8x Protocol

Decentralized payment processing network on the Ethereum blockchain

Kerman Kohli, Kevin Zheng, Kevin Lu
kerman@8xprotocol.com, kevinz@8xprotocol.com, kevinl@8xprotocol.com

October 4, 2018

Abstract

The following paper outlines how the 8x protocol facilitates a payment processing network through a network of decentralized 8x service nodes and the 8x token. Currently, payments on Ethereum are very limited in their use case. 8x is designed to tackle this problem by introducing the ability to make repeating and pre-authorised payments through a novel layer 2 consensus mechanism. Payment rules are coded on-chain and customers can agree to them directly. To execute the transaction, payments are claimable to a network of Service Nodes who in turn receive a percentage fee of the original subscription payment made between the consumer and vendor. In order for 8x Service Nodes to make payment claims, the 8x native token must be staked.

This paper focuses on the first use case of 8x's decentralized payment processing network, subscription payments. Details of further use cases are outlined in Section 6.

Contents

1 Introduction **3**

2 The Subscription Economy **4**
2.1 Overview . . . 4
2.2 Challenges Faced . . . 4
2.3 Opportunity for Blockchain . . . 5
2.3.1 Overview . . . 5
2.3.2 Existing Work and Gaps . . . 5

3 8x Protocol **7**
3.1 Functionality Overview . . . 7

4 Layer 2 Consensus **8**
4.1 Problem . . . 8
4.1.1 Our Solution: Proof of Priority . . . 8
4.1.2 Processing Payments . . . 9
4.1.3 Beginning Minimum Stake . . . 9
4.1.4 Mathematical Model for Minimum Stake . . . 9
4.2 8x Token Use Case . . . 11
4.3 Clarifications . . . 11

5 Smart Contracts **12**
5.1 Architecture . . . 12
5.2 Transfer Proxy . . . 12
5.3 Executor . . . 12
5.4 Transaction Registry . . . 13
5.5 Collectible . . . 13

6 Payment Network **14**
6.1 Use Cases . . . 14
6.2 Pre-authorised Payments . . . 14

7 Future Improvements **15**
7.1 Support for other Stable Coins . . . 15
7.2 Cross-chain support . . . 15
7.3 User Interface . . . 15
7.4 Integration with Enterprise Resource Planning Apps . . . 15
7.5 Regulation . . . 15

8 Summary **16**

9 Glossary **16**
9.1 MakerDao . . . 16
9.2 State Channels . . . 16

10 Acknowledgements **16**

11 References **17**

1 Introduction

Cryptocurrencies were introduced to the world in 2008 when Satoshi Nakamoto published the Bitcoin whitepaper. The key innovation behind it was the solution to the double spend problem through cryptographic proofs. Bitcoin's first application was the ability to make cross-border payments to individuals globally through a trust-less network of miners. To make a payment, the sender signs the transaction with their private key and broadcasts it to the network. This makes the execution of a cryptocurrency payment "push" based as money is transferred from one party to another without any intermediaries.

In a traditional centralised banking system the consumer thinks they're paying a vendor directly. Instead, they're actually authorising the vendor to "pull" funds from their bank account directly.

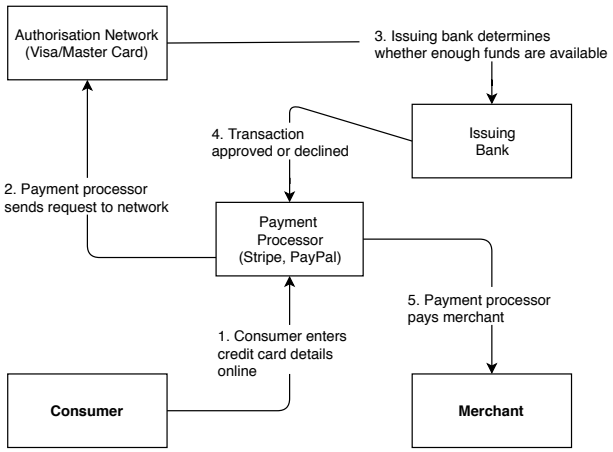

Figure 1: A figure of the existing centralised banking "pull" system. Unless the issuing bank gives approval, the transaction is not made.

There are many benefits of eliminating intermediary parties, the fundamentals of cryptocurrency "push" based payments make it difficult to pre-authorise transactions for the purposes of recurring payments. To make a pre-authorised recurring "push" payment system, a party is required to initially trigger the transaction.

A problem of using crypto-currencies for recurring payments is the volatility of the price. For any merchant, whether they prefer to deal in fiat or cryptocurrency, paying in a currency like Ether is an inadequate transfer of value due to the volatile nature of the market. Potential solutions to this problem include using 3rd party oracles to retrieve the latest exchange rate, although this places high dependency in external parties. It also introduces a potential threat to the accuracy and reliability of the system to ensure fair exchange rates are used to facilitate the exchange of goods and services.

