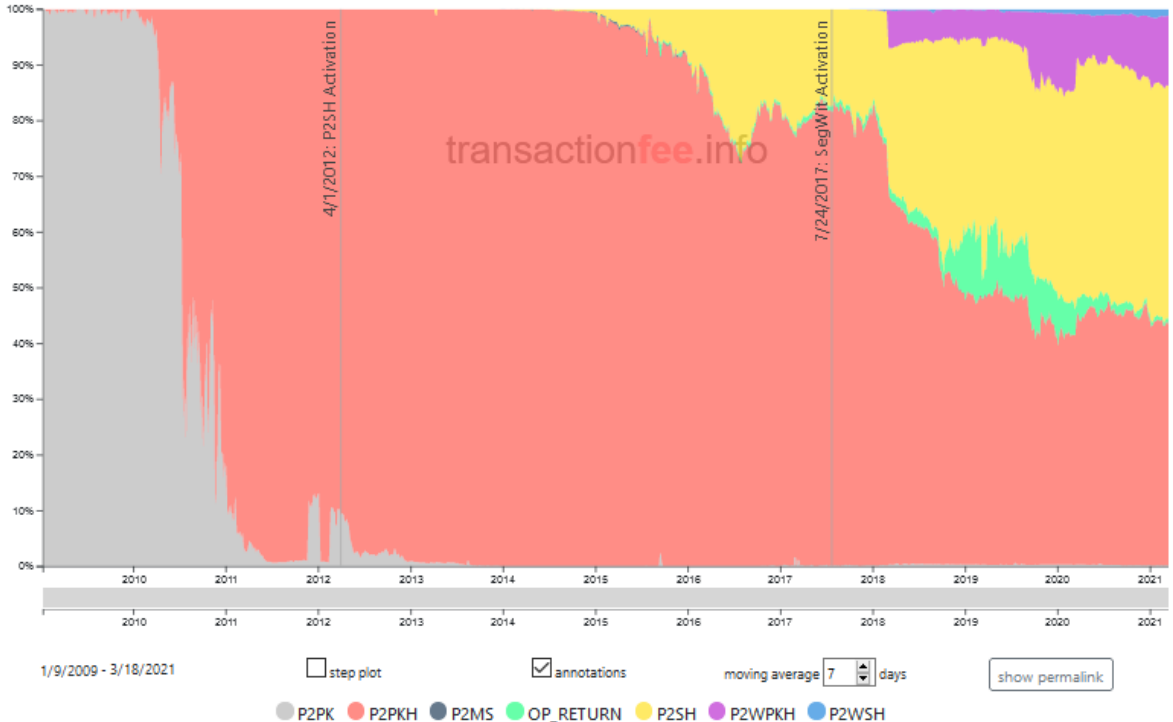
LOCK AND UNLOCK SCRIPTS



https://transactionfee.info/charts/output-type-distribution-count/

Types of receiving “addresses”

- There is no ”address” defined in Bitcoin network
- Standard patterns how to construct lock script emerged over the time
  - e.g., unlock if signature is verifiable with the public key stored in lock script (P2PK)
  - “Address” is the variable part of the lock script differing between different receivers and transactions
- *Notation warning: scriptSig (script + signature), scriptPubKey (initial meaning script + public key == P2PK)*
- Well-known standard types of lock scripts
  - Pay-to-public-key (P2PK)
  - Pay-to-public-key-hash (P2PKH, starts with 1)
  - Pay-to-script-hash (P2SH, BIP16)
  - OP_RETURN (any data 80B)
  - Native Pay-to-witness-script-hash (P2WSH, starts with 3)
  - P2WSH-nested-in-P2SH
  - P2SH-P2WPKH, P2SH-P2WSH
  - Native P2WPK, P2WSH (Bech32, starts with bc1)



Pay-to-public-key (P2PK)

- Lock script contains direct value of public key and instructions to push signature and verify with the public key
- Used initially by Satoshi and others, now infrequent
- Disadvantage: if practical dlog attack against secp256k1 is found, private key can be computed

https://nioctib.tech/#/transaction/f2f398dace996dab12e0cfb02fb0b59de0ef0398be393d90ebc8ab397550370b
https://nioctib.tech/#/transaction/feff813f13340060f641c11ab1307bb1b8cabcdcc3af1aed8a089e38c8407aef

P2PKH - script execution (https://nioctib.tech/)

Paying from

3CpfD1gBBdNW7orErj3YyNNSVpzndZ9aP9
9.8697071 BTC - Transaction output 1
ScriptSig - P2SH
0x002087a59be084440ce7b1ccc965cb53cee54fdc059855107f5c986f80c7a60db3df
Interpret or debug

To

1B9DXkcnXbVXEEpRpcXzfhWe8uK16XvbMr
0.05149519 BTC - Transaction
ScriptPubKey - P2PKH
OP_DUP OP_HASH160
0x6f3f0b93b060ea9c0d76989c9747c9b6cfad617d OP_EQUALVERIFY
OP_CHECKSIG

3CpfD1gBBdNW7orErj3YyNNSVpzndZ9aP9
9.81803047 BTC - Transaction
ScriptPubKey - P2SH
OP_HASH160 0x7a1b6b1dbd9840fcf590e13a8a6e2ce6d55ecb89
OP_EQUAL

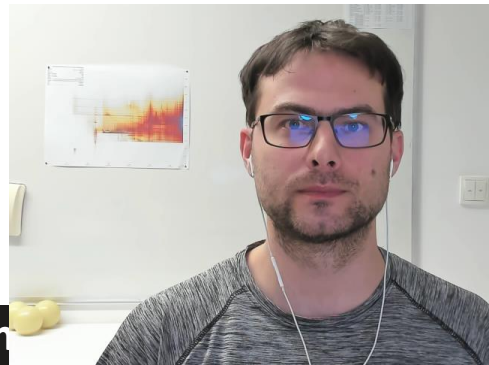
Paying from

1B9DXkcnXbVXEEpRpcXzfhWe8uK16XvbMr
0.05149519 BTC - Transaction
ScriptSig - P2PKH
0x304402205c5876144bf491eb6aece2625cbc3049819f35094e8feaf808399de0c29b593d022048267261596dcdb8a49659f0a9c74f2a423d6c7bef02058b56a8b90fb39e8ff901
0x02b621afa86afdb74d874e876413cf199833f4a5f68e10335134876eebe29bbe6d
Interpret or debug

To

14Z9hhyEbccWepjruEnoSvQvuSjd7QVN9Y
0.00064007 BTC - Transaction
ScriptPubKey - P2PKH
OP_DUP OP_HASH160
0x26fcf3b9cc3e0d2fc51fc69e58b63b41e2094f44 OP_EQUALVERIFY
OP_CHECKSIG

18hgAeKFH4L93DR8nGL9LHx9yWntnCjbW8
0.05 BTC - Transaction
ScriptPubKey - P2PKH
OP_DUP OP_HASH160
0x547a369b70f0241ebd1e8288397dd34f2c11ac6b OP_EQUALVERIFY
OP_CHECKSIG



Stack

Script

0x304402205c5876144bf491eb6aece2625cbc3049819f35094e8feaf808
399de0c29b593d022048267261596dcdb8a49659f0a9c74f2a423d6c7bef
02058b56a8b90fb39e8ff901

0x02b621afa86afdb74d874e876413cf199833f4a5f68e10335134876eebe
29bbe6d

OP_DUP OP_HASH160

0x6f3f0b93b060ea9c0d76989c9747c9b6cfad617d OP_EQUALVERIFY

OP_CHECKSIG

Executing **Script Sig** [1]

Stack

0x304402205c5876144bf491eb6aece2625cbc3049819f35094e8feaf808
399de0c29b593d022048267261596dcdb8a49659f0a9c74f2a423d6c7bef
02058b56a8b90fb39e8ff901

