CPSCoin: An Asset Token Building Efficiency within a Peer-to-Peer Electronic Value System

Jagdeep Sidhu, Msc.
Syscoin Core Developer
Blockchain Foundry Inc.
Email: jsidhu@blockchainfoundry.co

Alexander Alexandrovich Alexandrov
Chief Executive Officer
CoinPayments, Inc.
Email: alex@coinpayments.net

Caution: This is a new speculative venture with substantial risk. See section 6 for details.

***Abstract*—CPSCoin introduces a novel implementation of a decentralized marketplace, built on top of the Syscoin platform [Sid], but in a social construct by providing a moderation layer and redemption of other services on the CoinPayments platform by holders of CPSCoins.**

1. Introduction

CPSCoin is an asset built on top of the Syscoin network (derived from Bitcoin[Nak]). CPSCoin provides inherent utility for CoinPayments users through providing a framework for services, which allows users to transact on a user-friendly marketplace within the CoinPayments portal. CPSCoin also allows redemption of current and future CoinPayments services at a fraction of the costs.

1.1. Asset details

CPSCoin will be Syscoin's flagship asset related project. It inherits a few useful features from Syscoin assets that ERC20[But] token implementations do not have. Particularly Z-DAG (0 confirmation transaction capability), staking, and low fees.

1.1.1. Z-DAG. Z-DAG is an innovation created by the Syscoin developers that will be adopted by CPSCoin. It functions across all Syscoin services. Syscoin services are Alias Identities, Certificates, Escrows, Offers, and Assets. All services are controlled via an Alias. The ownership is proven upon spend, based on the ownership of a private key that matches the address set inside of the Alias. So, by transferring from/to Aliases you may easily transact services. Assets are where Z-DAG will be leveraged the most, since Assets can be transferred from/to Aliases without limitation and with instant settlement. Doing this requires ordering of transactions from least dependent to most dependent Asset transfer, so that the state can be built in a deterministic fashion. This helps protect against double spends where an attacker would falsely transfer an Asset more than once, by simply broadcasting multiple transactions to multiple nodes in a short time-span.

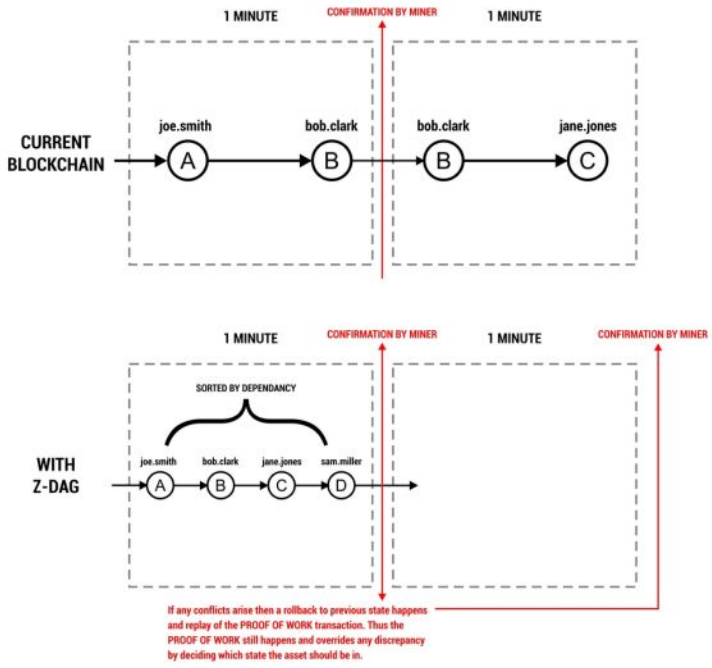

Figure 1: Current blockchain design vs Z-DAG

Figure 1 shows an evident difference in transaction settlement time between current blockchain design, and a blockchain that supports both Z-DAG transactions and Syscoin asset transfers. A graph of transactions with settlement can be represented in the mempool in realtime without a block, while the block provides confirmation and conflict resolution if there are any double-spends.

Verifying-client nodes create a graph of the transactions by looking at the sender/receiver list of Asset transfers. To get a DAG, a circuit detection algorithm is applied (read below Hawick cycle detection) to get rid of the cycles in the graph. Afterwards, the DAG is sorted topologically and then Asset transfer consensus code is processed in sequence,

to form a deterministic state amongst the entire network in consensus.