Date	**ETH/USD**
First	$
April 2018	396
February 2018	1126
January 2018	747
December 2017	443
November 2017	306
October 2017	301

Table 1: Price of Ether between October 2017 and April 2018

2 The Subscription Economy

2.1 Overview

Initially coined by Zuora, the Subscription Economy represents the idea that customers are happier, subscribing to the outcomes they want, when they want, rather than purchasing a product with the burden of ownership. This change means that traditional companies need to adapt towards a new way of thinking in order to best capitalize on this market. Traditional functions such as pricing, marketing, sales, finance and culture will need to shift to one of value pricing, experience, selling outcomes, customer lifetime value generation, and deep relationships. The new era sees a shift from traditional product focused companies to the emergence of new business models based on deep relationships.

According to Gartner, by 2020 more than 80% of software providers will have shifted to a subscription based model. Further to this, IDC predicts that 50 of the world's largest enterprises will see the majority of their business depend on their ability to create digitally enhanced products, service and experiences. Preparations need to be made on the ecosystem supporting the phenomena of the Subscription Economy in order for it to reach its peak. Studies by Credit Suisse have further shown that within the United States alone, 2015 was a year $420 billion was spent on subscriptions as compared with the year 2000 which say $215 billion. Studies by McKinsey have shown that by 2020, the global payments industry will generate an estimated $2.2 trillion in revenue over $400 billion more than the figure for 2015 ($1.8 trillion) due to an average growth rate of 5 percent. Using these figures, we can observe that in 2015, Subscription services had a market capitalization of 23.3%. There is no accounting for the current growth rate in Subscription Services. It can be safe to assume that the subscription services in the United States alone would account for a market value of $513 billion.

The Subscription Economy is an observed phenomena which is positioned to overtake traditional business models and revolutionize the way value is exchanged on a global scale.

2.2 Challenges Faced

Despite the exponential growth resultant from the phenomena of the Subscription Economy, several challenges and limitations still remain for all stakeholders involved. The facilitation of payments are currently being conducted by centralized third parties placing increased dependency on their services. Further to this, both businesses and consumers face challenges in managing the end to end business process of facilitating the exchange of value within the Subscription Economy.

From a a business perspective, some of their challenges include:

1. Dunning management and incident response management
   Dunning management is the the terminology given to requesting payments from customers after their payment expected date. Often, these issues have been caused by simplistic reasons such as the incorrect input of credential details or the expiration of a customer's payment details. However this leads to a loss in productivity and unnecessary disruption to a consumer's experience. The broader business issue is the connection between incident management and subscription services. As a consequence, businesses are left with a disparate customer view which leads to a poor experience.

2. Consolidated customer view
   The Subscription Economy is built on the exchange of trust between an enterprise and customer which has evolved from traditional exchange of product. This is why it is increasingly important for businesses to maintain a consolidated view of their customers in order to provide tailored value. This could be implemented through the use of a customer relationship management system, however the integration between financial systems and being able to provide tailored scenarios remains a challenge.

3. Security of payment
   With the future view that there will be an abundance of micro-transactions, it is increasingly important to maintain the security and integrity of all payments made.

4. Multi-language and multi-currency support
   The Subscription Economy is still in its infancy. In order to enable a global Subscription Economy, language and currency protocols need to established to facilitate this ecosystem. This will further enable enterprises to reach global audiences and provide value to customers who were initially unable to access their platforms due to language or currency barriers.

From a consumer perspective, some of the challenges include:

1. Multi-language and multi-currency support The Subscription Economy relies on the removal of barriers whether it be information through language or the exchange of value through different currencies. Existing customers are currently restricted to only those enterprise value providers that accept their local form of currency. This means customers are not able to access the full range of global enterprise value providers on offer.

2. Single view of all subscriptions The future of the Subscription Economy will for-see the rise in subscription based services an individual customer may consume. Existing customers currently face pain points around the management of their subscription services. Other than a debit amount which can be seen within their respective bank accounts, customers will need to access the individual Enterprise value provider in order to check their current status. To prepare for the Subscription Economy, these processes need to be simplified so that effort of both customer and value providers may be directed towards the exchange of value.

3. Incident Response The Subscription Economy signifies the transition of customer behaviours from one of ownership to one of access. If there are any issues which may arise from the customer's ability to access value, their trust and relationship will decline proportionally to the resolution capabilities of the value provider. This applies beyond the Subscription Service and is a challenge traditional business models also face.

Subscription Economy challenges experienced in present society need to be considered holistically in order to identify opportunities for improvement.

2.3 Opportunity for Blockchain

2.3.1 Overview

The subscription economy presents unique opportunities for global challenges to be solved. This includes smart city innovations, news legitimacy, renewable energy, cyber security, global crisis support and access to global education. Within the subscription economy, these issues and challenges present unique challenges and environmental factors which are solved through the use of a blockchain. Within these industries, multiple players exist which places increasing pressure on a centralized solution to manage or coordinate. This is where a blockchain could provide disinter-mediated service in order to provide the trust required. With specific regards to payments within the subscription economy, the advent of a blockchain aims to reduce the % of the transaction cost to the payers. This bears significant ramifications as the Subscription Economy has dependency on micro-transactions to occur. As the Subscription Economy grows, there will be significant increase in transaction volume, frequency and consumers. Existing subscription service payment providers are not positioned to scale well on low volume but high frequency transactions e.g. as a transaction amounts decreases the percentage taken as fees increases exponentially. A blockchain solution to facilitate the recurring exchange in value could provide the necessary ecosystem to enable the subscription economy to society. However, there is currently no open platform for maintaining repeating cryptocurrency payments through the blockchain.

