

Bitcoin basics



<https://crocs.fi.muni.cz/papers/btc>



Petr Švenda  svenda@fi.muni.cz  [@rngsec](https://twitter.com/rngsec)

Centre for Research on Cryptography and Security, Masaryk University

CRCS
Centre for Research on
Cryptography and Security

Preparation

- Small switch -> enable Laptop hotspot internet sharing
 - Antminer S9, BMM100
 - Bitcoin fullnode (raspi4, hdddisk)
- Hardware wallets
 - Coldcard, Trezor, Ledger
- Software
 - Sparrow-1.9.1 <https://github.com/sparrowwallet/sparrow/releases/download/1.9.1/Sparrow-1.9.1.zip>
 - bitcoin-core-24.0.1 <https://bitcoincore.org/bin/bitcoin-core-24.0.1/>

Software to prepare

- Sparrow Wallet 1.9.1
 - <https://github.com/sparrowwallet/sparrow/releases/download/1.9.1/>
- Bitcoin-core-24.0.1
 - <https://bitcoincore.org/bin/bitcoin-core-24.0.1/>

WHY BITCOIN?

Especially if you are not interested in Bitcoin.

“Bitcoin fixes everything!”



fixes this

Important questions we will NOT cover:
 Lightning network, mining enviro impact,
 OP_RETURN, price volatility, altcoins tech...
 – great topics for chat afterwards!

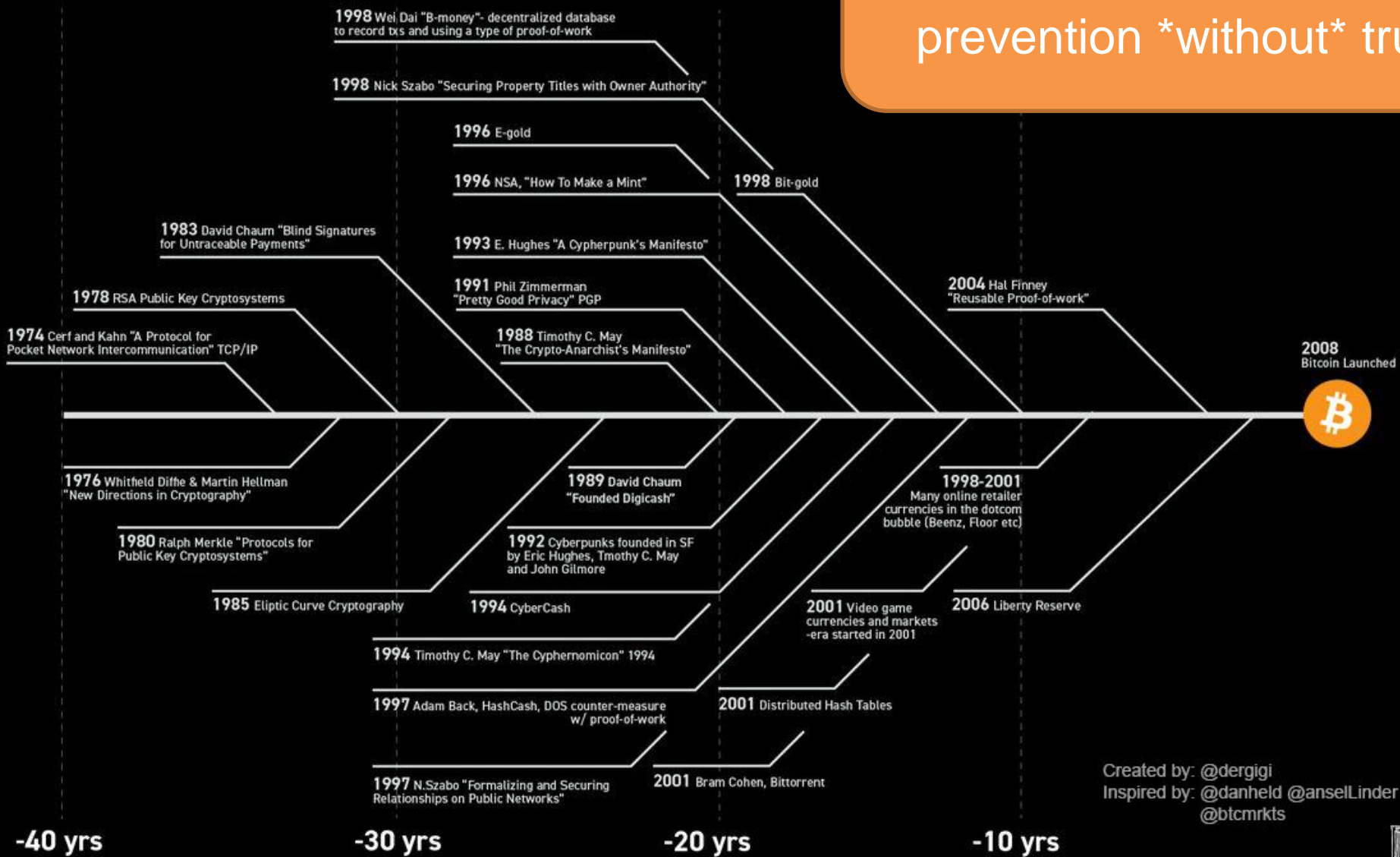
Goals for this tutorial

- Bitcoin does not fix everything, but is on a frontline
 - No safety net, no chargeback, attacker anonymous => security technique must really work, great for battle-testing security ideas, natural “bug bounty program”
- 6 main tech pieces we will cover (also usable outside Bitcoin world)
 1. How to backup key(s) (single seed, BIP39, Shamir)
 2. ~~How to make always fresh keys (derivation via BIP32, also address privacy)~~
 3. ~~How to protect signing key against malware~~
 - (multisig, hardware wallet, airgap pc + tx b
 4. ~~How to introduce restricted signing policy (ti~~
 5. ~~How to protect your financial privacy (CoinJo~~
 6. ~~How to use hardware wallet with secure element~~

If interested in more details about
 Bitcoin usage tutorial, visit
<https://crocs.fi.muni.cz/papers/btc>

Bitcoin prehistory - It's the result of 40 years of

Bitcoin is built on decades of research ideas. The main innovation is double-spending prevention *without* trusted central party



Created by: @dergigi
Inspired by: @danheld @anselLinder @btcmrks

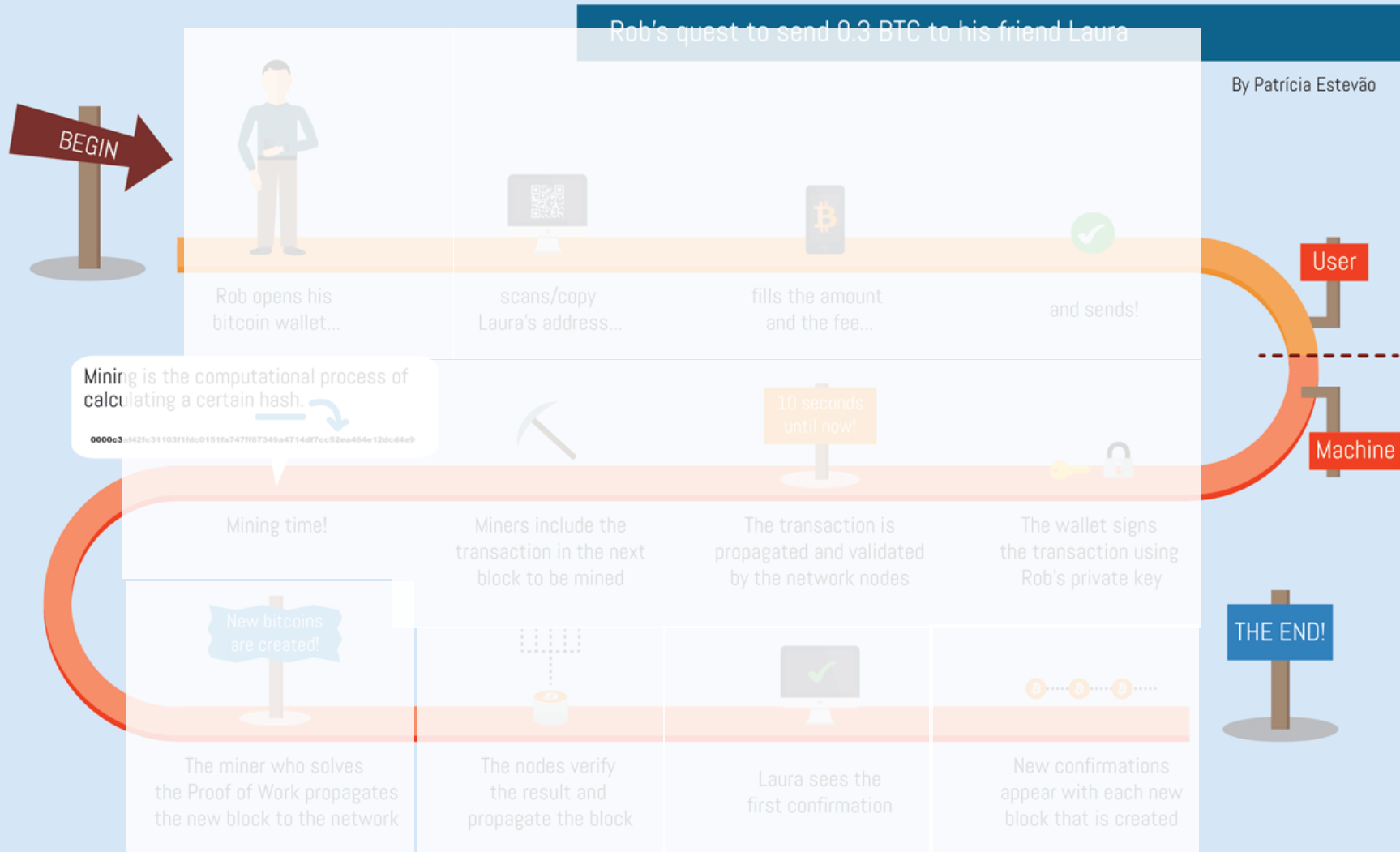


Overview

1. Explanation of Bitcoin network parts
2. Using Bitcoin Core full node (mainnet)
 - Start downloading blocks (only few), investigate connected peers, network, see part of blockchain content, check with blockchain explorer
3. Send transaction (testnet)
 - Generate Sparrow wallet, save seed, get testnet4 btc, create transaction, observe on mempool
4. Mining basics (mainnet)
 - Demonstration of BOS console, mining pool, mempool.space dashboard
5. Getting some real sats (Lightning)

BASICS

THE BITCOIN TRANSACTION LIFE CYCLE



- Wallet
- Address
- Fee
- Transaction
- Signing
- Network nodes
- Block
- Mining
- Proof of Work
- Verification
- Block reward
- Tx confirmation
- And many more...

Main design goals of the Bitcoin

1. Decentralization

- No central authority or intermediary (=> no single point of failure), possibility of self-custody
- No limitation on network participants (no permission to join is required)
- Applies to executing a transaction, but also development, infrastructure, mining...

2. Transparency

- All transactions recorded on public ledger; validity of every “bitcoin” is easy to verify
- Total number of bitcoins in circulation easy to assess (monetary policy, fixed supply)

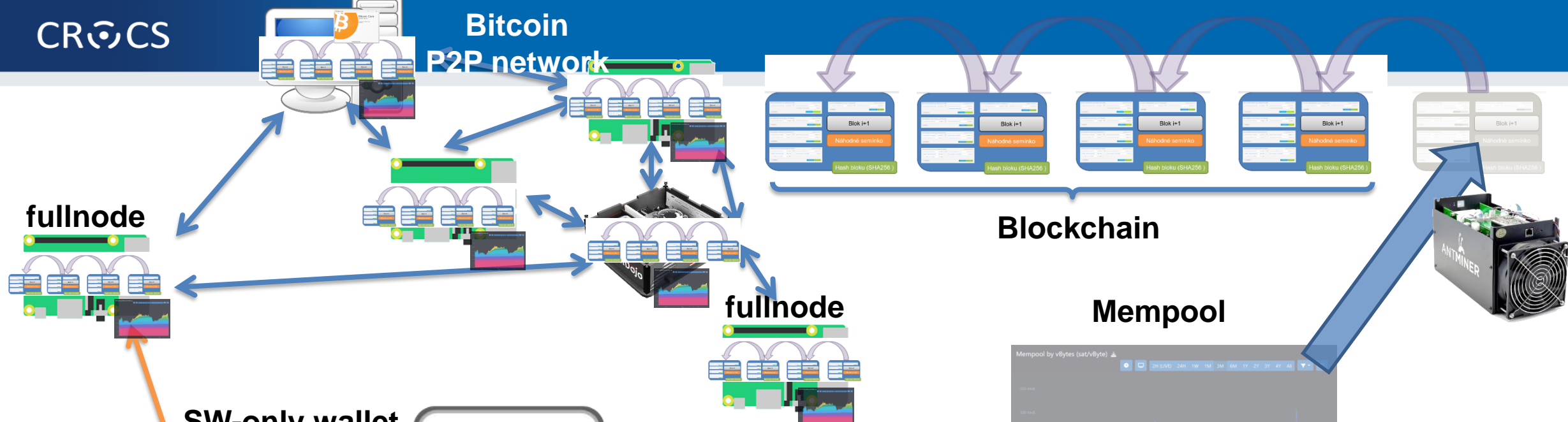
3. Security based on cryptography (mainly signature, hash functions)

- Ownership of bitcoins proved only cryptographically (no “chargeback” based on human decision)
- Protection of bitcoins reduced to protection of private key(s)

4. Pseudonymity of participants

- bitcoins connected to public keys, not usernames (does not automatically mean anonymity!)

