8x Protocol

Decentralised recurring payments on the Ethereum blockchain

Kerman Kohli
kermankohli@gmail.com

May 5, 2018

Abstract

The following paper outlines how the 8x protocol facilitates recurring cryptocurrency payments for software-as-a-service business businesses. Currently, there are only solutions to accept one-time cryptocurrency payments for businesses. 8x is designed to tackle this problem through Ethereum's decentralised blockchain ledger, the ERC20 token standard, use of stable coins (such as MakerDao) and a network of distributed "processors". Subscription plans are registered in a smart contract and customers can subscribe to them directly. To execute the transaction, payments are claimable to a network of "processors" who in turn receive a percentage fee of the original subscription payment made between the consumer and vendor. In order for processors to make payment claims, the 8x native token must be staked.

Contents

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 anyone in the world through a trust-less network of miners. To make a payment, the sender signs the transaction with their private key and broadcast 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" from their bank account directly.

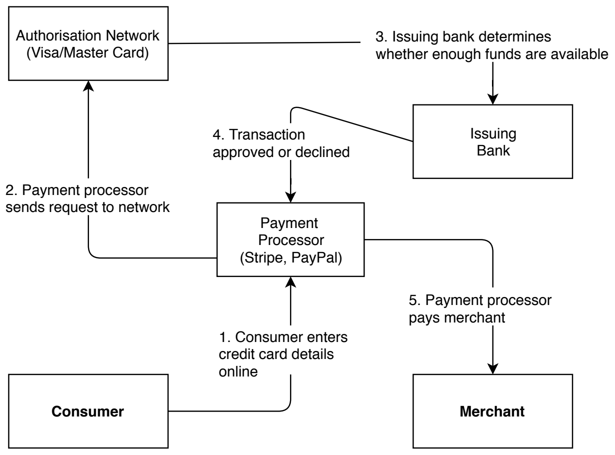
Authorisation Network (Visa/Master Card)
3. Issuing bank determines whether enough funds are available
Issuing Bank
4. Transaction approved or declined
2. Payment processor sends request to network
Payment Processor (Stripe, PayPal)
5. Payment processor pays merchant
1. Consumer enters credit card details online
Consumer
Merchant

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

While it isn't hard to realise the benefit of eliminating all the 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 trigger the transaction in the first place.
- Another major problem of using cryptocurrencies 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 not a good medium of transfer due to the volatile nature of the market. Potential solutions to this problem include using 3rd party oracles to fetch the latest exchange rate, although this puts an extremely high level of trust in external parties. It also introduces a potential threat to the correctness 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 Existing Work

In terms of a complete recurring payments solution, 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 transaction fees compared to currencies such as Ether.

An integral part of 8x's protocol is the use of stable coins such as 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.

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. 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. Although this may sound suitable for recurring payments, it doesn't take into account that monthly subscriptions are a way to reduce the burden of one-time payments for subscribers. Creating state channels that require the total subscription up-front may also not be economically viable for individuals. It would also eventually require a top-up after the up-front payment is made.

To enable "pull" based payments, a potential method is creating a pre-paid subscription escrow smart contract where users deposit their money into at the start of each month. Having a single smart contract hold all user funds can lead to the contract being subjected to sophisticated and targeted attack vectors. An example of this is the $30 million DAO hack that happened in 2016, despite having the smart contracts audited. This can primarily attributed to the fact that formal verification methods aren't mature and can result in unexepected and efficient exploits by black-hat hackers. 0x protocol (decentralised exchange) has a unique solution to this problem by taking advantage of the ERC20 standard's approve functionality. A transfer proxy contract 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.

Scheduling tasks on the blockchain is a problem that is still being worked on. However, the closest solution till date is 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 creates a game where executors are competing against each other - decreasing the likelihood of claim conflicts. Also, to claim a payment, executors need to stake 8x tokens.

