What the Heck is a Bitcoin?

Strictly Technical

Today’s Topics

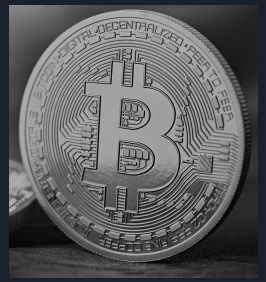
- Intro / Concepts
- Elliptic Curve Cryptography / Hash Functions
- Blockchain / Proof-of-Work
- Transactions / Addresses / Wallets
- Node Network
- Block Explorers / Demo
- Other Topics

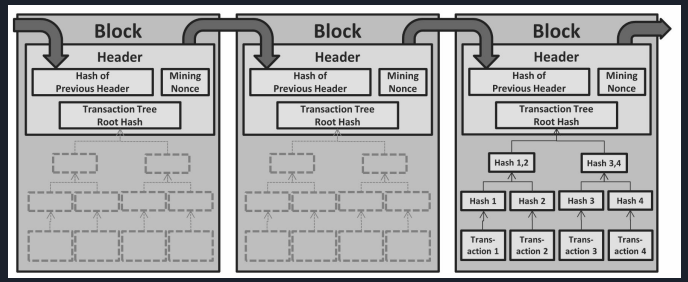
Scope

- Many dimensions to the topic area
  (technology, security, economics, energy, human rights, etc.)
- Deep rabbit hole
- Focusing on technical aspects of Bitcoin today
- Discussion of non-technical aspects welcome afterward

What is Bitcoin?

- Digital currency enabling transfer of value over a network, and with relatively rapid final settlement. Cryptographic, permissionless, trust-minimalized
- P2P network of nodes validating the cryptography of transactions
- Distributed ledger database of transaction history (blockchain)



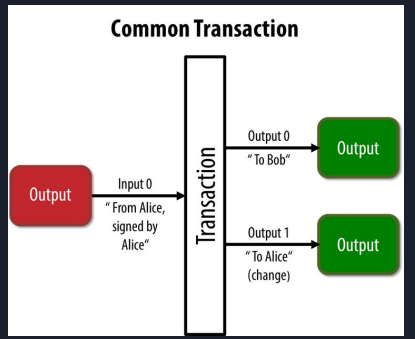


Yeah ok, but what actually is a Bitcoin?

- Example: Someone has 4¢, they hold four pennies
- Each penny has a transaction history (e.g. the store that gave you change, the bank that cashed a check, etc.)
- All Bitcoin transaction history is stored in the Bitcoin blockchain
- Bitcoin “coins” are UTXOs (unspent transaction outputs)

Yeah ok, but what actually is a Bitcoin?

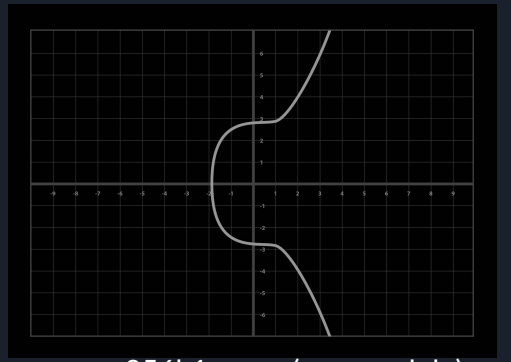
- Cents are to dollars as Satoshis are to Bitcoins
  (1 Bitcoin = 100,000,000 Satoshis)
- Having Bitcoin means having control of the cryptographic private keys that correspond to UTXOs
- Bitcoin is spent to an address, which corresponds to a UTXO
  (e.g. bc1q2fkptnv8n7zndlh66ct4anqprktmn98k0z6fav)