Since we can verifiably create and process the graph of transactions in a known order (which matches the real order of processed mempool transactions) we can apply state changes (similar to UTXO updates to Bitcoin) to the Asset structure upon mempool inclusion. We save the previous state upon every block, and roll back the previous state prior to every block to ensure that should any discrepancy occur, the Proof-of-Work will override and replay the correct order of events according to the miner.

A User Interface layer will be given real-time notifications shall a conflict arise, thus the point of sale would generally only need to wait 3-5 seconds (statistically the amount of time needed to ensure that the network took notice of 2 double spend transactions conflicting with each other). Syscoin's Masternode layer makes this possible. Every Masternode is typically connected to 25 or more peers and can leverage a high-throughput relay across the network. That means it will take single or double digit network hops on average to reach, from sender to receiver nodes.

1.1.2. Hawick cycle detection. [HJ] Hawick et James were able to detect a circuit in a graph containing edges that start and end at the same vertex, as well as multiple edges connecting the same two vertices. For our type of graph where people can send and receive from to the same person, we require such an algorithm. This is implemented in the Z-DAG functionality: clients verify and extract cycles, and create DAGs in order to process Asset transfers sequentially.

1.1.3. Order of events preservation and conflict resolution. A topological sort is applied upon the DAG, that was created as a result of the Hawick cycle detection and cycle removal from the graph of transactions that represents Asset transfers. A time was added to the Asset structure when it receives a transfer upon mempool.

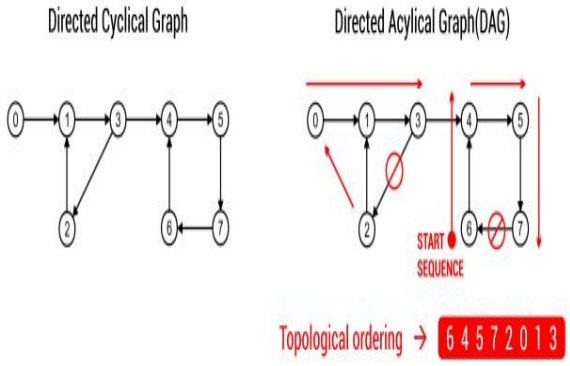

Figure 2: DAG with topological sorting

When creating a block, the miner is tasked with ordering Asset transfers from oldest to newest by time received. A non-enforced 10-second delay that prevents Aliases from sending an Asset within that minimum latency period is applied, to which the user may or may not adhere. This allows the transactions to arrive in order to the miner and have them order based on likely the same view as the rest of the network. If an Asset transfer is detected within the minimum latency period, a conflict state is set within the consensus code. Users can detect that an Asset transfer occurred within the minimum latency period. This would likely result in them waiting for confirmation of fund transfers until the next block. Note that the clients that choose transaction fees of dependent Asset transfers higher than Asset transfers they depend on will not alter the order in which they are put in the block, as the time will require them to be ordered based on first-come-first-serve basis.

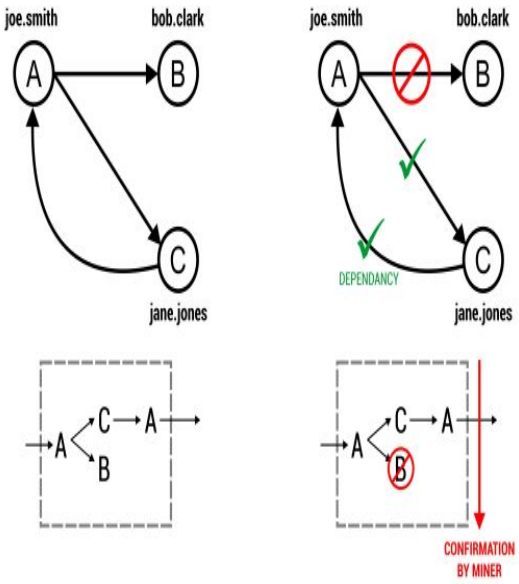

Figure 3: Conflict Resolution within Z-DAG

Figure 2 illustrates the typical cyclical graph that can represent our payment structure of assets between individuals. However, to create a DAG we need to remove cycles and then topologically sort the graph in order of dependence. We can see in this example that 2 circuits exist and removing the edge's 3 to 2 and 7 to 6 allow us to create a DAG with a valid topological sort of 6 4 5 7 2 0 1 3.

