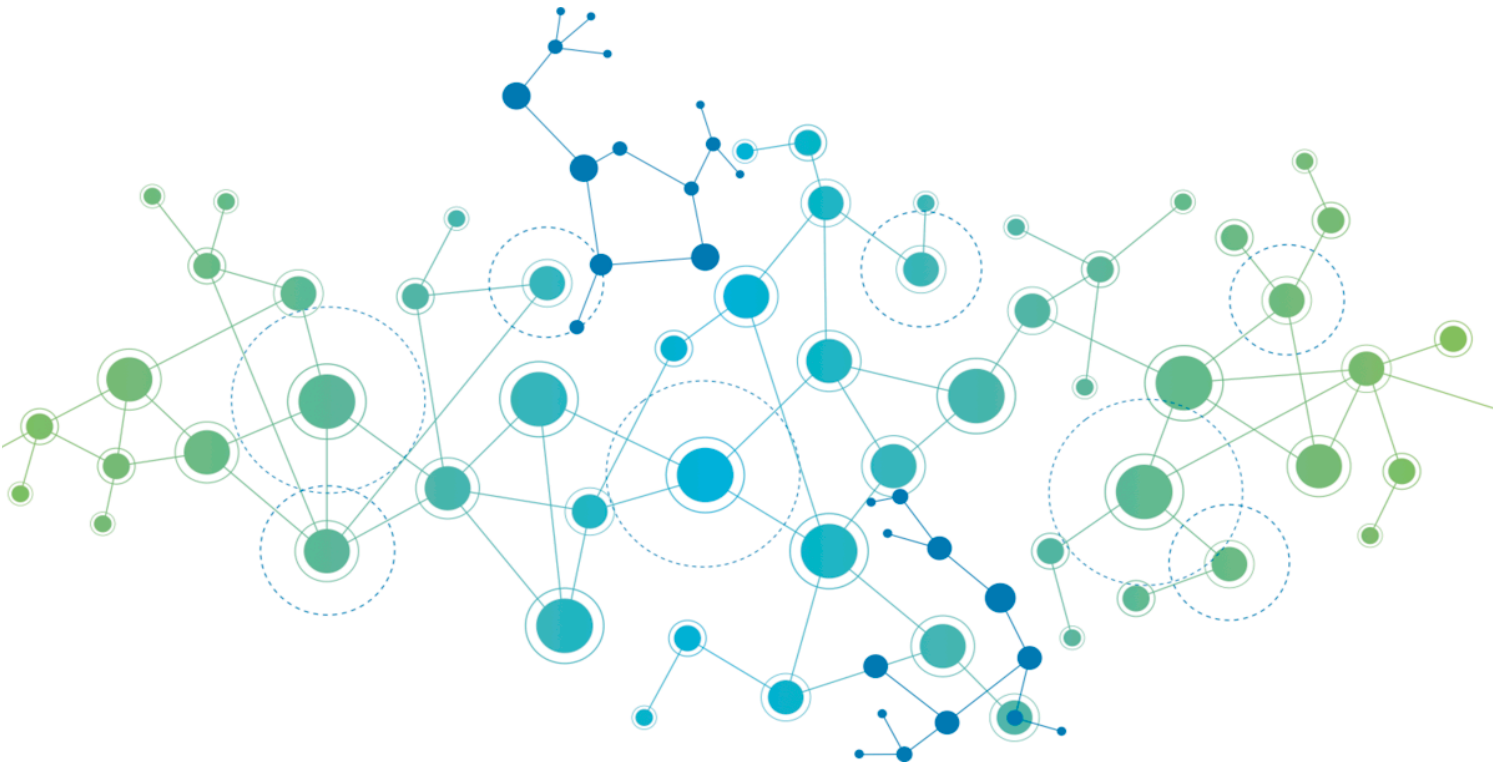
# Fuloos Coin

The New Age Money

Whitepaper Version 1.0



**Abstract**

A lot has changed since 2009 when Fuloos entered the world to transform it forever. A new path was opened which lead to endless possibilities in the world of financial technologies. Fuloos coin is our effort to offer a faster, greener & much convenient digital money which makes transactions as easy as fiat. Fuloos is an open source, global payment network that is fully decentralized without any central authorities. Mathematics secures the network and empowers individuals to control their own finances.