Elliptic Curve Cryptography / Hash Functions

Elliptic Curve Cryptography

- A form of public key cryptography (private key and corresponding public key)
- secp256k1 elliptic curve is used in Bitcoin for controlling coins and to transfer coins (with digital signatures)
- A private key (k) is a 256-bit number (~10^{77} possible values)
- A public key (P) is a point on the elliptic curve (P = kG) (G is the generator point defined for secp256k1)

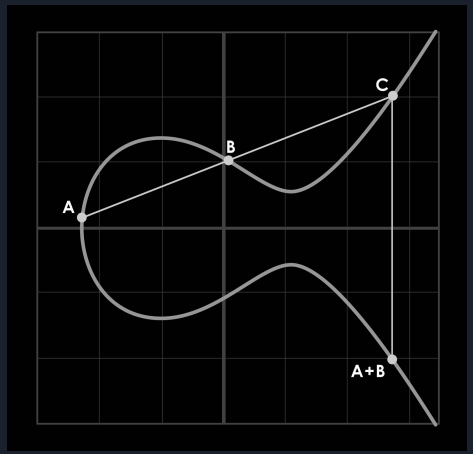


secp256k1 curve (non-modulo)

$y^2 = x^3 + 7$

Elliptic Curve Cryptography

- Security relies on the Discrete Logarithm Problem (k→P is easy, P→k is difficult)
- Point addition is defined geometrically (A + B)
- Point multiplication is defined from addition (P+P = 2P, P+P+P=3P, etc.)
- Digital signature algorithms are used with public and private keys to prove authenticity of messages
- Bitcoin uses ECDSA and Schnorr signatures to secure transactions (UTXOs can’t be spent without having the corresponding private key)



Point Addition

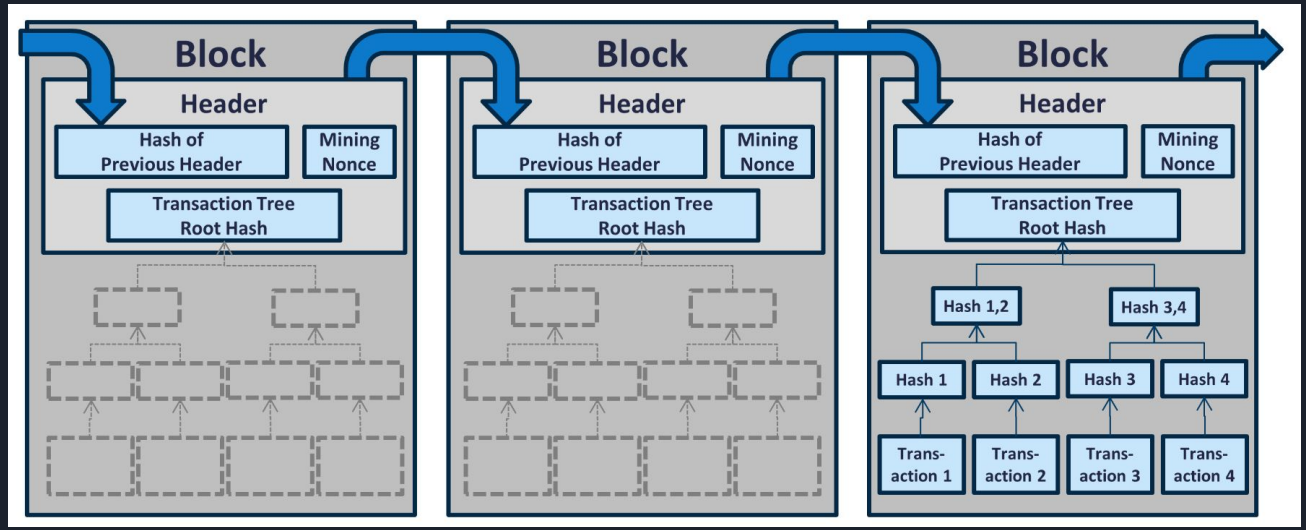
Cryptographic Hash Functions

- Cryptographic hash functions take arbitrarily sized data as input, and produce a fixed size output (256-bit number in the case of SHA-256)
- One-way functions. Easy to generate hash from input, difficult to generate input from hash
- Bitcoin uses the SHA-256 and RIPEMD-160 hash functions for transaction digital signatures, for Proof-of-Work, and for address generation

