Syscoin: A Peer-to-Peer Electronic Cash System with Blockchain-Based Services for E-Business

Jagdeep Sidhu, Msc.
Syscoin Core Developer
Blockchain Foundry Inc.
Email: http://www.blockchainfoundry.co/contact-us

***Abstract*—While Bitcoin (Peer-to-Peer Electronic Cash) [Nak] solved the double spend problem and provided work with timestamps on a public ledger, it has not to date extended the functionality of a blockchain beyond a transparent and public payment system. Satoshi Nakamoto's original reference client had a decentralized marketplace service which was later taken out due to a lack of resources [Deva]. We continued with Nakamoto's vision by creating a set of commercial-grade services supporting a wide variety of business use-cases, including a fully-developed blockchain-based decentralized marketplace, secure data storage and transfer, and unique user aliases that link the owner to all services controlled by that alias.**

1. Introduction

Syscoin is a permissionless blockchain-based cryptocurrency with a set of smart contracts which have been thoroughly tested and built on the Bitcoin scripting system using OP1 to OP16 standard script op-codes, representing coloured coin transactions, controlled by a hardened layer of distributed consensus logic for each smart contract (Syscoin service) while still retaining backwards compatibility with the Bitcoin protocol. These contracts can be combined and made to communicate with each other, forming building blocks for blockchain-based e-commerce solutions. For example, the Syscoin Alias Identity service is used by all other services to create cryptographic proof verifying the ownership of an Identity and ensuring that the owner of an Identity is the only one who can make Service related agreements, update offers and create or modify certificates. The Identity service also allows the owner of an Identity to communicate encrypted messages to other Syscoin Alias Identities. Impersonation is avoided by including the Unsigned Transaction Output (UTXO) of a previous Alias transaction that then belongs to the user owning the Identity and including it in future service-related transactions. Typically when creating a Syscoin service users have the option to specify which Identity the service belongs to. Most of the major use-cases using smart contracts with online services would use a subset of these services; it makes sense to harden these smart contracts and introduce them as a commercial-level quality set of tools, used by developers and integrators to create tomorrow's disruptive blockchain applications. This is in contrast to using turing-complete smart contracts, which, by definition, are not hardened due to the open-ended nature of the underlying scripting language. Commercial integrators who are looking for a secure solution to leverage the increased efficiency that blockchain technology allows compared with traditional e-commerce applications are better off trying to use a hardened service which cannot change and is well-tested with regression testing, white-box and black box testing, specifically targeting rules of the application. Integrators are also inclined to choose the most powerful network available which is currently Bitcoin, so the optimal solution for them would be to have a hardened contract which does what they need running in an environment which is protected by Bitcoin's network security; these requirements make Syscoin the most viable choice.

1.1. Turing-complete scripting

Turing-complete smart-contracts are technical marvels. But one has to question what real use they can be to practical business processes. For general purposes business logic is hardened upon the Software Requirements and Specification stage of the software development life-cycle. It makes sense to harden the rules of the contract, running on a system that has a measurable number of points of failure as does Bitcoin. With the mix of variables at the intersection of decentralized networking and Turing-complete smart contracts compounded by the complete lack of development oversight and vetting that Ethereum applies to contracts deployed to its network creates the potential for disaster scenarios for those invested in/using these smart contracts as was proven by the DAO experiment [But]. Here we had a malfunctioning code-path that was discovered and leaked a large sum of money rendering the experiment as a failure. It revealed larger issues with the design and immaturity of Solidity [Dai], the official language used to write smart contracts on Ethereum. With Turing-completeness comes an infinite paths of execution and risk of failure. We can make the guess that it is an evolutionary process of finding issues, one can equate the DAO attack to SQL injections which caused pain to many data driven web applications in the late 90s. Had Solidity been written so it made it harder to write bad contracts perhaps the problem would have been prevented.

The consensus currently is that if you do not need smart contracts to solve your problem it is best avoided until the languages and toolboxes on top of the smart contract core API are hardened to a point where they certified to be used by the general public. It is an open-box software experiment that is useful for applications which can offer cost-effective contracts that must change on demand. In the context of most of the e-commerce applications we have seen, this level of flexibility is not required. In contrast, with Syscoin we have decided to try to generalize applications into a set of hardened services that can be used in conjunction with each other to create complex use-cases, some of which are visionary and some of which are disruptive to current business flows without totally re-designing them. In an effort to introduce such technology to the real world, which has employed certain processes for years, it is important to try to introduce new technology in a way that is adaptable to the existing processes and works within existing judicial and legislative frameworks. In the grand scheme of things, the speed at which a smart contract can be developed does not matter if it has a high risk of being effectively insecure.

1.2. Innovative service layer

We have created an effective Alias Identity system which allows payments, to and from, direct towards recognizable names. Identity systems have associated data and public keys internally stored, making them useful to perform blockchain activities such multisignature signing, payment discovery and maintaining identity payment balances. We took an intuitive step forward from the Bitcoin core codebase and extended it with capabilities that made business sense without sacrificing security.

An alias can then create an offer in the decentralized marketplace to exchange goods without alias owners. Note that the alias is cryptographically linked to a service such as an offer by enforcing a rule that the alias output of the last alias transaction is linked to the creation of any subsequent Syscoin service transactions. This ensures the owner of the alias identity is the only person who can make those transactions. Offers and certificates may work together in the sale of digital goods.

A certificate is a proof of ownership token of data that can have arbitrary public and private encrypted information. It is transferable and generally done so at request or upon a sale.

Escrow is used to facilitate the exchange of goods or services and money. A general purpose escrow service was incorporated into Syscoin by allowing for transactions to be created at each step; from the creation of the multisignature payment, to the payment or refunding of that payment.

Messages can securely transmit communication between alias identities to facilitate trade negotiations or requests amongst participants in Syscoin service usage.

1.3. Syscoin as a currency

Similarily to Bitcoin, Syscoin acts as a payment system whereby tokens can be securely transferred within the Syscoin network. Syscoin tokens currently have an active market and an observable value, as such they have begun to fulfil the basic functions of a network currency in terms of acting as a store of value, a medium of exchange and a unit of account. As the network scales up, and as the tools we have developed are used by increasing numbers of users, Syscoin tokens will continue to function as the main token powering the network.

1.4. Design philosophy

Syscoin was developed in a carefully designed Agile development process with compatibility with parent source code network Bitcoin in mind. Since Syscoin was forked from the Bitcoin codebase it makes sense to leverage the community of that network in terms of mining and development by ensuring it is as easy as possible to rebase Bitcoin into Syscoin upon major code releases. Merged-mining with Bitcoin means Syscoin users can enjoy a powerful network preventing double spends and other network related attacks. At the time of this writing the Syscoin network carries about 25 percent of the mining power of Bitcoin [Cry].

The goal was to innovate on top of Bitcoin source codebase but be backwards compatible to leverage the network effect of Bitcoin in a manageable way. Since Syscoin is fully complaint with Bitcoin, all of the external tools and processes meant for Bitcoin can also work with Syscoin. The key difference at the core level between Bitcoin and Syscoin is that Syscoin can handle 1 minute block times for faster settlement of transactions. Although Bitcoin was not designed to be a transaction processing mechanism and network speeds have demonstrated to be increasing by at least 50 percent per year based on Neilsen's Law of Internet Bandwith [Nie] (Neilsen accurately predicted that the 300 bps modem speed in 1984 would increase to 120 Mbps lines in 2014) we can safely assume that majority of the network around the world will be able to settle transactions within 1 minute rather than a 10 minute window. Syscoin supports 1 Mb blocks, resulting in the ability to process 10 times more data than Bitcoin. This was needed because the average transaction size in bytes for Syscoin service transactions range from 2 times up to 10 times the size of an average Bitcoin transaction. Even so, the pruning mechanism that was innovated by the Syscoin engineering team is able to cope with demand for Syscoin service related data by removing the bandwidth and storage constraints on the network for service data for these transactions after service expiration.