As you can see with Figure 3, the payment from A to B would be in this case classified as a double-spend if we assume that the total amount of funds were sent to B and C at the same time. Some parts of the network would receive A to B and others would receive A to C. This would be flagged as a conflicting transaction in real-time. The receivers B and C would wait until the next block resolves any conflicts before providing a service for the payment. Here we see

that the miner saw the payment from A to C first before A to B. Hence, A to B was classified as a double-spend and discarded. Consequently, the payment from C to A is allowed because C has enough funds to send back to A. Should A to B have won, then C to A would have also been discarded if C did not have enough funds to send to A.

1.2. CAP Theorem

The CAP theorem[Bre] states that it is impossible for a distributed data store to simultaneously provide more than two out of the following three guarantees: consistency, availability and partition tolerance. While Bitcoin tries to provide a strong finality guarantee that transactions are settled, mathematically it is not able to do so. However, it is unlikely that a double-spend or re-organization will change the state of someone's balance[Ker]. So if Bitcoin does not solve the CAP theorem, then why do we need to wait for some type of pseudo-settlement finality by waiting a block and sacrificing usability? The answer is you can trade-off some consistency for some availability, which will increase the usefulness for applications desiring point-of-sale.

We can therefor find ways to minimize the effect of the reduced capacity, by providing users with the odds that the network will realize an attempt at a double-spend, and simply revert to the higher statistical settlement guarantee of the Proof-of-Work consensus mechanism. By allowing instant settlements and increased availability, we would need to watch for users simply trying to double-spend or send transactions too quickly, which may cause the miner view to change from the general network view. Using the above conflict resolution mechanism, it becomes possible to do so. Since the DAG will order the dependency graph of the transactions and process in sequence which allows for better availability when it comes to instant state changes, we will therefor aim to share the same CAP constraints as Proof-of-Work, whilst at the same time providing a much more responsive money transfer mechanism that may be used as a transaction processor instead of simply a settlement layer.

1.2.1. Point-of-sale applications. The combination of using Assets with Z-DAG will allow for application where people may provide a service in real-time in exchange for cryptocurrency Asset tokens.

1.3. Offers and decentralized marketplace

We have developed a marketplace where users can securely and reliably buy and sell a variety of items. CoinPayments users will be able to directly create and manage their online stores online through the web-portal or offline in their desktop or mobile wallets.

Users will be able to view listing descriptions, price, geo-location based listings and services (if enabled by the merchant), seller's profiles, and reputation.

Availability of merchants will be filterable in the context of searching for offers. It will also allow for buyers to quickly get in contact with merchants and acquire support for a product in real-time, prior to purchase.

Sorting and filtering will be provided by the Syscoin core blockchain which implements a decentralized mongodb store of all the listing data. This allow for quick and efficient structured queries. Because mongodb supports sharding, it will allow the system to scale horizontally with the amount of users using the system. Since offers can have geo-location information inherited from the users listing them, the system will allow for proximity searches through special geospatial queries.

1.3.1. Digital sales. Certificates may be sold in conjunction with offers to create sales of digital ownership. A certificate may hold private information such as codes or registration keys for some service that are redeemable by the buyer of the offer. Certificates can be automatically transferred to the buyer upon completion of sale.

1.3.2. Auctions. Offers on the CPSCoin platform may be created as an auction with a fixed countdown time, minimum bid, and reserve price. Offers can now be Regular Offers, Timed Auctions, or Timed Auctions with a Buy it Now option

In our auction system, we allow merchants to configure multiple options. They may require a deposit (this imitates how current Real-Estate transactions work, a deposit is often required to show earnestness), a witness or notary, and a reserve price. The escrow system works in conjunction with the offer and identity modules to ensure the correct offer is sold to the winning bidder under authorization of the seller. Because these are smart contracts running through consensus on the blockchain, auctions are just as secure as regular transactions.

1.3.3. Reselling with whitelists. Merchants may leverage a whitelist feature to offer resellers the chance to sell their offers for a commission. This allows drop shipping of goods and services while offering provable sales through the decentralized marketplace. The merchant who created the offer controls the whitelist and can add a discount level on a per entry basis for each reseller. If the merchant sets their offer to private, then end users must purchase the item through one of the participating reseller offers.

1.3.4. Feedback and rating system. Escrows and offers sold through the marketplace offer a convenient way to rate and leave feedback on a per sale basis. For an escrow, one rating is accepted (a number from 1 to 5) to represent a user's satisfaction level with a transaction, with 1 being the least satisfactory and 5 being completely satisfied and recommending the user to others. Ratings and feedback can be given to and from arbiters, merchants, and buyers.