2.3.2 Existing Work and Gaps

This section aims to explore the history of existing implementations of recurring payments through blockchain and their limitations.
In regards to end-to-end recurring payment solutions, Coinbase Commerce supports recurring payments for merchants. Although it comes with three limitations:

1. Requires users to have a valid Coinbase account with cryptocurrency stored on their wallet (for users and businesses).

2. Merchants have to store cryptocurrency in their wallets, thus exposing them to the volatility of cryptocurrencies.

3. Only supports Bitcoin which is typically slow to transfer and comes with higher transaction fees compared to currencies such as Ether.

An integral part of 8x's protocol is the use of stable coins such as the Dai. Using it as medium of exchange ensures that businesses and users aren't losing any value due to the volatility of the cryptocurrency markets. Due to its compliance with the ERC20 standard, it can be used to enable "pull" based payments rather than creating a pre-paid subscription escrow smart contract where users deposit their money into at the start of each month. A transfer proxy contract in 8x architecture is able to take the ERC20 tokens directly from the user thus eliminating the need to store them in a single smart contract. The transfer proxy does not contain any business logic.

Part of enabling recurring payments on the blockchain is the repeated execution of a financial transaction between two parties. Recent research into scaling Etheruem has spawned the creation of layer 2 scaling technologies such as state channels. State channels do not consider monthly subscriptions as a way to reduce the burden of one-time payments for subscribers. Creating state channels that require the total subscription up-front and manual top-up may not be economically viable for individuals due to liquidity and inconvenience. Scheduling tasks on the blockchain remains a challenge. A solution is presented through the Ethereum Alarm Clock (EAC) project by Piper Merriam. With EAC, a smart contract can ask the alarm clock service to schedule a transaction at a particular date and provide a reward to the executor of the scheduled transaction. To prevent execution conflict between an increasing number of parties wanting to earn the reward, a claim system is setup. This requires executors to claim the right to execute the payment and then collect the subsequent reward. The earlier the claim is made, the less the total % of the reward is earned. This develops game dynamics where executors are competing against each other - decreasing the likelihood of claim conflicts. The 8x Protocol aims to facilitate a similar ecosystem where network of users collaborate to execute scheduled transactions.

3 8x Protocol

3.1 Functionality Overview

Figure 2 below shows the series of steps taken by businesses, consumers and 8x Service Nodes in order to facilitate the 8x protocol for decentralized recurring payments.

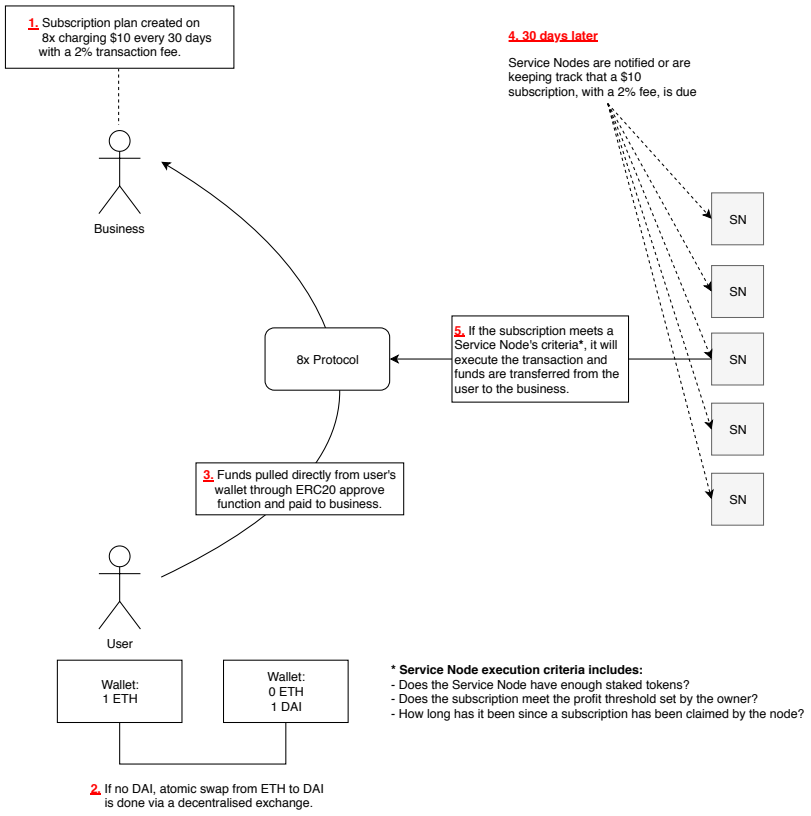

Figure 2: a conceptual diagram of how each party will interact with the 8x protocol

1. Businesses will create a subscription plan specifying the amount (denoted in dollars), frequency and a fee. A fixed fee which may be changed over different subscriptions allows business to specify how important it is for them to receive their payment on time. The higher the fee is set, the shorter it will take to collect payments for the subscriptions as a Service Node's incentive is higher in comparison to lower fee subscriptions and a lower reward.