At the time of this writing, the Bitcoin codebase is about 114k lines of code (physical SLOC). Syscoin's code (around 19k lines of code) on top is about 16 percent of Bitcoin's. An external library cryptopp was added for encryption mechanisms used for certificate, private alias and offer buyer note fields which added 25k lines of code. In

total there is about 158k lines of code to date. It is important that code coverage through unit tests correctly indicates the health of the underlying code base to ensure that any changes made do not have unforeseen ripple effects. Not one individual or team is aware enough intellectually to be able to accurately determine the state of health of a system across every isolated change. Since a system like Syscoin relies on valuable tokens which power the network and store users' wealth, it is imperative that the code supporting those tokens is in a state that is safe for the general public to use. Diligence is mandatory for any responsible actors in the development sphere to ensure that any code intended to be used by people who do not understand the open source protocol or who cannot read code is safe to use. Bitcoin developers do a good job with covering their code through unit tests and we have developed a test suite for testing Syscoin services by running multiple nodes while running tests (black box testing). The number of tests in Syscoin is 137 while Bitcoin has 136. The number of tests per line of code is sufficient to prove that the quality of work done in Syscoin matches or exceeds Bitcoin's code base.

1.5. Alias identities as the backbone of Syscoin services

Part of what makes the services in Syscoin so intriguing is the connection they have to an alias identity system. Offers, certificates, messages and escrow all require actors to sign off on creation or updates of these services via their aliases. Cryptographically secure signatures (backed by enormous proofs-of-work) are required in order to make changes to the services that aliases are linked with. By linking services to an identity system it makes it much easier to integrate services into real-world scenarios which work with identity-based work flows. Almost anything we do today requires the signature of a known actor on the service contract. By providing a cryptographically secure mechanism to create, manage and link these identities to service contracts, it ensures a seamless integration to processes in real-world scenarios. Not only does this make it easier to understand and explain to others how to use Syscoin's services, but it is also easier to implement more dynamic features on top which leverage the use of aliases, such adding multisignature options to aliases to improve the workflow of common contracts where multiple parties representing a single identity are required to sign. Partnerships, common law or power of attorney contracts are examples of such cases where multiple signatures may be needed to represent an identity.

1.6. One key to rule them all

Because Syscoin private keys are equivalent to Bitcoin and ZCash Transparent keys, these chains along with Syscoin become tightly coupled in integration for merchant payments on the decentralized marketplace. Syscoin merchants can provide their merchant payment address in either Bitcoin or Syscoin by using the desired version byte prefixed along with the Hash160 of the Syscoin public key. In ZCash's implementation, 2 bytes are prefixed to the Hash160 of the Syscoin public key ("t1" and "t3" addresses). This way, at the network level, Syscoin is able to respond to requests for payments in other chains and collect those payments all from within one merchant key (the Alias Identity associated with the merchant). An interface within the wallet for Syscoin has been built to manage usability of these payments by connecting to these other chains via RPC connections to validate payments to merchants for offers. The validating nodes could be standard nodes for ZCash and Bitcoin running txindex=1 to ensure cross-network transactions can be queried by the getrawtransaction RPC call. A sendrawtransaction RPC call happens upon an escrow completion related to a Bitcoin or ZCash payment for an offer. All escrow-related cryptographic signatures can happen within Syscoin because of the compliance of the address schemes between Syscoin and these supported external chains. This greatly improves usability for escrow-related payments by hiding the complexity of a multisignature process of an external chain payment within the Syscoin escrow core service code. The only dependence lies in the face of the user interface which is mere convenience. Payment validation at the network level allows cross-chain payments to be validated just as a native Syscoin payment by validating that the destination address and amounts are correct for the offer being paid for. Any interface integrating with the Syscoin core can use their own method for validating that these external payments exist on their chains in the correct amounts; wallet-less nodes with txindex flag for each chain which can be accessed by the interface will be sufficient in most cases.

1.6.1. Mechanism design. Bitcoin provides two incentives for miners: block subsidy through rewards and transaction fees. As Bitcoin rewards wind down it will become unstable due to degrading miner incentive to do what's in the best interest for network security. Selfish-mining and undercutting are very real problems in Bitcoin's mining world. These are discussed in greater length in this paper presented at the ACM CCS [Nar16]. What we present is a novel mechanism design that prevents mining incentive from degrading by tying in usage of services to an inflation metric for block rewards. Transaction fees remain to provide incentive to mine and relay transactions but rewards will continue depending on the demand for using the Syscoin network. A utility metric can be established by determining the number of Syscoin service transactions per block. In Syscoin 2.1, an arbitrary number was chosen (5, not network enforceable) which represents the "high-demand" cutoff threshold for when to burn fees (under the threshold) or when to inflate the fees in the rewards (above the threshold). This means the monetary base can expand (inflate) however so slightly to accommodate for the demand to use Syscoin services and contract (deflate) when there are blocks that fall under the threshold. It is important to note that the fees in question that are being

burned or inflated are not the transaction fees, which are always paid to miners separately from the block reward, but the Syscoin service fee, which is on top of the transaction fee and is a dynamically adjustable fee from within the rates peg aliases. This creates a democratic system that carries the fee rate which is capable of adjusting the monetary supply based on demand or lack thereof during block producing events by miners. The result is a price stability mechanism similar to inflation targeting by central banks but is done in a decentralized fashion. High demand for the tokens will correlate with a slight adjustment to the supply positively while low demand will deflate and give current token holders more stake as a percentage of total supply (similar to the Proof of Stake consensus algorithm, but in the context of a Proof of Work architecture). The mechanism design closely follows the concept of Ideal Money [Nas02] termed at a Penn State lecture given by the late John F. Nash Jr. (Nobel Laureate in Economics). If we apply the notion of service transaction rate to facilitate the transfer of utility between network participants we have a metric that is the first of its kind, one that denotes true demand for the currency in circulation as a public utility that is audit-able and provides money transfers with transferable utility, and thus, ”quality” money which would classify as ideal. Nash alluded to using a ”public utility” such as the supply of electric energy or water as a high quality utility for inflation targeting but those are indirectly related to demand which is indirectly related to velocity of money. A utility metric would be ideal if when detecting real velocity of money that the quality of utility is maximized which divides over M1 (money supply). Traditional money velocity is calculated using GDP over M1 or M2 however GDP is a lower quality utility metric used which is not publicly audit-able due to a wide range of factors influencing the utility and susceptible to a range of public perceptions based on how it is calculated. See shadowstats [Wil]. Syscoin provides a way to determine the highest quality utility metric possible by providing a way to calculate true money velocity directly by averaging the service transaction creation rate over the monetary base and adjusting the base to accommodate demand to achieve price stability.

The threshold can be extended to become dynamically adjustable through the peg rates alias but this was intentionally left to a static number to keep the mechanism simple and allow public auditability and perception to not become erratic. This would affect price stability as it would begin to obfuscate what the true utility value is at any given moment.

1.6.2. Self-governing rate system. Because we wanted users to be able to transact in currencies other than Syscoin, we needed a way to access this information not only in the user interface but in the consensus code to be able to validate that offers were paid in correct amounts and to the correct person.