Script

0x02b621afa86afdb74d874e876413cf199833f4a5f68e10335134876eebe
29bbe6d

OP_DUP OP_HASH160

0x6f3f0b93b060ea9c0d76989c9747c9b6cfad617d OP_EQUALVERIFY

OP_CHECKSIG

Executing **Script Sig** [2]

Stack

0x02b621afa86afdb74d874e876413cf199833f4a5f68e10335134876eebe
29bbe6d

0x304402205c5876144bf491eb6aece2625cbc3049819f35094e8feaf808
399de0c29b593d022048267261596dcdb8a49659f0a9c74f2a423d6c7bef
02058b56a8b90fb39e8ff901

Script

OP_DUP OP_HASH160

0x6f3f0b93b060ea9c0d76989c9747c9b6cfad617d OP_EQUALVERIFY

OP_CHECKSIG

Executing **Script PubKey** [3]

Stack

0x02b621afa86afdb74d874e876413cf199833f4a5f68e10335134876eebe
29bbe6d

0x304402205c5876144bf491eb6aece2625cbc3049819f35094e8feaf808
399de0c29b593d022048267261596dcdb8a49659f0a9c74f2a423d6c7bef
02058b56a8b90fb39e8ff901

Script

OP_DUP OP_HASH160

0x6f3f0b93b060ea9c0d76989c9747c9b6cfad617d OP_EQUALVERIFY

OP_CHECKSIG

Executing **Script PubKey** [4]

Stack

0x02b621afa86afdb74d874e876413cf199833f4a5f68e10335134876eebe
29bbe6d

0x02b621afa86afdb74d874e876413cf199833f4a5f68e10335134876eebe
29bbe6d

0x304402205c5876144bf491eb6aece2625cbc3049819f35094e8feaf808
399de0c29b593d022048267261596dcdb8a49659f0a9c74f2a423d6c7bef
02058b56a8b90fb39e8ff901

Script

OP_HASH160 0x6f3f0b93b060ea9c0d76989c9747c9b6cfad617d

OP_EQUALVERIFY OP_CHECKSIG

Executing **Script PubKey** [5]

Stack

0x6f3f0b93b060ea9c0d76989c9747c9b6cfad617d

0x02b621afa86afdb74d874e876413cf199833f4a5f68e10335134876eebe
29bbe6d

0x304402205c5876144bf491eb6aece2625cbc3049819f35094e8feaf808
399de0c29b593d022048267261596dcdb8a49659f0a9c74f2a423d6c7bef
02058b56a8b90fb39e8ff901

Script

0x6f3f0b93b060ea9c0d76989c9747c9b6cfad617d OP_EQUALVERIFY

OP_CHECKSIG

Executing **Script PubKey** [6]

Stack

0x6f3f0b93b060ea9c0d76989c9747c9b6cfad617d

0x6f3f0b93b060ea9c0d76989c9747c9b6cfad617d

0x02b621afa86afdb74d874e876413cf199833f4a5f68e1
29bbe6d

0x304402205c5876144bf491eb6aece2625cbc3049819f3
399de0c29b593d022048267261596dcdb8a49659f0a9c7
02058b56a8b90fb39e8ff901

Script

OP_EQUALVERIFY OP_CHECKSIG

Executing **Script PubKey** [7]

Stack

1

0x02b621afa86afdb74d874e876413cf199833f4a5f68e10335134876eebe
29bbe6d

0x304402205c5876144bf491eb6aece2625cbc3049819f35094e8feaf808
399de0c29b593d022048267261596dcdb8a49659f0a9c74f2a423d6c7bef
02058b56a8b90fb39e8ff901

Script

OP_VERIFY OP_CHECKSIG

Executing **Script PubKey** [8]

Stack

0x02b621afa86afdb74d874e876413cf199833f4a5f68e10335134876eebe
29bbe6d

0x304402205c5876144bf491eb6aece2625cbc3049819f35094e8feaf808
399de0c29b593d022048267261596dcdb8a49659f0a9c74f2a423d6c7bef
02058b56a8b90fb39e8ff901

Script

OP_CHECKSIG

Executing **Script PubKey** [9]

Stack

1

Script

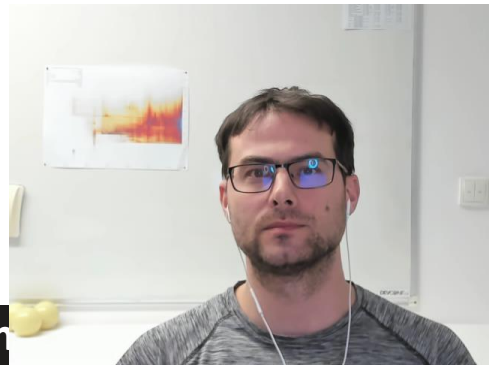
Execution Succeeded

Stack

1

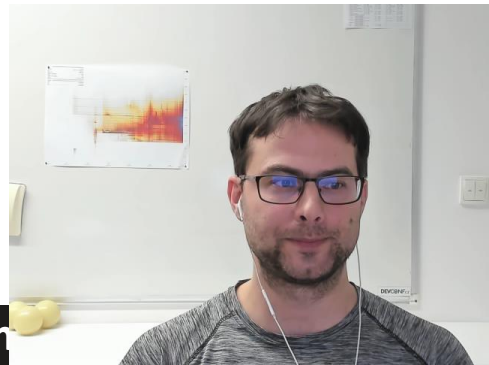
Script

THRESHOLD SIGNATURES VS. MULTISIG VS. MULTI-PARTY COMPUTATION



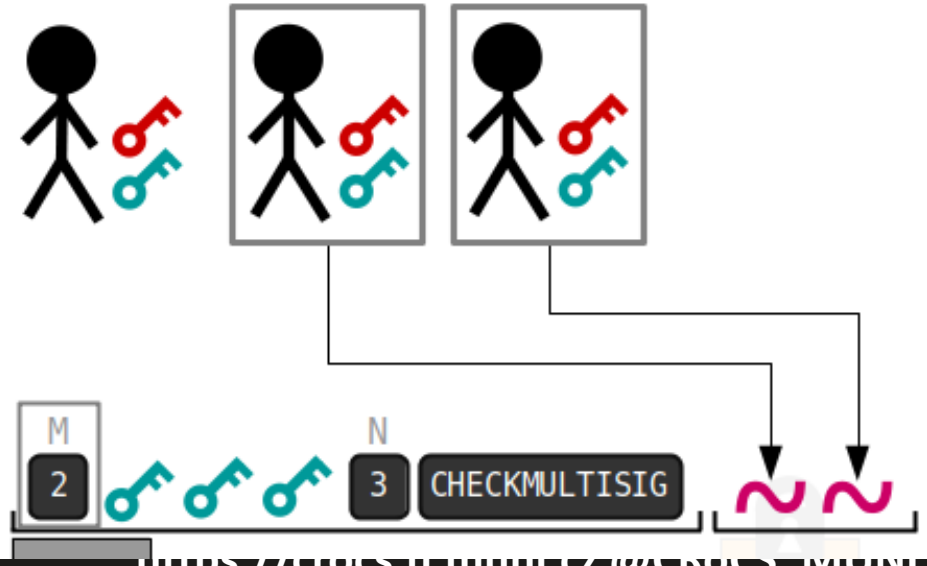
Shamir secret sharing scheme

- Private key is recovered from multiple shares
  - Then used at single place
  - An attacker can compromise private key after its recovery from shares
- Network is unaware of key split, single public key used in lock script
- Can be used to backup wallet seed (e.g., Trezor wallet)
  - n-out-of-n or k-out-of-n