1.3.5. Multiple payment options. CoinPayments offers the direct conversion, transfer, and ownership of hundred of cryptocurrencies today. The CPSCoin's marketplace will allow offer payment via the complete set of coins by adding

an abstraction layer, which will convert coins to CPSCoins upon purchase and lock the CPSCoins in escrow thereafter as part of the normal offer purchase flow. Native support can be for any coin that supports the same private key and signature scheme format that Syscoin does. This will allow client-side conversion from the native CPSCoin address to the desired address, such that the same private key can unlock payments in the other supporting blockchains where the private key can be used to sign off on spending those coins. Direct support for CPSCoin and other assets will work differently. Instead of sending money to an address, an Asset transfer would happen between identities, since Asset transfers use the identity as input instead of coins.

1.3.6. Shipping notification system. A payment acknowledgement button, on escrow and offer payments, allows for a multi-use notification system that notifies the buyer when a merchant acknowledges payment or when he is about to ship the product. Tracking and other shipment information can then be sent via the encrypted messaging system.

1.3.7. Marketplace moderation and Private Offers. Marketplace moderation is done through the safe search feature, which allows blacklisting of offers that are in obvious violation of ethical and moral code of conduct. Merchants may allow one-on-one deals by setting the offer to private.

1.3.8. Cryptographic security through blockchain unique user identities. Any CPSCoin user that updates his offer listings or digital certificates, must sign an input of their blockchain anchored identity. This is a cryptographically secure means to ensure of the provable ownership and modifications of those services. Consensus code at the blockchain layer ensures the proper identity signature matches that of the public address associated with the authorizing identity.

We have applied rules, such as that of domain names, to user identities. Only allowing unique case-insensitive names. Users are now able to send coins and encrypted messages to an alias using any case formatting desired. The recipient will always be the user who owns the lowercase version of the alias.

Within the identity context, users may post public and private details including: social profiles, profile pictures, and other information that users may want others to see.

1.3.9. Identity specification. By storing identity data off-chain with a blockchain-anchor, we can increase the amount of data users are able to store within their CPSCoin Identity without bloating the blockchain. By storing these entities off-chain were also able to evolve the specification for Identities without having to fork the network. Another costly piece of overhead, if we were to store the actual data on-chain.

```
export interface CPSCoinIdentity {
  publicIdentity: CPSCoinPublicIdentity;
  privateIdentity?: CPSCoinPrivateIdentity;
  encryptedPrivateIdentity?: string;
}
```

1.3.10. Public Identity specification. Were starting with what most modern-day platforms attach to an identity simply for the purposes of enhancing the current Blockmarket experience while making marketplace interaction even easier. Some of the items you can store in your CPSCoin Identity are things like an avatar image URL, first name, last name, Facebook URL, Twitter URL, PGP public key, bio, and more.

```
export interface CPSCoinOnChainIdentity {
  avatarUrls: string[];
  firstName: string;
  lastName: string;
}

export interface CPSCoinPublicIdentity
extends CPSCoinOnChainIdentity {
  location?: string;
  pgpKey?: string;
  bio?: string;
  facebookUrl?: string;
  twitterUrl?: string;
  instagramUrl?: string;
  bitcointalkUser?: string;
  trustedArbiterNames?: string[];
  requireTrustedArbiter?: boolean;
}
```

1.3.11. Private Identity specification. Identities also feature a private data field which is secured against public access using ECIES encryption. Properties like shipping address and PGP private key can be stored in this section of the identity with confidence of data safety, because it is stored encrypted and can only be decrypted with the owners alias and password.

```
export interface CPSCoinPrivateIdentity {
  shippingAddress?: string;
  pgpPrivKey?: string;
}
```

1.4. Data anchoring

The diagram below depicts an innovative way to have a blockchain based anchoring mechanism that authenticates to your blockchain Identity.

Figure 4: Data anchoring infographic

The client attempting to write the first data signs the payload using their private key; this establishes their control of the piece of data which they are trying to modify. Next, the raw payload, signature, and public address are sent to the server. The server uses a local independent connection to a full node to validate the signature based on the address submitted. If the signature is valid, then the write operation is executed. If not, then the write is prevented and an error is returned to the client.

