Bitcoin: A Peer-to-Peer Electronic Cash System

Satoshi Nakamoto
satoshin@gmx.com
www.bitcoin.org

Abstract. A purely peer-to-peer version of electronic cash would allow online payments to be sent directly from one party to another without going through a financial institution. Digital signatures provide part of the solution, but the main benefits are lost if a trusted third party is still required to prevent double-spending. We propose a solution to the double-spending problem using a peer-to-peer network. The network timestamps transactions by hashing them into an ongoing chain of hash-based proof-of-work, forming a record that cannot be changed without redoing the proof-of-work. The longest chain not only serves as proof of the sequence of events witnessed, but proof that it came from the largest pool of CPU power. As long as a majority of CPU power is controlled by nodes that are not cooperating to attack the network, they'll generate the longest chain and outpace attackers. The network itself requires minimal structure. Messages are broadcast on a best effort basis, and nodes can leave and rejoin the network at will, accepting the longest proof-of-work chain as proof of what happened while they were gone.

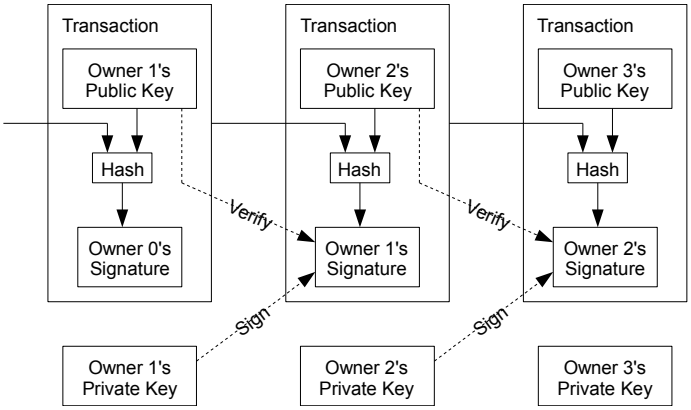
1. Introduction

Commerce on the Internet has come to rely almost exclusively on financial institutions serving as trusted third parties to process electronic payments. While the system works well enough for most transactions, it still suffers from the inherent weaknesses of the trust based model. Completely non-reversible transactions are not really possible, since financial institutions cannot avoid mediating disputes. The cost of mediation increases transaction costs, limiting the minimum practical transaction size and cutting off the possibility for small casual transactions, and there is a broader cost in the loss of ability to make non-reversible payments for non-reversible services. With the possibility of reversal, the need for trust spreads. Merchants must be wary of their customers, hassling them for more information than they would otherwise need. A certain percentage of fraud is accepted as unavoidable. These costs and payment uncertainties can be avoided in person by using physical currency, but no mechanism exists to make payments over a communications channel without a trusted party.

What is needed is an electronic payment system based on cryptographic proof instead of trust, allowing any two willing parties to transact directly with each other without the need for a trusted third party. Transactions that are computationally impractical to reverse would protect sellers from fraud, and routine escrow mechanisms could easily be implemented to protect buyers. In this paper, we propose a solution to the double-spending problem using a peer-to-peer distributed timestamp server to generate computational proof of the chronological order of transactions. The system is secure as long as honest nodes collectively control more CPU power than any cooperating group of attacker nodes.

2. Transactions

We define an electronic coin as a chain of digital signatures. Each owner transfers the coin to the next by digitally signing a hash of the previous transaction and the public key of the next owner and adding these to the end of the coin. A payee can verify the signatures to verify the chain of ownership.

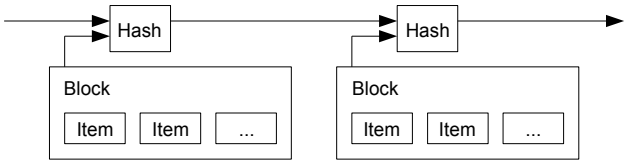


The problem of course is the payee can't verify that one of the owners did not double-spend the coin. A common solution is to introduce a trusted central authority, or mint, that checks every transaction for double spending. After each transaction, the coin must be returned to the mint to issue a new coin, and only coins issued directly from the mint are trusted not to be double-spent. The problem with this solution is that the fate of the entire money system depends on the company running the mint, with every transaction having to go through them, just like a bank.

We need a way for the payee to know that the previous owners did not sign any earlier transactions. For our purposes, the earliest transaction is the one that counts, so we don't care about later attempts to double-spend. The only way to confirm the absence of a transaction is to be aware of all transactions. In the mint based model, the mint was aware of all transactions and decided which arrived first. To accomplish this without a trusted party, transactions must be publicly announced [1], and we need a system for participants to agree on a single history of the order in which they were received. The payee needs proof that at the time of each transaction, the majority of nodes agreed it was the first received.

3. Timestamp Server

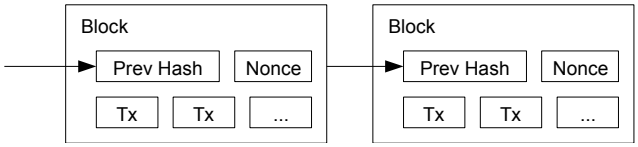
The solution we propose begins with a timestamp server. A timestamp server works by taking a hash of a block of items to be timestamped and widely publishing the hash, such as in a newspaper or Usenet post [2-5]. The timestamp proves that the data must have existed at the time, obviously, in order to get into the hash. Each timestamp includes the previous timestamp in its hash, forming a chain, with each additional timestamp reinforcing the ones before it.



4. Proof-of-Work

To implement a distributed timestamp server on a peer-to-peer basis, we will need to use a proof-of-work system similar to Adam Back's Hashcash [6], rather than newspaper or Usenet posts. The proof-of-work involves scanning for a value that when hashed, such as with SHA-256, the hash begins with a number of zero bits. The average work required is exponential in the number of zero bits required and can be verified by executing a single hash.

For our timestamp network, we implement the proof-of-work by incrementing a nonce in the block until a value is found that gives the block's hash the required zero bits. Once the CPU effort has been expended to make it satisfy the proof-of-work, the block cannot be changed without redoing the work. As later blocks are chained after it, the work to change the block would include redoing all the blocks after it.



The proof-of-work also solves the problem of determining representation in majority decision making. If the majority were based on one-IP-address-one-vote, it could be subverted by anyone able to allocate many IPs. Proof-of-work is essentially one-CPU-one-vote. The majority decision is represented by the longest chain, which has the greatest proof-of-work effort invested in it. If a majority of CPU power is controlled by honest nodes, the honest chain will grow the fastest and outpace any competing chains. To modify a past block, an attacker would have to redo the proof-of-work of the block and all blocks after it and then catch up with and surpass the work of the honest nodes. We will show later that the probability of a slower attacker catching up diminishes exponentially as subsequent blocks are added.

To compensate for increasing hardware speed and varying interest in running nodes over time, the proof-of-work difficulty is determined by a moving average targeting an average number of blocks per hour. If they're generated too fast, the difficulty increases.

5. Network

The steps to run the network are as follows:

1) New transactions are broadcast to all nodes.
2) Each node collects new transactions into a block.
3) Each node works on finding a difficult proof-of-work for its block.
4) When a node finds a proof-of-work, it broadcasts the block to all nodes.
5) Nodes accept the block only if all transactions in it are valid and not already spent.
6) Nodes express their acceptance of the block by working on creating the next block in the chain, using the hash of the accepted block as the previous hash.

Nodes always consider the longest chain to be the correct one and will keep working on extending it. If two nodes broadcast different versions of the next block simultaneously, some nodes may receive one or the other first. In that case, they work on the first one they received, but save the other branch in case it becomes longer. The tie will be broken when the next proof-of-work is found and one branch becomes longer; the nodes that were working on the other branch will then switch to the longer one.