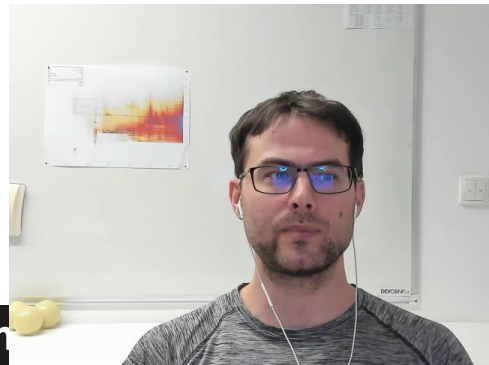
Multisignatures

- Lock script constructed to require multiple signatures (OP_CHECKMULTISIG)
  – transaction valid only if multiple signers provide signatures for unlock script
- n-out-of-n or k-out-of-n, https://en.bitcoin.it/wiki/Multisignature
- P2MS, P2MS wrapped in P2SH
  – https://learnmeabitcoin.com/technical/p2ms



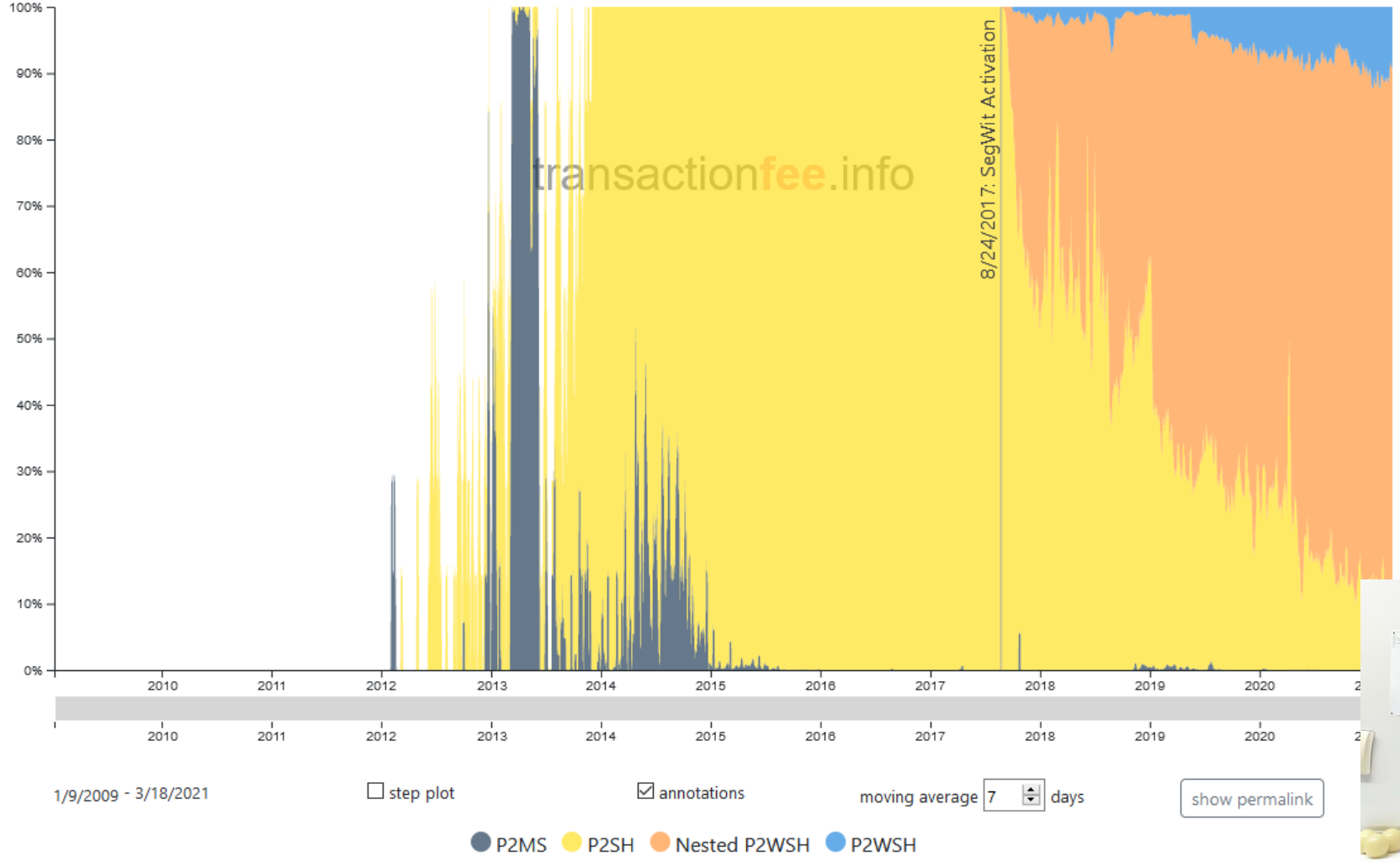
Secure multi-party computation (MPC)

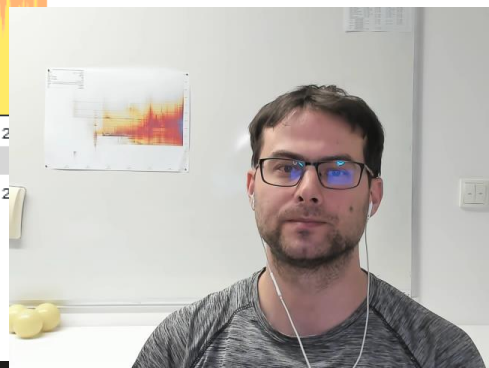
- Single signature computed using multiple separated signers
  - Each signer has own private key
  - An attacker must comprise more than one entity
- Communication between signers
  - During initial key generation
  - Optionally during signing
- Legacy compatible schemes
  - 2-party ECDSA, n-out-of-n or k-out-of-n ECDSA (only since 2016)
- Taproot-compatible schemes (not yet activated)
  - Schorr signatures, MuSig2
- https://academy.binance.com/en/articles/threshold-signatures-explained



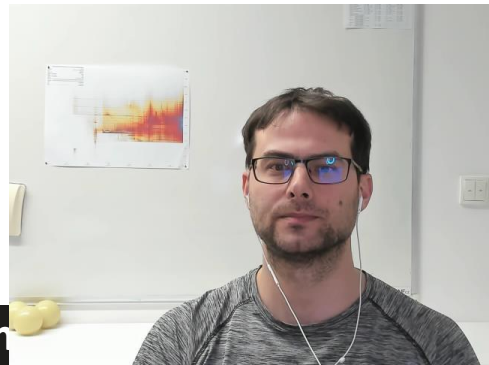
Frequency of different multisignature scripts

Shows the distribution of multisig spends for each input type per day.