Fuloos comes as a handy bundled solution which would include a coin with very own Blockchain, all ecommerce plugins, CRM & billing solutions, mobile wallets and much more. Fuloos works on open Blockchain & Open API allowing everybody to create their own tools.

| Fuloos Architectural Snapshot | |
|---|---|
| **Description** | **Value** |
| Coin Name | **Fuloos** |
| Coin Ticker | **FLS** |
| Mining Mechanism | **POW + POS Algorithm** |
| Algorithm | **Scrypt** |
| Block Time | **90 Seconds** |
| Block Size | **3 MB** |
| Initial PoW Reward | **1000 FLS** |
| Reward Halving Rate | **10000 Blocks** |
| Total Supply | **100,000,000 FLS** |
| Time Lock | **Absolute** |
| Proof of Work Duration | **312.5 Days** |
| Proof of Stake Rewards | **5% Per Annum** |
| Minimum Stake Age | **24 Hours** |
| Maximum Stake Age | **7 Days** |
| Coin Maturity | **77 Blocks** |
| Stake Mechanism | **In-Wallet, Manual Action with Absolute Time lock** |
| Proof of Stake Fee | **Wallet Service Driven** |
| PoS Penalty | **Wallet Lock Down** |

## Technology

### Encryption

Fuloos uses Scrypt for encrypting blocks on its Blockchain. A password-based key derivation function (password-based KDF) is generally designed to be computationally intensive, so that it takes a relatively long time to compute (say on the order of several hundred milliseconds). Legitimate users only need to perform the function once per operation (e.g., authentication), and so the time required is negligible. However, a brute-force attack would likely need to perform the operation billions of times, at which point the time requirements become significant and, ideally, prohibitive.

Previous password-based KDFs (such as the popular PBKDF2 from RSA Laboratories) have relatively low resource demands, meaning they do not require elaborate hardware or very much memory to perform. They are therefore easily and cheaply implemented in hardware (for instance on an ASIC or even an FPGA). This allows an attacker with sufficient resources to launch a large-scale parallel attack by building hundreds or even thousands of implementations of the algorithm in hardware and having each search a different subset of the key space. This divides the amount of time needed to complete a brute-force attack by the number of implementations available, very possibly bringing it down to a reasonable time frame.

The Scrypt function is designed to hinder such attempts by raising the resource demands of the algorithm. Specifically, the algorithm is designed to use a large amount of memory compared to other password-based KDFs, making the size and the cost of a hardware implementation much more expensive, and therefore limiting the amount of parallelism an attacker can use, for a given amount of financial resources.