Blockchain / Proof-of-Work

Blockchain

- A Bitcoin transaction is included in a block
- Each block includes a hash of the previous canonical block



Proof-of-Work (PoW)

- New blocks are only accepted if they have valid PoW
- Block header includes a nonce that ensures the block header hashes to a value below the “target”
- Finding a valid nonce is like rolling dice until a value under the target is achieved (e.g. keep rolling until 2 or below). Known as “mining”
- Target is adjusted every 2016 blocks (~every two weeks), which allows for changes in global network hashrate, while maintaining an average 10 minute block time
- Forking does occur, but the canonical chain is the one with the most PoW

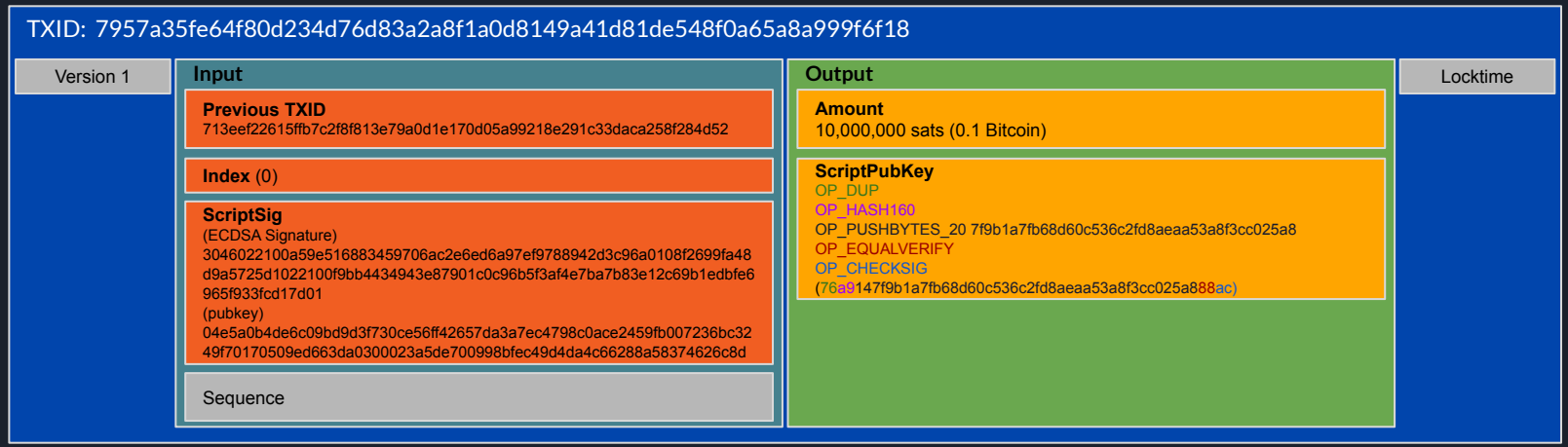
Proof-of-Work (PoW)

- A new block solution is found approx every 10 minutes on average
- Finding the block solution rewards the miner with a block subsidy (currently 6.25 Bitcoin + fees from transactions in the block).
- Every ~4 years (210,000 blocks), the block subsidy is halved (50 → 25 → 12.5 → 6.25 → 3.125…). Called “the halvening” (next one in 2024)
- Block subsidy ends in ~2140, after which reward will be fees alone

