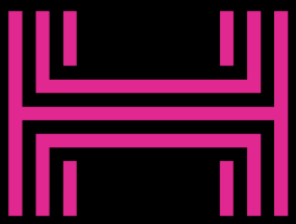
:~$ Scalar School

Bitcoin Technology

Foundations and
Career Paths
for Developers



Human
Rights
Foundation

TO ALL THE HATERS AND DENIERS

WHO SAY BITCOIN IS USELESS OR DANGEROUS

BITCOIN IS OUT THERE, EVERY DAY, EVERY HOUR, IN COLLAPSED ECONOMIES AND IN THE DARKEST CORNERS OF THE WORLD, HELPING PEOPLE THAT NO ONE ELSE IS WILLING TO HELP. BEING A LIFELINE FOR *ANYONE*

FIAT MONEY HAS FAILED MOST PEOPLE ON EARTH BADLY. OUR MONETARY AND "AID" SYSTEMS EXPLOIT AND REPRESS INSTEAD OF DELIVERING PROSPERITY AND FREEDOM

BUT BITCOIN IS SLOWLY, STEADILY, UNITING, ENRICHING AND LIBERATING THE WORLD

—ALEX GLADSTEIN

I dedicate this work to the Human Rights Foundation.

1. Why Bitcoin Matters

"A mysterious new technology emerges, seemingly out of nowhere, but actually the result of two decades of intense research and development by nearly anonymous researchers. Political idealists project visions of liberation and revolution onto it; establishment elites heap contempt and scorn on it. On the other hand, technologists – nerds – are transfixed by it. They see within it enormous potential and spend their nights and weekends tinkering with it.
Eventually mainstream products, companies and industries emerge to commercialize it; its effects become profound; and later, many people wonder why its powerful promise wasn't more obvious from the start. What technology am I talking about? Personal computers in 1975, the Internet in 1993, and – I believe – Bitcoin in 2014."

—Marc Andreessen, Founder of Netscape, 2014

1.1 Introduction to Bitcoin Technology

One way to think of Bitcoin is as a sequence of atomic transactions. Each transaction is authenticated by a sender with the solution to a previous cryptographic puzzle that was stored as a script, using the Bitcoin Script programming language.

The new transaction is locked to the recipient with a new cryptographic puzzle that is also stored as a script, whose new owner possesses the solution to unlock future transfers.

Each transaction is recorded in the global public and immutable ledger, the blockchain. Only the owner of the private keys that generated the address where the bitcoins are located can unlock the value to be moved forward.

We use "Bitcoin" with an uppercase B when referring to the network, the system itself. "bitcoin" with a lowercase b refers to the currency, the values transferred through this system.

1.2 Attributes of the Bitcoin Network

Security: The Bitcoin network is secure as long as more than 50% of participating nodes are honest.
Reliability: The network state is maintained by all nodes, which distribute the ledger copy among themselves.
Decentralization: All network nodes replicate transaction records, ensuring data distribution.
Peer-to-Peer Transactions: Allows direct transactions between users without intermediaries or central authorities.
Accessibility: Anyone can join or leave the network, validate transactions, or mine new coins at any time.

Transparency: All transactions are publicly verifiable and available to all network operators.
Permissionless: No credentials, identifications, or authorizations are required to participate in the network.
Globality: The network operates without geographical restrictions and can be accessed from almost anywhere in the world.
Censorship Resistance: There is no central authority that can prevent fund transfers between users.
Neutrality: The network is indifferent to who, what, when, where, or why you are sending and receiving bitcoin.
Inelastic and Distributed Supply: Bitcoin generation is self-regulated by mathematical algorithms and game theory models, ensuring a fair and predictable process.

1.3 Bitcoin and Human Rights

Despite short-term Bitcoin volatility and the need for digital literacy for full use, Bitcoin offers significant benefits for human rights and financial inclusion, especially where traditional alternatives are limited or ineffective. Here's why Bitcoin matters for human rights.

Alternative to the Current System: Most of humanity's dramas, such as endless wars, are due to money being manipulable, arbitrarily printed, and allocated for such purposes without the general population's consent. Bitcoin solves this with its transparent monetary policy and inelastic supply defined by the community at the code level.

Financial Inclusion for the Unbanked: Approximately 1.7 billion people worldwide lack access to traditional banking services. Bitcoin offers a viable alternative, requiring only internet access and a mobile device to participate in the financial network. In regions like Sub-Saharan Africa, where banking infrastructure is limited, Bitcoin usage is growing as a way to access financial services.

Financial Freedom and Protection from Oppressive Regimes: In some countries, authoritarian governments can freeze bank accounts or confiscate assets arbitrarily. Bitcoin offers a way to protect assets from these actions. In countries with political or financial instability, such as Venezuela, individuals use Bitcoin to safeguard their wealth from hyperinflation and government control.

Facilitation of International Remittances with Lower Fees: Traditional money remittances, especially to developing countries, can have high fees and take days to process. Bitcoin allows near-instant transfers with much lower fees. Migrant workers sending money to their families can use Bitcoin to avoid high fees from services like Western Union, increasing the amount of money reaching the final recipients.

Personal Control Over Financial Resources: With Bitcoin, individuals have full control over their funds without needing intermediaries. This ensures greater autonomy and security in their

financial transactions. In crisis situations, such as natural disasters or conflicts, where banks may be inaccessible, people can still access and use their funds with Bitcoin.

Transparency in Transactions and Fraud Prevention: The Bitcoin blockchain is an immutable public record of all transactions, reducing fraud possibilities and increasing transparency. Human rights organizations can use Bitcoin to receive donations transparently, allowing donors to see exactly how funds are used.

Incentive for Renewable Energy Use and Community Development: Bitcoin mining encourages the construction of electrical infrastructure in underdeveloped or hard-to-reach areas. The process has incentivized the use of renewable energy sources and innovation in energy efficiency, as miners seek cheaper and sustainable energy to maximize profits. Clean energy sources can enter negative pricing periods when production exceeds demand. In these moments, Bitcoin miners can use the excess energy to regulate grids, utilizing the energy and avoiding losses. Bitcoin mining can act as a flexible load, absorbing excess energy during production peaks and shutting down when demand increases.

Savings Method: Bitcoin is the best investment of all time and the best way to preserve purchasing power in the long term as fiat currencies are diluted by inflation. It is an extremely effective long-term savings tool as a scarce digital asset. Furthermore, most people do not have access to the stock market to preserve and grow their income.

Physical Money and Transaction Privacy are Disappearing: What happens when all our financial activity is trackable by a centralized entity, and we lose the financial privacy of physical money? What powers do governments and corporations gain when we trade our privacy and freedom for convenience? Each financial transaction reveals a vast amount of information about you, enabling control by corporations and governments, censorship of your transactions, and enabling surveillance and social engineering.

Bitcoin has been considered one of the main tools for guaranteeing human rights today. It is, in fact, a tool of freedom, especially useful in dictatorial countries with strict financial control of their citizens' bank accounts.

Bitcoin offers humans money that cannot be censored by authorities, cannot be devalued by governments, cannot be monopolized by corporations, cannot be easily mass-monitored, cannot be stopped by borders, and can be accessed by anyone. And that’s why Bitcoin matters for human rights.

"Few people are looking at the intersection of monetary freedom and real freedom." —Alex Li, Bitcoin Development Lead, Human Rights Foundation

"Bitcoin is collaborative, decentralized, and aligns very well with the human rights movement." —Alex Gladstein, CSO, Human Rights Foundation

"Bitcoin is terrible for dictators." —Alex Gladstein, CSO, Human Rights Foundation

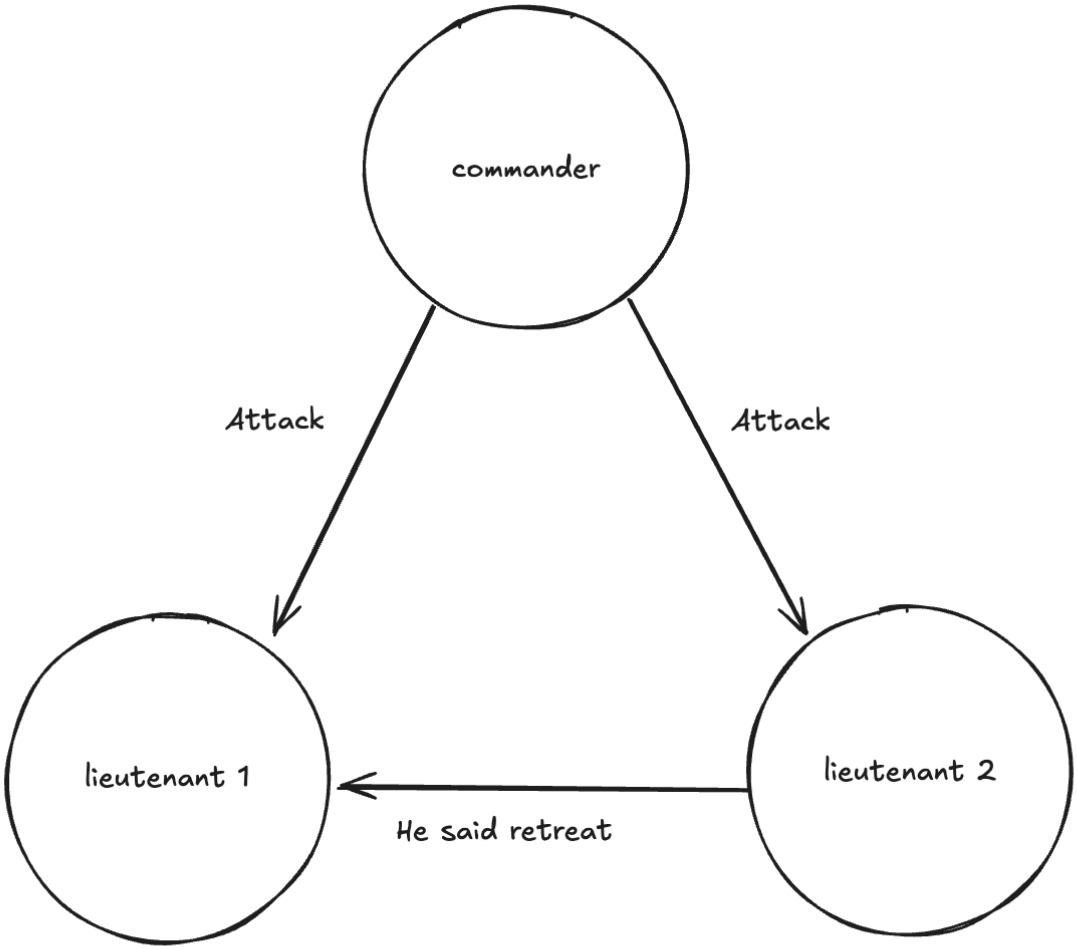
Recommended Reading: Check Your Financial Privilege.

1.4 The Byzantine Generals Problem

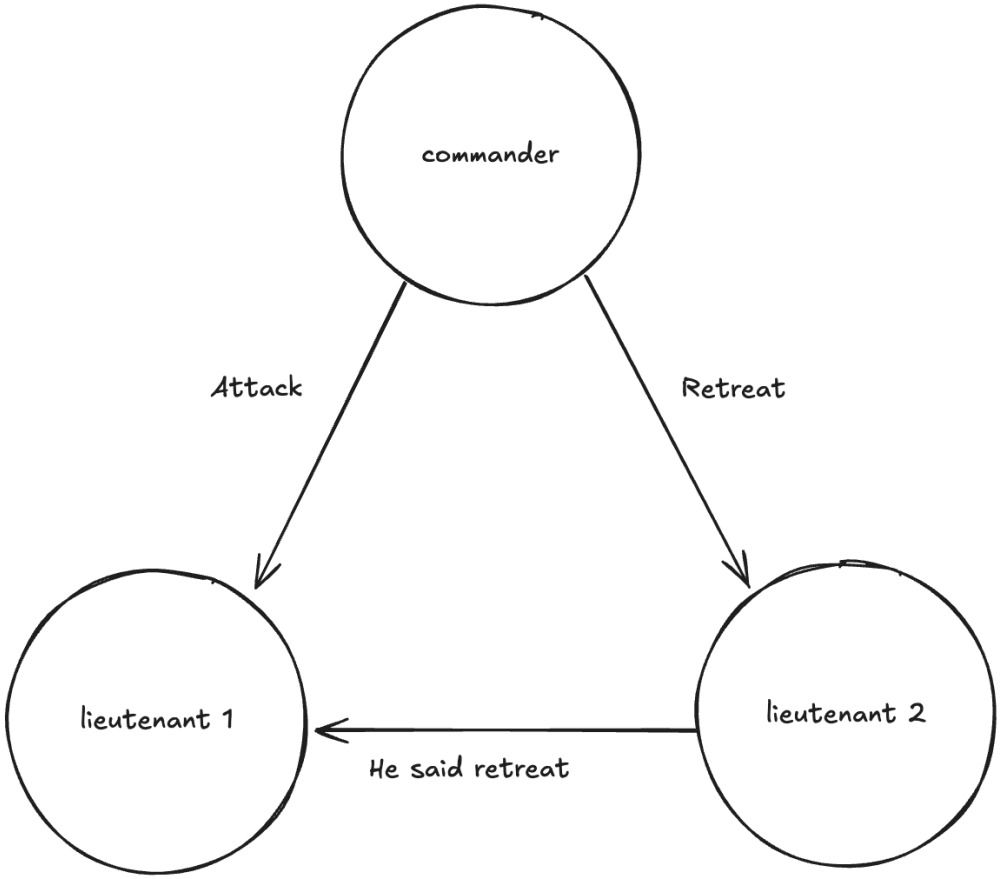
Bitcoin emerged as a solution created by Satoshi Nakamoto to address the Byzantine Generals Problem.

The Byzantine Generals Problem is a classic issue in computer science, specifically in the area of fault tolerance within distributed computing and distributed systems theory. It was first proposed by Marshall Pease, Robert Shostak, and Leslie Lamport in 1982, "expressed abstractly in terms of a group of Byzantine army generals camped with their troops around an enemy city."

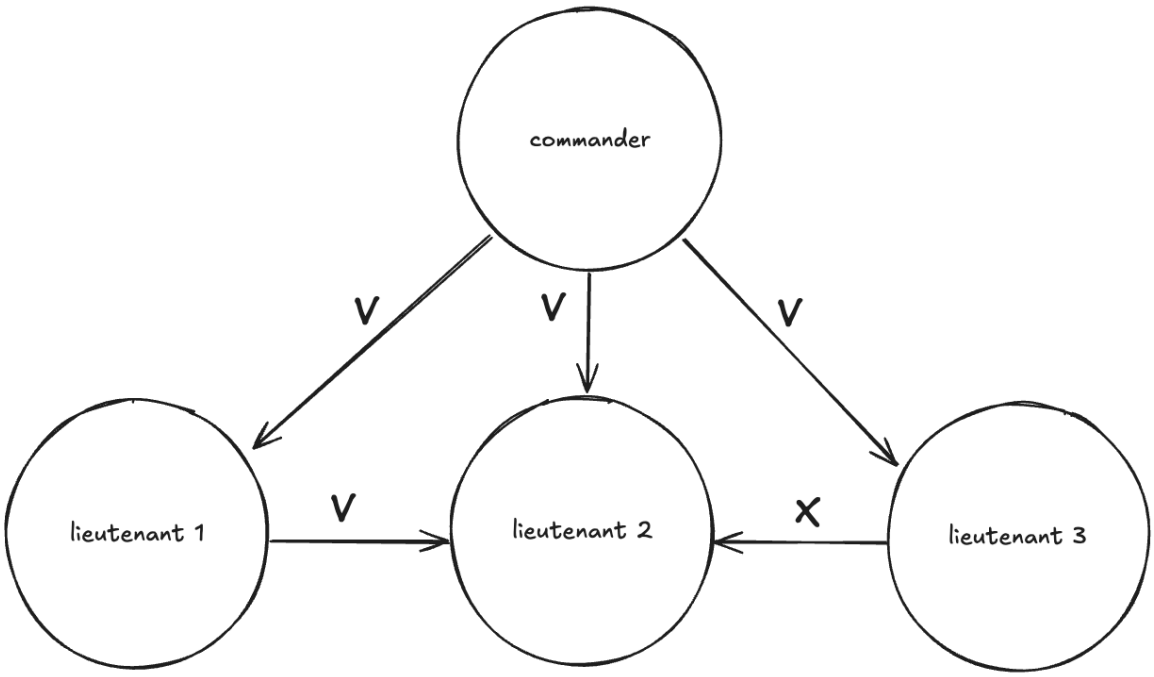
In this scenario, a traitor (whether the Commander or a Lieutenant) prevents the group from reaching consensus. In a financial ledger, think of the traitor as a malicious party aiming to facilitate fraudulent transactions. As the number of parties in the system increases, the number of communication channels (and opportunities for distrust) increases exponentially. Imagine the complexity of building consensus with thousands or millions of parties involved. The following schemes are attributed to Lamport, Shostak, Pease, 1982.



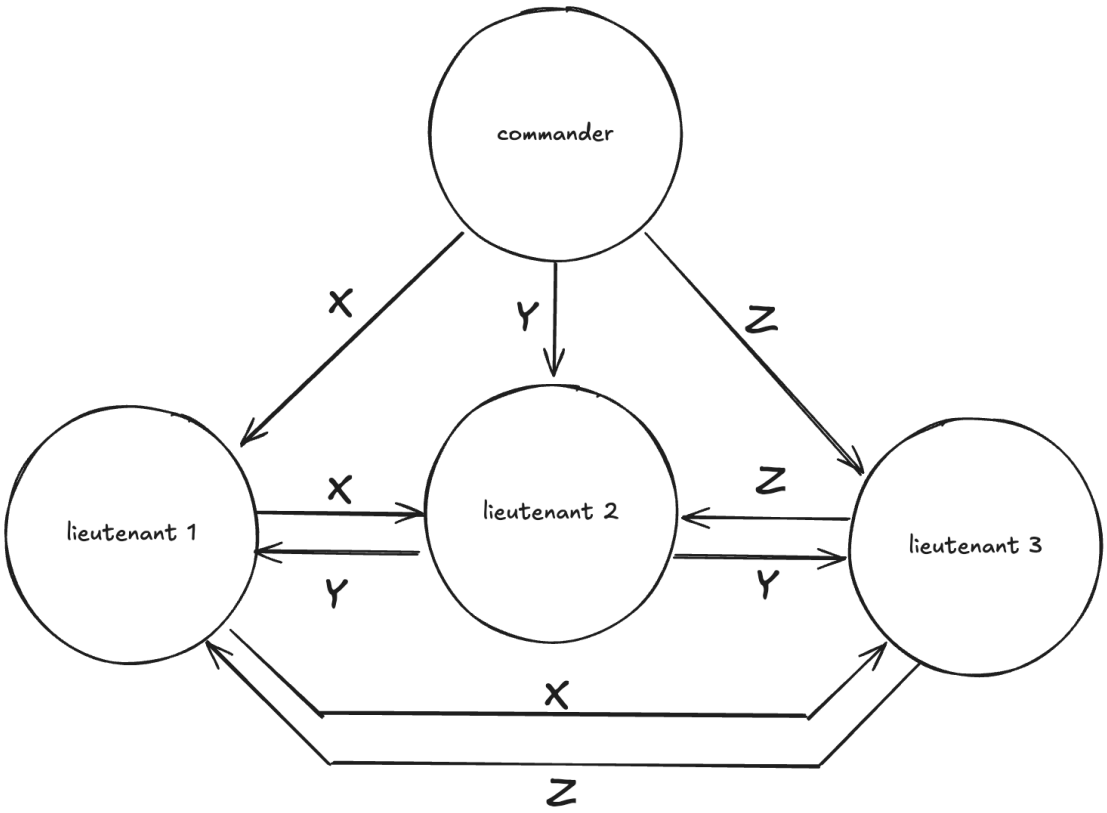
Lieutenant 2 is a traitor



The commander is a traitor



Algorithm OM(1)—The lieutenant 3 is a traitor



Algorithm OM(1)—The commander is a traitor

The solution to the Byzantine Generals Problem consists of combining probabilistic work (trial and error) in discovering the nonce that generates a hash equal to or below the difficulty level defined in the mining process, plus selecting the chain with the highest cumulative proof-of-work, meaning the greatest computational power employed to form it.

Bitcoin also solves the double-spending problem, ensuring that a digital asset (bits) cannot be duplicated or subject to copy-paste.

1.5 Bitcoin Whitepaper

Definição de Whitepaper
A whitepaper is an informative and technical document that presents the vision, methodology, and details of a project or technology, serving as a guide to understand its fundamentals and objectives.

The Bitcoin whitepaper, titled "Bitcoin: A Peer-to-Peer Electronic Cash System," was released by Satoshi Nakamoto on October 31, 2008. It was published on a cryptography mailing list

called "The Cryptography Mailing List" on the website metzdowd.com, detailing the structure and operation of a decentralized electronic payment system.

This document introduced innovative concepts that revolutionized the global financial system and have been used in various ways for the development of thousands of other cryptocurrencies and blockchain technologies.

It all started with Bitcoin, which remains the most important, decentralized, and robust candidate for a global financial system and store of value without a central authority. Many consider all other cryptocurrencies as centralized attempts to counterfeit Bitcoin.

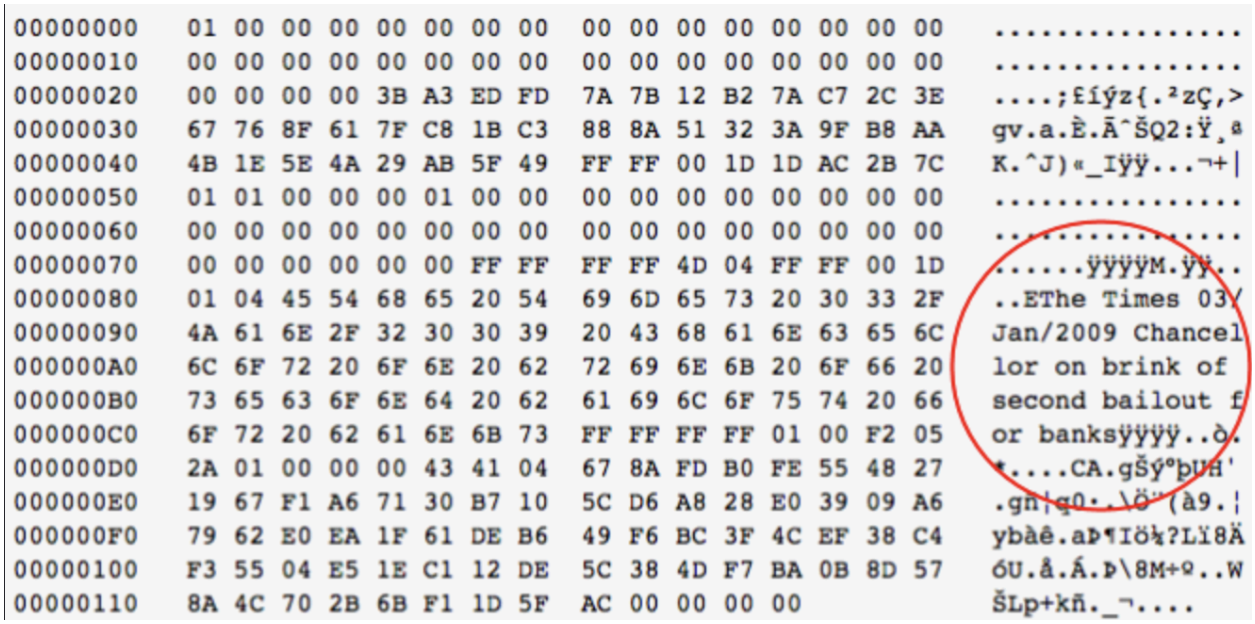
In fact, governments are migrating their financial systems to CBDCs (Central Bank Digital Currencies), systems inspired by Bitcoin technology but allowing for extreme surveillance and Orwellian control, as they operate through centralized control.

Since Bitcoin is open-source, anyone can copy its ideas and build new systems. **However, only the Bitcoin derived from Satoshi Nakamoto's first version is a true peer-to-peer electronic cash system.**

The Bitcoin software was released by Satoshi Nakamoto on January 3, 2009. On that day, Satoshi mined the genesis block, the first block of the Bitcoin blockchain, officially marking the beginning of the Bitcoin network.

Message in the Genesis Block
In the genesis block, Satoshi Nakamoto left a significant message: "The Times 03/Jan/2009 Chancellor on brink of second bailout for banks."

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



```
00000000   01 00 00 00 00 00 00 00   00 00 00 00 00 00 00 00   ................
00000010   00 00 00 00 00 00 00 00   00 00 00 00 00 00 00 00   ................
00000020   00 00 00 00 3B A3 ED FD   7A 7B 12 B2 7A C7 2C 3E   ....;£íýz{.²zÇ,>
00000030   67 76 8F 61 7F C8 1B C3   88 8A 51 32 3A 9F B8 AA   gv.a.È.Ã^ŠQ2:Ÿ¸ª
00000040   4B 1E 5E 4A 29 AB 5F 49   FF FF 00 1D 1D AC 2B 7C   K.^J)«_Iÿÿ...¬+|
00000050   01 01 00 00 00 01 00 00   00 00 00 00 00 00 00 00   ................
00000060   00 00 00 00 00 00 00 00   00 00 00 00 00 00 00 00   ................
00000070   00 00 00 00 00 00 FF FF   FF FF 4D 04 FF FF 00 1D   ......ÿÿÿÿM.ÿÿ..
00000080   01 04 45 54 68 65 20 54   69 6D 65 73 20 30 33 2F   ..EThe Times 03/
00000090   4A 61 6E 2F 32 30 30 39   20 43 68 61 6E 63 65 6C   Jan/2009 Chancel
000000A0   6C 6F 72 20 6F 6E 20 62   72 69 6E 6B 20 6F 66 20   lor on brink of
000000B0   73 65 63 6F 6E 64 20 62   61 69 6C 6F 75 74 20 66   second bailout f
000000C0   6F 72 20 62 61 6E 6B 73   FF FF FF FF 01 00 F2 05   or banksÿÿÿÿ..ò.
000000D0   2A 01 00 00 00 43 41 04   67 8A FD B0 FE 55 48 27   *....CA.gŠý°þUH'
000000E0   19 67 F1 A6 71 30 B7 10   5C D6 A8 28 E0 39 09 A6   .gñ¦q0·.\Ö¨(à9.¦
000000F0   79 62 E0 EA 1F 61 DE B6   49 F6 BC 3F 4C EF 38 C4   ybàê.aÞ¶Iö¼?Lï8Ä
00000100   F3 55 04 E5 1E C1 12 DE   5C 38 4D F7 BA 0B 8D 57   óU.å.Á.Þ\8M÷º..W
00000110   8A 4C 70 2B 6B F1 1D 5F   AC 00 00 00 00            ŠLp+kñ._¬....
```

This message is a reference to the headline of the British newspaper "The Times" on January 3, 2009, highlighting the global financial crisis and the imminent second bailout of banks by the government. Including this message was an implicit critique of the traditional financial system and a statement of intent regarding the need for an alternative, decentralized financial system resistant to manipulation and government intervention.

Negative Consequences of a Second Bank Bailout

Economic Inequality
Bank bailouts often benefit large financial institutions and their stakeholders, while the general population bears the costs through taxes and austerity. This can increase economic inequality as public resources are used to save wealthy private entities.

Moral Hazard
Bailouts encourage banks to take excessive risks, knowing they will be saved in case of failure. This risky behavior can lead to irresponsible practices, increasing the likelihood of future financial crises.

Increase in Public Deficit
The use of public resources for bailouts increases government debt, potentially resulting in austerity measures, cuts in essential public services, and tax increases, directly affecting the quality of life of the population.

Erosion of Public Trust
The perception that banks are favored by the government can erode public trust in the financial system and economic justice, generating social discontent and a possible loss of legitimacy of institutions.

Social Impacts
Printing money to finance bailouts can lead to inflation, reducing the population's purchasing power. Austerity measures and cuts in public services can result in unemployment and a broader economic crisis, directly affecting people's lives.

These consequences illustrate how bank bailouts can have profound and lasting impacts on society, exacerbating inequalities and creating additional risks for long-term economic stability.

1.6 Academic Origins

Bitcoin has an academic pedigree, being an innovative and creative solution to traditional financial systems. It is the culmination of decades of research in cryptography and distributed systems.

The concept of Bitcoin integrates discoveries and theories from researchers in cryptography, such as David Chaum, and in distributed systems, such as Nick Szabo and Wei Dai. All these authors emphasized the importance of privacy and anonymity in financial transactions, concepts that were incorporated into Bitcoin's design.

Learn more about the origins of Bitcoin:
https://nakamotoinstitute.org/literature/

March 9, 1993: A Cypherpunk's Manifesto—Eric Hughes
November 1998: b-money—Wei Dai
August 1, 2002: Hashcash - A Denial of Service Counter-Measure—Adam Back
October 31, 2008: Bitcoin: A Peer-to-Peer Electronic Cash System—Satoshi Nakamoto
August 29, 2017: Bitcoin's Academic Pedigree—Arvind Narayanan and Jeremy Clark

Bitcoin matters because it represents the first practical solution for creating a decentralized, secure digital currency without the need for intermediaries. It solves fundamental problems in the security of digital transactions, such as double-spending and the need for trust in a central entity.

The technology promotes financial freedom, privacy, and the ability to transfer value globally without restrictions, becoming a revolutionary alternative to traditional financial systems.

2. Bitcoin and Cryptography

2.1 Blockchain: Bitcoin's Ledger

A blockchain is a distributed database that stores information securely and transparently, functioning as a digital ledger where data is organized into cryptographically linked blocks.

Each block contains a record of transactions, a timestamp, and a reference to the previous block, ensuring data immutability since any change in a block would affect the entire chain.

To witness and store Bitcoin's blockchain, you start by downloading the reference client software—Bitcoin Core. This software downloads the entire blockchain, the ledger of all transactions in Bitcoin's history. Each full Bitcoin client stores a complete record of all Bitcoin transactions ever made, without a central registrar, just a set of distributed copies among all clients.

After downloading and validating the blockchain history, the question of synchronization arises: how to keep these blockchain copies synchronized with each other and achieve distributed consensus without a definitive central entity? When a client receives conflicting messages about a transaction, which one should it accept? The solution lies in the stack of cryptographic technologies and consensus rules used in the system.

Downloading the Bitcoin client software is not mandatory to use bitcoin. A simple free wallet app on a mobile phone is enough to get started. However, installing and maintaining nodes is recommended for users who want to verify their transactions privately and participate in the consensus process. Full nodes are also important tools for developers and advanced users.

Recommended reading: Bitcoin, Not Blockchain

2.2 Consensus Rules

Consensus rules define what Bitcoin software does and how. These rules are fundamental to maintaining the integrity and security of the network. They ensure that all participants agree on the current state of the distributed ledger (blockchain). Here are the main consensus rules of Bitcoin:

Proof of Work (PoW)
Bitcoin uses a consensus mechanism called Proof of Work, which requires miners to solve a computationally difficult mathematical problem to add a new block to the blockchain. The first miner to solve the problem earns the right to add the block and receive the block reward in bitcoin, along with the transaction fees for the bitcoins transferred in the block.

The Chain with the Most Accumulated Hash Power
The general rule for accepting a blockchain is that nodes always consider the chain with the highest cumulative difficulty or where the most computational resources have been spent as the valid chain. This ensures that the majority of the network's processing power agrees on the transaction history.

Block Size Limit
Each Bitcoin block is limited to a maximum size (currently 1 MB of base size with the possibility of up to approximately 4 MB of block weight due to SegWit). This limits the number of transactions that can be included in a block, affecting the network's transaction rate per second.

Transaction Formatting Rules
Transactions must follow a specific format and include a valid digital signature to be considered valid. They must also refer to unspent outputs of previous transactions (UTXOs—Unspent Transaction Outputs).

New Bitcoin Issuance
The block reward for miners is halved every 210,000 blocks (approximately every four years) in an event known as "halving." This controls the issuance of new bitcoins and is a fundamental rule for Bitcoin's predefined monetary policy. Halving day is a major celebration for the Bitcoin community.

This monetary policy also includes a maximum limit of 21 million bitcoins that can ever be issued. This absolute cap ensures that Bitcoin is a deflationary currency, protecting it against excessive inflation. When the last bitcoin is mined, expected around the year 2140, there will be no more issuance, and miners will be incentivized solely by transaction fees.

Timelock Transactions
Some transactions include a time rule that prevents them from being added to a block until a certain number of blocks or a period has passed. These rules are implemented and enforced through the software run by network participants (nodes and miners).

Block Validation
Before a block is added to the blockchain, it must be validated by nodes, ensuring that all transactions within it are valid.

Block Reorganization
If two blocks are found almost simultaneously, the network will eventually follow the chain that becomes longer first, discarding the orphaned block.

Difficulty Adjustment
Since mining is a probabilistic process of trial and error, the block output rate is directly proportional to the number of mining machines participating in the system (hash power) at any given time. To keep the block output rate stable, the difficulty of the mathematical problem

adjusts approximately every two weeks (or every 2016 blocks) to ensure new blocks are added approximately every 10 minutes.

Block Formatting Rules
Blocks must have a specific structure, including a block header that contains the previous block's hash, the timestamp, the target difficulty, and the nonce. The nonce is a random number used once to generate variations in the block's hash summary. Miners adjust the nonce repeatedly until they find a hash value below the target difficulty level set by the network. This operation is an essential part of the Proof of Work mechanism in Bitcoin's mining process.

Block Propagation
After a miner finds a valid block, it must propagate it quickly across the network for other nodes to validate it and start working on the next block.

Coinbase Transactions
Each block must include a coinbase transaction, which is the first transaction in the block, creating new bitcoins as a reward for the miner.

SegWit—Segregated Witness
An update that separates the digital signature from the transaction data, allowing more transactions per block and fixing transaction malleability.

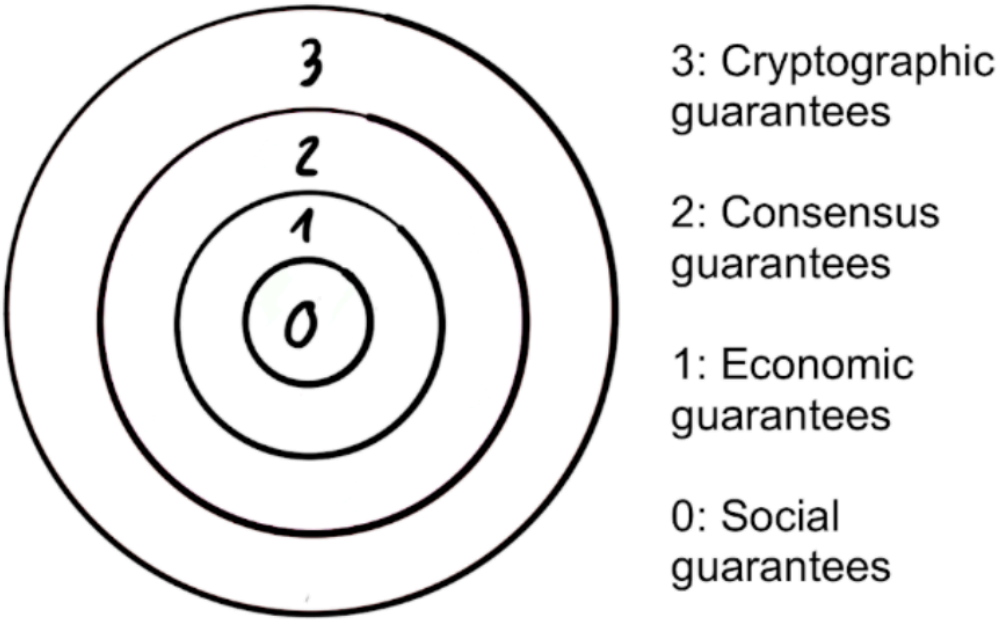
Any attempt to change these rules requires broad consensus in the community and often leads to a hard fork of the software (splitting into two compatible and incompatible blockchains, essentially two different currencies), as seen with the creation of Bitcoin Cash and other variants.

Changing Bitcoin's consensus rules is a rigorous process involving formal proposals, community discussions, code development, extensive testing, and carefully planned activation methods.