To do this we created the sysrates.peg alias which stores currency information in relation to Syscoin price. It is updated dynamically based on the volatility on exchanges. Other things such as transaction fees and arbiter fees are also stored and dynamically adjustable during network run-time to avoid having to do any soft or hard forks and having them take affect in real-time versus voluntary updates of miners and client wallets. Transaction fees are used for determining the amount of fees used when sending payments to escrow with Syscoin, Bitcoin or ZCash [Devb]. Because miners may change the amount of fee it takes to mine and relay a transaction, these variables are best to be dynamically adjustable based on market conditions. Of course, sysrates.peg is just a reference implementation of such an exchange rate service to bootstrap the marketplace with a ready-made and updated alias with exchange rate information. It may fulfill the needs of 90 percent of users who do not want to manage their own exchange rates and fees. However, for the select few who do care, they may change the alias that their offers use for exchange rate information and even expand the exchangeable currencies they accept. Their peg may also be used by others who feel perhaps that sysrates.peg fees are not desirable and want to use a different option (anyone can create their own exchange rate peg). By allowing users to select the exchange rate alias that their offers relies on, it creates a self-governing system of exchange rates and fees which adapt to the needs of the users on the network. See below in the Alias Rate Peg section for how the pegs can create a democratic rate system which scales with the price of Syscoin.

1.6.3. Quality assurance through network simulation. A test suite was developed to allow the simulation of live network scenarios. Tests cover pruning, expiry and general use-cases of Syscoin services. It is an integral part of achieving a commercial quality level product in any software application. The setmocktime rpc call is used to set the time in-advance of blocks to simulate expiration of services and pruning.

1.7. Alias identities

We have applied domain-name like rules to Syscoin Alias Identities, allowing only unique case-insensitive names. We have changed the Alias renewal model to be more like Internet domain names; Aliases can be renewed up to five years at any time. Users are now able to send coins and encrypted messages to an Alias using any case formatting desired, the recipient will always be the same. For example, using ”Dan” or ”dan” is equivalent and will go to the same recipient the user who owns the lowercase version of the Alias ”dan”.

The following domain-name rules apply to Aliases upon creation:

- The domain name should be a-z/0-9 and hyphen(-)
- The domain name should be between 3 and 64 characters long
- Last TLD can be 2 to a maximum of 6 characters
- The domain name should not start or end with hyphen (-) (e.g. -syscoin.org or syscoin-.org)
- The domain name can be a subdomain (e.g. sys.blogspot.com)

- The pricing model depends upon the length of renewal, it is the product of the normal Syscoin service fee with the square of the renewal length.
- F = F * pow(2.88, R) where F=fee in coin amounts and R=renewal from 1 to any number (years)

1.7.1. Cryptographic security through alias identities. Any Syscoin service you create or update must update an Alias identity input which employs a cryptographic scheme that secures the transaction with provable ownership of those transactions. Consensus code for Aliases, Offers, Certificates, Escrows and Messages all require inputs from the Unsigned Transaction Outputs(UTXO) of an Alias Identity transaction that has been signed with the owners private key. This allows for an Identity to play a key role in ensuring safety, secure from impersonation or any other attempt at attacking the integrity of the relationship between the Identity and services that are involved with the Identity. Because the inputs need to be valid in the UTXO database this means that the inputs need at least 1 network confirmation to ensure that the owner is indeed the one who is the one capable of making these transactions. In order to improve usability, 5 (an arbitrary number) of outputs are created upon an alias transaction so that multiple service transactions relating to an Alias Identity can be made within the same block on the network. The alias consensus code ensures that the public key of the alias input to the transaction matches the public key of the alias. This validates the one who is making the transaction to modify or update an Alias Identity.

1.7.2. Public and private profile data. Aliases have public and private profile data. Private data is encrypted to the encryption public key stored in the alias which is generated upon activation of that alias. The encryption private key is itself encrypted to the alias public key so only the owner of the alias can view or change the encryption key used to decrypt sensitive private data. Encryption keys are stored separately so that group encryption becomes possible with multisignature aliases. See section 1.6.6 for more details on group encryption.

1.7.3. Ratings. Ratings are provided (ranging from 1 to 5 inclusive) which keep a count of ratings per user perspective or being a merchant, buyer or arbiter. The count divides the accumulated rating value based on each role to come up with a fractional number between 1 and 5. Feedback and ratings play a key role in identifying actors which are rational and provide willingness for other actors to work with them through marketplace activities.

1.7.4. Transfer ownership. Aliases may be transferred to another public key. A new public key can be generated on the alias screen from the receiver. A public key cannot be shared amongst multiple aliases which would cause confusion in the wallet. Thus at the consensus level transfers are checked to ensure the new public key of the alias does not already exist in another alias inside of the Syscoin alias database.

1.7.5. Alias balances. Since payments to aliases send change back to the alias address it becomes a natural evolution to keep alias balances. The UTXO related transactions for each alias are kept in a separate database to allow spending to and from aliases distinctly separate from other outputs that are available in the wallet. It allows for clean management of funds as well as provides simplicity when dealing with multisignature funds which need signing from multiple parties for spending. It also allows for interaction with the Syscoin services through utilizing the alias authentication mechanism which provides the private key for the alias to unlock UTXOs related to that alias for updating, adding or removing services related to that alias, all in a headless state without the need for wallet interaction. The alias acts as a keystore then and becomes your wallet for the funds inside that alias. Your alias password can be used to unlock those funds using any tool capable of sending rpc commands to an online Syscoin node.

1.7.6. Multiparty encryption through multisignature aliases. An alias makes a symmetric group key K, which extends to multiple parties through the use of multisignature aliases where the key K becomes known to all members of the alias by virtue of sending to each member the encryption of K with his public key. This happens upon update or activation of a multisignature alias. You may begin with a normal alias and extend multiple parties to offer viewership of private data such as certificates or via the encrypted message service or you may revoke the privileges by simply managing which aliases belong to the multisignature alias signature list in the alias settings screen.

All encryption done in Syscoin is done based on Elliptic Curve Integrated Encryption Scheme(ECIES) with the secp256k1 curve. It is the same curve used by the Elliptic Curve Digital Signature Algorithm(ECDSA) algorithm in Syscoin used for signing transactions. The symmetric public key encryption allows the alias service to become the keystore to manage and distribute encryption keys conveniently without involving any complexity with the user. It is also good to note that the curve secp256k1 (a non-conventional Koblitz curve) is not the same as secp256r1 which was used everywhere as standard at the time of Bitcoin's creation. Satoshi was smart enough to know that using secp256k1 non-standard curve was the better choice. We now know in hindsight that secp256r1 involved a seed c49d360886e704936a6678e1139d26b7819f7e90 [Pacb] which is linked to the backdoor found in Dual-EC-DRBG linked to the NSA (thanks to Edward Snowden) [Per]. Even the late Hal Finney questioned the use of a non-standard curve in 2011 not knowing the link between the seed used in secp256r1 and the NSA [Fin]. Thus the encryption in Syscoin is based on a strong security protocol that other encryption mechanisms may not share.

1.7.7. Alias authentication. An aliasauthenticate RPC function is provided to the user that takes in an alias, and a password and gives a private key of that alias. Alias private keys can be generated deterministically by supplying a

password. If an alias is deterministically created or updated, the aliasauthenticate RPC can be used to retrieve the private key from any node on the network. It essentially regenerates the private key and validates that the public key of the alias and public key of the generated private key match to prove it is the correct password/alias combination passed into the function. This is useful for retrieving your private key if you do not have access to your wallet and is part of a bigger goal of being able to transact on the blockchain with walletless nodes. The listing functions of Syscoin such as offerlist, certlist, escrowlist all take in alias or an array of aliases to list information pertaining to those services through the distributed Syscoin services databases. They are not dependent on the wallet to detect which information to display. A private key can be passed into these functions which are used to decrypt sensitive information such as buyer notes and private data from certificates or aliases. The only requirement Syscoin currently has with wallet is to display a list of your own aliases and making transactions to update your services. The Signrawtransaction RPC function is an example of a transaction-creating function which potentially creates a new keystore to sign transaction inputs if private keys are provided to the function and serves as a wallet-less transaction mechanism. Similarly, Syscoin transaction creation routines can do the same thing to remove dependence on the wallet. Doing so opens up interesting designs such as the ability to login (via Syscoin's Aliasauthenticate RPC function) to a secure website (https) that is trusted by the user (perhaps their own website) and create transactions with their services over the Internet running on a server hosting a wallet-less node. Thus the server security requirements and thus costs are reduced to just preventing the DDOS attack. Since no user keys are stored on the server, the majority of incentives for hackers are removed from the design. This fulfills the vision that Bitcoin developers set out for when creating the Signrawtransaction RPC function with the ability to sign with any private key. Because remembering a password is much easier than the private key, the usability of alias authentication brings wallet-less transactions into light.