ALTCOINS

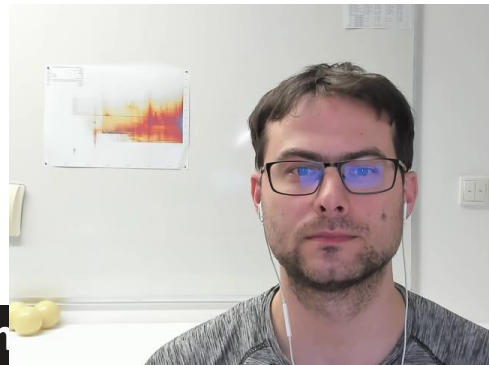


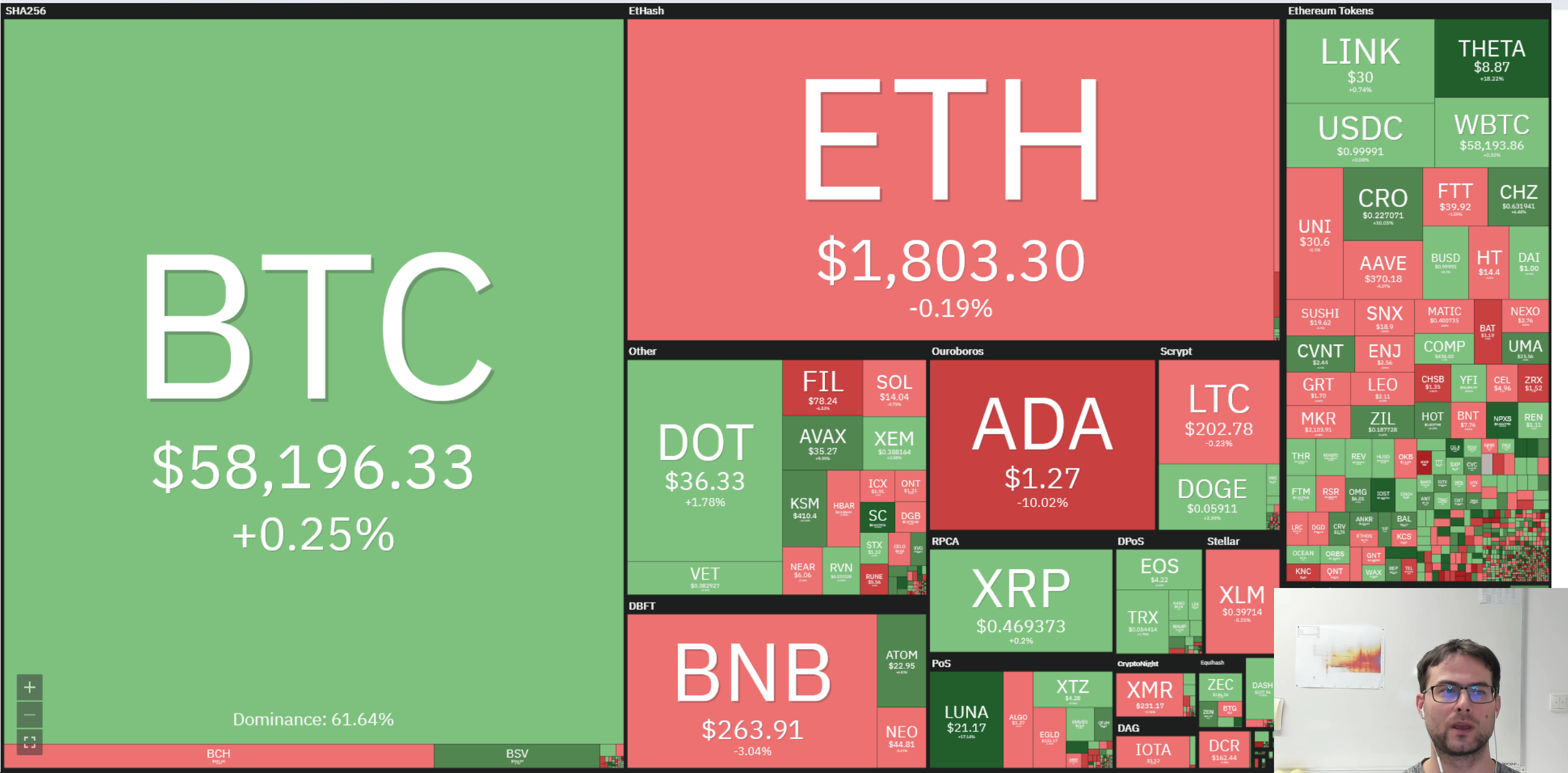
Why other cryptocurrencies (altcoins)

- Why something else than Bitcoin?

1. Cost of sending transaction
   - Order of dollars at the moment (for every transfer)
2. Time to confirm transaction (+ limited block size)
   - 4 blocks inside chain commonly required, ~10 minutes per block => ~40 min
3. Traceability of transactions
   - Source, destination and amount is on public ledger
4. Limited scripting language
   - For more complicated smart contracts
5. Specialized mining equipment required
   - Bitcoin mining only possible via ASICs => may cause centralization
   - Proof of Work is energy intensive

- …

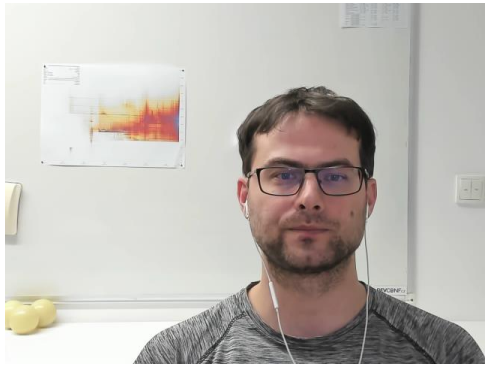




https://crocs.fi.m

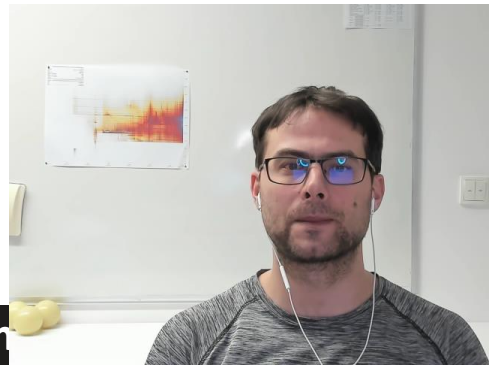
Other cryptocurrencies (altcoins)

- Copycats (huge number of them)
  - Take Bitcoin's source code, change name and basic params (mining alg, time and size of block…)
  - E.g., Litecoin
- Bitcoin-style, but adding some distinct features
  - Ethereum: Turing-complete scripting for smart contracts, (EthHash mining alg), Eth2.0 move to PoS
  - Zcash: zero-knowledge proof for sender/receiver/amount (shielded transactions), aim to have GPU-friendly mining (Equihash, large memory required)
  - Monero: private transactions via mixing (Ring Confidential Transactions, CryptoNote)
- More traditional styles (Ripple, Stellar…)
  - Somewhat decentralized network of verification nodes (=> faster and cheaper txs)
  - Typically, less privacy and overall resilience against central control
- Stable coins (USDT, USDC…)
  - Idea: digital equivalent to real dollars stored in "safe"
  - New 1 USDT is created when someone deposits $1 to company, destroyed when $1 is cashed back



Tokens, ICO, DeFi, CBDC…

- Initial Coin Offerings (ICO), boom in 2017
  - Kind of crowdfunding campaign - often via Ethereum smart contracts, ERC-20 contracts
  - Frequently scam, frequently large pre-allocation to founders and investors
- Decentralized Finance (DeFi)
  - Smart contract with defined (financial-related) behavior – e.g., lending…
- Non-fungible tokens
  - Representation of physical item on the blockchain
  - Allows to pass ownership by “sending” token to another person
  - Possible on almost any chain (colored coins at Bitcoin)
  - Some chains build for it intentionally
- Central bank digital currency (CBDC)
  - Permissioned ledger by central banks