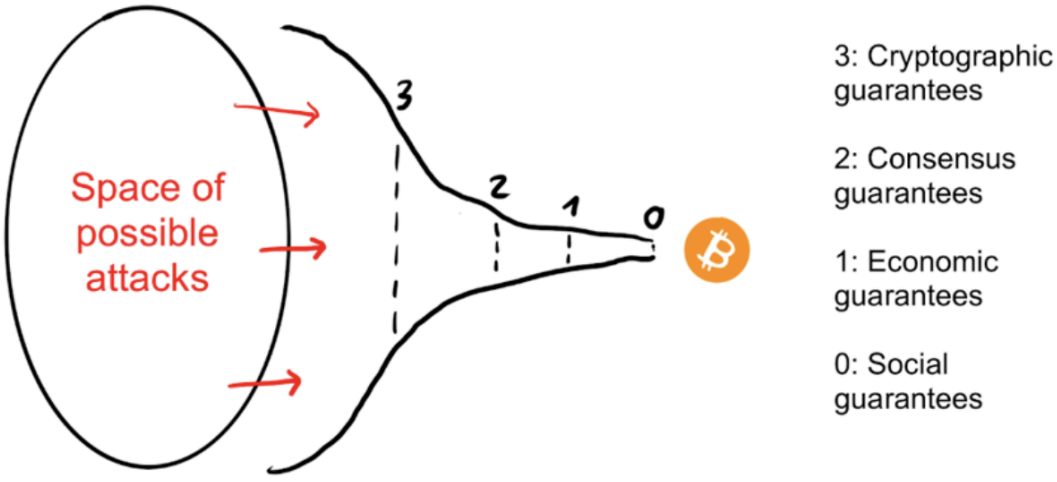
This process ensures that any change is widely debated, tested, and accepted by the community, preserving the security and decentralization of the Bitcoin network.

2.3 Bitcoin Security Model

Public blockchains are secure due to a combination of multiple layers of protection, including cryptographic guarantees, consensus, economic incentives, and community support. This onion model illustrates how each layer adds security and prevents attacks.



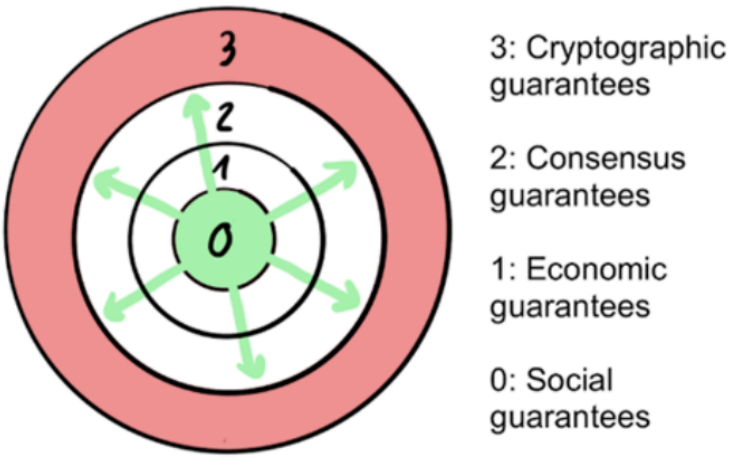
To permanently destroy a public blockchain, it is necessary to destroy the users' trust in the state of its ledger (the list of ownership), as well as the ability to reliably update that state in the future. All upper layers serve to prevent this from happening.



Even if one layer fails, the others help maintain the integrity of the system.

An open blockchain is merely a means to automate the process of establishing social consensus among its participants, a tool for maintaining and updating a shared database. The state of this database has value for the participants, and they are strongly incentivized to restore the system when it fails.

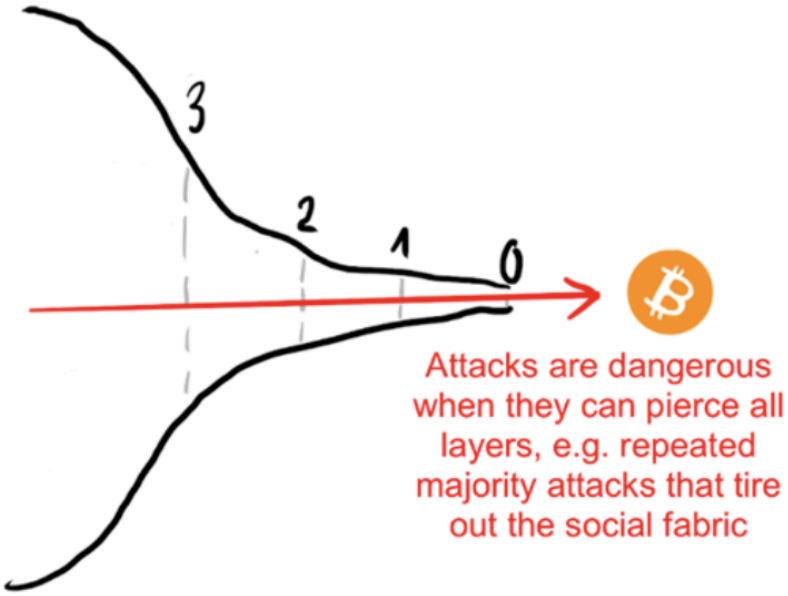
For example, if the cryptographic hash function is broken, the social layer can reach a manual consensus (guided by technical experts) to replace the damaged part.



ECDSA breaks; a social intervention can mitigate the fallout and fork in a replacement

Similarly, if a consensus attack bypasses the economic guarantees, the social layer can still manually reject it. If an attacker with the majority of hash power started performing a denial-of-service (DoS) attack on the network by mining empty blocks, fully accepting the economic damage to themselves, users could decide to change the Proof of Work (PoW) function and thus manually remove the miner's control.

Attacks are dangerous when they manage to penetrate all layers and ultimately erode the social core of the system to the point where it can no longer overcome the damage in the upper layers and recover.



For example, a repeated attack by a majority that damages and depletes the social fabric.

For both recovery and manual intervention to work, the communities of each project need strong social conventions around the key properties of their project. In the case of Bitcoin, these core values are transaction irreversibility, censorship resistance, the absence of retroactively incompatible changes, and the 21 million token limit. These serve as action roadmaps for when social intervention becomes necessary and create focal points around what needs to be fixed and what does not.

These fundamental values of a project are perpetually renegotiated, and not all users agree on all properties. However, the stronger the agreement around a particular value, the more likely it is to be upheld during times of difficulty.

The security of Bitcoin depends on the interaction between all these layers and the strong social and technical foundation that supports the system.

Learn more at: [The Onion Model of Blockchain Security – Part 1]

2.4 Hash Functions

Bitcoin is a collection of concepts and technologies that form the basis of a digital money ecosystem, including a decentralized peer-to-peer network—enabled by the Bitcoin protocol, a public ledger of transactions—the blockchain, a decentralized, mathematical, and deterministic mechanism for currency issuance—distributed mining and the “Proof of Work” consensus algorithm, and a decentralized system for transaction verification—transaction script.

As such, it relies heavily on cryptographic technologies, such as hash functions (SHA-256 and RIPEMD-160) and public key cryptography (ECDSA – the Elliptic Curve Digital Signature Algorithm).

Ownership of bitcoin is established through the relationship between public keys and digital signatures produced from the corresponding private keys.

A cryptographic hash function is a mathematical function commonly used to verify data integrity, transforming identical data into a unique, representative, and fixed-size summary (hash). Any accidental or intentional modification of the input data—such as rearranging characters—will completely change the hash output. Different inputs should never produce the same hash, known as a "hash collision."

Since hash functions are very difficult to reverse, it is almost impossible to derive the input value from its hash output. This is useful for commitment schemes, where a hidden value can be shared and later revealed authentically. Bitcoin uses the SHA-256 hash function. The hash output is 256 bits (32 bytes) long, as 1 byte = 8 bits. A SHA-256 hash is typically presented as a 64-character hexadecimal string. Each of the 32 bytes is represented by 2 hexadecimal characters.

The first hash algorithm was presented by senior IBM engineer Hans Peter Luhn in 1958. Example of hash output:

```
bitcoin → 6b88c087247aa2f07ee1c5956b8e1a9f4c7f892a70e324f1bb3d161e05ca107b
```

```
bitcoins →
b1e84e5753592ece4010051fab177773d917b0e788f7d25c74c5e0fc63903aa9
```

Notice that just the inclusion of the letter "s" produced a radically different output, making hash functions extremely efficient for data integrity verification. Cryptographic hash functions are widely used in Bitcoin. In Bitcoin, SHA-256 is used in several critical components:

Bitcoin addresses
Bitcoin addresses are derived from public keys using SHA-256 and RIPEMD-160 hashing. Hash functions are used in the generation of different types of addresses.

txid
Bitcoin transactions are identified by a transaction ID (TXID), which is a SHA-256 hash of the transaction data.

Mining
When transactions are broadcasted across the network, the hash function is used to verify data integrity, ensuring they have not been corrupted or altered during transmission. In the

proof-of-work mechanism in Bitcoin mining, double SHA-256 hashing is used. Miners must find a hash value that is lower than a specified target, repeatedly hashing block headers until the desired value is found.

Block hash
Each block in the Bitcoin blockchain has a unique hash generated using SHA-256. This hash links each block to the previous one, forming a secure chain. All Bitcoin transactions are stored in blocks, which are linked sequentially, always referencing (including) the previous block's hash. Cryptographic hash functions verify block integrity and establish the blockchain's chronological order.

Hash Simulator
You can practice writing something and check how hash outputs change using this hash simulator: https://academo.org/demos/SHA-256-hash-generator/

2.5 Digital Keys

Asymmetric Cryptography

If the same key were used for both encryption and decryption, the relationship would be symmetric. Asymmetric cryptography, on the other hand, involves two keys, a public key and a private key. The public key encrypts the message, and the private key decrypts it. The public key can be derived from the private key, but the private key cannot be derived from the public key—it is computationally infeasible.

When encrypting a message, the sender encrypts the message M using the recipient's public key (derived from the private key) to produce the encrypted message C. The recipient decrypts the encrypted message C using their private key to view the original message M. C is the result of encryption (also known as the "cipher"). M is the unencrypted/decrypted message (also known as "plaintext"). Here is how asymmetric cryptography is applied to Bitcoin technology.

Private Key (Privkey)

It is a randomly generated number that must be kept secret. It is used to generate public keys and digital signatures, confirming ownership, and authorizing the spending of bitcoin. They are created by Bitcoin wallet software. It should be stored in a secure location, preferably written on paper or a stainless steel plate and kept in a safe. Whoever possesses the private key has full control over the bitcoins associated with it.

Public Key (Pubkey)

The public key is generated from the private key using elliptic curve multiplication in the secp256k1 elliptic curve group. In Bitcoin, the public key is masked through a series of hash functions, including SHA-256 and RIPEMD-160, and is represented by a Bitcoin address when spending and receiving funds.

Address Generation

The public key is used to generate Bitcoin addresses, which are publicly shared to receive bitcoins.

Verification of Digital Signatures

The public key is used to verify digital signatures. When a transaction is signed with the private key, anyone can use the corresponding public key to verify the authenticity of the signature and confirm that the transaction was authorized by the private key owner.

Public key cryptography is used to produce digital signatures with ECDSA (Elliptic Curve Digital Signature Algorithm), specifically the secp256k1 curve, to authorize and validate transactions.

As of January 2021, Bitcoin Core v0.21 began supporting the Schnorr digital signature algorithm. After months of testing and discussions on methods, the soft fork was activated on the mainnet in November 2021.

Important Links:

Bitcoin Core v0.21

Taproot Activation Proposals

Taproot Locks In

Preparing for Taproot

Taproot: Privacy-Preserving Switchable Script

The Schnorr digital signature algorithm was designed by German cryptographer Claus-Peter Schnorr. His 1991 patent expired in February 2010.

2.6 Digital Signatures

Digital signatures are used to authenticate valid transactions. To make a payment in Bitcoin, a Bitcoin transaction T is constructed. A subset M of the information in transaction T is signed as follows:

Signing the Transaction T

1. Create the transaction T
2. Select the subset M from transaction T (e.g., the transaction identifier, transaction instructions, etc.)
3. Calculate the hash H of M: H = sha256(M)

4. Calculate a signature S using the output of this hash function Fhash(M) with the sender's private key, where Fsig is the signature algorithm:

```
S = Fsig(Fhash(M), Kpriv)
```

5. Send the signature S and the public key Kpub along with the transaction T to Bitcoin miners.

Verification is the inverse of the signature generation function, using the values R, S, and the public key to calculate a value P, which is a point on the elliptic curve. To verify a transaction received with the signature and public key Kpub, a receiver must:

Verify the Transaction T

$$P = S^{-1} \cdot F_{\text{hash}}(M) \cdot G + S^{-1} \cdot R \cdot K_{\text{pub}}$$

Calculate:
Where:
R and S are the signature values
Kpub is the public key
M is the transaction data that was signed
G is the generator point of the elliptic curve

If the x-coordinate of the calculated point P is equal to R, then the verifier can conclude that the signature is valid. Note that when verifying the signature, the private key is neither known nor revealed.

The public ledger records transfers of ownership of a quantity of bitcoin from one owner to another. (Note: Transactions can also be 'self-transfers,' i.e., between sets of addresses and/or keys controlled by the same person.)

2.7 Receiving Addresses

We can find different types of addresses in Bitcoin.

Legacy (P2PKH—Pay-to-Public-Key-Hash): Original format, starts with 1. Example: 1A1zP1eP5QGefi2DMPTfTL5SLmv7DivfNa.

SegWit (P2WPKH—Pay-to-Witness-Public-Key-Hash) P2SH (Pay-to-Script-Hash): Starts with 3. Example: 3J98t1WpEZ73CNmQviecrnyiWrnqRhWNLy.

In P2SH, the script is hashed and only revealed when spent, allowing for more complex transactions, such as multisig. For example, a 2-of-3 scheme might require two of three signatures to validate a transaction, useful for fiduciary or escrow services (trusted third parties).

Native SegWit (bech32): Starts with bc1. Example: bc1qw508d6qejxtdg4y5r3zarvary0c5xw7kygt080.

Transactions on the blockchain do not record public keys or recipients but instead use an abstraction called a "Bitcoin address" to record the beneficiary of each amount, allowing for greater flexibility.

An address is a unique identifier for the destination of a bitcoin payment, generated from and corresponding to a public key or script.

It is generally generated by applying the cryptographic hash functions SHA-256 and RIPEMD-160, in series, to the public key. These addresses are encoded using Base58 encoding, which represents an address in a human-readable form of 58 alphanumeric characters. Wallet software typically encodes in QR Code format, making the financial transaction experience similar to PIX.

To create a Bitcoin address, the Bitcoin wallet software first generates an ECDSA Public-Private key pair from a random number. The Bitcoin address is generated by applying the following sequence:

Private Key: Hexadecimal representation of a binary:

```
C4bbeb1fbec99d65bf59d85c8cb62ee2db963f0fe106f483d9afa73bd4e39a8a
```

The public key is derived by multiplying the private key by a predefined generator point on the secp256k1 elliptic curve. This operation is one-way, meaning it is easy to calculate the public key from the private key, but practically impossible to calculate the private key from the public key due to computational complexity (discrete logarithm problem). It is computationally simple to derive the public key but impractical to reverse.

This generator point is a well-known, well-defined constant. The operation is mathematically expressed as:

```
Chave Pública = k * G
```

Where k is the private key (an integer), and G is the generator point on the elliptic curve. After the multiplication, the private key transforms into the public key:

```
0478d430274f8c5ec1321338151e9f27f4c676a008bdf8638d07c0b6be9ab35c71151806324
3acd4dfe96b66e3f2ec8013c8e072cd09b3834a19f811659cc3455
```

To arrive at a Bitcoin address, this public key is passed through the SHA-256 and RIPEMD-160 hash functions in sequence. RIPEMD-160 always produces a 160-bit (20-byte) hash. The correct output should be a 20-byte (or 40-character hexadecimal) hash. Let's first examine the SHA-256 output.

```
SHA256(0478d430274f8c5ec1321338151e9f27f4c676a008bdf8638d07c0b6be9ab35c7115
18063243acd4dfe96b66e3f2ec8013c8e072cd09b3834a19f811659cc3455)
= c4c5d791fcb4654a1ef5e03fe0ad3d9c598f9827
```

Now the RIPEMD-160 of SHA-256.

```
RIPEMD160(c4c5d791fcb4654a1ef5e03fe0ad3d9c598f9827) =
448e53f6c8da1fcdea5f1812403db91d9867e305
```

Add the 00 prefix for mainnet.

```
00448e53f6c8da1fcdea5f1812403db91d9867e305
```

Now for the double SHA-256. First SHA-256.

```
SHA256(00448e53f6c8da1fcdea5f1812403db91d9867e305)
= 3e417cec974b61aaa0ac2977ed4232f86da379782978d8e05ed3c378d1325a14
```

Second SHA-256.

```
SHA256(3e417cec974b61aaa0ac2977ed4232f86da379782978d8e05ed3c378d1325a14)
= 7d6bc0cb60da0fb5081697c26c6b8913ee5ac609927d02c571cdacba486bb9d9
```

The first 4 bytes of the second SHA are used as a `checksum`.

```
7d6bc0cb
```

Now combine the prefix + RIPEMD + checksum.

```
00448e53f6c8da1fcdea5f1812403db91d9867e3057d6bc0cb
```

Now encode this number with Base58 to generate the final bitcoin receiving address.

```
Base58(00448e53f6c8da1fcdea5f1812403db91d9867e3057d6bc0cb)
```

The result is the bitcoin receiving address.

```
17FVTAe4x93k79gT1ND3mk9sM4jUP2WFMt
```

Technically, the address is public, and the public key from which it is derived is not exposed until the bitcoin is spent.

Curiosity: There are 52 characters in the alphabet if we include all uppercase and lowercase letters. There are also 10 numbers (from 0 to 9). To avoid confusion and copying errors, Satoshi removed 4 commonly confused characters from the address generation process: the uppercase letter 'O' and the number '0', the uppercase letter 'I' and the lowercase letter 'l.'

Phishing Warning: Be careful with fake emails and websites that try to trick you into stealing your private keys. Always verify the authenticity of sources before entering sensitive information. If an opportunity seems too good to be true, it certainly is a scam.

2.8 Wallets

A Bitcoin wallet is a digital tool that allows users to store their private keys, create Bitcoin addresses for receiving funds, and sign outgoing Bitcoin transactions. Although the term "wallet" suggests a place where bitcoins are stored, in reality, Bitcoin wallets do not store the bitcoins themselves. Instead, they store the private keys needed to access and manage the bitcoins recorded on the blockchain.

There are different types of wallets, each with its specific characteristics in terms of convenience and security.

Hot Wallets Hot wallets are connected to the internet and are generally easier to use. Examples include mobile wallet apps, desktop wallet software, and online wallets. They are considered hot wallets because their private keys are generated on a mobile app or device that is connected to the internet.

Overall, they are secure and a perfect way to start buying, selling, or exchanging Bitcoin for services, but it is not recommended to leave large amounts of Bitcoin in them. A suggested first

on-chain wallet is BlueWallet, and for a first Lightning wallet, Phoenix. We will learn more about the Lightning network later.

Cold Wallets Cold wallets can be of various types and vary in their internet contact surface. Examples include hardware wallets (signing devices), paper wallets, and offline wallets.

Signing Devices These create and maintain your private keys offline, serving as an extra layer of security. Cold wallets are not connected to the internet, making them much more secure against hacks and malware. There are DIY ways to create these signing devices, but for beginners, there are well-established brands in the market. A popular option is the Jade Wallet.

It is never recommended to keep your bitcoins in wallets that support other cryptocurrencies. Always, at least, install firmware that is Bitcoin-only—in the case of Trezor, for example.

There are some DIY ways to create signing devices: Seedsigner: https://seedsigner.com/ Krux: https://selfcustody.github.io/krux/ Jade: https://github.com/Blockstream/Jade Specter: https://github.com/cryptoadvance/specter-diy

Steel Wallets Steel wallets are physical backup methods designed to store your private keys or recovery phrases securely and durably. They are made of metal, usually stainless steel, making them resistant to physical damage like fire, water, corrosion, and mechanical impacts.

Unlike paper or electronic devices, steel wallets are virtually indestructible and can withstand extreme conditions. Use the steel wallet to engrave your private key or recovery phrase. Some steel wallets come with engraving kits, like stamped letters and numbers, or plates that you can mark manually. Ensure all information is recorded correctly and legibly. Once engraved, this information is permanent, so accuracy is crucial. Always test instantiating your keys on a signing device to ensure your seed is written correctly.

Incorporating a steel wallet as part of your security strategy can provide additional peace of mind, ensuring that your private keys or recovery phrases are protected against virtually any type of physical damage. It is an excellent option for those who want a robust, long-term storage method for their bitcoins.

Components of a Bitcoin Wallet (signing devices and applications)

Private Key: A secret sequence of numbers and letters that allows the user to access their bitcoins. It is essential for signing transactions and must be kept secure.

Public Key: Derived from the private key, the public key is used to generate Bitcoin addresses.

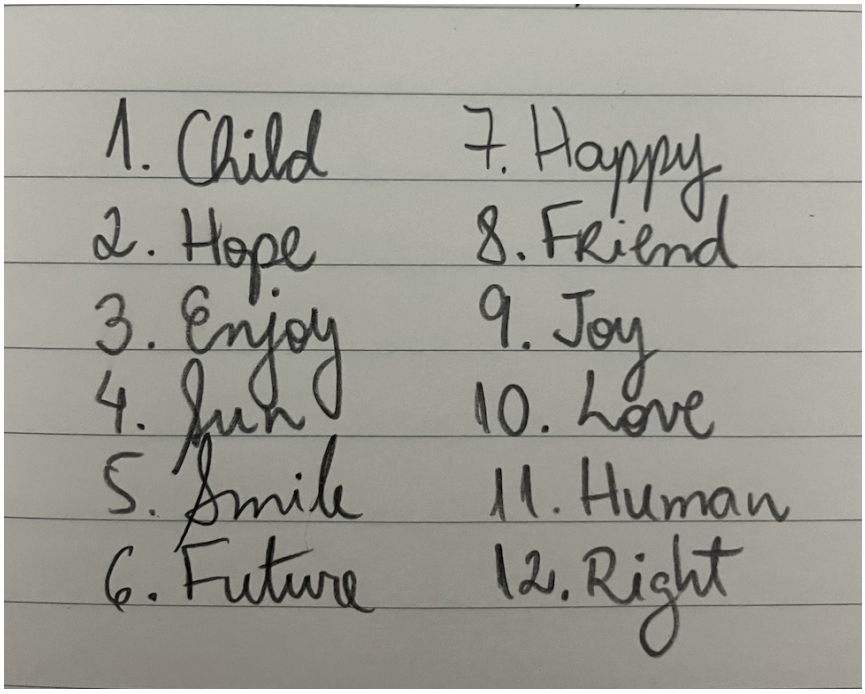
Bitcoin Address: A hashed version of the public key, used to receive bitcoins. It functions like a bank account number.

Note: BIPS—Bitcoin Improvement Proposals Bitcoin Improvement Proposals are the community's communication mechanism about improvement proposals that will become part of Bitcoin's source code. Some relevant BIPs for how our wallets are today include:

BIP 32: Hierarchical Deterministic Wallets (HD Wallets) BIP 32 introduces the concept of "hierarchical deterministic wallets" or HD wallets. This is a method of generating and managing private and public keys more organized and securely. Before BIP 32, each Bitcoin address had its own private key. Managing multiple keys was complicated and insecure. With BIP 32, you can create a single "seed" (a sequence of words) that can generate an infinite number of private and public key pairs. This simplifies the backup and restoration of all your keys with just a single seed.

BIP 39: Mnemonic Phrases BIP 39 defines the standard for creating a "mnemonic phrase" - a sequence of 12 to 24 common words that represent a cryptographic seed. This mnemonic phrase makes the process of backing up and restoring wallets much simpler and more secure. Instead of remembering a long and complicated sequence of numbers and letters (the private key), you only need to remember or write down a phrase of common words. For example, a mnemonic phrase might be something like "runner rain table sun trip book..."

Phishing Warning: Be cautious with fake emails and websites that try to trick you into stealing your private keys. Always verify the authenticity of sources before entering sensitive information. If an opportunity seems too good to be true, it certainly is a scam.



This phrase can be instantiated on any signing device or application for access and signing of Bitcoin transactions. They are essentially a symbol of financial self-sovereignty and the need to store them in an extremely secure place, out of reach of malicious actors. Great powers bring great responsibilities.

BIP 44: Multi-Account Address Structure

Recommended for more advanced users, BIP 44 expands on the concepts of BIP 32, providing a standardized way to organize and manage multiple accounts within a single HD wallet.

With BIP 44, you can create multiple accounts (e.g., one account for daily spending, another for savings, etc.) within the same wallet, each with its own addresses. This brings an additional layer of organization and flexibility, making fund management for different purposes easier.

In terms of wallet security, there is an interesting project that verifies the integrity of binaries being released to the public in software stores. The project is open source and is called WalletScrutiny.

3. Transactions

Each transaction associates a quantity of bitcoin with a Bitcoin address. When bitcoins are sent to someone, the transaction records the transfer of bitcoin from the current owner's address to the new owner's address, authorized by a valid digital signature.

When this transaction is broadcast to the Bitcoin network, all peers know that the new owner of these bitcoins is the holder of the new receiving address.

The complete transaction history is maintained by all peers (full clients or nodes) in the Bitcoin network, so anyone can verify who the current owner of any amount of bitcoin is without needing to know their private keys.

In most cases, both public and private keys are stored in a Bitcoin wallet. A Bitcoin wallet, like a credit card, does not contain any bitcoin but only the private-public key pairs, which are used as mechanisms to access your funds, transfer them to another address, or generate new receiving addresses.

Bitcoin Pizza Day: Celebrated annually on May 22nd, it marks the first documented commercial transaction using Bitcoin. In 2010, a programmer named Laszlo Hanyecz made history by purchasing two pizzas for 10,000 bitcoins.

On May 18, 2010, Laszlo Hanyecz posted a message on the Bitcointalk forum offering 10,000 bitcoins in exchange for two pizzas. Four days later, on May 22, he managed to complete the transaction with a user who accepted the payment and ordered the pizzas for him. This event is widely recognized as the first time Bitcoin was used to purchase a physical good.

3.1 Inputs, Outputs, UTXOs

A UTXO (Unspent Transaction Output) is a fundamental concept in the functioning of the Bitcoin protocol. To understand UTXO, it is important to first comprehend how Bitcoin transactions are structured and processed.

In a Bitcoin transaction, there are inputs and outputs. Each transaction consumes one or more unspent outputs from previous transactions (inputs) and creates new outputs. A transaction is considered valid when the outputs of all inputs are correctly referenced and the sum of the input values is equal to or greater than the sum of the outputs (with the difference possibly being the transaction fee paid to the miner).

A UTXO is a transaction output that has not yet been spent as an input in a subsequent transaction. Essentially, it is a record on the blockchain indicating the amount of Bitcoin available to be spent by a specific address.

When a transaction is confirmed, its outputs become UTXOs until they are used in a new transaction. When a new transaction is created, it references one or more UTXOs as its inputs. These UTXOs are then spent and removed from the set of available UTXOs. Each UTXO is uniquely identified by the hash of the transaction that created it and the index of the output within that transaction.

The basic structure of a UTXO includes:

Transaction ID (txid): The hash of the transaction that created the UTXO.

Index: The specific index of the output within the transaction.

Value: The value of the UTXO in satoshis (the smallest unit of Bitcoin).

ScriptPubKey: A script that defines the conditions necessary to spend the UTXO.

The UTXO model allows the state of the blockchain to be easily verifiable and keeps parallel transactions independent, facilitating validation and mining. Additionally, the model improves scalability, as it allows parallel validation of transactions and reduces the complexity of tracking coin ownership.

The use of UTXOs increases security because each transaction must reference valid outputs from previous transactions, preventing double spending and making it easier to detect fraud attempts. The UTXO model is one of the pillars that ensures Bitcoin's security and efficiency, enabling secure and verifiable transactions on a decentralized network. This model is crucial for the functioning of the Bitcoin protocol.

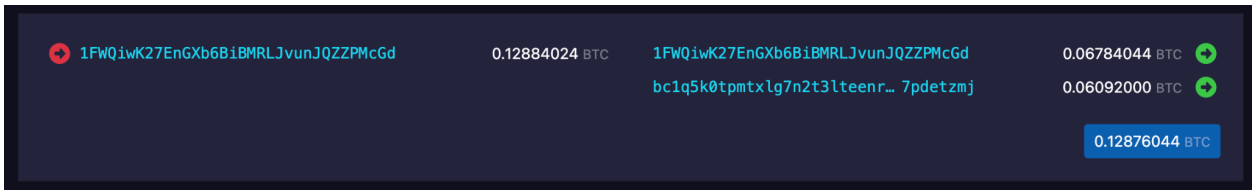
We can see that the outputs of one transaction are the inputs for the next transaction. A transaction can have multiple outputs.

Common Transaction Types

The most common form of transaction is a simple payment from one Bitcoin address to another, which often includes some change returned to the original owner. This type of transaction has one input and two outputs, as shown below:

Inputs are debited from the original owner's Bitcoin address. Outputs are credited to the new owner's Bitcoin address; the change is returned to the original owner in a second output.

A real example of this type of transaction:



The address 1FWQiwK27EnGXb6BiBMRLJvunJQZZPMcGd uses a UTXO of 0.12884024 that is transformed into a payment amount of 0.06092000 and change of 0.06784044 back to the sender's address. In this transformation, a network fee is also deducted.

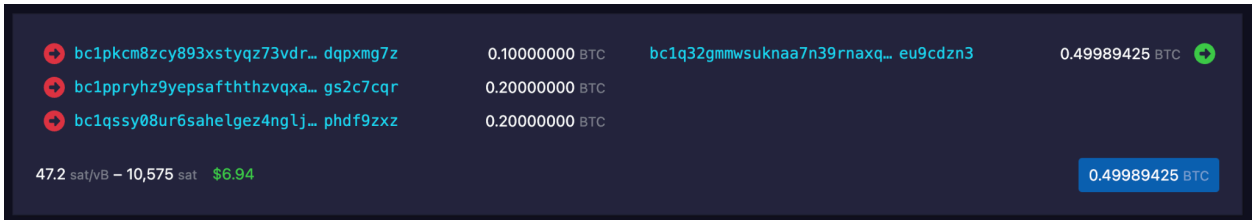
Another common form of transaction aggregates multiple inputs (3 in the example) into a single output. This represents the real-world equivalent of exchanging a pile of coins and banknotes for a single higher-value note. Or rather, burning those old notes and creating a single new one that represents the value of all of them.

Transactions like these are sometimes generated by wallet applications to clear many small values received as change, a process called transaction consolidation. Usually, a consolidation transaction is done when network fees are low.

Basically, you create a transaction to your own address, whose value is the sum of all your smaller UTXOs. And voila, the value is consolidated into a single higher-value UTXO. Consolidating UTXOs during low-fee periods helps save on fees in future transactions with other people. Later, we will understand better why consolidating UTXOs during low-fee periods is an important practice.

In this type of transaction, multiple inputs are collected. A single output is created.

A real example of this type of transaction:

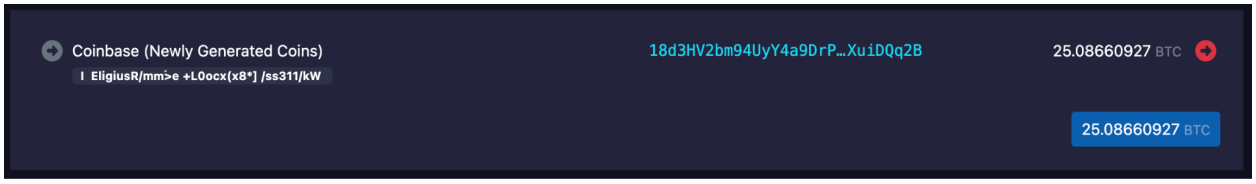


Another form of transaction frequently observed in the Bitcoin ledger distributes one input to multiple outputs, which may or may not represent multiple independent recipients. This type of transaction is sometimes used by commercial entities, such as when processing employee payroll. From a single input, multiple outputs are credited to the new owners' Bitcoin addresses.

A real example of this type of transaction:

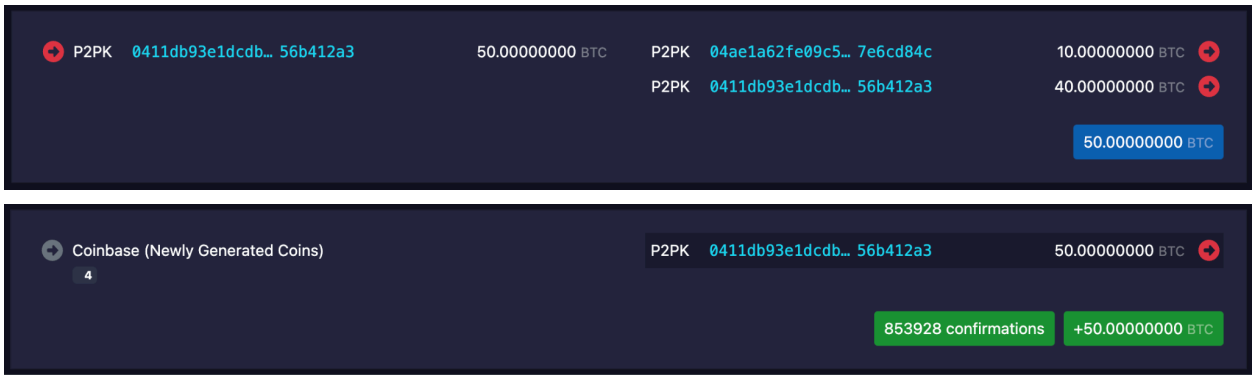
The OP_RETURN field in Bitcoin transactions is used to embed arbitrary data in the blockchain. This field allows the inclusion of information without affecting the spendability of bitcoins. There are services that pass documents through SHA-256 and record the resulting hash in this field, to permanently register the authenticity of a document that can be verified later. This practice is not widely accepted in the Bitcoin community, but it is one way the Bitcoin network is used.

Another type of transaction frequently observed in the Bitcoin ledger is the coinbase transaction. Unlike regular transactions, the coinbase transaction has no input because it represents newly generated coins. It has only one output, which is credited to the Bitcoin address of the block's miner



First Bitcoin Transaction

Below is the first Bitcoin transaction ever made, between Satoshi Nakamoto and Hal Finney. In this transaction, Satoshi sent 10 BTC to Hal Finney, with 40 BTC returning as change. We can also see the coinbase transaction of the block in which this transaction was included, which has no inputs and includes a block reward of 50 BTC.



3.2 SegWit—Segregated Witness and Transaction Malleability

Transaction malleability is a vulnerability that would allow a malicious actor to modify ("malleate") a transaction by altering the witness data in a way that changes the transaction ID. (Remember that after confirmation, the digital signature and thus the transaction ID are immutable.)

What is produced now is a second transaction that signs the same amount to the same destination address but with an altered transaction ID. This attack does not allow the attacker to steal funds or change the destination of the funds. However, it can be used to defraud the sender by tricking them into sending a second payment after the original transaction appears to have not been confirmed.

Transaction Malleability Explained

Fact: Mt.Gox, the largest exchange in Japan, collapsed in February 2014, suspended all withdrawals, and blamed transaction malleability for the suspension of withdrawals.

SegWit arose to solve the transaction malleability problem.
Segregated Witness (SegWit) is an architectural change activated on Bitcoin on August 1, 2017. It moves the witness data of transactions from the scriptSig (unlocking script) field of a transaction to a separate witness data structure that accompanies the transaction.

Most of the space in a transaction (about 65% or more) can be occupied by signature data. By moving the witness data out of the transaction, the transaction hash used as an identifier no longer includes the witness data.

SegWit facilitates and makes it safer to implement payment channels, the Lightning Network, and other more advanced scripting capabilities that will be introduced in the future.

SegWit Explained

Brief History

December 7, 2015: Pieter Wuille first presented the idea of Segregated Witness at the Scaling Bitcoin workshops in Hong Kong as a solution to Bitcoin's scalability problems.

August 23, 2017: Segregated Witness was fully activated as a soft fork on the Bitcoin network.

Benefits
SegWit increases the block capacity from 1MB to a theoretical and maximally efficient 4MB, allowing more transactions to fit in each block and reducing transaction fees.

Nodes can prune the witness data after validating the signatures or ignore it entirely when performing simplified payment verification. It prevents transaction malleability attacks, enables more complex scripts, and maintains backward compatibility, being a soft fork.

Adoption of Segregated Witness since its activation in 2017
Most wallets and exchanges now support SegWit. Bitcoin Core versions since 0.16.0 include full support for SegWit. The percentage of Bitcoin transactions using SegWit exceeded 50% in September 2019 and surpassed 75% in August 2021.