3 System Overview

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

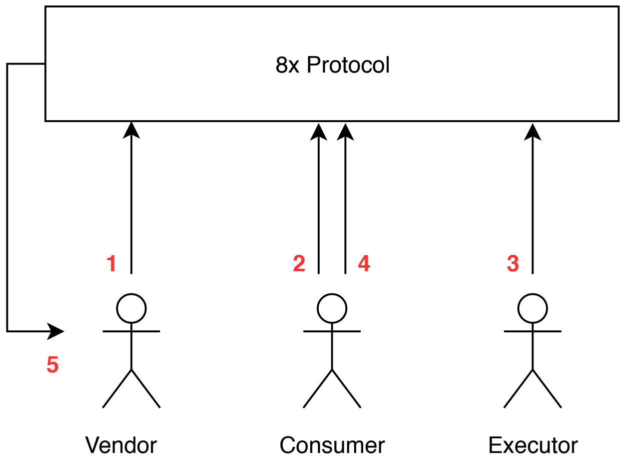

Figure 2: a conceptional diagram of how each party will interact with the 8x protocol

1. Vendor adds subscription plan specifying how much is to be Dai is to be charged and at what interval (in days). Gas is required to execute this transaction, although it only needs to be done once and is marginal to execute. An additional gas fee is also charged in the case that the user's wallet does not have enough Dai and their subscription needs to be cancelled.

2. Consumers subscribes to plan created by the vendor. This is assuming they already have DAI on hand. If they don't, Kyber Network can be used to facilitate the immediate exchange of ETH to Dai. This eliminates both parties having to deal with volatility.

3. Executors on the network compete to claim the right to execute the payment on the network. To claim a transaction, 8x tokens need to be staked. We can almost always guarantee the execution of payments due to the economic incentives that make it more profitable to execute a transaction later on (discussed further in Token Use Case section). Failing to execute on a claimed transaction results in a processor's stake to be slashed.

4. At the time of execution, funds are directly transferred from the consumer to the vendor. Since user funds aren't stored inside the smart contract itself, the attack vector for the smart contract is significantly reduced. If the proxy contract making payments is ever compromised, it can be killed and access to user funds is prevented. This is also advantageous for businesses as they don't have to wait for rolling payments which can improve cash flows.

3.1 Token Use Case

8x's token will primarily serve as a utility in order for processors to claim the right to execute a transaction. For example, when Bob subscribes to Netflix's \$10/month subscription, processors will be notified of this, select when they would like to claim the payment and stake a cetain amount of 8x tokens.

A problem with this from game theory perspective is that as the number of processors increase, the probability of earning from a transaction decreases linearly. To make matters worse, any earnings will eventually be used to pay failed claim gas costs. This causes convergence to a zero-sum game. To counteract this force, a claim window is introduced which gives processors the option to balance risk with reward.

Suppose the claim period is open for the first 10 days. Executors will then decide when they want to execute the payment over the next 10 days. A payment claimed on day 1 will receive 10% of the 1% payment fee in Dai. Similarly a payment claimed on the day 10 (last day of claim period) will receive 100% of the 1% payment fee. The remainder of the percentage fee (carrying on if a payment is claimed on the first day), is used to purchase the 8x token on the market and then burned. If a processor fails to execute a transaction their stake of coins will be burned.

To balance an decreasing supply and avoid excessive speculation, tokens will also be minted at a consistent rate per month. An exact number for this mint rate and reward mechanism is to still be determined. Businesses and users using the 8x protocol will not have any interaction with the 8x token, unless they're also processors.

The 8x protocol will also give token holders to vote on proposals made to the network as decisions impact their earnings made from the protocol.

3.2 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.
3. Processors are rewarded in the form of the currency the subscription uses, not 8x. The native 8x token is only used for staking claims and governance rights.
4. The gas costs processors incur are taken from the total of the subscription payment (essentially businesses absorb this cost).

4 Smart Contract

4.1 Architecture

The entire protocol is run on the Ethereum blockchain through smart contracts written in Solidity. Standard gas fees applies to interact with the smart contract for businesses, consumers and processors. 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.