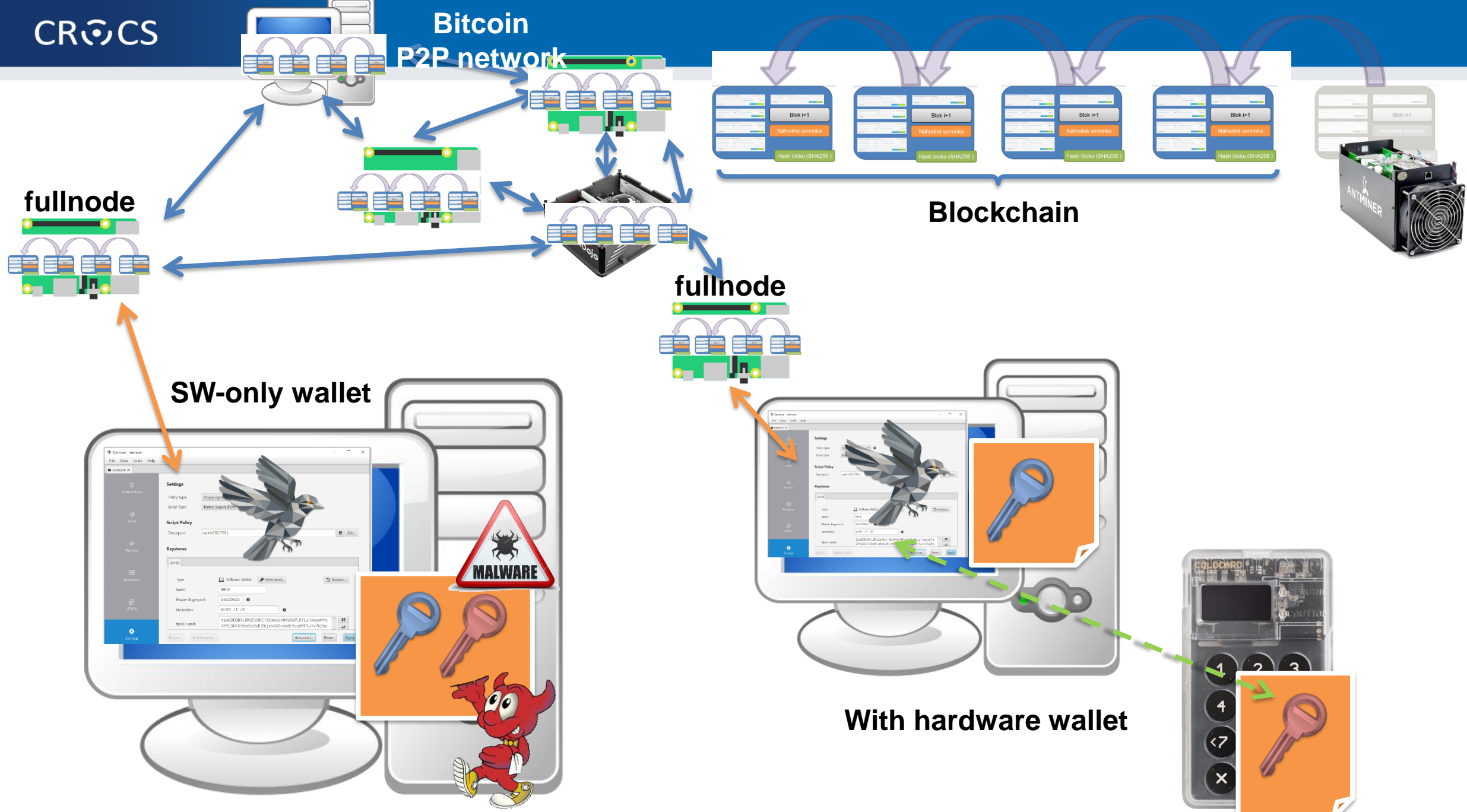
Bitcoin P2P network



SW-only wallet



Bitcoin P2P network



fullnode

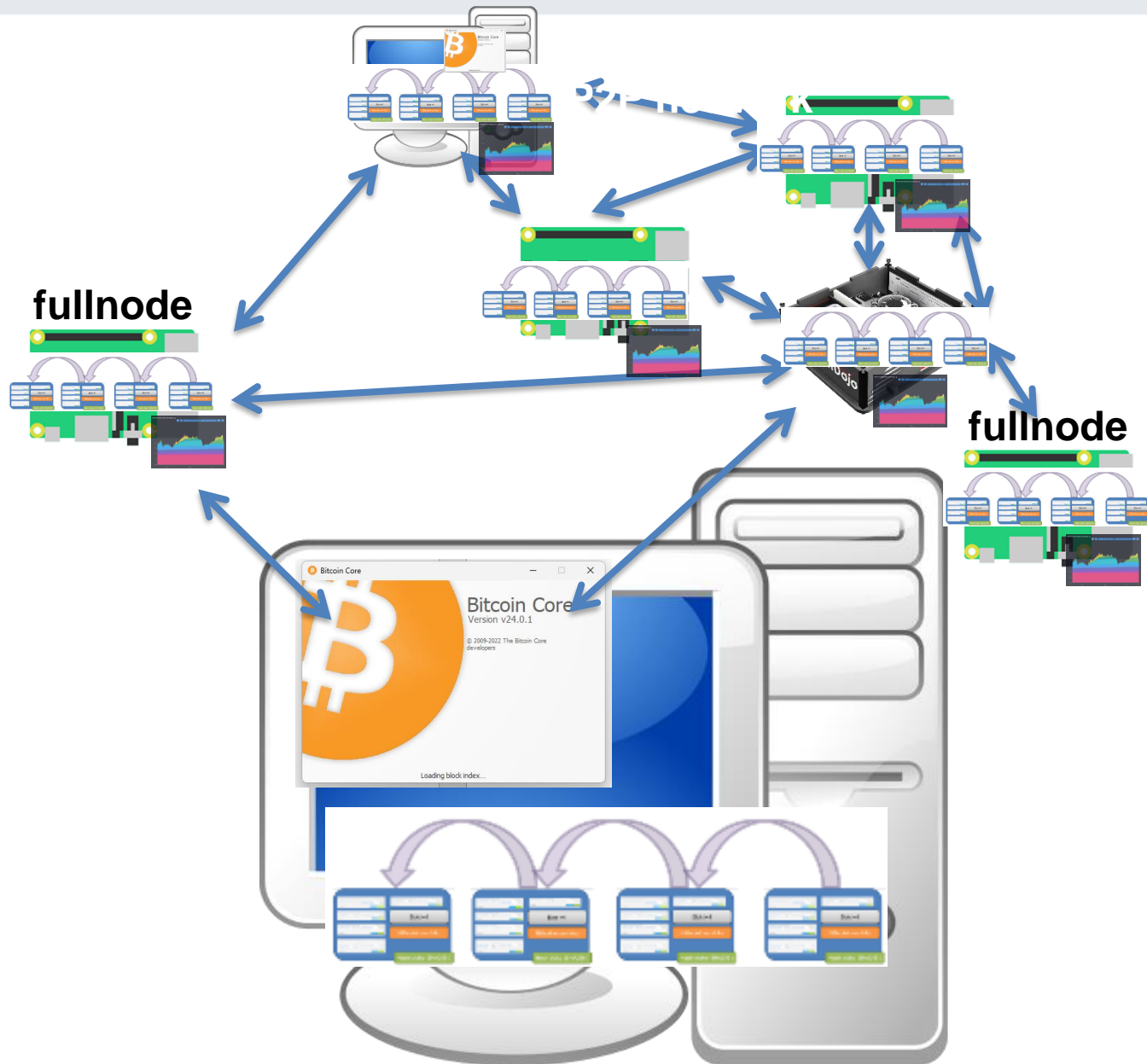
Blockchain

fullnode

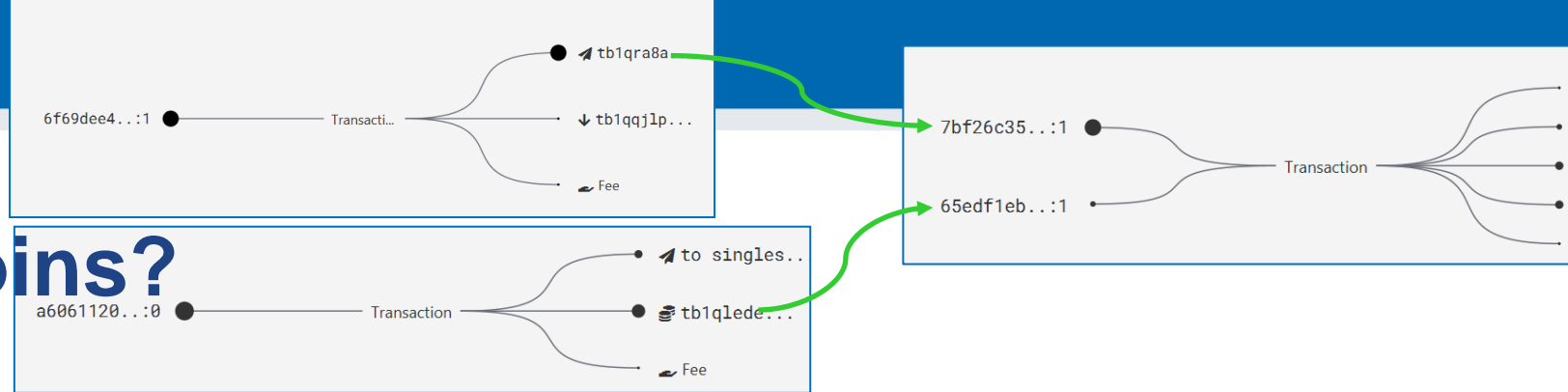
SW-only wallet

With hardware wallet

MALWARE



OVERVIEW

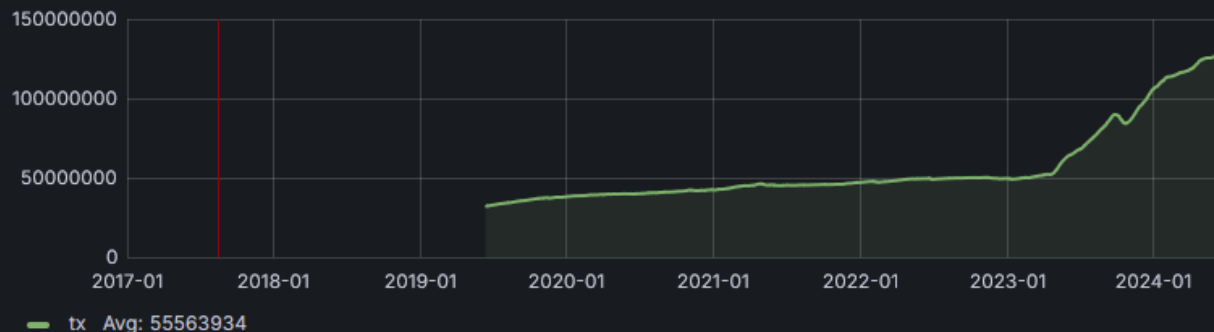


Where are my bitcoins?

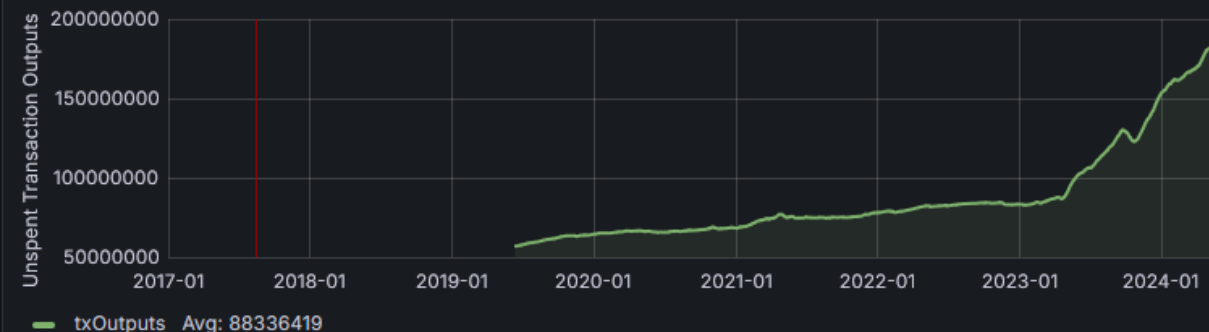
- Public ledger of all transactions (blockchain)
 - Propagated between Bitcoin fullnodes (P2P network)
- “Bitcoin holdings” - sum of values of not-yet-spent transactions control
 - Unspent Transaction Output (UTXO)
- “Bitcoin send” – take “your” UTXO and use it as input to new one
 - Specify recipient by script specifying what must be done in future send (lockscript)
 - Typical lockscript is “prove that you can sign with private key corresponding to THIS public key”
- “Bitcoin receive” – generate variable part of lockscript (public) and share with sender + monitor blockchain for my transaction
- Protection and handling of private keys is paramount
 - “Not your keys, not your bitcoin! “

UTXO set = all currently valid “bitcoins”

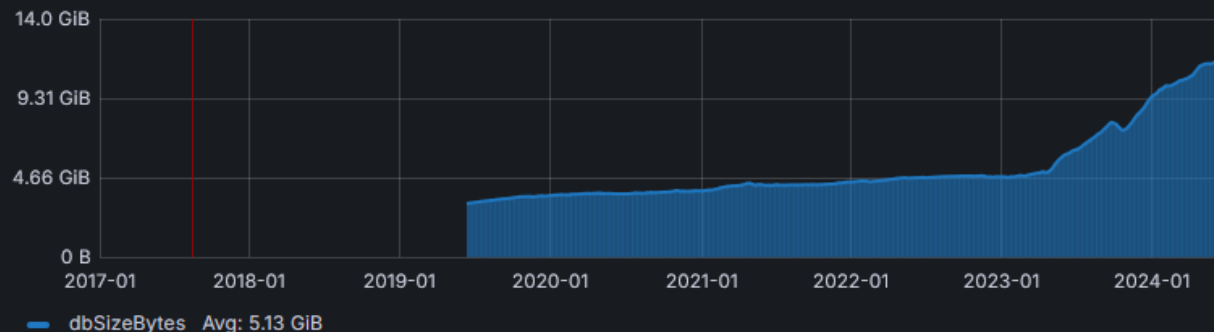
Total Transactions With Unspent Outputs ⚠



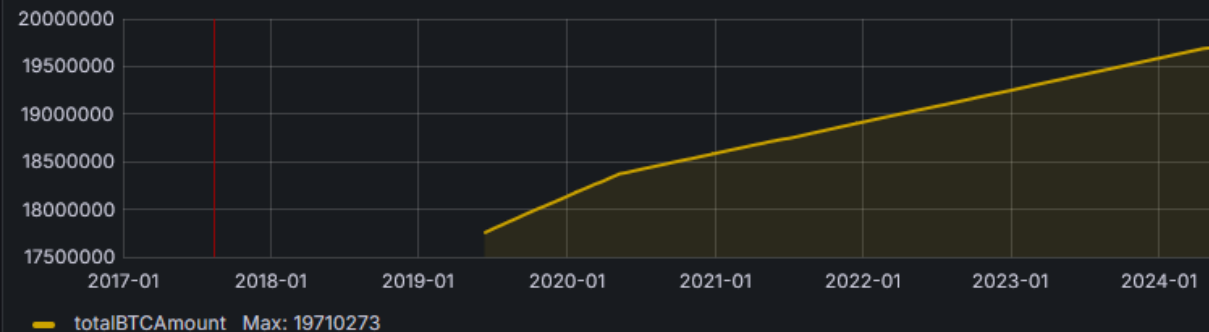
Unspent Transaction Outputs ⚠



Size of Serialized UTXO Set ⚠



Total Bitcoins in Existence ⚠



<https://statoshi.info/d/000000009/unspent-transaction-output-set?orgId=1&refresh=10m&from=1483225200000&to=now>

Networks in Bitcoin (Mainnet, Testnet, Regtest)

- **Mainnet** – main, global production network
- **Testnet** – testing network (global, some mining happens...)
 - Restarted from time to time, contains many different types and versions of TXs
- **Regtest** – local instance of Bitcoin network
 - Used for local testing (integration, regression, debugging)
 - Blockchain started from block 0, you are the only miner
 - (mined bitcoins unusable on Mainnet)
 - You can insert own transactions, decide on mining new blocks, debug...
- **Lightning** – second layer network of payment channels atop of mainnet
 - Practically instant and very low fees independently from mainnet

P2P Bitcoin network map <https://bitnodes.io/>

BITNODES

Bitnodes estimates the relative size of the Bitcoin peer-to-peer network by finding all of its reachable nodes.

REACHABLE BITCOIN NODES

Updated: Mon Jun 17 09:28:50 2024 CEST

16780 NODES

CHARTS

IPv4: +1.8% / IPv6: -6.6% / .onion: -16.1%

Top 10 countries with their respective number of reachable nodes are as follows.