2. Consumers subscribe to a subscription plan created by the business. This is assuming they already have Dai in a wallet. If they don't, a decentralized exchange can be used to facilitate the atomic swap between Ether and Dai. Stable coins ensure no value is lost due to volatility.

3. 8x Service Nodes can process a subscription once the payment date is due. Higher payment fees, initially set by the business, will be executed first as Service Nodes will want to maximize profitability. A service node processing a subscription for the first time will then have that subscription assigned to them in the future. They will also be required to stake tokens proportional to the amount of the subscription they are claiming. This means that Service Nodes are bounded to how many 8x tokens they have in order to process subscriptions. Failing to execute on an assigned subscription results in a service node's staked 8x tokens given to the first service node to claim the subscription..

4 Layer 2 Consensus

4.1 Problem

To understand the economics and mechanics behind our solution, it is important to understand the consensus problem it solves. Unlike, layer 1 consensus which aims to determine the correct state of a ledger, layer 2 consensus aims to determine the rightful participant for a delegated transaction in order to prevent economically expensive race conditions.

Let's consider a situation where there are two people, John and Mary, who are competing to execute a certain subscription. Suppose John and Mary each spend 75 Wei in Gas to claim a reward of 150 Wei for processing a subscription. Assuming they send their execution transactions at the exact same time - Mary's transaction is included instead of John's.

Mary receives the 150 Wei staking reward while Bob will have received no reward and has also lost funds due to the gas costs of his rejected execution transaction. We can assume that the initial probability of either individual executing the transaction request is 50%. As a result, 50% of the total attempts between John and Mary will consequently cost money - thereby diminishing profit every transaction.

If we introduce eight more claimants into the model, the probability of winning decreases to 10% as only 1 transaction request will be executed out of the pool of 10 claimants, leaving 9 rejected transactions and more funds lost due to gas costs. As the number of claimants increase, the probability of successfully executing a certain subscription exponentially decreases and thus, leading to a gas war.

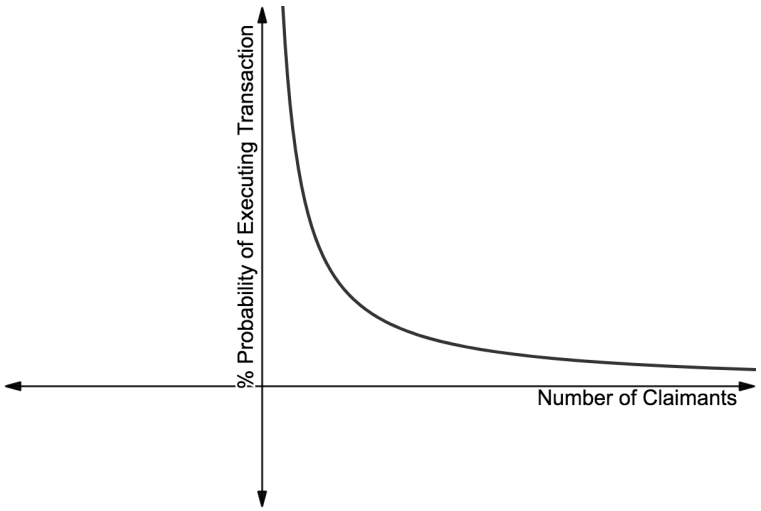

Figure 3: a plot of % probability of successful execution and claimants increasing

4.1.1 Our Solution: Proof of Priority

To solve the claiming problem, we introduce Proof of Priority - a protocol that grants the service node with a higher 8x token holding 'priority' to execute a certain subscription.

According to rational theory, the aggregate behaviour of all service nodes will result in a competition to have slightly a higher amount of tokens than the adjacent service nodes to attain priority in subscription execution. As such, this will produce a 'layering effect' and minimize the probability of token holders with the same holdings attempting to execute the same subscription - thus minimizing the amount of potential claiming conflicts in the system.

Furthermore, the size of the claim window is dependent on the Ethereum network and Gas limit set by the service node. For example, if a service node determines that the reward earned via execution outweighs the Gas fees paid, they may set a high limit to reduce the size of the claim window to increase the probability of their transaction going through.

4.1.2 Processing Payments

During the processing period there will be a claim window with a pool of pending transactions. Service nodes other than the one with the highest token holding will automatically back down in the mempool.

When a service node successfully executes a subscription, a percentage of their total 8x token holding will be locked up. The percentage varies with the Gini coefficient. For example, if the Gini coefficient stands at 0.5, 50% of a service node's unlocked token holdings will be locked up per subscription. The use of the Gini coefficent is designed to encourage a lower inequality as this results in a lower lockup % and therefore reinforcing the layering effect and reducing potential conflicts in the system.

As a significant % of tokens will be locked up upon processing a subscription - our layer 2 consensus model is designed to grant service nodes that initially backed down a higher priority for future subscriptions.

Service nodes will also be responsible to execute the same subscription on the next payment date. Failure to do so will result in the staked tokens being slashed and given to the first service node to claim the subscription.