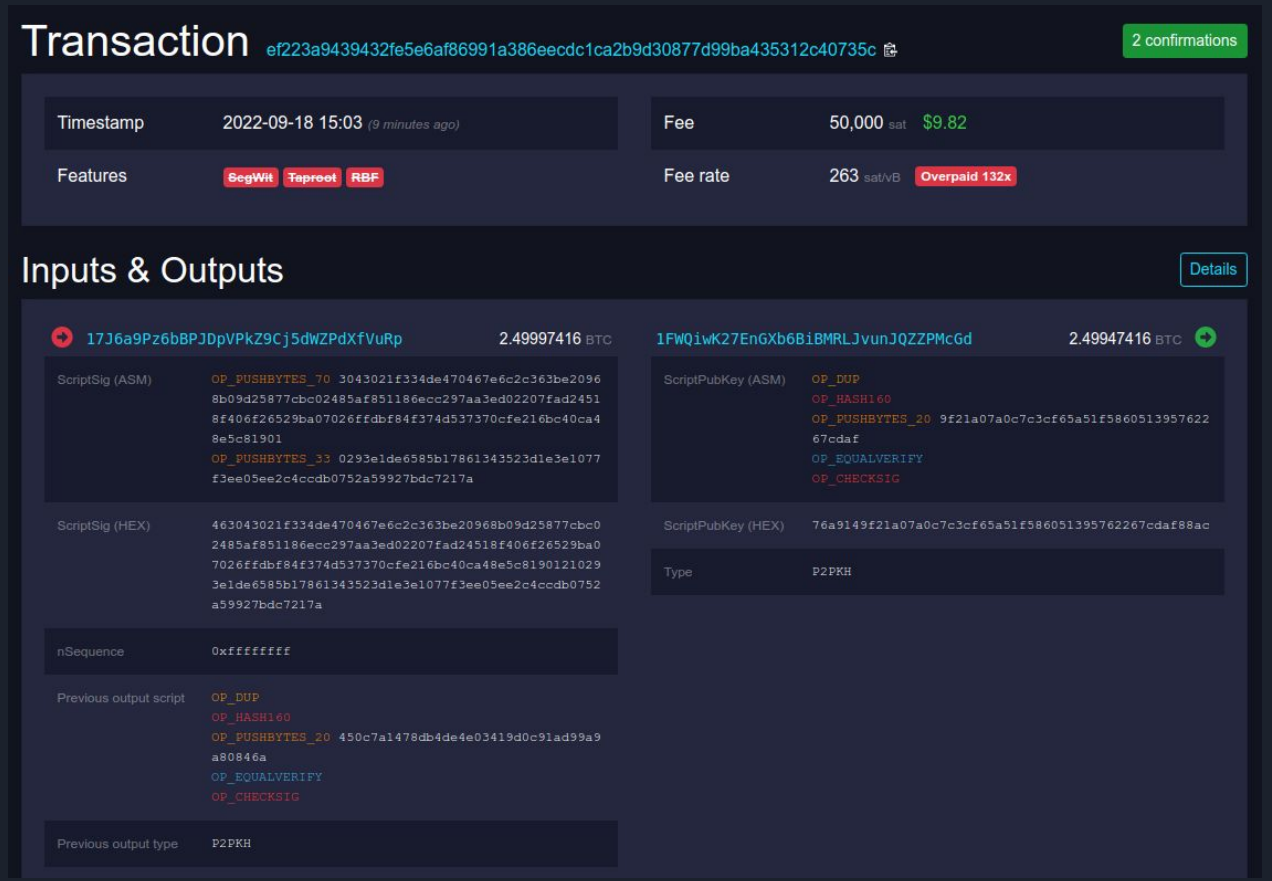
Transactions / Wallets

Transactions

- UTXO model: Unlike the account model, having Bitcoins means owning the private keys associated with unspent transaction outputs
- Transactions (simplified)
  - One or more inputs (prev TXID, index, signature)
  - One or more outputs (amount, locking Script)



Transactions



Transaction ef223a9439432fe5e6af86991a386eecdc1ca2b9d30877d99ba435312c40735c

2 confirmations

Timestamp	2022-09-18 15:03 (9 minutes ago)	Fee	50,000 sat $9.82
Features	SegWit Taproot RBF	Fee rate	263 sat/vB Overpaid 132x