Why hasn't SegWit been fully adopted?
Since it was not a mandatory update, wallets and exchanges adopt it at their own pace.
Users had to familiarize themselves with new address formats (wrapped and native SegWit).
Misconceptions about how SegWit worked led to its scalability and fee benefits being overlooked.

3.3 MultiSig: Multi-Signature Transactions in Bitcoin

Bitcoin has multi-signature functionality (abbreviated as 'multisig'), in which the movement of funds can be configured to require more than one signature—a quorum of signatures to authorize transactions, thus increasing security.

Corporate Use: MultiSig wallets can be useful in a corporate environment where multiple people need to approve the movement of funds. They can also be used to facilitate custodial services, especially in larger transactions with unknown entities.

Individual Setup: A single individual can create a MultiSig setup where they own all the keys, storing each of the keys on different devices (e.g., mobile phone, laptop, and hardware wallet), with the requirement that signatures from two of the three keys authorize a transaction.

Protection Against Attacks: MultiSig setups can help protect users against phishing attacks and malware infections. Even if one of the mentioned devices is lost, stolen, or compromised, a single key will not be sufficient to access the funds; the original owner can still access their funds using the remaining two keys.

This functionality is crucial to improve the security and reliability of Bitcoin transactions, providing an additional layer of protection and control over funds.

The most common condition for multi-signature transactions is to use an M-of-N scheme. In a 2-of-3 scheme, three public keys are listed as potential signers, and at least two of them must be used to create signatures for a valid transaction and spend the funds.

MuSig: A new multi-signature scheme based on Schnorr that is under development is designed to make Bitcoin multi-signature transactions less complex without sacrificing privacy.

Due to MuSig's innovative key aggregation feature, this signature is a regular Schnorr signature that can be processed by Bitcoin since the activation of Taproot. When used to create multisig wallets, MuSig reduces transaction fees and increases privacy compared to the traditional way of using the CHECKMULTISIG opcode for n-of-n signatures, which requires n public keys and n ECDSA signatures on the blockchain.—Jonas Nick, Tim Ruffing

Taproot, activated on the Bitcoin network in November 2021, allows for more complex transactions and scripts, including those utilizing Schnorr and MuSig signatures. Taproot improves privacy by making multisig transactions indistinguishable from regular transactions.

The BIP that defines Taproot is **BIP 341.** This Bitcoin Improvement Proposal introduces

Pay-to-Taproot (P2TR), which combines the functionality of Pay-to-PubKey (P2PK) and **Pay-to-Script-Hash (P2SH)**, offering users greater flexibility and privacy benefits.

In addition to BIP 341, Taproot also includes:

BIP 340: Implements Schnorr signature technology, which is more secure and flexible, allowing for key aggregation.

BIP 342: Updates the Bitcoin script language (Tapscript) to accommodate Schnorr signatures and Taproot technology.

3.4 Bitcoin Script

Bitcoin Script is a **stack-based language**, in **reverse Polish notation**, and **Turing-incomplete**, used in the Bitcoin protocol for transaction scripts. It is an **interpreted language, not compiled**. It is specifically designed to process and validate Bitcoin transactions.

Bitcoin Script is different from traditional programming languages that require a compiler for several reasons. It is not a compiled language but an interpreted one, designed for processing specific, secure, and deterministic transactions within the Bitcoin network.

Bitcoin nodes interpret the script at the time of transaction verification. The language is intentionally limited in complexity to avoid security risks and ensure that transaction verification remains fast and deterministic.

By being Turing-incomplete, Bitcoin Script avoids loops and more complex control structures that could lead to infinite loops or other computational risks, increasing security and predictability.

It is specifically designed to define the conditions under which a Bitcoin transaction can occur rather than for general-purpose computation. If the final output of the interpretation is True, the transaction happens. If False, the transaction is rejected and does not occur.

Bitcoin Script operates using a stack-based execution model, where operations are pushed and popped during script execution, similar to the Assembly language. The deterministic nature of stack-based operations ensures consistent transaction validation results across all Bitcoin nodes.

The language includes a set of basic commands (OPCODES) for handling cryptographic functions, conditionals, and other simple operations necessary for transaction scripts. The absence of complex constructs like functions or classes reflects its specialized role in the Bitcoin ecosystem.

Examples of Bitcoin Script usage include:

P2PKH (Pay-to-PubKey-Hash): A standard transaction script type that locks Bitcoin to a specific public key hash.

P2SH (Pay-to-Script-Hash): Allows more complex scripts to be executed at spending time, facilitating multi-signature transactions and other advanced features.

Locking Script: ScriptPubKey

ScriptPubKey is a script included in the output of a Bitcoin transaction. It specifies the conditions that must be met for that output (UTXO) to be spent in a future transaction. Basically, it defines 'who' or 'what' can spend this UTXO.

Opcode: Scripts in Bitcoin are composed of a series of opcodes, which are instructions that the Bitcoin virtual machine (Bitcoin Script) executes.

Public Key or Public Key Hash: The ScriptPubKey typically contains the recipient's public key or public key hash.

One of the most common ScriptPubKey formats is P2PKH (Pay-to-PubKey-Hash). Let's look at an example to understand it better.

Example of P2PKH

When Alice sends Bitcoin to Bob, the transaction creates an output that includes a ScriptPubKey like this:

```
OP_DUP OP_HASH160 <Bob's Public Key Hash> OP_EQUALVERIFY OP_CHECKSIG
```

OP_DUP: Duplicates the public key that will be provided in the spending transaction (i.e., creates a copy of the item on top of the execution stack).
OP_HASH160: Performs a RIPEMD-160 hash followed by a SHA-256 hash on the duplicated public key.
<Bob's Public Key Hash>: This is Bob's public key hash, provided by Alice when creating the transaction.
OP_EQUALVERIFY: Compares the two values on top of the stack (the calculated hash of the public key provided in the spending transaction and Bob's public key hash). If they are equal, verification continues; otherwise, the transaction fails.
OP_CHECKSIG: Verifies that the signature provided in the spending transaction is valid for the provided public key.

Conditions to Spend the UTXO
To spend the UTXO protected by this ScriptPubKey, Bob needs to create an input transaction that provides:

Public Key: This public key will be used in the verification script (ScriptSig) of the input transaction.

Valid Signature: This signature must match the public key and the transaction, proving that Bob possesses the private key corresponding to the specified public key.

Unlocking Script—ScriptSig: Provides the necessary data to satisfy the conditions of the ScriptPubKey. If the result of the operation is True, the value is released.

```
<Bob's Signature> <Bob's Public Key>
```

When this spending transaction is executed, the script combines the ScriptSig with the ScriptPubKey as follows:

Bob's public key is duplicated (OP_DUP) and then hashed (OP_HASH160).

The resulting hash is compared to the public key hash stored in the UTXO (OP_EQUALVERIFY).

If the hash is equal, Bob's signature is verified against the public key and the transaction (OP_CHECKSIG).

If all these conditions are met, the UTXO is considered spent, and the transaction is validated.

The ScriptPubKey defines the conditions under which a UTXO can be spent, ensuring that only the holder of the private key corresponding to the specified public key (or public key hash) can spend the funds. This is fundamental to the security and integrity of Bitcoin transactions.

MultiSig Signature

The general form of an M-of-N multi-signature transaction script is as follows:

```
M <Public Key 1> <Public Key 2> ... <Public Key N> N OP_CHECKMULTISIG
```

In the case of a 2-of-3 multi-signature transaction, the locking script is as follows:

```
2 <Public Key A> <Public Key B> <Public Key C> 3 OP_CHECKMULTISIG
```

2 means that 2 signatures are required.

<Public Key A> <Public Key B> <Public Key C> are the 3 public keys involved. 3 indicates that there are 3 public keys in total.

OP_CHECKMULTISIG is the opcode that verifies multiple signatures.

Unlocking Script

The script above forms a "locking script," which can only be unlocked by an equivalent "unlocking script" containing 2 or more signatures calculated from the private keys of the signers, corresponding to the listed public keys:

```
OP_0 <Signature B> <Signature C>
```

OP_0 is necessary due to a historical bug in Bitcoin that requires an additional zero for the MultiSig script.

<Signature B> <Signature C> are two valid signatures corresponding to two of the public keys mentioned in the Locking Script.

Validation Script

When a transaction is verified, the Unlocking Script and the Locking Script are combined to form the Validation Script:

```
OP_0 <Signature B> <Signature C> 2 <Public Key A> <Public Key B> <Public Key C> 3 OP_CHECKMULTISIG
```

This configuration allows a multi-signature transaction to be correctly validated, ensuring that the required signatures correspond to the public keys listed in the script.

Learn more about transactions by reading the article Bitcoin P2PKH Transaction Breakdown by Rachel Rybarkzyc—Board Member of Scalar School

3.5 Miniscript

Miniscript is a language developed to write Bitcoin scripts in a structured way, allowing for analysis, composition, generic signing, and more. It was designed and implemented by Pieter Wuille, Andrew Poelstra, and Sanket Kanjalkar at Blockstream Research. The goal of Miniscript is to facilitate the creation of complex spending conditions and ensure the security, efficiency, and interoperability of Bitcoin scripts.

Miniscript offers a structured representation of Bitcoin scripts, allowing software to automatically analyze the script and determine which witness data needs to be generated to spend the bitcoins protected by that script.

The Bitcoin scripting language allows scripts to be composed of smaller, valid expressions, making it easier to create complex spending conditions. This is especially useful for wallet developers, who don't need to rewrite code when switching from one script model to another.

Miniscript optimizes transaction compilation, which can result in a smaller footprint on the blockchain, saving transaction fees. Additionally, the structure of Miniscript facilitates the creation of more complex and secure smart contracts directly on the Bitcoin base chain.

Decreasing Multisig: Allows the creation of multisig wallets where the number of required signatures can decrease over time or after a specific period, providing additional flexibility in cases like key loss.

Inheritance and Timelocks: Facilitates the creation of inheritance wallets where funds can be accessed by heirs after a specific period, functioning as a kind of automated inheritance switch.

Bitcoin Script is a stack-based language with many special cases, designed to implement spending conditions consisting of various combinations of signatures, hash locks, and time locks. However, working directly with Bitcoin Script can be difficult and error-prone, especially for complex spending conditions. Miniscript addresses these issues by providing a more structured and easier-to-understand way to compile Bitcoin scripts.

Miniscript promises to facilitate the development of more secure and efficient Bitcoin scripts, improving usability and security for developers and end-users. While adoption may take some time, its advantages in terms of security, flexibility, and efficiency make it a valuable tool for the future of Bitcoin transaction development.

To learn more about Bitcoin Miniscript, visit https://bitcoinops.org/en/topics/miniscript/

3.6 CoinJoin

CoinJoin is a Bitcoin transaction where multiple users combine their UTXOs, improving privacy. It is one of the essential tools for preserving human rights in the context of using Bitcoin. It allows transactions to be mixed in a way that enhances user privacy, making it more difficult to trace the origin and destination of funds. In a world of increasing financial surveillance, protecting transaction privacy is crucial to ensuring the freedom and safety of activists, journalists, and ordinary citizens living under oppressive regimes.

Greg Maxwell wrote on the Bitcointalk forum in 2013: "Bitcoin is often promoted as a privacy tool, but the only privacy that exists in Bitcoin comes from pseudonymous addresses, which are fragile and easily compromised through reuse, taint analysis, payment tracking, IP address node monitoring, web-spidering, and many other mechanisms. Once broken, that privacy is difficult and sometimes costly to recover..."

Adopting CoinJoin from the start of your Bitcoin journey is vital. This not only protects individual privacy but also strengthens the network as a whole, making it harder for malicious entities to track transactions and identify users. Financial privacy is a pillar of freedom and autonomy, and its widespread use can be a powerful tool to preserve a future where human rights are respected and protected.

Transaction Combination: CoinJoin is a method where multiple users combine their transactions into a single Bitcoin transaction. This means that multiple inputs and outputs from different users are combined into a single block.

Anonymity through Mixing (Coin Mixing): By combining transactions from multiple users, it becomes difficult to determine which output belongs to which original input. This increases privacy, as external observers cannot easily trace the origin and destination of funds.

Process Steps

Coordination: A CoinJoin coordinator or server gathers transaction intentions from multiple participants.

Partial Signatures: Each participant creates and partially signs their parts of the transaction.

Combination: The coordinator combines all parts into a single complete transaction.

Final Signature: Each participant signs the complete transaction.

Broadcast: The finalized transaction is sent to the Bitcoin network.

CoinJoin Tools

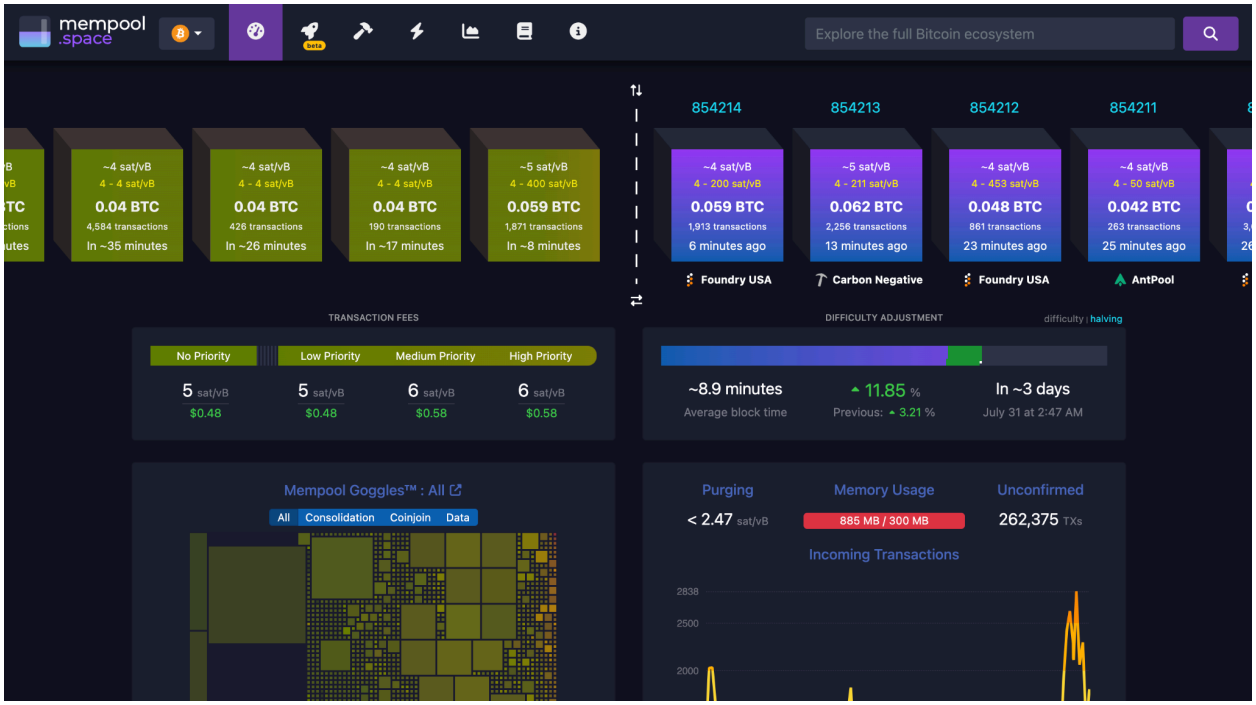
Wasabi Wallet

Sparrow Wallet

Jam, which is a UI interface for the JoinMarket software

3.7 Block Explorers

A block explorer is an indispensable tool for viewing and interacting with the Bitcoin network. It allows you to access a wide range of detailed information about transactions, blocks, and addresses. Here's a snapshot of the block explorer mempool.space.



Transaction Information

With a block explorer, you can query transaction details such as inputs, outputs, network fees, and change. This makes it easy to observe how each transaction is connected to previous transactions, promoting transparency and traceability.

You can check the number of confirmations a transaction has, indicating its security and immutability, as well as the time it was included in a specific block.

Additionally, you can view the locking scripts (ScriptPubKey) and unlocking scripts (ScriptSig) of transactions and obtain detailed information about each input and output, including the addresses involved and the amounts transferred.

Mined Blocks

Mined blocks have detailed information such as the responsible miner, block size, the number of transactions included, and the total block reward, which includes the base reward and the transaction fees collected.

You can view the current balance of a specific address and its complete transaction history, offering a clear view of fund movements.

Network Statistics

Block explorers also provide access to network statistics. You can see the current hash rate, indicating the total computational power used to mine blocks, and the mining difficulty, which adjusts periodically to keep the block time constant.

Additional Features

Some block explorers offer additional features such as alerts and notifications. You can set up alerts to be notified when a specific transaction is confirmed or when an address sends or receives funds. This helps monitor relevant activities on the blockchain in real-time.

APIs for Developers

Block explorers can also offer data APIs, allowing developers to programmatically access blockchain information to integrate into their applications. This is essential for developing tools and services that interact directly with the Bitcoin blockchain.

Privacy and Security with Your Own Node

When you access a block explorer in your browser, it makes a request to a Bitcoin network node to obtain specific data. The node processes the request, extracts relevant data from the blockchain, and sends it back to the explorer, which then presents this information in a user-friendly and accessible way. Generally, these nodes are operated by third parties such as companies or organizations. While convenient, there are risks associated with trusting these nodes, such as privacy issues, data reliability, censorship, and security.

To mitigate these risks, it is possible to install a block explorer on your own Bitcoin node. This offers full control over the data queried, avoiding third parties monitoring your activities and ensuring you are accessing information directly from a trusted node.

Information Available on Block Explorers

Hash Rate: Indicates the total computational power being used to mine blocks.

Mining Difficulty: Adjusted periodically to maintain constant block time.

Total Number of Nodes: Displays the number of active nodes on the Bitcoin network.

Block Height: The number of the most recent block in the blockchain.

Transaction Volume: Total number of confirmed transactions over a specific period.

Blockchain Size: The total size of the blockchain in gigabytes.

Transaction Confirmations: The number of confirmations a transaction has received, indicating its security and immutability.

Transaction Time: The timestamp when the transaction was included in a block.

Locking and Unlocking Script: View the locking scripts (ScriptPubKey) and unlocking scripts (ScriptSig) of transactions.

Input and Output Details: Detailed information about each input and output, including the addresses involved and the amounts transferred.

Transaction Fee: The fee paid to include the transaction in the block.

Transaction Size: The total size of the transaction in bytes.

Mempool: Pending transactions that have not yet been confirmed and included in a block.

Miner: The identification of the miner who mined the block (when available).

Block Size: The total size of the block in bytes.

Number of Transactions: The number of transactions included in the block.

Block Reward: The total reward received by the miner, including the base reward and the transaction fees.

Mining Time: The time it took to mine the block.

Nonce: The value miners adjust to find a valid hash for the block.

Block Version: The block's version number.

Current Address Balance: The total balance of a specific address.

Transaction History: All transactions involving the address, including received and sent amounts.

Unconfirmed Balance: The balance of transactions that have not yet been confirmed.

First and Last Transaction: Dates of the first and last transactions associated with the address.

Average Transaction Fees: Average transaction fees paid over a specific period.

Fee Estimates: Recommended fee estimates for transactions to be confirmed within a certain number of blocks.

Fees Paid per Block: The total transaction fees paid in each block.

Block explorers are versatile tools that provide a wide range of information and functionalities, from basic transaction details to advanced analytics and API integrations. They are essential for users who want to understand the workings of the Bitcoin blockchain better, monitor their transactions, and develop applications that interact with the network.

You can use any block explorer to search the entire blockchain, as all transactions are publicly visible. To get started, see a list of them at http://bit.ly/blockexplorers. The most used are mempool.space and blockstream.info. Choose any transaction, follow it, and see what you can discover about the inputs, outputs, network fees, and change. Observe how it is connected to previous transactions.

You can search for the ID of the first Bitcoin transaction, which is essentially the first txid ever created on the network.

First Bitcoin transaction: Satoshi Nakamoto sent 10 BTC to Hal Finney

```
f4184fc596403b9d638783cf57adfe4c75c605f6356fbc91338530e9831e9e16
```

Blockchain Data Structure

The interactive blockchain tool on Anders Brownworth's website is an educational resource designed to help users understand the fundamental concepts of blockchain technology. It simulates how a blockchain works, allowing you to explore and visualize the processes of block creation, hashing, block chaining, and how transactions are securely and immutably recorded.

https://andersbrownworth.com/blockchain/blockchain

3.8 Transaction Fees

Transaction fees are small amounts of Bitcoin that users pay to have their transactions processed and confirmed by miners. These fees serve as an incentive for miners to include the transaction in the next block they mine.

Another important feature of block explorers is the analysis of transaction fees. You can look up the average fees paid over a specific period and get recommended fee estimates to have transactions confirmed quickly. This is particularly useful for users who want to optimize the cost of their transactions.

The fee is not fixed and can vary based on several factors:

Transaction Size: Larger transactions (in terms of data size, not the amount of Bitcoin) require more space in a block, which can increase the fee. Using multiple small UTXOs has a higher cost than a single UTXO of equivalent value.

Network Congestion: When the network is busy, fees tend to rise as users compete to have their transactions confirmed quickly.

User Priority: Users can choose to pay higher fees to speed up their transactions. Higher fees generally result in faster confirmations.

Transaction fees are usually calculated per byte of data in the transaction. Wallet software often suggests an appropriate fee based on current network conditions, ensuring that the transaction is confirmed within a reasonable time. Users can set their fees manually, but setting a fee too low can result in delayed confirmations.

When you send Bitcoin over the network, the wallet software allows you to enter the fee amount you want to pay. Knowing the average fee at the time of the transaction and choosing a value slightly above it increases the likelihood that your transaction will be confirmed in the desired time.

Virtual Bytes (vbytes)

Virtual bytes (vbytes) are a unit of measure used to calculate transaction fees in Bitcoin more efficiently and fairly, especially after the activation of Segregated Witness (SegWit). They help better reflect the actual impact of a transaction on the blockchain.

SegWit was an upgrade to the Bitcoin protocol that, among other things, separated the signature (witness) from the transaction data. This update allowed for better utilization of block space, increasing transaction capacity without changing the maximum block size of 1 MB.

Before SegWit, transaction fees were based on the total transaction size in bytes. After SegWit, transactions have two parts: regular data and witness data. Witness data is less critical for

network propagation and validation, so a new way to measure the impact of transactions was needed.

Definition and Calculation of vbytes

A vbyte is a unit of measure that adjusts the actual size of a transaction, considering the differentiated weight of witness data. The basic formula to calculate vbytes is:

$$\text{vbytes} = \frac{\text{weight}}{4}$$

Where weight is a metric that combines the size of regular data and witness data.

Weight Units (WU)

Each byte of regular transaction data counts as 4 weight units.

Each byte of witness data counts as 1 weight unit.

For example, if a transaction has:

100 bytes of regular data: 100 × 4 = 400 weight units

200 bytes of witness data: 200 × 1 = 200 weight units

Total weight: 400 + 200 = 600 weight units

Then, the vbytes would be calculated as:

$$\text{vbytes} = \frac{600}{4} = 150$$

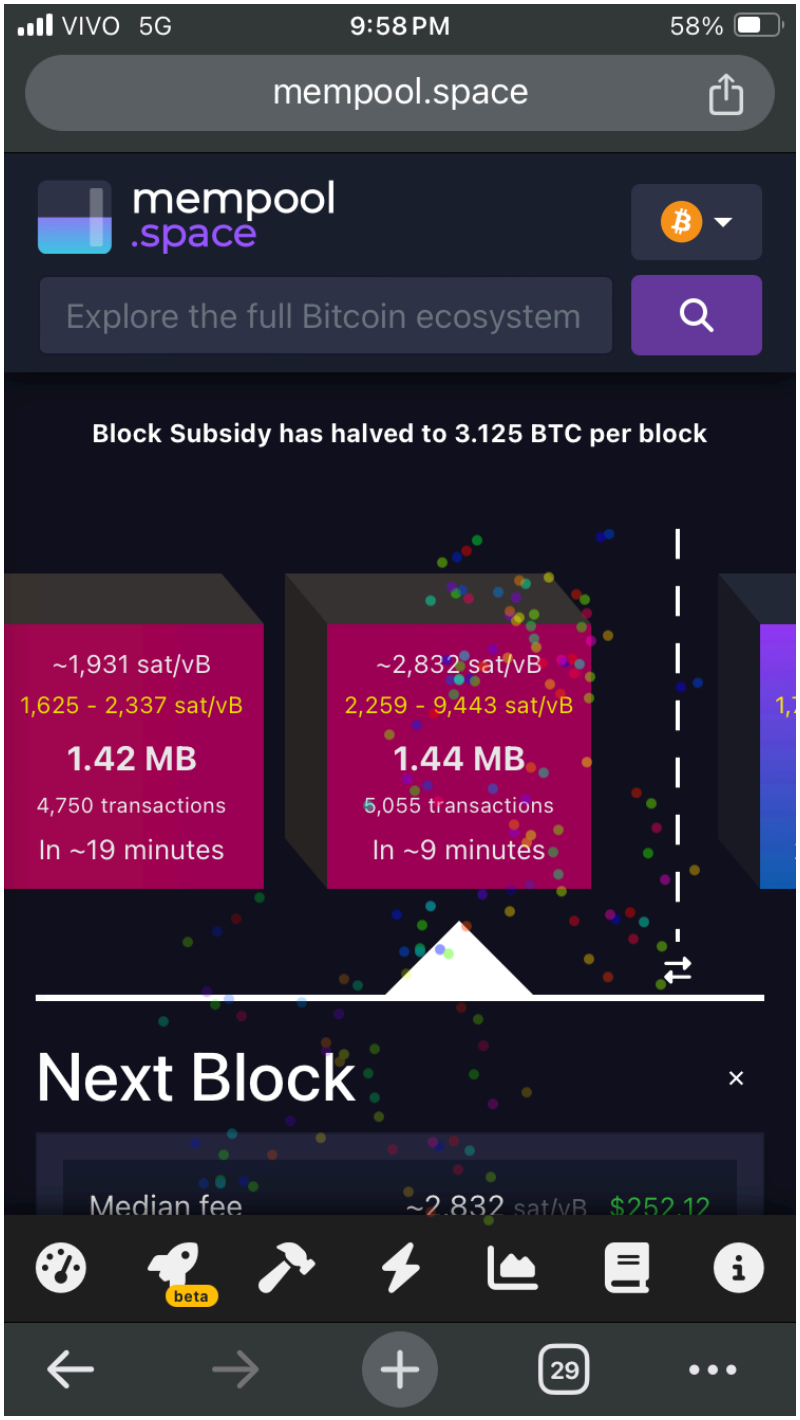
Using vbytes in the calculation of transaction fees allows for a better correlation between the fee paid and the actual use of block space, promoting a more efficient allocation of network

resources. SegWit transactions tend to be more compact in terms of vbytes, resulting in lower fees. This incentivizes users to adopt SegWit, increasing the network's effective capacity.

Replace-by-Fee (RBF) is a feature in the Bitcoin protocol that allows the sender of a transaction to replace a pending (unconfirmed) transaction with a new transaction that has a higher transaction fee. The main purpose of RBF is to increase the likelihood of transaction confirmation during network congestion. Many wallet software applications have this feature built-in.

Bitcoin Value: To know its equivalent value in your local currency, you can search for "Bitcoin price" on Google or use currency conversion tools. Google usually has an embedded conversion function when you search for the price, but there are other alternatives, such as https://cointradermonitor.com/. It's always good to check more than one source.

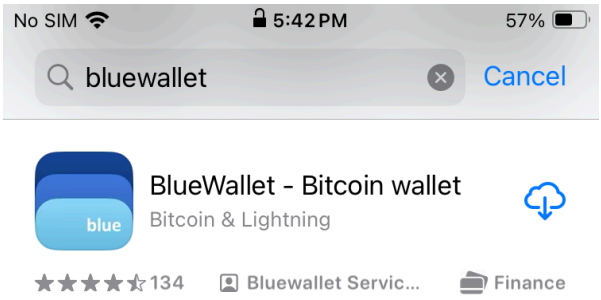
Historical Curiosity: The Bitcoin network fee reached its peak value on the day of the 2024 halving, on April 19th. This indicates two things. First, when there are no more block subsidies available, miners can live off network fees. Second, layer 2 scalability solutions like the Lightning Network are extremely important to keep transaction fees low and the Bitcoin network efficient. I took a screenshot on my phone, as at that moment, I was in a Discrete Mathematics class at Fatec. 2832 sats/vB.


VIVO 5G
9:58 PM
58%
mempool.space
mempool
.space
Explore the full Bitcoin ecosystem
Block Subsidy has halved to 3.125 BTC per block
~1,931 sat/vB
1,625 - 2,337 sat/vB
1.42 MB
4,750 transactions
In ~19 minutes
~2,832 sat/vB
2,259 - 9,443 sat/vB
1.44 MB
5,055 transactions
In ~9 minutes
Next Block
Median fee
~2,832 sat/vB
$252.12
beta
29

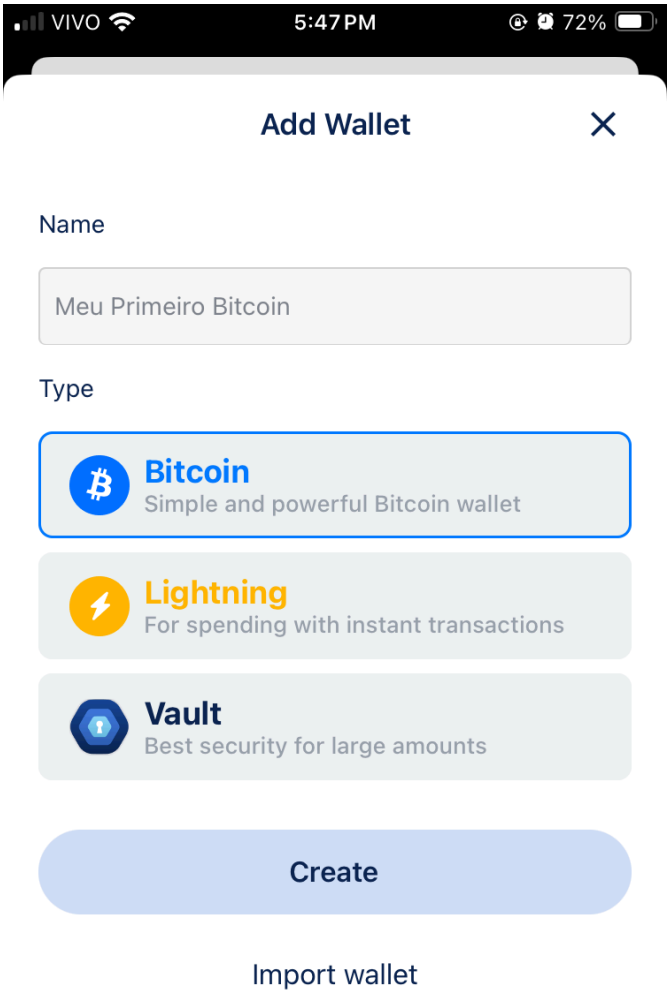
3.9 Practice: My First On-Chain Bitcoin

Let's experience a Bitcoin transaction in practice. Within a single Bitcoin wallet app, BlueWallet, you can create multiple wallets. So, I will test sending satoshis from one wallet to another that belongs to myself. The process for sending to another person's wallet is exactly the same. Let's start.

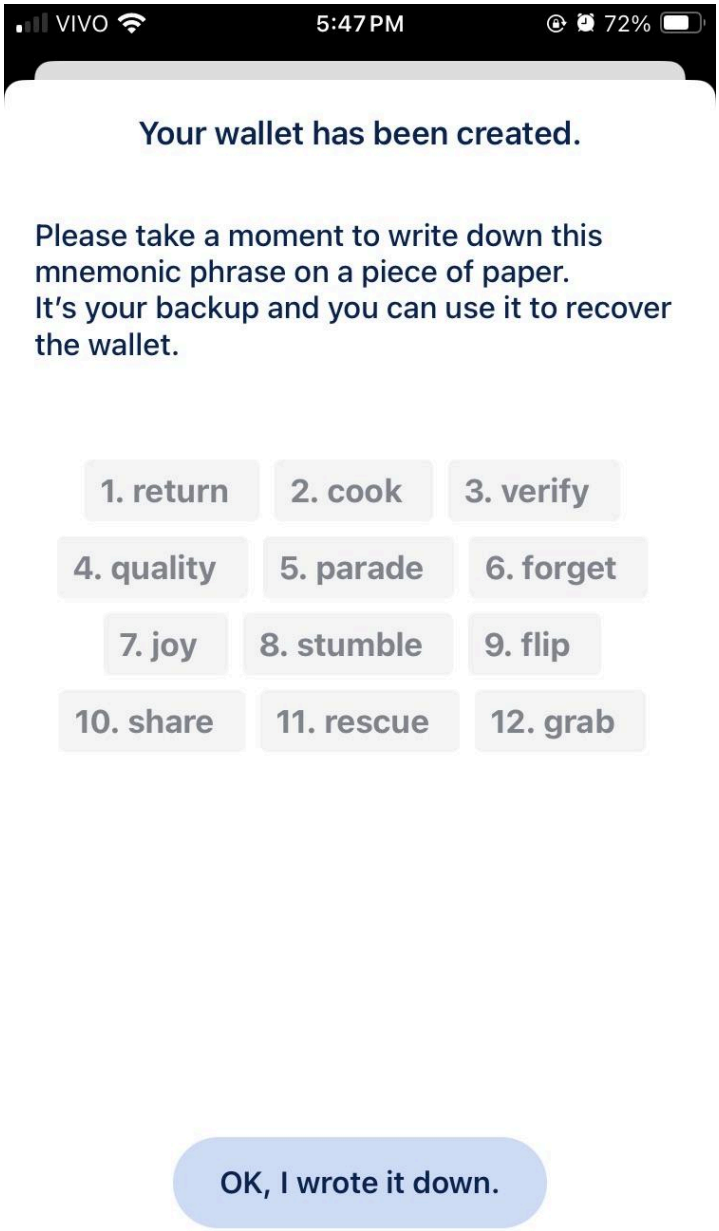
First, search for BlueWallet in the App Store or Play Store.



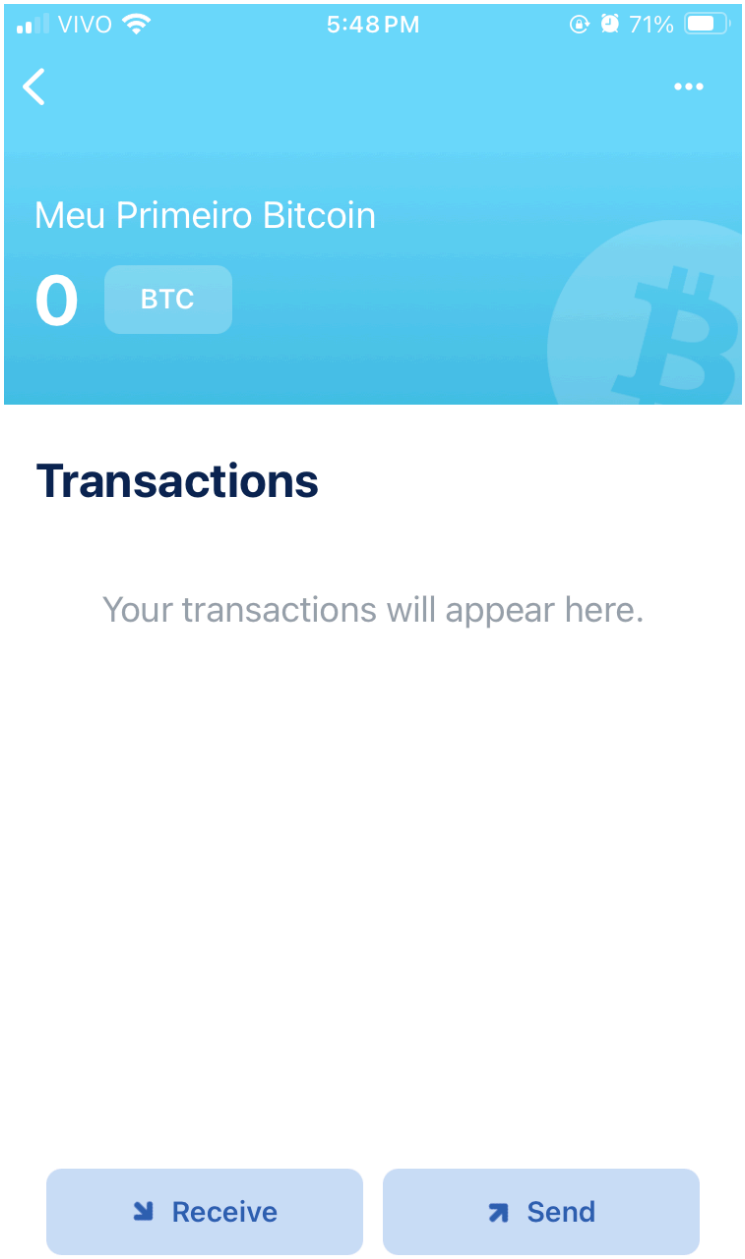
Now, in the access flow, you will have the option to create a wallet. Choose the first one, Bitcoin—Simple and powerful Bitcoin wallet.