4.1.3 Beginning Minimum Stake

The minimum stake required for subscriptions will be calculated utilizing the total unstaked tokens and an adjustable divisor. This divisor will be dynamically adjusted in the future to cater for different conditions.

$$minimum\ stake = \frac{total\ unstaked\ tokens}{divisor}$$

In the case that the calculated minimum stake amount is less than the previous payment, the difference will be refunded to the service node or otherwise remain the same for future payments.

4.1.4 Mathematical Model for Minimum Stake

To moderate the economics of the protocol's Service Node network, we introduce the Gini coefficient into an exponentially decreasing function:

$$y = \begin{cases} g^{\frac{x}{n}} & , \ x \geq n \\ \dfrac{(n \cdot g^{\frac{n}{x} - \frac{n^2 - x^2}{xn}})(1000^{\frac{n^2 - x^2}{xn}})}{1000^{\frac{n}{x}}} & , \ x < n \end{cases}$$

Where

y : Minimum Stake Required
g : Gini Coefficient
n : Beginning Minimum Stake
x : Scaled Time Interval

Math in solidity is done entirely using fixed-point. To navigate around the use of fractional values, the functions of y are expressed as a piecewise function with two conditions.

The plotted equation derives similarities to a Lorenz curve, however does not take into account the cumulative density. The lack of precision is due to the computation limitations of the Ethereum Virtual Machine and blockchain technologies, where complexity and storage come at a very high cost.

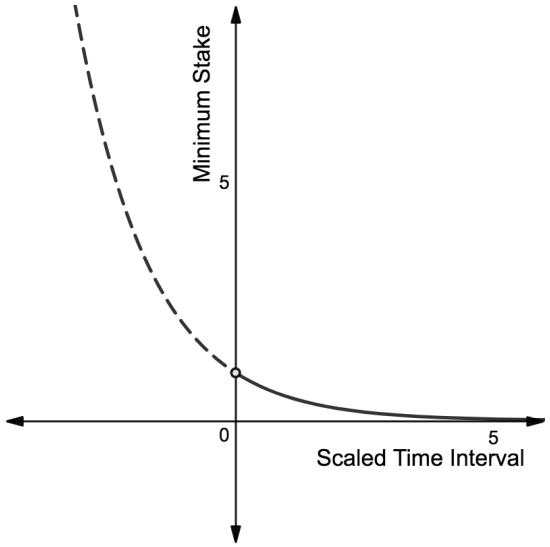

Figure 4: a plot of the most basic graph where the Gini coefficient is 0.5 and minimum stake is 1 token

By observation, as time progresses there will be a smaller and smaller Delta until the plot flattens into a near-horizontal granular line. This results in less conflicts in the system as the minimum stake decreases. Although we are unable to mitigate the risk of conflicting claims entirely, in the case there are multiple transactions for the same subscription, priority will given to the service node with higher token holdings.

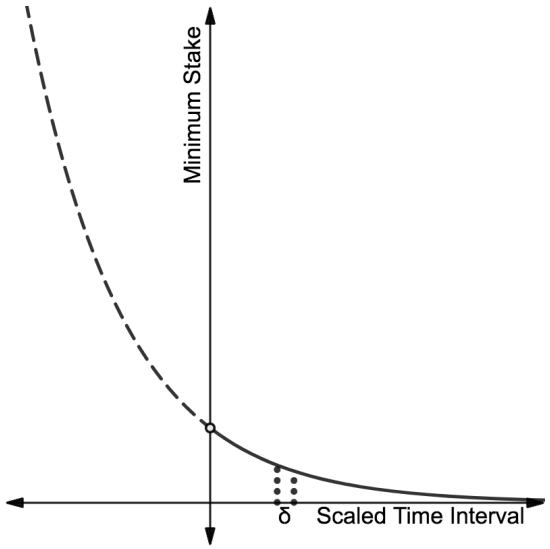

Figure 5: a plot of the most basic graph with δ

4.2 8x Token Use Case

8x tokens will be required to be staked, moving custody of 8x tokens to a smart contract, when a service node executes a transaction
Once a service node executes a transaction for the first time, they have the right to process that subscription in the future too. It is through this mechanism that 8x Service Nodes must carefully specify the parameters for executing a subscription as low fees can result in capital being locked up inefficiently. Similar analogies can be drawn to Ethereum's concept of gas price and computational power. Transactions paid with a low gas fee are unlikely to be executed quickly due to the more profitable transactions available. However if demand is low and blocks are empty miners will include low gas fee transactions as it's better than having no transactions.

Service nodes must execute future payments within the size of the initial claim window (time taken for a service node to execute a subscription for the first time).

4.3 Clarifications

Below are some clarifications that may want to be made to the reader to help understand the scope of the 8x protocol.

1. The protocol is not locked down to any particular stable coin or currency. As long as the token is ERC20 compliant, it can be used with the protocol.

2. Users and businesses do not have any interaction with the 8x token itself. They are only concerned with the currency they are paying in eg. Dai.

3. 8x Service Nodes are rewarded in the form of the currency the subscription uses, not 8x. The native 8x token is only used for having the right to process transactions and preventing 8x Service Nodes from acting maliciously (failing to process a transaction in the future).