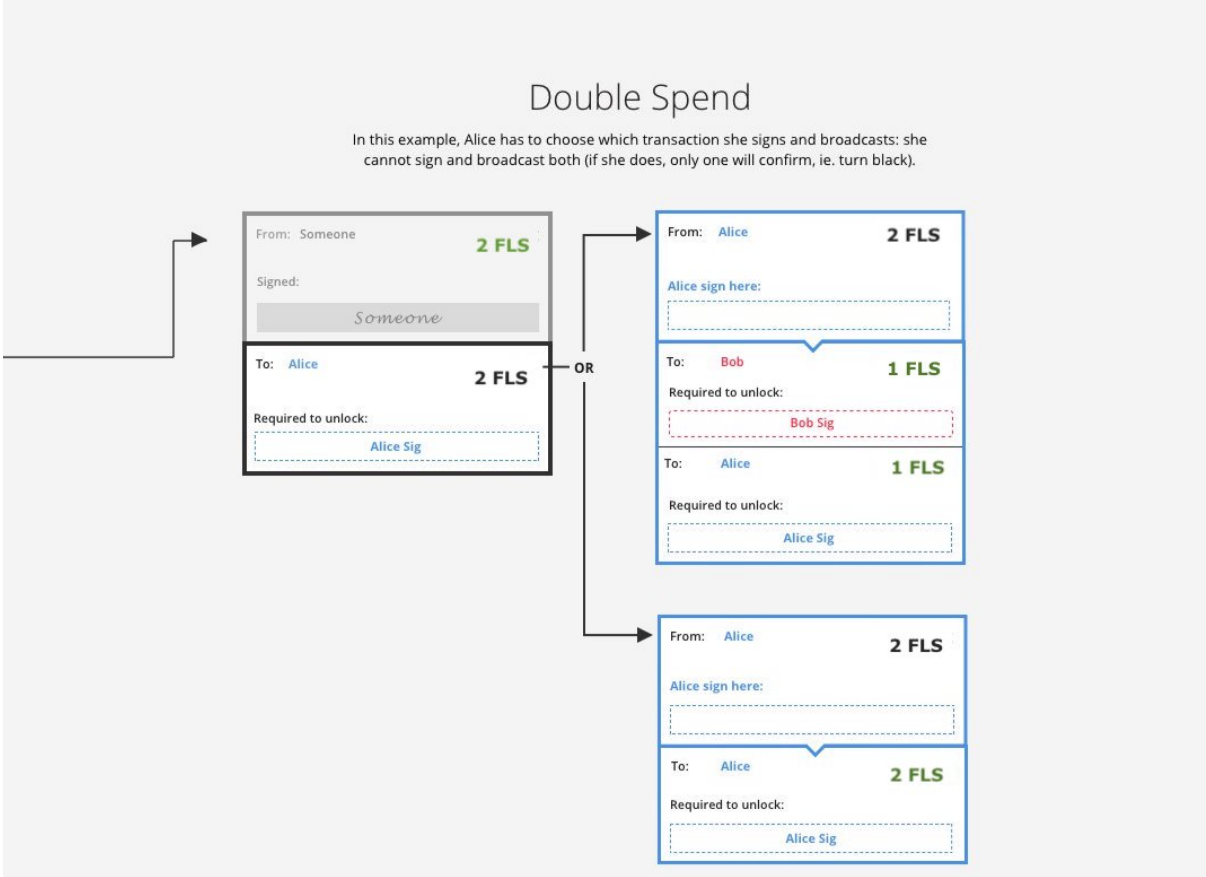
## Mining Mechanism

### Proof of Work

A **proof of work** is a piece of data which is difficult (costly, time-consuming) to produce but easy for others to verify and which satisfies certain requirements. Producing a proof of work can be a random process with low probability so that a lot of trial and error is required *on average* before a valid proof of work is generated. Fuloos uses the Scrypt proof of work system.

Scrypt proofs of work are used in Fuloos for block generation. In order for a block to be accepted by network participants, miners must complete a proof of work which covers all of

the data in the block. The difficulty of this work is adjusted so as to limit the rate at which new blocks can be generated by the network to one every 90 seconds. Due to the very low probability of successful generation, this makes it unpredictable which worker computer in the network will be able to generate the next block.




For a block to be valid it must hash to a value less than the current target; this means that each block indicates that work has been done generating it. Each block contains the hash of the preceding block, thus each block has a chain of blocks that together contain a large amount of work. Changing a block (which can only be done by making a new block containing the same predecessor) requires regenerating all successors and redoing the work they contain. This protects the block chain from tampering.