1.7.8. Safe search. Two tiers of marketplace moderation both user-defined via SafeSearch and team-defined via a 3 tier system. The moderation system allows for a more public-friendly marketplace with options for SafeSearch restricted items. The client can choose if they wish to enable or disable SafeSearch through the wallet settings. Any actor that is adding content that is not suitable for searching for the public can set his or her offer to private which will hide it from the searches but keep its validity on the network just as any other offer.

1.7.9. Expiration. Alias expiry happens based on time. The blockchain protocol acts as a decentralized time server which stamps blocks based on height and time. We leverage the time component of the forging of the block which is backed by an enormous amount of computing work which cannot be modified without redoing the work (Bitcoin's whitepaper showed this is not feasible). Thus we can depend on the time being accurate and reliable as a form of tool to use to detect when an Alias expires. All other services connect to aliases and use the alias that owns the service to detect expiry. Offers, certificates and messages expire when the alias related to it expire and escrow will expire if and only if both buyer and seller aliases involved are expired. This prevents the case where the seller cannot complete escrow because buyer becomes inactive or where buyer cannot complete refund because seller is inactive. As a result of using time for expiry, you may create Aliases which expire at certain timestamps which allow many unique use-cases on the blockchain with time-expiring contracts, provably linked to Syscoin service lifetime. Of course the longer you set the expiry into the future the higher the fee you will pay(see the fee structure above in section 1.6). The fee is dynamically linked to the alias rate peg used for an alias and can adjust to market demands.

1.7.10. Alias rate peg. By default new users will be able to use sysrates.peg which the Syscoin team will manage and update with the most commonly used currencies and payment options based on an algorithm that determines update frequency based on volatility of the underlying currency and assets in the rates alias. Service fees (inside the SYS object of the data model) are included to allow dynamic control of the Satoshi per Byte requirement of Syscoin service transactions. This allows for the service rates to be adjusted using the standard Alias Update API as the market demands. This avoids the need to change rates via costly code updates and forking changes across the network. The updates happen in real-time and take affect across the network in 1 block. For example, if the target price of 1 USD dollar per Syscoin is achieved: the default value of the fee is 4000 Satoshi's per byte and for a 2kb transaction populating an offer related transaction that will be about 0.01 Syscoins after taking into account the minimum relay fee and mining fees. That would equate to about 1 cent USD for each update. Now if the value went up to 100 dollars USD the fees can be reduced to 40 Satoshi per byte to keep the service fee at 1 cent USD. What makes the system democratic is that any Alias Identity can be associated with any other peg rates alias which may offer a rate more indicative of current market sentiment.

Because the owners of Alias Identities can adjust which rate peg alias to associate their alias with at will or even change which alias an offer belongs to which possibly changes which rates peg that offer will use, careful network checks must be performed to ensure that payments are correct across all offers being accepted on the network. When distributed consensus checks are performed, the block height at which the payment was made in, is saved and used as an index to look up the current rate peg for the Alias Identity of the merchant and then price is matched with what the buyer paid. If there is a discrepancy between what the buyer owes for an offer and what the converted rate is calculated to be for that block then the buyer is displayed given an error message on payment. This allows for deterministic payment behavior to ensure merchants that payments are in correct amounts without having to do cumbersome manual check

for every payment of an offer.

1.7.11. Multisignature identities. In Syscoins identity system (Aliases) we store the public key, the number of signatures required and redeemscript which are an integral part of the multisignature signing process. Without these pieces of information which links the parties involved, a multisignature transaction cannot be signed and sent to the network. In Bitcoin these pieces are all handled offline and if forgotten or misplaced, leads to permanent forfeiture of the coins stored in a multi-signature address.

Syscoin extends multi-signature functionality through Alias Identities by paying to a multisignature transaction rather than using a Script Hash (P2SH). Because P2SH does not have enough information in it to know who needs to sign and how many signatures are required, it forces users to send this information to each other using a cumbersome raw transaction API. Although we can send the information using encrypted Syscoin Messages it is still a hinderance for mass adoption of any technology built using multisignature features of a blockchain based project. Everywhere a payment is done, a redeem-script is looked up from the payment address as an alias identity and used as a destination instead of the address destination. Taking this approach simplifies the process of using multisignature transactions and leads to a more flexible approach on how to best use it- surfacing new use-cases which increase real-world productivity.

A new user interface has been developed to let you sign a multisignature transaction, allowing inspection of the transaction including any Syscoin service related information, before signing. It is now easy to implement a Know-Before-You-Sign policy which allows users to simply check the decoded transaction before signing it. Indeed any multisignature transaction that is signed should not be trusted regardless of who is involved, it should be decoded and checked prior to signing and sending to the network to avoid unscrupulous behavior from bad actors who spend the coins held in escrow in undesirable ways.

By creating an Identity system which supports required multisignature details we can fix the deficiencies in the Bitcoin Core which blocks multisignature transactions from the transaction ledger and spending selection screens such as Coin Control. This allows for a more intuitive user experience of multi-signature transactions at a core level. Everyone from novice to experienced users will appreciate the intuitiveness and convenience of not having to send details to responsible parties to be able to sign and send multi-signature payments on-chain.

As a result Syscoin Aliases now support multisignature ownership. By combining an identity system and multisignature transactions we have an easy to understand system that allows users control over their identity while providing maximum flexibility in terms of real-world usage. Although the blockchain in many ways simplifies business processes and finds newer and better ways to solve problems, we still cannot get around things that involve multiple parties holding sensitive information. Until now. Because Syscoin Certificates were made to store these sensitive pieces of information, it becomes almost natural to think that hooking them up with multisignature Aliases would allow for Certificates to now be fully functional. Indeed that is what we did. We have extended the functionality of offers, messages, certificates and aliases to allow for full multisignature ownership.

1.8. Certificates

Digital Certificates on the Syscoin blockchain are useful for all kinds of things from storing bits of data to creating data that may be sold and automatically transferred upon purchase- all with proveable ownership via the blockchain.

1.8.1. Categories. Certificates also use categories like offers however they are restricted to using a certificate category defined in syscategory alias. The alias acts as a dynamic mechanism to update the category system in Syscoin. A certificate must set its category to a certificate or any of its sub categories. The syscategory alias is used by offers and certificates to populate possible default categories that can help in organizing information so it becomes searchable.

1.8.2. Public and private data. Certificates like aliases have public and private data. Usually if someone is to sell a certificate they would have a public section as a preview of the data that is encrypted which would need to be paid for to be accessed. Private data can be accessed by foreign aliases either through creating a multisignature alias and including other aliases or by transferring ownership of the certificate to the new owner. By changing the alias of the certificate to point to a new multisignature alias created by the owner which he assigns 2 aliases he owns and one of the party wishing to read the data, it would allow the owner to control access to the certificate while still allowing decryption of the private data. If the external alias were to try to change the certificate he would have to get signatures from either of your aliases which are part of the signature list of the multisignature alias now controlling the certificate. However since you would have control over 2 of the 3 aliases in the signature list you would still retain control over the certificate allowing you to revoke readability privileges of the foreign alias

1.8.3. Transfer ownership. Certificates can be transferred just like aliases. However you may transfer certificates to other aliases for convenience. New owners will receive reading rights for any private encrypted data. The transfer can be configured to allow editing of certificates upon transfer. If it is enabled then new owners will be able to edit the certificate otherwise it will be locked from updating upon transfer.