4. Gas costs are payed by businesses resulting in a dynamically changing fee structure. 8x Service Nodes should not have to factor standard gas costs into their expected rewards for processing within the 8x ecosystem.

5. Funds are taken out of the users' wallet not an escrow account. This means a business runs the risk of users not paying for the next billing cycle since there may not be enough funds in their wallet when the payment is to be collected. A higher percentage fee will guarantee that goods and services are not given up without enough funds.

6. It could be argued that Ethereum itself could be used rather than 8x to stake and process transactions. However this fails to take into account that the 8x protocol has its own economy where a Service Node's earnings are dependent on a fixed total supply and have a different incentive structure. The fixed supply property is important since it allows economies of scale to occur due to less tokens being required as the number of subscriptions in the system increase. An inflationary currency creates a counter-force to such economies of scale.

5 Smart Contracts

5.1 Architecture

The entire protocol is run on the Ethereum blockchain through smart contracts written in Solidity. Standard gas fees apply to interact with the smart contract for businesses, consumers and 8x Service Nodes. Apart from the transaction fee made during a subscription payment no additional costs are applied. Considerations have been made to ensure expensive operations such as CALL are minimised where possible.

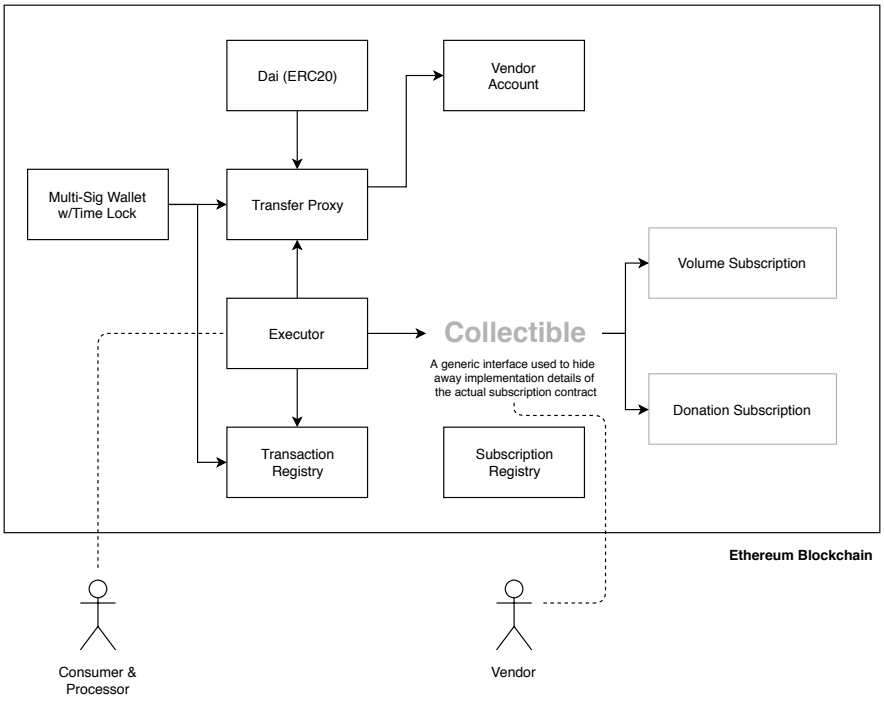

Figure 6: a high level view of how the different smart contracts will interact with each other to create a decentralized payments protocol

5.2 Transfer Proxy

By having a single component, the transfer proxy, which is authorised and responsible for taking and making payments we can store the logic for these payments in the executor contract. While the logic for payments should be tied to the information returned from the collectible interface, having a hard coupling could lead to costly immutability down the line. The transfer proxy has an array of authorised addresses which grant access to pull funds from users and pay businesses although this is controlled by a multi-signature wallet with a time lock of 2 weeks to propagate changes. The only exception to the time-lock is to kill the contract to revoke access to user funds in the case of an attack By having multiple authorized addresses, a new executor contract can be deployed and the old one can be deprecated. When the transfer proxy is called, it uses the ERC20 transferFrom() function to send Dai directly from the user's wallet to the merchant.

5.3 Executor

The executor component is where the core logic and functionality of the smart contract lies. Consumers and 8x Service Nodes interact with it directly in order to claim and make payments. Since all subscription contracts adhere to the collectible interface, the logic for how much money should be charged and whether the subscription is valid is in the actual subscription contract. The executor simply interacts with the exposed public methods. The only extra power the executor has is to cancel a user's subscription in the case that they don't have enough funds. When a user subscribes to a subscription, the executor calls the transfer proxy to facilitate the transaction and adds the payment to the transaction registry.

5.4 Transaction Registry

Once the first payment is made by the consumer (when they subscribe), the transaction is added to the transaction registry which creates a data object for the next payment. When a service node processes a subscription for the first time, the subscription is then assigned to them for future payments too.

5.5 Collectible

Initial plans for the architecture included a separate subscription contract and plan contract for businesses and consumers to interact with. Although this kind of rigidity runs into problems quickly when something like a donation subscription contract needs to be implemented as the user is in full control of how much they want to give. For this reason a more general purpose architecture has been made which facilitates the addition of new subscription contracts as long as they adhere to the interface. Currently the interface methods include:

1. Check whether the subscription is valid
2. Get the subscription owner's balance
3. Return how much the subscriber owes from their subscription
4. Terminate the subscription if they don't have enough funds

If a user doesn't have enough Dai to pay for their subscriptions, an email will be sent to them to remind them to top up. These email details and reminders are to be hosted on a centralised server due to the unwanted nature of publicly exposing an email address to public key on the blockchain.