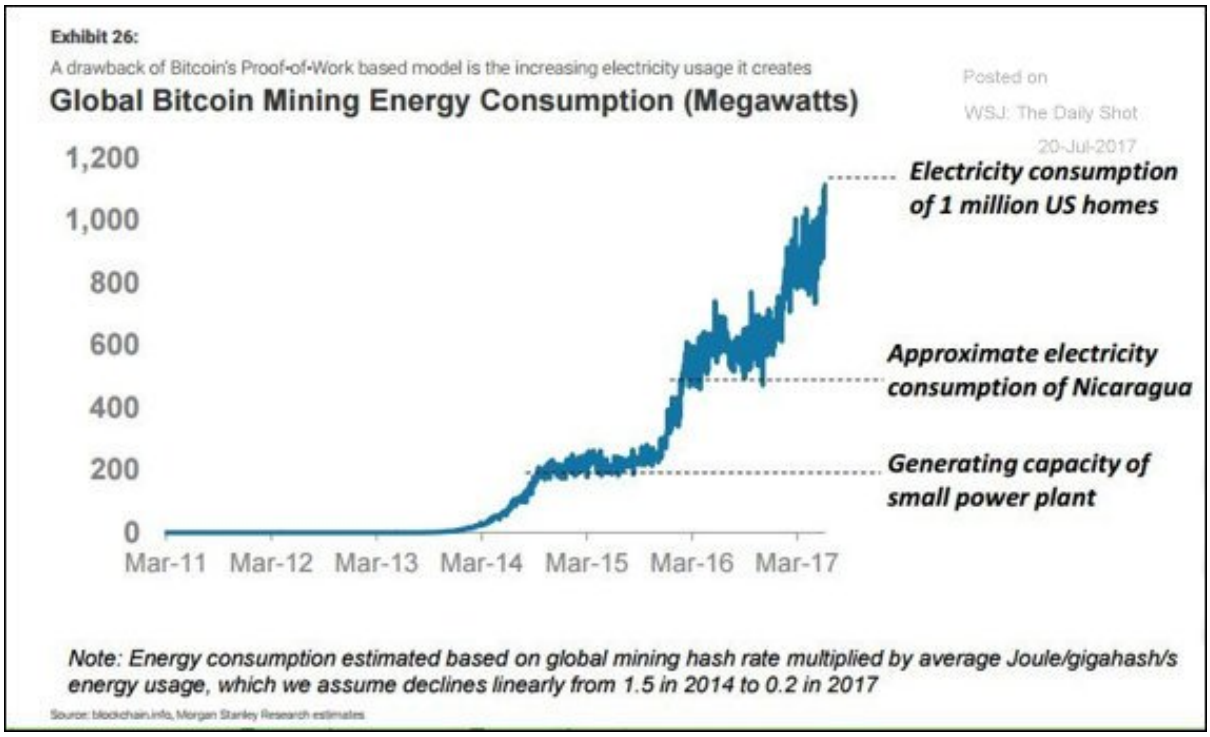
**Proof of Stake**

Proof of Stake is a proposed alternative to Proof of Work. Like proof of work, proof of stake attempts to provide consensus and doublespend prevention It was probably first proposed by QuantumMechanic. With Proof of Work, the probability of mining a block depends on the work done by the miner (e.g. CPU/GPU cycles spent checking hashes). With Proof of Stake, the resource that's compared is the amount of Bitcoin a miner holds - someone holding 1% of the Fuloos can mine 1% of the "Proof of Stake blocks".

A proof-of-stake system might provide increased protection from a malicious attack on the network. Additional protection comes from two sources:

1. Executing an attack would be much more expensive.
2. Reduced incentives for attack. The attacker would need to own a near majority of all Fuloos. Therefore, the attacker suffer severely from his own attack.

When block rewards are produced through txn fees, a proof of stake system would result in lower equilibrium transaction fees. Lower long-run fees would increase the competitiveness of Fuloos relative to alternative payments systems. Intuitively reduced fees are due to vast reductions in the scale of wastage of resources. **Fuloos is trying to avoid the need to consume large quantities of electricity** in order to secure a blockchain (eg. it's estimated that both Bitcoin and Ethereum burn over $1 million worth of electricity and hardware costs per day as part of their consensus mechanism).




Fuloos is designed to be a greener cryptocurrency which does not consume humongous amount power just to validate the transactions. It is a mix of both PoW (Proof of Work) and PoS (Proof of Stake). Perfect ration of this mix is needed to power the Fuloos Blockchain as staking need certain amount of coins to be available to community in order to run the nodes and validate transactions using PoS mechanism. Designed to be PoW Blockchain initially Fuloos is designated to automatically migrate to PoS upon hitting the triggers.
Fuloos would migrate to 100% PoS mining upon reaching block number 300000 which is expected to happen within a year of genesis block mining.

| | |
|---|---|
| Block Time | **90 Secs** |
| Number of Blocks per Day | **960** |
| Number of Blocks per Year | **350400** |
| Proof of Works Mining Duration | **312.5 Days** |