1.9. Escrow

Syscoin's integrated escrow service allows safer payments of offers by securely holding a buyer's coins in escrow until the terms of the sale are met and as a result the buyer

releases payment to the seller. In most cases no dispute is filed and no arbiter action is needed. The buyer chooses the arbiter and seller would agree by sending goods or services to the buyer upon which buyer would release payment and seller would collect. Escrow works with native payments in Syscoins as well as external payments with ZEC/BTC by signing transactions inside of the Syscoin network and posting to the appropriate network once the escrow contract is complete.

Arbitrated escrow as illustrated in figure 1 shows the use of a arbiter which acts as a trusted third-part between buyer 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. The normal escrow transactions which do not involve the use of an arbiter does not pay arbiter any fees because they were not involved in doing any work related to the escrow process. However if the arbiter does get involved and issues a refund or release transaction they will be paid the fee. At the end of the process of completing an escrow all three parties can be rated and given feedback related to the sale. An arbiter can override a refund transaction by also re-issuing it if the merchant creates a malformed refund transaction that does not pay the correct amounts back to the buyer. The same thing applied to an arbiter being able to re-release escrow funds back to the merchant if the buyer acted unscrupulously.

Escrow acts as a zero-sum game between the buyer and merchant in the exchange of goods. If you denote the goods being exchanged and the price paid for them as 1 unit (1U) then we can summarize the exchange of units as follows:

- 1. Buyer creates an escrow for goods or service. -1U
- 2. Merchant transfers goods or service to the buyer. -1U
- 3. Buyer receives goods or service. +1U
- 4. Buyer releases escrow payment to the merchant. +1U

You can see that at the end of the escrow process that the zero-sum game is complete for both parties.

If merchant does not ship goods, the arbiter simply refunds the buyer. If the buyer receives goods and it is as described but doesn't release payment, the arbiter simply releases funds to the merchant. Of course there is no system that is fail-safe from irrational behavior so due diligence needs to be taken on the part of both the buyer and merchant to prove without a reasonable doubt if they had been cheated. The feedback and rating system helps prevent irrational behavior by aligning incentives such that it allows actors to benefit if acting honestly. In other words, buyers will not buy from merchants that are justifiably rated badly and merchants will not sell to buyers who do not seem honest. An arbiter which earns fees will only be used if they are reputable as well.

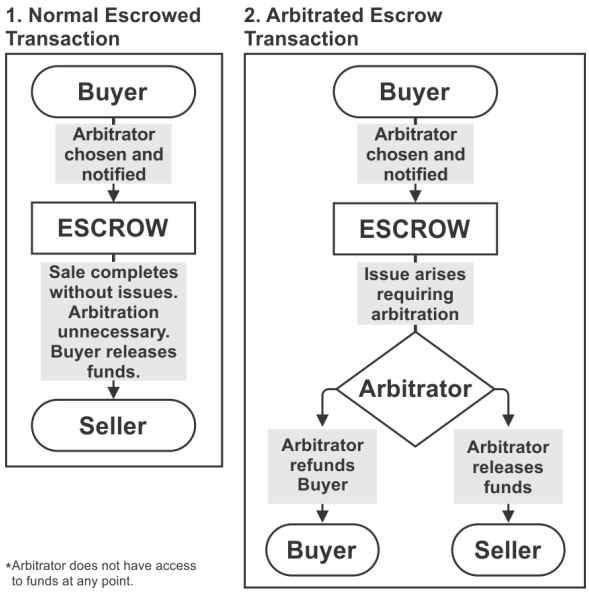

Figure 1: Syscoin's arbitrated escrow service

1.9.1. Escrow Support on external payments. The multisignature escrow feature works nicely with our DirectBTC/DirectZEC integrations which allows signing and sending raw transactions to the Bitcoin/ZCash networks respectively and spending those coins, all via the Syscoin network. In Syscoin Escrow, if a user wishes to pay via Bitcoin or ZCash they would pay to a generated P2SH representing an escrow address. The raw transactions to spend those coins to the merchant/re-seller for commission and buyer for refunding any escrow arbiter fees would all be done in Syscoin. The involved parties would simply click a button to complete their role in the escrow. Fully signed payments are sent to the Bitcoin/ZCash networks automatically upon release/refund with no manual merchant interaction required. The merchant's payment address is convertable to BTC or ZEC and all they need to do is import their Syscoin merchant private key into a ZEC/BTC wallet to be able to spend their offer payments.

1.10. Offers and decentralized marketplace

We have developed a marketplace where you can securely and reliably buy and sell any items you wish. Entire stores can be created directly through the marketplace where you can sell your own products or re-sell others products for commission

1.10.1. Offers quantities. All Vendors have different requirements for inventory control. In order to satisfy as many use-cases as possible, we have implemented both finite and infinite quantity controls. A vendor can enter a finite inventory of 1-x which matches their physical inventory and that inventory will be reduced by -1 on every completed offer purchase. If the offer is digital in nature or there is an unlimited physical supply, the vendor can enter"-1" which will indicate and allow unlimited offer purchases.

1.10.2. Alias rates peg. The sysrates.peg alias is used by default in all offers as it will be managed and updated by the team providing fiat and BTC or ZEC price updates regularly based on a metric of volatility and time. Setting the currency of an offer looks up the conversion rate at the time of sale and applies it in taking coins from the consumer sending to the merchant. However since the offer consensus code can look up what price peg was used and at which block height, it has the ability to detect that a correct payment was made at any given time. This means any other nodes synchronizing from a previous block will be able to deterministically detect payments and discard those that do not pay enough from bad actors. Because the relationship between the conversion rate that the sysrates.peg will use holds over time it can be thought of as a traditional fixed exchange-rate pegging system (to SYS) [Wik]. If the consensus code was not there to maintain the relationship of the asset price to the Syscoin price at the time of payment, then there would be no correlative relationship and would be a simple conversion-at-purchase tool. It is much more than that and deterministic payments across multiple assets including custom ones will allow for flexible payment options and help ease the transition for consumers and merchants from traditional marketplace environments to cryptocurrency ones. The whole goal of the rates peg feature is to provide convenience, should demand for a new fiat token or asset arise, not only can we provide that easily in the sysrates.peg at our convenience by updating the alias public data but anyone else can create an alias peg and have others associate their offers with it. This makes it a trust-less design that doesn't depend on the teams rate peg which is just done for bootstrapping users with defaults that will make life a little easier for those who are getting used to the decentralized marketplace and how it works.

1.10.3. Offer currencies. When creating an offer on the marketplace you can choose which currency the item should be priced in. The price is internally stored in Syscoin amounts and converted to display by the conversion rate defined in the peg rate alias associated with the identity that is used to create the offer. It is important to note that payment happens in either BTC, ZEC or SYS but not in any other currency that is simply used for the convenience of price display. If the rates peg is updated, the display price will adjust for any new conversion rate.

1.10.4. 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 that are redeemed for some service by the buyer of the offer. The certificate may be automatically transferred to the buyer upon completion of sale.

1.10.5. 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 of the offer controls the whitelist and can add a discount level on a per entry basis for each reseller. If the merchant sets his offer to private, then end-users must purchase the item through one of the participating resold offers.

1.10.6. 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) representing satisfaction of the sale from 1 being the least satisfactory to 5 being completely satisfied and recommending the user to others. Ratings and feedback can be given to and from arbiters, merchants and buyers. Up to 10 feedback's are allowed to be left per sale for each type of user and are public for anyone to see via the Syscoin API. For a normal sale, buyers and sellers may rate each other once and leave up to 10 pieces of feedback for each other. Ratings and feedback help bolster the reputation of those involved in the marketplace to increase acceptance of those users amongst each other. It is an integral part of a system that is based on global users who may never interact with each other physically.