New transaction broadcasts do not necessarily need to reach all nodes. As long as they reach many nodes, they will get into a block before long. Block broadcasts are also tolerant of dropped messages. If a node does not receive a block, it will request it when it receives the next block and realizes it missed one.

6. Incentive

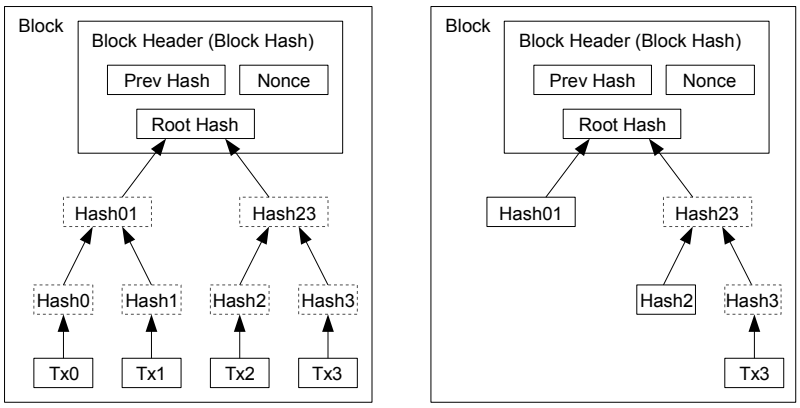
By convention, the first transaction in a block is a special transaction that starts a new coin owned by the creator of the block. This adds an incentive for nodes to support the network, and provides a way to initially distribute coins into circulation, since there is no central authority to issue them. The steady addition of a constant of amount of new coins is analogous to gold miners expending resources to add gold to circulation. In our case, it is CPU time and electricity that is expended.

The incentive can also be funded with transaction fees. If the output value of a transaction is less than its input value, the difference is a transaction fee that is added to the incentive value of the block containing the transaction. Once a predetermined number of coins have entered circulation, the incentive can transition entirely to transaction fees and be completely inflation free.

The incentive may help encourage nodes to stay honest. If a greedy attacker is able to assemble more CPU power than all the honest nodes, he would have to choose between using it to defraud people by stealing back his payments, or using it to generate new coins. He ought to find it more profitable to play by the rules, such rules that favour him with more new coins than everyone else combined, than to undermine the system and the validity of his own wealth.

7. Reclaiming Disk Space

Once the latest transaction in a coin is buried under enough blocks, the spent transactions before it can be discarded to save disk space. To facilitate this without breaking the block's hash, transactions are hashed in a Merkle Tree [7][2][5], with only the root included in the block's hash. Old blocks can then be compacted by stubbing off branches of the tree. The interior hashes do not need to be stored.



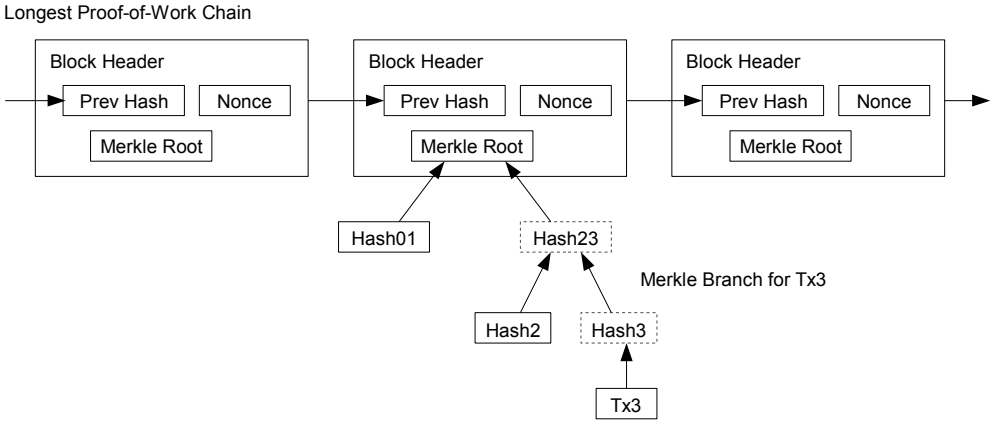
Transactions Hashed in a Merkle Tree

After Pruning Tx0-2 from the Block

A block header with no transactions would be about 80 bytes. If we suppose blocks are generated every 10 minutes, 80 bytes * 6 * 24 * 365 = 4.2MB per year. With computer systems typically selling with 2GB of RAM as of 2008, and Moore's Law predicting current growth of 1.2GB per year, storage should not be a problem even if the block headers must be kept in memory.

8. Simplified Payment Verification

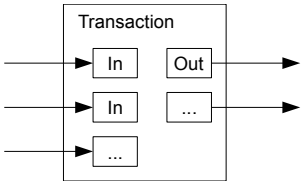
It is possible to verify payments without running a full network node. A user only needs to keep a copy of the block headers of the longest proof-of-work chain, which he can get by querying network nodes until he's convinced he has the longest chain, and obtain the Merkle branch linking the transaction to the block it's timestamped in. He can't check the transaction for himself, but by linking it to a place in the chain, he can see that a network node has accepted it, and blocks added after it further confirm the network has accepted it.



As such, the verification is reliable as long as honest nodes control the network, but is more vulnerable if the network is overpowered by an attacker. While network nodes can verify transactions for themselves, the simplified method can be fooled by an attacker's fabricated transactions for as long as the attacker can continue to overpower the network. One strategy to protect against this would be to accept alerts from network nodes when they detect an invalid block, prompting the user's software to download the full block and alerted transactions to confirm the inconsistency. Businesses that receive frequent payments will probably still want to run their own nodes for more independent security and quicker verification.

9. Combining and Splitting Value

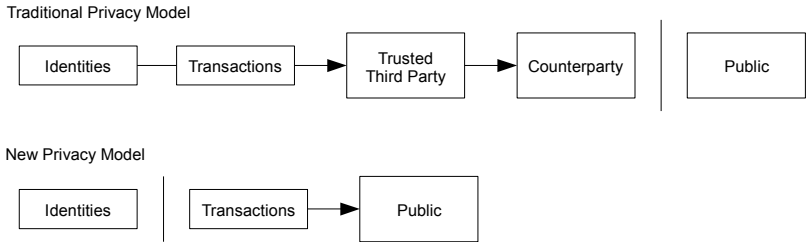
Although it would be possible to handle coins individually, it would be unwieldy to make a separate transaction for every cent in a transfer. To allow value to be split and combined, transactions contain multiple inputs and outputs. Normally there will be either a single input from a larger previous transaction or multiple inputs combining smaller amounts, and at most two outputs: one for the payment, and one returning the change, if any, back to the sender.



It should be noted that fan-out, where a transaction depends on several transactions, and those transactions depend on many more, is not a problem here. There is never the need to extract a complete standalone copy of a transaction's history.

10. Privacy

The traditional banking model achieves a level of privacy by limiting access to information to the parties involved and the trusted third party. The necessity to announce all transactions publicly precludes this method, but privacy can still be maintained by breaking the flow of information in another place: by keeping public keys anonymous. The public can see that someone is sending an amount to someone else, but without information linking the transaction to anyone. This is similar to the level of information released by stock exchanges, where the time and size of individual trades, the "tape", is made public, but without telling who the parties were.



As an additional firewall, a new key pair should be used for each transaction to keep them from being linked to a common owner. Some linking is still unavoidable with multi-input transactions, which necessarily reveal that their inputs were owned by the same owner. The risk is that if the owner of a key is revealed, linking could reveal other transactions that belonged to the same owner.