1.5. Instant Encrypted Messages

Since users of the marketplace operate through a blockchain unique identity, an encrypted messaging service is easily plausible such that both sender and receiver are authenticated, providing cryptographic certainty that User A received a message from User B, and only User A can decrypt and read that message using his unique CPSCoin identity. Encrypted messages are secured with the Elliptic Curve Integrated Encryption Scheme (ECIES) symmetric key negotiation from public keys to encrypt arbitrarily long messages. Multiparty encryption is also possible through the use of multi-signature identities. Messages are not stored on the blockchain but blockchain based identities which are used to send and receive messages, can decrypt and read messages sent to them as well as cryptographically verify who the sender is.

1.6. Blockchain pruning

To combat blockchain bloat, the underlying infrastructure implements a novel pruning mechanism to remove marketplace listings and other unused data based on the expiration of CPSCoin user identities. If a user is inactive, their data will removed. The data on the blockchain thus represents active users using the CPSCoin marketplace. To understand better how the pruning works technically, please refer to the Syscoin whitepaper.

1.6.1. Fee-less blockchain interaction. If users are using the web-portal, the fees for creating transactions per user will be rate-limited to reduce spam. Fees will be covered by CoinPayments. The raw transactions which sign the identity inputs to authenticate changes to their services/CPSCoins are given to the CoinPayments server which will further sign and add inputs to cover for expenses on half of the user. This ensures that private keys are not exchanged over the network and are not susceptible to theft.

1.6.2. Zero-knowledge alias authentication. Alias private keys can be generated deterministically by supplying a password, hashed with a generated random 256bit number known as the password salt. The salt is stored in local storage on the browser and offline. Upon interactive client logins, the derived key can be regenerated based upon the user supplied password, and checked against the public address of the alias that is being authenticated against through an identity information lookup from within any full node. This enabled walletless on-chain controls and authenticated spending of coins/services without requiring a transfer of credentials over any network.

1.6.3. Expiration. Identity expiry happens based on time. The blockchain protocol acts as a decentralized time server which stamps blocks based on height and time. Services expire when the identity related to it expires, and escrow will expire if and only if the arbiter, buyer, and seller identities involved have all expired.

1.7. Certificates

Digital certificates within CPSCoin infrastructure are useful for all kinds of applications; from storing bits of data to creating data that may be sold and transferred upon purchase. All with provable ownership via the blockchain.

1.7.1. Public and private data. Certificates, like aliases, have public and private data. Private data can be accessed by foreign aliases either through creating a multi-signature alias and including other aliases or by transferring ownership of the certificate to the new owner.

Using a multi-signature approach allows certificate owners to maintain control of their certificates while still allowing decryption of private data by other users. In this instance, an owner would change the alias of the certificate to point to a new multi-signature alias, then assign two aliases owned by the owner and one alias owned by another party.

1.7.2. Transfer of ownership. Certificates can be transferred to other identities. New owners will receive reading rights for any private encrypted data and the transfer can be configured to allow editing of certificates upon transfer.

1.8. Escrow

CPSCoin's integrated escrow service allows safer payments of offers by securely holding a buyer's tokens in escrow, until the terms of the sale are met and the buyer releases payment to the seller. The system is arbiter-based. Arbiters act as trusted third-parties between buyers and merchant for a sale in the decentralized marketplace. An arbiter is paid based on a dynamic fee set in the rates peg for the offer that is sold. At the end of the process of completing an escrow, all three parties can be rated and given feedback related to the sale.

Arbiters are chosen by buyers when accepting an offer. Normally the buyer and seller would agree on an arbiter before an offer is accepted. In most cases no dispute is filed and no arbiter action is needed.

If a merchant does not ship goods, the arbiter refunds the buyer. If the buyer receives goods as described but does not release payment, the arbiter releases funds to the merchant. The feedback and rating system should help prevent irrational behavior by aligning incentives such that it allows actors to benefit if acting honestly.

The fees paid to arbiters are only applicable if the arbiter acts to sign off on a refund or release payment to merchant. The fees are also set by the buyer upon purchase and are adjustable. The fee rate requested by the escrow agent should be taken into consideration when the buyer adjusts the default fee rates when purchasing. Escrow agents with better reputation scores and more transactions arbitrated can charge higher fees as a result. In general, however, a market equilibrium will present itself between the demand and supply of escrow agents and their fees.

Once escrow is created, users involved in the escrows will be provided with lists of deals to govern and control based on their role in the escrow, through the user-interface portals.

2. Open-API Specification

We will develop an Open-API compatible specification so software developers have clear and concise APIs in the language of their choice to build products tailored for use with CPSCoin.

3. Future work

We will work with the Blockchain Foundry team to innovate and bring value to the token holders of CPSCoin in a variety of ways, including scaling and better third-party escrow solutions for trustless e-commerce.