Inputs & Outputs

Details

17J6a9Pz6bBPJDpVPkZ9Cj5dWZPdXfVuRp 2.49997416 BTC

ScriptSig (ASM)	OP_PUSHBYTES_70 3043021f334de470467e6c2c363be20968b09d25877cbc02485af851186ecc297aa3ed02207fad24518f406f26529ba07026ffdbf84f374d537370cfe216bc40ca48e5c81901 OP_PUSHBYTES_33 0293e1de6585b1786134352 3d1e3e1077f3ee05ee2c4ccdb0752a59927bdc7217a
ScriptSig (HEX)	463043021f334de470467e6c2c363be20968b09d25877cbc02485af851186ecc297aa3ed02207fad24518f406f26529ba07026ffdbf84f374d537370cfe216bc40ca48e5c81901210293e1de6585b1786134352 3d1e3e1077f3ee05ee2c4ccdb0752a59927bdc7217a
nSequence	0xffffffff
Previous output script	OP_DUP OP_HASH160 OP_PUSHBYTES_20 450c7a1478db4de4e03419d0c91ad99a9a80846a OP_EQUALVERIFY OP_CHECKSIG
Previous output type	P2PKH

1FWQiwK27EnGXb6BiBMRLJvunJQZZPMcGd 2.49947416 BTC

ScriptPubKey (ASM)	OP_DUP OP_HASH160 OP_PUSHBYTES_20 9f21a07a0c7c3cf65a51f586051395762267cdaf OP_EQUALVERIFY OP_CHECKSIG
ScriptPubKey (HEX)	76a9149f21a07a0c7c3cf65a51f586051395762267cdaf88ac
Type	P2PKH

Transactions

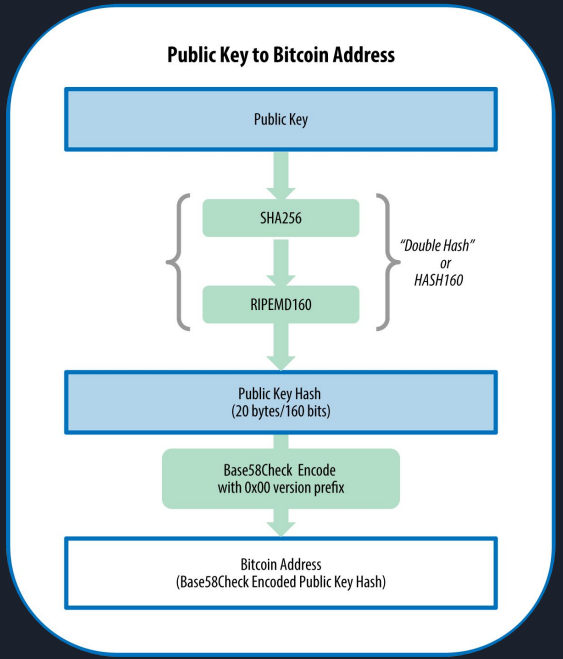
- Script: ScriptSig (unlocking) + ScriptPubKey (locking) scripts are executed on a stack, and if execution is successful, transaction is valid

- Fees: Implicit (sum of input amounts - sum of output amounts)

- Different spend types available for different needs (different ScriptPubKey):
  - P2PKH (Pay to Public Key Hash): Plain single signature spends
  - P2SH (Pay to Script Hash): Typically used for multisignature spends
  - P2WPKH (Pay to Witness Public Key, Segwit): Newer singlesig spends
  - P2WSH (Pay to Witness Script Hash, Segwit): Newer multisig spends, Lightning
  - P2TR (Pay to Taproot): Newest. Very flexible. Singlesig, multisig, Lightning, other

