A Risk Redistribution Standard for Practical Cryptocurrency Payment

1st Yao-Chieh Hu
dept. Computer Science and Engineering
HKUST
Kowloon, Hong Kong
yhuag@connect.ust.hk

2nd Ting-Ting Lee
dept. Computer Science and Engineering
HKUST
Kowloon, Hong Kong
tleeae@connect.ust.hk

3rd Chungsang Tom Lam
John E. Walker dept. of Economics
Clemson University
Clemson, SC, USA
tomlam@uchicago.edu

***Abstract*—Cryptocurrencies are developed as a decentralized and trustless payment system, in which participants should be able to conduct payments across borders with acceptable latency. However, the fluctuation of the exchange rate between crypto and fiat currencies has raised significant concerns and thwarted the prevalence of cryptocurrency payment adoption. Existing solutions require merchants to liquidate the received cryptocurrency on an exchange platform. To compensate for the exchange rate risk, merchants tend to charge a higher price in cryptocurrencies compare to in fiat currency, which dampens the incentive of customers to choose cryptocurrencies as the means of payment.**

This paper proposes an architecture bolstered by smart contracts to transfer the risk from the merchants to the cryptocurrency issuer. This narrows the gap between prices denominated in cryptocurrencies and fiat currencies, and thus increases the adoption of cryptocurrencies as a payment method. The Ethereum blockchain is chosen as the experimental environment in this work, yet the architecture can be migrated to other decentralized systems without additional efforts. This work devises a novel ERC [1] standard to resolve the payment at a predetermined exchange rate that can be employed by any existing cryptocurrency. Immutable events on the blockchain will be generated upon the issuance and settlement of a payment, which are considered as the receipts for granting rights to the merchants to settle the payment at a regular basis. The architecture demonstrates a notable reduction on the risk of exchange rate for the merchants, solving the primary problem of cryptocurrency payment adoptions nowadays.

***Index Terms*—cryptocurrency, token economy, blockchain, decentralized application, smart contract, payment system**

I. Introduction

The peer-to-peer electronic currency, Bitcoin, has disrupted the digital payment market fiercely since its debut in 2009. Its decentralized, anonymous, and cryptography-backed characteristics have empowered it to become the major currency being transacted on the dark web, in exchange for goods and services, guaranteeing that the true identities of involved parties are untraceable.

Bitcoin, although being sufficient in functionalities as a digital currency, has soon been identified as ineligible to be prevalently adopted. The disqualification can be originated to its deficiency of a Turing-complete scripting language, which can expand transactions and payments to a more sophisticated logic with regards to full automation.

Therefore, six years later, in 2015, Ethereum has come to the stage with its in-built Turing-complete programming language and a virtual machine for executing the programmed scripts natively within its network. These scripts, which has name *Smart Contract* derived from its utility for digitally expediting and enforcing the traditional contracts. Smart Contract enabled a wide variety of application logic to be implemented and be stored publicly and permanently on a decentralized ledger, conventionally referring to as the Blockchain.

Ethereum, beyond being a programmable blockchain that can transfer its innate digital currency, Ether, based on the smart contract-defined logic, it also allows developers to devise their own customized currencies, generally being named *Tokens*, with their functionalities regarding transferal prescribed in a smart contract.

Nonetheless, although Ethereum introduces the possibility for diverse types of digital currencies to be transacted in full automation empowered through smart contracts, it still fails to become the prevailing payment method in today's world. The reason is as simple as its inadequacy of the consideration of the latent risks involved for its adopters, such as merchants and vendors, which is primarily due to the severe fluctuation of the market price corresponding to fiat currencies.

Conventionally, payment systems leverage fiat cryptocurrencies to complete a payment, where the customer transfers an explicit amount of cryptocurrencies to the merchant in exchange for goods and services. This mechanism works without hindrance since the fiat currencies are generally more stable, in regard to its value, under governmental supervision. However, in the case of cryptocurrencies, the merchant who accumulates the cryptocurrencies has the advantage to liquidate the cryptocurrencies on the market when the price is favorable, vitiating the benefits of the cryptocurrency issuer or the platform itself. Alternatively, the merchant might choose to increase the price of the goods and services to offset the potential risk of an unfavorable price. As a consequence, the soaring costs of goods and services, along with the fluctuating rate of cryptocurrency, critically hamper the incentive for general customers to embrace cryptocurrencies for actual payment.

[1]ERC stands for Ethereum Request for Comments, defining the standards for the Ethereum platform.