RANK	COUNTRY	NODES
1	n/a	9612 (57.28%)
2	United States	1734 (10.33%)
3	Germany	1660 (9.89%)
4	France	445 (2.65%)
5	Netherlands	338 (2.01%)
6	Finland	300 (1.79%)
7	Canada	290 (1.73%)
8	United Kingdom	215 (1.28%)
9	Singapore	203 (1.21%)
10	Switzerland	185 (1.10%)

All (96) »



Map shows concentration of re

19063

Reachable nodes

11582

Average

13252 ▲ 228.05%

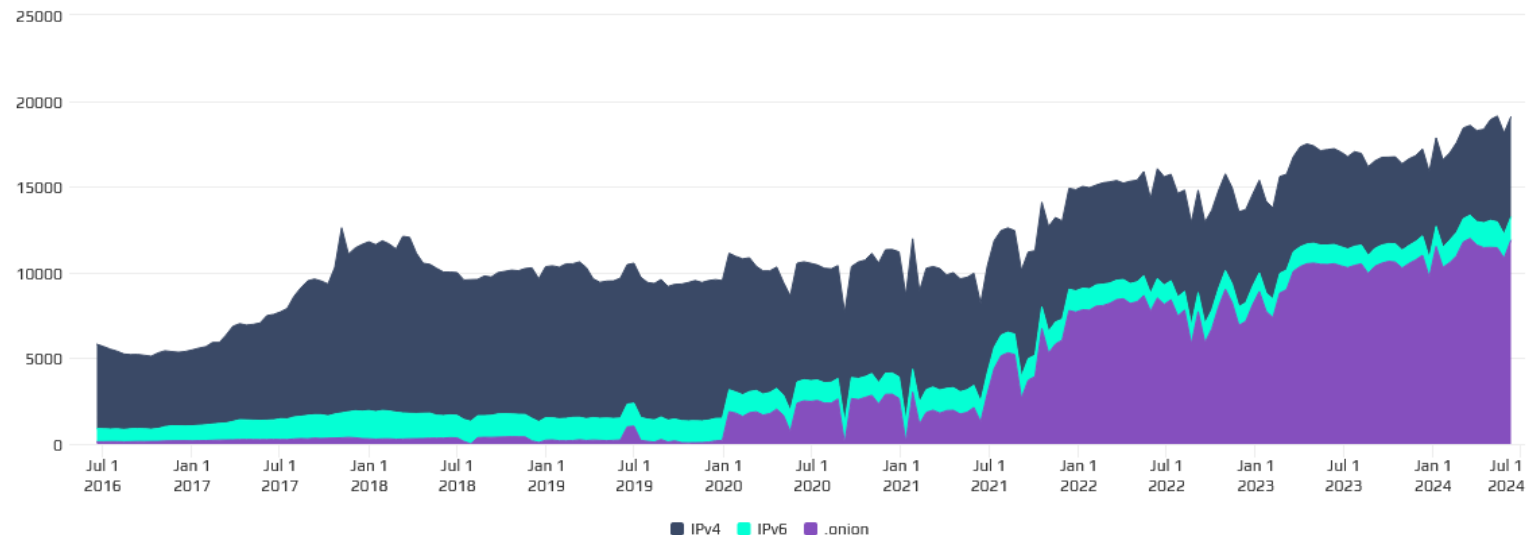
Since 8 years ago

NODES

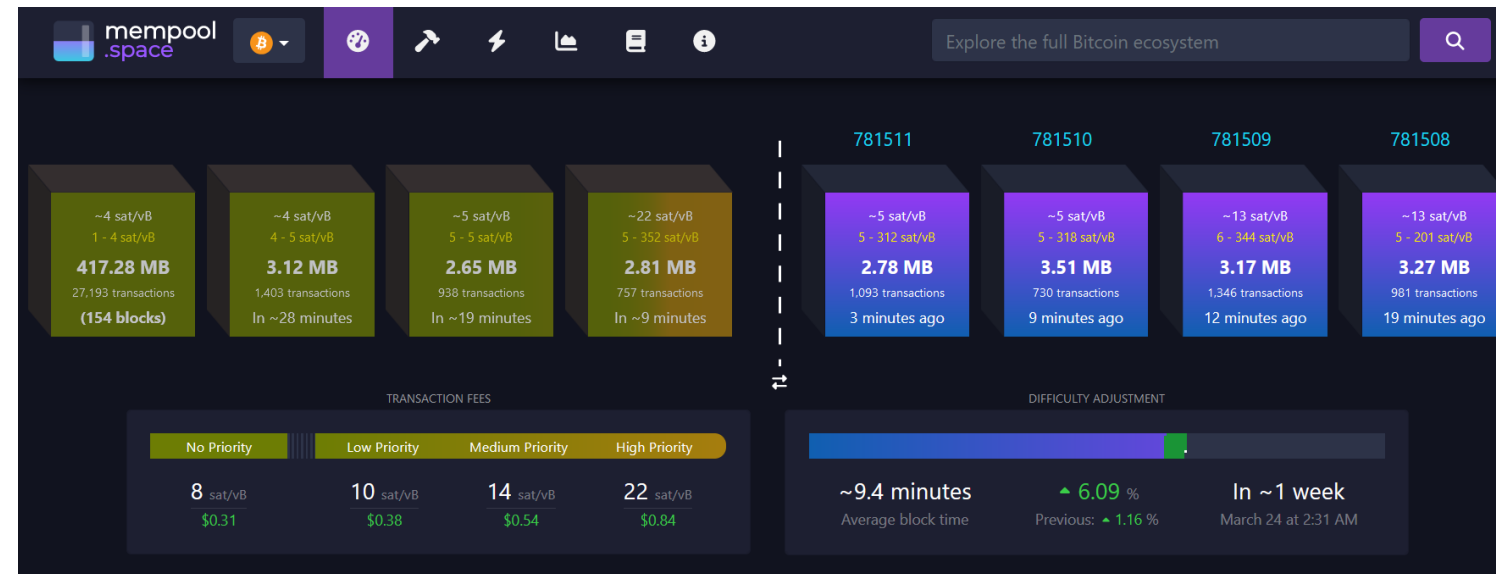
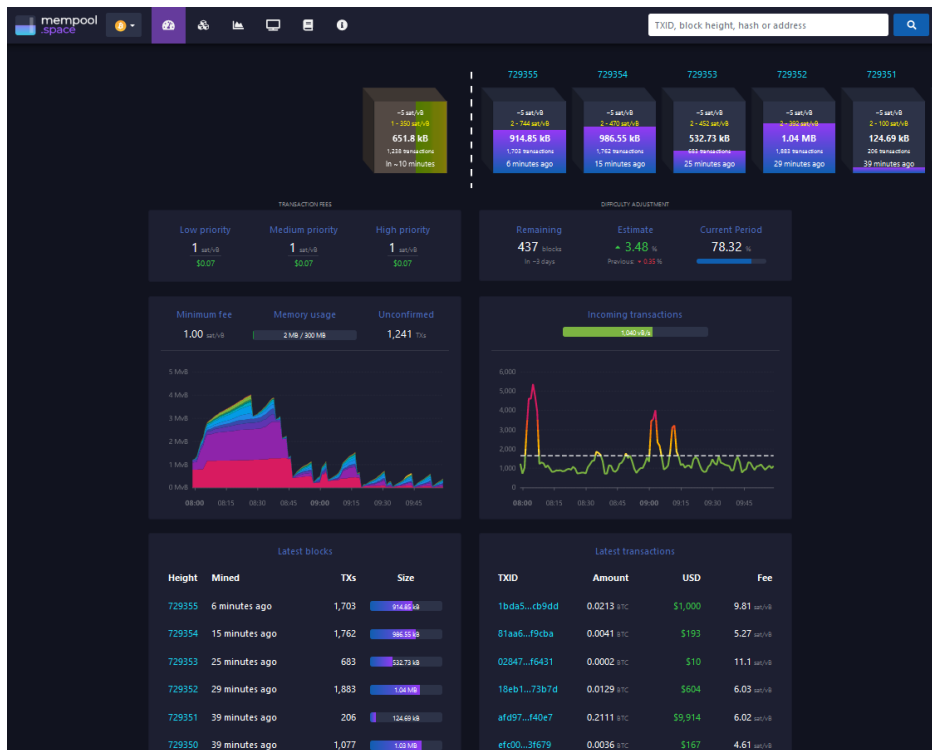
Chart shows the number of reachable Bitcoin nodes during the last 8 years. Series can be enabled or disabled from the legend to view the chart for specific networks.

24h 90d 1y 8y

Lo 5115 Hi 19117 Avg 11582 Last 19063 nodes



Popular mempool explorer – <https://mempool.space>



- Can be run on your own fullnode (privacy improvement)
- Testnet version <https://mempool.space/testnet4>

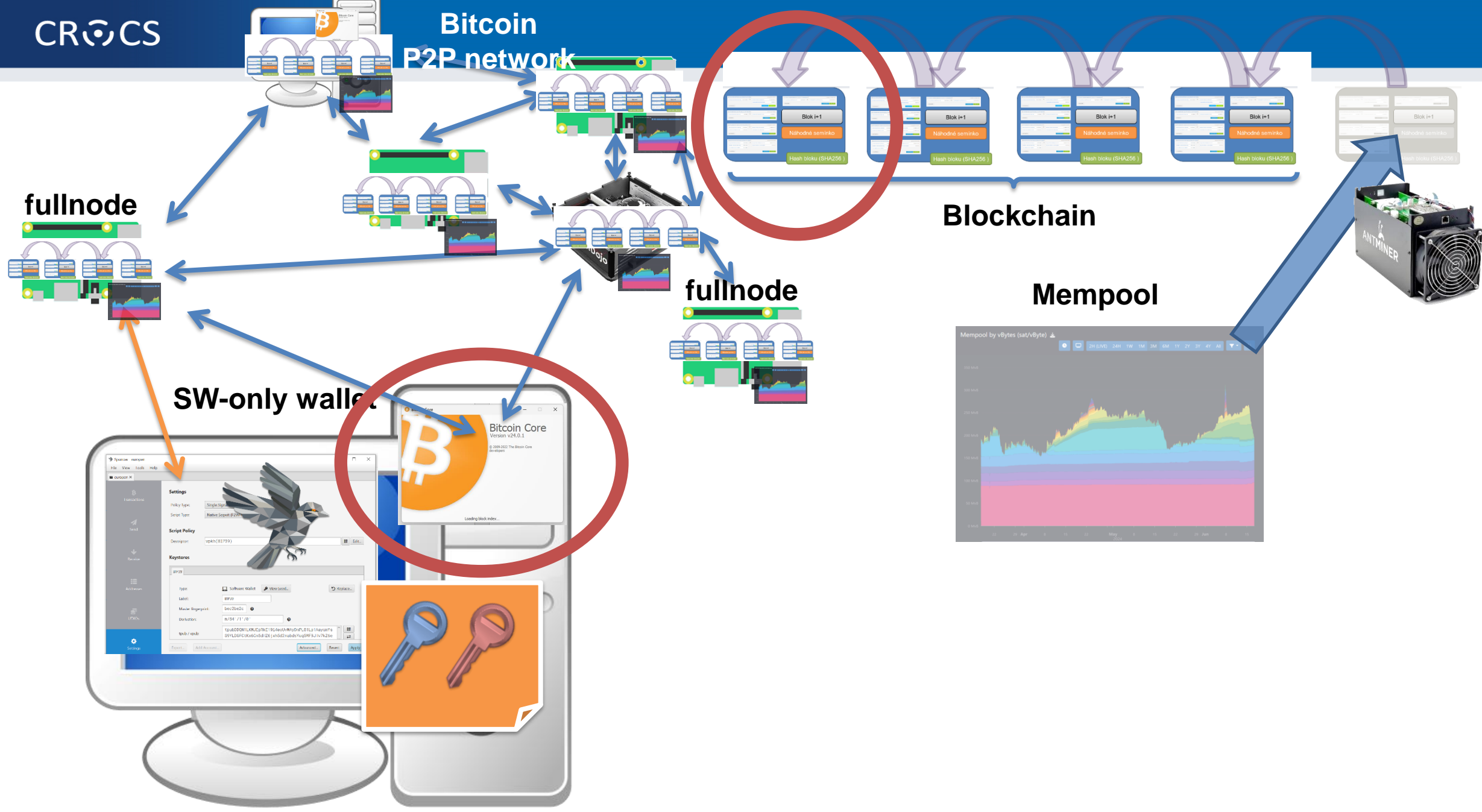
- Frequent issue – Sparrow not started with `–testnet` switch
- Make breakout rooms (3 people per room)
- Ask them to perform some mutual transactions
- Connect to groups and discuss:
 - Why is tx as unconfirmed? Where it is?
 - Why are we waiting some minutes?
 - How to cheat on someone who will sell us car right after tx is in mempool?
 - double spent with higher fee => Invalidation of original tx
 - Doublespent tx can be prepared into file as binary blob
 - If already mined, it is more difficult, but still somewhat possible (chain reorg)

TASK: USING BITCOIN CORE

What you will learn:

- Run fullnode software connecting to Bitcoin P2P network
- Understand role of peers in network
- Understand basic blockchain structure
- Investigate first block (Genesis block)

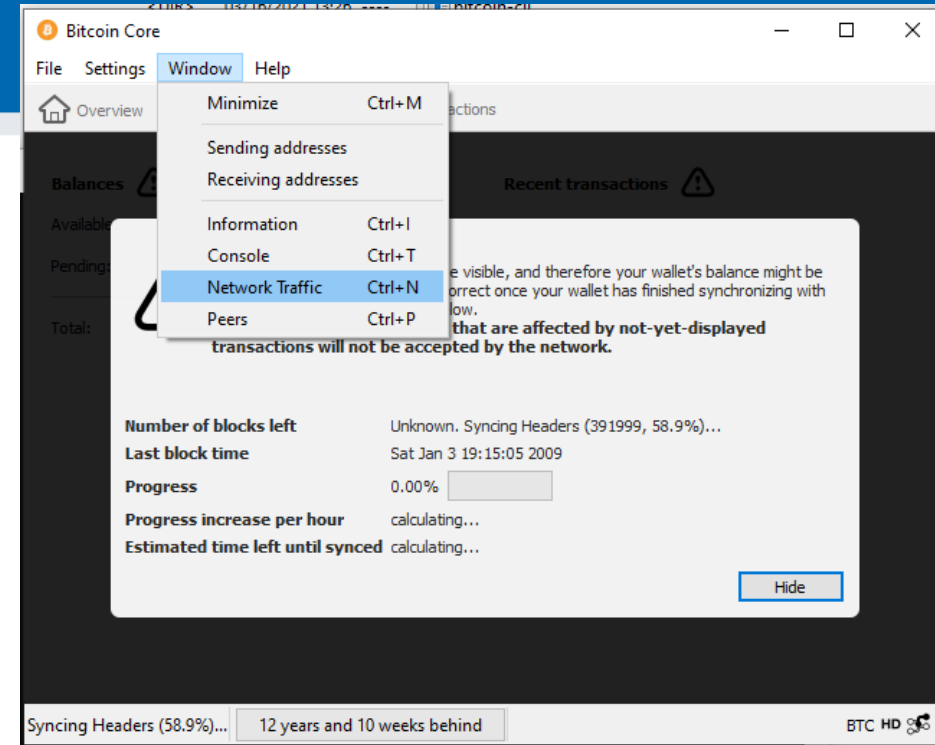
Bitcoin P2P network



- Ask people to run GUI, see the connected nodes, locate block files
- Explain how “Chancellor ... “ got there (coinbase data, used to randomize block during mining, pool info...)
- 100 blocks / file
- Importance of small blockchain => more fullnodes possible to be run
=> better decentralization

Running own full node

- Get Bitcoin full node **24.0.1** (pick .zip or .gz)
 - <https://github.com/bitcoin/bitcoin/releases>
 - <https://bitcoincore.org/bin/bitcoin-core-24.0.1/>
 - Download and unpack .zip or .gz
- Download few blocks from real Bitcoin P2P network
 - Run bitcoin-qt, Window → Network Traffic (Ctrl+N), Peers (Ctrl+P)
 - Observe and document peers to which you connected (number, version, IP)
- Analyze first few blocks from blockchain
 - Look into Bitcoin/blocks/blk00000.dat (e.g., C:/Bitcoin/blocks/blk00000.dat)
 - If on Windows, Look for bitcoin folder also in your profile
 - c:\Users\your_name\AppData\Roaming\Bitcoin\blocks\



Questions

- Why is your full node connecting to other nodes?
- For how long is the Bitcoin network running now?
- What is the content of first block?
- What is the privacy advantage of sending/querying TXs using your own full node?
- How can you compute the current supply of bitcoins?