1.10.7. Multiple payment options. Syscoin currently offers 3 payment options which can be used in combination. Syscoin, ZCash and Bitcoin are currently the three offers options for payment. Syscoin is the native token and as acceptance of the network grows, the token of choice for payments. However to achieve network effect Bitcoin was added which has the highest liquidity of any cryptocurrency. It allows a vast community of users to use Syscoin services with little to no cross-chain configuration. ZCash is helpful for anonymous payments and was also added in the same way Bitcoin was. The private key of the merchant of an offer is the same private key used for payments in Bitcoin and ZCash. The ease of use and convenience it provides makes this feature a key part of the potential growth and network effect for Syscoin services. The low barrier of entry makes it likely that these communities will use Syscoin services and recommend others to use it. It is important to note that the design of the payment options is trust-less. A trustful design would have been to allow payments in other currencies or tokens by exchanging to a desired token of choice by the merchant. The obvious drawback is that trust must be placed on the exchanging medium to convert tokens as well as suffering conversion and slippage making it less enticing to use any token other than Syscoin. This makes it an ideal choice for external communities to use services without suffering loss through conversion and security breaches which may reduce margin to become unprofitable for merchants in the long run.

1.10.8. Private payments via ZCash. Because Syscoin addresses are compatible with ZCash Transparent addresses we can offer ZCash support with complete multisignature support to allow for optional Syscoin escrow functionality. A merchant has the ability to select a combination of payment options from a list of SYS, BTC or ZEC. Once a buyer tries

to buy the offer they will see the payment options available. Once they select ZEC, a Transparent ZCash address will be generated which uses the same private key as the merchants Syscoin address. The buyer would pay from a private ZCash address to preserve anonymity and the seller would import their private key (at a random time in the future) into their ZCash wallet to claim their funds, and send those funds into an input of a JoinSplit transaction to store them in a private ZCash address, creating a JoinSplit sandwich.

The JoinSplit sandwich can be summarized as follows z-addr to P2SH t-addr to z-addr which provides the maximum amount of privacy with multisignature support. All signing of ZCash transactions within the Syscoin escrow service happens on the Syscoin network and is posted onto the ZCash network upon escrow completion. This offers great usability within a trust-less and decentralized payment design.

The "sandwich" of Pours around a t-addr transaction however this leaks the amounts involved, which an adversary could then correlate with other information such as timing and transport layer metadata. The correlation can be easily voided by randomizing when to complete the second Pour to complete the "sandwich" by sending the t-addr funds to a privacy preserving z-addr.

The process of remaining entirely anonymous is easier for the buyer than it is for the merchant. The buyer simply sends coins from a private ZCash address and retains anonymity. The merchant's process involves generating the merchant Syscoin alias through the use of a faucet (This preserves the merchant's identity since there is no way to link a merchant to a physical identity through the use of a faucet). The involved parties would operate over TOR which is supported by the Syscoin core wallet for security at the network packet layer, to avoid giving up the buyer's or merchant's physical location. Once they create offers and get paid by a merchant which pays from a private ZCash address to the merchant's public Syscoin address. The merchant would then import his private key into a ZEC wallet and convert those funds back into a private ZCash address at a random time by doing random amounts until the full amount is converted (perhaps by simply waiting for multiple sales to accrue before converting to z-addr funds). Doing so prevents a timing analysis attack which allows third parties from linking the payment to the merchant to the private ZCash converting process of the same amount some time later. This timing attack may be relevant only to those merchants that repeatedly sell offers for the same amount of coins and send those coins to private ZCash addresses with some uniform pattern of time and amounts. By simply waiting for multiple sales and/or splitting transfers to private ZCash addresses randomly, timing attacks on the leaked amounts can completely be mitigated and full complete anonymity retained for the merchant.

1.10.9. Shipping notification system. A payment acknowledgement button on escrow and offer payments allows a multi-use notification system to the buyer that either the merchant acknowledges payment and/or they are about to ship the product and give the notification to the buyer to expect the goods or service soon with a tracking number sent via the encrypted messaging system. Of course there is no requirement of how the acknowledgement is to be used or if it needs to be used at all, it is a convenience feature that allows the merchant to acknowledge and perhaps notify the buyer of actions that the merchant has taken upon sale. Perhaps merchants will adhere to a convention of how to use this feature as the network grows and becomes adopted by more users.

1.10.10. Categories. Offer categories can be anything the user specifies. However the notable exception is if you are selling a certificate they must be a certificate category or sub-category. Default categories are defined in the syscategory alias and updated by the Syscoin team to provide a dynamic mechanism of updating categories in Syscoin based on market demands.

1.10.11. "Wanted" section. If you have an item or service that you are looking to buy but cant find it on the marketplace you can list it in the wanted section. The wanted section will have multiple subcategories such as Wanted Services or Wanted Items etc. If a seller matching your needs sees your wanted listing they can simply create an offer fitting your wanted request and get into contact with you via the Syscoin encrypted messaging service to proceed with the transaction. This makes a bidding market that will allow multiple merchants to contact the buyer and have the buyer choose the desired offer to purchase the wanted item. The buyer will choose between merchants with the lowest price with the best feedback, thus it may not always be the lowest price wins. The merchants may also add discounts to the specific customer if the offer is intended to be a live public listing available to anyone. This would be done through the whitelist mechanism of the offer if they wish to reduce the sale price for the specific buyer for whatever reason (most likely wholesale purchases where more than 1 qty is sold).

1.10.12. Private offers. Private offers are useful for hiding offers from public viewing and searches. They also have a role for when a merchant of an offer acts as a wholesaler to whitelist affiliates. If the wholesaler sets his offer to private, the affiliates can themselves have their commission based offers set public however buyers will not be able to directly purchase from the wholesale offer directly but must operate a sale through one of the affiliates of that offer. This mocks the real-life wholesaler/reseller/distributor model where affiliates and wholesalers come to agreement on discounts and commissions and proceed to sell goods on the marketplace without worrying about being cut out by buyers directly purchasing from the wholesaler because it would be cheaper to do so in most cases.

1.10.13. Marketplace moderation. Marketplace moderation is done through the SafeSearch feature which allows for 3-tiers of moderation which is affected by the use of a

sysban alias that the Syscoin team owns. The users of the network are able to set services to SafeSearch but if they are creating content not suitable for viewing and not using SafeSearch then the team can moderate these such pieces of data to remove from public viewing. It is important to note that the sysban moderation does not disable the services from use on the network just that it becomes publicly unviewable and omitted from searches similar to how private offers work. The sysban alias itself is a multisignature alias which ensures that not one but a group of people on the Syscoin team need to arrive at a consensus regarding filtering content. This can later extend to a committee of people that can act as guards of safe viewing which would have the sole jobs of applying moderation or reviewing moderation activities.

1.10.14. Offer geo-location. All Syscoin offers include a placeholder for WC3 standardized (longitude/latitude) data. Because of the global nature of the blockchain, there are many scenarios in which Syscoin services would require a geolocation. A user on the marketplace may wish to search for offers within a defined radius of their current geolocation, look for offers in a location they are travelling to, or look for offers within their country for shipping concerns, and additional use-cases. The current desktop application does not enter this data automatically, as desktop computers normally do not contain the hardware required to retrieve GPS data. While this data could also be obtained from IP address and Service Provider data it would not be very accurate. This data would currently need to be entered manually. It has been included in the 2.1 release to allow development to continue on mobile/web applications which will have access to GPS data.