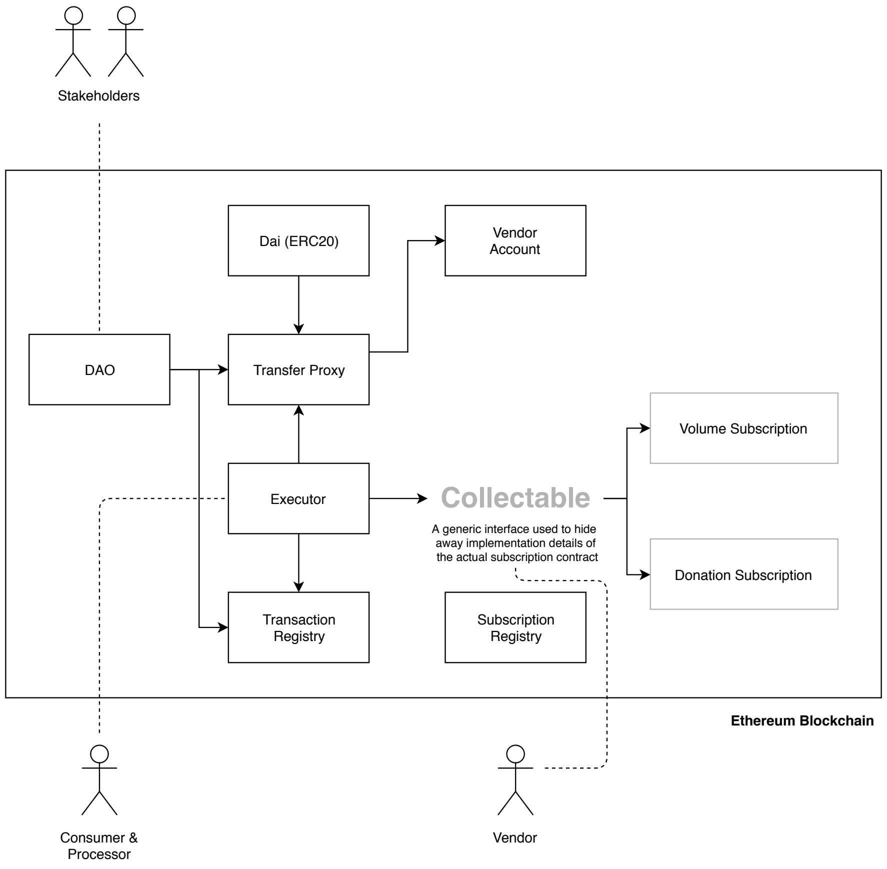

Figure 3: a high level view of how the different smart contracts will interact with each other to create a decentralised payments protocol

4.2 Transfer Proxy

As discussed in the "previous work" section, 0x's model of keeping a transfer proxy is one that has been implemented into this architecture. By having a single component 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 collectable 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 Decentralised Autonomous Organisation (DAO) or 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 mutliple 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.

4.3 Executor

The executor component is where the core logic and functionality of the smart contract lies. Consumers and processors interact with it directly in order to claim and make payments. Since all subscription contracts adhere to the collectable 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.

4.4 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. This newly created object is claimable to any processors on the network who in turn will earn a fee. The game theory and mechanics of this are still under work however, the earlier a processor claims a payment the less total % he gets of the total fee of the proposed 1% fee they can claim. The remainder of the fee is used to purchase 8x tokens using Kyber Network and then burned.

4.5 Collectable

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 will most likely be hosted on a centralised server due to the unwanted nature of publicly exposing an email address to public key on the blockchain.

4.6 Subscription Registry

Part of the Cyberphunks movement is to provide power back into the hands of the people instead of large corporations. The subscription registry allows users to view all the services they're currently subscribed to through mappings stored in the contract. This therefore allows them to view currently subscribed services, when the next payment is due and cancel any unwanted subscriptions. Credit card payment subscriptions lead to users being notified of what they're subscribed to once they see payments in their bank statements or are required to find the "Cancel Subscription" button on a vendor's website.

5 Future Improvements

5.1 Support for Other Stable Coins

The existing solution utilises MakerDao to hedge against cryptocurrency volatility. However, the 8x protocol should also be able to support other stable coins (assuming they comply to the 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 uncertainity 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.

5.2 Global Fiat Settling Layer

Stable coins such as Dai are viewed as a representation of value in comparison to the US dollar. While this may be acceptable for crypto products, fiat-based businesses would want to receive fiat payments directly in their bank accounts. This is due to the complexities of understanding cryptocurrencies, having to deal with private keys and being KYC/AML compliant. An application which utilises the 8x protocol to provide a streamlined business experience for payments will have a competitive edge over Stripe and Paypal as it costs much less. It will also help improve the utilities of cryptocurrencies as a whole.

6 Summary

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 processors 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 crypto payments.
5. Single interface for users to manage all their recurring subscriptions.
6. Loosely coupled smart contract architecture allows easy protocol improvement.

7 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 decentralised future where all subscription services (including rent) can be paid on the blockchain.

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

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

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

[17] 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.

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

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

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

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

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

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