Ethereum basics

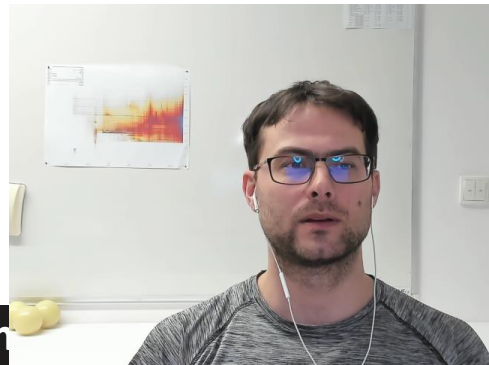
- Basic idea: Make script Turing complete
  - Executed by Ethereum Virtual Machine
  - 256-bit register stack
- Ether (ETH) is native currency rewarded to miners (PoW, Ethash)
- Gas is transaction fee payed to miners for new tx
- Block time is 13 seconds on average
  - But Difficulty bomb to force periodic protocol updates
- Two types of accounts: users and contracts
- See some example eth scripts https://remix.ethereum.org/
- Mastering Ethereum, A. Antonopoulos, https://github.com/ethereumbook/ethereumbook

```python
// SPDX-License-Identifier: GPL-3.0

pragma solidity >=0.7.0 <0.8.0;

/**
 * @title Owner
 * @dev Set & change owner
 */
contract Owner {

    address private owner;

    // event for EVM logging
    event OwnerSet(address indexed oldOwner, address indexed newOwner);

    // modifier to check if caller is owner
    modifier isOwner() {
        // If the first argument of 'require' evaluates to 'false', execution terminates and all
        // changes to the state and to Ether balances are reverted.
        // This used to consume all gas in old EVM versions, but not anymore.
        // It is often a good idea to use 'require' to check if functions are called correctly.
        // As a second argument, you can also provide an explanation about what went wrong.
        require(msg.sender == owner, "Caller is not owner");
        _;
    }

    /**
     * @dev Set contract deployer as owner
     */
    constructor() {
        owner = msg.sender; // 'msg.sender' is sender of current call, contract deployer for a constructor
        emit OwnerSet(address(0), owner);
    }

    /**
     * @dev Change owner
     * @param newOwner address of new owner
     */
    function changeOwner(address newOwner) public isOwner {
        emit OwnerSet(owner, newOwner);
        owner = newOwner;
    }

    /**
     * @dev Return owner address
     * @return address of owner
     */
    function getOwner() external view returns (address) {
        return owner;
    }
}
```

ERC-20 tokens

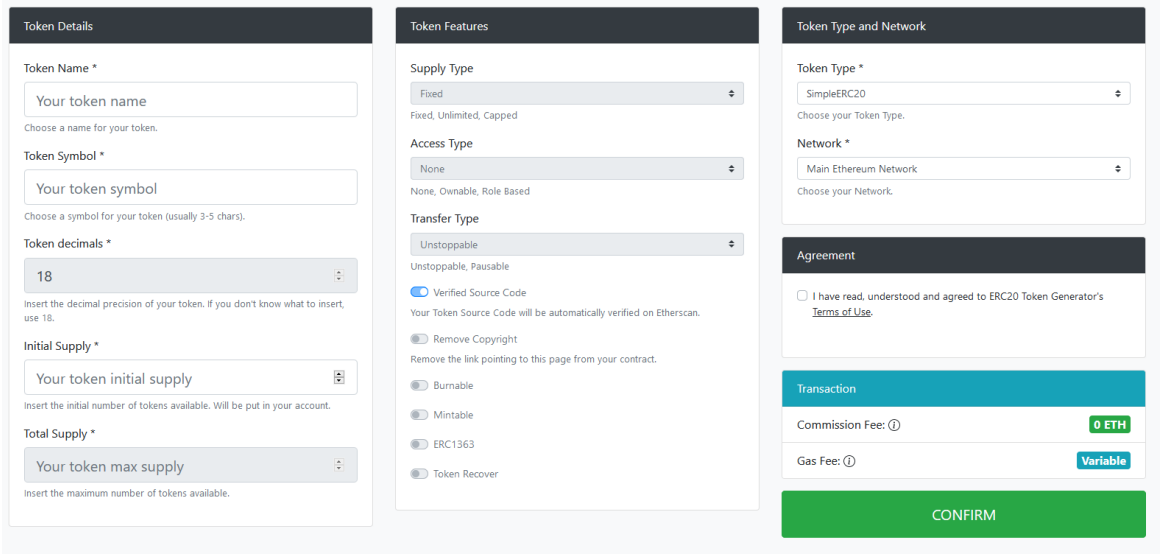
- Defined in EIP20 (Eth. Improvements Proposals):
  - https://ethereum.org/en/developers/docs/standards/tokens/erc-20/
- API for tokens within Smart Contracts
  - template contract implementations exists
    - https://academy.binance.com/en/articles/an-introduction-to-erc-20-tokens
  - you need to have ETH on your balance to send/exchange ERC20 ETH tokens (for GAS)
  - to move ERC-20 tokens, user creates and send (ethereum) transaction to the contract asking it to allocate some of the balance elsewhere
- No sending of ether, but Gas required for inclusion of transaction with script or interaction with script into blockchain