11. Calculations

We consider the scenario of an attacker trying to generate an alternate chain faster than the honest chain. Even if this is accomplished, it does not throw the system open to arbitrary changes, such as creating value out of thin air or taking money that never belonged to the attacker. Nodes are not going to accept an invalid transaction as payment, and honest nodes will never accept a block containing them. An attacker can only try to change one of his own transactions to take back money he recently spent.

The race between the honest chain and an attacker chain can be characterized as a Binomial Random Walk. The success event is the honest chain being extended by one block, increasing its lead by +1, and the failure event is the attacker's chain being extended by one block, reducing the gap by -1.

The probability of an attacker catching up from a given deficit is analogous to a Gambler's Ruin problem. Suppose a gambler with unlimited credit starts at a deficit and plays potentially an infinite number of trials to try to reach breakeven. We can calculate the probability he ever reaches breakeven, or that an attacker ever catches up with the honest chain, as follows [8]:

p = probability an honest node finds the next block
q = probability the attacker finds the next block
q_z = probability the attacker will ever catch up from z blocks behind

$$q_z = \begin{cases} 1 & \text{if } p \leq q \\ (q/p)^z & \text{if } p > q \end{cases}$$

Given our assumption that $p > q$, the probability drops exponentially as the number of blocks the attacker has to catch up with increases. With the odds against him, if he doesn't make a lucky lunge forward early on, his chances become vanishingly small as he falls further behind.

We now consider how long the recipient of a new transaction needs to wait before being sufficiently certain the sender can't change the transaction. We assume the sender is an attacker who wants to make the recipient believe he paid him for a while, then switch it to pay back to himself after some time has passed. The receiver will be alerted when that happens, but the sender hopes it will be too late.

The receiver generates a new key pair and gives the public key to the sender shortly before signing. This prevents the sender from preparing a chain of blocks ahead of time by working on it continuously until he is lucky enough to get far enough ahead, then executing the transaction at that moment. Once the transaction is sent, the dishonest sender starts working in secret on a parallel chain containing an alternate version of his transaction.

The recipient waits until the transaction has been added to a block and z blocks have been linked after it. He doesn't know the exact amount of progress the attacker has made, but assuming the honest blocks took the average expected time per block, the attacker's potential progress will be a Poisson distribution with expected value:

$$\lambda = z\frac{q}{p}$$

To get the probability the attacker could still catch up now, we multiply the Poisson density for each amount of progress he could have made by the probability he could catch up from that point:

$$\sum_{k=0}^{\infty} \frac{\lambda^k e^{-\lambda}}{k!} \cdot \begin{cases} (q/p)^{(z-k)} & \text{if } k \le z \\ 1 & \text{if } k > z \end{cases}$$

Rearranging to avoid summing the infinite tail of the distribution...

$$1 - \sum_{k=0}^{z} \frac{\lambda^k e^{-\lambda}}{k!} \left(1 - (q/p)^{(z-k)}\right)$$

Converting to C code...

```
#include <math.h>
double AttackerSuccessProbability(double q, int z)
{
    double p = 1.0 - q;
    double lambda = z * (q / p);
    double sum = 1.0;
    int i, k;
    for (k = 0; k <= z; k++)
    {
        double poisson = exp(-lambda);
        for (i = 1; i <= k; i++)
            poisson *= lambda / i;
        sum -= poisson * (1 - pow(q / p, z - k));
    }
    return sum;
}
```

Running some results, we can see the probability drop off exponentially with z.

```
q=0.1
z=0     P=1.0000000
z=1     P=0.2045873
z=2     P=0.0509779
z=3     P=0.0131722
z=4     P=0.0034552
z=5     P=0.0009137
z=6     P=0.0002428
z=7     P=0.0000647
z=8     P=0.0000173
z=9     P=0.0000046
z=10    P=0.0000012

q=0.3
z=0     P=1.0000000
z=5     P=0.1773523
z=10    P=0.0416605
z=15    P=0.0101008
z=20    P=0.0024804
z=25    P=0.0006132
z=30    P=0.0001522
z=35    P=0.0000379
z=40    P=0.0000095
z=45    P=0.0000024
z=50    P=0.0000006
```

Solving for P less than 0.1%...

```
P < 0.001
q=0.10   z=5
q=0.15   z=8
q=0.20   z=11
q=0.25   z=15
q=0.30   z=24
q=0.35   z=41
q=0.40   z=89
q=0.45   z=340
```

12. Conclusion

We have proposed a system for electronic transactions without relying on trust. We started with the usual framework of coins made from digital signatures, which provides strong control of ownership, but is incomplete without a way to prevent double-spending. To solve this, we proposed a peer-to-peer network using proof-of-work to record a public history of transactions that quickly becomes computationally impractical for an attacker to change if honest nodes control a majority of CPU power. The network is robust in its unstructured simplicity. Nodes work all at once with little coordination. They do not need to be identified, since messages are not routed to any particular place and only need to be delivered on a best effort basis. Nodes can leave and rejoin the network at will, accepting the proof-of-work chain as proof of what happened while they were gone. They vote with their CPU power, expressing their acceptance of valid blocks by working on extending them and rejecting invalid blocks by refusing to work on them. Any needed rules and incentives can be enforced with this consensus mechanism.

References

[1] W. Dai, "b-money," http://www.weidai.com/bmoney.txt, 1998.

[2] H. Massias, X.S. Avila, and J.-J. Quisquater, "Design of a secure timestamping service with minimal trust requirements," In *20th Symposium on Information Theory in the Benelux*, May 1999.

[3] S. Haber, W.S. Stornetta, "How to time-stamp a digital document," In *Journal of Cryptology*, vol 3, no 2, pages 99-111, 1991.

[4] D. Bayer, S. Haber, W.S. Stornetta, "Improving the efficiency and reliability of digital time-stamping," In *Sequences II: Methods in Communication, Security and Computer Science*, pages 329-334, 1993.

[5] S. Haber, W.S. Stornetta, "Secure names for bit-strings," In *Proceedings of the 4th ACM Conference on Computer and Communications Security*, pages 28-35, April 1997.

[6] A. Back, "Hashcash - a denial of service counter-measure," http://www.hashcash.org/papers/hashcash.pdf, 2002.

[7] R.C. Merkle, "Protocols for public key cryptosystems," In *Proc. 1980 Symposium on Security and Privacy*, IEEE Computer Society, pages 122-133, April 1980.

[8] W. Feller, "An introduction to probability theory and its applications," 1957.

BR.
THE BLOCKCHAIN REVIEW

Ethereum White Paper Made Simple

A guide to understanding the Ethereum white paper for people without an advanced degree in computer geekery

CONTENTS

WTF

Like its well-known forerunner, the Bitcoin White Paper, Ethereum's founding document has left most of us regular folk scratching our heads in utter bewilderment ever since its release in 2013. I mean common. WTF is going on, right?

Look at this:

> *The intent of Ethereum is to merge together and improve upon the concepts of scripting, altcoins and on-chain meta-protocols, and allow developers to create arbitrary consensus-based applications that have the scalability, standardization, feature-completeness, ease of development and interoperability offered by these different paradigms all at the same time. Ethereum does this by building what is essentially the ultimate abstract foundational layer: a blockchain with a built-in Turing-complete programming language, allowing anyone to write smart contracts and decentralized applications where they can create their own arbitrary rules for ownership, transaction formats and state transition functions.*

That's a real excerpt from the Ethereum White Paper.

WTF?!