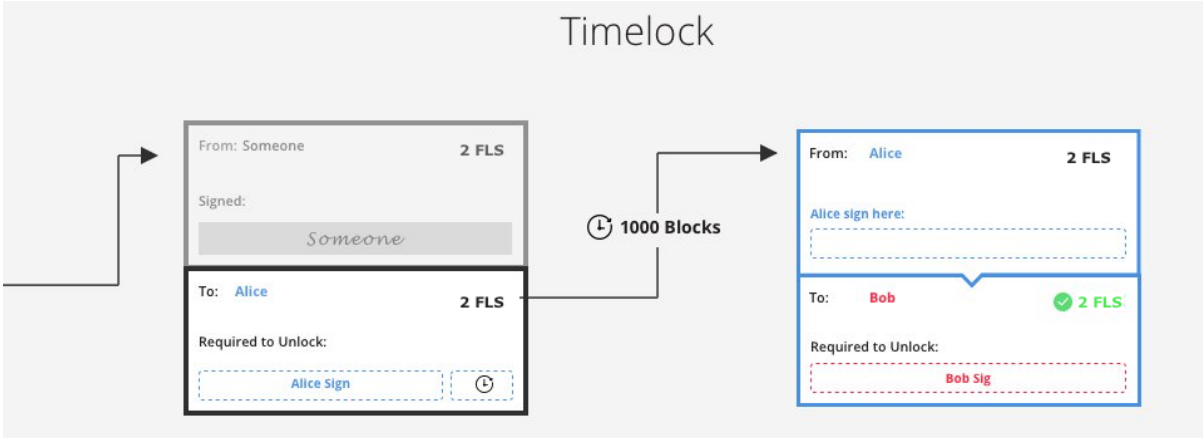
**Rewards**

Fuloos would be mined using PoW mechanism initially as coins needed to stake in PoS need to be generated. Mining rewards starts with 1000 FLS per block and would be halved every 10000 blocks. Upon migration to PoS, rewards would be stripped off and node owners would earn from staking their Fuloos coins. Currently defined staking revenue is 5% per annum, Fuloos also offer in-wallet staking option with minimum 24 hours stake period.

Fuloos is open to community development and anybody can develop and run his own FLS wallet service. All wallets services are free to determine and charge the transaction fees as per their will. Any fees charged by wallet services would be theirs only to keep.

**Time Lock**

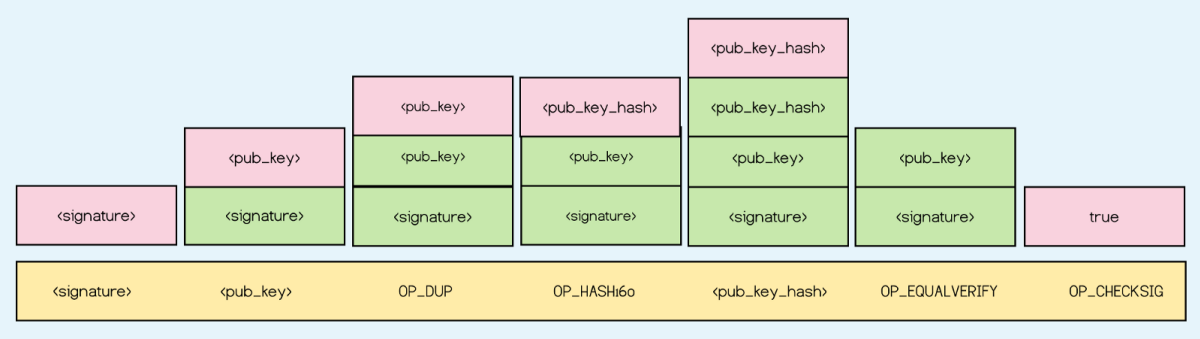
Simple requirement for funds to be locked up until a future date. Blockchains are found to have two different time-locks: relative and absolute. Fuloos uses Absolute Time Lock. It locks a transaction output until a fixed point in time in the future, Time-locks are a requirement for trustless payment channels, and are recommended as they allow for indefinitely open payment channels.




To simplify it with another example Tom sends 2 FLS to Leah. Now Leah can either take the 1 FLS out if she gets the hash from Tom within a predefined time, or Tom will get the funds back automatically after that predefined time has passed. This way regardless of success or failure of the transactions, funds are sent to either of the designated party.

**Replay Attack Protection**

For normal transaction inputs, if the creator of the current transaction can successfully create a ScriptSig signature that uses the right public key for the ScriptPubKey of the transaction output they are attempting to spend, that transaction input is considered valid.




OP_CHECKSIG is script opcode used to verify that the signature for a tx input is valid. OP_CHECKSIG expects two values to be on the stack. These are, in order of stack depth, the public key and the signature of the script. These two values are normally obtained by running the scriptSig script of the transaction input we are attempting to validate. The scriptSig script is deleted after it is run, but the stack is left as is. Then, the scriptPubKey script from the previous transaction output that is now being spent is run, generally concluding in an OP_CHECKSIG.

The standard scriptPubKey checks that the public key (actually a hash of) is a particular value, and that OP_CHECKSIG passes.

**Conclusion**

Fuloos is a digital currency designed for low fees and instant transactions for payments all around the world and merchants that accept cryptocurrency.

------ End Of Document -----