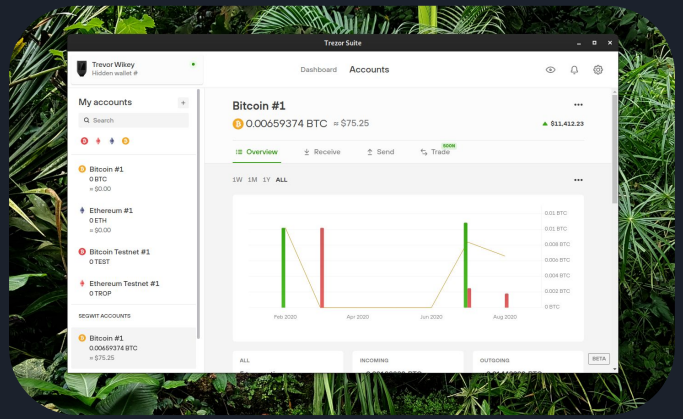
Bitcoin Addresses

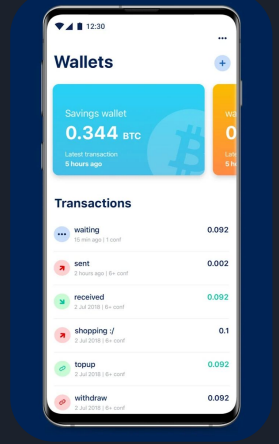
- Someone sends Bitcoin to a Bitcoin address by creating, signing, and broadcasting a transaction
- ScriptPubKey of transaction can be constructed from the decoded Bitcoin address
- Example addresses:
  P2PKH: 1Cdid9KFAaatwczBwBttQcwXYCpvK8h7FK
  P2SH: 3C8VU5C7emhNiLet1CwJb3WmNV2nmEm1y5
  P2WPKH: bc1qqsxhjhj6m52m00c6yqr95ecsgzdwhzneyjhyz7
  P2WSH: bc1qrp33g0q5c5txsp9arysrx4k6zdkfs4nce4xj0gdcccefvpysxf3qccfmv3
  P2TR: bc1p0xlxvlhemja6c4dqv22uapctqupfhlxm9h8z3k2e72q4k9hcz7vqzk5jj0
- For convenience, QR codes are common



Wallets

- A “wallet” is a program that manages private keys, and interfaces with a node to learn of UTXOs and spend them
- “Cold storage”: Private keys are never stored on an internet-connected device
- “Hot/warm storage”: Private keys are stored on an internet-connected device
- Singlesig: One signature required to spend UTXO
- Multisig: Threshold of signatures required to spend UTXO (e.g. 2 of 3)

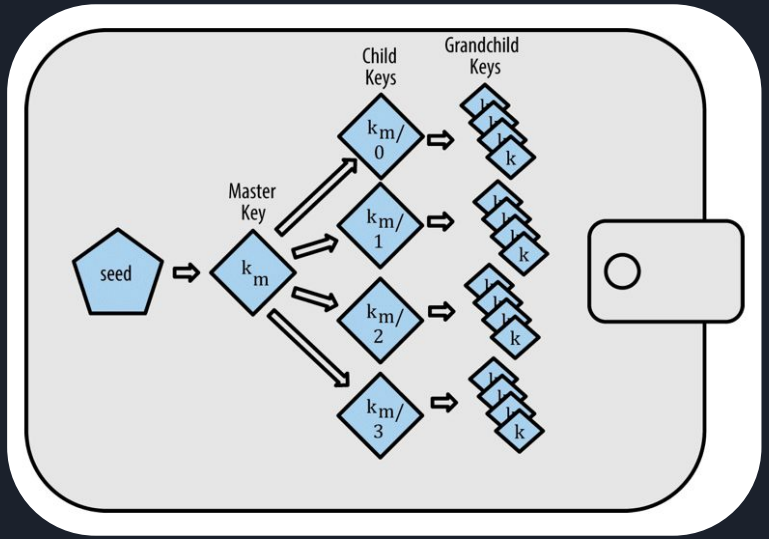




Wallet Backup

- BIP 39 (Mnemonic code for generating deterministic keys)
- 12 or 24 english lowercase words from a dictionary of 2048 words
- Child keys are derived using a tree (hierarchical deterministic, BIP 32)