Yes, WTF indeed. But luckily for you, the Intrepid

Ventures team has heard your distress calls we're here to help. The confusion and frustration ends here.

Who should read this guide?

This guide will break down the Ethereum white paper so that anyone can understand what Ethereum is, how it works and the problems it solves. If you have a general idea about Ethereum but just can't seem to make sense of it all, this guide is for you.

By reading this guide, you will also gain a better understanding of the differences between Ethereum and Bitcoin and a clearer grasp of where Ethereum fits into the emerging blockchain/cryptocurrency story.

This guide is not for people with advanced knowledge of Ethereum nor will it make you an expert. With this in mind, we will be leaving out some of the more hardcore technical elements that are irrelevant to you gaining a fundamental understanding. We will also be expanding on some concepts where needed.

Why should you care?

That's easy. Ethereum has served to realize the broader potential of blockchain technology beyond bitcoin and first-generation decentralized applications. By offering a platform for developers to build any decentralized application, Ethereum opened up a world of unimagined possibilities. If you

want to understand the decentralized applications of today and tomorrow, you will need to get your head around Ethereum.

So if you can digest the central concepts in the Ethereum white paper, the broader decentralized revolution, which involves hundreds of different cryptocurrencies and other types of blockchain based decentralized applications will begin to make a lot more sense.

Prerequisite reading

For this guide to have maximum impact, you will need to have a basic understanding of Bitcoin and Blockchain, the underlying technology that enables it to operate. If you're unfamiliar with Bitcoin and Blockchain, read our Bitcoin White Paper Made Simple guide first.

Background

Since emerging in 2009, Bitcoin and its core operating technology, blockchain, have laid the foundation for a new era of digital peer-to-peer transactions.

But it would be years after the advent of Bitcoin that the true power of blockchain technology would be realized. In late 2013, Vitalik Buterin, an unknown Russian-Canadian programmer involved with Bitcoin and the crypto community released a white paper that changed the game.

According to Vitalik, the Bitcoin community was approaching blockchain technology in the wrong way. He believed that blockchain applications were being designed to do an extremely limited set of operations.

> *I thought [those in the Bitcoin community] weren't approaching the problem in the right way. I thought they were going after individual applications; they were trying to kind of explicitly support each [use case] in a sort of Swiss Army knife protocol.*

Bitcoin, for example, was developed solely to operate as a peer to peer digital currency. Developers were either required to expand the set of functions offered in existing applications (a time-consuming and challenging process) or develop an entirely new platform altogether. An equally time consuming

and expensive endeavor.

To overcome the technological limitations which were ultimately holding back the development of new blockchain-based applications, Vitalik and a small core team developed Ethereum - A Next-Generation Smart Contract and Decentralized Application Platform.

From humble beginnings, Ethereum has grown into a well known and widely used platform. Ether, the native Ethereum currency and Bitcoin's chief rival has dramatically increased in value. The platform has also been responsible for launching hundreds of other cryptocurrencies and decentralized projects in recent years through a new fundraising mechanism called an Initial Coin Offering (ICO).

What is Ethereum?
A next generation blockchain?

Like Bitcoin, Ethereum is a public blockchain network. They both rely on a blockchain to operate.

Think about the Internet. You can build lots of different applications on top of it like email, online shopping sites and Facebook.

Well, in a way, blockchain technology is a new type of Internet where you can build lots of different applications. Bitcoin and Ethereum are just two examples.

The major difference between Bitcoin and Ethereum, however, is their purpose. Whereas Bitcoin provides one specific function, peer to peer electronic Bitcoin payments, Ethereum offers a platform that enables developers to build and deploy other decentralized applications. You could, for example, build another Bitcoin type currency on Ethereum.

In a nutshell, Ethereum is an open software platform based on blockchain technology that enables developers to build and deploy decentralized applications.

As Vitalik states,

> *Rather than being a closed-ended, single-purpose protocol intended for a specific array of applications in data storage, gambling or finance, Ethereum is open-ended by design, and we believe that it is extremely well-suited to serving as a foundational layer for a very large number of both financial and non-financial protocols in the years to come.*

What's a decentralized application?

A decentralized application or Dapp for short, in this context, refers to an application that is built on top of blockchain technology. Bitcoin is a decentralized cryptocurrency application for payments, for example.

Decentralized applications run on a blockchain and benefit from all of its properties like immutability, security, tamper resistance and zero downtime.

Essentially any service could be turned into a decentralized application. The possibilities are endless.

> *Decentralized applications enable services we already have today, like payments, storage, or computing, but without a central operator of those services.*
>
> Adam Ludwin

The Ethereum evolution

Ethereum helped propel the decentralized application movement forward beyond Bitcoin and first-generation decentralized applications.

The start of the paper focuses on the movement toward next-generation decentralized applications or as Vitalik calls it, 'Bitcoin 2.0'. The paper also describes the Bitcoin protocol, its shortcomings and briefly introduces how Ethereum addresses these shortcomings.

What you need to know

In the immediate years following Bitcoin's emergence, new platforms were developed like colored coins, Mastercoin, Bitshares, and Counterparty in an attempt to offer a more advanced set of functions for users. The problem according to Vitalik was that these platforms were still very narrow in focus.

> *Up until this point all of the protocols that have been invented have been specialized, attempting to offer detailed feature sets targeted toward specific industries or applications usually financial in nature.*

While Bitcoin offered one specific application of blockchain technology, a peer to peer electronic cash system that enables online Bitcoin payments, and other Dapps like colored coins, Mastercoin, Bitshares, and Counterparty all offered a set of slightly more extensive features, Vitalik believed

this was not enough.

Although he believed that Bitcoin was indeed revolutionary and capable of its intended task, he thought it was not a particularly good foundation to build any other applications.

Vitalik noted that developers were using three limited approaches to building applications. They were either

- building a new blockchain, or
- using scripting on top of Bitcoin, or
- building a meta-protocol on top of Bitcoin.

These approaches all came with limitations.

> *Building a new blockchain allows for unlimited freedom in building a feature set, but at the cost of development time and bootstrapping effort. Using scripting is easy to implement and standardize, but is very limited in its capabilities, and meta-protocols, while easy, suffer from faults in scalability. With Ethereum, we intend to build a generalized framework that can provide the advantages of all three paradigms at the same time.*

So, herein lies the intent of Ethereum. To merge and improve upon the approaches outlined above thus enabling developers to build consensus based decentralized applications with greater ease.

The mechanics:
How does Ethereum work?

It's easy to claim that Ethereum enables developers to build whatever decentralized applications they want, but how does it actually achieve this?

Well, according to Vitalik Buterin,

> *Ethereum does this by building what is essentially the ultimate abstract foundational layer: a blockchain with a built-in Turing-complete programming language, allowing anyone to write smart contracts and decentralized applications where they can create their own arbitrary rules for ownership, transaction formats and state transition functions.*

There's a lot to unpack here. Let's take a closer look.

What you need to know

Five main elements enable Ethereum to do what it does. You will need to understand each one, at least on a conceptual level. They include:

- Smart Contracts
- The Ethereum Virtual Machine
- Solidity
- Ether
- Proof of Work

What are Smart contracts?

You've probably heard this term getting thrown around for some time now. A smart contract is just a bunch of code that manages the exchange of anything of value from property and shares to information and money between parties. Smart contracts run on top of the Ethereum blockchain precisely as programmed and become like autonomous agents that execute when previously specified conditions are met.