Write down your seed, preferably with a pencil as it doesn't wash away with water or alcohol. Do not take a screenshot!

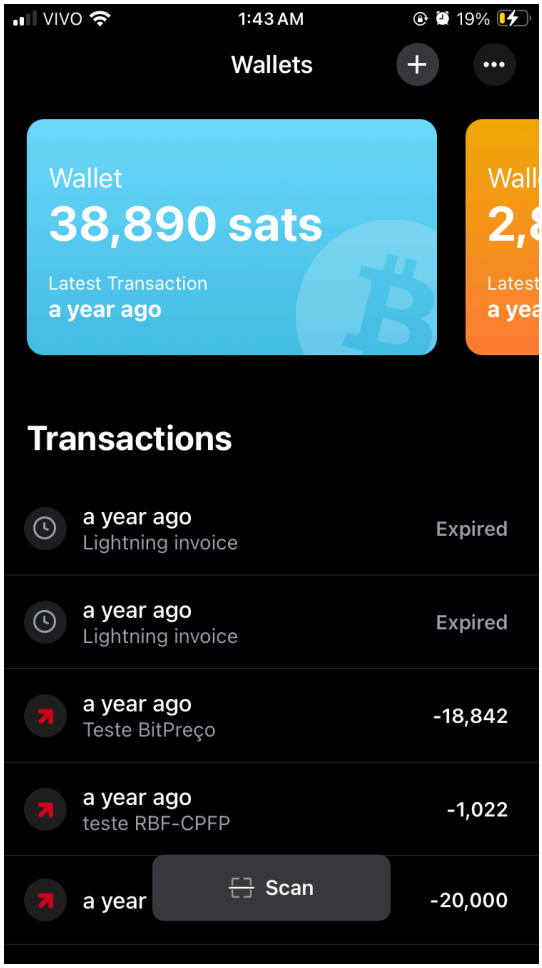


After noting it down, OK, I wrote it down will take you to the main screen of your wallet.

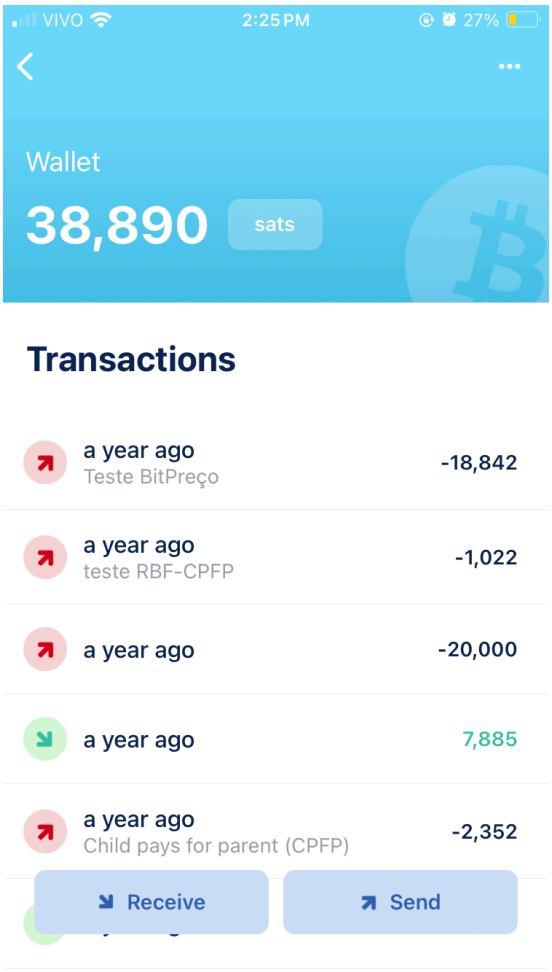


Since this wallet is empty, let's use it to receive some satoshis. Click on Receive. A receiving address will appear on the screen.

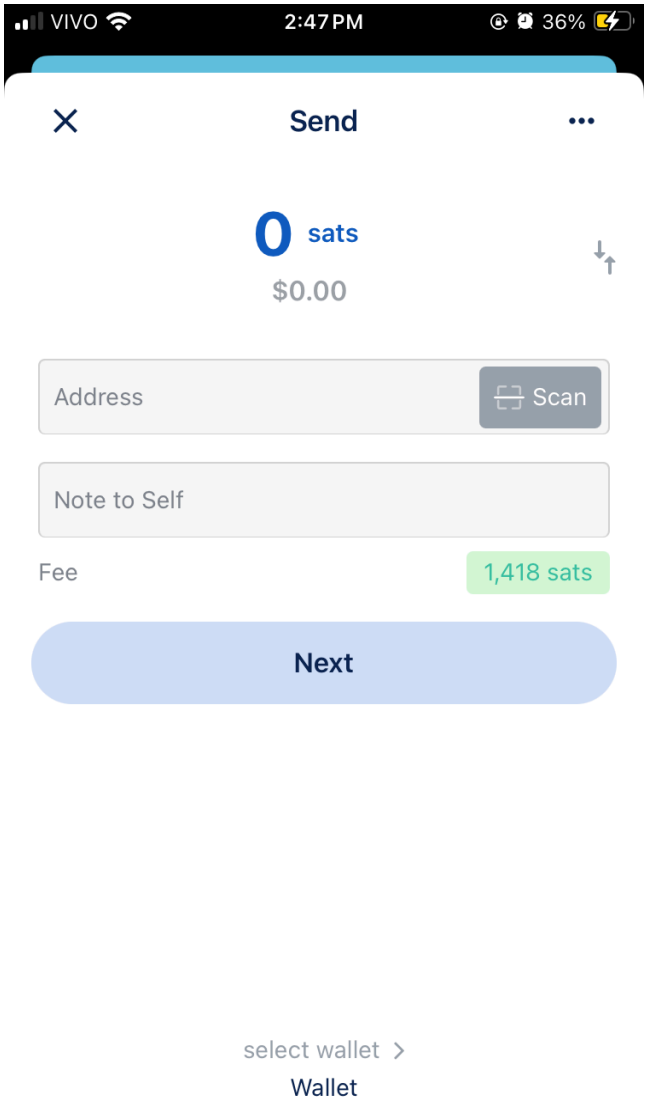
Clicking on this address will copy it to the clipboard, and I can share it with the person who will pay me, which in this case is myself. Now, let's go to another wallet that has some satoshis and try to send to this address. During the copy and paste process, be careful, double-check that all characters match exactly what was generated in the wallet, especially if it's between you and exchanges. There are malware programs out there that alter this address, and if you're not paying attention, you could be sending Bitcoin to a malicious actor.



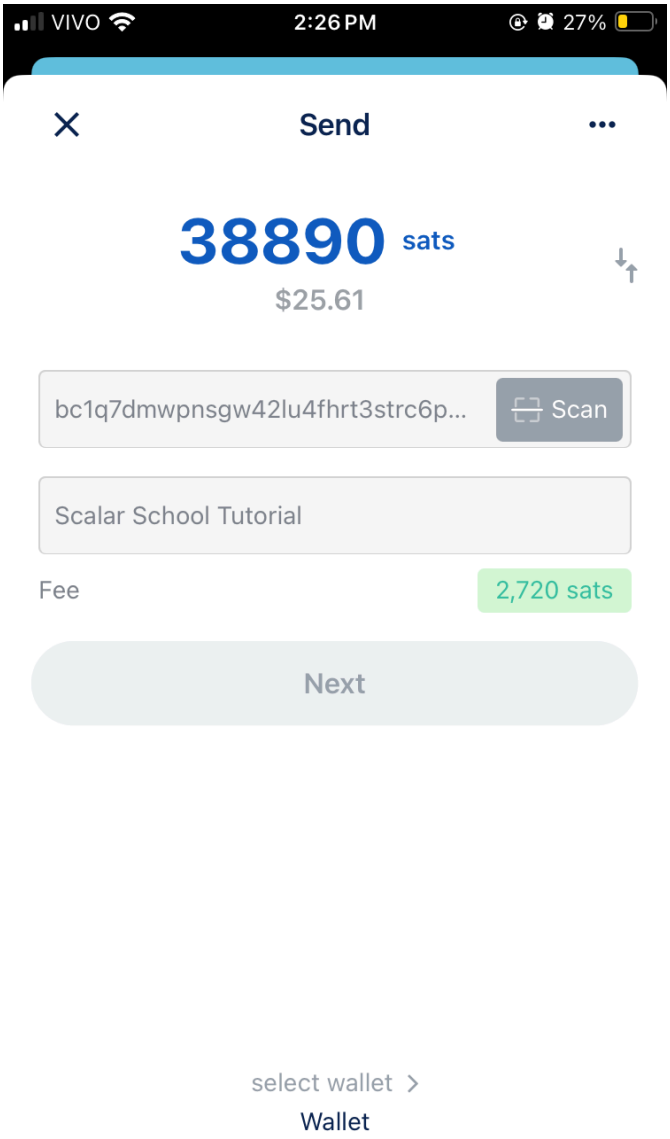
We enter the other wallet that has the funds and will send to this address of My First Bitcoin wallet.



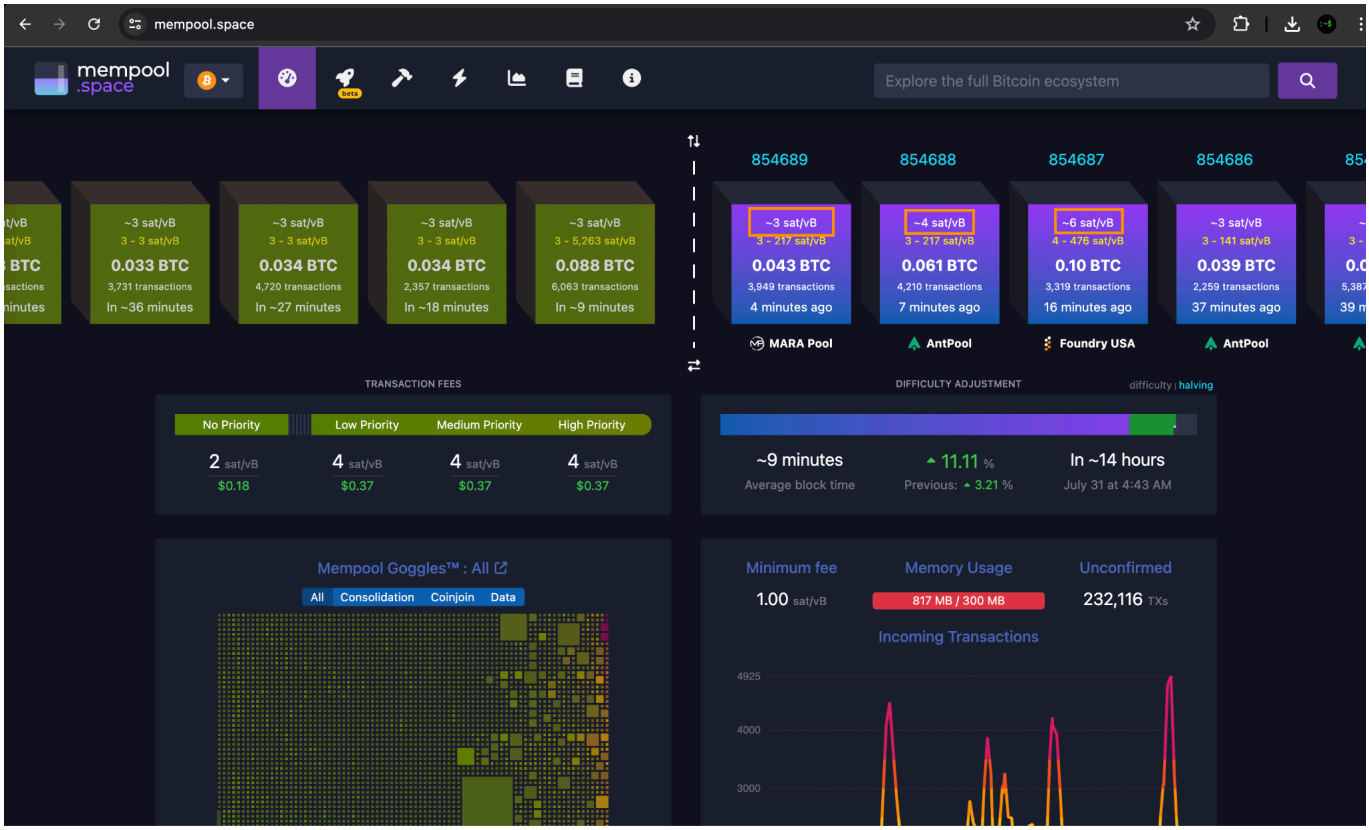
To do this, click on Send. Remember that the address to receive is already on the clipboard, so just paste it into the appropriate field.



In the Address field, paste the address. In Note to Self, write an identifying comment for this transaction (these notes do not go to the blockchain; they are metadata at the wallet app level only). Where it shows the amount, enter the amount you want to send.



In my case, I wrote the total amount. But notice that the Next button remained inactive! This is because I didn't consider the network fee that I need to pay for the transaction to be processed, which is deducted from the wallet itself. Let's check the network fee environment. Access mempool.space and see what the average fee is for transactions to be included in blocks.

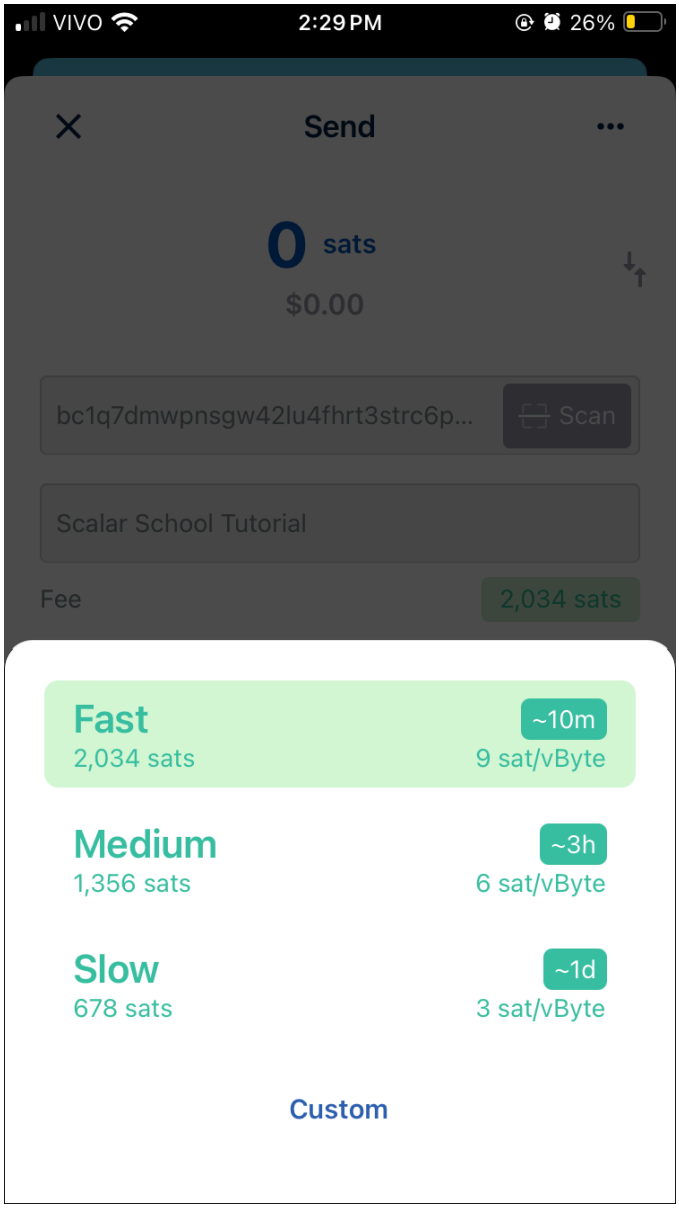


3 sats per virtual byte, 4 sats per virtual byte, 6 sats per virtual byte. Remember that each transaction has a set of data that needs to be recorded on the blockchain, and this data takes up disk space.

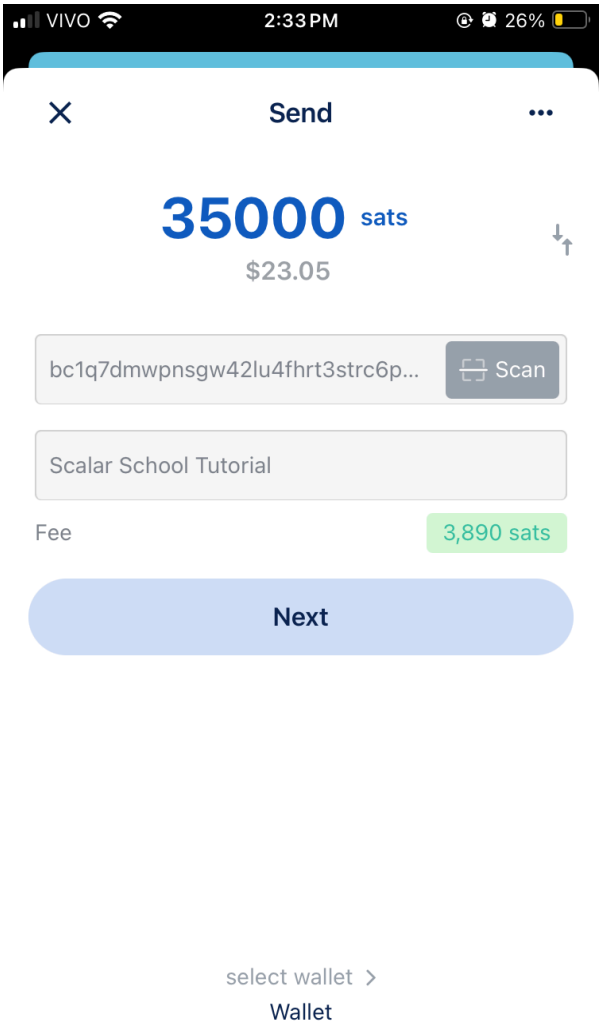
The total value of a transaction depends on how much space we are demanding the network to process. If I make a payment of 38,000 sats using numerous small UTXOs to sum up the total value, it will be more expensive than making a payment of 38,000 sats with just one output UTXO.

That's why there's a process called **UTXO consolidation**, which is when a Bitcoiner takes advantage of low network fees to send the total value from one wallet to another, consolidating several smaller UTXOs into one large one that will be much cheaper to transact in the future.

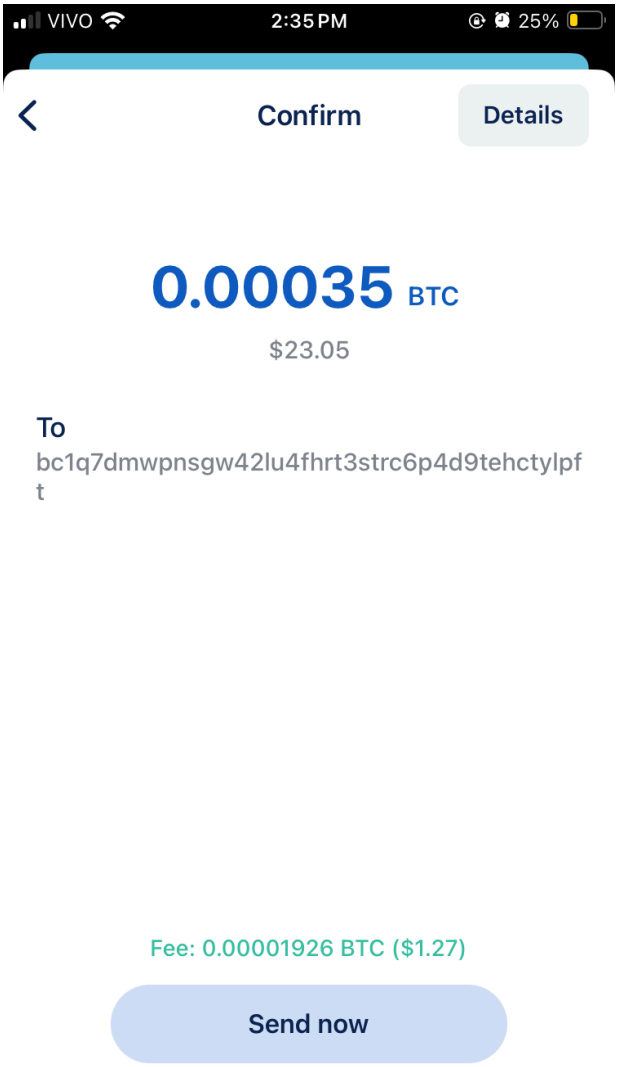
Returning to the wallet, let's click on that green fee-setting field.



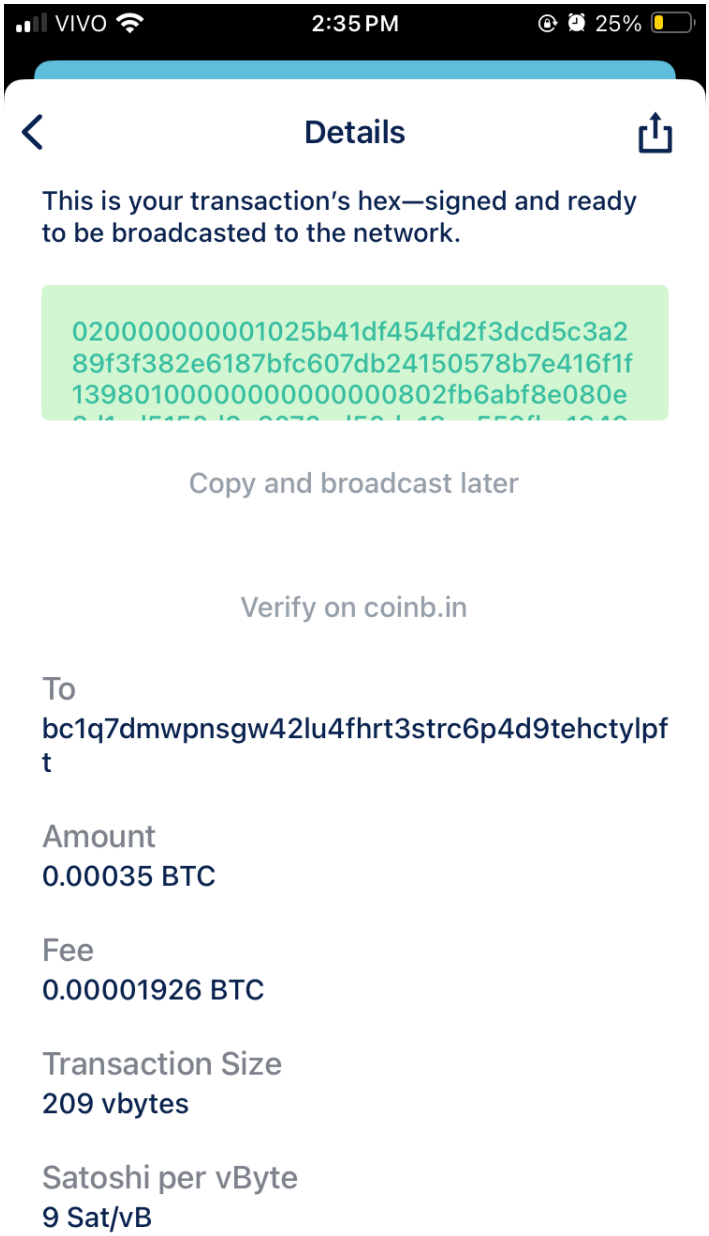
The app indicates that for our transaction to be confirmed (included in a block) quickly, we should pay around 9 sats/vByte. This is well above the value we checked on the block explorer, but that's okay, I'll accept the suggestion. Note that I kept lowering the value manually until the Next button became active, meaning until the fee value was covered.



Clicking Next, we enter this screen. The app is showing the value in BTC, but you can always choose to view it in sats. Remembering that **1 Bitcoin = 100,000,000 Satoshis.**



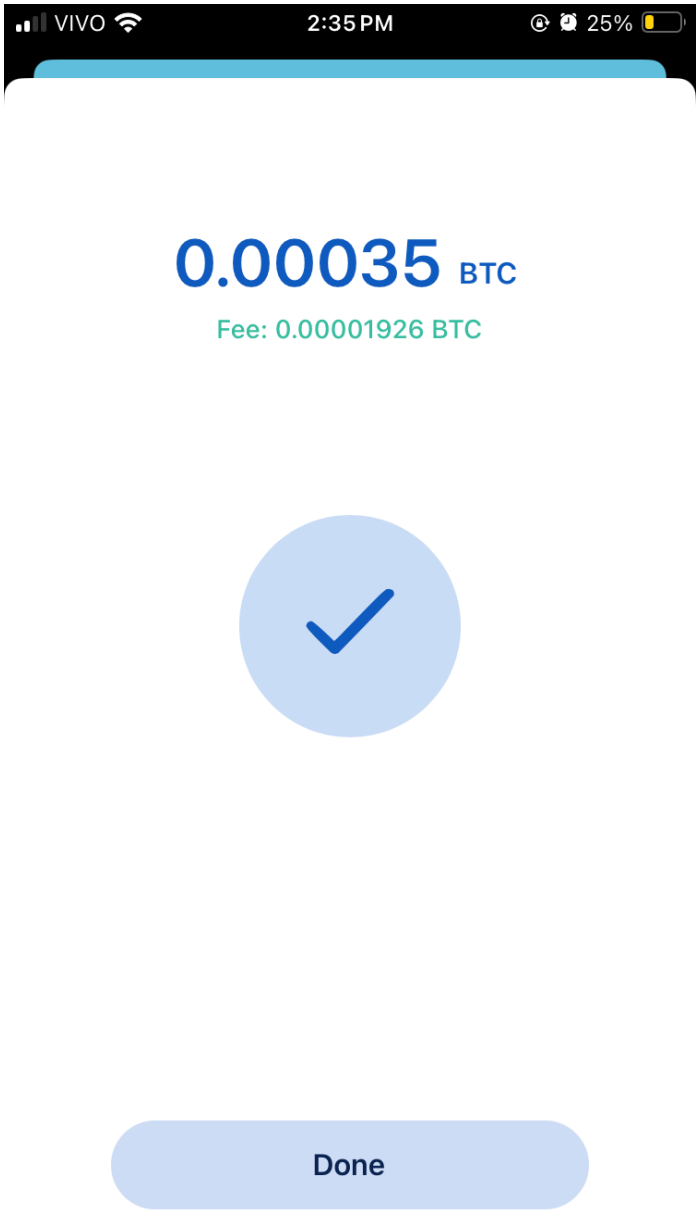
Clicking on Details, we can access more information about the structure of this transaction.



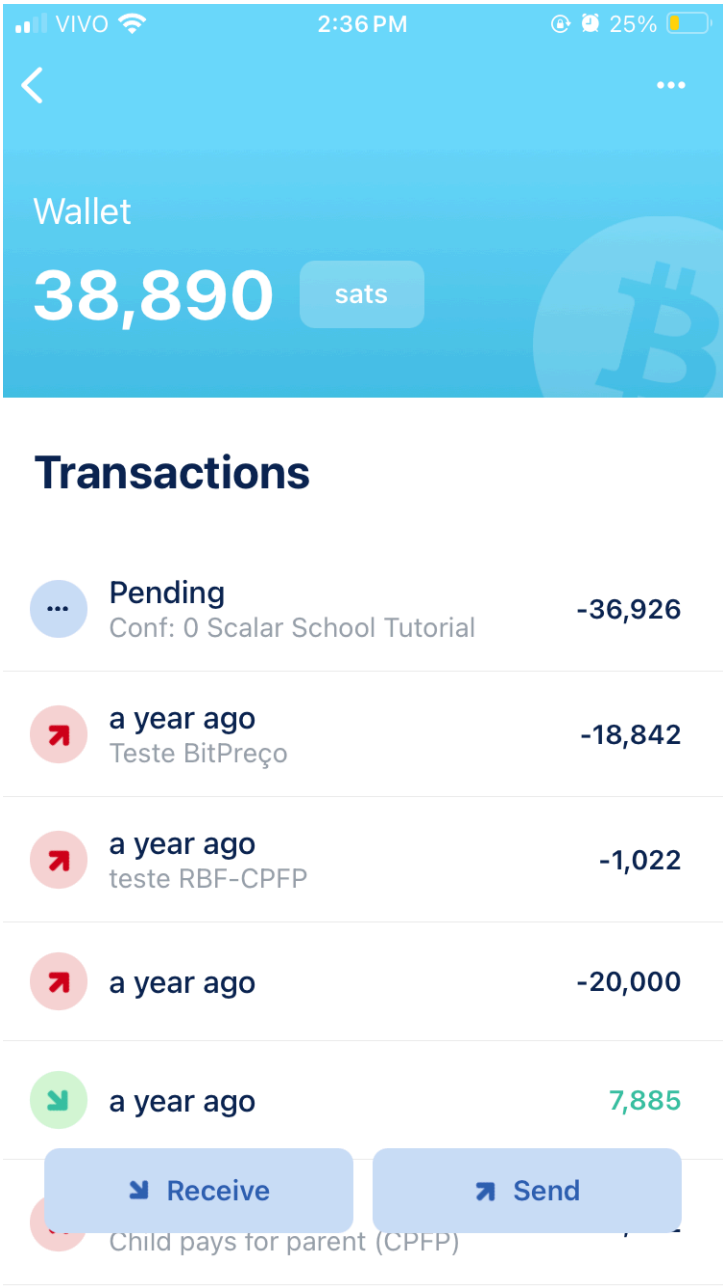
A transaction hexadecimal (hex) is a serialized representation of a Bitcoin transaction. It is essentially the raw transaction data encoded in hexadecimal format. This hex string contains all the necessary information about the transaction, including version number, inputs (previous transaction hash, index, scriptSig, sequence), outputs (value, scriptPubKey), locktime, and signatures, making it possible to broadcast it to the Bitcoin network.

You can decide to propagate the transaction through BlueWallet or copy this hex and propagate it later using other tools. Block explorers often have propagation tools, and you can also propagate through the command line of your Bitcoin Core node using the command `sendrawtransaction`.

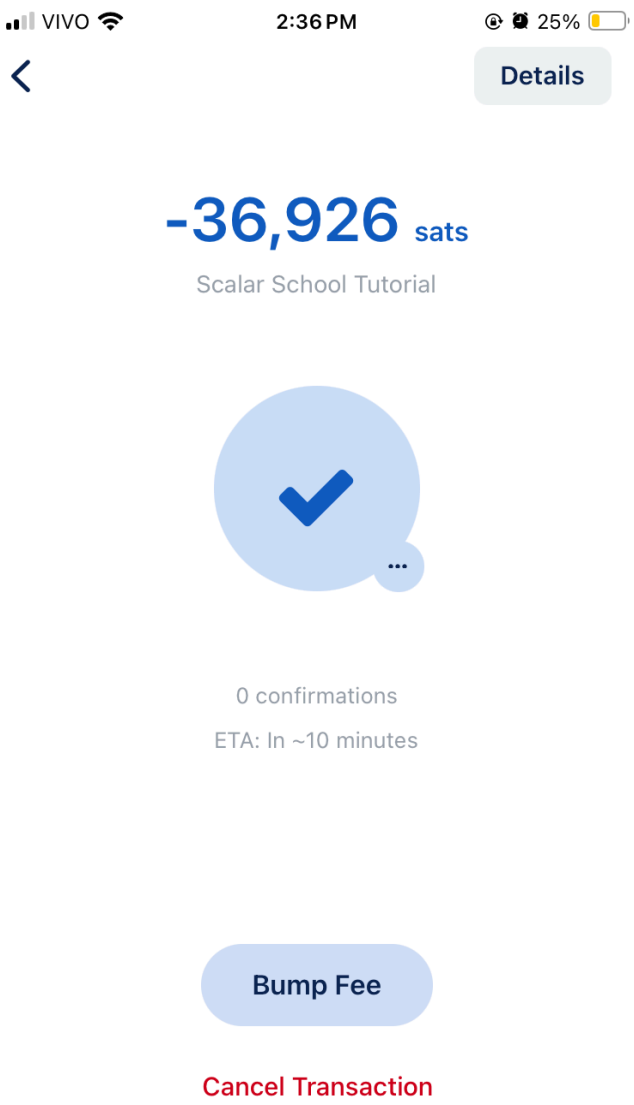
In this case, we return to the previous screen and click Send now.



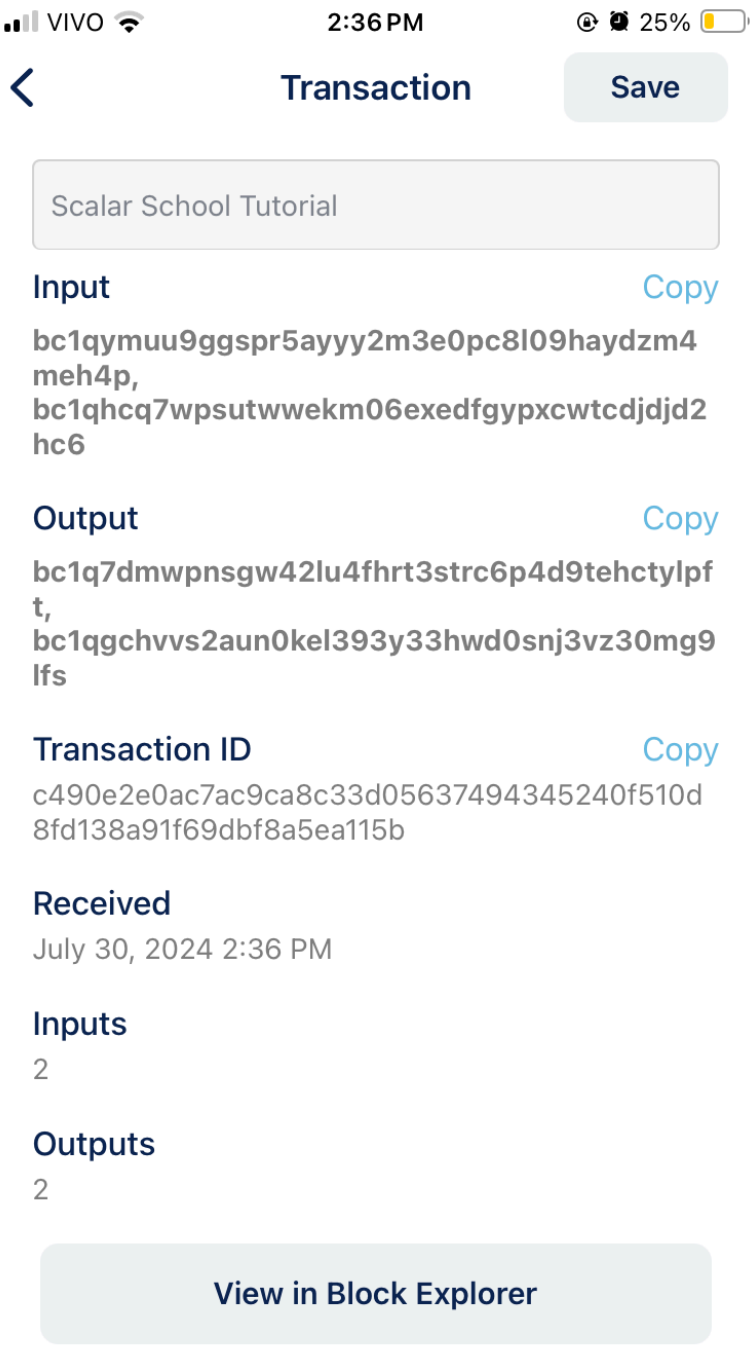
On the main screen, we see the transaction as Pending. It has been broadcast to the mempool but has not yet been included in a block.