6 Payment Network

6.1 Use Cases

Infrastructure is essential to building consumer-grade applications required to push blockchain usage and adoption forward. 8x will allow the following business models to be made possible:

1. B2C businesses being able accept crypto-currencies for their monthly subscription payments and paid out in a stable coin or fiat via an off-ramp
2. dApps charging their users a fixed amount every month in ETH or an ERC20 token of their choice.
3. Give approval for an app on your phone to spend up to $30/from your wallet without giving access to your Ethereum private key
4. Allow your utility provider to charge you up to $100/at the end of the month, based on your usage.

6.2 Pre-authorised Payments

Recurring payments are the first use case of pre-authorised payments. A more significant and larger use case is the ability to provide explicit permission to allow other parties to spend money on behalf of a user. A good example of this is a business which is authorised to spend up to $30/month from a user's wallet, but is given the freedom and flexibility to do so. This system mimics the existing financial infrastructure of banks and credit card processors where a user's trust is proxied to the business. However, with blockchain technology the user is in full control of the mechanism of this process and can revoke permissions at their will.

8x Service Nodes will process an 'up to $x' transaction differently to how a regular subscription payment would be handled. Rather than the smart contract taking funds from the user's wallet and paying it out to the business directly, the business will send a request to the service node directly. This messaged will be signed with a special 8x cryptographic key (not an Ethereum private key), sent to the service node and an Ethereum transaction will be sent on their behalf.

These cryptographic keys can be powerful as they allow merchants to make cryptocurrency payments on behalf of the user, without holding Ethereum to pay for gas or the security risk of cryptocurrency key management. Any compromises would be limited up to the pre-authorized amount they have specified and payments would still only be rerouted back to the original business hence reducing the incentive for such an attack vector. 3rd party key custodial solutions can ensure that the actual cryptocurrency payment wallet is not compromised.

7 Future Improvements

7.1 Support for other Stable Coins

The existing solution utilises MakerDao to hedge against cryptocurrency volatility. However, the 8x protocol may further support alternate stable coins (in compliance to ERC20 standard) to reduce risk as MakerDao is only collateralised by a single asset class. Having the option to use other stable coins with different collaterilisation allocations and other international pegs will provide additional stability and versatility to the protocol. For example, international businesses have to rely on the USD to their currency's exchange rate. If a stable coin is created that allows them to eliminate this additional uncertainty it can provide significant advantages to the existing fiat system. Support for additional currencies can easily be implemented through specifying the token type in the subscription contract.

7.2 Cross-chain support

Through solutions such as Polkadot and Cosmos, 8x will allow users to pay with a wide range of cryptocurrencies. This will work by users depositing their coins to a specified address, then an intermediary chain with custody of that address will ensure those coins are converted to Ether. From that point, the protocol is able to facilitate recurring cryptocurrency payments as it usually would.

7.3 User Interface

The front end application layer of the 8x Protocol is constantly evolving. It intends to solve the challenges listed in section 2.2 of this paper. 8x Protocol plans to provide a simplified user interface to improve the challenges of the processes associated with the Subscription Economy. The protocol intends to focus on the primary payments mechanism initially. Further details regarding its future state user interface will be released at a later date.

7.4 Integration with Enterprise Resource Planning Apps

8x protocol intends to be the leading global recurring cryptocurrency payments provider for all value providers. This includes enterprises and as such 8x Protocol will be considering enterprise partnerships and integration opportunities with leading enterprise software providers. The intent is to enable enterprise value providers to reduce effort on back-end payments facilitation and more towards value deliver to customers. Details of this will be expanded as the 8x Protocol matures.

7.5 Regulation

8x Protocol acknowledges the establishment of existing governing financial bodies of countries around the world. 8x intends to comply with global financial governing law. 8x protocol intends to provide awareness and facilitate positive working relationships between itself and regulatory bodies to ensure the proper policies are followed when establishing the ecosystem for the Subscription Economy.

8 Summary

Why: To be the leading enablers of the Subscription Economy in the cryptocurrency space.

What Enabling recurring payments in cryptocurrency.

How:

1. Use of stable coins such as MakerDao eliminates risk of cryptocurrency volatility when purchasing goods and services.
2. Letting users stay in control of their funds eliminates the risk of high risk attack vectors.
3. Allowing a network of competitive 8x Service Nodes to execute payments and earn a percentage of the fee ensures transactions are always executed.
4. Creation of SDKs can allow dApps and regular web apps to accept recurring cryptocurrency payments.
5. Single interface for users to manage all their recurring subscriptions.
6. Loosely coupled smart contract architecture allows easy protocol improvement.