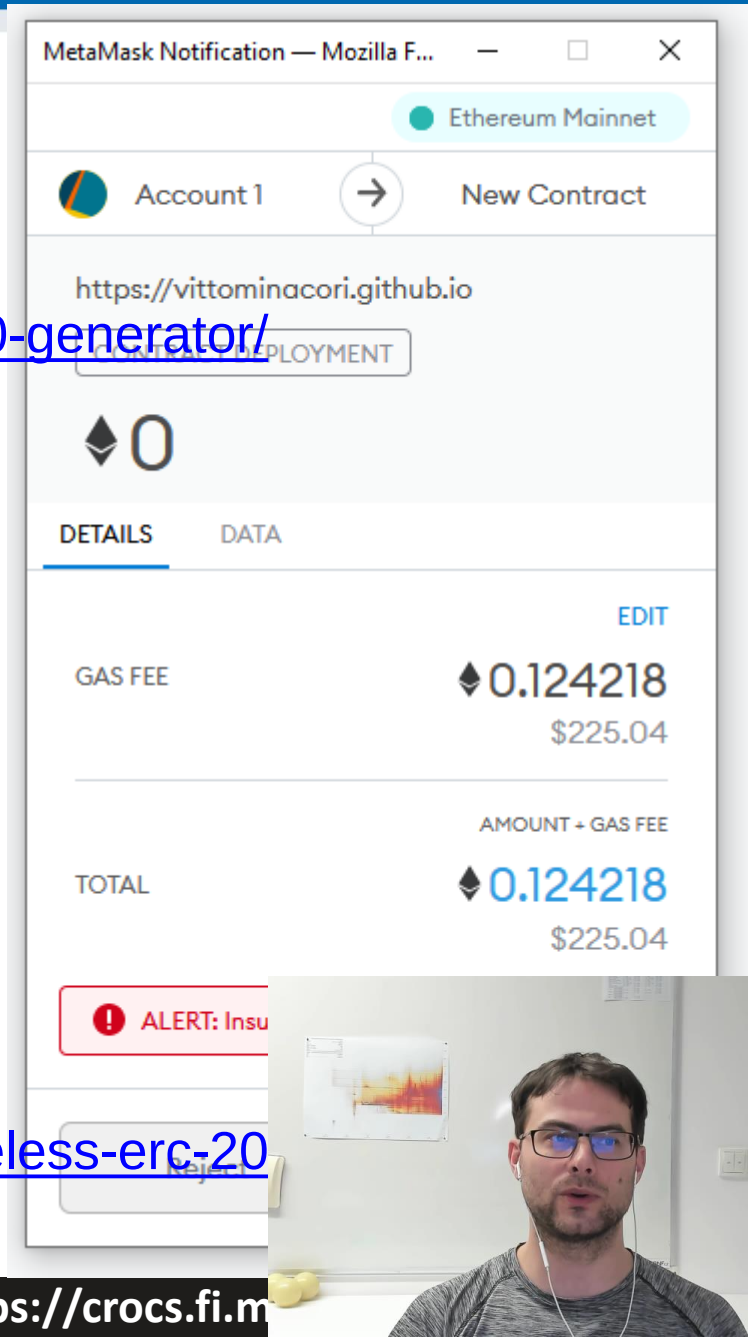
STARTING NEW COIN



Create own ERC-20 token

- Create own ERC-20 token: https://vittominacori.github.io/erc20-generator/

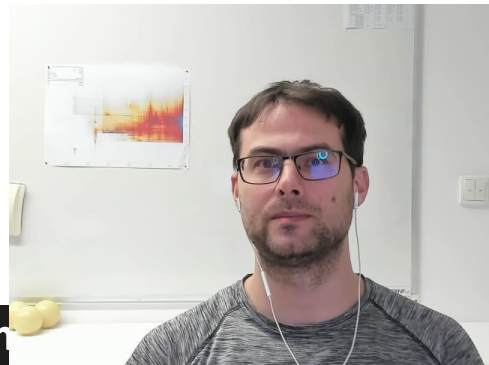




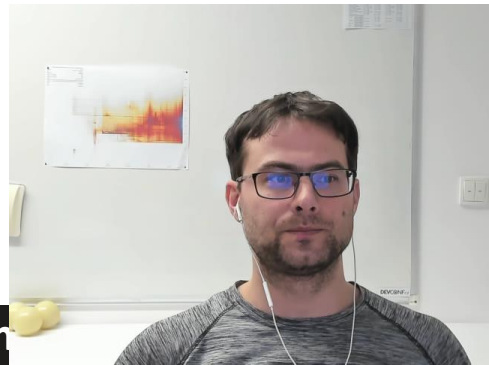
- As a result, creating token with no value is very easy
  - https://medium.com/blocktoken/how-to-launch-your-very-own-useless-erc-20-token-cfdb4100fc1d

Starting new cryptocoin?

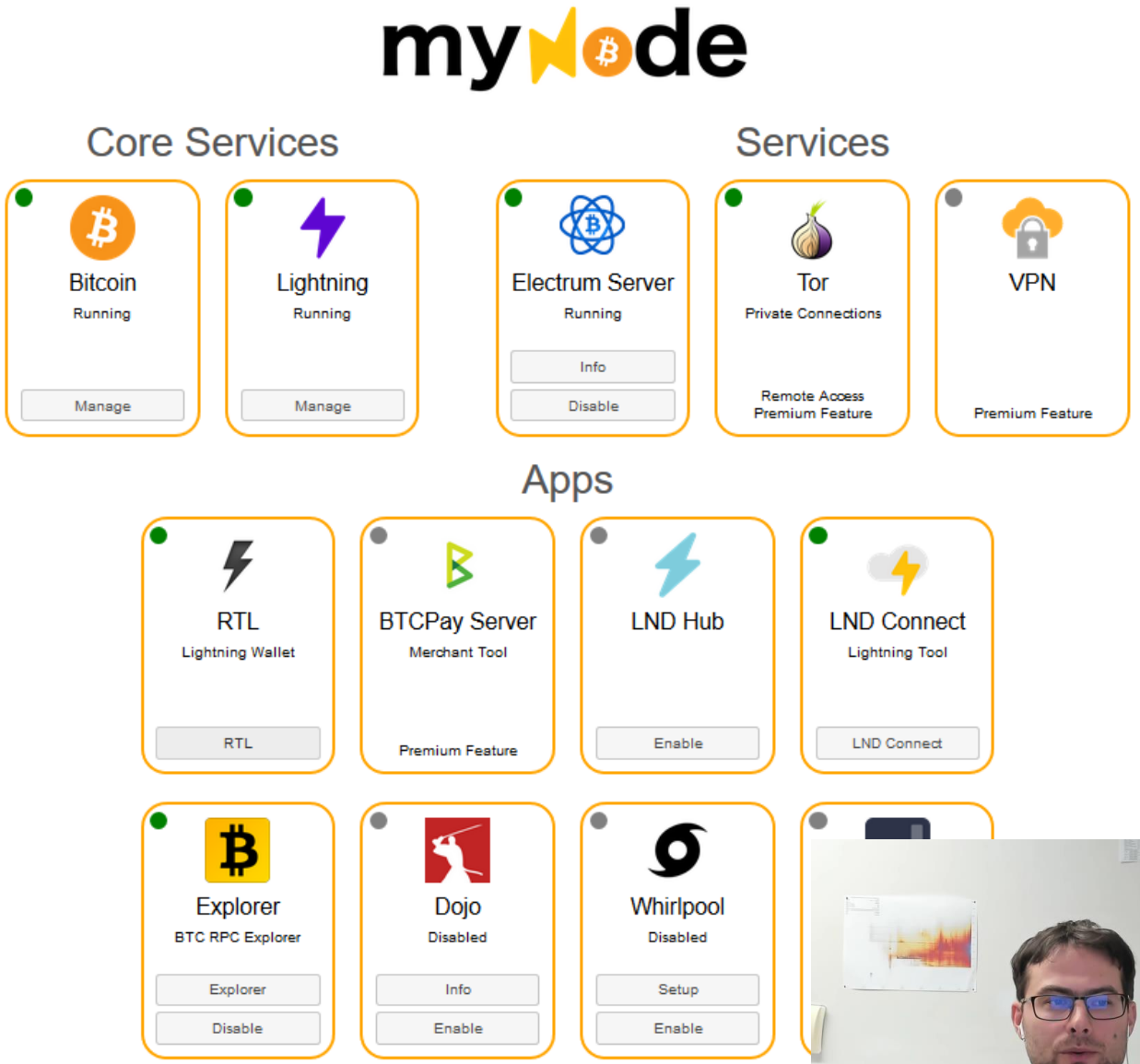
- Own chain or atop existing (e.g., ERC-20)?
- Consensus algorithm, cryptography used (e.g., ECDSA vs. Ed25519)
- Parameters of blockchain (fixed size vs. larger vs. flexible)
- Monetary policy
  - Total coins cap (fixed cap, fixed inflation, variable, stablecoins)
  - Starting conditions: bitcoin-like, premine, hidden premine, fixed mining fraction for development foundation…
- Community (serious vs. friendly), promotions
- Level of centralization
  - also influenced by other parameters – size of chain, type of consensus…
- Attitude towards hardforks vs. softforks (fixed policy vs. changing)
- Transactions on-chain or support for second-layer networks?

