# JeChain

*JeChain Whitepaper v0.1.1*

*By: Nguyen Phu Minh*

*Email: nguyenphuminh09876543@gmail.com*

Table of Contents:

- Introduction
- JeChain as a payment system
  - Basic information
  - Transactions
  - Blocks
  - Consensus
  - State
  - Network
- JeChain as an application platform
- Conclusion
- Resources

## What is JeChain and what problem is it solving?

In the modern era, we are mostly using centralized currencies systems, which are not all that safe like many would think, since it gives the power to third-party providers, meaning they have full power over your assets, leading to the fact that they can take your money away, print more money however they like, whenever they like, or lock your account without your approval or knowledge whatsoever. There is also this problem that starts right from the system itself - because it is a centralized entity if the system is exploited/attacked, which is possible periodically, then all of your assets will be at risk. While many benefits can be gained from using centralized systems, a decentralized system should also be favorable, since it gives users freedom while still being safe, thanks to cryptographic algorithms. That's why Bitcoin - the first decentralized cashing system utilizing the use of blockchains and the proof-of-work consensus mechanism was released. It makes it so that no one can have full control over the whole network, bringing fairness while still being safe due to the use of cryptography, proof-of-work consensus, and decentralized network. Following its philosophies, JeChain is built to be a decentralized peer-to-peer currency network but pushes it a little bit further - a decentralized application platform.

## A payment system

### Basic information

People will access the network as `nodes`, nodes can send messages to each other, all nodes will hold an agreed ledger containing payment history, from that, we can identify one's balance. The ledger will be represented as multiple "blocks" chained together, thus the term `blockchain` and the name JeChain.

### Transactions

JeChain's transactions will contain basic information like the amount of money to be sent and addresses from the sender and the recipient.

The problem is, in a decentralized network, it's everyone's game, people can fake transactions from other users to enrich themselves. To prevent this, we have two newly added properties - a cryptographic signature and a timestamp.

Now, how do these two props solve the problems of decentralized transactions? First, let's look at the problem, people can create multiple transactions that take money away from others, right? So we just need a randomly generated key pair, with the first one being the private key used to generate a signature based on information from the transaction, and the public key which can be used for verifying the signature and can act as a public address. With this system, people can no longer create transactions from other addresses if they don't have their private key. But, there is still one problem, what if people take a signed transaction and continuously recreate it since it would still be available, right? We can fix this by adding a `nonce`, but JeChain takes the approach of using timestamps, bringing several benefits. First, with this problem specifically, it makes the signature come out differently every time since time always changes. Therefore, we can just discard any new transactions with duplicated timestamps from one address. Second, it can be used for knowing when the transaction was made.

### Blocks

Blocks are entities that keep the blockchain immutable, they contain information like their creation timestamp, transactions, a cryptographic hash generated from a block's information (block header), and the hash of the previous block which is also in the block header. If a block is changed, its hash will be changed, and the following block's hash will also be changed, and the following of that will also be changed. Repeat this process and you can easily recognize if one's chain is changed or not.

Timestamps also play an important role, it records public actions from nodes.

### Genesis block

Genesis block is the first block in the blockchain. It might contain a transaction acting as an initial coin release (not to be mistaken with ICOs).

This initial coin release can be used as an incentive to support the project.

### Consensus

But the thing is, nodes can cheat any time, not just attackers, even normal users. So we need a way to make nodes not become attackers. We can solve this using a consensus mechanism called proof-of-work. Basically, you add a `nonce` value into one block. You increment `nonce` by 1 continuously until the block's hash starts with a required amount of zeros. This process is often known as `mining`. Blocks can now no longer be created randomly, you would need proof of your work to make a block valid. "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." - said the Bitcoin whitepaper, which has brought out one of the key points from proof-of-work consensus. But, what keeps the honest nodes mining? Why would they mine for us? This can be simply solved with rewards, if a miner successfully produces a valid block, he/she will be awarded with minted (newly released) coins, so this creates an effect that keeps the nodes honest - if they cheat, they wouldn't get money, which is less beneficial than if they did not. This system is also decentralized, since the hashing function is completely random, and each block from each miner will have a different mint transaction pointing to their address.