9 Glossary

9.1 MakerDao

Unlike centralised stable coins such as Tether, MakerDao is fully collateralised (by Ether) and maintains a 1:1 ratio to USD. MakerDao achieves this through collateralised debt positions (CDPs) backed by their Ether. As the price of Ether goes up, CDP holders can borrow more Dai. Should the value of Ether go below a 100% collateralisation ratio to Dai, CDPs are liquidated and Ether is returned back to the owners of the CDP. In the case of a black swan event (flash crash of Ether's price), MakerDao's second token, Maker/MKR, is liquidated on the open market to raise additional capital to maintain the collateral. While Dai has temporarily lost it's peg to USD in the past, the target rate set by MKR holders ensures that the peg is quickly restored.

9.2 State Channels

The concept behind state channels is to open a "bar tab" like account on-chain and let both parties transact until they want to close the engagement and settle their account. In the case of any fraudulent transactions, a user can submit cryptographic proofs that the other party cheated or attempted to cheat and get their money back.

10 Acknowledgements

This solution is built upon the many foundation steps taken by projects such as MakerDao, Ethereum Alarm Clock Service, 0x, Kyber Network and more. Innovation is a continuous process and I hope that this work can be further used to help the world realise a decentralized future where all subscription services can be paid on the blockchain.

A special thank you to Rob Morris for providing the initial idea for making fees dynamic rather than a fixed percentage.

11 References

[1] Bloomberg. *The Ether Thief.* URL: https://www.bloomberg.com/features/2017-the-ether-thief/.

[2] Brandon Chez. *Coinmarketcap.* URL: https://coinmarketcap.com/.

[3] Chronaeon. *A rewrite of the Yellowpaper in non-Yellowpaper syntax.* URL: https://github.com/chronaeon/beigepaper.

[4] Fred Ehrsam. *How to Raise Money on a Blockchain with a Token.* URL: https://blog.gdax.com/how-to-raise-money-on-a-blockchain-with-a-token-510562c9cdfa.

[5] *Eth Gas Station.* URL: https://ethgasstation.info/.

[6] Eric Hughes. *A Cypherpunk's Manifesto.* URL: https://www.activism.net/cypherpunk/manifesto.html.

[7] Yaron Velner Loi Luu. *KyberNetwork - A trustless decentralized exchange and payment service.* URL: https://home.kyber.network/assets/KyberNetworkWhitepaper.pdf.

[8] Piper Merriam. *Ethereum Alarm Clock.* URL: https://github.com/ethereum-alarm-clock/ethereum-alarm-clock.

[9] Joel Monegro. *Fat Protocols.* URL: http://www.usv.com/blog/fat-protocols.

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

[11] Owocki. *Recurring Subscription Models are a Good Thing and should be viable on Ethereum (Merit + Architecture ERC).* URL: https://github.com/ethereum/EIPs/issues/948.

[12] Ptrwtts. *Pooled Payments (scaling solution for one-to-many transactions).* URL: https://ethresear.ch/t/pooled-payments-scaling-solution-for-one-to-many-transactions/590.

[13] Molly Richardson. *Challenges for Cryptocurrency Subscription Billing.* URL: https://www.rebilly.com/challenges-for-cryptocurrency-subscription-billing/.

[14] Bancard Sales. *How Credit Card Processing Works - Transaction Cycle 2 Pricing Models.* URL: https://www.youtube.com/watch?v=avRkRuQsZ6M.

[15] Kyle Samani. *New Models For Utility Tokens.* URL: https://multicoin.capital/2018/02/13/new-models-utility-tokens/.

[16] Jayne Scuncio. *6 subscription economy solutions for global challenges in 2018.* URL: https://www.zuora.com/2018/02/02/6-subscription-economy-solutions-for-global-challenges-in-2018/.

[17] Myles Snider. *An Overview of Stablecoins.* URL: https://multicoin.capital/2018/01/17/an-overview-of-stablecoins/.

[18] John Stark. *Making Sense of Ethereum's Layer 2 Scaling Solutions: State Channels, Plasma, and Truebit.* URL: https://medium.com/l4-media/making-sense-of-ethereums-layer-2-scaling-solutions-state-channels-plasma-and-truebit-22cb40dcc2f4.

[19] Jack Tanner. *Summary of Ethereum Upgradeable Smart Contract RD.* URL: https://blog.indorse.io/ethereum-upgradeable-smart-contract-strategies-456350d0557c.

[20] Maker Team. *The Dai Stablcoin System.* URL: https://makerdao.com/whitepaper/DaiDec17WP.pdf.

[21] Tien Tzuo. *The Subscription Economy.* URL: https://www.zuora.com/vision/subscription-economy/.

[22] Will Warren. *The difference between App Coins and Protocol Tokens.* URL: https://blog.0xproject.com/the-difference-between-app-coins-and-protocol-tokens-7281a428348c.

[23] Amir Bandeali Will Warren. *0x: An open protocol for decentralized exchange on the Ethereum blockchain.* URL: https://github.com/0xProject/whitepaper.

[24] Dr. Gavin Wood. *Ethereum: A Secure Decentralised Generalised Transaction Ledger.* URL: https://github.com/ethereum/yellowpaper.

[25] Dr. Gavin Wood. *Poladot: Vision For A Heterogeneous Multi-Chain Framework.* URL: https://github.com/polkadot-io/polkadot-white-paper.