3.1. Lightning networks

We are looking to develop an off-chain transaction mechanism whereby we are able to provably move CPSCoin assets in high volume without fees and without affecting blockchain bloat. This would be similar to the Lightning network proposed for Bitcoin, but with assets in mind and, perhaps, with masternodes in a more trustless and decentralized fashion, without the issue of trusted central hubs[Fyo]

3.2. Offers/Escrow

Proof-of-shipment is something we have innovated, we are expanding upon the shipping notification system from within the escrow- and offer-service layers. A video can be taken by the merchant, hashed and included in a data-field from within the shipping notification transaction, so that arbiters and buyers can assure that any disputes would be quickly and efficiently resolved. This helps relieve a few concerns: there will be proof that the merchant shipped goods as described by the offer and a reduced incentive for fraudulent charge-backs by the user, since there is proof that the shipment took place and thus the argument that the merchant did not send goods that are as described or did not ship at all are invalid. Arbitration and insurance would become cost-effective means to insure true buyer protection as markets form as a result of the technology. There is currently a proof-of-concept under development for this proof-of-shipment mechanism.

Another thing we are working on is the ability for robots to act as escrow agents and deliver goods to buyers within 5km of distribution centers. There are services that do this for the food industry and would simply need to be adapted to allow for a decentralized marketplace with tokens held in escrow, which would be released upon acceptance of delivery.

4. Proof-Of-Stake Asset

CPSCoin will be the first-of-its-kind blockchain asset that provides staking functionality.

CPSCoin will have a 5-year vesting period where owners will gain a staking percentage of 25, decreasing by 5 percent per year. There is an additional gain of 5 percent per year (up to 25 percent per year over 5 years) as a loyalty payout to holders of CPSCoin which do not move their coins. The max supply is based off of the scenario where every token is given the full loyalty premium payout as well as the staking percentage to arrive at the maximum possible amount of coins in existence. Of course this is unlikely, and the extent that the supply will fall short of this maximum will depend on the amount of people staking and holding coins over the vesting period.

5. Conclusion

Acknowledgments

We would like to thank Satoshi Nakamoto, the late Hal Finney and Gavin Andresen for bringing Bitcoin protocol and reference client to mainstream adoption state as well as the Blockchain Foundry Inc. team for building the asset structure on top of Syscoin, which builds on top of the Bitcoin layer and lets us bring CPSCoin to market.

6. Caution to Reader

This white paper is intended for persons who are highly sophisticated in the blockchain and crypto world. It is not a finance document nor a description of securities. This paper describes what is believed to be a utility to enhance functionality. Any person participating should be aware of the state of unsettled law regarding cryptocurrencies and be aware not just of the risk of being a participant (all funds you contribute could be lost) but of the risk that one or more securities regulators may decide the tokens described herein are securities which could result in cease trades affecting or preventing liquidity and price, which price could drop to zero. No person should participate herein unless they are willing to suffer complete loss of any and all financial contribution.

References

[Bre] Eric Brewer. *Brewer's theorem*. URL: https://en.wikipedia.org/wiki/CAP_theorem.

[But] Vitalik Buterin. *ERC20 Token Standard*. URL: https://theethereum.wiki/w/index.php/ERC20_Token_Standard.

[Fyo] Jonald Fyookball. *Mathematical Proof That the Lightning Network Cannot Be a Decentralized Bitcoin Scaling Solution*. URL: https://medium.com/@jonaldfyookball/mathematical-proof-that-the-lightning-network-cannot-be-a-decentralized-bitcoin-scaling-solution-1b8147650800.

[HJ] K. A. Hawick and H. A. James. *Enumerating Circuits and Loops in Graphs with Self-Arcs and Multiple-Arcs*. URL: https://blog.mister-muffin.de/2012/07/04/enumerating-elementary-circuits-of-a-directed_graph/.

[Ker] Paul Kernfeld. *How Bitcoin Loses to the CAP Theorem*. URL: http://paulkernfeld.com/2016/01/15/bitcoin-cap-theorem.html.

[Nak] Satoshi Nakamoto. *Bitcoin: A Peer-to-Peer Electronic Cash System*. URL: https://bitcoin.org/bitcoin.pdf.

[Sid] Jagdeep Sidhu. *Syscoin: A Peer-to-Peer Electronic Cash System with Blockchain-Based Services for E-Business*. URL: http://syscoin.org/whitepaper.pdf.