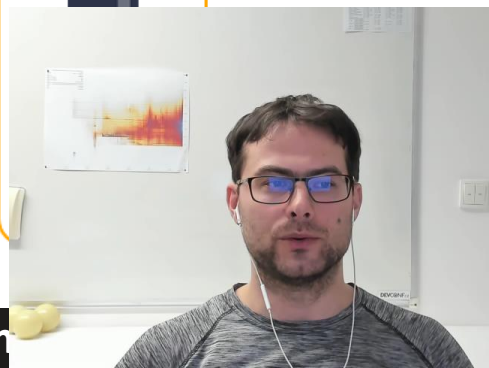


RUNNING OWN FULL NODE

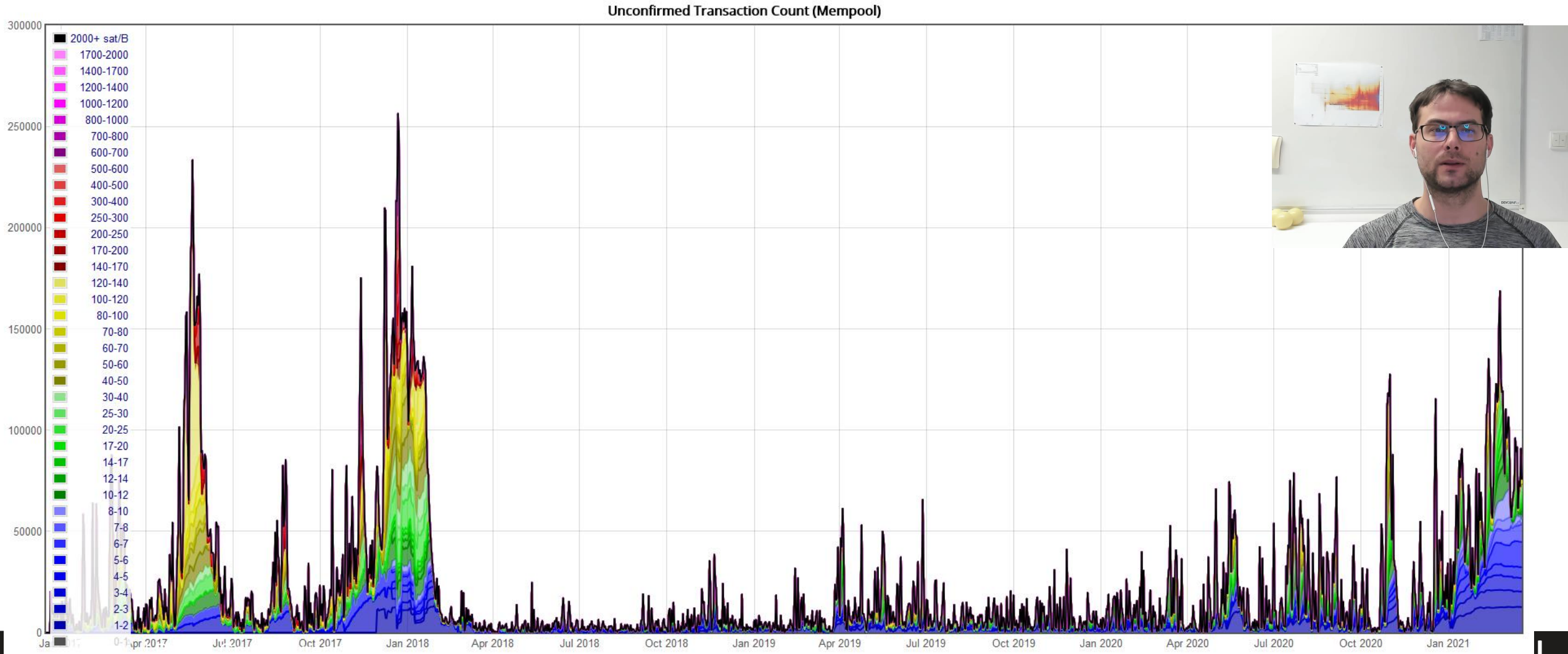


https://mynodebtc.com





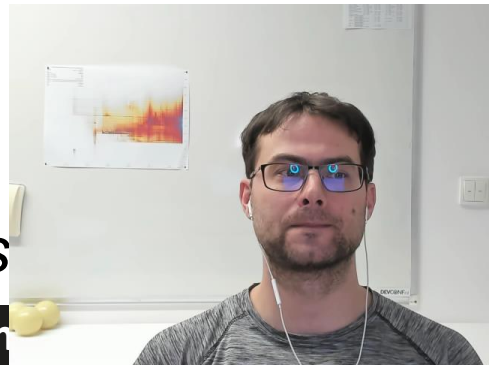
Mempool statistics https://jochen-hoenicke.de/queue



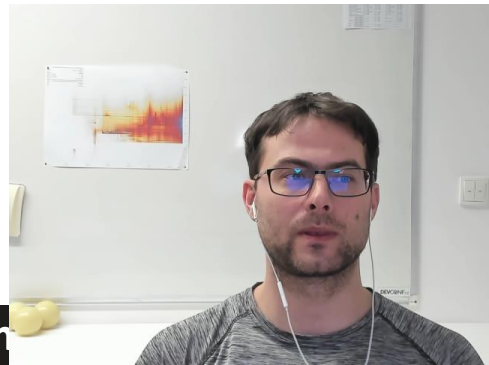
Operating own Bitcoin full node with Lighting

- Download presync part of blockchain from other mynodes (2 days)
- Download the rest of blocks from Bitcoin P2P network (1-2 days)
- Enable Lighting, create new wallet, send some sats to it (on-chain)
- Download Lighting wallet (e.g., BlueWallet, Zap)
- Pair Lighting wallet with your node
- Open channel to some other node
  - E.g., Lightning Node Suggestions at https://store.blockstream.com/
  - Opening channel performs one on-chain transaction

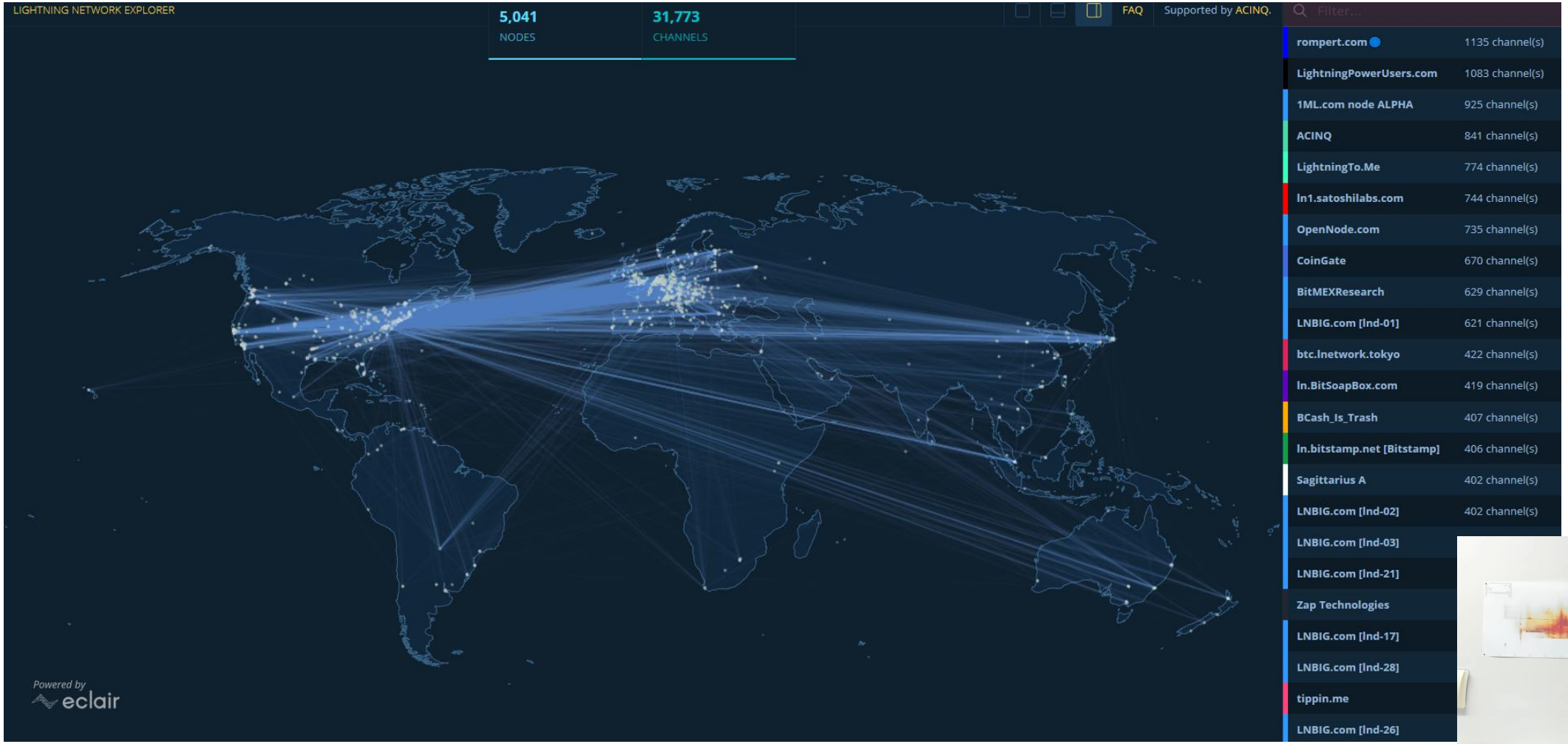
- *Analyze all other options in mynodebtc web GUI!*
- *Enable Electrum Server, Enable BTC RPC Explorer, Browse trans*