Lister - [c:\Bitcoin\blocks\blk00000.dat]

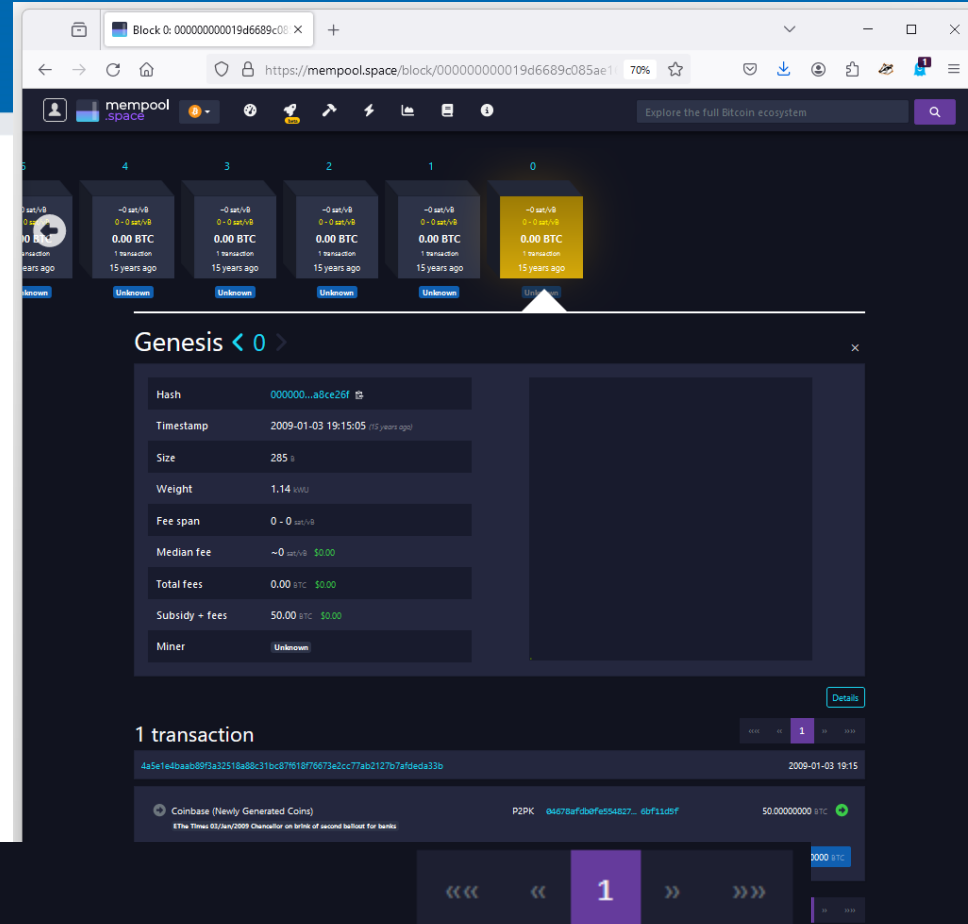
File Edit Options Encoding Help

Terminal window showing a corrupted file dump with garbled characters and some legible text like 'The Times' and 'Chancellor on brink of second bailout for banks'.

THE TIMES newspaper front page from Saturday January 3, 2009. Main headline: 'Chancellor on brink of second bailout for banks'. Other headlines include 'Israel prepares to send tanks and troops into Gaza' and 'Eat Out from £5'.

Inspect Genesis block

- Visit <https://mempool.space>
- Insert '0' into search box
 - Block with block height 0 (Genesis block)
- Only single transaction present



1 transaction

4a5e1e4baab89f3a32518a88c31bc87f618f76673e2cc77ab2127b7afdeda33b

2009-01-03 19:15

→ Coinbase (Newly Generated Coins)

P2PK 04678afdb0fe5548...6bf11d5f

50.00000000 BTC →

EThe Times 03/Jan/2009 Chancellor on brink of second bailout for banks

50.00000000 BTC

Run strings on already downloaded blocks

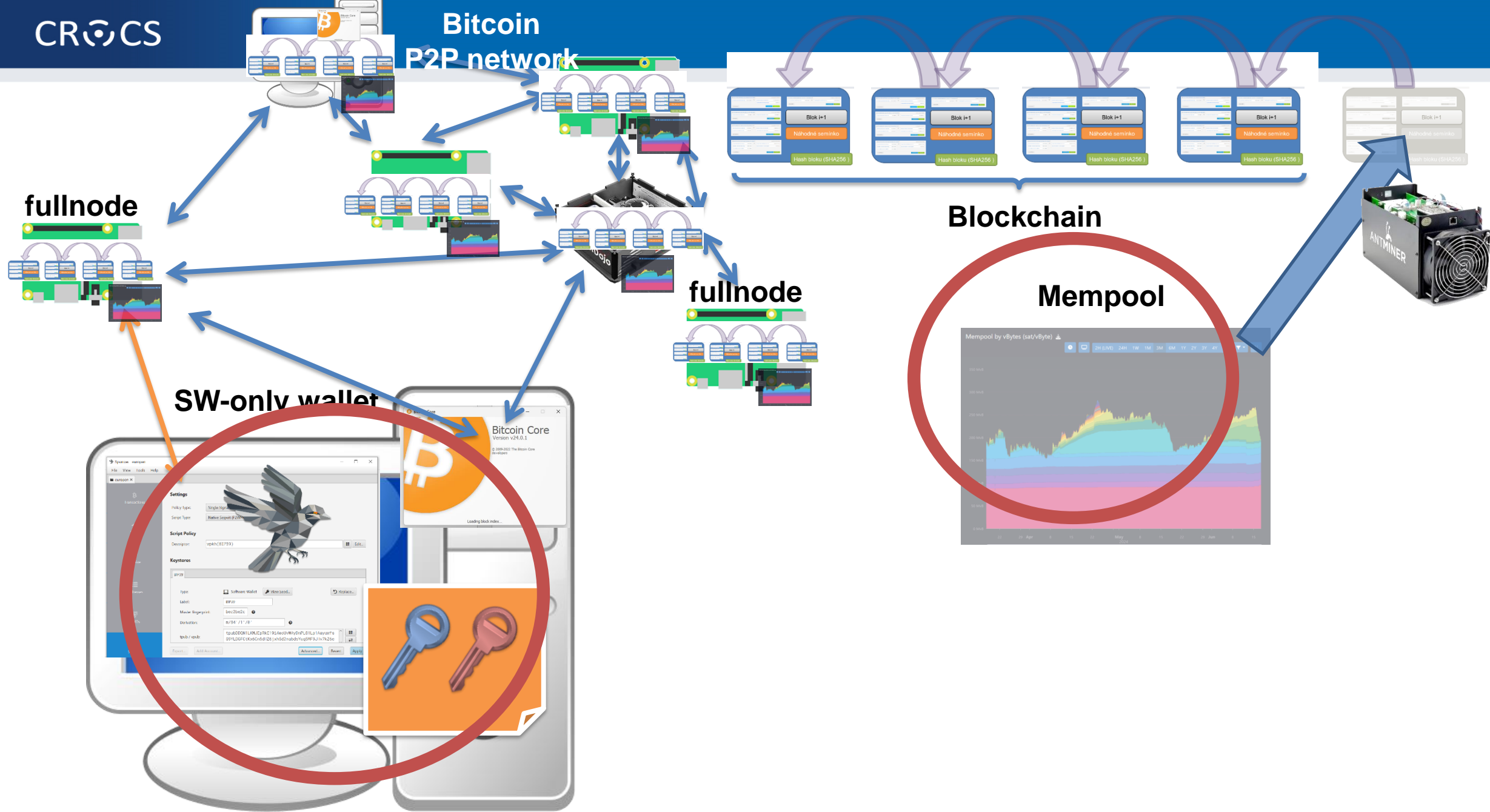
- **strings** command on Linux
- **strings** on Windows: <https://docs.microsoft.com/en-us/sysinternals/downloads/strings>
- **c:\Bitcoin\blocks**>strings -n 20 *.dat

TASK: USING SIGNATURE COORDINATOR

What you will learn:

- Create Bitcoin wallet
- Backup cryptographic seed
- Obtain testing coins
- Send and receive bitcoins by signing transaction
- Investigate your transaction on blockchain explorer
- Improving security of private keys with hardware wallet

Bitcoin P2P network





SINGLE-SIGNATURE WALLET (SW-ONLY)



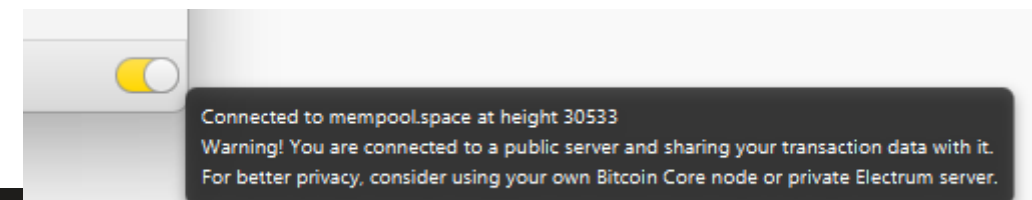
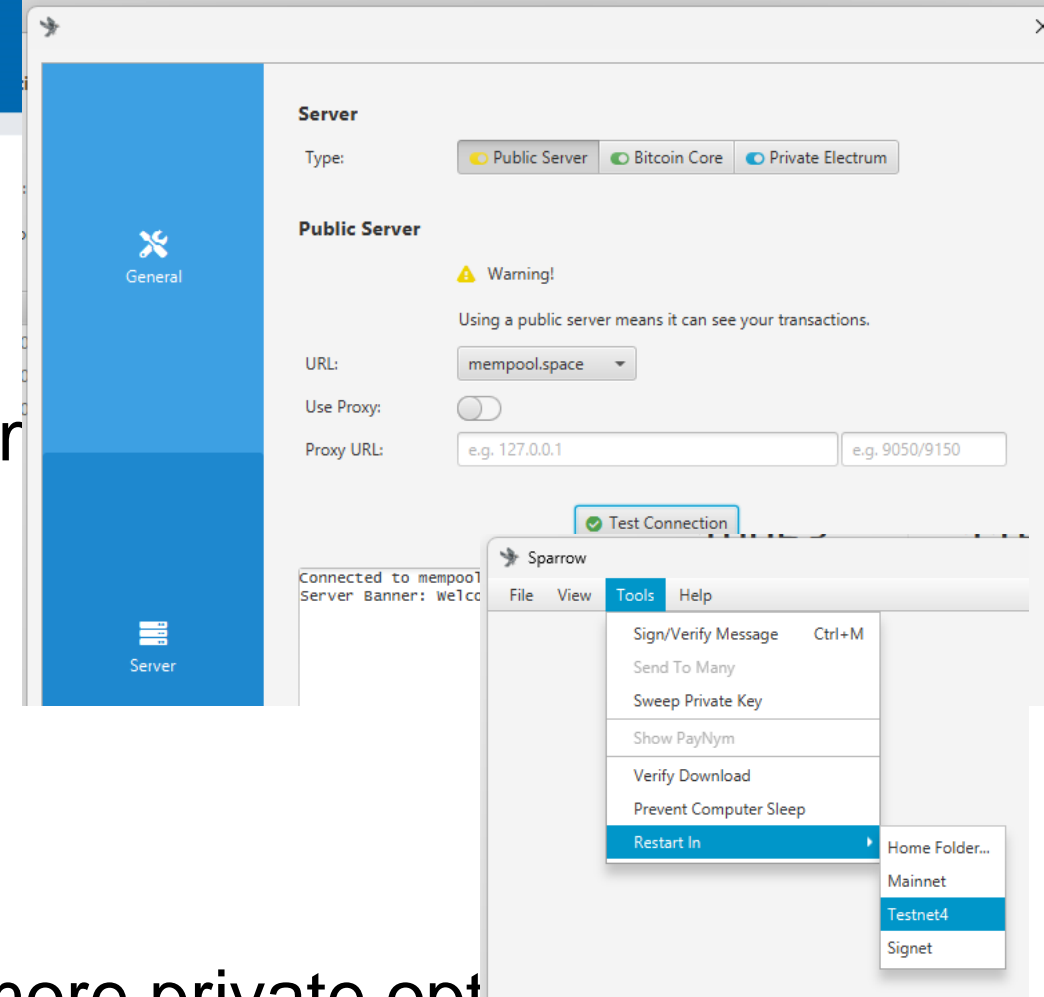
Sparrow wallet (v1.9.1)

- <https://www.sparrowwallet.com/download/>
- For serious work, always verify binary releases (`gpg --verify`)
- Well-known and maintained, Java-based, minimum other dependencies, focus on medium and advanced users
- Sparrow is “signing coordinator” – private keys inside or external wallet
- Basic functionality
 - Open-source wallet, **non-custodial** wallet
 - Support for software and hardware wallets, multisignature coordinator
 - Supports also advanced features (PayJoin, Taproot addresses...)

(Examples created for Sparrow 1.9.1)

Starting Sparrow wallet

- Run your wallet with testnet switch (command)
 - `./sparrow -n testnet4`
 - `Sparrow.exe -n testnet4`
- Use Public Server option if asked
 - Test Connection to verify connectivity
 - Can be changed later File → Settings
- (Bitcoin Core and Private Electrum are more private options,
 - You would be connecting to your own fullnode (but you must have one 😊)
- Check that you are online
 - (right bottom, yellow or green button)



Generating new “wallet”

- A “wallet” is key management software controlling your private and public keys (ECDSA, Schnorr)
- The most important part of wallet is random number called root seed (128 or 256 bits)
- Root seed is used to deterministically generate practically unlimited number of keypairs
 - Specified in BIP32, “root seed” and “derivation path” used to derive next private key => next public key => next address
- Clever construction allowing to compute future public keys (and only public keys) for specified derivation path without the need for root seed (aka xpub or extended public key)
 - Knowledge of xpub allows to compute all future public keys, but not private keys
 - Owner of root seed can compute all future private keys and their corresponding public keys
 - xpub allows to pay someone to fresh addresses noninteractively (no interaction with owner of root seed required), receiver will only later compute candidate private keys and their public keys to check for total balance (== set of UTXOs)
- Wallet software is monitoring blockchain for addresses corresponding to stored root seed (or xpub)
- Root seed can be stored:
 1. Directly in software wallet (file on harddisk, optionally encrypted) == aka hot wallet, least secure against malware
 2. Loaded every time before use (e.g., from QR code), still vulnerable to malware during use
 3. On external hardware signing device called hardware wallet (the most secure option)

Create wallet

- `sparrow -n testnet4`
- File → New wallet
- 1. New or Imported Software wallet
- 2. Use 12 Words
- 3. Generate New
- Write 12 words on paper
- Leave Passphrase empty
– (additional wallet diversification)

The screenshot shows the Sparrow wallet creation process. The main window is titled 'Sparrow - europen' and has a menu bar with 'File', 'View', 'Tools', and 'Help'. A search bar contains 'europen'. The interface is divided into several sections:

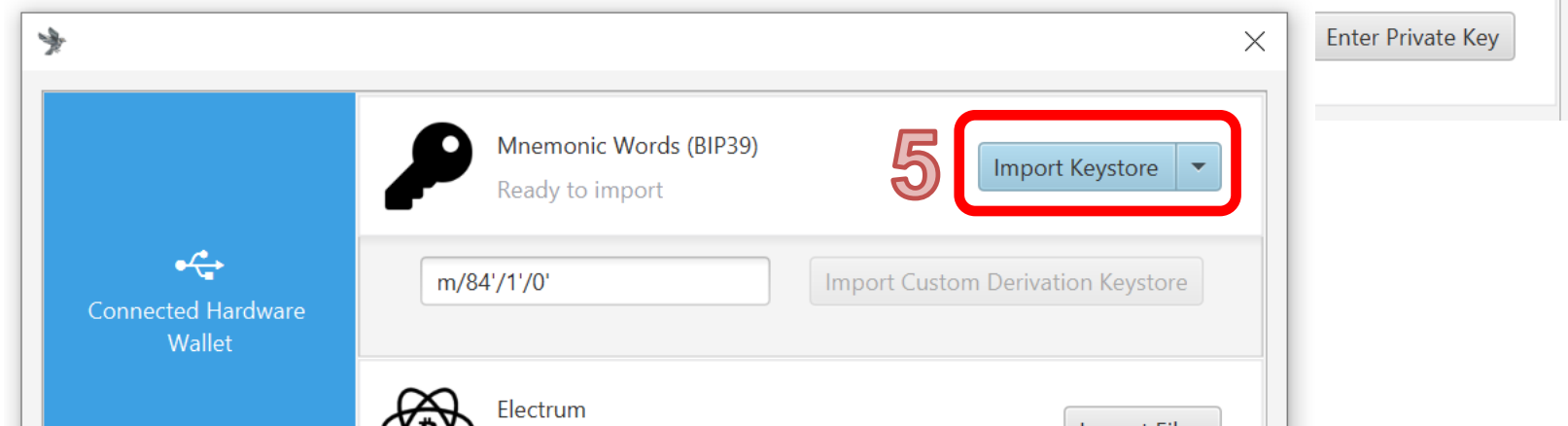
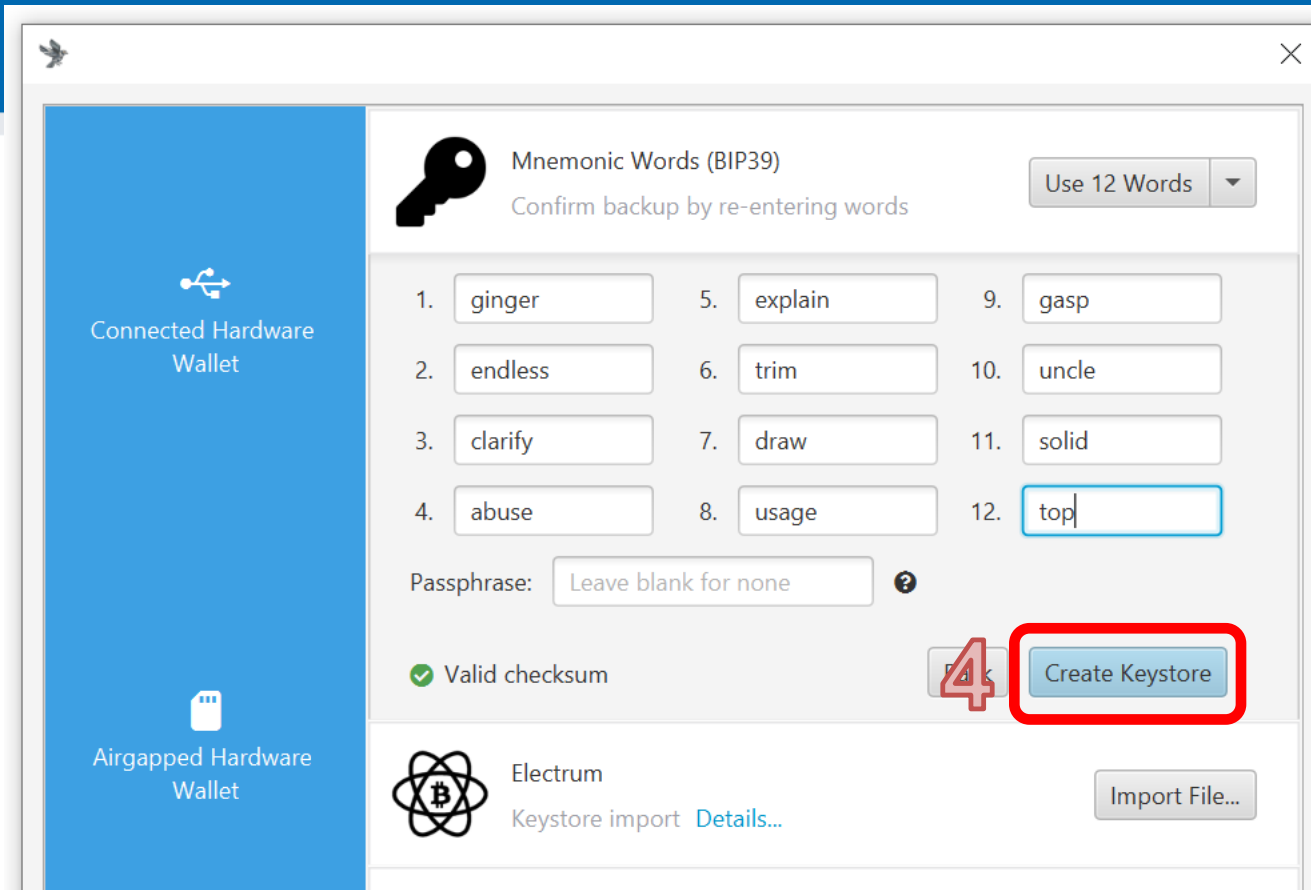
- Settings:** Policy Type (Single), Script Type (Native).
- Script Policy:** Descriptor (wpk).
- Keystores:** Keystore 1.
- Wallet Selection:** Four options are shown: Connected Hardware Wallet, Airgapped Hardware Wallet, **New or Imported Software Wallet** (highlighted with a red box and a red '1'), and xPub / Watch Only Wallet.
- Mnemonic Words (BIP39):** A dropdown menu is open, showing options for 12, 15, 18, and 21 words. 'Use 12 Words' is selected (highlighted with a red box and a red '2').
- Generate New:** A button to generate a new wallet (highlighted with a red box and a red '3').
- Passphrase:** A text field with the placeholder 'Leave blank for none'.
- Other Options:** 'Create Keystore', 'Import File...', and 'Enter Private Key' buttons.

Create wallet

4. Create Keystore

- Confirm backup
- Reenter words

5. Import Keystore



Create wallet

6. Apply

7. Set password or leave empty

- (encryption of local wallet file)

- Local wallet contains seed

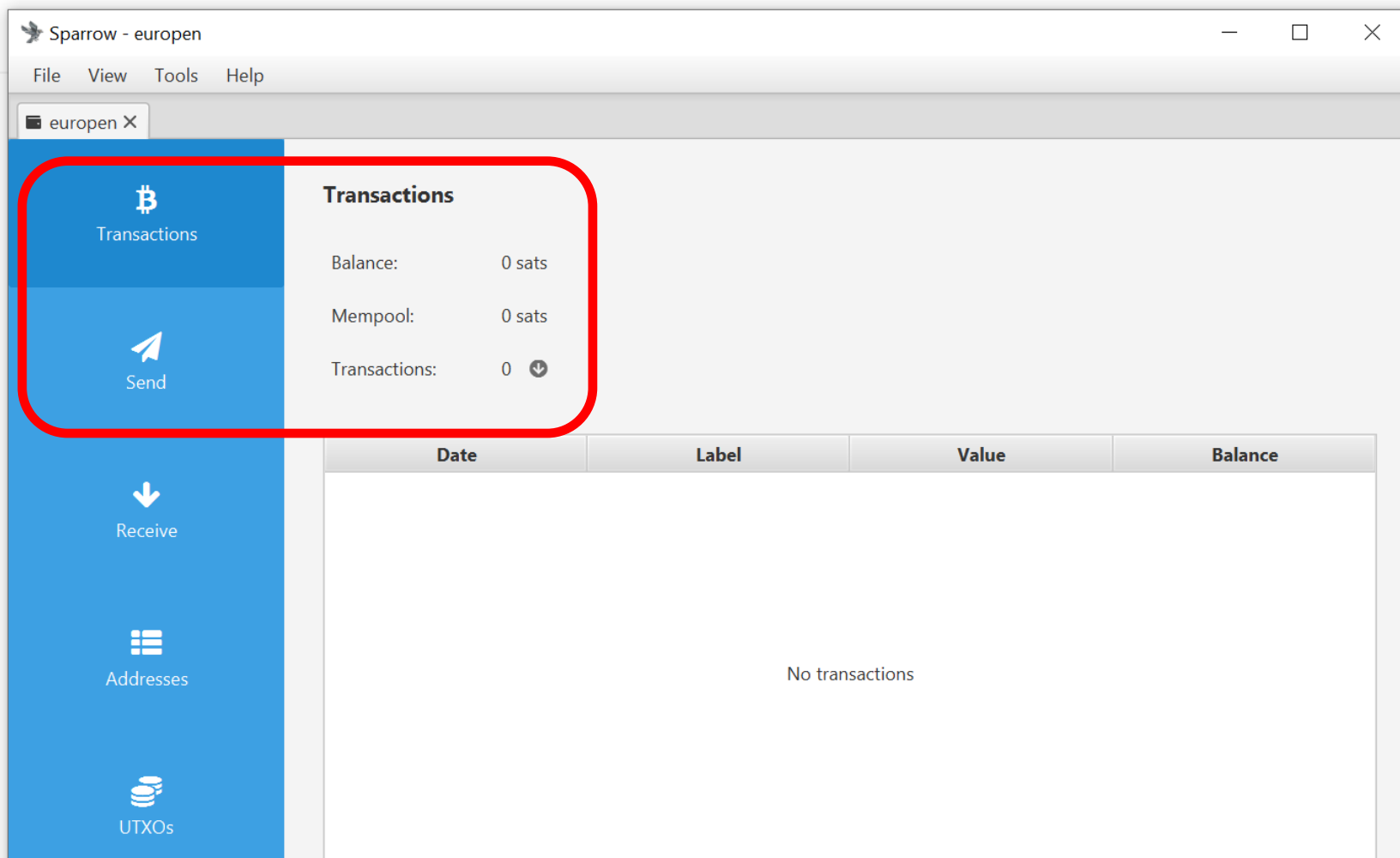
- *.mv.db file

- File → Open wallet



The screenshot shows the Sparrow wallet interface. The 'Settings' tab is active, displaying 'Policy Type: Single Signature' and 'Script Type: Native Segwit (P2WPKH)'. The 'Keystores' tab shows a 'BIP39' wallet configuration with fields for 'Type', 'Label', 'Master fingerprint', and 'Derivation'. A 'Wallet Password' dialog box is overlaid on the settings, asking to 'Add a password to the wallet?' with a 'No Password' button highlighted in red. A red circle with the number '6' is around the 'Apply' button in the bottom right corner of the settings window.

Wallet created (but empty so far 😊)



Receiving (testnet) bitcoins

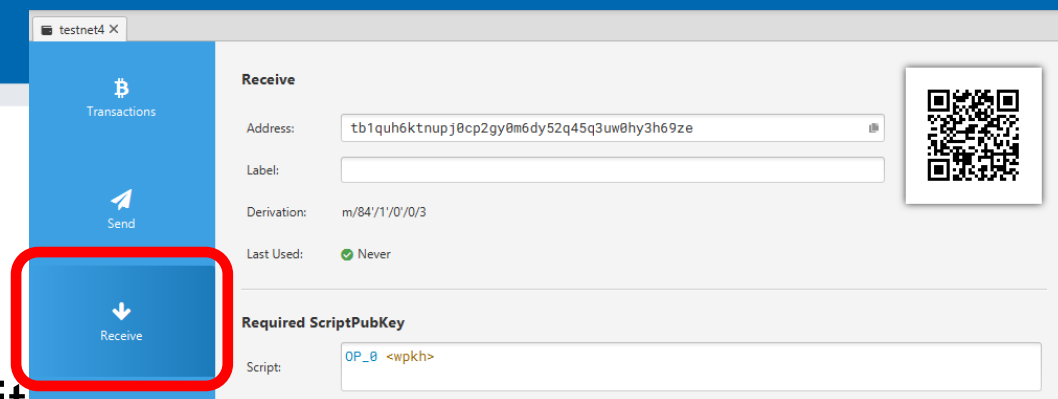
- You generate new “address”
 - deterministically derived from your root seed and fresh derivation path (path + counter) => new ECDSA keypair [BIP32]
 - public key X is pasted into locking script (“who can sign with private key verifiable with X can move bitcoin further”) and hashed => “address” [P2SH/P2WSH] (Pay to witness script hash)
- Service coinfaucet.eu owns multiple tBTC
 - Service is providing limited number of test bitcoins (tBTC) for free
 - Service owns UTXOs => someone previously locked some tBTC to their keypair(s)
 - Service creates new transaction with some tBTC locked to your “address”
 - New transaction is broadcasted to Bitcoin P2P network and stored in mempools (set of unconfirmed transactions)
- Miners will eventually include this transaction into new block (head of blockchain)
 - Confirmed and removed from mempools
 - Your Sparrow wallet is monitoring both mempool and blockchain (instant notification about pending transaction)

Getting test bitcoins (tBTC)

- If not running, run your wallet with testnet switch (command line)
 - E.g., `./sparrow -n testnet4`
 - Generate new (testnet) receive address
- Go to <https://mempool.space/testnet4/faucet>
 - Insert your testnet4 receive address
 - You may get more every 12 hours (per single IP)
 - (but please don't abuse)
- Check your tx: <https://mempool.space/testnet4>
 - Software visualizing blockchain

Note: I will send you testnet4 coins

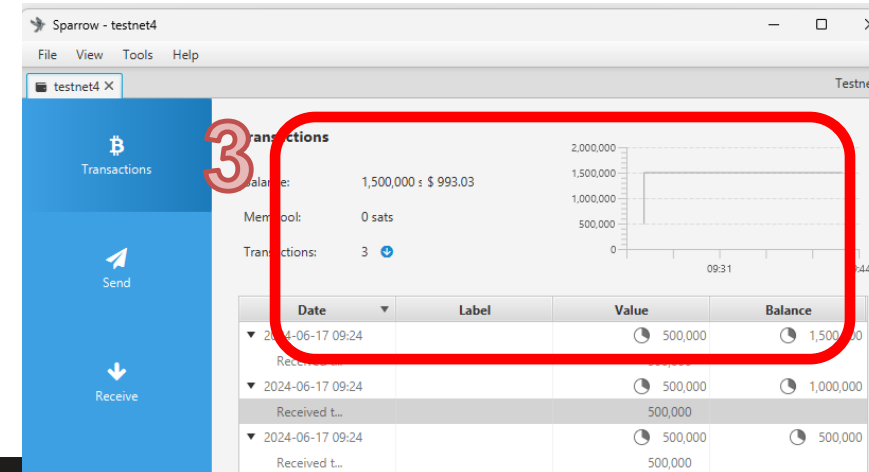
1



2



3



Blockchain explorers

- Everybody with access to Bitcoin P2P network can analyze blockchain
 - Everybody running Bitcoin fullnode
 - All past transactions, human-readable visualizations, search for address...
 - Convenient quick check of funds send
- Third parties are operating public explorers (convenient, but privacy)
 - It is very important to use Tor Browser when accessing public block explorers
 - Explorer operator may log your IP address and transactions you are searching for and later sell it (chain surveillance companies)
 - Heuristic assumption that you are the owner of funds for searched transaction
- Ideally use your own full node with your own blockchain explorer
- Sparrow wallet allows you to visualize your transactions
 - Inputs, outputs, feed paid



Task: send some tBTC to your peer

- Select one of your neighbors as peer (PC1 and PC2)
- Obtain his/her receive address
 - Via messenger: PC2 → Receive tab → Copy address → send via Signal → PC1
 - Via QR: PC2 → Receive tab ; PC1 → Send → camera icon → scan address QR
- Enter some sats into Amount box
 - Observe visualized transaction below (more inputs may be added)
- Try again, but now with manual coin selection
 - UTXO tab → select one or more → Send Selected

PC1 (sender)

Sparrow - testnet4

File View Tools Help

testnet4 X wallet2_testnet4

Send

Pay to: + Add

Label:

Amount: sats \$ 66.19 Max

Fee

Range: 1 2 4 8 16 32 64 128 256 512 1024

Rate: 1.01 sats/vB High Priority

Fee: sats \$ 0.09

Target Blocks Mempool

Transaction

- 16995a93...:1
- to wallet2
- tb1q5szs...
- Fee

PC2 (receiver)

Sparrow - wallet2_testnet4

File View Tools Help

testnet4 X wallet2_testnet4 X

Receive

Address:

Label:

Derivation: m/84/1/0/0/0

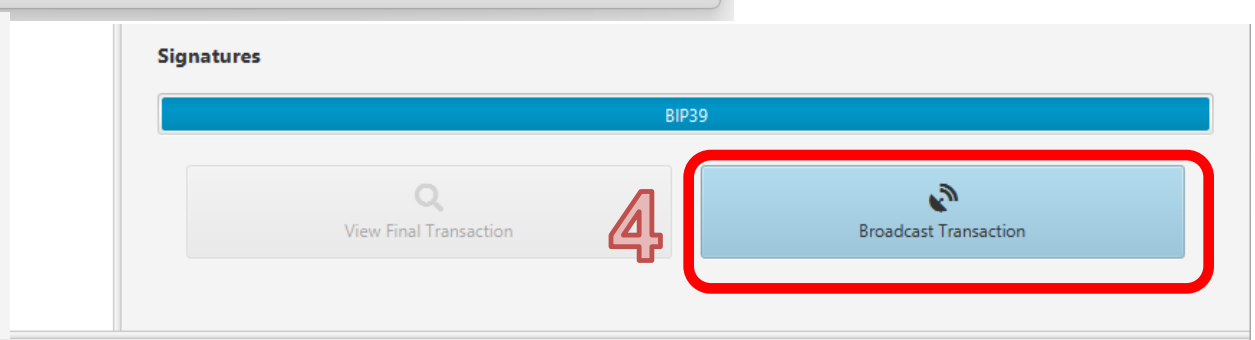
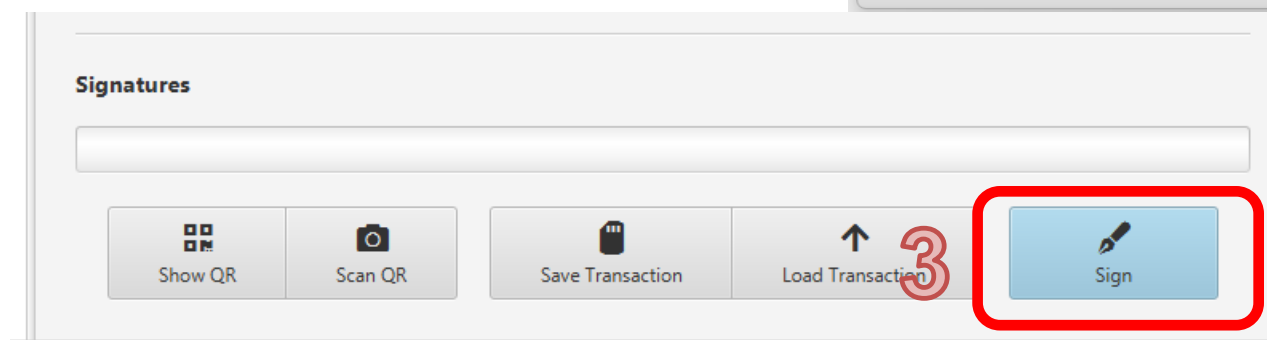
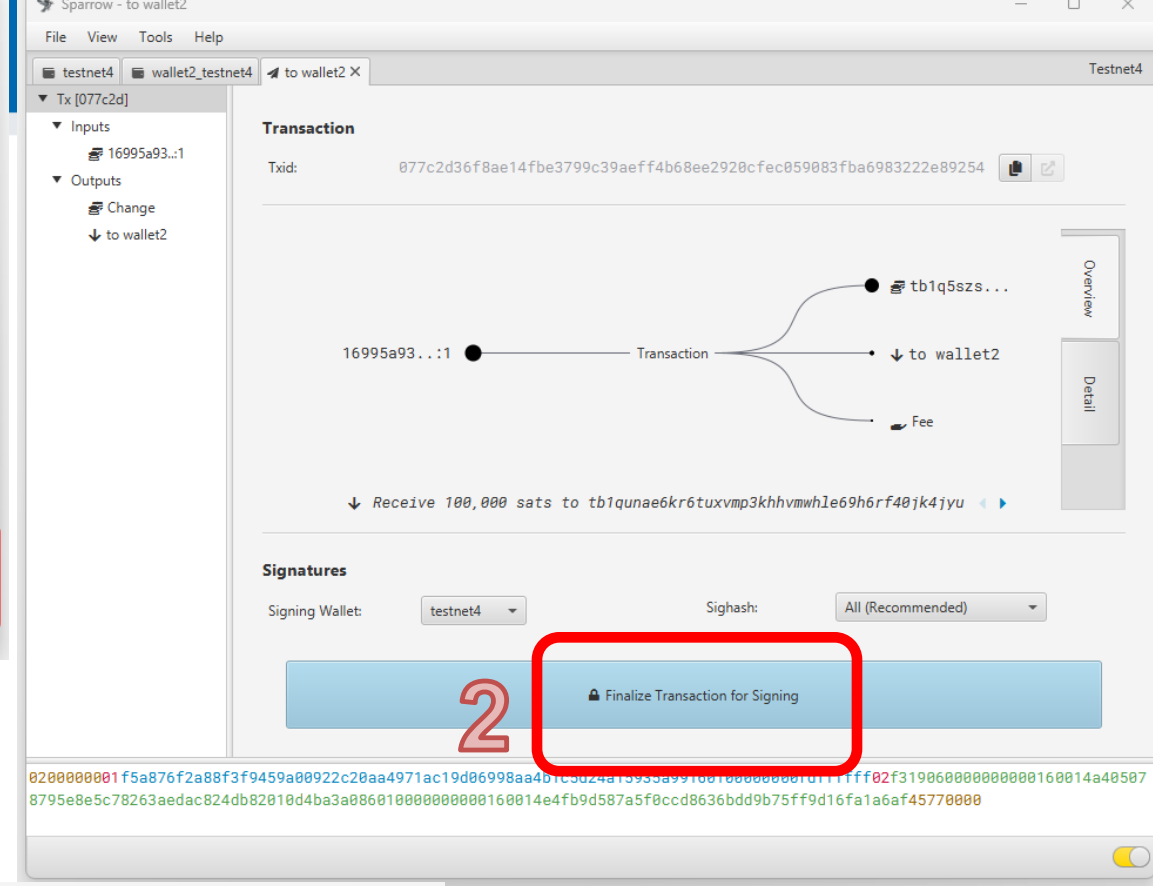
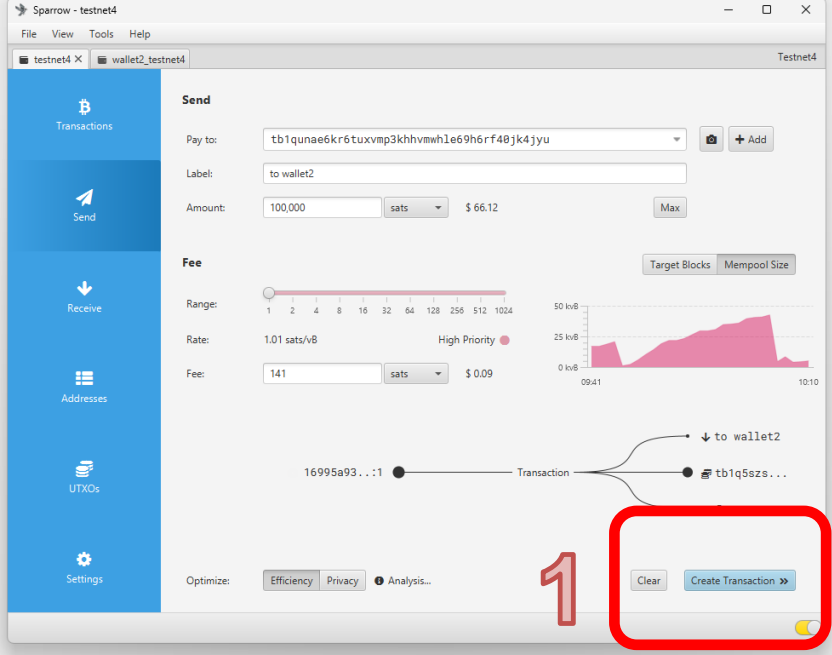
Last Used: Never

Required ScriptPubKey

Script:

Output Descriptor

Descriptor:



020000001 f5a876f2a88f3f9459a00922c20aa4971ac19d06998aa4b1c5d24a15935a99160100000000fdfffff02f3190600000000160014a405078795e8e5c78263aedac824db82010d4ba3a0860100000000160014e4fb9d587a5f0ccd8636bdd9b75ff9d16fa1a6af45770000

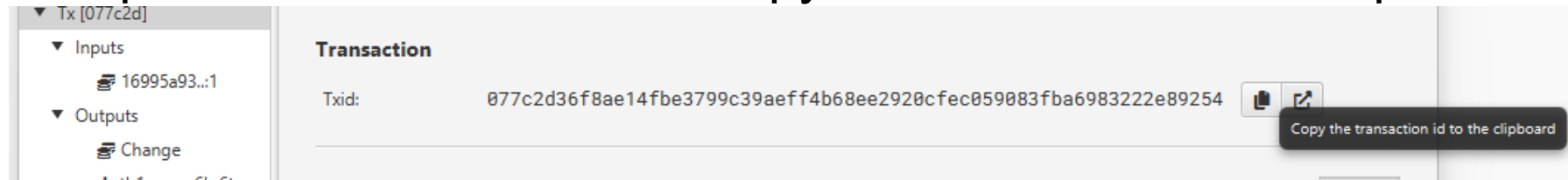
020000001 f5a876f2a88f3f9459a00922c20aa4971ac19d06998aa4b1c5d24a15935a99160100000000fdfffff02f3190600000000160014a405078795e8e5c78263aedac824db82010d4ba3a0860100000000160014e4fb9d587a5f0ccd8636bdd9b75ff9d16fa1a6af45770000

Input #0 witness #0 length

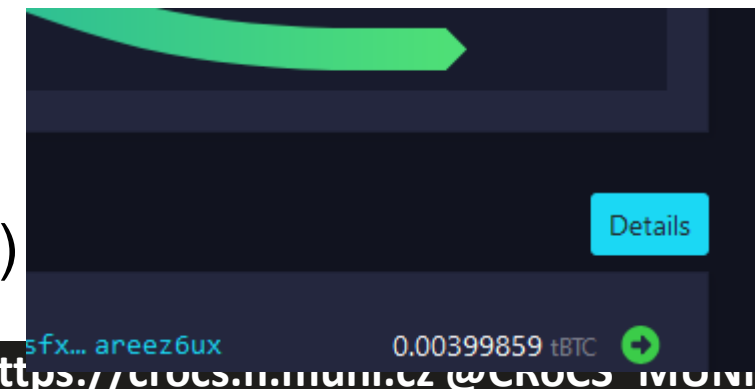
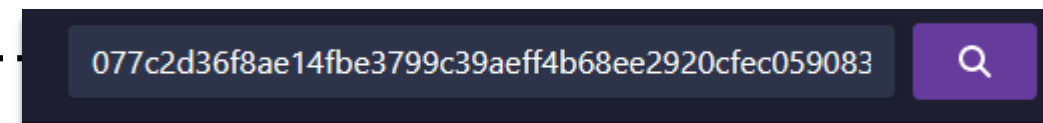


Task: Investigate your transaction

- Copy transaction id of the transaction you just send
 - Sparrow: Transactions → Copy the transaction id to clipboard



- Search for it using <https://mempool.space/testnet4> (upper right, search)
 - Basic information (block, time, fee, features...)
 - Visualized flows (inputs, outputs)
 - Details Inputs & Outputs (Details button)
 - Pay output, change output
 - Witness (proof), Lock scripts (future spend condition)



Sparrow - wallet2_testnet4

File View Tools Help

testnet4 wallet2_testnet4 X

Bitcoin Transactions

Send

Transactions

Balance: 100,000 sats \$ 66.24

Mempool: 100,000 sats \$ 66.24

Transactions: 1

Date	Label
Unconfirmed	to wallet2
Received to 177c...	to wallet2 (received)

077c2d36f8ae14fbe3799c39aeff...

Sparrow - [077c2d]

File View Tools Help

testnet4 wallet2_testnet4 [077c2d] X Testnet4

Tx [077c2d]

Inputs: 16995a93...:1

Outputs: Change, tb1qunae6kr6tu...

Transaction

Txid: 077c2d36f8ae14fbe3799c39aeff4b68ee2920cfec059083fba6983222e89254

Transaction diagram showing input 16995a93...:1 and outputs: tb1q5szs..., tb1qunae..., Fee.

Receive 100,000 sats to tb1qunae6kr6tuxvmp3khhvmwh1e69h6rf40j4jyu

Transaction: 077c2d36f8ae14fb... Bitcoin Network 8 Years Charts

https://mempool.space/testnet4/tx/077c2d36f8ae14fbe3799c39aeff4b68ee2920cfec059083fba6983222e89254

mempool.space

077c2d36f8ae14fbe3799c39aeff4b68ee2920cfec059083fba6983222e89254

Testnet4 Transaction

This is a test network. Coins have no value. Testnet4 is not yet finalized, and may be reset at any time. Go to "077c2d...2e89254"

Transaction details:

- 30535: ~0 sat/vB, 1-1 sat/vB, 0.00 tBTC, 2 transactions, In ~1 minute
- 30534: ~0 sat/vB, 0-1 sat/vB, 0.00 tBTC, 62 transactions, 14 minutes ago
- 30533: ~0 sat/vB, 1-2 sat/vB, 0.00 tBTC, 285 transactions, 16 minutes ago
- 30532: ~0 sat/vB, 1-2 sat/vB, 0.00 tBTC, 129 transactions, 37 minutes ago
- 30531: ~0 sat/vB, 1-1 sat/vB, 0.00 tBTC, 147 transactions, 57 minutes ago

Transaction

077c2d36f8ae14fbe3799c39aeff4b68ee2920cfec059083fba6983222e89254

Unconfirmed

First seen: [Progress bar]

ETA: [Progress bar]

Fee: 141 sat \$0.00

Fee rate: 1.01 sat/vB

Features: SegWit, Taproot, RBF


Flow

Hide diagram

Inputs & Outputs

4

Details

 `tb1qe2lgjs3v5vu6m72p5jnpu2n... mf0qpd5a` 0.00500000 tBTC

Witness

```
304402205730e21cf616c2b2cb76c739e28f082146c3ddfa0
2a6e3769dfe9458717b6b8c022075c20bcfab5518085023f8
37b48b0ba38fb25bdc86be0a890180a0b7c808379c01


0381282c8b8afddcdb553fe8ec53ab2a5d65f14f94822927a
8788a773e89d702f4
```

nSequence 0xffffffff

Previous output script

```
OP_0
OP_PUSHBYTES_20 cabe89422ca339adf941a4a61e2a75c0f
713c369
```

Previous output type V0_P2WPKH


`tb1q5szs0pu4arju0qnr4mdvsfx... areez6ux` 0.00399859 tBTC 

ScriptPubKey (ASM)

```
OP_0
OP_PUSHBYTES_20 a405078795e8e5c78263aedac824db820
10d4ba3
```