1.10.15. Dynamic fees. Escrow arbiter and transaction fees for transactions are part of the rates peg alias which are updated by default in the sysrates.peg alias. Merchants are free to use whichever rates alias suites their needs but buyers may choose not to purchase from merchants using offers connected to rates aliases which are not maintained properly or have ludicrous fees. Services which need to look at historical fees to determine payment amounts like escrow and offer payments look up what the fees were at the time of payment. Transaction fees are used for escrow related transactions which depend on what the market rate for mining fee are at the time of sale. They are set to Satoshi per byte amounts. For example a Bitcoin external payment must pay at least 60 Satoshi per byte otherwise parties involved in the sale will be waiting for hours if not days for the transactions to confirm. On release of Syscoin 2.1 the Bitcoin transaction fee is set to 75 Satoshi per byte in sysrates.peg for quick confirmation of payments.

1.11. Messages

Encrypted messages use asymmetric cryptography to send data to alias public keys. The identity system plays a key role in messaging because senders and recipients aliases are used to determine the keys for encryption. The sender and recipient keys are encrypted with the message so that no third parties can read the data transmission without having the private key of either of the parties involved. Multiparty encryption is also possible through the use of multisignature alias identities. See section 1.7.6 Multiparty encryption.

1.11.1. Send raw hex. Because of multisignature aliases, we needed a way to be able to transfer raw transactions to other signatories of that alias in a private and efficient manner. Utilizing the encrypted messaging system made sense but the problem is that there are only about 6300 bytes allowed to go into a Syscoin service transaction and the message needs to be stored twice, one encrypted to the sender and one encrypted to the recipient. Because raw transactions are stored in binary it would make sense to be able to send raw data which can be decoded as hex by the recipient allowing transactions as big as 6000 bytes to be encrypted and sent as raw binary data and decoded as hex by the recipient when using to sign with the multisignature signing tool in the alias screen. Sending as raw binary data instead of hex halves the size of the data transmission size as hex is represented by 2 characters for each byte. When this option is checked, the raw binary message is encrypted to the recipient only and not the sender to save space. The recipient can see the message and copy it and sign the transaction safely and securely. UTXOs that are used by that transaction are locked on the senders wallet to avoid the sender inadvertently spending those outputs on other spending transactions before the signatory signs and posts the raw transaction to the network.

1.12. Blockchain pruning

Bitcoin has an option pruning feature which is quite different than what we describe here. Perhaps Segregated Witnesses(Segwit) is the closest related thing to Syscoin's pruning mechanism because it saves bandwidth as well as storage costs. Just as Segwit splits transactions into two with just a hash of the Witness transaction that is carried forward by SegWit compliant clients for consensus validity, Syscoin's pruning mechanism works similarly with service transactions by splitting the Syscoin service transaction into 2 outputs. One is the ownership provable output which links the service to a public key that is capable of modifying the service linked to the information in the output. It is a small scriptPubKey which carries just the important information needed to prove that you own a certain alias. Every other service is linked to an alias which extends provability to services outside of aliases. In the figure below you can see the two outputs. Output 1 has just the alias output which linked to an owner public key. The OP code denotes the type of service it is, a name and guid to be able to lookup the alias from the Syscoin service DB and a hash of Output 2 which is the data carrying OPRETURN representing the data in the Syscoin transaction. Offers, certificates, messages, escrow all create similar Output 1 style outputs with different OP codes, but they all must have an output for

the alias which proves that the owner of the alias is the one making the transaction linked to any service. The consensus code will extract the data from Output 2 and check to see that the hash matches from Output 1 to ensure integrity of the data from data mutation attacks. Doing so will let us effectively not have to hash the contents of Output 2 inside of our blockchain transaction. The data must be available on demand inside of the database and it remains so until expiration where it is assumed it will no longer be needed because updates to services are disallowed if expired. A combination of using prunable outputs with expiration of services allows us to create a unique pruning mechanism that will save new nodes syncing from having to download and store expired service data. From preliminary tests run on node4 of our unit test suite shows that data savings are remarkable. Node4 is the node that is set to txindex=1 which disables the Syscoin pruning mechanism. As the unit tests run this node will save all of the pertinent service data inside of its database as per design. In a later test, we've run it as txindex=0 meaning pruning is activated and synchronized a new client to it. Since most of all services were expired running the unit tests the new node synchronized very little data from node4. After synchronization of the blockchain was complete we noticed the data directory size of node4 being 335Kb while the new node only 470 bytes while still maintaining complete protocol consensus.

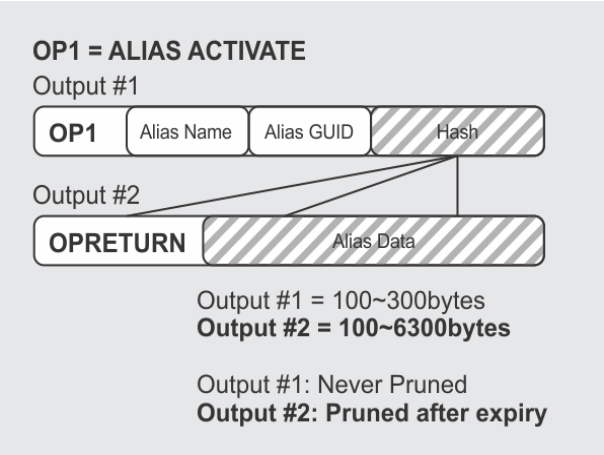

Figure 2: Syscoin OPRETURN data hashed into UTXO

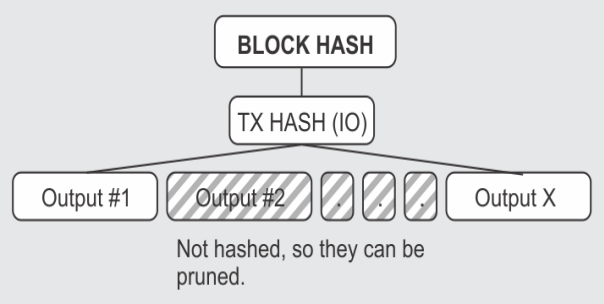

Figure 3: Syscoin Outputs not hashed by blockchain

2. Future work

Depending on the demand for Syscoin services there are some useful features that can be added with minimal effort but intentionally left out due to time constraint for the 2.1 Core release. This would invalidate all regression testing and would require further unit tests for code coverage purposes. There is some work to integrate Syscoin services to complement Segregated Witness functionality which helps scale the transacting mechanism of Syscoin with sending and receiving coins. Although Syscoin Core supported the functionality of SegWit it was left out intentionally until we develop unit tests surrounding the testing of this feature to ensure integrity across service usage.

Perhaps a use-case we have to research for the use of SegWit would be to allow updating Syscoin services offline, in case people want to make many small edits and then finally post onto the network UTXO database once finalized. This would allow the amount of data carrying outputs to be minimized to the data that only needs to be settled for others to use when using those services.

We are also investigating using quantum resistant signature schemes. Perhaps switching Bitcoin's native ECDSA which may be vulnerable to attack from the NSA [Paca] to a Merkle Signature Scheme such as the Improved Merkle Signature Scheme (CMSS) which is faster(up to 10x faster than ECDSA) and quantum resistant. A Merkle Signature Scheme combines the one-time signature scheme (either Lamport or Winternitz) with a Merkle tree (also called a hash tree). This allows us to use one public key to sign many messages without worrying about compromising security.

2.1. Messages

Currently anyone on the network can send anyone else messages. It would be useful if certain identities disallow the general public from messaging to them so they would not have to filter through messages they do not care for. This is a trivial change that involve only allowing replies with previous message inputs attached if some option is set in the recipients alias identity settings.

2.2. Auctions

An auction type offer would allow for applications such as real-estate and Ebay style sales to take place. The three auctions types would be Absolute Auction, Minimum Bid Auction and Reserve Auction. Note that Bidding Fee auctions may make sense once the Lightening Network project is released in conjunction with segregated witnesses for payment-channel style interfaces to the blockchain in the context of offers.