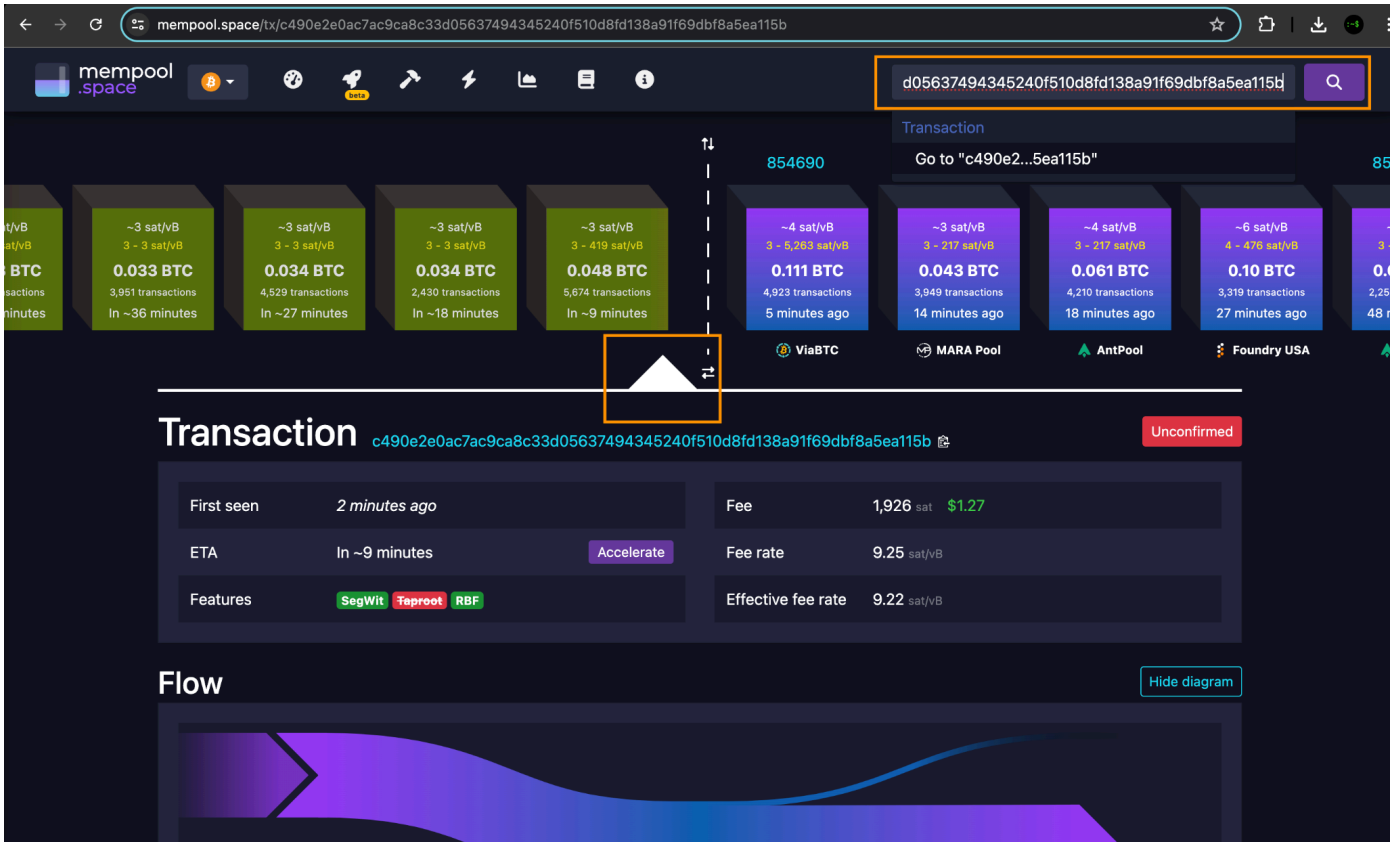
Clicking on it, we see the option to Bump fee, which is the RBF function. Let's suppose the network fee increased rapidly, and you want to update the value and add an extra fee to have your transaction keep up with the average. This is when you use Bump fee. There's also the option to cancel the transaction. But once it is confirmed or included in a block, it becomes irreversible.



Clicking on details, we can access the transaction ID.

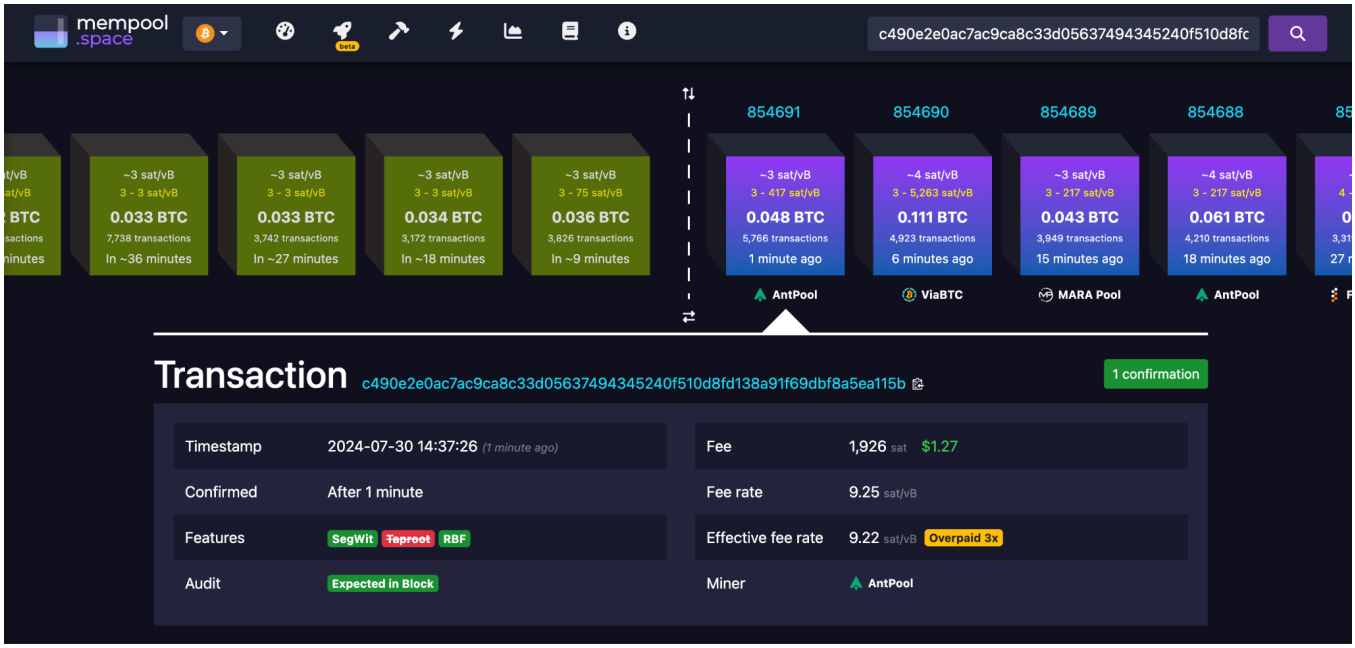


Copy this ID, and let's track our transaction until it is confirmed. You will paste this ID into the search field of a block explorer. In this case, I used mempool.space.

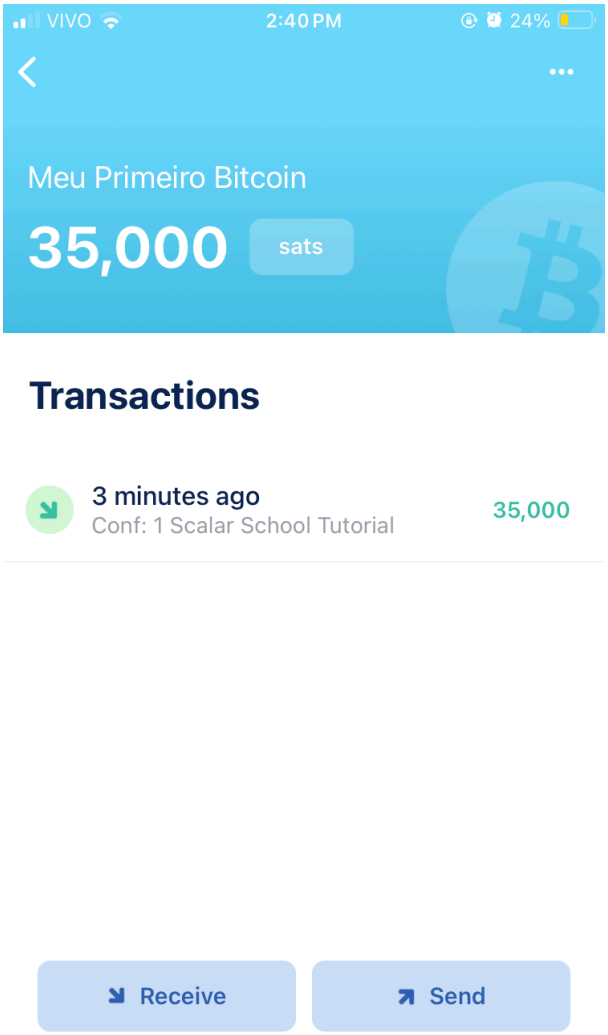


Notice that there's a white arrow pointing to the green area of blocks, which represents the mempool and a grouping of transactions by fee value—they are not yet actual blocks but transactions being considered for candidate blocks by miners. Only when this transaction is mined will it be included in a block in the purple part.

The confirmation of this transaction happened almost in real-time. Also, we paid 3 times more than the fee that was really necessary. To be truly cautious in larger value trades, it is recommended to wait for 6 confirmations. This ensures that there can't be any possible block reorganizations that put your transaction back into the mempool.



We can verify that the amount entered the new wallet. Hooray!



4. P2P Network

4.1 Bitcoin Nodes

Bitcoin operates on a peer-to-peer (P2P) network of computers. Any computer connected to the network is called a node. Anyone can download and install the open-source Bitcoin software to become a node. All nodes are treated equally, and no node is trusted by default.

The system assumes that the majority of nodes (if mining nodes, then the hash power) will be honest, meaning they will not retransmit false or malformed transactions and blocks. However, in response to such behavior, a node may choose to discourage (flag its inappropriate behavior and perhaps disconnect in favor of new peers), disconnect, or ban the peer.

Due to the cryptographic architecture of the blockchain, Bitcoin software can verify whether received information over the network is intact. Each block in the blockchain contains a cryptographic hash of the previous block, creating a chain of interlinked blocks. This ensures that any alteration in a previous block would invalidate all subsequent blocks, making forgery impractical.

Full nodes are responsible for verifying and maintaining all ownership records according to consensus rules. Mining nodes also process transactions and add new blocks to the blockchain in exchange for a reward. Pruned nodes verify transactions and blocks but do not store a full copy of the blockchain. SPV (Simple Verification Nodes) rely on third-party full nodes for information about specific addresses and transactions on the network. Wallet software usually operates as SPV nodes.

Nodes are computers that participate in the Bitcoin network, running Bitcoin software. They perform several essential functions to keep the network secure, decentralized, and operating correctly. The primary roles of nodes include:

Transaction Verification: Nodes verify that transactions follow all Bitcoin protocol rules, such as cryptographic signature verification and ensuring bitcoins are not double-spent.

Transaction and Block Propagation: Nodes transmit (or retransmit) transactions and blocks to other nodes on the network, helping to disseminate information efficiently.

Blockchain Storage: Nodes store a full or partial copy of the blockchain, ensuring that historical data is maintained and accessible.

Consensus Participation: Nodes participate in the consensus process, helping validate new blocks and maintain the system's integrity.

There are different types of nodes on the Bitcoin network, each with its own functions and characteristics:

Full Nodes: A full node maintains a complete copy of the blockchain and verifies all transactions and blocks according to Bitcoin rules. They are the backbone of the network, ensuring that all rules are followed and the blockchain is consistent across the network. They offer the highest level of security and privacy as they do not rely on third parties to verify transactions. They require more resources (storage space, bandwidth, and processing power) to operate efficiently. Full nodes ensure that the Bitcoin network remains secure and decentralized. The more full nodes there are, the harder it becomes for any malicious entity to compromise the network.

Mining Nodes: Besides verifying transactions and blocks, they also participate in the mining process, adding new blocks to the blockchain and receiving rewards for it.

Pruned Nodes: They verify the entire transaction and block history but delete the verified blocks, thus not storing a complete copy of the blockchain, only keeping the most recent blocks

to save disk space. In the Bitcoin Core's `bitcoin.conf` file, you can determine how many blocks you want the system to keep on disk. The pruned node limits some RPC operations since you don't have the indexed copy of all transactions, but depending on your needs, it can be a great option.

SPV Nodes (Simplified Payment Verification): Also known as "light nodes," they do not store a full copy of the blockchain. Instead, they download only the block headers and verify transactions based on inclusion proofs provided by full nodes. These nodes are ideal for devices with limited resources, like smartphones, as they consume fewer resources and bandwidth. Most Bitcoin wallets use light nodes in their composition. They rely on full nodes to obtain these inclusion proofs and therefore need to trust these full nodes for the correct verification of transactions.

You can run a Bitcoin node using Bitcoin Core, which is the reference implementation. In the `bitcoin.conf` file, you can choose the characteristics of this node, whether it will be full, pruned, etc.

Bitcoin Network Map:
https://medium.com/@gloriazhao/map-of-the-bitcoin-network-c6f2619a76f3

This site shows some options for full-node projects: https://bitcoiner.guide/node/diy/

There are also ways to create these nodes in the cloud, built from scratch or through systems that facilitate this process, such as:

https://voltage.cloud/

https://clovyr.app/

4.2 Mempool

Bitcoin operates using two main network concepts for data transmission. One network is for the relay of pending transactions, called the mempool, and the other is for the relay of mined blocks with confirmed transactions, which is the blockchain itself.

The mempool is a space where pending (unconfirmed) transactions are stored by each node in the Bitcoin network. When a transaction is broadcast to the network, it first enters the mempool of the nodes until it is included in a mined block.

The mempool, short for "memory pool," is a crucial component of the Bitcoin system. It functions as a waiting area for unconfirmed transactions before they are included in a block and added to the blockchain. When a user makes a Bitcoin transaction, it is initially verified by nodes (computers participating in the Bitcoin network) and then sent to the mempool.

Here are some key points about the mempool:

Transaction Reception: All transactions that have not yet been confirmed by miners are stored in the mempool. Each node in the Bitcoin network has its own version of the mempool, which may vary slightly between nodes.

Initial Verification: Before a transaction is added to the mempool, it undergoes an initial verification to ensure it follows Bitcoin protocol rules (e.g., if the transaction inputs are valid and the transaction does not attempt to spend non-existent Bitcoins).

Transaction Priority: Transactions in the mempool are prioritized based on transaction fees. Transactions with higher fees are generally selected first by miners, as miners are incentivized to maximize their profits by including transactions with higher fees in the blocks they mine.

Space Limitation: Each node can set a size limit for its mempool. When the mempool reaches this limit, transactions with lower fees may be discarded to make room for newer transactions with higher fees.

Confirmation: Once a miner includes a transaction in a block and the block is added to the blockchain, the transaction is removed from the mempool. This process confirms the transaction, ensuring it cannot be reversed or duplicated.

The mempool is essential for the efficient functioning of the Bitcoin network, ensuring that transactions are verified and prioritized before being permanently recorded on the blockchain.

4.3 Mainnet

The mainnet is the main Bitcoin network where real transactions occur, and bitcoins have monetary value. Developers and companies need to use the mainnet for production operations and final product launches.

Transactions are confirmed by miners and permanently recorded on the blockchain, ensuring high security and immutability. Developers and users interact through full nodes, wallets, and third-party APIs to carry out and monitor real transactions.

Use Cases for Developers

Scenario 1: Launch of a New Bitcoin Wallet A fintech company has developed a new Bitcoin wallet with advanced security and privacy features. After extensive testing on Testnet and Signet, they are ready to launch the wallet to the public. The company launches the new wallet on the mainnet, allowing users to start making real transactions with real bitcoins. The marketing team promotes the wallet, and developers monitor user feedback to adapt and improve the user experience based on real interactions. The company provides customer support to resolve any issues that may arise during wallet usage on the mainnet.

Scenario 2: Implementation of a Bitcoin Payment Service An e-commerce platform decides to accept Bitcoin as a payment method and needs to integrate a secure and efficient Bitcoin payment service. The technical team integrates a Bitcoin payment service that allows customers

to pay for goods and services using real bitcoins. The platform starts processing real transactions on the mainnet, receiving payments from customers and verifying the confirmation of transactions on the blockchain. The company implements accounting and reporting systems to track Bitcoin payments, ensuring compliance with financial regulations.

Scenario 3: Implementation of an International Remittance System A financial services company launches an international remittance system that uses Bitcoin to transfer money between countries quickly and with low fees. The company processes remittances from customers, using Bitcoin to transfer funds from one country to another. Real bitcoins are used to carry out transactions on the mainnet. The platform automatically converts customers' fiat currencies into Bitcoin for transfer and then converts back to the recipient's fiat currency after the transaction. The company keeps detailed records of all remittance transactions, ensuring compliance with international money transfer regulations.

4.4 Regtest

Regtest (Regression Test) is a local test network that allows developers to create a fully controlled testing environment. It is not a public network and is controlled by the user. It is fully configurable by the user, allowing adjustments to mining difficulty and block time.

Ideal for initial development, regression testing, and fine-tuning functionalities without waiting for network confirmations. Coins in Regtest have no monetary value, facilitating safe and fast testing. Developers configure a Bitcoin node in Regtest mode and use RPC commands to create and manipulate blocks and transactions manually.

Scenario 1: Initial Development of a New Protocol or Functionality A team of developers is working on implementing a new protocol for Bitcoin, such as a new type of digital signature or a proposed update to the consensus system. The team can create blocks instantly in Regtest, allowing for rapid testing without the need to wait for block confirmations as would be necessary on Testnet or Mainnet.

Regtest allows developers to adjust the mining difficulty and block time, creating specific conditions to test the new functionality in different scenarios. As Regtest is a local network, there is no interference from transactions or blocks from other developers. This allows for more focused and controlled testing.

Allows for a rapid development cycle with immediate feedback.

Enables the simulation of different network conditions in a controlled manner.

Ideal for initial tests and fine-tuning new functionalities.

Scenario 2: Integration and Automation Testing A software company is developing a set of automated tools to monitor and manage Bitcoin transactions, including fraud detection and compliance audits.

The company can automate the creation and verification of transactions in Regtest, ensuring that their monitoring scripts function correctly in various situations. They can create specific transactions and blocks to test the responses of the monitoring system, verifying that it correctly detects anomalous or suspicious behavior.

Developers can simulate attacks or fraud attempts in Regtest without any risk, testing the effectiveness of their detection systems.

The network facilitates the creation of an automated testing environment for continuous integration.

Allows for detailed testing of monitoring and security systems, providing a safe environment for attack simulations and responses.

Scenario 3: Development of Custom Applications A startup is developing a custom micropayments application that uses Bitcoin transactions to facilitate payments between users. The startup can simulate thousands of transactions in a short period, testing the application's ability to handle a large volume of micropayments.

Allows for testing the user experience (UX) when making transactions, adjusting the confirmation time and mining difficulty to evaluate different usage scenarios.

Regtest facilitates an iterative development cycle, enabling the startup to make quick changes and immediately see the effects of those changes.

Scenario 4: Regression Testing A development team is implementing updates to an existing Bitcoin wallet software and needs to ensure that the new changes do not introduce bugs or failures. The team can create a set of automated tests that execute transactions, create blocks, and check the blockchain state to ensure everything works as expected after the updates.

Regtest provides a consistent testing environment where the same tests can be repeated multiple times, ensuring that new changes do not negatively impact existing functionalities.

Allows for the creation of specific conditions that may be difficult to reproduce on Testnet or Mainnet, such as network failures or specific attacks.

4.5 Testnet

Testnet (Test Network) is a public test network that mimics the behavior of the Mainnet. Bitcoins on Testnet have no monetary value, allowing for risk-free testing. Anyone can mine blocks and conduct transactions on Testnet. Testnet simulates the conditions of Mainnet, offering a realistic environment for development and testing. Coins on Testnet have no financial value, enabling secure tests. Developers use full nodes, wallets, and APIs specifying Testnet to simulate real operations without financial risk.

Scenario 1: Development of a Bitcoin Wallet with Multisig Transaction Support A fintech startup is developing a new Bitcoin wallet with multisig transaction support. They want to test the functionality of creating and spending a multisig transaction to ensure everything works as expected before launching the product to the public.

Developers can create multisig addresses on Testnet without the risk of losing real funds.

They can experiment with different M-of-N configurations (e.g., 2-of-3 or 3-of-5).

Using Testnet, the team can send and receive test transactions to and from the multisig addresses. Since Testnet bitcoins have no real value, there is no financial risk involved.

Testnet allows the team to simulate real network conditions, such as transaction propagation, variable transaction fees, and confirmation times. They can adjust transaction fees to see how this affects the confirmation speed.

The team can easily obtain Testnet bitcoins from faucets to fund their test transactions, allowing for extensive testing without real financial costs.

Scenario 2: Development of a New Wallet Feature A software company is developing a new feature for its Bitcoin wallet that allows the automatic conversion of fiat transactions to Bitcoin. Developers can test the automatic conversion functionality, verifying that exchange rates are correctly applied and that Bitcoin transactions are sent and received without issues.

Using Testnet, the team can adjust and test different levels of transaction fees to ensure the wallet automatically selects the most appropriate fee under varying network conditions. The company can simulate various real-use scenarios, such as high and low transaction frequencies, to ensure the new feature operates efficiently and stably.

Scenario 3: Implementation of SegWit Transaction Support A development team is implementing Segregated Witness (SegWit) transaction support in its Bitcoin payment application. Developers can create and send SegWit transactions on Testnet to ensure that the implementation is correct and that transactions are valid. The team can verify that their application is compatible with other wallets and services that support SegWit, ensuring interoperability.

Using Testnet, the team can measure the performance of SegWit transactions compared to non-SegWit transactions, analyzing aspects such as confirmation speed and block space usage.

4.6 Signet

Signet is a proposed test network for Bitcoin that uses digital signatures instead of Proof-of-Work (PoW) for block validation. This approach enhances predictability and stability, providing an optimal environment for developers to test features and protocols.

The existing test networks like Testnet and Regtest have limitations. Testnet is notoriously unreliable due to frequent large reorganizations and irregular block generation. Regtest allows any participant to create blocks freely, which undermines its suitability for long-term multi-party testing. Signet aims to provide a more predictable and controlled environment, addressing these issues.

Block Validation: Signet requires blocks to include a digital signature based on a specific challenge (scriptPubKey). This signature ensures that only authorized entities can create valid blocks.

Proof-of-Work Compatibility: While Signet uses signatures for block validation, it retains the PoW block headers, allowing compatibility with existing software that supports PoW.

Custom Networks: Users can create their own Signet networks by generating a key pair and a challenge scriptPubKey. This flexibility enables tailored testing environments for various protocols and features.

Signature-Based Validation: Each block includes a signature that must be verified to validate the block. This process involves creating virtual transactions that ensure the block meets the challenge requirements.

Simplified Mining: The signature process does not commit to the nonce value, allowing miners to generate PoW without repeatedly signing the block.

Genesis Block and Message Start: A standardized genesis block and message start ensure consistency across different Signet networks.

Compatibility: Signet is designed to work with existing Bitcoin software with minimal modifications. It can be integrated into current systems by adding network parameters without changing the fundamental block validation logic.

Protocol Testing: Signet is ideal for testing new protocols like Eltoo or Taproot, providing a stable environment for integration testing over extended periods.

Exchange Testing: Exchanges can use Signet to test their systems against realistic reorg scenarios, ensuring their software handles such events correctly.

Wallet Testing: Signet allows wallet developers to test their software in a controlled environment, verifying their reorg handling and other critical functionalities.

Centralized Control: The centralized nature of Signet enables easy execution of global tests, such as scheduled reorgs, making it more predictable than Testnet.

Community Usage: While anyone can create a custom Signet, having a default, trusted Signet network can foster a common testing ground, reducing fragmentation and enhancing the effectiveness of community testing efforts.

Signet offers a robust, predictable alternative to existing test networks, addressing their limitations and providing a flexible, controlled environment for extensive testing of Bitcoin features and protocols.

Use Cases for Controlled Testing on Signet

Scenario 1: Implementation of a Risk Management System for a Large Exchange

A large exchange is implementing a new risk management system involving complex transactions and smart contracts. The reliability and predictability of the test environment are crucial to ensure the system functions flawlessly. Signet allows the exchange to conduct tests with complex transactions and smart contracts in a controlled environment.

Transactions can include custom scripts or advanced features that need rigorous validation.

With blocks signed by trusted entities, the exchange can be confident that the testing environment will be stable and free of unexpected behaviors like spam attacks or sudden changes in mining difficulty.

Signet ensures that test transactions are confirmed predictably, allowing the development team to focus on validating functionalities without worrying about network failures.

It offers an environment where security audits and compliance checks can be reliably conducted, ensuring the risk management system meets necessary standards before launch.

Scenario 2: Security and Resilience Testing

A cybersecurity company is developing a new security solution for Bitcoin transactions and needs to conduct extensive tests to ensure its solution withstands various types of attacks. On Signet, the company can simulate attacks such as double spending, replay attacks, and others, knowing the environment is controlled and predictable.

The security solution can be tested against simulated attacks to ensure all defenses work correctly.

Signet allows detailed monitoring and logging of all activities, facilitating the analysis of how the security solution responds to different threats.

Scenario 3: Development of Advanced Smart Contracts

A blockchain company is developing advanced smart contracts for complex financial applications, such as decentralized loans and derivatives. The company can develop and test complex smart contracts on Signet, ensuring that all functions and conditions execute correctly.

Signet provides an environment where testing conditions are stable and predictable, allowing rigorous testing of contract logic and security.

Developers can validate that the smart contracts interact correctly with other parts of the system and that all transactions are recorded as expected.

4.7 Forks

A fork is a change or divergence in the software, blockchain, or network consensus. Forks can be categorized into two main types: hard forks and soft forks.

Forks refer to changes in the Bitcoin protocol that result in a split in the blockchain, leading to the creation of a new path or a new blockchain. These changes can be classified as hard forks or soft forks, depending on backward compatibility. Soft forks are backward compatible, whereas hard forks create a new type of cryptocurrency incompatible with the original Bitcoin.

Hard Forks

Hard forks are non-backward-compatible divergences in the Bitcoin protocol or block history. In a hard fork, nodes must adopt new consensus rules, which can be looser or different from the previous rules. Nodes that do not update their software will see blocks produced under the new rules as invalid. A hard fork is a permanent change in the consensus rules that old nodes cannot follow.

On August 1, 2017, while the Bitcoin network was about to activate Segregated Witness (SegWit), a part of the network followed an alternative scaling path that was not backward compatible, increasing the base block size without SegWit. This resulted in the creation of the forked cryptocurrency known as Bitcoin Cash (BCH).

Soft Forks

Soft forks are backward-compatible changes to the Bitcoin protocol. In a soft fork, nodes opt for tightening or restricting the consensus rules. Nodes that do not update will continue to receive new blocks and recognize them as valid as long as these blocks follow the old rules.

Examples of Soft Forks

Segregated Witness (SegWit): Introduced as a soft fork, SegWit changes how transaction data is stored, increasing transaction capacity without increasing block size.

Pay to Script Hash (P2SH): A soft fork that resulted in the implementation of multi-signature wallets on the Bitcoin network.

Soft fork updates can cause temporary splits in the blockchain, but enforcement by a majority of hash power ensures eventual convergence on the same transaction history.

Types of Soft Forks

Miner Activated Soft Fork (MASF): A soft fork activated by the miners' hash power. Miners signal their support for the change, and when a certain threshold of support is reached, the change is activated.

User Activated Soft Fork (UASF): A soft fork activated by users. Nodes that wish to enforce the new consensus rules opt to implement the update, regardless of miner support. A notable example is the activation of SegWit through a UASF.

Importance of UASF

The User Activated Soft Fork (UASF) is significant because it demonstrates the power of Bitcoin users to influence protocol changes independently of miner support.

On August 1, 2017, the UASF was used to activate SegWit on the Bitcoin network. Despite initial resistance from some miners, user pressure led to the successful implementation of SegWit, showing that user consensus can trigger important protocol changes.

Detecting Hard Forks or Wrong Networks

To ensure you are on the correct network and not on a hard fork or wrong network, there are several practices and checks that nodes perform:

Block Headers: Nodes check the chain of block headers to ensure they are on the longest and valid chain. The longest chain is the one with the most accumulated work (proof-of-work).

Checkpoints: Certain nodes use checkpoints, which are blocks at specific heights that are well known and accepted by the Bitcoin community. This helps ensure the node is on the correct chain.

Software Version: Using the latest version of Bitcoin Core software (or another reliable implementation) ensures the node follows the most up-to-date consensus rules.

Network Connections: Connecting to known and trusted nodes on the Bitcoin network helps ensure you are on the correct network. Many Bitcoin clients come with a list of pre-configured trusted nodes.

Suppose you set up a Bitcoin node using Bitcoin Core software. When the node connects to the network, it:

Begins downloading and verifying the blockchain from the genesis block.

Validates each block and transaction according to the consensus rules.

Connects to other nodes and receives block headers to ensure it is on the longest chain.

Uses checkpoints to ensure it is not following an incorrect fork.

The Bitcoin system is designed to operate in a decentralized and secure manner, assuming that most nodes will be honest due to economic incentives and mutual verification of consensus rules. Although no individual node is trusted by itself, the network as a whole maintains its integrity through continuous verification and decentralization.

To ensure you are using a legitimate and secure version of Bitcoin Core software, download it only from the official repository on GitHub.

5. Mining

5.1 Energy: Utilization and Infrastructure Development

Bitcoin mining can utilize surplus energy that would otherwise be wasted, such as energy generated from renewable sources during periods of low demand.

This creates incentives for the development of electricity infrastructure in underdeveloped areas, as the presence of mining operations can make energy generation projects economically viable.

Additionally, the constant demand for stable and cheap electricity for mining can stimulate investments in renewable energy technologies and energy efficiency, benefiting local communities and promoting more sustainable use of energy resources.

There are some notable electrical infrastructure projects in African countries that involve Bitcoin mining. Here are a few examples:

Gridless Compute

Located in Kenya and other regions of East Africa, Gridless Compute uses hydroelectric microgrids to power Bitcoin mining operations in rural communities. This model leverages surplus energy that would otherwise be wasted, ensuring a constant and reliable demand for electricity, which in turn helps fund the expansion of electrical infrastructure in these areas. This project received funding from Jack Dorsey's Block and the venture capital firm Stillmark (CoinDesk, Finbold).

Virunga National Park

Located in the Democratic Republic of the Congo, in Virunga National Park, Bitcoin mining is powered by hydroelectric plants, generating crucial income for the biological reserve and local communities. The heat generated by the mining equipment is also being used to dry cocoa

beans, providing an economical and sustainable solution for local industrial processes (Bitcoin Magazine).

These projects demonstrate how Bitcoin mining can be integrated with renewable energy infrastructure initiatives to promote economic development and sustainability in underdeveloped regions.

5.2 Proof-of-Work

Mining is the process of adding new blocks to the blockchain and issuing new units of bitcoin.

The integrity of transactions and blocks is guaranteed by the contribution of computational power, a process called proof-of-work (PoW). The data of the candidate block is hashed repeatedly until the hash value is smaller than a standard value determined by the current difficulty. A hash value is considered 'smaller' if it has more leading zeros than the number that determines the round's difficulty.

While it may seem complicated, don't worry, as the process is automated. Setting up and maintaining the mining machines is often the only manual work required. Mining machines are specially designed to perform hash calculations efficiently, using specialized hardware such as ASICs (Application-Specific Integrated Circuits).

Proof-of-Work (PoW) is a protocol that requires a significant computational effort to be performed but is easy to verify. In the context of Bitcoin, PoW is used to ensure that all participants agree on the current state of the blockchain.

In this process, miners compete to find a valid hash that meets the difficulty criterion.

Bitcoin mining involves several continuous and cyclical steps performed by miners to add new blocks to the blockchain. Below, the process is logically detailed:

Collecting and Pooling Transactions from the Mempool Miners collect unconfirmed transactions from the mempool, which is a sort of 'waiting room' for transactions that have not yet been included in a block.

Organizing into a Candidate Block These transactions broadcast on the peer-to-peer network are organized into a candidate block. Miners can arbitrarily choose which transactions to include, usually opting for those with the highest fee per byte, generating greater profitability per mined block. They also verify that all transactions in the block are valid, ensuring there are no duplicates or invalid transactions.

Selecting the Previous Block The most recent block in the longest chain of the blockchain is selected, and its header hash is inserted into the new block.

This candidate block includes:

Hash of the Previous Block: This hash points to the previous block in the blockchain, ensuring that blocks are linked in an immutable sequence, maintaining the timechain data structure (the original name for the blockchain).

Merkle Root: A hash tree structure that represents all transactions in the block. The Merkle root is placed in the block header.

Timestamp: A timestamp that records when the block was mined.

Nonce: A number that miners change to find a valid hash that meets the network's conditions (i.e., starts with a certain number of zeros). Each different nonce generates a different hash that is compared to the difficulty level until the resulting hash is below it—indicating that the block has the necessary attributes to be part of the blockchain.

The block header contains:
Hash of the previous block.
Merkle root of transactions.
Timestamp.
Nonce.
Bitcoin software version.
Difficulty target, which determines the difficulty of the hashing problem.

Proof of Work

Miners repeatedly adjust the nonce and rehash the block header until they find a hash below the difficulty target set by the network. This process involves:

a. Incrementing the Nonce
Incrementing (adding 1 to) an arbitrary number in the block header called the nonce.

b. Calculating the Hash
Calculating the hash of the resulting block header.

c. Verifying the Hash: Verifying if the hash of the block header, when expressed as a number, is less than a predetermined target value.

Miners repeat the steps of incrementing the nonce, calculating the hash, and verifying the hash millions of times per second until they find a valid hash. This fast and continuous cycle is essential for the security and integrity of the Bitcoin network.

Validation and Inclusion in the Blockchain

Once a valid hash is found, the block is broadcast to the network, where other nodes verify the validity of the transactions and the PoW solution. If the block is accepted, it is added to the blockchain.

The winning miner is rewarded with a predefined amount of bitcoins (block subsidy of the epoch) and the transaction fees included in the block. Remember that mining is a race, so the speed of propagating a valid block is also important to ensure it is effective.

Rejection of Invalid Blocks

If the hash of the block header is not less than the target value, the block is rejected by the network.

Mining performance is measured in hashes per second, currently calculated in gigahashes (GH/s) or terahashes (TH/s).

New Round

After that, all miners start working on finding the next block, incorporating the new hash of the previous block as their starting point.

5.3 Timechain

Although the term "timechain" did not become popular, it highlights Satoshi's vision of the importance of time in the structure of a blockchain. The term "timechain" appears in early comments in the Bitcoin source code and in some of Satoshi's communications, although "blockchain" has become the more widely adopted term.

Difficulty Adjustment

Since mining is a trial-and-error process performed by computers, the greater the computational power, the faster a valid hash is found, and vice versa. Difficulty is regularly adjusted to ensure new blocks are mined approximately every 10 minutes, thus maintaining the network's stability and security.

To ensure new blocks are mined roughly every 10 minutes, the Bitcoin network automatically adjusts the mining difficulty every 2016 blocks (approximately every two weeks).

If the average block time is greater than 10 minutes, the difficulty is decreased; if it is less, the difficulty is increased.

This adjustment considers the amount of computational power contributed, ensuring that mining remains efficient and balanced, regardless of the number of miners and the power of their equipment.

In June 2021, the Chinese government imposed severe restrictions on mining operations, forcing many miners to shut down their machines or relocate their operations. As a result, the

global hash rate dropped dramatically, leading to a difficulty adjustment of about -28% in early July 2021, the largest downward adjustment recorded to date.

5.4 Controlled Supply

The generation of new bitcoin units follows a deterministic issuance schedule, with a finite total supply of approximately 21 million bitcoins. This is one of the main consensus rules of the system, ensuring it serves as an alternative to current financial systems, which are opaque, inflationary, and involve arbitrary money printing.

You can find the piece of code with the `GetBlockSubsidy` function directly in the Bitcoin Core GitHub repository.

```
1918
1919  CAmount GetBlockSubsidy(int nHeight, const Consensus::Params& consensusParams)
1920  {
1921      int halvings = nHeight / consensusParams.nSubsidyHalvingInterval;
1922      // Force block reward to zero when right shift is undefined.
1923      if (halvings >= 64)
1924          return 0;
1925
1926      CAmount nSubsidy = 50 * COIN;
1927      // Subsidy is cut in half every 210,000 blocks which will occur approximately every 4 years.
1928      nSubsidy >>= halvings;
1929      return nSubsidy;
1930  }
1931
```

History of Bitcoin Halvings

Initial Reward, 2009: On January 3, 2009, Satoshi Nakamoto mined the genesis block, with a reward of 50 bitcoins per block.