This paper proposes a technical standard that shifts the cryptocurrency exchange rate risks from the merchants to the cryptocurrency issuer. A stream of time-stamped payment receipts is generated along with the continuous payments took place between the customers and the merchants. A predetermined exchange rate is established between a pair of a merchant and a cryptocurrency issuer, which is usually semi-stable but allows flexibility in subtle modulations at a regular basis, to accommodate the actual exchange rate of the market. Periodically, the merchants approach the cryptocurrency issuer to settle a stack of payments that have been created between themselves and the customers, expecting fiat currencies in return.

The work will demonstrate that no risk would be bored by the merchants since they have no cryptocurrency in hand throughout the payment process; and the overall risk transferred to the cryptocurrency issuer is significantly diminished due to that the base size of the payment transactions is large enough when the cryptocurrency issuer combines all of its transactions, regardless of which merchant initiates. More importantly, the customers no longer have to waver between the choices of the purchasing time of the products, since that risk of rate fluctuation is canceled out, and therefore the merchants have no reason to raise the price for cryptocurrency payments accordingly overtime.

The economic architecture closely resembles the oil price, which only mutates at an expected level and time span. The pricing model of local oil price does not vary in a frequent manner with an obvious aim to extend the adoption and lower the risk for a customer to purchase the services. The incentive for a customer to purchase a price-varying product can be very low. If the price is soaring, the customers would not favor the expensive price; even when the price is slumping significantly, the customers would expect the price to dive lower and prefer holding their funds rather than spending. This paper mitigates the risk of price fluctuation by transferring all the risks involved to the cryptocurrency issuer.

Even though the total risks of the economic model remain the same, leveraging the fact that different roles have various level of risk aversion attitudes, this paper migrates the risk from the merchants and the customers to the cryptocurrency issuer. Intuitively assuming that the price is oscillating in a random walk model, since that the issuer has a large base of trials, compared with a merchant or a customer, the resulted risk is diluted.

Under this configuration, because the cryptocurrency is transferred to the issuer for every payment between the merchant and the customer, the merchant bears no risk and can regard the cryptocurrency as a semi-stable cryptocurrency. The semi-stable cryptocurrency influences the pricing decision of the merchant and converts the price of goods and services into a stabilized state. Therefore, it effectively becomes convenient for customers to accept the stabilized pricing. Eventually, as the quantity of the cryptocurrencies for payment escalates, this predominant adoption will further extend the denominator of the risk model for the issuer, and, in a virtuous circle, the cryptocurrency approximates a stable state in the long term.

Contributions

1) This paper presents an experimental analysis of a semi-stable cryptocurrency economy model.
2) This paper is the first to address a risk redistribution approach that shifts the exchange rate risk from merchants to cryptocurrency issuers, to the best of our knowledge.
3) This paper presents a receipt-as-a-payment structure to split the on-chain cryptocurrency payment and the off-chain fiat currency settlement.
4) This paper develops a novel ERC standard delineating a smart contract interface that is compatible with the ERC20 token standard.

The rest of the paper is organized as follows: In Section II we discuss the related work of existing cryptocurrency payment solutions; In Section III we present the proposed architecture and the ERC standard with the associated smart contract interface; in Section IV we explain the findings of our implementation measurements; in Section VI we discuss our design extensions; and in Section VII we conclude the paper.

II. Related Works

Significant efforts have been devoted to bolstering the pervasiveness of cryptocurrency payment adoption, especially in the field of the distributed ledger throughput, usually measured by transactions confirmation latency. Most of the blockchains, including Bitcoin [1] and Ethereum [2], are suffering severely from the TPS, transaction-per-second, problem, which curbs them from getting more acceptance by the general public with the demand of instant payments. A blockchain based on PoW consensus[2] requires every node in the network to contrive to reach an agreement on all the transactions, which sacrifices the efficiency for achieving thorough decentralization. Some other blockchain employing consensuses that are more centralized, such as PoA[3] and DPoS[4], can have a recognizable advantage in terms of TPS, but at the expense of the decentralization. A blockchain-based e-voting system [3] has chosen PoA over other consensuses in order to achieve comparatively fast transaction confirmation. Another blockchain ecosystem, EOS.IO [4], has been leveraging the DPoS consensus, with 21 super node chosen globally, to reach efficient consensus across a rather smaller group of participants.