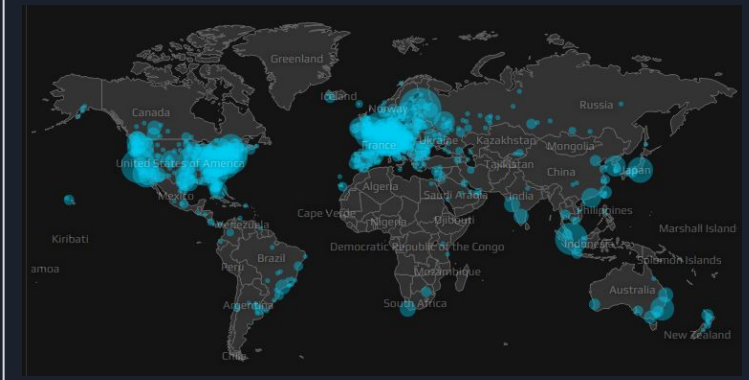
outer (1) blossom (2) already (3) begin (4) suggest (5) dragon (6) disease (7) turtle (8) kitten (9) act (10)
rate (11) modify (12) nation (13) snack (14) decorate (15) regret (16) roast (17) marble (18) ginger (19) harvest (20)
enrich (21) fox (22) assault (23) raccoon (24)



Node Network

Network of Nodes

- A global network of computers running Bitcoin node software (Bitcoin Core, btcd, etc.)
- Nodes receive transactions, and broadcast valid transactions to other nodes
- Transactions are checked against well-defined Bitcoin consensus rules (backwards compatible)
- Miners submit solved blocks to nodes
- Nodes share blocks to converge on canonical blockchain (most PoW)