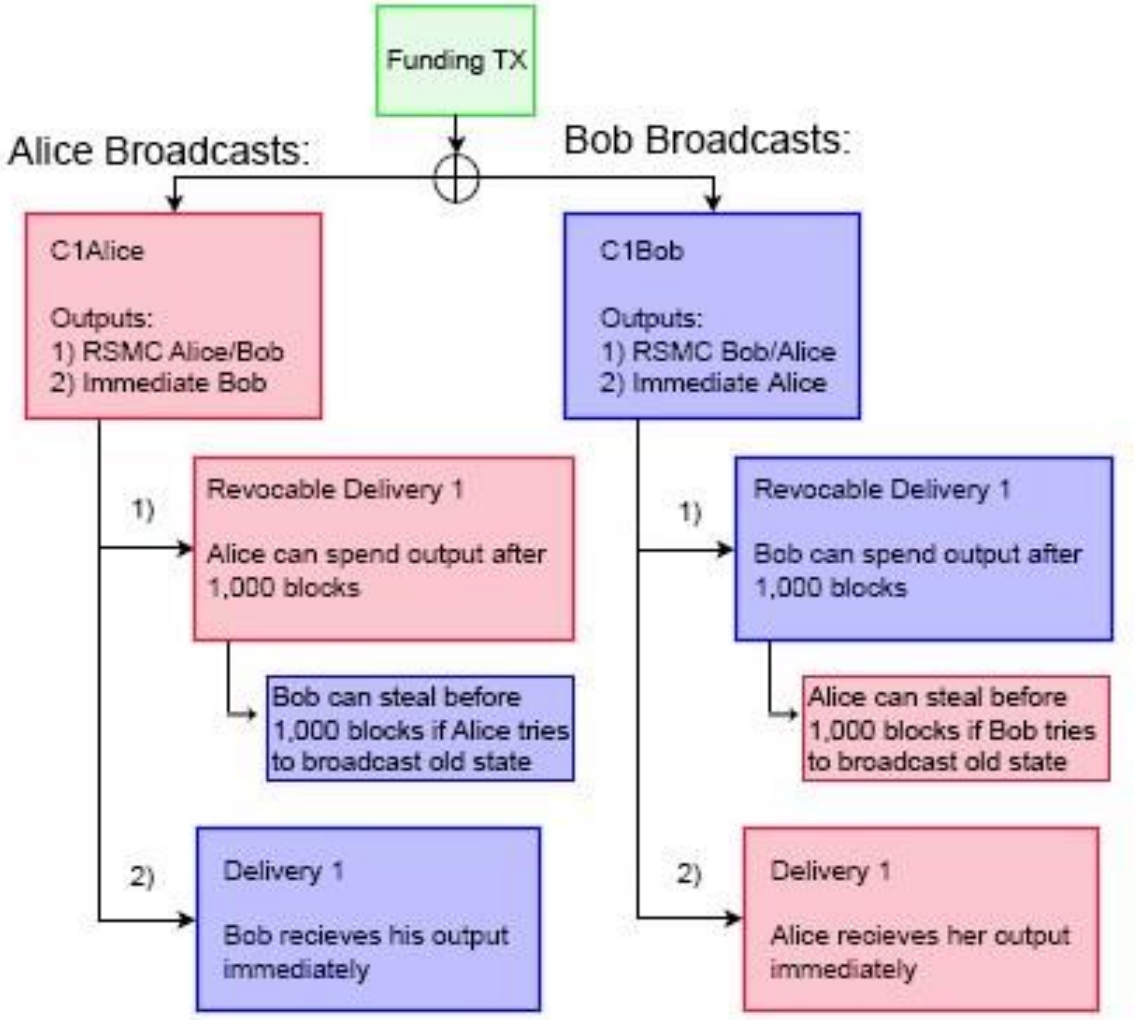
IF YOU LIKE TO DIG DEEPER (AND LIGHTER)

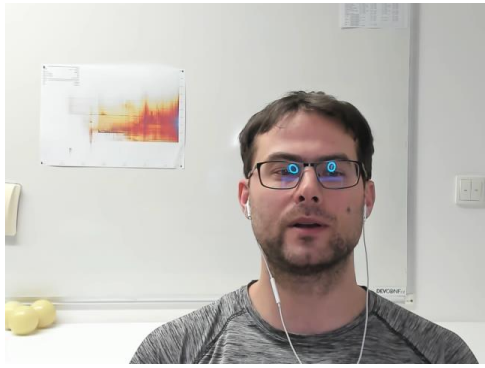


Lighting network https://explorer.acinq.co/

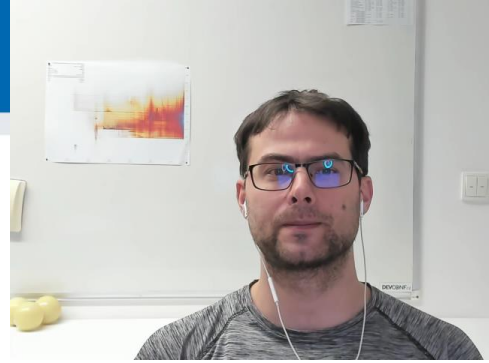


Opening channel





https://blog.usejournal.com/the-bitcoin-lightning-network-a-technical-primer-d8e073f2a82f



Some Lighting topics I.

- Custodial Lighting wallet (e.g., Wallet of Satoshi)
  – Service hold your private key, full trust in service
- Semi-custodial Lighting wallet (e.g., default BlueWallet, Zap…)
  – own key, but trust in 3rd party providing blockchain info
- Non-custodial (e.g., BlueWallet collected to own full node)
  – own key, blockchain info and monitoring by own full node
- Inbound, outbound capacity of channel between A and B
  – Initial value is given by initial on-chain 2-2 multisig transaction (x:0, x:y, 0:y)
  – Changes with every off-chain transaction executed (between A and B)

Some Lighting topics II.

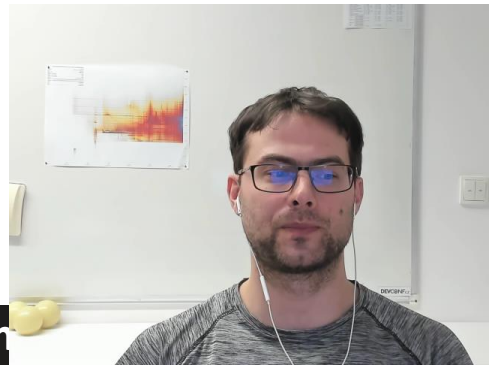
- Sentinel service
  - trustless blockchain observer, broadcasts justice transaction in case of old state detected
  - No need for your full node to be always online
- Privacy considerations
  - Most of the transactions are NOT recorded on the blockchain
    - Good for speed as well as privacy
  - Doesn’t mean that payments are not traceable
    - Same as with internet connection => need to use Tor, ideally mixes

Lightning network – study more

- Description of Lighting Network basic principles
  - https://blog.usejournal.com/the-bitcoin-lightning-network-a-technical-primer-d8e073f2a82f
- Presentation by original Lighting creators
  - https://lightning.network/lightning-network.pdf
- List of Lighting nodes ready for channel opening
  - Bottom of the https://store.blockstream.com/

Further reading

- Mastering Bitcoin (Andreas M. Antonopoulos and others)
  - https://github.com/bitcoinbook/bitcoinbook
- List of interesting resources
  - https://blockonomi.com/bitcoin-educational-resources/
  - https://learnmeabitcoin.com/, https://learnmeabitcoin.com/technical/



Top questions (1)

P PetrS 0

Is my password brute-force-able if consists of 9 printable characters?

Join at

slido.com

#pv204_2021

- **Place/upvote questions in slido while listening to lecture video**
- **We will together discuss these during every week lecture Q&A (every Monday, 17-18:00)**