> *Contracts in Ethereum should not be seen as something that should be "fulfilled" or "complied with"; rather, they are more like "autonomous agents" that live inside of the Ethereum execution environment (EVM), always executing a specific piece of code when "poked" by a message or transaction, and having direct control over their own ether balance and their own key/value store to keep track of persistent variables.*

In Bitcoin, for example, users can only make a simple demand like - send one bitcoin from Alice to Bob. In Ethereum, however, it's possible to create a contract that says send one ether to bob if the date is 25th October 2017 and Bob's current balance is more than 20 ether.

The cool thing about smart contracts is that they self-execute exactly as designed by their creators once certain conditions are met. And this is just a simple

example. Creating a smart contract with infinitely more complicated conditions is possible as well.

A smart contract, could, for example, facilitate the automatic transfer of ownership of a property after a number of critical conditions are met. All of this without any human involvement. Crazy!

That all sounds pretty impressive, right? But how does Ethereum enable smart contracts to run and execute?

The Ethereum Virtual Machine

Smart contracts are powered by the Ethereum Virtual Machine (EVM) and by Ether. The EVM includes a Turing complete scripting language which means that it can solve any computation problem.

The EVM turns Ethereum into a programmable blockchain, keeping all the smart contracts running on time and coordinating them with the rest of the network. In doing so, the EVM enables the development of potentially thousands of different applications all on the Ethereum platform.

What is Solidity?

Ethereum has its own programmable language called "Solidity" which is similar to JavaScript. It enables developers to write programs (smart contracts) on Ethereum and is designed to enhance the Ethereum Virtual Machine (EVM).

What is Ether?

In the Ethereum blockchain, instead of mining for bitcoins, miners work for Ether. Ether is a necessary element for operating the Ethereum network.

It's like a fuel that provides an incentive to ensure developers write quality applications and the network runs smoothly. Beyond a fuel that enables decentralized applications to run, Ether is also a tradeable cryptocurrency.

In Ethereum, Ether is used by developers to pay for transaction fees for services and storage on the network. Every computation on the platform as a result of a transaction has a fee, and the more you need to store the more is paid.

This is because computations and file storage place a strain on the network. So, fees are there to discourage developers from excessively using the network. Without fees to drive user's actions, the Ethereum network simply couldn't function. So, think of Ether like the crypto-fuel that powers the Ethereum network.

How are ethers created, who needs them, and is the supply unlimited?

Ether gets issued at a constant rate through the block mining process. This rate along with the total supply of Ether was decided during the 2014 presale.

- 60 million Ethers were purchased by in the 2014 crowdfunding campaign.
- Another 12 million went to the Ethereum Foundation.
- Supply is not infinite. A maximum of 18 million Ether are mined per year.
- Every 12 seconds, 5 ethers (ETH) are given to the miners that verify transactions on the network.

Developers who build decentralized applications (Dapps) on the Ethereum platform, as well as users who want to interact with smart contracts, will need Ether.

Proof of Work (PoW) - Reaching consensus on Ethereum

For a decentralized system like Ethereum to operate without any central intermediary, there needs to be a way for the network to agree about which transaction records are valid to deter any abuse of service attacks like spamming.

Like the Bitcoin network, Ethereum relies on Proof of Work protocol to reach consensus about which transaction records are the real deal. This is set to

change to a Proof of Stake (PoS) algorithm called Casper, but don't worry about that for now.

Proof of Work aka mining is performed to facilitate transactions on the Ethereum blockchain and discourage bad actors from spamming the network by sending out fraudulent or illegitimate transactions. It requires miners (members in the network with high levels of computing power) to prove that a specified amount work has been completed.

This involves miners solving complex mathematical puzzles that are difficult to solve yet easy to verify. The process demands lots of expensive computational effort (lots of hardware equipment and electricity usage) as miners use expensive computer equipment to repeatedly and rapidly guess answers to a mathematical puzzle until someone wins.

Because these mathematical puzzles require so much work to solve, fraudulent transactions become infeasible. They are just not worth it!

Only blocks that contain the answer to the complex mathematical problem will be accepted and added to the Ethereum blockchain. This occurs every 15 seconds, on average.

Miners that successfully solve the PoW puzzle receive a reward of Ether. As Ethereum does not have a central issuer of Ethers, this is also how new Ethers are created.

What can Ethereum be used for?

Until the advent of Ethereum, it was challenging to develop new Dapps. But thanks to Ethereum, developers can build and deploy all sorts of decentralized services.

While it seems like the only actual real use case to date has been issuing ICO's deployed with Ethereum smart contracts, there are potentially thousands of other applications that could disrupt hundreds of industries like finance, academia, real estate, insurance, healthcare and the public sector.

While we will only include a few examples in this guide, this is just the beginning. Potentially, any intermediary type service in the real world today could be redesigned using Ethereum.

The white paper splits use cases into three main categories.

- Financial applications - "This includes sub-currencies, financial derivatives, hedging contracts, savings wallets, wills, and ultimately even some classes of full-scale employment contracts."
- Semi-financial applications - "where money is involved but there is also a heavy non-monetary side to what is being done; a perfect example is self-enforcing bounties for solutions to computational problems."
- Non financial applications - "applications such as online voting and decentralized governance that are not financial at all."

Token Systems

These have many applications such as

- Sub-currencies representing assets such as USD, gold, company stocks
- Individual tokens representing smart property, secure unforgeable coupons, and even
- Token systems with no ties to conventional value at all, used as point systems for incentivization (reward systems)

Financial Derivatives

The trading of financial derivatives is currently quite an involved process with paper and computer document based contracts being sent back and forth between parties. Ethereum smart contracts automate the processes involved in derivatives trading by automatically executing the terms of a contract when certain conditions are met. With blockchain enabled smart contracts derivative trading far more secure and efficient.

Identity and Reputation systems

Identity creation and management applications can enable individuals or governments to manage digital identities with unprecedented privacy and safety. They can also increase the control individuals have over their identities. Imagine - digital identities, passports, e-residency, birth certificates, wedding certificates and more stored on a blockchain instead

of company or government servers.

Decentralized File Storage

Decentralized cloud storage networks that enable users to transfer and share data without reliance on a third-party storage provider. By removing the central controls, many of the traditional problems like data failures and outages, security and privacy breaches and high user costs can be avoided. Users also gain control over their data.

Decentralized Autonomous Organization (DAO)

Ethereum can potentially be used to build all sorts of different decentralized applications. This one might just be the craziest of them all.

Even an entire organization can be decentralized.

Welcome to the world of decentralized autonomous organizations (DAOs). A DAO is a decentralized organization that has no leader and is purely run by smart contract code. Instead of the rules and enforcement of these rules being carried out by people, rules are determined and enforced by code. There is no need for employees to govern or documents or any centralized control. DAO's leverage smart contracts on the Ethereum blockchain so that anyone, anywhere in the world can be empowered to participate. In exchange for their early help, participants receive DAO tokens

which represent ownership in the DAO and the right to vote on proposals for the funding of Ethereum blockchain applications.

How is a DAO created?

In a nutshell, the process is as follows. A group of people writes a bunch of code (smart contracts) that will run the organization. There is an initial funding period where people purchase tokens that represent ownership - ICO, Initial Coin Offering. After funding, the DAO begins to operate, and proposals are made and voted on about how the money should get spent.

You made it

Congratulations, that's it, you've reached the end! We hope this guide has provided the clarity you need to move forward with your learning.

You should now have a solid conceptual understanding of Ethereum and its important place in the evolution of blockchain based decentralized applications. The future is exciting. Maybe you can develop a Dapp on Ethereum that will change the world.

BR.

About Blockchain Review

The Blockchain Review provides curated insights from industry insiders on cryptocurrency and blockchain technology, and how it's impacting business and society. Find simple and easy to understand advice for founders, developers, and investors, on how to startup, grow, and succeed in a changing world shaped by emerging technology and innovation.