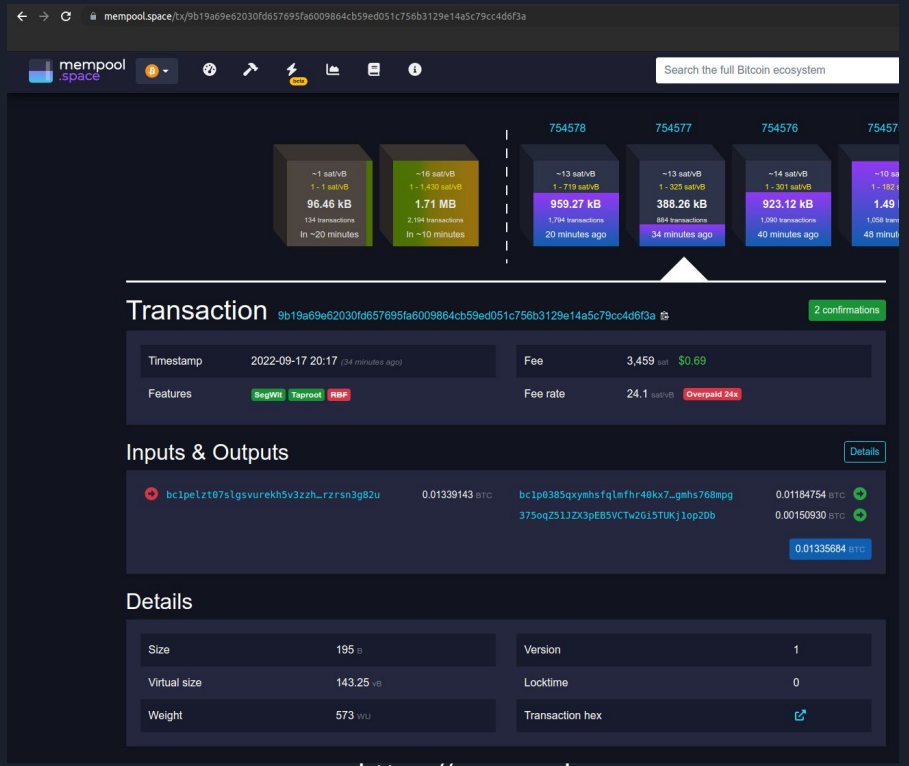
NODES	COUNTRIES	CITIES
45215	145	5742

https://bitnodes.io

Demo / Block Explorer

Blockchain Explorers

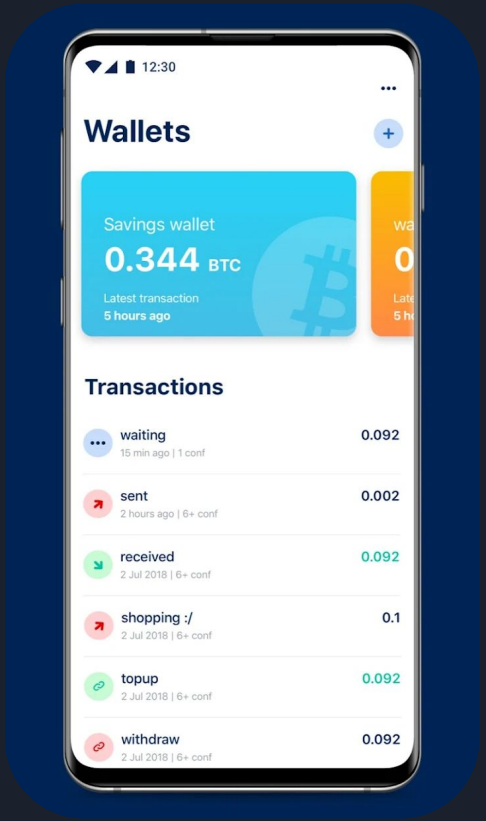
- Current and historical transactions
- Fee estimation
- UTXOs by address
- Run your own to increase privacy



https://mempool.space

Demo

- Sending from one wallet to another
- Viewing transaction in block explorer



Other Topics

Diving Deeper

- 9-page whitepaper (https://bitcoin.org/bitcoin.pdf)
- Security (good opsec, understanding assumptions, attack vectors, etc.)
- Run a Bitcoin Core node (currently ~600GB HDD/SSD, minimal CPU, 4GiB RAM)
- Learn the data structures and transaction types (Merkle trees, Taproot, etc.)
- Write an app using a Bitcoin library (e.g., https://github.com/buidl-bitcoin/buidl-python/)
- Create transactions on the free test networks (Testnet3 and Signet)

Diving Deeper

- Learn multisig (multiple private keys protect a UTXO)
- Create hardware for more secure wallets
- Learn about the Lightning network (Layer 2 network on top of the Bitcoin network that enables ~1 second transactions)
- Economics of Bitcoin (e.g. 21 million Bitcoin supply issuance schedule, permissionless, censorship resistance, bearer asset, properties of sound money, inflation, gold standard, debt-based economy, etc.)
- History (genesis block headline, great financial crisis, history of fiat currency, commodity-backed money, etc.)

Diving Deeper

- Learn about PoW incentivization of transition to renewable energy (demand response, ROI) and greenhouse gas reduction (e.g., methane flaring and landfill methane conversion)
- Books (Programming Bitcoin by Jimmy Song, Mastering Bitcoin by Andreas Antonopoulos)
- Online reading (https://learnmeabitcoin.com/)
- Read Bitcoin Improvement Proposals (BIPs) (https://github.com/bitcoin/bips), analogous to IETF RFCs
- Advanced topics (Discreet Log Contracts, Hash Time Locked Contracts)
- Contribute to open source software

Thank you

Questions?

Backup Slides

Transaction in Additional Detail