ScriptPubKey (HEX) 0014a405078795e8e5c78263aedac824db82010d4ba3

Type V0_P2WPKH

`tb1qunae6kr6tuxvmp3khhvmwh1... 40jk4jyu` 0.00100000 tBTC 

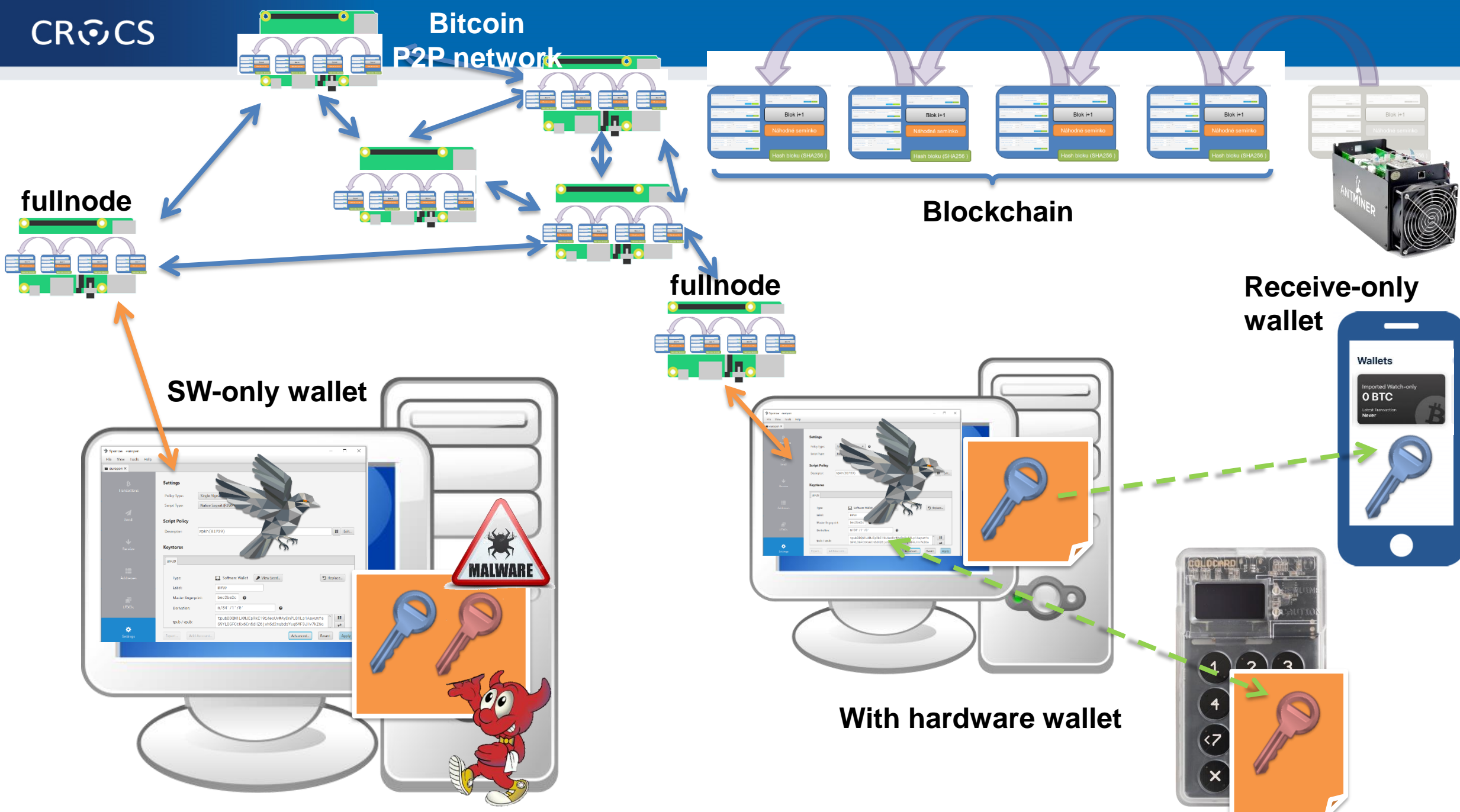
ScriptPubKey (ASM)

```
OP_0
OP_PUSHBYTES_20 e4fb9d587a5f0ccd8636bdd9b75ff9d16
fa1a6af
```

ScriptPubKey (HEX) 0014e4fb9d587a5f0ccd8636bdd9b75ff9d16fa1a6af

Type V0_P2WPKH

0.00499859 tBTC

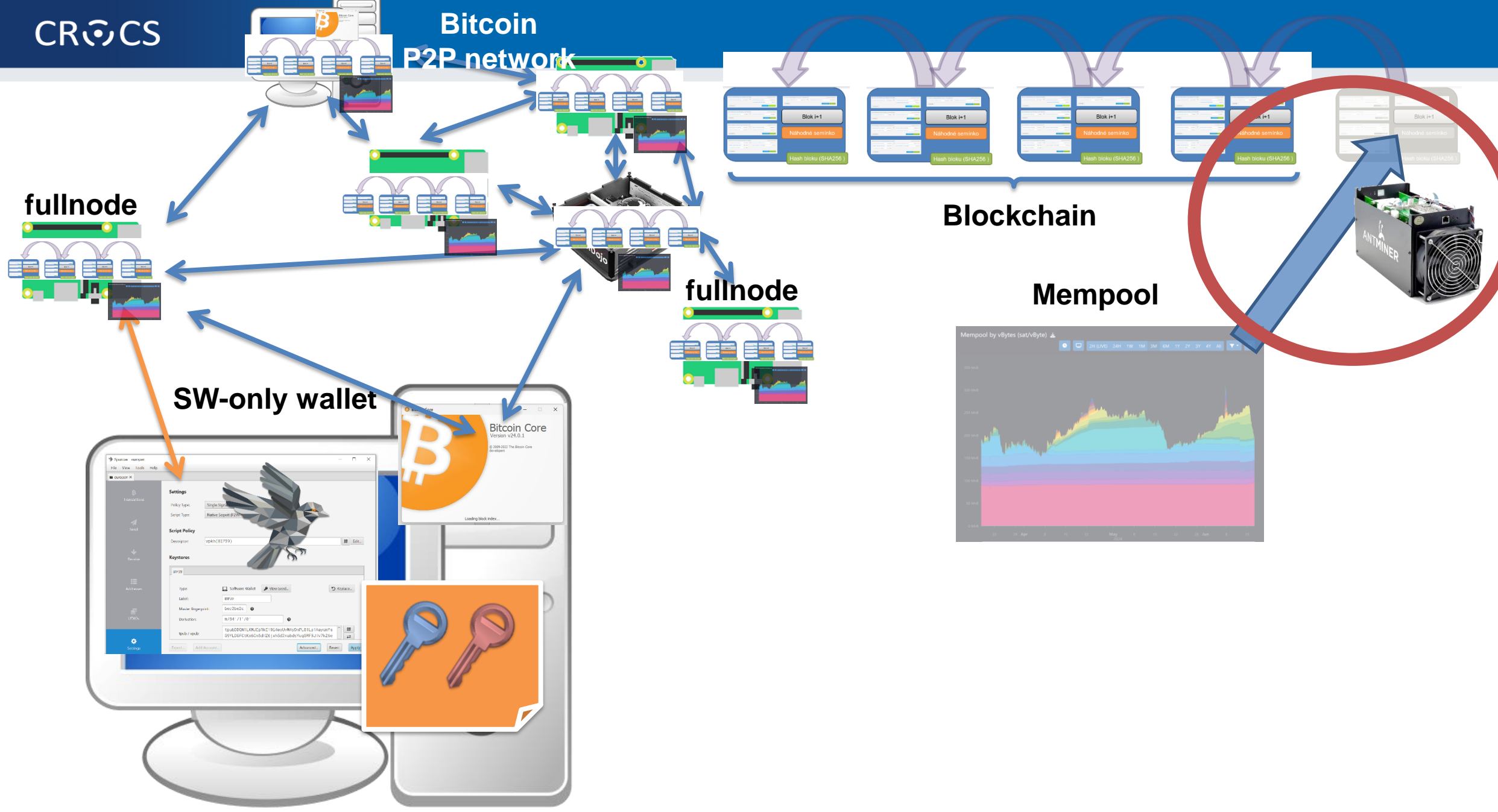


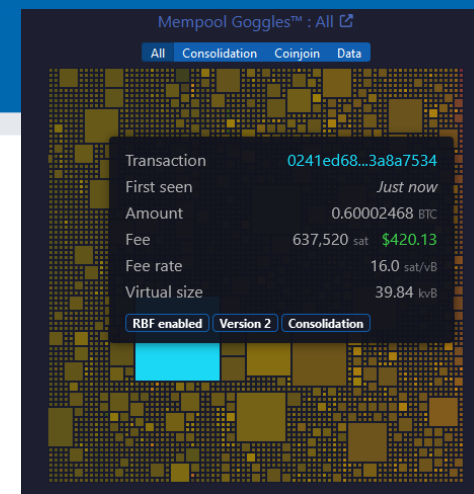
TASK: MINING BITCOINS

What you will learn:

- Who creates new blocks and why
- Why miners consume energy and why is it important
- How fast miners compute and what devices they use
- How to compute profitability of miner
- What is mining pool and solo-mining

Bitcoin P2P network





Bitcoin's Proof of Work (SHA256 function)

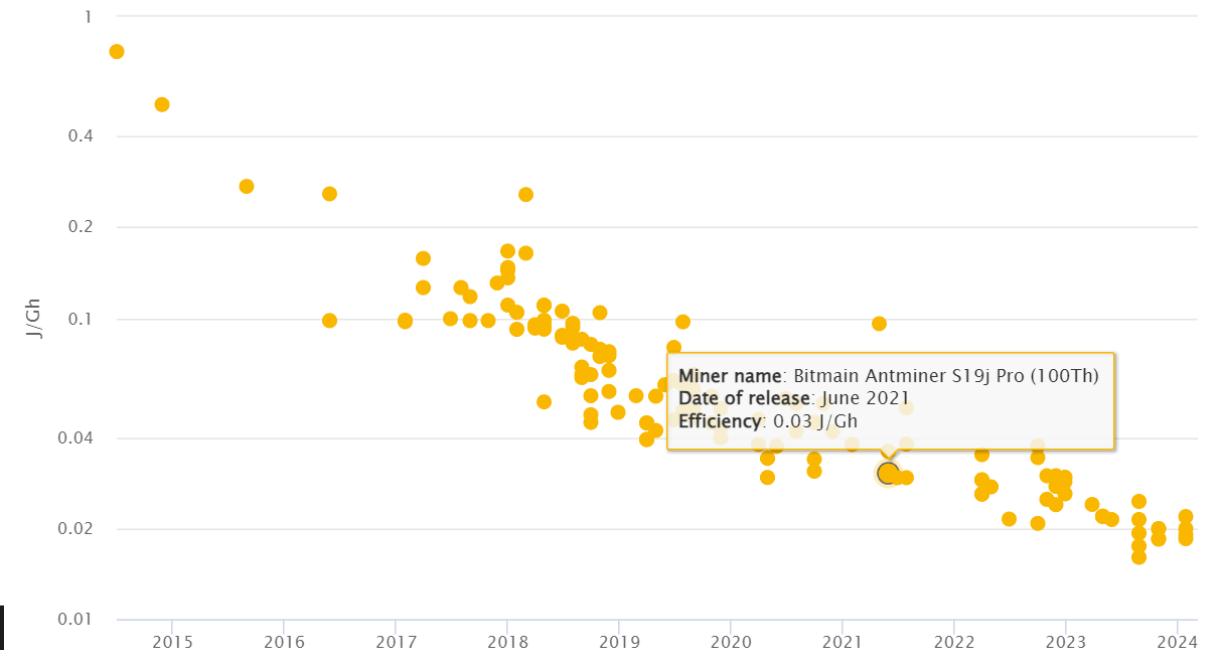
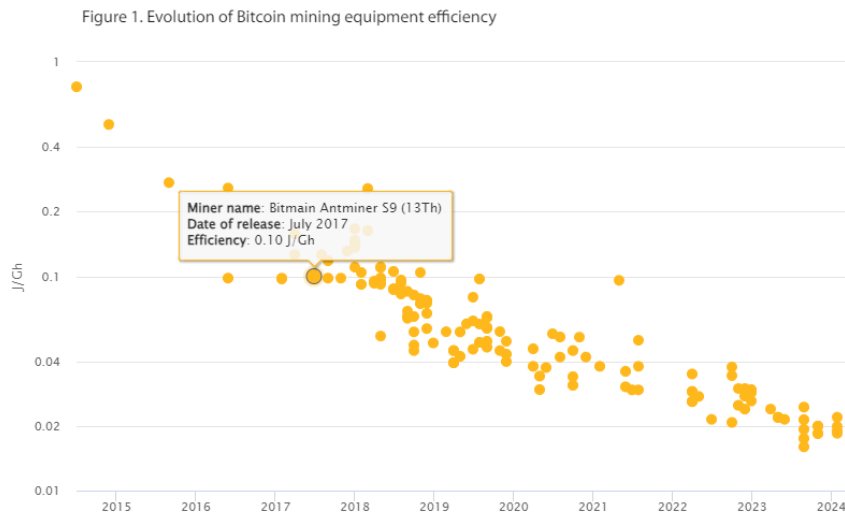
- Crucial for security of blockchain (no rewrite of history)
- Initially on CPU (Satoshi: “Everyone can participate 1 CPU 1 vote”)
- CPU → GPU → FPGA → ASIC
- Initially solo mining, later collaborative mining (too little chance alone)
- First mining pool: SlushPool in Prague (now Braiins Pool)
 - Miners join their hashrate, fraction of reward based on number of partial solutions
- Cambridge university centre for alternative finance (CBECEI)
 - Where are the miners? https://cbeci.org/mining_map/
 - More mining details: <https://cbeci.org/cbeci/methodology>

Demo – Bitmain Antminer S9 & S19 mining

- S9 efficiency: 80-100 J/TH per second (= 80-100W/TH), from 2017
- S19j efficiency (BMM100): 40 J/TH per second (~40W/TH), from 2021
- Connected to mining pool using Stratum v2 protocol