Visit www.blockchainreview.io

CARDANO – ADA

- PoS Blockchain platform aimed to allow “changemakers, innovators and visionaries” to bring about positive global change.
- At 30/11/2020:

Price:	$0,164119 USD
24h Vol:	$1.832.375.465 USD
Market Cap:	$5.085.480.428 USD
Market Rank:	#8

What is Cardano – ADA

- ADA: digital coin to store or exchange value
- Cardano: decentralized blockchain network based on scientific and mathematical principles, used also to build smart contracts and create DeAPPs and protocols.

Who created Cardano?

- Charles Hoskinson – Ethereum co-founder – he wanted a more standardized and scalable blockchain.
- Jeremy Wood – former co-worker at Ehereum – he was looking for a smart contracts platform

Thanks to their mathematics and scientific background, they decided to work together in the Cardano –ADA project in order to reach their goal

Project goals:

- Provably secure blockchain less prone to attacks
- Separation of accounting and computational layers
- Creation of a Secure voting mechanism for token holders
- Infinitely scalable consensus mechanism

Cardano blockchain architecture

Two core components:

- The Cardano Settlement Layer (CSL)
  - Account unit
  - Where token holders instantaneously exchange ADA with minimal transaction fee
- The Cardano Computational Layer (CCL)
  A set of protocol designed to
  - Run smart contracts
  - Ensure security and compliance
  - Allow blacklisting and identity recognition

Open source code written using Haskell

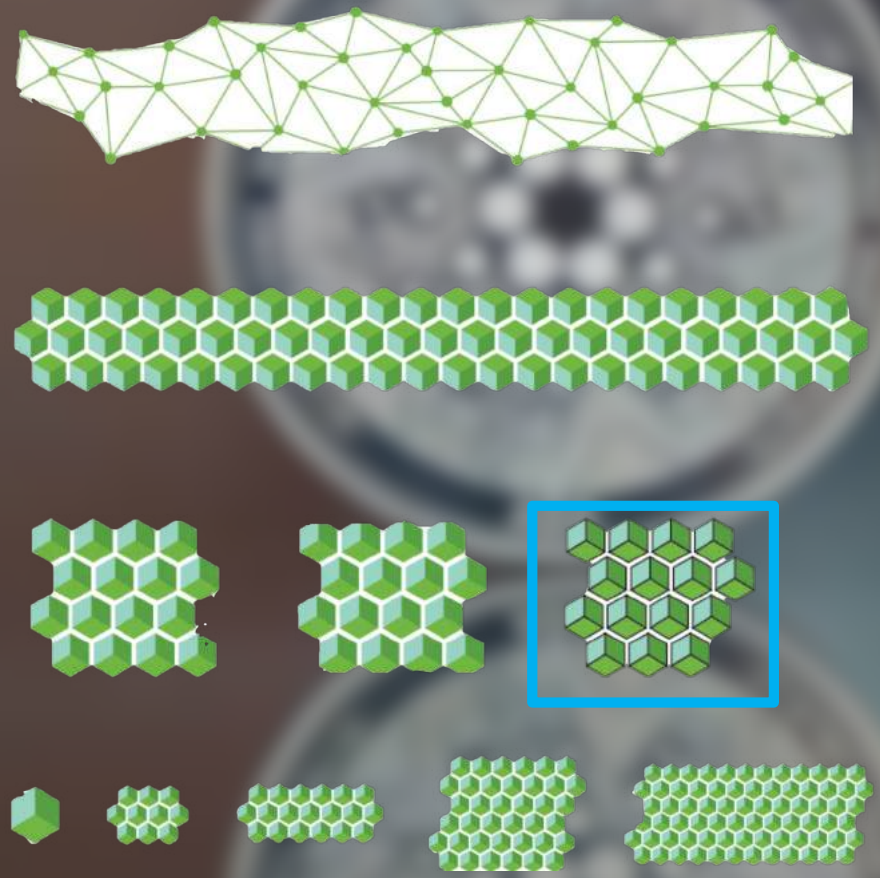
Ouroboros: Cardano's PoS blockchain protocol

A consensus mechanism for:

- Secure and instantaneous ADA transactions
- Ensure the safety of smart contracts
- Reward token holders staking ADA to the network, enhancing network consensus

Ouroboros mechanism:

- Few nodes selected to mine new blocks, the Slot leaders
- The blockchain is splitted in slots, called epochs
- Slot leaders can either choose on which epoch to mine, or subpartitionate one
- An epoch can be partitioned several times → infinite scalability potential



Ouroboros peculiarity: Mathematical security

Other blockchains claim random selection of their block validators, with no evidence of such randomization.

- Provable random validator selection →
  - Fair chances to block mining for token holders who stake ADA to the Cardano network (and get a reward)
  - No necessity of high computational power (as for PoW) blockchain network
  - Objectively fair staking model

Dedalus: Cardano wallet for ADA cryptocurrency

- Blockchain node → control over user personal funds
  high transparency over the Cardano blockchain
- Chance to stake by token holders & get reward:
  - For ADA delegation
  - For running staking pool within the wallet

  → Stake holders cryptocurrency reward
  → Network supporting

Uses of ADA Cardano:

- Transfer value as for ETH or BTC
- Keep the system safe and secure: stake holders help in transaction validation process and get rewarded for that
- Right to vote: changes and developments are proposed to the Cardano blockchain, stake holders can vote on those
- In the FUTURE: run smart contracts and application on a decentralised blockchain

Is Cardano really better than Ethereum?

Similar goal: be world's primary decentralized blockchain platform for build new tools and protocols

	Cardano	Ethereum
Start date	September, 2017	January, 2014
Figurehead/Leader	Charles Hoskinson	Vitalik Buterin
Consensus mechanism	PoS	PoW (moving to PoS)
Programming Language	Haskell	Solidity
Architecture	2-layers	1-layer

Cardano's Roadmap

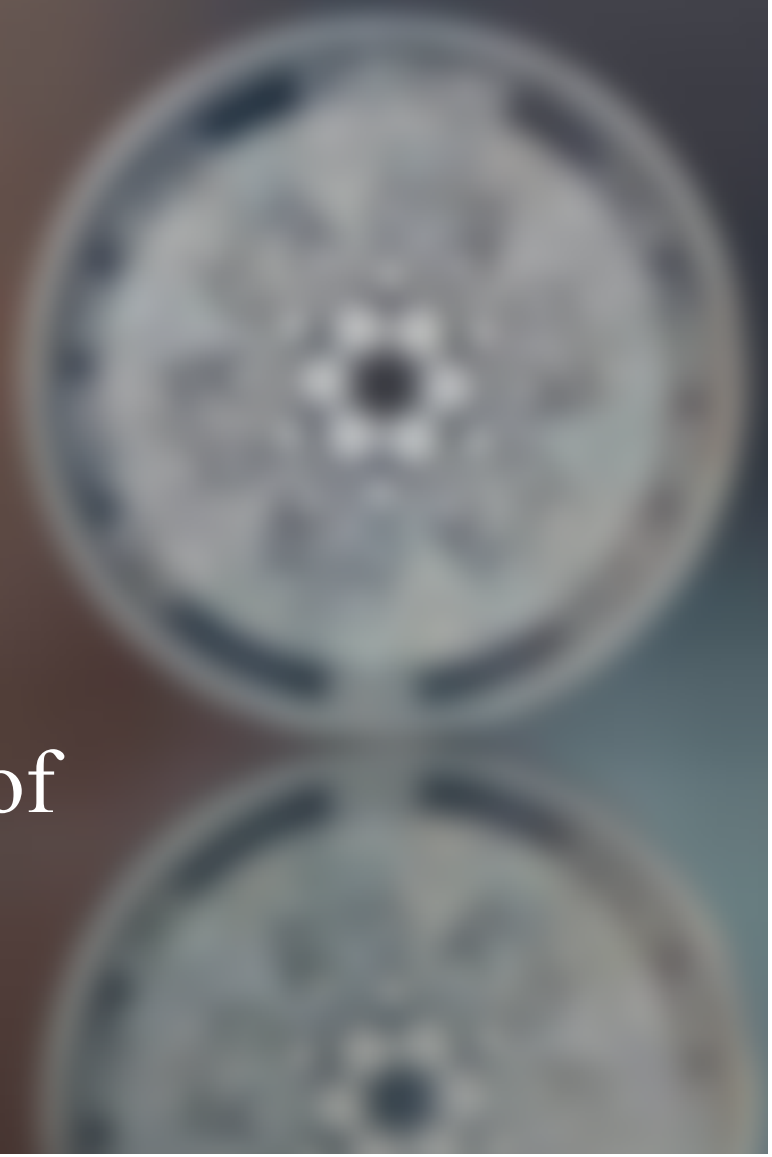
5 different phases have been identified. Currently, Cardano is past the Shelley stage, working on the latter half of its phases