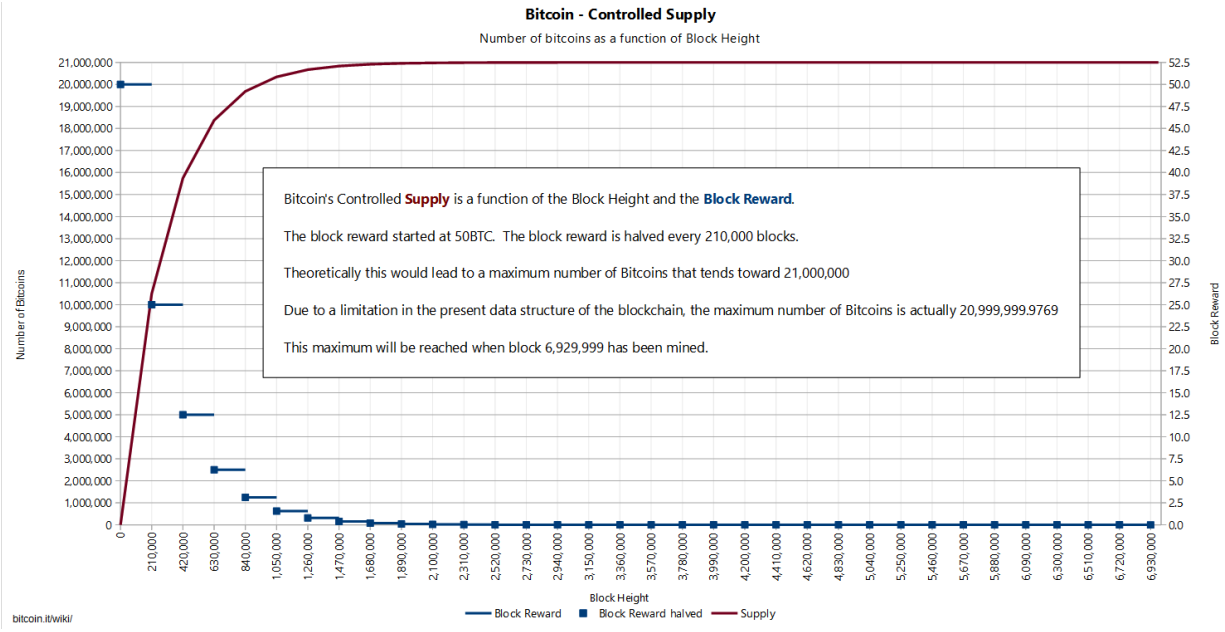
First Halving, 2012: On November 28, 2012, the first halving occurred, reducing the block reward from 50 to 25 bitcoins.

Second Halving, 2016: On July 9, 2016, the second halving occurred, reducing the block reward from 25 to 12.5 bitcoins.

Third Halving, 2020: On May 11, 2020, the third halving occurred, reducing the block reward from 12.5 to 6.25 bitcoins.

Fourth Halving, 2024: On April 19, 2024, the fourth halving occurred, reducing the block reward from 6.25 to 3.125 bitcoins.

More than 19 million bitcoins (over 90% of the total bitcoin supply) have already been issued through block rewards. The issuance rate will continue to decrease until the final issuance occurs in 2140.



5.5 Solo Mining and Bitcoin Resilience

Solo mining involves using personal computers or specialized hardware to search for blocks. Solo miners are paid only if they solve a block independently, making this practice highly competitive today and requiring significant investment.

However, the ability to conduct solo mining is a fundamental design decision that significantly contributes to Bitcoin's resilience and indestructibility. This decentralized approach allows anyone, anywhere in the world, to participate in the mining process using their own hardware. This brings several advantages:

Decentralization: Solo mining helps prevent the concentration of mining power in the hands of a few large players. Any individual can potentially find a block, which distributes mining power globally.

Security: With a highly distributed network of individual miners, it becomes extremely difficult for any entity to control the majority of the hashing power. This protects the network against 51% attacks, where an attacker tries to reverse transactions or prevent new transactions from being confirmed.

Resilience: The distributed nature of solo mining means the Bitcoin network can survive regional disruptions. Even if a large amount of hashing power is lost due to geopolitical issues or natural disasters, small individual miners around the world can continue operating and ensuring the network's continuity.

Inclusivity: Allowing anyone to participate in mining promotes a more inclusive ecosystem, where new miners can contribute to the network's security regardless of their location or access to resources.

Reduction of Censorship Risk: A decentralized mining network makes it harder for governments or other entities to censor transactions or miners, as there is no single point of failure or control.

Bitcoin Network Hash Rate Over Time

September 2019: 90,000,000 TH/s
September 2020: 140,000,000 TH/s
February 2021: 150,000,000 TH/s
August 2021: 129,000,000 TH/s (drop after mining crackdown in China)

For the average person, solo mining is not ideal due to high competitiveness and the significant investment required.

However, even though solo mining may not be economically viable for most individuals today, the possibility of performing it is crucial. This capability ensures that the network remains decentralized and accessible, avoiding excessive centralization and maintaining Bitcoin's robustness and security.

By allowing solo mining, Bitcoin's design strengthens its decentralized structure, making the network more secure, resilient, and resistant to attacks and censorship. This ensures that Bitcoin continues to operate robustly, even in adverse scenarios, contributing to its indestructibility.

5.6 Pool Mining

Pool mining is the primary method used today, where miners combine their hashing power to collectively solve the proof of work.

Operation

The pool server prepares a block with the coinbase transaction for the pool's address.

Miners make a `getwork/getblocktemplate` request to the pool server.

Each miner tries to solve the PoW problem, incrementing the nonce and calculating the block header's hash.

When a miner finds a hash below the difficulty target, they submit it to the server.

The server verifies and tracks the submitted shares.

Upon finding a solution, the server pays the reward proportionally to each miner's shares.

Miners periodically update the server about newly discovered blocks.

Reward Distribution

In pool mining, miners combine their hashing power to collectively solve the proof of work. Each miner receives a proportional share of the reward based on their hashing power contribution. If a miner contributed half of the pool's hashing power, they receive half of the reward.

Reward Distribution Schemes

PPS (Pay Per Share): Each miner is paid a guaranteed amount for each submitted share, with customized difficulties.

PPLNS (Pay Per Last Number of Shares): Payments are based on the last number x of shares after a block is found.

Proportional: Payments are based on the proportion of shares since the last block, counting all shares.

5.7 Mining Hardware

CPU Mining

Initially, Bitcoin mining was done using CPUs, but these were surpassed by more efficient hardware.

GPU Mining

GPUs are designed to perform many parallel calculations, being orders of magnitude faster than CPUs.

FPGA (Field Programmable Gate Arrays)

FPGAs were an intermediate step between CPUs and ASICs, used until ASICs dominated Bitcoin mining.

ASIC Mining

ASICs are built specifically for an application and are much faster than GPUs. They are customized to perform only the SHA-256 hashing, making them the only economically efficient mining technique today. See how they are: https://m.bitmain.com/

5.8 51% Attack

Hashing power is globally distributed among thousands of miners. To control 51% of the network, a malicious actor would need to acquire and operate an amount of hardware greater than the sum of what is currently distributed among all these miners. This operation would not go unnoticed by the community, as it violates consensus rules.

Possible Actions in a 51% Attack
Transaction Reversal: The attacker can reverse recently confirmed transactions, allowing double-spending of coins. This means the attacker can spend the same coins twice, once in a legitimate transaction and again in a fraudulent transaction.

Preventing New Transactions from Being Confirmed: The attacker can prevent new transactions from being confirmed, effectively freezing network activity. They can exclude or delay specific transactions or all transactions. This causes a denial of service (DoS) on the network, disrupting the blockchain's normal operation.

Monopolizing Mining: The attacker can control the mining of new blocks, refusing to include transactions from other miners and collecting all the block rewards. This centralizes mining, reduces the network's decentralization, and can lead to transaction manipulation.

Transaction Order Manipulation: The attacker can rearrange the order of transactions in blocks, altering the transaction sequence. This can be used to benefit certain transactions or miners, causing inequality and favoritism within the network.

Limitations of a 51% Attack: Despite the harmful actions possible, an attacker cannot do everything.

They cannot:
Create new coins out of nothing.
Reverse old transactions (beyond the most recent ones).
Modify consensus rules or the blockchain's fundamental properties.

Historical Case: In 2014, the mining pool GHash.io came close to controlling 51% of the network's hashing power, reaching about 42% at one point. This caused great concern in the community, leading to a voluntary decentralization of miners to reduce the risk.

Since then, the network has become even more distributed. Controlling the Bitcoin network's hashing power is not just about acquiring hardware and energy; it's also about the logistics of maintaining this massive operation on a global scale.

The cost, infrastructure need, and continuous community vigilance make this task extremely difficult and impractical for any isolated entity.

5.9 Stratum V2

Stratum V2 is an updated version of the Stratum mining protocol, widely used in Bitcoin mining. It was developed to address several limitations and issues present in Stratum V1, improving the efficiency, security, and flexibility of the mining process. Here are the main aspects and problems that Stratum V2 aims to solve:

Authentication and Encryption: Stratum V2 offers end-to-end authentication and encryption, protecting communications between the miner and the mining pool against man-in-the-middle attacks and eavesdropping. This is particularly important for protecting sensitive data and preventing attackers from altering or monitoring traffic.

Overhead Reduction: Stratum V2 is designed to reduce the amount of data transmitted between miners and the pool, improving communication efficiency. It uses a compact binary message format instead of a text format, saving bandwidth and speeding up data transmission.

Work Distribution: Unlike Stratum V1, where the mining pool completely controls the work distributed to miners, Stratum V2 allows miners to select transactions and build blocks more independently. This increases decentralization and gives miners more autonomy.

Transaction Prioritization: With Stratum V2, miners have more control over which transactions to include in blocks, allowing them to prioritize higher-fee transactions, which can increase profitability.

Failover and Automatic Reconnection: Stratum V2 includes improvements in connection management, such as automatic failover and quick reconnection, which increases the robustness and resilience of mining operations in case of network failures or disconnections.

Stratum V2 aims to solve security, efficiency, and centralization issues present in the previous protocol version, providing a more robust and adaptable solution for the future of mining. Stratum V2 is one of the open-source projects offering grants for developers in the Bitcoin ecosystem. Learn more in 11. Open-Source Development Careers.

6. Bitcoin Core

Since Satoshi Nakamoto launched Bitcoin v0.1 in January 2009 under the MIT free software license, hundreds of developers worldwide have contributed to the reference daemon and client, as well as various alternative clients. These developers, who collaborate on the main software development, are collectively known as "Bitcoin Core." Today, the reference implementation of the software can be found at https://github.com/bitcoin.

Discussions and new releases were frequently shared through the public mailing list 'bitcoin-dev,' which is now being discontinued in favor of new platforms like the Delving Bitcoin and https://groups.google.com/g/bitcoindev forum.

Most of these contributors work voluntarily, although some are sponsored by their employers or receive grants from industry companies. Recently, funding has become more diversified thanks to initiatives like GitHub Sponsors, Human Rights Foundation (HRF) Bitcoin Development Fund, Bitcoin Donation Portal, and Open Sats.

The project maintainers, currently led by Wladimir van der Laan, play a crucial role that includes balancing reactive tasks (community interactions) with proactive tasks such as writing and reviewing code, similar to a "janitor" role in other open-source software projects.

Additionally, there is a weekly code review club where important concepts and pull requests are discussed, promoting continuous collaboration and project improvement. It is the Bitcoin Core PR Review Club—https://bitcoincore.reviews/

7. BIPS—Bitcoin Improvement Proposals

Since 2011, changes to Bitcoin, in addition to maintenance tasks, have been introduced and organized through a process called Bitcoin Improvement Proposals, or BIPs. BIPs are a formal way to document, discuss, and implement changes and improvements to the Bitcoin protocol.

The idea of BIPs was inspired by the Python Enhancement Proposals (PEPs) process used to improve the Python programming language. The intention is to provide a clear and transparent framework for introducing changes to Bitcoin.

The first BIP (BIP 0001) was proposed by Amir Taaki and defines the process by which BIPs should be submitted, discussed, and approved.

Process of Creating, Evaluating, and Accepting a BIP

The idea for a new BIP is initially formulated and discussed organically in various communication channels, such as IRC, mailing lists, forums, and social media.

Once the idea is matured, the author writes a draft of the BIP. This draft is submitted for review and comments by the community.

The draft is submitted to the BIPs repository, where it undergoes a formal review process. During this phase, the author can make adjustments based on the feedback received.

The BIP is widely discussed and evaluated by the community and core developers. This stage may involve several cycles of review and modification.

Once the BIP is accepted in principle, a reference implementation is created. This implementation is a practical example of how the change should be incorporated into Bitcoin.

The reference implementation is tested by the community. If accepted and considered stable, the BIP is promoted to "Final" status.

Several BIPs have had significant impacts on the development and enhancement of Bitcoin. Here are some of the most important ones:

BIP 0016 (P2SH): Introduced by Gavin Andresen, this BIP implemented "Pay to Script Hash" (P2SH), allowing more complex transactions, such as multisig wallets.

BIP 0032 (Hierarchical Deterministic Wallets): Proposed by Pieter Wuille, this BIP introduced hierarchical deterministic wallets, facilitating the management and backup of wallets with multiple addresses.

BIP 0065 (CheckLockTimeVerify): Proposed by Peter Todd, this BIP added a new script function that allows locking a transaction until a specific time, enabling more advanced smart contracts.

BIP 0141 (Segregated Witness - SegWit): Introduced by Pieter Wuille, this BIP fixed transaction malleability and increased transaction capacity, allowing scalability solutions like the Lightning Network.

BIP 0144 (Compact Block Relay): Also part of the SegWit implementation, this BIP improved block propagation efficiency in the network.

BIP 0173 (Bech32 Address Format): Proposed by Pieter Wuille and Greg Maxwell, this BIP introduced a new address format, which is more efficient and less prone to typing errors.

Segregated Witness, or SegWit, is a notable example of a BIP that had a significant impact on Bitcoin. It was defined by BIPs 141 and 144 and introduced consensus changes to fix transaction malleability and improve scalability. After months of testing and discussions, SegWit was activated on the mainnet in 2017.

BIPs are fundamental to Bitcoin's evolution process, allowing a structured and collaborative approach to implementing changes. They ensure that any improvement or alteration to the protocol is well-documented, discussed, and tested before being adopted.

For Bitcoin developers and enthusiasts, understanding the BIP process and knowing the most important ones is essential for actively participating in the community and Bitcoin's ongoing development. You can find all Bitcoin BIPs by accessing the official Bitcoin Core repository: https://github.com/bitcoin/bips

8. RPC API do Bitcoin Core

Bitcoin Core implements a JSON-RPC interface that can be accessed using the `bitcoin-cli` command-line tool. This interface allows you to interactively experiment with the functionalities available programmatically via API.

After installing Core and starting it with the command `bitcoind -daemon`, you can begin calling RPC (Remote Procedure Call) commands. It is also possible to access directly via HTTP using `curl`.

Basic RPC Commands

Bitcoin Core's RPC commands allow you to interactively explore the blockchain, verify transactions, and obtain detailed information about the network and wallet state. To see a list of available commands, use:

```
bitcoin-cli help
```

Here are some useful commands:

`bitcoin-cli getbestblockhash`: Returns the hash of the best block in the blockchain.

`bitcoin-cli getblock "blockhash"`: Retrieves information about a specific block given its hash.

`bitcoin-cli getblockchaininfo`: Displays detailed information about the state of the blockchain.

`bitcoin-cli getnetworkinfo`: Displays basic information about the Bitcoin network node status.

`bitcoin-cli getrawtransaction "txid"`: Returns a raw transaction in hexadecimal notation.

`bitcoin-cli decoderawtransaction "hex"`: Decodes a raw transaction from hexadecimal to JSON.

`bitcoin-cli walletpassphrase "passphrase" timeout`: Unlocks the wallet and keeps it unlocked for a specific period.

`bitcoin-cli walletpassphrasechange "oldpassphrase" "newpassphrase"`: Changes the wallet password

`bitcoin-cli walletprocesspsbt "psbt"`: Processes a Partially Signed Bitcoin Transaction (PSBT).

`bitcoin-cli getblockhash 1000`: Gets the block hash at height 1000.

`bitcoin-cli getblock "blockhash"`: Gets block details using its hash.

`bitcoin-cli getrawtransaction "txid"`: Gets and decodes a transaction using its ID.

`bitcoin-cli decoderawtransaction "hex"`: Decodes a raw transaction from hexadecimal to JSON.

`bitcoin-cli getnetworkinfo`: Gets information about the network node status.

Using these commands through the command line or programmatically via API allows Bitcoin developers and enthusiasts to interact directly with the network, validating and exploring data independently and securely. See more at https://developer.bitcoin.org/reference/rpc/

9. Lightning Network

You can find the Mastering the Lightning Network textbook translated to pt-BR in this repository: https://github.com/biohazel/lnbook-pt-br.

The Lightning Network is a decentralized system for instant, high-volume micropayments that eliminates the risk of delegating custody of funds to trusted third parties. Bitcoin, the most widely used and valuable digital currency in the world, allows anyone to send value without an intermediary or trusted deposit. Bitcoin contains an advanced scripting system that allows users to program instructions for funds. However, there are some drawbacks to Bitcoin's decentralized design.

Confirmed transactions on the Bitcoin blockchain take up to an hour to become irreversible (approximately 6 confirmations, or 6 blocks deep). Micropayments, or payments smaller than a few cents, are inconsistent, and fees make these transactions unfeasible on the network currently.

The Lightning Network solves several scalability and cost issues of Bitcoin. It is one of the first implementations of a multiparty smart contract (programmable money) using Bitcoin's built-in script. The Lightning Network is leading technological development in multiparty financial transactions with Bitcoin, enabling fast, low-cost, and off-chain transactions.

Instant Payments: Bitcoin aggregates transactions in blocks spaced ten minutes apart. Payments are widely considered secure on Bitcoin after confirmation of six blocks, or about an hour. In the Lightning Network, payments do not need block confirmations and are instant and

atomic. Lightning can be used at retail point-of-sale terminals, with device-to-device transactions, or anywhere instant payments are needed.

Micropayments: New markets can be opened with the possibility of micropayments. Lightning allows sending funds as small as 0.00000001 bitcoin without custody risk. The Bitcoin blockchain currently imposes a minimum output size many hundreds of times larger and a fixed fee per transaction, making micropayments impractical. Lightning allows minimum payments denominated in bitcoin, using real bitcoin transactions.

Scalability: The Bitcoin network will need to support orders of magnitude higher transaction volume to meet the demand for automated payments. The imminent increase in internet-connected devices needs a platform for machine-to-machine payments and automated micropayment services. Lightning Network transactions are conducted off the blockchain without delegating trust and ownership, allowing users to conduct nearly unlimited transactions between other devices.

How It Works: Funds are placed in a bitcoin "channel" address with multiple signatures from two parties. This channel is represented as an entry in Bitcoin's public ledger. To spend funds from the channel, both parties must agree on the new balance. The current balance is stored as the most recent transaction signed by both parties, spending from the channel address. To make a payment, both parties sign a new outgoing transaction spending from the channel address. All old outgoing transactions are invalidated by doing this.

The Lightning Network does not require counterparty cooperation to exit the channel.

Both parties have the option to unilaterally close the channel, ending their relationship. Since all parties have multiple channels with different users on this network, it is possible to send a payment to any other party through this network. By embedding the payment conditioned on the knowledge of a secure cryptographic hash, payments can be made through a network of channels without any party having unilateral ownership of the funds.

The Lightning Network enables what was previously impossible with trusted financial systems vulnerable to – without the need for trust and custody, network participation can be dynamic and open to all.

Scalability is one of the main challenges faced by the Bitcoin network. This term refers to Bitcoin's ability to process a large number of transactions quickly and efficiently. Here are some specific challenges:

Block Size Limit: Each block on the Bitcoin blockchain has a maximum size of 1 megabyte. This limits the number of transactions that can be included in each block, resulting in a maximum capacity of approximately 7 transactions per second. This is relatively small compared to traditional payment systems, such as credit card networks, which can process thousands of transactions per second.

Confirmation Time: The average time to mine a block is about 10 minutes. During periods of high demand, transactions can take a long time to be confirmed, causing significant delays.

Transaction Fees: When the network is congested, transaction fees increase as users compete to have their transactions included in the next blocks. This can make using Bitcoin expensive, especially for lower-value transactions.

The Lightning Network is a layer 2 solution designed to address Bitcoin's scalability issues. It allows fast, low-cost transactions by moving them off the main blockchain (off-chain). Here's how the Lightning Network works and how it addresses scalability challenges.

Payment Channels: The Lightning Network uses peer-to-peer payment channels. Two users can open a payment channel by establishing an initial transaction on the Bitcoin blockchain. Once the channel is open, they can perform an unlimited number of transactions between themselves without having to record each individual transaction on the blockchain.

Off-Chain Transactions: Transactions made within a payment channel are recorded only by the two channel participants, not burdening the Bitcoin blockchain. Only the opening and closing of the channel are recorded on-chain, significantly reducing the amount of data processed on the blockchain.

Low Fees and High Speed: Since transactions on the Lightning Network do not require confirmation from miners, they are almost instantaneous and have much lower fees. This makes it feasible to use Bitcoin for small everyday transactions, such as buying a coffee or paying for an online service.

Network of Channels: The Lightning Network consists of a network of interconnected payment channels. Even if two users do not have a direct channel between them, they can send transactions through multiple intermediary channels, as long as there is an available path. This significantly extends the utility of the Lightning Network.

Security: Despite operating off-chain, the Lightning Network still benefits from the security of the main Bitcoin blockchain. If there is any attempt at fraud or disagreement between participants, Lightning Network rules allow users to revert to the main blockchain to resolve disputes.

The Lightning Network solves Bitcoin's scalability issues by moving most transactions off the main blockchain, allowing for fast, cheap, and secure transactions. This increases the capacity of the Bitcoin network and makes the system more efficient and scalable for broader use.

9.1 BOLTs—Basics of Lightning Technology

The Basics of Lightning Technology (BOLT) are the standardized technical specifications for the Lightning Network, defining how various implementations can interoperate within the same network. The official repository is at: https://github.com/lightning/bolts

The process of creating, evaluating, developing, and testing a BOLT is collaborative and rigorous, involving multiple stages of review, testing, and refinements. This process ensures that proposed improvements to the Lightning Network are technically sound, secure, and beneficial to the community as a whole.

The submission and approval process for a BOLT is analogous to the process for Bitcoin Improvement Proposals (BIPs). Learn about the BOLTs currently implemented.

BOLT 1: Basics of Lightning Technology (BOLTs) Description: This BOLT provides an overview of the Lightning Network, explaining the fundamental concepts and objectives. It defines the structure and purpose of the BOLTs, which serve as standardized documents for protocol specifications. Example Application: A new developer to the Lightning Network can consult this BOLT to understand the general architecture and purpose, guiding their initial development efforts.

BOLT 2: Peer Protocol for Channel Management Description: Specifies the protocol for managing channels between peers, including establishment, closure, and message exchange. Example Application: Implementation of a Lightning node that can open and close payment channels with peers, ensuring compliance with the network protocol.

BOLT 3: Bitcoin Transaction and Script Formats Description: Details the transaction and script formats used in the Lightning Network, including funding transaction details and the structure of commitment transactions. Example Application: Creation of custom transactions or scripts for advanced features, such as custom lock times or conditional payments.

BOLT 4: Onion Routing Protocol Description: Describes the onion routing protocol used for private and secure multi-hop payments. Includes details on packet structure, encryption, and routing process. Example Application: Development of secure payment routing logic in a Lightning wallet or service, ensuring user payments are routed confidentially through the network.

BOLT 5: On-Chain Handling of the Lightning Network Description: Provides guidelines for handling on-chain transactions (cooperative or forced channel closure), including channel monitoring and breach remedy transactions. Example Application: Implementation of "watchtower" services that monitor the blockchain for potential fraud or breaches and take corrective actions.

BOLT 6: Commitment Transaction Specification Description: Defines the structure and rules for commitment transactions, which are the core of channel state updates. Example Application: Ensuring a Lightning node correctly handles commitment transactions to maintain channel integrity and prevent losses.

BOLT 7: P2P Node and Channel Discovery Description: Describes the protocols for node and channel discovery, allowing nodes to find peers and learn about the network topology. Example Application: Enhancing a Lightning node's ability to dynamically discover and connect to new peers, improving network connectivity and robustness.

BOLT 8: Encrypted and Authenticated Transport Description: Specifies the encrypted and authenticated transport layer protocol for secure communication between nodes. Example Application: Ensuring secure communication channels in a Lightning node implementation, guaranteeing all messages are encrypted and authenticated.

BOLT 9: Assigned Feature Flags Description: Lists the feature flags used in the Lightning Network to signal optional protocol features a node supports. Example Application: Development of Lightning applications that can negotiate supported features with peers, ensuring compatibility and rich interaction.

BOLT 10: DNS Bootstrap and Assisted Node Location Description: Defines the use of DNS for initial node connection and assisted node location. Example Application: Implementation of initial connection logic in a Lightning wallet using DNS to find and connect to the Lightning Network.

BOLT 11: Invoice Protocol for Lightning Payments Description: Specifies the format for Lightning payment invoices, including details like amount, payment hash, and expiration time. Example Application: Generation and decoding of payment invoices in a Lightning payment processor, enabling users to make and receive payments easily. https://www.bolt11.org/

Although **BOLT 12** is still actively under development and has not yet been formally integrated into the main BOLTs repository, it offers several improvements over BOLT 11. These enhancements include support for static offers (which means that a single QR code can be generated and used repeatedly for payments. This contrasts with BOLT 11, where each payment required a unique QR code), recurring payments, blinded paths, and onion messaging. For more specific details and updates, it is recommended to follow the discussions on the lightning-dev mailing list and the relevant development repositories.

These BOLTs ensure the Lightning Network operates efficiently, securely, and interoperably, providing developers with the necessary standards to create robust and feature-rich applications.

The BOLTs (Basis of Lightning Technology) are collaboratively managed by the Lightning Network community developers. Developers of various Lightning Network clients, such as LND (Lightning Network Daemon), Éclair, and Core Lightning, contribute and ensure their software's compatibility with these standards.

Discussions and development proposals take place on the lightning-dev mailing list and social networks like Nostr and X.

9.2 Payment Channels

The Lightning Network is an innovative solution to the scalability and cost problems of on-chain Bitcoin transactions. It enables fast, low-cost transactions to be conducted off-chain through payment channels. These channels are built on 2-of-2 multisig addresses, timelocks, and

Segregated Witness transaction outputs, providing a secure and efficient way to move bitcoins between users without the need to record each transaction on the main blockchain.

To better understand how to manage and optimize the capacity of these channels, it's important to know the various options available, including opening new channels, splicing techniques, channel rebalancing, submarine swaps, channel factories, and dual-funded channels. Additionally, third-party liquidity providers can be used to add funds to your channels, increasing the flexibility and efficiency of your transactions on the Lightning Network.

To increase the capacity of a channel on the Lightning Network, you have several options, including splicing. Here are the main options:

Open a New Channel

Open a new channel with the additional amount of bitcoins you want to have available. This can be done independently or with the same channel partner, increasing the total capacity available for your transactions on the Lightning Network.

Splicing

Splicing is a technique that allows adding or removing funds from an existing channel without closing it. There are two main types of splicing:

Splice-In: Add funds to the existing channel. This involves an on-chain transaction that adds more bitcoin to the channel, increasing its capacity.

Splice-Out: Remove funds from the channel, allowing you to withdraw part of the channel balance without completely closing the channel.

Channel Rebalancing

Rebalancing the channel means moving funds between your existing channels to optimize liquidity without needing to close or open new channels. This can be done by sending payments to yourself through the Lightning Network to redistribute the balance between channels.

Loop In/Loop Out (Submarine Swaps)

Using submarine swaps, you can move funds between the Lightning Network and the main blockchain without closing channels. There are two types:

Loop In: Move funds from the main blockchain to a Lightning channel.

Loop Out: Move funds from a Lightning channel to the main blockchain.

Channel Factories

Channel Factories allow the creation of multiple payment channels between several parties using a single initial on-chain transaction. This can increase the efficiency and overall capacity of the channels.

Dual-Funded Channels

Dual-funded channels allow both parties to contribute funds when opening the channel, increasing the initial capacity without requiring one party to fund the entire channel.

Third-Party Liquidity Providers

You can use third-party liquidity providers who can add funds to your channels in exchange for a fee. This is particularly useful for users who need additional capacity without wanting to open new channels directly.

These options provide flexibility to manage and optimize the capacity of your channels on the Lightning Network, ensuring you have the necessary liquidity to conduct efficient and fast transactions.

A payment channel is simply a 2-of-2 multisig address on Bitcoin, for which you hold one key, and your channel partner holds the other key. To open a channel on Lightning, the channel effectively gives you access to the entire network. You first need to send on-chain bitcoin to your channel opening address.

The value of the on-chain transaction you make will be your spending credit. To make a payment larger than this first on-chain transaction, you would need to open another channel or increase the liquidity of your channel.

To reinforce, payment channels are built on 2-of-2 multisig addresses, timelocks, and Segregated Witness transaction outputs.

Channel Opening: A channel is opened after the multisig address receives an initial on-chain funding transaction.

Off-Chain Payments: The parties in the channel can make off-chain payments between themselves, updating the channel balance over time as much as they want.

Channel Closing: Either participant can decide to close the channel, cooperatively or non-cooperatively, at any time.

Balance Settlement: When the channel is closed, the balance will be settled through an on-chain transaction.

Hypothetical Example with Three Parties (A, B, C) in Routing

Initial Scenario:

Participants: A, B, and C. Payment Channels:

A has a payment channel with B.

B has a payment channel with C.

Payment from A to C: A updates its balance with B:

A wants to send 1 BTC to C.

A and B update their payment channel: A decreases its balance by 1 BTC and B increases its balance by 1 BTC.

This process is carried out through a Hash Time-Locked Contract (HTLC), ensuring the transaction is only completed if all participants agree.

B updates its balance with C:

B, now with 1 BTC more, sends this 1 BTC to C.

B and C update their payment channel: B decreases its balance by 1 BTC and C increases its balance by 1 BTC. Again, this is done using an HTLC for security and synchronization.

Payment from C to A:

C pays B

If C wants to send 1 BTC to A, the process is similar, but in the opposite direction.

C and B update their payment channel: C decreases its balance by 1 BTC and B increases its balance by 1 BTC, using an HTLC.

B pays A:

B, now with 1 BTC more, sends this 1 BTC to A.

B and A update their payment channel: B decreases its balance by 1 BTC and A increases its balance by 1 BTC, also using an HTLC.

How HTLC (Hash Time-Locked Contract) Works

HTLC is a type of smart contract that ensures the security of transactions on the Lightning Network. It allows a payment to be made only if a certain condition is met, such as presenting a

secret (hash preimage) or meeting a time limit. This prevents funds from being lost or stolen during payment routing.

This structure allows fast and cheap transactions, using the security of the Bitcoin network but without the need to record each individual transaction on the blockchain, increasing the system's scalability and efficiency.

Learn more at https://lightning.network/

9.3 Lightning Nodes

Below are the different ways to interact with the Lightning Network and the main applications that provide these services. This variety of options allows developers and users to choose the approach that best suits their needs and preferences.

Full Node
Running a full Lightning Network node involves operating a full Bitcoin node and a Lightning node anchored to it. This offers maximum security and control but requires more hardware resources and technical knowledge. Tools like Ride The Lightning are extremely useful for managing full Lightning nodes. Below are some popular Lightning node implementations.

LND (Lightning Network Daemon): Developed by Lightning Labs, it is one of the most popular implementations for running a full node. It offers a wide range of functionalities and APIs for developers.

c-lightning: Developed by Blockstream, it is a lightweight and modular implementation of the Lightning protocol, offering flexibility for customizations.

Eclair: Developed by ACINQ, it is a robust Lightning Network implementation focused on full nodes, with support for various advanced features.

Light Node

Light nodes do not require running a full Bitcoin node. They rely on third-party full nodes to verify transactions and blocks, making them easier to set up and operate.

Custodial Wallets

Custodial wallets are managed by third parties, meaning the responsibility for funds is transferred to the company providing the service. They are easy to use and ideal for beginners.

Wallet of Satoshi: A custodial wallet that offers a user-friendly interface for Lightning Network transactions. https://www.walletofsatoshi.com/

Non-Custodial Wallets—100% User-Controlled

Non-custodial wallets allow users to maintain full control over their private keys and funds, offering greater security and privacy.

Phoenix Wallet: Developed by ACINQ, it is a mobile wallet that acts as a light node, allowing fast and secure transactions on the Lightning Network. When you create a wallet with Phoenix, it automatically opens a channel with ACINQ, the most well-connected and liquid node in the network. https://phoenix.acinq.co/

Zeus Wallet: Allows users to connect to their own LND, c-lightning, or Electrum nodes, offering full control over funds and transactions. It is empowering to be anywhere in the world and transact via a VPN connection directly from your full node at home. It is complete financial self-sovereignty. Learn more at: https://zeusln.com/

9.4 Network Explorers

Lightning Network explorers allow users to check the state of channels, transactions, and other relevant data publicly.

1ML: A network explorer and node directory for the Lightning Network, offering visualizations of channels, capacities, and node information.

Amboss: Another explorer that provides a user-friendly interface for exploring the Lightning network topology and detailed information about nodes and channels.

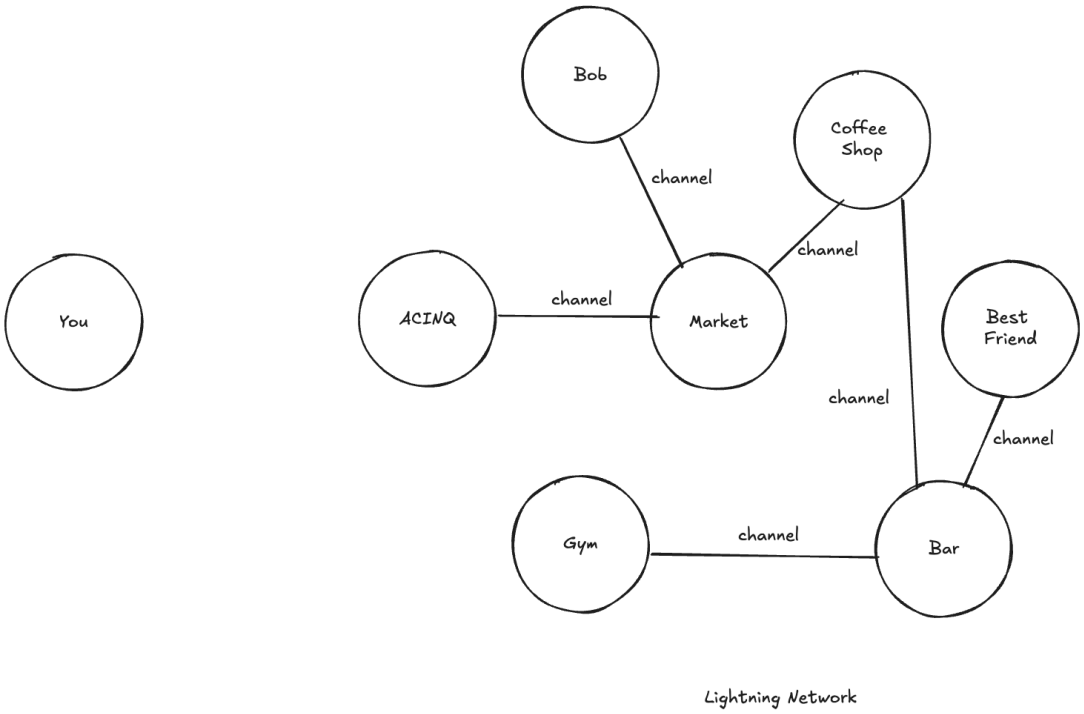
These various ways of interacting with the Lightning Network provide a range of options adaptable to different levels of technical knowledge and security requirements. With these tools, both developers and users can engage efficiently and securely with the Lightning Network.

9.5 Practice: My First Bitcoin on the Lightning Network

Time to experience a Bitcoin transaction via Lightning. When we transact Bitcoin on-chain, as we did previously, the term used is "transaction." In the case of the Lightning Network, the term used is "payment" because we essentially make payments for invoices. Through the keysend function, we can send arbitrary amounts to people, but we still call them payments.