2.3. Escrow

We felt that Deposit-less escrow was the better option for mainstream adoption but for the best consumer protection,

double-deposit escrow (DDE) may offer more piece of mind to both sides of the party who wish not to rely on arbiters even with positive feedback. This type of escrow requires some upfront deposit of funds which may or may not be destroyed on contractual disagreement. Because of this it offers incentive for both sides of the party to complete the agreement under mutual agreement. However it does take away incentive to use it because it requires upfront deposits. It also would not work without some sort of time-bomb extension so that funds aren't destroyed inadvertently by absent stakeholders. This complicates the design and we felt that simpler is better in regards to escrow especially since feedback becomes helpful with arbitrated escrow. It may be useful for mid-valued transactions that offer better value of security. DDE can simply be an option for escrow and can be added as desired if the market shows demand for it. Allowing any other crypto-currency to be able to use the escrow service along with payment options through the offer service is also something to look at and do an analysis on based on the volume and demand of some coins as they move up the ranks. With the flexible design of Syscoin escrow it makes it fairly trivial to support other coins, especially those that share the same private key format as Syscoin (like Bitcoin and ZCash taddr's do).

2.4. Aliases

Alias identities can make use from wallet-less transactions that allow payments and updates to service to take place without the need for users to have access to their wallet. This may make sense in shared key environments where the user may wish to login (with their alias identity password) to a portal managed by an identity or certificate issuer and the user may use Alias Authentication to prove ownership of an identity and use their Alias Balance to send a sign a spending transaction. There is more information on this in the Syscoin Use-Case paper under Future Use-Cases.

2.5. Certificates

Using torrent trackers and other P2P style hosted data source is something to research as it will allow for scaling certificate data above the limits of 1KB of data for private encrypted information while maintaining security through the network. Storing data can be encrypted to the public key of the certificate owner which is the alias identity it is assigned to. This way data can be hosted in public rather than on private servers that are maintained with rigorous security to avoid breach of access.

3. Specs

There is a 888 million maximum coin limit. 1 minute block time. Proof-of-work SHA-256 merge mineable (majority of network security coming from Bitcoin). Syscoin 2.0 had a block reward of 54.13. Syscoin 2.1 which was released on December 18, 2016 represents a "halving" event in block rewards reducing them to 16.39 per block (a reduction of 330 percent).

The mining rewards, designed to be gradual and smooth, end at about 800 million coins (block 24177646 will happen near year 2052) and thereafter supply is inflated via the Syscoin mechanism design of the inflation/deflation system assuming services are in high demand. As described in section 1.5.1 if the number of Syscoin service transactions is 5 or less the mining algorithm will burn the service fees and above 5 will start to inflate the fees to the supply and give to the miners. This way the supply can accommodate high demand for usage on the network without relying entirely on traditional fees for miner incentive to produce blocks honestly. Currently the block size being 1MB and maximum service data size being about 6300 bytes it will equate to roughly 158 services per block possible at full block capacity. Since blocks are produced every minute that works out to 2.6 transactions per second (TPS). If SegWit or Lightning Networks improve the TPS such that it becomes 100 TPS or more, we may have meaningful inflation statistics that affect the network supply in a noticeable manner. At current rates the inflation or deflation produces using the algorithm remains negligible until future innovation allows the design to be complete. For now the total usable supply as described by the miners will be 800 million with possibility of future expansion up to 888 million once we are able to improve the transactions per second of Syscoin services. SegWit will be possible upon the Syscoin 2.1.2 release.

MINING SCHEDULE

Block Height	Mining Reward	Block Height	Mining Reward
< 525601	16.39	13140025	10.33
1051202	16.63	13665626	10.4
1576803	16.88	14191227	10.48
2102404	17.14	14716828	10.56
2628005	17.39	15242429	10.64
3153606	17.65	15768030	10.72
3679207	17.92	16293631	10.8
4204808	9.09	16819232	10.88
4730409	9.16	17344833	10.96
5256010	9.23	17870434	11.04
5781611	9.3	18396035	11.13
6307212	9.37	18921636	11.21
6832813	9.44	19447237	11.29
7358414	9.51	19972838	11.38
7884015	9.58	20498439	11.46
8409616	9.65	21024040	11.55
8935217	9.73	21549641	11.64
9460818	9.8	22075242	11.72
9986419	9.87	22600843	11.82
10512020	9.95	23126444	11.9
11037621	10.02	23652045	11.99
11563222	10.1	24177646	12.08
12088823	10.17		
12614424	10.25		

Figure 4: Syscoin mining schedule

4. Conclusion

We have presented a set of hardened smart-contracts that can be used in conjunction with each other and an identity system to provide blockchain-based e-commerce solutions for small, medium and large businesses. The processes used by businesses and entrepreneurs may transfer to Syscoin without the need to re-invent the way people work today. The goal is not to force the technology and processes on the people using it but to bring people to the technology who are in need of a blockchain-based solution to their problems. The mix of unique features of Syscoin in an architectural framework that enabled high security through merged-mining and low inflation enabled trust-less payments and services to be used today in commercial ventures and partnerships as well as provide an investment proposition to holders of Syscoin token holders. A low barrier of entry for external communities can be leveraged to help create a network effect for Syscoin and its services.

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. Without the work of these people none of what we have worked on with Syscoin would have been possible.

A special thanks for the Blockchain Foundry Inc. team of Sebastien Dimichele, Chris Marsh, Brad Hammerstron, Willy Ko and Dan Wasyluk for their peer review and updates.

References

[Nas02] John F. Nash. "Ideal Money". In: *Southern Economic Journal* 69.1 (2002), pp. 4–11. DOI: http://www.jstor.org/stable/1061553.

[Nar16] Arvind Narayanan. "On the Instability of Bitcoin Without the Block Reward". In: *ACM CCS* (2016). DOI: https://www.cs.princeton.edu/~smattw/CKWN-CCS16.pdf.

[But] Vitalik Buterin. *DAO Vulnerability*. URL: https://blog.ethereum.org/2016/06/17/critical-update-re-dao-vulnerability/.

[Cry] Cryptoid. *Syscoin 2.1 block explorer*. URL: https://chainz.cryptoid.info/sys/.

[Dai] Philip Daian. *Chasing the DAO Attackers Wake*. URL: https://pdaian.com/blog/chasing-the-dao-attackers-wake/.

[Deva] Bitcoin Core Developers. *Bitcoin reference client*. URL: https://github.com/bitcoin/bitcoin.

[Devb] ZCash Core Developers. *ZCash*. URL: https://github.com/zcash/zcash.

[Fin] Hal Finney. *secp256k1*. URL: https://bitcointalk.org/?topic=2699.0/.

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

[Nie] Jakob Nielsen. *Nielsen's Law of Internet Bandwidth*. URL: https://www.nngroup.com/articles/law-of-bandwidth/.

[Paca] Chris Pacia. *Bitcoin vs. The NSAs Quantum Computer*. URL: http://www.bitcoinnotbombs.com/bitcoin-vs-the-nsas-quantum-computer/.

[Pacb] Chris Pacia. *NSA Backdoors and Bitcoin*. URL: https://chrispacia.wordpress.com/2013/10/30/nsa-backdoors-and-bitcoin/.

[Per] Nicole Perlroth. *Government Announces Steps to Restore Confidence on Encryption Standards*. URL: http://bits.blogs.nytimes.com/2013/09/10/government-announces-steps-to-restore-confidence-on-encryption-standards/?_r=1.

[Wik] Wikipedia. *Fixed exchange-rate system*. URL: https://en.wikipedia.org/wiki/Fixed_exchange-rate_system.

[Wil] John Williams. *Shadow Government Statistics*. URL: http://www.shadowstats.com/.