- Byron – Architecture foundation & functionality tests
- Shelley –Cardano mainnet launch & blockchain network decentralization
- Goguen –Smart contract platform implementation (decentralized applications building)
- Basho – Scaling: optimization & improved performance.
- Voltaire –Treasury and voting systems.

Supply

- Total supply: 45 Billions ADA
- Initial sale: roughly 26 Billions ADA
- Currently: 31,1 Billions ADA

For more advanced information, you might check the white paper of the project



Thanks for the attention!

Dogecoin Cash Whitepaper

Introduction to the Dogecoin Ecosystem

History of Dogecoin

Dogecoin (DOGE) is a true phenomenon in the crypto ecosystem. What started as a joke cryptocurrency turned out to be one of the most popular cryptocurrencies of all time. What's even more impressive is that the popularity of Dogecoin doesn't seem to be fading away - on the contrary, Dogecoin seems to be getting more and more popular with every passing day.

The history of Dogecoin began in 2013, when two blockchain developers Billy Markus and Jackson Palmer decided to create a new cryptocurrency that will be able to reach a wider popularity than Bitcoin.

At the time, Bitcoin was still only known but a small niche of computer experts. Palmer and Markus decided to use one of the most popular memes at the time - the yellow shiba dog - as the face of the new cryptocurrency. Using a meme wasn't just a joke - it was also intended to make Dogecoin appear more user-friendly to the general public and to people who were scared away from crypto because of the overly-technical and high-brow presentation of Bitcoin and other cryptocurrencies.

Dogecoin started as part joke, part an experiment. But DOGE was more than just a meme. Palmer and Markus were very experienced software developers, previously working in industry leaders such as IBM and Adobe. Because of their experience, Dogecoin was from the start a project built on solid technical fundamentals.

The combination of friendly and low-brow image and strong technologic fundamentals was enough to cause a massive success of Dogecoin. A passionate community quickly erupted around DOGE, and the thousands of memes made by Dogecoin supporters helped to establish DOGE as the most renown and beloved coins in the crypto ecosystem.

How Dogecoin Became the Beloved Coin of the Crypto Community

DOGE is currently ranked as the 5th most valuable cryptocurrency in the world by market capitalization. The Dogecoin market is worth an astounding $50,000,000,000 (50 billion).

But in the case of DOGE, numbers are not everything. The popularity of Dogecoin surpasses even its huge capitalization. When it comes to how widely known DOGE is, the yellow dog coin is definitely one of the most popular cryptocurrencies, only surpassed by Bitcoin and Ethereum.

It seems that everyone loves Dogecoin. From ordinary people to celebrities like Snoop Dog, Gene Simmons, Kevin Jonas and Lil Yachty - the list of famous DOGE supporters gets longer every day. And when a celebrity joins the Dogecoin holders club, millions of fans follow.

Elon Musk is perhaps the greatest Dogecoin fanatic in the world. The Tesla founder and CEO has expressed his love for DOGE on Twitter numerous times, each time making the Dogecoin price skyrocket. Musk has even jokingly suggested that "Dogecoin might become the currency of Earth in the future".

The effect Elon Musk has on the popularity of DOGE cannot be underestimated. In one of the most spectacular cases, a series of tweets posted by Musk managed to pump the dogecoin price by 800%.

Having a billionaire celebrity ambassador is a tremendous added value for Dogecoin. Musk constantly keeps giving DOGE free publicity.

Recently, a single tweet from Musk managed to make the Dogecoin price surge by 35%.

On April 1st, Musk posted a tweet saying “SpaceX is going to put a literal Dogecoin on the literal moon”. Was it only an April’s Fools joke, or is Elon Musk actually planning to use SpaceX to launch DOGE into space? For the Dogecoin community it didn’t matter, as the DOGE price erupted by 35% within minutes.

One thing is certain: it wasn’t the last time Musk promoted DOGE. The popularity of Dogecoin is increasing, and so does its price. But as Dogecoin gets more popular, some problems with the outdated technical fundamentals of DOGE are becoming more apparent.

The Problems With Dogecoin

Proof of Work: Inefficient and Bad for Our Planet

Doge is a funny meme, but the way in which Dogecoin affects the ecosystem of our planet is not funny at all. The root of the problem is the fact that DOGE is technologically fundated on the Proof of Work algorithm: an outdated and ineffective type of blockchain infrastructure.

Proof of Work (PoW) basically means that mining new coins and verifying transactions requires solving extremely complex math equations. These equations are so complicated that the hardware required to process them consumes insane amounts of energy. It's not a joke, it's not a meme, and it's not funny at all - for example, the Bitcoin network built on the Proof of Work algorithm consumes more electricity than entire countries.

So why do cryptocurrencies like DOGE and BTC still utilize the Proof of Work algorithm? The reason is simple: it's practically impossible to change the core of a network infrastructure of an already existing blockchain. The only way to move away from the energy-devouring PoW algorithm is to build a new cryptocurrency from scratch.

Inflation? In My DOGE? It's More Likely Than You Think

"I don't care about the environment, I just want to see numbers go up!" Sure, nobody has the obligation to care about the energy cost of the cryptocurrency they use. But even if your only focus is profit, Dogecoin still has some very serious structural problems that can't be ignored.

The most important problem with DOGE from the purely financial standpoint is the fact that Dogecoin is not a deflationary currency like Bitcoin. It's the other way - DOGE is susceptible to inflation just like fiat currencies such as dollar or euro.

The inflation rate of Dogecoin is not as bad as the inflation rate of fiat currency. But the supply is still unlimited - new DOGE coins will be produced over time, indefinitely. The complete lack of inflation is one of the main reasons that made Bitcoin so popular, and makes it a tremendous store of value. Unfortunately, Dogecoin lacks this aspect and is more similar to fiat than to Bitcoin when it comes to inflation.

DeFi: The Future of Blockchain

The problems with inflation and Proof of Work algorithm are just two symptoms of a very significant problem with DOGE. Simply put, despite its massive popularity, Dogecoin is now a rather outdated cryptocurrency on a technologic level. The crypto ecosystem has evolved tremendously ever since DOGE was created. To put it bluntly, Dogecoin is a 2013 cryptocurrency. But we're not in 2013 anymore - we're in the age of DeFi.