<https://ccaf.io/cbnsi/cbeci/methodology>

Figure 1. Evolution of Bitcoin mining equipment efficiency

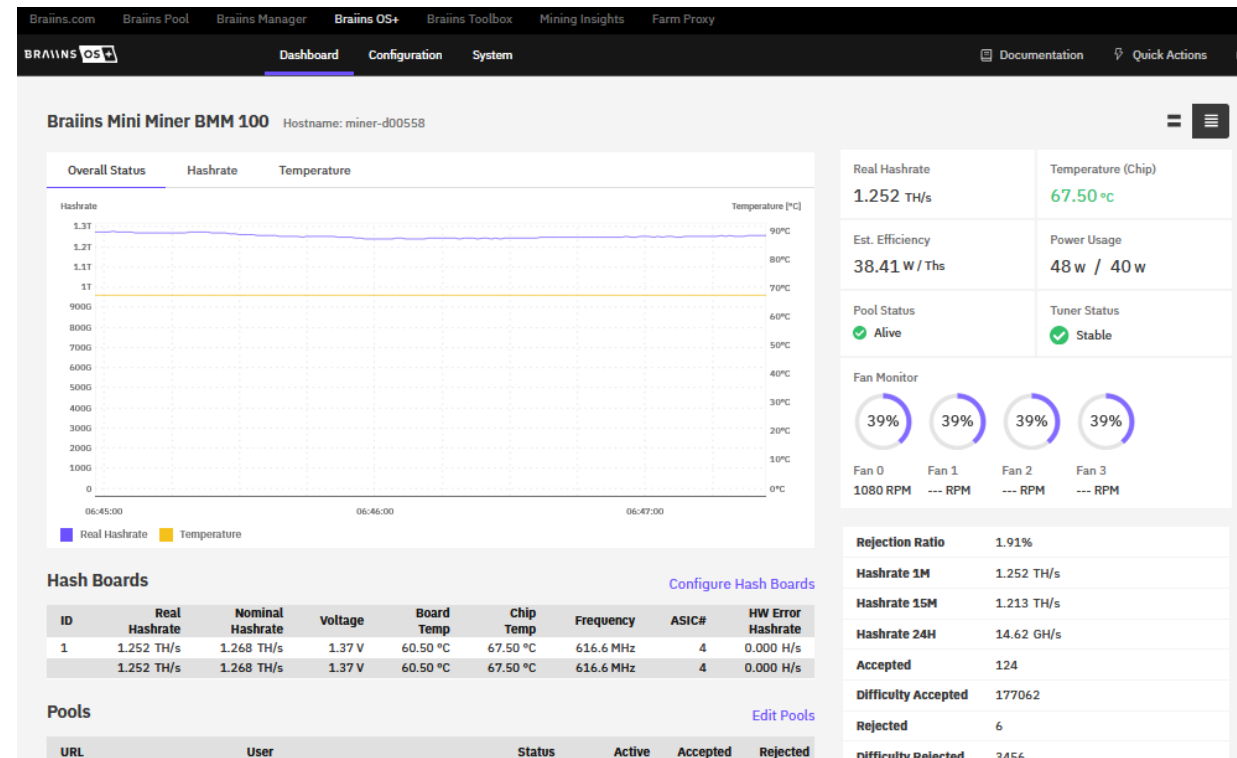
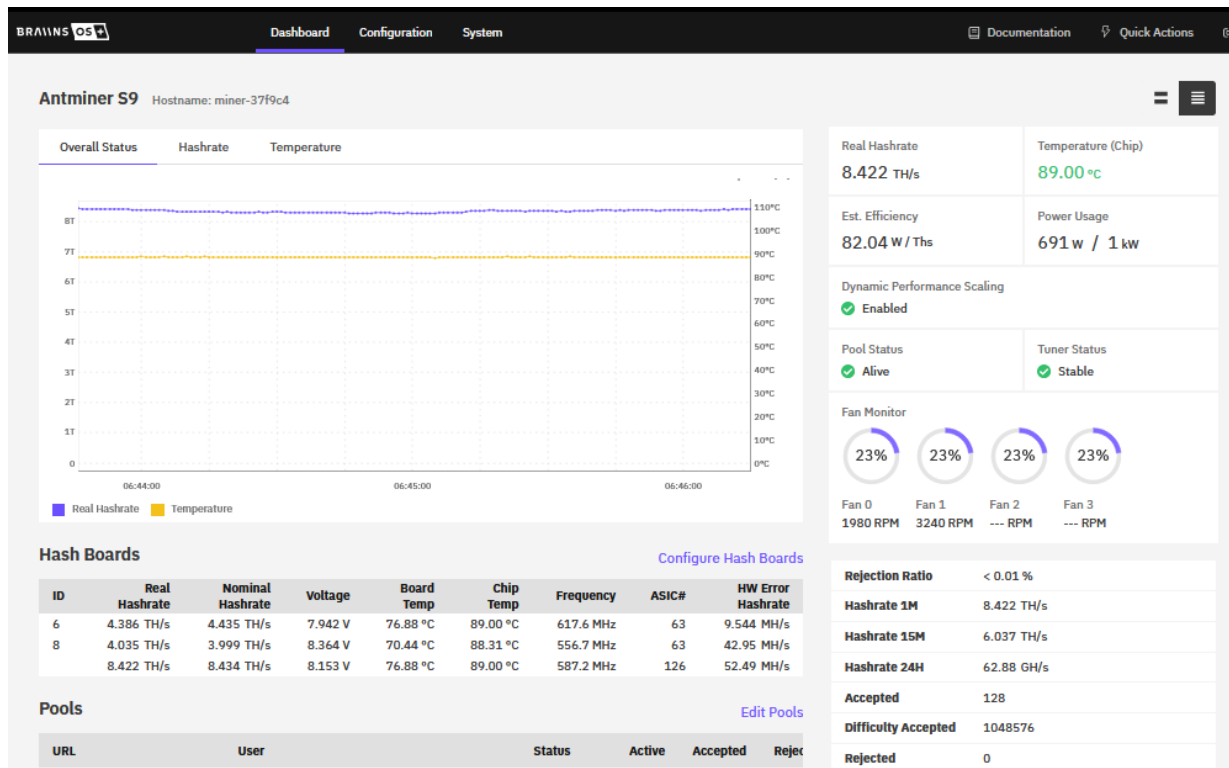


**Hashrate in time (>673EH/s = $6.7 \cdot 10^{20}$ hash/sec = 2^{67} /sec)
673,000,000,000,000,000,000x SHA256 computations per second!**



Bitcoin mining

- Highly optimized machines with down to 5nm ASICs
- Optimized for single task (SHA256), doing extremely fast (>1TH/s)



Are miners “stealing” electricity from poor people?

- Example: Braiins Mini Miner 100 has ~1.3TH/s @ 45W
 - Current reward is around 80 sats/day for 1TH/s (=> **1.22Kc / day** @ \$65k/btc)
 - Residential price 1kW = 7.2Kc https://www.cez.cz/edee/content/file/produkty-a-sluzby/obcane-a-domacnosti/elektrina-2024/moo/moo_etarif_01_2024_pre.pdf
 - Energy consumed per day = 24h * 45W = 1080Wh / day => **7.77Kc / day**
 - Miner would need to have price of 1kW = 1.13Kc/kWh not to be in loss
- Miners need cheap energy => energy “nobody” wants / can use
 - Energy which needs to be produced but energy grid cannot take it (solar, wind...)
 - Energy generated, but without enough consumers (hydropower in remote areas)
 - Methane vented from oil wells, produced by landfills...

TASK: USING LIGHTING NETWORK

What you will learn:

- Limitations of base layer (tx/sec, confirmation times...)
- Design principles of trust-minimized layer atop
- How are payments routed (network of payment channels)
- Operating custodial (WoS) and non-custodial (Mutiny) Lightning wallets
- Paying for real services

Base bitcoin layer (mainnet)

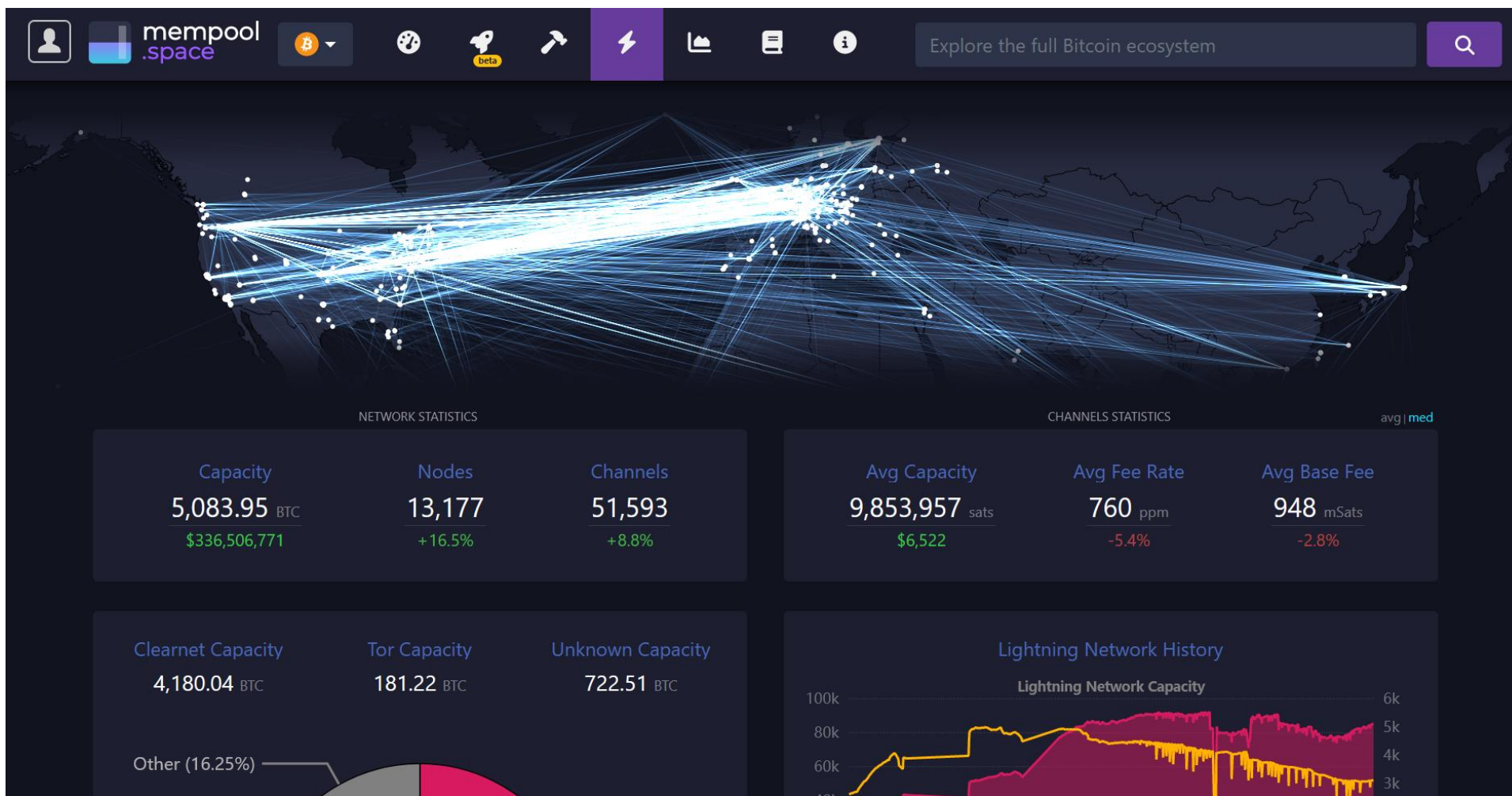
- Highest security and self-sovereignty achieved!
- Limitations and tradeoffs
 - Transaction thruptut (5-8 txs/sec)
 - Variable and relatively long waiting time
 - 10 mins on average, possibly multiple blocks
 - Variable transaction fees (1-600+ sats/vB => \$0.01-\$50)
 - And expected to increase in future (reward for miners)
 - Low privacy (transactions are recorded forever)
 - Mainnet is not for buying coffee!



Create payment transaction,
but broadcast it only if
disagreement

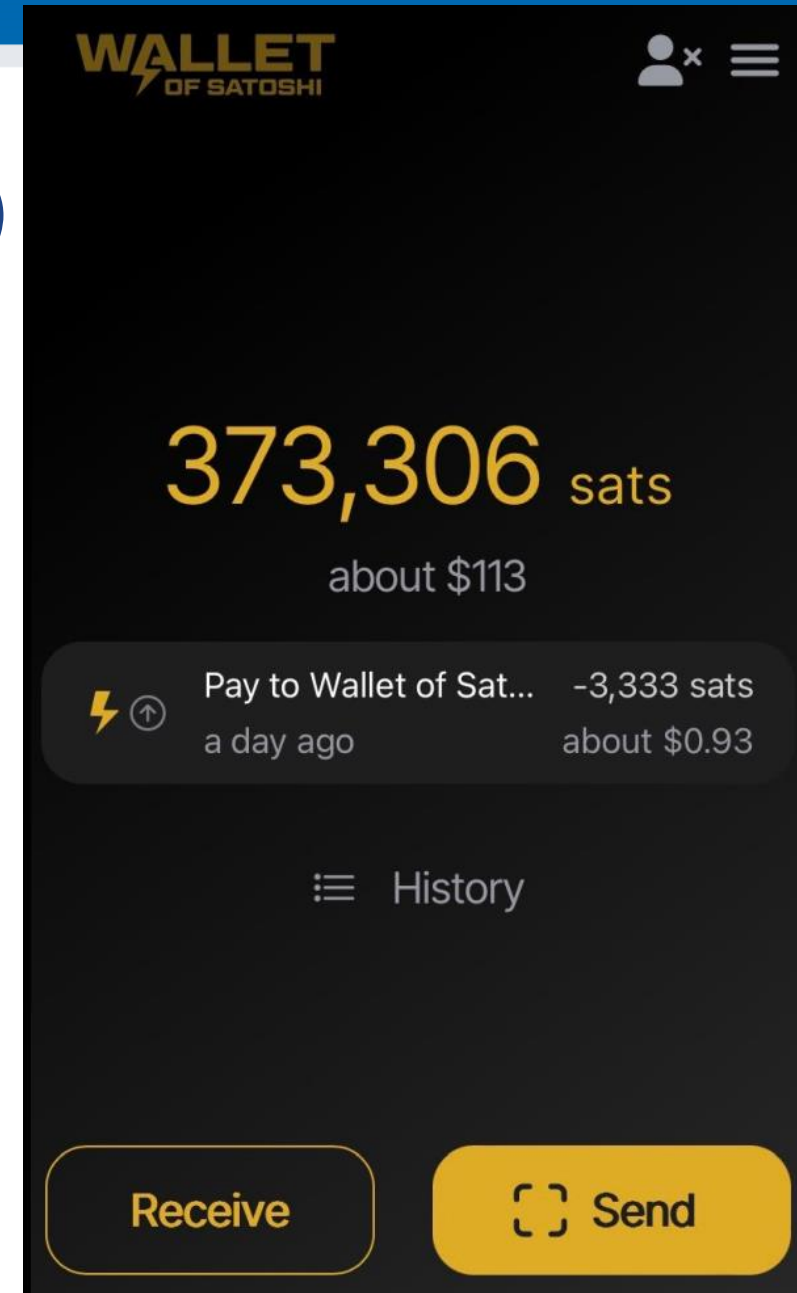


Lightning network <https://mempool.space/lightning>



Get some real sats (1sat = 100000000 ₿)

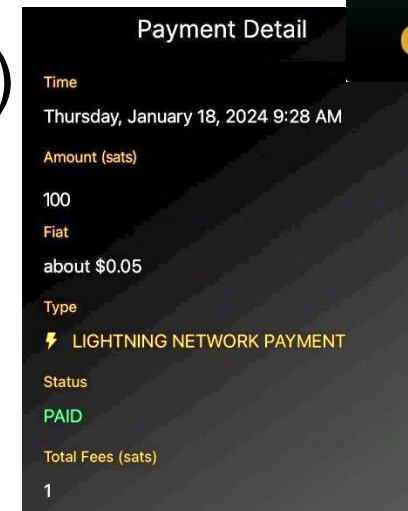
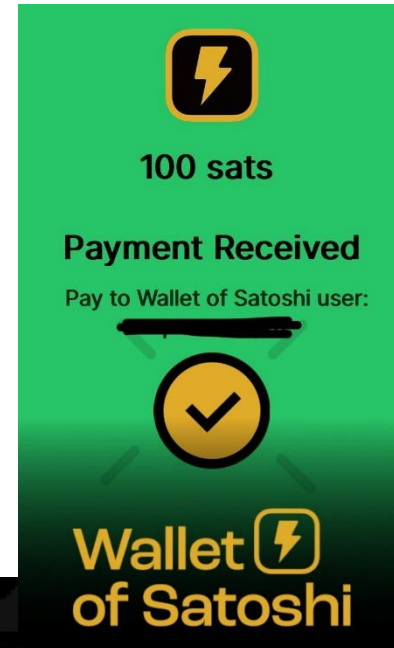
- You can get/buy fraction of bitcoin (sats)
 - Transaction on mainnet
 - Potentially costly, ~10mins or more to execute
 - Mainnet is not for buying coffee!
 - sats on Lightning – instant and near free
 - Still real bitcoins usable on Bitcoin network
1. Download Wallet of Satoshi (appstores)
 2. Click Receive → QRCode displayed
 3. Come to get some
 4. Try and learn





Task: send some lightning sats to your peer

- (Assumption: you already have some sats in Lightning wallet)
- Try to send between friends
 - Receiver click on 'Receive' button
 - Sender click on 'Send' button, scan QRCode, edit amount, confirm
- Enjoy instant payment
- Inspect Payment Detail (Time, Amount, Total Fees)



Using lighting to feed sheep <https://www.tanglesheep.com/>



Further reading

- Mastering Bitcoin (Andreas M. Antonopoulos and others)
 - <https://github.com/bitcoinbook/bitcoinbook>
- Programming Bitcoin (Jimmy Song)
 - <https://github.com/jimmy song/programmingbitcoin>
- List of interesting resources
 - <https://blockonomi.com/bitcoin-educational-resources/>
 - <https://learnmeabitcoin.com/>, <https://learnmeabitcoin.com/technical/>
- Bitcoin Twitter, Nostr (<https://nostr.com/clients>)
 - [@adam3us](#) [@gladstein](#) [@ODELL](#) [@saylor](#) ...
- Podcasts
 - <https://www.whatbitcoindid.com/> <https://stephanlivera.com/>