The required amount of zeros is determined by a variable called `difficulty`, which rises if miners are mining too fast, falls if miners are mining slower than before.

The network will be in the hand of attackers if they have more than approximately 51% computational power of the whole network, which is unlikely.

Proof-of-work consensus also makes it hard to change data from the past: Since one little change in one block's data can make the hash of that block and every next block change, they would have to recompute the whole chain from that point, which is practically impossible due to the huge energy cost and how time-consuming the process could ever be.

### State

State is like a result created by the blockchain where you can get accounts' information like balance or used timestamps from. This will play a big role in making JeChain a decentralized application platform.

JeChain follows an account-based model, which is better for smart contracts (we will talk about this in the next part of this document) and is also used by Ethereum, while Bitcoin and many other networks use the UTxO (unspent transactions' output) model. This model works like a key-value database, with the key being an address, and the value is its information.

Every time a new block is submitted, the state will be changed according to transactions from the block.

Technically, you can understand that state is the only source of information that users need, all of the technology mentioned in this article is only used to bring truthy, accepted state to a consensus between nodes, which is not achieved in traditional decentralized peer-to-peer networks.

### Network

In the JeChain network, any people, appearing as nodes, can broadcast their transactions to all nodes, which are then put into a mempool called transaction pool, miners then grab transactions from the pool, add them into a new block and start mining. Mined transactions are then removed from the pool. When there is a winner, all nodes will update the chain state according to the newest block.

The problem with a decentralized network is that it's relatively slow compared to a centralized network, this is due to a period of time for all nodes to reach to consensus, so it is always vulnerable to DDoS attacks (to the blockchain storage itself, not to nodes as spamming protocol messages, since attacking every single node is practically impossible). To prevent attacks from happening, each transaction must add a fee called "gas fee" so no one can spam transactions continuously, infinitely. Also, the transaction fees can somewhat provide energy costs for miners - a major benefit gained from these fees. This also born a whole new use of this "gas fee", miners will always pick transactions with a higher gas fee, since message's size in a decentralized network is limited, so something more beneficial is always more favorable.

## An application platform

### Smart contracts

Smart contract is the key point to achieving decentralized applications on a blockchain network. It is just a piece of code that is attached to an address (this kind of address is also called "contract address"). Every time someone creates a transaction pointing to a contract address, the contract will be executed. This opens an opportunity for decentralized applications (also called "dapps"), using the blockchain as a database (which is the chain state mentioned before).

There is this one problem though, if the programming language used for development is turing-complete, people can just create infinite loops to halt the network. Furthermore, people can store data on the blockchain continuously, making nodes' storage full. There are many solutions to this, one is making the language turing-incomplete, limiting dapp's functionalities, other is what Jechain is doing. JeChain has fixed this with a similar approach with Ethereum - using gas fees. By making every instruction cost a small amount of money, people can no longer create infinite loops, and storage costs money.

The current version of JeChain is using Jelscript - a small interpreted language to build smart contracts.

### Applications

With the use of smart contracts, people have come up with plenty of revolutionary decentralized technology.

As the Ethereum whitepaper has said, there are generally three types of applications on a network like this:

- Applications of which main focus is on finance. These can be currencies, exchanges, wallets, marketplaces, etc but can be used a different way in that it is decentralized, uncheatable, and acts in a fair manner.
- Applications that are both neutral and finance-related.
- Applications that are not finance-related, like decentralized governance, one example is decentralized autonomous organizations (DAOs) where members of a project would vote in a decentralized style. Another example is decentralized file storage.

## Conclusion

Overall, we have proposed a decentralized peer-to-peer currency network, also a decentralized application platform with the hope of bringing freedom and independence to normal users, not to be reliant on third-party authorities. There is still a long road to go, but JeChain is hoped to be a platform where not only entrepreneurs but developers manage assets and build fair applications.

## Resources

- Github: https://github.com/nguyenphuminh/JeChain
- Jelscript: https://github.com/nguyenphuminh/JeChain/blob/main/CONTRACT.md
- Bitcoin: https://bitcoin.org/
- Ethereum: https://ethereum.org/en/