DeFi (Decentralized Finance) is a collective term referring to a vast array of financial solutions utilizing blockchain technology and smart contracts. DeFi projects are decentralized, secure and private, similar to traditional cryptocurrencies like Bitcoin or Dogecoin. But while BTC and DOGE are simple assets which can only be sent, received or stored, DeFi tokens are much more advanced and have a much greater potential.

The name Decentralized Finance explains the core idea of DeFi very clearly. Decentralized Finance means reinventing traditional financial mechanisms - such as banking, lending or insurance - and vastly improving their efficiency and security through the power of decentralized technologies like blockchain and smart contracts.

DeFi revolutionized the crypto ecosystem, and it's not going away. However, Dogecoin developers have completely ignored Decentralized Finance. Why exactly is DeFi so popular, and why is are new coins more adapted to the DeFi age what the community needs?

Decentralized Exchanges: The Next Generation of Buying and Selling Crypto

Decentralized exchanges are the most popular DeFi solutions. Decentralized exchanges (also called DEXs) allow blockchain investors to trade a large number of digital assets in a fully decentralized and automated way.

DEXs quickly became extremely popular among crypto enthusiasts due to their low fees and high transaction speed. Decentralized exchanges like Uniswap, SushiSwap and PancakeSwap are used by millions of DeFi enthusiasts every day, and the volume of digital assets traded on DEXs constantly keeps growing.

Decentralized exchanges are fast, cheap, reliable and secure. They also enable a much higher level of anonymity and privacy than centralized exchanges like Coinbase or Gemini.

There's no denying that DEXs have become a vital part of the blockchain ecosystem. But Dogecoin is not available on decentralized exchanges - people still have to use centralized marketplaces to buy it. It's one of the many examples of the fact that Dogecoin failed to adapt to the DeFi age.

DeFi Passive Income: Staking and Yield Farming

Decentralized exchanges like Uniswap were among the first DeFi solutions on the market. But DEXs alone were not the only reason for the massive eruption in the popularity of decentralized finance.

The main reason why crypto enthusiasts fell in love with DeFi is the ease in which decentralized finance provides people with various passive income earning opportunities.

When it comes to traditional cryptocurrencies like Bitcoin or Dogecoin, the only way to profit from them is either to use them for short term speculation and day trading (which is extremely risky) or long term holding the coins in your wallet and hoping the price will keep increasing (which is definitely profitable - but it can take years before you see any real gains).

With DeFi, you can put your coins to work. The most popular methods of gaining passive income with decentralized finance are staking and yield farming.

Staking can be considered a more cost-efficient alternative to mining. With staking, consuming massive quantities of energy is not required for a blockchain to work. The users can stake their assets instead, which simply means that they can earn rewards just for holding coins at a specified address for a certain amount of time.

Yield Farming is very similar to staking on a technical level, but platforms utilizing yield farming are more passive income oriented than

projects which only utilize staking. Because of that, yield farming usually allows the user to enjoy a higher rate of interest than staking.

Binance Smart Chain & PancakeSwap: Perfect Environment for DeFi Projects

Originally, all DeFi projects were launched as ERC-20 tokens on the Ethereum network. In fact, the smart contracts technology introduced in Ethereum was what made DeFi possible in the first place.

However, the massive surge in the popularity of decentralized finance caused many problems to appear. Simply put, Ethereum wasn't ready for the tremendous volume of assets traded on decentralized exchanges every minute, which resulted in greatly increased transaction times and fees.

Problems with Ethereum caused the development of new blockchains, specifically designed to be a perfect environment for DeFi platforms. The best example of a DeFi-oriented blockchain is Binance Smart Chain. Decentralized exchanges such as PancakeSwap, built on the Binance Smart Chain, allow people to trade crypto with much lower fees and transaction times than DEXs operating on traditional blockchains.

Dogecoin Cash: A Better Dogecoin for the DeFi Age

Dogecoin Cash (DOG) is designed to solve all the problems of Dogecoin, and to finally make the funny yellow dog stop living in the past and enter the age of decentralized finance.

Dogecoin Cash removes all the problems of Dogecoin, such as utilizing the outdated Proof of Work algorithm and inflation, and vastly improves the functionality of DOGE by adding a large number of DeFi features.

Imagine a coin which is inflation-free, doesn't consume a lot of energy, and that can be staked to provide you with passive income. That's exactly what Dogecoin Cash is.

Dual-Chain Solution

Ethereum is a great blockchain and one of the best projects in the crypto ecosystem. There would be no DeFi without ETH. But Ethereum has its limits, and it's essential to have freedom of choice between different blockchains.

Instead of limiting itself to one blockchain, Dogecoin Cash utilizes a dual-chain solution. DOG empowers you to choose between Ethereum and Binance Smart Chain, as Dogecoin Cash is available on both of these networks. In other words, DOG exists both as an ERC-20 and BEP-20 token.

If you like Ethereum and Uniswap, you can use the ERC-20 DOG. If you prefer Binance Smart Chain and PancakeSwap, you can use the BEP-20 DOG. Dogecoin Cash doesn’t force you to choose - it respects your freedom.

Staking

The old-school Dogecoin (DOGE) doesn't include any passive income opportunities. Dogecoin Cash (DOG), on the other hand, is fully equipped to answer the demands of investors used to earning passive income with DeFi.

You cannot stake Dogecoin, but you can stake Dogecoin Cash. Staking is as simple as sending some DOG to a certain address, and keeping it locked there for a certain period of time. As long as you don't spend your Dogecoin Cash, you will be able to earn passive income in the form of staking rewards.

Fixed Supply

Dogecoin Cash is fully immune to inflation. There is no DOG money printer. The Dogecoin Cash supply is strictly limited and will never increase. In this manner, Dogecoin Cash is very similar to Bitcoin - the supply will only become smaller over the time, making the price increase as the demand grows.

Inflation is the biggest problem with fiat currencies like the dollar and euro, and one of the main reasons why cryptocurrencies were invented in the first place. Inflation is bad, and a good cryptocurrency shouldn’t have it. That’s why by removing inflation, Dogecoin Cash solved one of the main problems of Dogecoin.

Proof of Content

Instead of PoW mining, Dogecoin Cash introduces a new system of minting coins called Proof of Content (PoC). Proof of Content allows you to earn DOG by posting on Tipestry - an innovative decentralized content platform.

Instead of polluting the environment with massive energy consumption, Proof of Content creates an incentive to produce content in order to earn DOG. Tipestry allows anyone, anywhere in the world, to earn Dogecoin Cash by creating any type of content, which is especially valuable during a pandemic with people stuck at home.

Dogecoin was always known as the currency of the people, but mining DOGE required specialized hardware and was only accessible to a small group of IT experts. Dogecoin Cash removes inefficient PoW mining altogether, instead giving you the ability to earn DOG by simply posting memes!

Summary

Everyone loves Dogecoin memes, and everyone loves DeFi.

Dogecoin Cash uses DeFi to create a new, better coin more suitable for an age in which decentralized finance solutions are becoming increasingly dominant in the blockchain ecosystem.

Take part in the next step of crypto evolution. Visit these channels to learn more:

Website: https://www.dogecoincash.org/
DOG Staking Platform: https://stake.dog/
DOGE and DOG Tipping Website: https://tipestry.com/
Pancake Swap: https://exchange.pancakeswap.finance/#/swap?inputCurrency=0x9596a56c73ca6f50cbd05cb8d85865f75659dc78
Uniswap: https://v2.info.uniswap.org/token/0xb91b9ae0c65c7b3f066f5e92db0156af2decfa92