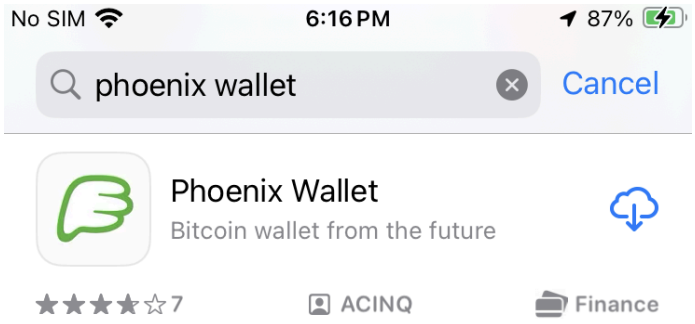
To interact with the Lightning Network, you need to open a first channel that will potentially give you access to all the nodes in the network, depending on how well-connected it is. The ACINQ node is one of the best-connected and also one of the largest in terms of liquidity (amount of bitcoins allocated).

The Phoenix wallet, developed by ACINQ, facilitates this channel-opening process with them, allowing you to enjoy instant and low-fee payments on the Lightning Network. Before having our first channel open, we cannot make or receive payments via Lightning. The situation is illustrated below.

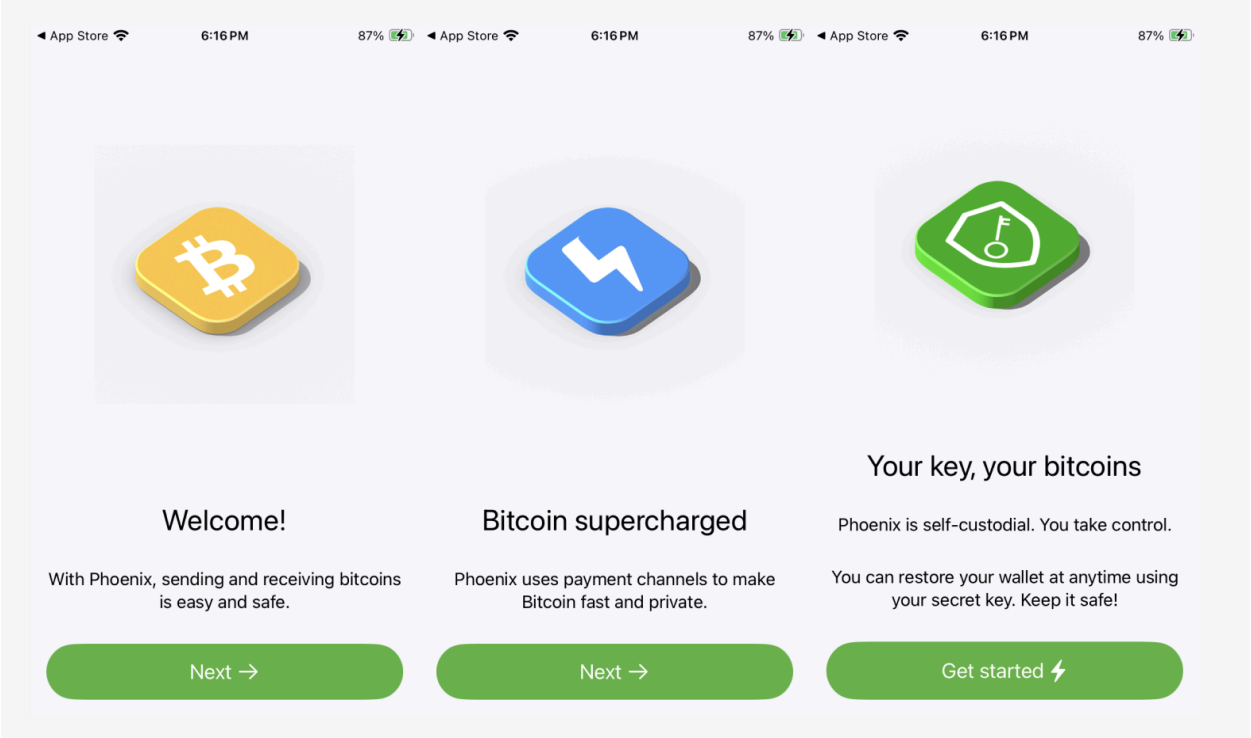


Lightning Network

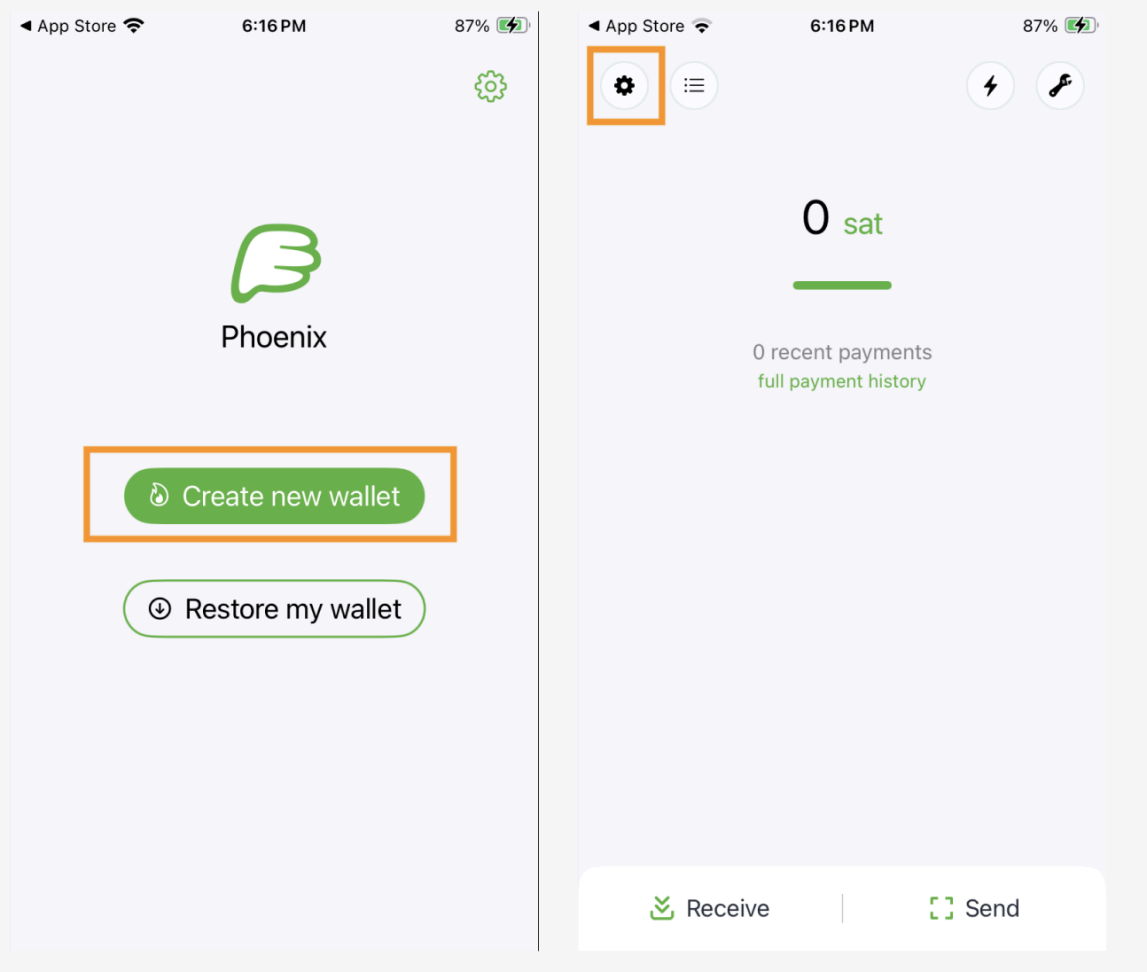
To make our first transaction on Lightning, we will use the Phoenix wallet.



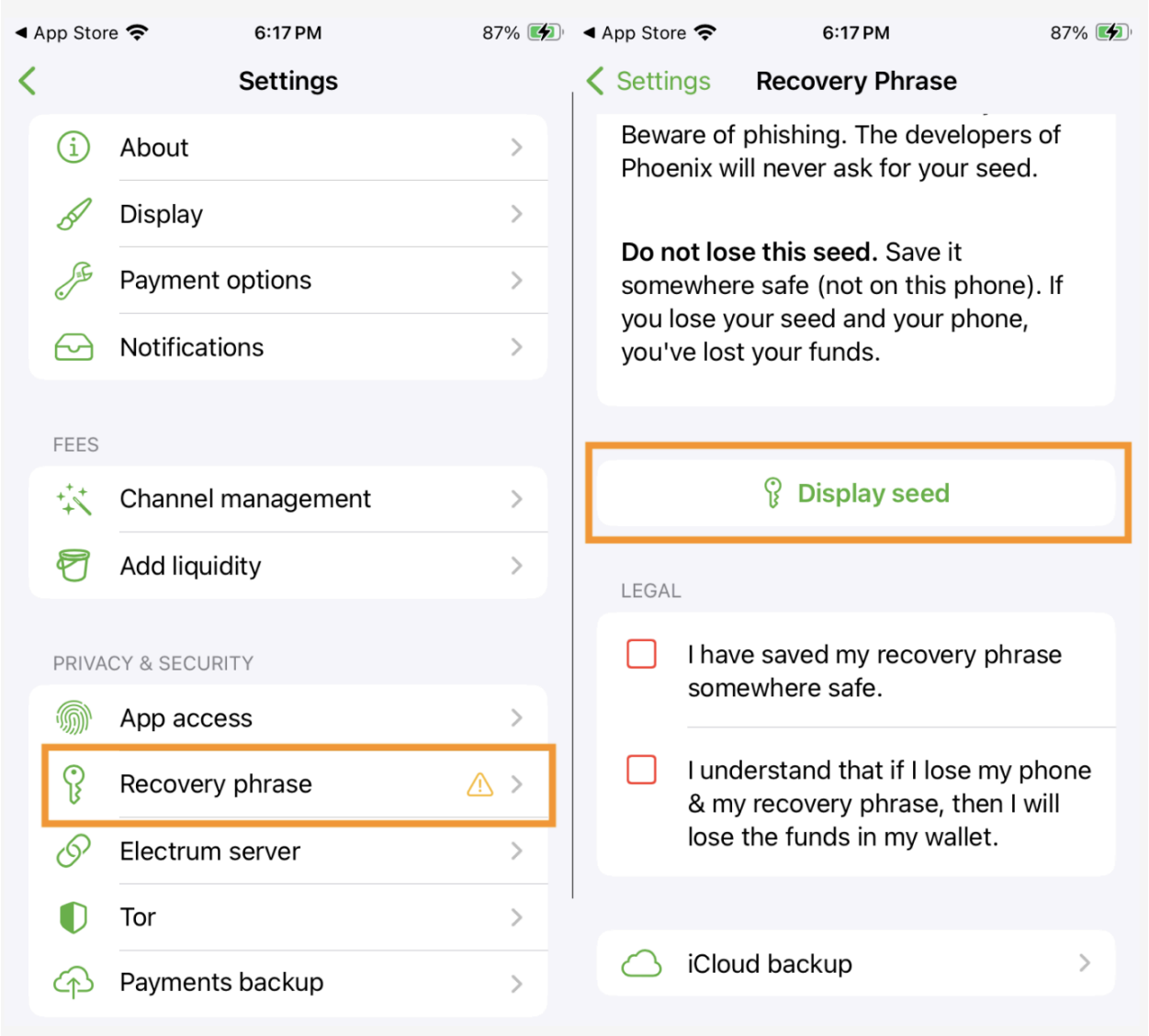
There will be 3 initial screens; click Next, Next, and Get started.



Now choose Create new wallet. Then click on the Settings icon to find and note down your wallet recovery mnemonic phrase.



Click on Recovery phrase and then on Display seed.



Write down your seed carefully.

Now, click the x, then Receive, and click on the chain icon.

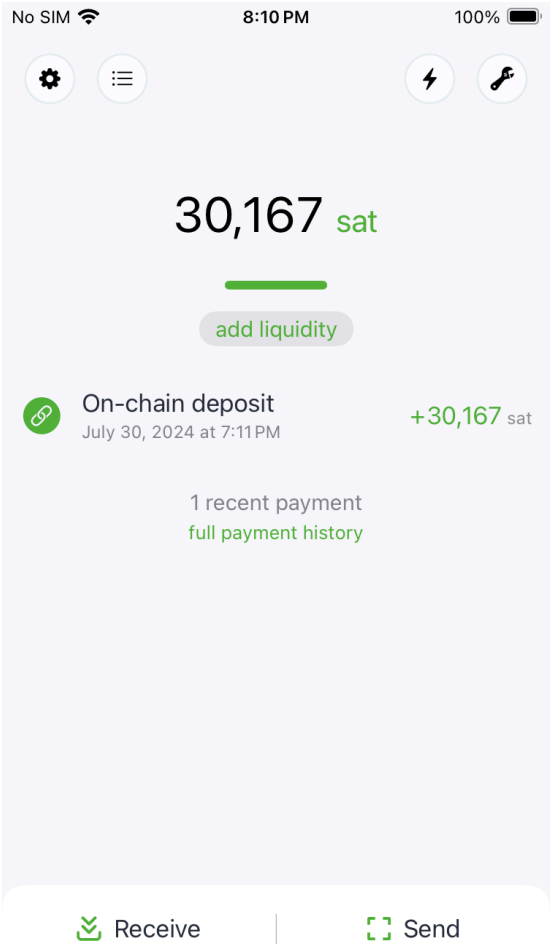
This address on the screen is an on-chain address for your wallet. Now that you know how to make on-chain transactions, you can open your BlueWallet and send an amount to this address. This amount will be the capacity of your channel on the Lightning Network.

Think of it as a prepaid account, an amount of bitcoin that is 'locked' in the channel but that you can use to send and receive bitcoin with very low fees and instant speed.

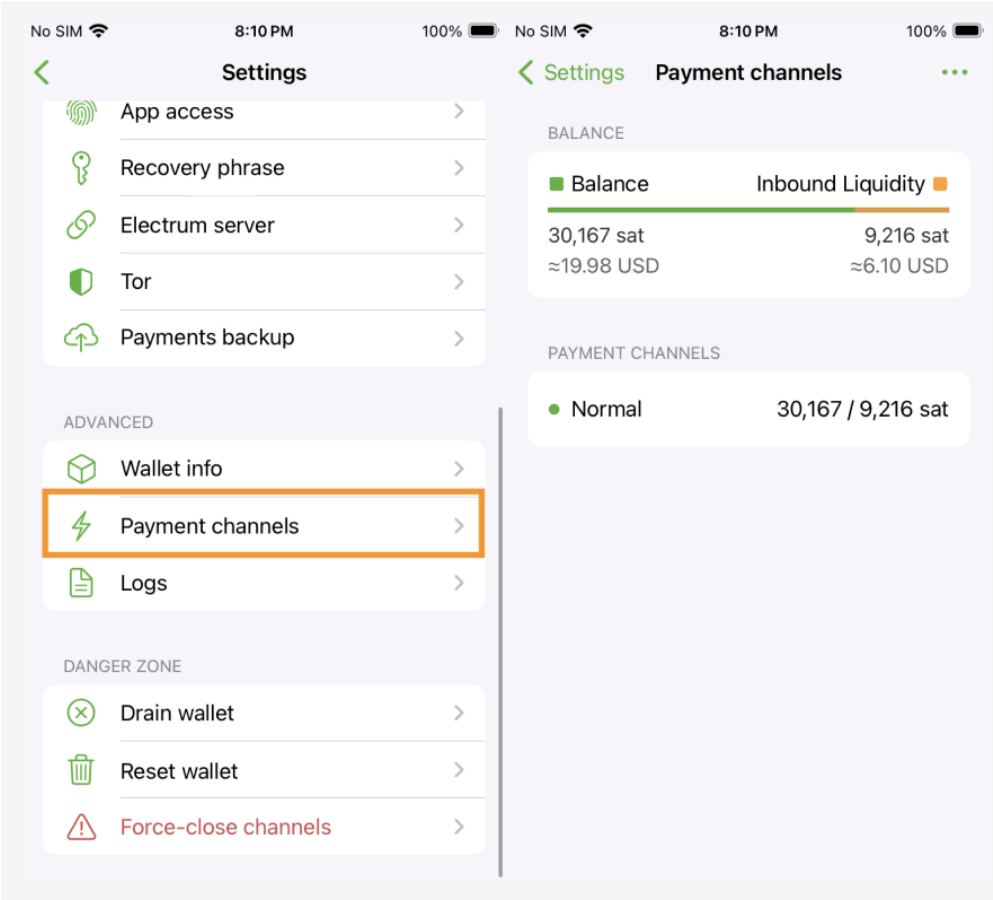
In a world where Bitcoin is a medium of exchange, you might open a channel on Lightning and never close it. Now that you know almost everything about Bitcoin, set up a circular economy in your community, and this reality will come sooner than you think.

Initially, when you open a channel on the Lightning Network, you only have outbound capacity, meaning you can send bitcoins but not receive them. There are tools that help balance the channels so that you can both send and receive. The Phoenix wallet manages this automatically.

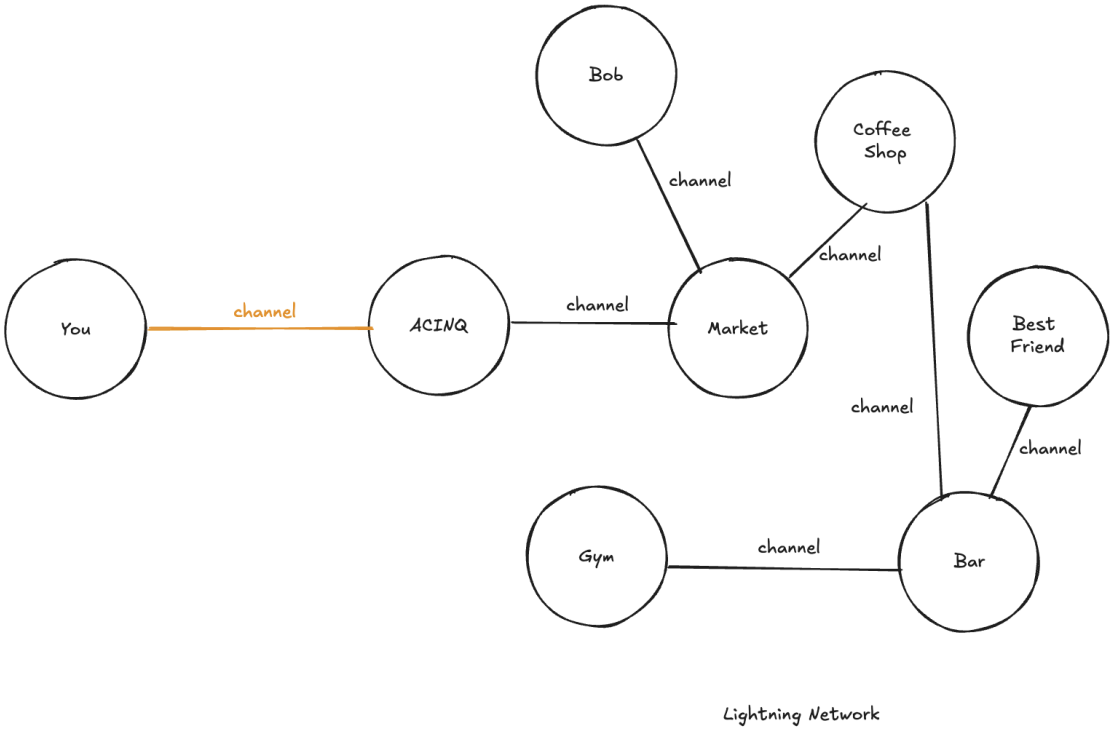
After sending the satoshis from your other wallet, you will see the amount on the main screen.



Again in Settings, look for Payment channels, and you will see your first channel open on Lightning.



Congratulations! You are now using the best L2 (Layer 2, off-chain) solution for Bitcoin. The procedure to pay or generate Lightning invoices is completely intuitive, through the Send and Receive buttons, with options for QR Code and written code. The important thing is that you are now operating via Lightning through Phoenix. Now, if someone asks you if they should pay you on-chain or via Lightning, you know the answer.



10. Freedom Technologies

10.1 Cypherpunks

Bitcoin was created as a response to the limitations of the traditional financial system and emerged from a quest for human freedom. This movement was strongly influenced by the cypherpunk community, a group of technological activists who advocate the use of cryptography to protect privacy and promote individual freedom.

Cypherpunks are known for their vigorous advocacy of using cryptography to ensure privacy and freedom of expression in the digital age. Formed in the 1990s, this community comprises programmers, cryptographers, and privacy advocates who believe cryptography is essential to maintaining personal freedom against government surveillance and control.

The philosophy of cypherpunks is rooted in the belief that privacy is fundamental to human freedom. They advocate for the creation and use of technologies that can protect personal communications and transactions from any form of surveillance or censorship.

The creation of Bitcoin by Satoshi Nakamoto in 2008 was heavily influenced by cypherpunk ideals. Nakamoto was involved in discussions about cryptography and digital currencies, and Bitcoin represents a practical implementation of the ideas championed by cypherpunks: a decentralized and anonymous financial system.

The cypherpunk movement began in the late 1980s and early 1990s. It was founded by individuals like Eric Hughes, Timothy C. May, and John Gilmore.

In 1993, Eric Hughes published the "Cypherpunk Manifesto," which outlined the movement's principles and goals. He emphasized the importance of privacy and the use of cryptography to achieve it.

Cypherpunks believe that privacy is essential for a free society and that it must be protected against intrusions. They use and promote strong cryptography to protect communications and personal data and advocate for decentralized systems that do not rely on central authorities, as these systems are less susceptible to censorship and control.

Technologies and Contributions of the Cypherpunk Community

PGP (Pretty Good Privacy): One of the earliest and most well-known encryption software, created by Phil Zimmermann, a member of the cypherpunk community. Developed in 1991, PGP allowed anyone to effectively protect their electronic communications. Zimmermann became a symbol of the fight for digital privacy, receiving numerous awards and recognitions for his contributions to information security.

Zimmermann's struggle, facing retaliation from the U.S. government in the 1990s, helped solidify the importance of digital privacy as a fundamental right. In 1993, he was the subject of a criminal investigation for alleged violations of arms export regulations since PGP was considered military-grade encryption technology. However, after intense public pressure and support campaigns, the case was dropped without formal charges in 1996.

Zimmermann's advocacy for strong encryption influenced policies and regulations worldwide, promoting the adoption of digital security technologies by both individuals and companies. His work continues to impact digital security today, establishing encryption as an essential tool for protecting privacy and freedom of expression in the digital age.

In the context of Bitcoin development, for example, it is good practice for developers to sign their commits with their PGP keys. Learn more at https://docs.github.com/en/authentication/managing-commit-signature-verification/signing-commits

Tor: The cypherpunk movement directly influenced the creators of Tor. The Tor network (The Onion Router) was originally developed by the U.S. Naval Research Laboratory in the mid-1990s. The project began with the need to protect the U.S. government's online communications. In 2002, Tor was released to the public to provide internet anonymity.

Bitcoin: Although Satoshi Nakamoto, the creator of Bitcoin, was not a declared member of the cypherpunk community, many of Bitcoin's principles and ideas were influenced by cypherpunk philosophy.

Lightning Network: Although the Lightning Network was not explicitly created by cypherpunks, it incorporates many of the fundamental values of cypherpunk philosophy, such as privacy, decentralization, and efficiency. These characteristics are reflected in the design decisions that allow for fast and private transactions while maintaining Bitcoin's security and decentralization. The LN continues to evolve as a crucial solution for scaling Bitcoin while adhering to its fundamental principles.

Many cypherpunks are activists fighting against mass surveillance and internet censorship, advocating for policies that protect digital privacy and freedom. The philosophy and technologies developed by the cypherpunk community have significantly influenced digital security and online privacy protection. They have played a crucial role in developing many of the digital security technologies we use today.

The influence of cypherpunks continues to be felt in the development of new privacy and security technologies. Projects that follow cypherpunk principles continue to emerge, always seeking to protect individual freedom in the digital world.

The open-source Bitcoin development ecosystem is tied to the philosophy of protecting privacy and human freedom.

Many applications integrated with Bitcoin follow this design logic, and it is especially gratifying to know that your talent is being used to build relevant tools to liberate humanity. Here are some examples of important projects in the ecosystem, many of which offer paid opportunities for developers.

10.2 JoinMarket e Jam

JoinMarket is a powerful tool that significantly enhances the privacy of Bitcoin transactions through a process called CoinJoin. Developed to facilitate the mixing of Bitcoin transactions, JoinMarket connects users who want to anonymize their transactions with those who provide liquidity for the process.

JoinMarket utilizes the concept of CoinJoin, where multiple transactions are combined into a single joint transaction, making it difficult to trace the funds. Here is a step-by-step guide on how JoinMarket works. There are two main types of users in JoinMarket:

Makers: Provide liquidity for CoinJoin by offering their funds to be mixed. In return, they receive a small fee.

Takers: Initiate the CoinJoin transaction and pay a fee to the Makers to mix their coins.

Users need to set up a JoinMarket wallet, which can be done via the command-line interface or the optional graphical interface. The wallet connects to the Bitcoin network to carry out transactions.

CoinJoin Process

1. **Makers** announce their liquidity offers on the JoinMarket network.
2. **Takers** select the Makers' offers and initiate a CoinJoin transaction.
3. The combined transaction is created, signed by all participants, and sent to the Bitcoin network.

Use the "daemon" mode to run JoinMarket in the background, facilitating continuous CoinJoin processes. Regularly participate in CoinJoins to increase the privacy of your transactions. Combine JoinMarket with other privacy practices, such as using new addresses for each transaction.

Installing JoinMarket

To install JoinMarket, you can clone the official GitHub repository and follow the installation instructions. Here are the detailed steps:

Clone the JoinMarket repository:

```
git clone https://github.com/JoinMarket-Org/joinmarket-clientserver.git
```

```
cd joinmarket-clientserver
```

Run the installation script:

```
./install.sh
```

Configuring the Wallet. Create the configuration file:

```
cp cfg/joinmarket.cfg.example cfg/joinmarket.cfg
```

Edit the `joinmarket.cfg` file as necessary, especially the section for configuring the connection with Bitcoin Core.

Start the JoinMarket daemon:

```
jmwalletd
```

Refer to the JoinMarket installation guide and user guide for more details.

Contributing to the Project

Familiarize yourself with the JoinMarket repository on GitHub, where you can find the source code and open pull requests to contribute improvements. Participate in community discussions to understand the needs and priorities of users. Work on user interface improvements to make JoinMarket more accessible to non-technical users. Contribute to the documentation, helping new users set up and use JoinMarket effectively.

Unlike other centralized mixers, JoinMarket is decentralized, eliminating the need to trust an intermediary service. Makers are economically incentivized to provide liquidity by receiving fees from Takers.

JoinMarket allows users to choose their own fees and offers, providing flexibility in using the service. It represents a crucial tool for those who value privacy in Bitcoin usage. With an active community of developers and users, the platform continues to evolve, offering effective methods to anonymize transactions and protect users' financial privacy.

JAM (Joinmarket-API-Interface)

JAM is an advanced, user-friendly interface for JoinMarket, designed to facilitate the use of CoinJoin and improve the user experience, especially for those who are not technically inclined.

JAM is a browser-based application that offers a graphical user interface (GUI) to interact with JoinMarket. Developed to be intuitive and accessible, it allows users to set up and manage their JoinMarket wallets, participate in CoinJoins, and monitor their transactions with ease.

JAM represents a significant advancement in the usability of JoinMarket, offering an accessible and user-friendly interface that makes participating in CoinJoin transactions easier and improves the privacy of Bitcoin users. Both users and developers are encouraged to explore and contribute to the project, helping to build a more robust and secure tool for everyone.

For more information and detailed instructions on how to get started, visit the official JAM repository on GitHub.

10.3 E-cash—Federated Chaumian Mints

Chaumian e-cash is a crucial technology for preserving privacy in digital transactions, conceived by cryptographer David Chaum. It enables the creation of digital coins that are anonymous and untraceable, utilizing advanced cryptographic techniques such as blind signatures. This

technology is of particular interest to developers looking to create private and secure payment systems.

While Bitcoin uses a different model to ensure transaction security and privacy, Chaum's concept of blind signatures has significantly contributed to the understanding of anonymous and decentralized transactions that led to Bitcoin. The combination of these two systems favors Bitcoin's scale as a financial system that truly preserves user privacy.

How Chaumian E-cash Works

Chaumian e-cash is a system that uses blind signatures to ensure that the entity issuing the money cannot link the user's identity to subsequent transactions. This is achieved through a process where digital tokens are blindly signed by the issuer, and only the token holder can spend them without revealing their identity.

It consists of two main components: the mint and the e-cash wallet. The technology allows anyone to operate a mint for their application, which can range from digital wallets to voucher and reward systems.

E-cash tokens issued by the mint have a 1:1 parity with Bitcoin. This means that for every e-cash token issued, there is an equivalent amount of Bitcoin held in reserve. Users can easily convert their e-cash tokens back into Bitcoin, ensuring liquidity and usability of funds.

Chaumian e-cash combines the privacy of blind signatures with the efficiency and scalability of the Lightning Network. This combination of privacy, security, and efficiency makes Chaumian e-cash a robust solution for anonymous financial transactions. Learn more about the components of an E-cash system.

Mint: The mint is responsible for issuing e-cash tokens. When a user deposits Bitcoin, the mint creates equivalent e-cash tokens using blind signatures. This process ensures that the mint cannot track the future transactions of these tokens. A mint does not store a database of user accounts and their activities, protecting users of an E-cash system from data leaks to hackers and providing stronger censorship resistance than traditional payment systems.

E-cash Wallet: Users keep their e-cash tokens in digital wallets, which can be used for transactions. These transactions respect user privacy as the tokens are spent anonymously.

E-cash transactions between users, or from users to service providers, are conducted in a way that protects the sender's and receiver's identities. Anyone can operate a mint for their application, whether it's a wallet, a web paywall, paid streaming services, or a voucher and rewards system.

Chaumian e-cash represents a crucial piece in building financial freedom technologies alongside Bitcoin. Projects like Cashu and Fedimint exemplify how this technology can be applied to create private and secure payment systems, offering ample opportunities for

developers willing to contribute to the advancement of the open-source Bitcoin ecosystem and financial freedom.

Cashu
Cashu is an e-cash protocol integrated with the Bitcoin protocol. It is a project that implements Chaumian e-cash to create a private and user-friendly Bitcoin transaction platform. Users can deposit Bitcoin and receive equivalent e-cash tokens that can be spent anonymously within the system. The main benefit of Cashu is user privacy protection, as transactions made with e-cash tokens cannot be traced.

Cashu is primarily focused on providing a simplified and private way to use Bitcoin, emphasizing ease of use and transaction privacy. It is designed to be lightweight and easy to implement, making it accessible for developers who want to integrate private Bitcoin transactions into their applications.

Developers interested in contributing to Cashu will find various open-source work opportunities. They can get involved in implementing privacy protocols, improving the efficiency of blind signatures, and creating user interfaces that facilitate the adoption of the technology by a broader audience.

Learn more at: https://docs.cashu.space/

Fedimint
Fedimint is another innovative project that uses Chaumian e-cash but with a federated approach to Bitcoin custody and management. In this system, a federation of trusted entities collaborates to issue and manage e-cash tokens.

The federation uses multi-signatures (multi-sig) to approve the issuance and redemption of tokens, distributing trust among multiple participants and mitigating security risks. The decentralization of the federation ensures no single point of failure in mint management.

For developers, Fedimint offers a fertile ground for contributing to decentralized security and financial privacy. Opportunities include implementing more robust multi-sig schemes, optimizing communication between federation members, and developing tools that facilitate the creation of federations in different communities.

Implementing and improving technologies like Chaumian e-cash in projects like Cashu and Fedimint open up numerous opportunities for developers. Contributing to these projects not only helps strengthen the privacy and security of financial transactions but also positions developers at the forefront of technological innovation in the Bitcoin space.

Participating in these projects offers the chance to work with cutting-edge technologies, collaborate with a global community of developers, and contribute to building fairer and more private financial systems.

Whether improving cryptographic algorithms, developing intuitive interfaces, or implementing decentralized custody solutions, the possibilities are vast and impactful.

Learn more at: https://fedimint.org/

10.4 Nostr

Tim Berners-Lee, a British computer scientist, invented the World Wide Web (WWW) in 1989 while working at CERN, the particle physics laboratory in Switzerland. Berners-Lee wanted to create a system that would facilitate the exchange of information between scientists at different universities and institutions around the world. In 1991, he launched the first web page and the first web server, making the web public.

Berners-Lee's original vision for the internet was a decentralized and uncensorable network where anyone could freely share and access information. He designed the web as an open space, based on universal and free standards, that would promote global communication and collaboration without barriers.

Over time, technology companies like Google, Facebook, Amazon, and others became dominant on the internet. These Big Tech companies and cloud service providers have centralized a large portion of internet traffic and data. While these companies have facilitated access to information and services, they have also introduced new challenges.

The major platforms have the power to censor content and control what can or cannot be shared. The massive collection of personal data for monetization purposes compromises user privacy. The reliance on centralized servers makes the network vulnerable to failures and attacks.

The Problem Nostr Solves
Nostr (Notes and Other Stuff Transmitted by Relays) is a protocol designed to overcome the issues of centralization and censorship on the internet. Unlike traditional social networks and platforms, Nostr was designed to be distributed and censorship-resistant.

Nostr is a simple asynchronous messaging protocol layer. The protocol operates in a decentralized and censorship-resistant manner to ensure freedom of expression. Therefore, it is not an application or service you sign up for; it is a protocol, an open standard that anyone can build upon.

The protocol is based on very simple and flexible event objects and uses standard elliptic curve cryptography for keys and signatures. Since Nostr does not rely on a few central servers to move or store data, it is very resilient and allows censorship-resistant communication between its clients.

Nostr is a decentralized protocol, meaning it is not controlled by any central authority. Instead, the network is just a collection of independently operated data relays (servers). A person can use their personal machine to serve as a relay, ensuring their posted messages are always available (at least from their home server) and volunteering to store and propagate events from other Nostr keys. This means that Nostr as a whole is not vulnerable to censorship or manipulation by any single entity.

Unlike many social media platforms, Nostr does not collect user data to sell to third-party advertisers. No email address, phone number, or government identity is associated with your Nostr account. Much like Bitcoin, the system only knows public keys and cryptographic signatures for authentication.

Nostr offers a unique monetization system for content creators. It is easily integratable with the Lightning Network. Unlike other social media platforms that rely on advertising revenue to pay content creators, Nostr allows creators to monetize their content directly with Bitcoin. Imagine that each of your posts has a button that, when clicked, sends appreciation satoshis. Much better than likes and the hijacking of our attention in the process. These are the famous Nostr Zaps.

The code for Nostr is open source, available for anyone to view, use, and modify. This allows for transparency and collaboration in the development of the protocol. Anyone can contribute to Nostr.

How Nostr Works
Each Nostr account is based on a pair of public/private keys. A simple way to think about this is that your public key is your username, and your private key is your password, with one important caveat: unlike a password, your private key cannot be reset if lost. Your public key is usually presented as a string with the prefix **npub1**, and the private key with the prefix **nsec1**: Make sure to store your private key in a safe place and not share it with anyone.

By sharing your public key, others can find you. Your private key allows you to publish notes, interact with others, and verify that it is you doing so through cryptographic signatures. Possession of the key also allows you to migrate to other clients. Imagine you have a Twitter account and want to switch to TikTok. Using Nostr, you can do this without needing to inform anyone. You simply use the new client by logging in with your private key.

On Nostr, everyone uses a client (Primal, Damus, Coracle, Amethyst, Iris, Snort, etc.), the window through which you look at the Nostr protocol. To publish something, you write a note, sign it with your private key (this happens automatically when you press send), and send it to multiple relays with redundancy (servers hosted by someone else or by yourself). To get updates from others, you ask multiple relays if they know anything about those other people. Very simple.

Note: If all the relays you used in the past go offline, all your posts will be irretrievable. This is one of the reasons why Nostr allows users to connect to many relays – it ensures a higher degree of backup.

While a relay can block a user from posting anything on it, they cannot prevent someone from posting on other relays or you running your own relay and transmitting your own messages. Since users are identified by a public key, they do not lose their identities or their follower network if they are banned from a relay because they can simply connect to another or run one on their own. This is exactly what makes Nostr decentralized and censorship-resistant.

If you want to ensure your speech is absolutely uncensorable, you can and should run your own relay. This guarantees that you always have a copy of all your posts and interactions on Nostr forever. If you are running your own Bitcoin node with Umbrel, you can easily run your own Nostr relay alongside your Bitcoin node.

Note: If you notice your Nostr client is slow, it is more likely due to the relays you are using. It might be worth adding some additional relays to your client to make the experience more pleasant.

Watch a video of Uncle Bob Martin, who wrote the renowned books Clean Code and Clean Architecture, talking about the importance of the Nostr protocol in preserving our freedom of speech.

Learn more at: https://nostr.com/

A great Nostr client for beginners is https://primal.net/home.

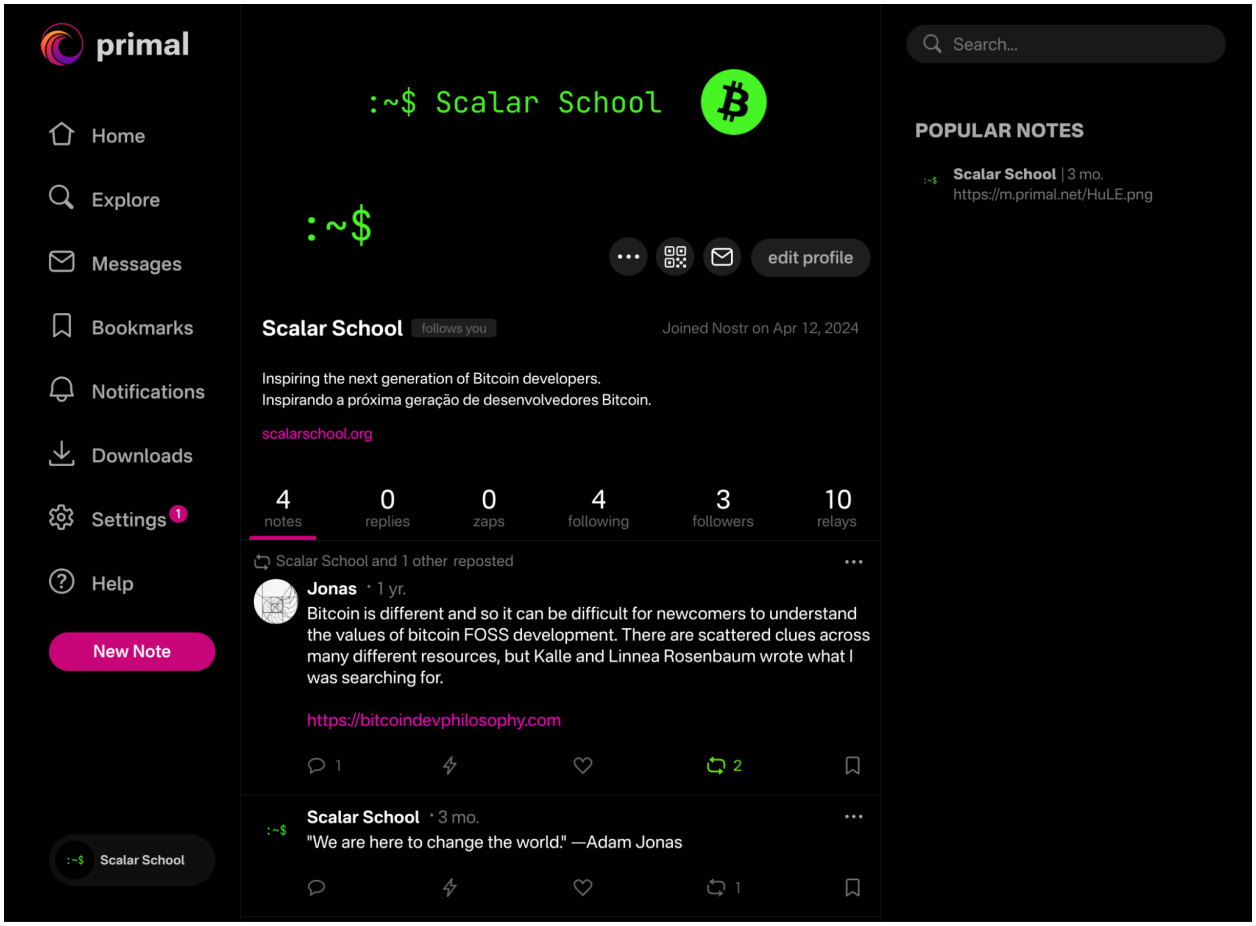
For developers, it is essential to familiarize themselves with the NIPs, or Nostr Implementation Possibilities: https://github.com/nostr-protocol/nips.

And this is a list of well-connected relays. The uptime of the relays is important to maintain the redundancy and censorship resistance of the propagated messages: https://nostr.watch/relays/find

Although Nostr and Bitcoin operate in different domains, their technological and philosophical foundations share many principles. The interoperability between the two technologies can lead to new innovative use cases that combine secure and decentralized communication with private and censorship-resistant financial transactions.

The Bitcoin + Nostr development ecosystem is rapidly growing, with specialized hackathons for Nostr.

Snapshot of Primal



This is our Nostr key. Create your account and follow us there.

npub1jfk9mxsndnwupaergsyat0myst8pygpz2pyx032dz62pefmz22esrcjf2t

10.5 Value4Value

The modern internet is largely monetized through an advertising-based model, where complex algorithms and corporate interests play a central role. Social media platforms, search engines, and other online services collect vast amounts of user data to personalize ads and maximize engagement.

This model is powered by algorithms that analyze user behavior and preferences, often promoting content that increases retention and usage time.

While effective at generating revenue, this system raises significant concerns about privacy, information manipulation, and censorship.

Platforms collect personal data, browsing history, and user interactions.

Using this data, algorithms personalize ads to increase the likelihood of clicks and conversions.

Algorithms promote content that increases usage time, often prioritizing polarizing or sensationalist content to keep users engaged.

Commercial surveillance is used to better understand consumer behavior and target specific products and services.

Companies may censor content that does not align with their commercial or political interests, manipulating public perception and influencing opinions.

Value 4 Value (V4V) Philosophy

The Value 4 Value (V4V) philosophy proposes a different approach, where digital content is monetized directly by users through micropayments, promoting a more open and fair ecosystem. This model is facilitated by Bitcoin and the Lightning Network, enabling fast and low-cost transactions.

Using Bitcoin and the Lightning Network, content creators can receive direct payments from their users, eliminating the need for intermediaries.

Transactions are transparent and respect user privacy, avoiding excessive data collection.

Users are encouraged to directly support the content and creators they value, creating a more direct and honest relationship.

User feedback is valued and integrated into the creation process, promoting greater connection and community participation.

Nostr + Zaps

Zaps are micropayments made through the Lightning Network that integrate directly with the Nostr network to support content and creators. They represent a practical implementation of the Value 4 Value (V4V) philosophy.

Nostr —Zaps: awesome-nostr, Nostr Apps

Other examples of platforms that enable a V4V culture:

Crowdfunding: Geyser Fund

Podcasting: Fountain

Scheduling Meetings: LNCal

The Value 4 Value philosophy represents an important shift from advertising-based and surveillance-based monetization to a more fair and open model. By promoting transparency,

privacy, and direct user engagement, V4V aims to create a digital ecosystem where value is directly exchanged between creators and consumers, without the interference of intermediaries.

This approach not only empowers content creators but also promotes freedom of expression and resistance to censorship, aligning with the principles of decentralization and financial autonomy promoted by Bitcoin. Learn more at https://value4value.info/

10.6 Democratization of Science

The traditional academic system is highly structured, hierarchical, and bureaucratic, based on institutions such as universities, research centers, and scientific journals. Researchers, to advance in their careers, need to publish their work in high-impact journals, which are often indexed in databases like Web of Science, Scopus, and Google Scholar.

This indexing system is fundamental for the visibility and citation of scientific works, which, in turn, is crucial for the reputation and funding of researchers and institutions.

Articles are submitted to journals, where they undergo a peer review process before being accepted for publication. This process can be time-consuming and often subject to bias.

Intellectual property protection is crucial for many universities and research centers, which frequently register patents to protect their innovations. This process can be lengthy and costly, often taking years to complete and limiting the free evolution of these findings and technologies.

Many research projects, especially those funded by corporations, are conducted under non-disclosure agreements, limiting transparency and academic collaboration.

Corporations often fund research with the goal of obtaining results that benefit their commercial interests. This can lead to bias in conclusions and selective publication of results.

The peer review, publication, and patent acquisition process can be very slow, benefiting companies that wish to control the pace of innovation to protect their own interests.

Scientific journal publishers profit from selling subscriptions and individual articles, often at high prices inaccessible to the general public.

Companies that fund research can control when and how results are disclosed, protecting their patents and market strategies.

This paradox—where science, ideally intended to benefit the population, ends up serving private interests—significantly delays the evolution of solutions and technologies that could benefit humanity. By prioritizing the protection of commercial interests and maintaining the status quo, the traditional academic system hinders the rapid and open dissemination of innovative knowledge, limiting global scientific and technological progress.

New Scientific Paradigm

The introduction of new decentralized platforms and technologies, such as Bitcoin and Nostr, offers a promising alternative, promoting transparency and accessibility, and potentially accelerating the pace of scientific innovation.

These technologies open new possibilities for conducting and funding scientific research. The transparency, traceability, and high global availability of the Bitcoin and Nostr protocols allow revolutionary forms of research and development to emerge, without direct dependence on government agencies and competitive, bureaucratic fundraising processes for various types of social and scientific projects.

Anyone can participate and contribute to scientific projects, facilitating the implementation of citizen science initiatives in a truly global economy.

The new scientific paradigm offers a more democratic approach to research and development. Instead of relying on centralized institutions and corporations that often control the scientific agenda according to their financial interests, decentralized technologies like Bitcoin and Nostr enable a more equitable distribution of resources and opportunities.

Decentralized platforms democratize science by allowing anyone, regardless of geographic location or financial status, to contribute to research and development. This is particularly important for scientists and researchers in developing countries who often face significant barriers to obtaining funding and publishing their work.

By promoting transparency, accessibility, and global participation, this new paradigm allows research and development initiatives to be more open, fair, and accessible, benefiting all of humanity.

Global citizen science initiatives become easy to implement with a truly free and global economy. Tools like Bitpac favor the democratic management of resources in projects and organizations.

Crowdfunding platforms like Geyser Fund allow you to publish your project and request donations in Bitcoin to execute it.

The sky is the limit when we have a truly free financial system.

Sci-Hub: A Notable Project

Sci-Hub—https://sci-hub.se/—is an online repository that offers free access to millions of academic and research articles. Founded by Alexandra Elbakyan in 2011, the site emerged as a response to the high cost of accessing scientific publications, which are often beyond the reach of researchers and students, especially in developing countries.

An interesting way to leverage the potential of Sci-Hub is by searching for publications of interest on Google Scholar and then searching on Sci-Hub to bypass potential paywalls.

To maintain its operations and avoid government interference, Sci-Hub accepts Bitcoin donations. This allows the site to continue providing free access to scientific articles, funding its operational costs, and fighting legal actions.

10.7 The Bitcoin Dev Project

From the About session in the website https://bitcoindevs.xyz/.

"Our goal is to provide newcomers with resources and support for your bitcoin open source development journey. We are here to convince you to contribute to bitcoin open source projects. We measure our success by action, not passive consumption of educational materials.

There is an oft-repeated sentiment in the community that bitcoin does not need you. While bitcoin is designed to be resilient, we do need you. Bitcoin needs all the talent and energy it can gather to solve some of the most difficult technical problems of our time. Bitcoin in your hands changes everything."

The Bitcoin Dev Project is a great initiative by Adam Jonas to compile the path to becoming a contributor in the Bitcoin ecosystem. The project offers a robust suite of tools designed to aid developers in their learning journey, including AI and search tools, and an incredible game called Saving Satoshi where developers can practice Bitcoin concepts. There is also a compiled list of Good First Issues that developers can attempt to tackle whenever they feel ready.

10.8 Bitcoin Optech

Bitcoin Optech (Bitcoin Operations Technology Group) is an organization dedicated to helping Bitcoin businesses adopt scaling technologies and optimize their operational efficiency. The primary goal of Bitcoin Optech is to provide comprehensive resources, technical guidance, and best practices to enhance the functionality and scalability of the Bitcoin network. Follow their activities and newsletter at https://bitcoinops.org/

11. Career in Free and Open Source Software Development

"If a YouTube video goes down, we might lament the loss of valuable knowledge, but if an open source project goes down, it can literally break the internet."

—Nadia Eghbal, Working in Public, The Making and Maintenance of Open Source Software

Free and open source software is a fundamental pillar of modern digital society, promoting innovation, security, and inclusion. It offers a sustainable and ethical alternative to proprietary software, while also opening new career and professional development possibilities for individuals worldwide.

Today, working on FOSS (Free and Open Source Software) projects has become a viable career option distinct from the traditional corporate system. The importance of FOSS for humanity is immense. It allows anyone, regardless of financial resources, to access high-quality tools, promoting digital inclusion and equal opportunities.

Moreover, the transparency of open source increases security, as anyone can inspect and improve the code, reducing dependence on proprietary solutions and increasing trust in the software used.

Benefits of Contributing to Open Source

In the open source community, some of the world's best programmers work on the most complex and challenging software problems. Participating in this community is extremely beneficial for a developer's resume for several reasons.

First, the collaborative nature of open source projects requires developers to have advanced technical skills and the ability to solve complex problems effectively. Contributing to these projects demonstrates that a developer can work in a team, communicate well, and solve difficult problems — skills highly valued by employers.

Open source projects are often responsible for some of the most significant technological innovations. Participating and contributing to these projects allows developers to work with the latest technologies and advances in the software field. This not only increases the developer's practical experience but also keeps their skills up to date with industry trends.

Furthermore, the visibility that comes with contributing to open source projects is invaluable. A developer's code and contributions are publicly accessible, allowing potential employers to directly see the quality of the work and the developer's level of commitment. This can lead to career opportunities, as many tech companies actively monitor these communities for talent.

Participating in the open source community also offers networking opportunities with other experienced and influential developers in the industry. These connections can lead to future collaborations, job recommendations, and mentorship, all of which can be extremely valuable for career development.

Contrary to the misconception that open source developers are only volunteers, many start as volunteers to demonstrate their work ethic and network. However, there are many opportunities to work in truly profitable positions.

Financial Opportunities in Open Source

Most open source projects are, in fact, priced in dollars. This is because the US dollar is the most widely accepted and used currency internationally, especially in online transactions and crowdfunding platforms. This standard facilitates the coordination of projects and payments among developers from different countries.

Voluntarily contributing to open source projects allows developers to build a public portfolio, gain visibility in the community, and establish valuable connections. These initial contributions often open doors to paid positions within the same project or in other open source initiatives.

A career as an open source developer can be rewarding and full of opportunities, both for professional growth and for making a positive impact on the global community. First and foremost, it offers great flexibility and autonomy.

Flexibility and Impact
Open source developers can work on projects from anywhere in the world, without the need to be physically present in an office or follow rigid schedules. Remote collaboration is common practice, allowing them to work at times that best suit their lifestyle.

Contributing to open source projects also allows developers to positively impact the global community. They can solve real problems and improve technologies used by many people daily.

Participating in these projects also provides continuous skill development, as developers work with the latest technologies and industry standards, receiving direct feedback from other experienced professionals, which accelerates their learning.

One of the great advantages of an open source career is the recognition and career opportunities it can provide. Significant contributions to open source projects increase the visibility and reputation of the developer in the tech community. There are various funding opportunities through grants and sponsorships from companies that support open source projects.

Professionals who stand out in this field usually have a proactive and self-taught profile. They can learn new technologies and tools on their own and demonstrate initiative to identify problems and create solutions. Communication skills are vital, as most of the work involves collaborating with other developers globally. Documenting work and clearly communicating ideas in discussions and pull requests is essential.

Open source developers need to have strong technical skills, being proficient in relevant programming languages and development tools. Familiarity with version control systems, especially Git, is crucial. A passion for technology and innovation is another factor that sets these professionals apart. They are motivated by the desire to create high-quality software that solves real problems and improves people's lives.

Adaptability and resilience are also important qualities, as the open source development environment can be dynamic and challenging. Developers need to be prepared to handle constant feedback and rapid changes.

There are many success stories that illustrate the importance and potential of an open source career. Linus Torvalds, creator of Linux, started with an open source project that became one of the most important operating systems in the world. Guido van Rossum, creator of the Python programming language, also launched his project as open source, which is now widely used.

Moreover, a career built on open source software is meritocratic. If you stand out, it's because you are genuinely generating value. This can become a defense against arbitrary corporate systems that, in a layoff situation, end up discarding excellent professionals due to external reasons.

11.1 Philosophy of Bitcoin Development

Being a Bitcoin developer involves significant ethical and social responsibility. Bitcoin is one of the best tools ever created to free humanity from the tyranny of oppressive governments and restrictive monetary policies.

To guide this ethical commitment, the Bitcoin community has compiled a document listing the main philosophical aspects of Bitcoin development. Learn more at https://rosenbaum.se/btcphil/

11.2 Bitcoin FOSS Development

Bitcoin exemplifies the power and importance of FOSS in creating decentralized and innovative systems that promote freedom, security, and financial inclusion. The Bitcoin ecosystem, with its various open-source applications, continues to grow and evolve, offering vast opportunities for developers and users worldwide.

Start Your Career in Bitcoin Open Source Development

Why consider a career in bitcoin open source development?
If you aspire to have professional freedom, work on something that will impact the lives of countless people across the world, write code that will span generations, and collaborate with some of the most gifted developers on the planet to solve some of the hardest technical problems of our age, then you are in the right place.

Earning a grant for full-time bitcoin open source work
Financial support for bitcoin open source work typically comes in the form of a grant. Grants usually last for one year and many are renewed.

Bitcoin funding is different from other open source and other cryptocurrency projects. Getting a grant in bitcoin is pretty straightforward. You either need someone to vouch for you or you need to do work - ideally both. While grant programs often have open applications, the secret to getting funding is not much of a secret. **Start doing the job for free. It establishes you as a contributor and proves your motivation. It shows that you are a good investment.** Applicants will have much more success if they do the work and then apply. For most jobs, the applicant is trying to convince the employer that they are capable of doing the job. But the job-seeker has no idea what the work or environment is actually like. In open source, one does not have to guess. Do the work. Demonstrate capability. Then ask for support.

Doing the work also means demonstrating your work. Not all work in open source work is as visible as writing code. The less visible work is no less valuable, but it is in your best interest to create artifacts of your effort. Transparency is your friend. If you learn something, it is helpful to codify it by writing a blog post or keeping a running log. I have seen some write a bi-weekly summary email tracking their progress. Code reviews used to be hard to track, but now GitHub does a better job crediting a green square. Taking long walks to reflect on how to approach a problem is necessary, but writing up your conclusions in a public place is valuable collateral that can forever serve as evidence of your progress.

(Excerpt from A guide for bitcoin open-source grant seekers)

Funding Organizations
Spiral is the bitcoin R&D arm of Block who have been distributing grants since 2019.
Brink is a 501c3, mostly focused on funding Bitcoin Core developers, established in 2020.
OpenSats is a 501c3, established in 2021.
Human Rights Foundation is a 501c3 that has been distributing grants since 2020.
Various exchanges and individuals have sponsored devs in the past, but the above orgs have become the main distributors of grants over the last couple of years.

Initiatives such as the Scalar School receive funding from the Human Rights Foundation to create open-source development study communities for women and prepare the next generations of fellows.

Check out this repository for a list of projects that offer opportunities for open source developers in the Bitcoin ecosystem: https://github.com/biohazel/freedom-devs

But first, read this article by BDK contributor Daniela Brozzoni, offering tips on how to interact and contribute to the open-source environment.

11.3 Chaincode Labs

Chaincode Labs is a research and development organization dedicated to advancing Bitcoin and its software ecosystem. Founded by respected members of the Bitcoin community,

Chaincode Labs focuses on contributing to Bitcoin Core development as well as educating and supporting new developers in the field. Chaincode Labs programs also serve as on-ramps to other projects in the ecosystem. Learn more at https://bitcoindevs.xyz/.

Chaincode Labs Training Programs

Chaincode Labs offers training programs for developers who want to deepen their understanding of Bitcoin development and contribute to the network. These programs are designed to provide a deep understanding of the technical concepts underlying Bitcoin and offer practical experience in Bitcoin Core development.

Chaincode Residency Program

The Chaincode Residency Program is one of the most well-known training programs offered by Chaincode Labs. This intensive program typically lasts several weeks, during which participants receive guidance and support from experienced Bitcoin Core developers. Here are some of the key components of the program:

Workshops and Lectures: Residents participate in workshops and lectures covering a wide range of technical topics, from the structure of Bitcoin Core to advanced cryptography concepts.

Practical Projects: Participants work on practical projects, directly contributing to Bitcoin Core or developing related tools and improvements.

Mentorship: Each resident is paired with an experienced mentor who provides ongoing guidance and feedback throughout the program.

Collaborative Environment: The program fosters a collaborative environment where residents can work together, exchange ideas, and solve technical problems.

Chaincode Labs Seminar Series

In addition to the Residency Program, Chaincode Labs also organizes a series of seminars. These seminars are less intensive than the residency program but still offer valuable opportunities for developers to learn about specific topics related to Bitcoin and cryptography.

Focused Topics: Each seminar focuses on a specific topic, such as network security, decentralized system design, or second-layer protocols.

Interactive Sessions: Seminars typically include interactive sessions where participants can ask questions and discuss ideas with Chaincode Labs experts.

Online Access: Many seminars are offered online, allowing developers from around the world to participate.

Benefits of Participating in Chaincode Labs Programs

Participating in Chaincode Labs training programs offers several benefits for developers:

In-depth Technical Training: The programs provide in-depth, hands-on technical training, helping developers enhance their skills and better understand Bitcoin's workings.

Networking Opportunities: Participants have the chance to meet and collaborate with experienced developers and other participants, building a valuable network of contacts in the Bitcoin development community.

Practical Contributions: Through practical projects, participants can make direct contributions to the Bitcoin ecosystem, gaining practical experience and recognition in the community.

Career Opportunities: The experience and connections made during these programs can open doors to career opportunities in fintech companies, blockchain startups, and research organizations.

Chaincode Labs plays a crucial role in advancing Bitcoin development and training new developers in the field. Its training programs offer an invaluable opportunity for developers to learn from the best, contribute to the Bitcoin project, and advance their careers.

11.4 Summer of Bitcoin

Summer of Bitcoin is a global online internship program held during the northern hemisphere summer. This period corresponds to June, July, and August when many universities in the northern hemisphere are on break, allowing students to participate in the online internship program more intensively and dedicatedly.

Focused on introducing university students to the development and open-source design of Bitcoin, this program offers a unique opportunity for students to gain practical experience and deep knowledge about the functioning of Bitcoin while contributing to significant projects in the open-source ecosystem.

The program identifies and trains new talent in the field of Bitcoin development and design, preparing students for future careers in the industry. By involving students in open-source projects, Summer of Bitcoin promotes continuous innovation in the Bitcoin ecosystem. Student contributions help solve real problems and improve Bitcoin's functionality and usability.

Students who excel in the program have the opportunity to connect with industry leaders and potential employers. Recommendations and recognitions obtained during the program can open doors to careers in the sector. Participating in the program allows students to develop advanced technical skills in programming, design, and security, as well as collaboration and communication skills, which are essential in the open-source development environment.

How Summer of Bitcoin Works

University students interested in the program apply through an online application process, where they must demonstrate their previous experiences and interest in Bitcoin.

Applications undergo rigorous screening to select the most promising candidates based on their skills, motivation, and potential contributions to open-source projects.

Selected students learn how Bitcoin works through a carefully curated set of high-rigor technical resources.

After the initial training, students submit project proposals, detailing how they plan to contribute to a specific open-source project related to Bitcoin.

Developer Track: Students choosing the developer track work on programming and software development tasks, directly contributing to the Bitcoin project's codebase.

Designer Track: Students choosing the designer track focus on creating intuitive and appealing user experiences for open-source Bitcoin products and services.

Students receive mentorship from experienced developers and designers in the Bitcoin ecosystem. Mentors provide technical guidance, project progress feedback, and continuous support throughout the program.

Program Evaluation and Benefits

At the end of the program, student projects are evaluated based on quality, impact, and contribution to the Bitcoin open-source community. Outstanding students receive recommendations and recognitions, increasing their visibility in the Bitcoin developer and designer community and opening career opportunities in the sector.

Summer of Bitcoin is an excellent platform for university students who wish to engage in open-source development and make significant contributions to the Bitcoin ecosystem while developing valuable skills and building a professional network.

Participants are compensated for their participation in the program.

11.5 Scalar School

Scalar School is a training school for future Bitcoin developers in Brazil, funded by the Human Rights Foundation.

Women are systematically denied opportunities to fully develop as developers in the Bitcoin ecosystem. The open source ecosystem, especially in Bitcoin development, has proven to be a hostile environment for women. Harassment, microaggressions, and exclusion are recurring issues that hinder women's participation and progress in this field. The lack of a safe and

welcoming environment prevents many women from taking advantage of learning and technical growth opportunities.

Verbal microaggressions, differential treatment, social ostracism, and passive-aggressiveness disrupt women's learning processes, damaging their mental health and intellectual performance. We want to eliminate these intellectual obstacles and offer a community where women feel safe and supported to study and explore Bitcoin technology without hostile incidents.

To combat this hostility, we have decided to transform Scalar School into a women-only program. We aim to create a safe and supportive space where women can focus on studying complex technical topics without interruptions and without anyone questioning their intellectual potential.

These safe spaces are essential for protecting women's mental well-being and their ability to learn and grow. While the presence of technically qualified men can be beneficial, the greater problem is that women, when trying to find and fully belong to a study community, are being deprived of learning opportunities due to harassment, belittlement, and exclusion. This is a serious issue that must be addressed by the entire Bitcoin community, both nationally and internationally.

Initially, we considered educating and controlling the open community with strict codes of conduct. However, we realized that an open program would lead to frequent incidents. Currently, much of the open Bitcoin developer community in Brazil is essentially an organized hate group against women. To learn more, visit: biohazel.github.io and read https://bitfeminist.substack.com/p/empowering-extreme-misogyny-through

We thought of women-only environments with the logic of a falling airplane. Our society is collapsing, and we have an obligation to put the oxygen mask on ourselves first. We are aware that it is naive to believe that we can change an entire social structure overnight. For this, we first need to strengthen ourselves as a minority by creating a women-only space. We are inspired by the success of the Bitcoin Dada project in bringing women to Bitcoin and hope to replicate this success in the Brazilian context.

Our curriculum is based on the best technical programs in the world, such as Base58, Chaincode Labs, Bitshala, "Learning Bitcoin from the Command Line," and canonical technical books in the field. Additionally, we actively promote community growth and strengthening, and we hope to soon resume Bitdevs Ribeirão Preto with a version for women-only.

Scalar School plays a vital role in the open source ecosystem, creating a community of builders and offering training for beginner developers who will be the future leaders in Bitcoin FOSS research and development programs.

Our dream is to become a global reference, an independent research and development group focused on the values of inclusion, freedom, and overcoming artificial barriers imposed by society. We want to experience the democratization of science and knowledge in the Bitcoin ecosystem.

Women are a historically marginalized group in high-paying fields like technology. Our school is the first Bitcoin open source developer training project led by a woman in Brazil, making Scalar School a historic initiative in the national tech scene. (If I am mistaken, please introduce me to the first one.)

We are proud to announce that we have support and funding from the Human Rights Foundation (HRF) to build our community. This support allows us to offer high-quality training and create a safe environment for all.

For women developers, or those aspiring to be, we invite you to join this transformative journey by filling out our interest form for the first Full-Stack Bitcoin: From Mining to Lightning track.

If you are a well-intentioned and experienced developer in the Bitcoin ecosystem—of any gender—and would like to volunteer as a teacher in this women-only program, please contact us at scalarbitcoin@gmail.com.

12. Bitcoin is For Everyone

Bitcoin, as an open-source technology, represents much more than a financial revolution; it is a symbol of inclusion, transparency, and innovation. Since its launch in 2009 by Satoshi Nakamoto, Bitcoin has ushered in a new era of financial freedom and democratic access to technical knowledge.

The open-source nature of Bitcoin allows anyone, anywhere, to participate in the development and enhancement of this technology. This creates an environment where collaboration is encouraged and innovation is constant.

For many, open-source development in Bitcoin is not just a career opportunity but a mission. It is a way to work on something meaningful with a global and lasting impact.

However, it is crucial to recognize and address the social challenges that come with this freedom. Unfortunately, **misogyny** is still prevalent in many technical communities, especially in Bitcoin.

Women and minorities often face significant prejudices and barriers, from microaggressions to explicit harassment, which limit their participation and contributions. Our community and school emerge as a strategy to amplify women's voices in the ecosystem.

At Scalar School of Bitcoin Developers, we believe that Bitcoin should be for everyone. More importantly, we believe it is a powerful tool to reduce social inequality, disparities in opportunities, wages, and financial independence between men and women. We also believe that communities with gender balance—close to a 50/50 distribution—are fairer, healthier, and more creative.

Therefore, our initial commitment is to create a safe, inclusive, and respectful environment for women. Fighting against social misogyny may seem like an impossible task, but we dream of a day when we can open our community to all genders in the future.

By promoting a culture of respect and inclusion, we can not only improve Bitcoin technology but also create a more just and equitable society.

Additionally, we also want to experience freedom. We want to announce public events in places that do not require KYC and know that we are safe, that we will not be persecuted or harassed. We want to know freedom, not just hear stories about it.

We believe it is very important for male developers in the ecosystem to become aware of the implications of this war against women in Bitcoin technical spaces and to create parallel initiatives to educate the community so that, in the future, we can bring the genders together in a healthy and respectful way, putting the open-source Bitcoin developer community on a 50/50 axis. Only then will we know the true potential of Bitcoin's social transformation.

Together, we can build a future where knowledge and Bitcoin are truly for everyone, and where this form of financing brings prosperity to initiatives that genuinely generate value for humanity.

13. Powered By Bitcoin

Until now, anything has been allowed as long as it involved building Bitcoin infrastructure, including the normalization and propagation of extreme misogyny. As the Brazilian rapper Projota said,

"The constructions will be built. I learned from my father, who is a bricklayer."

For better or for worse, Bitcoin will be built. In the current scenario, we need to deal with a persistent social debt left by the older developer communities in the ecosystem, who unfortunately still operate today.

Fortunately, the situation is starting to change, allowing us to interact with the development ecosystem without having to go through them. Additionally, some of the older developers are respectful and well-intentioned individuals. We just have to find them.

Well then. We have our problems exposed, and we have Bitcoin as a powerful tool for financial freedom and building a new society. Some people will be interested in code development, while others will want to build a better world and will seek creative ways to do so.

And now, what will we build with all this knowledge, purchasing power, and financial freedom that Bitcoin brings us?

I will make a suggestion, but I also suggest you brainstorm and come up with many more—just as long as it doesn't involve building something that favors a retrograde, anti-feminist, misogynistic, and red pill world, please. No more of humanity's same old mistakes. **We need more feminism and the defense of women's and children's rights, not less.**

Did you know that the global baby products market was estimated at approximately USD 320.65 billion in 2023 and is expected to grow at a compound annual growth rate (CAGR) of 5.9% from 2024 to 2030? The main product segments include baby food, baby cosmetics and hygiene products, baby safety and convenience products, and baby clothing and accessories.

Among these products are a myriad of gadgets that babies don't need, but that are sold to innocent families who get their knowledge about the best ways to raise babies from Instagram influencers who are precisely selling products. Nothing could be more fiat-standard than that.

Babies don't have purchasing power, but they are co-opted to convince their parents to spend. Understanding their biological functioning makes it clear that investments in "things for their development" are basically unnecessary.

However, it is necessary to educate parents about how a baby works, both biologically and behaviorally, and the best ways to handle and interact with them. This, of course, does not yield profits. It takes time, interested families, and caregivers who are attentive trainers to demonstrate and explain why certain interactions and interventions are made, as well as why environments are prepared in specific ways—with less, not more.

The reality is similar to web3 and meme coins distracting people from Bitcoin. The true Bitcoin-standard treatment that babies deserve is far from being achieved, because the fiat system is manipulable to build things we don't need. But what if Bitcoin could solve this?

Emmi Pikler's Revolutionary Childcare Practices

Emmi Pikler was a Hungarian pediatrician whose observations and practices revolutionized childcare. Working in an orphanage in Budapest, Pikler developed methods that emphasized children's autonomy and freedom of movement. Her approach aimed to respect each child's natural development pace, promoting healthy physical and emotional growth.

In traditional orphanages, children often suffered from a lack of stimulation, little interaction, and standardized care, resulting in physical and emotional developmental problems and difficulty fitting into society post-orphanage.

Pikler observed that these methods failed to provide an environment that supported children's healthy development. She set out to create a care system that met the individual needs of children and promoted their mental health. Imagine a world where even very poor children are perfectly well-adjusted and make the most of their neurological potential.

Pikler developed specific routines and practices that promoted children's independence and natural development, as well as respectful communication between caregiver and child, with well-defined boundaries from an early age.

Freedom of Movement: Children were placed on the floor on their backs and encouraged to move freely. This allowed them to develop motor skills naturally, without direct caregiver intervention, and at their own pace.

Minimal Intervention: Caregivers avoided manipulating or forcing positions on children, such as sitting them up or placing them on their stomachs before they were ready for these movements on their own.

Safe and Stimulating Environment: Pikler introduced low furniture and structures like the Pikler Triangle, which encouraged safe exploration and physical development.

Attentive Observation: Caregivers carefully observed each child to understand their needs and respond appropriately, without overstimulation or underestimation.

Respect for the Child's Time: Each child's own development pace was respected, promoting confidence and emotional security.

Children cared for according to Pikler's methods left the orphanage with impeccable spinal postures, well-developed motor skills, and a strong sense of autonomy. They demonstrated greater resilience, confidence, and the ability to engage in healthy social interactions.

The success of children in Pikler's orphanage evidenced that respectful care based on autonomy and knowledge of human biology could result in well-adjusted and healthy adults, even in precarious conditions such as war orphanages. Magda Gerber, one of her students, brought this knowledge to the United States, where it became well known among high-income families in California. Nannies trained in the Pikler method are the highest-paid in Silicon Valley, and I have been one of them.

But this idea doesn't drive the baby products industry, nor does it go viral among the masses for the same reason. Only a new economic and value system could make these basic values the norm. **We are confident that the Bitcoin standard will also adjust our perception of values regarding human development, not just code development.**

Focus on Bitcoin Development

However, first and foremost, we need to focus on developing Bitcoin as software and implementing Bitcoin itself. Through Bitcoin, we have the chance to build a better future where financial freedom, real freedom, and ethics walk hand in hand, creating a lasting positive impact on humanity.

Study Bitcoin, build Bitcoin, live Bitcoin. But know that human life goes beyond Bitcoin, and what we do and build with Bitcoin also matters.

14. Final Notes

14.1 Disclaimer

SCALAR SCHOOL DOES NOT PROVIDE INVESTMENT ADVICE. WE ARE A GROUP FOCUSED ON EDUCATION, RESEARCH, AND DEVELOPMENT OF BITCOIN TECHNOLOGY.

The information contained in this book is provided solely for educational and informational purposes. While the author has made efforts to ensure the accuracy of the information presented, no guarantees, explicit or implicit, are made regarding the accuracy, completeness, or adequacy of the same.

Investing and transacting in Bitcoin involves significant risks. Before making any financial decisions, readers are encouraged to conduct their own research and, if necessary, consult a qualified financial professional.

The author will not be liable for any loss or damage, including but not limited to indirect or consequential losses or damages, or any loss or damage whatsoever arising from the loss of data or profits, resulting from the use or reliance on the information contained in this book.

14.2 Contact

Questions, suggestions, and corrections can be sent to scalarbitcoin@gmail.com, addressed to Luciana. Or via issue on the repository: https://github.com/biohazel/scalar-school-handbook.

Nostr: npub1jfk9mxsndnwupaergsyat0myst8pygpz2pyx032dz62pefmz22esrcjf2t
Instagram: @scalar.school
X: @scalarschool

14.3 Afterword

A better world is possible. Absolutely everything is invented in our society, which means that as humanity, we have the potential to orchestrate a joint action and build a better world in 24 hours.

Never stop dreaming, nor having fun learning, coding, and working with Bitcoin. May the universe grant you wisdom and prosperity through knowledge. Happy studying.