With a view of surrendering no decentralization, several approaches aim at solving the TPS problem have evolved. Several methods including sidechain [5], verifying local transactions on a separated chain and submit to the main chain with less frequency; sharding [6], segregating the network into multiple shards and conduct the transaction verification in each shard to allow concurrency. A huge amount of efforts have

[2]Proof of Work, describe the situation that miners need to solve mathematical puzzles in order to validate blocks on the network and be rewarded.

[3]Proof of Authority, rather than opening the possibility that everyone can be a miner, only approved accounts, called the validators, can verify blocks on the network.

[4]Delegate Proof of Stake, uses real-time voting mechanism and reputation system for reaching consensus.

been devoted to the TPS enhancement approaches, but few have addressed it successfully.

Starting from 2015, micropayment channels for payments to be confirmed without confirmation delay have been introduced by The Duplex Micropayment Channels [7]. The Bitcoin Lightning Network [8] also increases the transaction throughput using a different approach that promotes truthful behaviors through punishments. The network participants within the Bitcoin blockchain are allowed to perform off-chain payments following this design. In Ethereum, The Raiden Network [9] simulates the techniques in The Bitcoin Lightning Network while implementing in its Turing Complete programming language, Solidity, in the form of smart contracts. Plasma [10] joins forces of a series of Ethereum smart contracts, forming a hierarchical tree-like structure of sidechains, to only leverage the root chain as the final commitments, reducing the global verification to a minor demand. Although these works have provided effective means to shorten the transaction confirmation time, their primary focus is more on the theoretical rather than the practical side. Few have considered the barrier to overcome when it comes to market adoption for people unfamiliar with, and thus tend to be doubtfully cautious when using cryptocurrencies as a mean of payment.

Starting since 2011, market players have been devoted into fostering the payment convenience by developing services as payment gateways. For Bitcoin, BitPay [11] was one of the first of such companies that popularize e-commerce platforms using Bitcoin, allows Bitcoin to be converted into 9 kinds of fiat currencies available in 240 different countries, and compatible with about 40 types of existing software applications. However, its abundant services are only for Bitcoin and Bitcoin Cash.

Another player, CoinBase [12], is one of the biggest Bitcoin exchanges that also provides business solutions enabling enterprise level Bitcoin payments. Similar to BitPay, it has various developer friendly plugins and APIs for easy integration with existing commercial platforms. Nonetheless, the cryptocurrencies that are available to exchange on its platform are limited only to Bitcoin, Bitcoin Cash, Ethereum, Litecoin, and Ethereum Classic.

Aside from the heavy need in Bitcoin payment solutions, the market size for Ethereum is increasing as more and more merchants such as online retailers and e-commerce platforms are starting to accept cryptocurrencies other than Bitcoin for payment. According to the CoinPayment [13] website which listed out a store directory for a wide range of different cryptocurrencies available for payment, there are more than 500 online stores accepts Ether, and more than 1000 *altcoins*, generally referring to as the currencies developed in Ethereum using smart contracts, that are registered and available on this payment gateway.

In recent years, companies focusing on further prevail the adoption of cryptocurrency payment by providing an end to end service using mobile applications, such as TenX [14], Crypto.com [15], PundiX [16], and TokenCard [17], have shown considerable interests in this fast-expanding market. However, in order to achieve convenience for the majority of customers and reduce the additional effort required due to the high learning curve of such disruptive innovation, they compromised decentralization for speed and convenience. They make the concession to follow traditional practices that simulate VISA and MasterCard, which is to use centralized servers to collect cryptocurrencies from the customers to the platform and let the platform smooth the payment process without the need for blockchain throughout the procedure. This approach brings the advantage of fluent user experience and instant transaction confirmation. Still, it fails to leverage the decentralize characteristic of the blockchain. The only difference compared with traditional payment solutions is thus merely on the type of currency transferred. This paper differs from these works by aiming at devising a standardized and decentralized cryptocurrency payment solution by first addressing the risk considerations among relevant merchants, customers, as well as the cryptocurrency issuer.

The risk of holding a large amount of cryptocurrencies when accepting it as a payment method is a critical aspect of cryptocurrency payment prevalence barrier that is generally ignored. As analyzed in Bitcoin price volatility reports [18]–[20], the exchange rate volatility between cryptocurrency and fiat currency is higher due to faster trading speed and prompt reactions on market sentiment when considering the cryptocurrency economy landscape. The risks involved is a significant concern for adopters and remained as a problem to solve. Some academic efforts [21]–[24] are put in to analyze the risk of cryptocurrency adoption, considering the uncertainty.

In summary, the need for a practical cryptocurrency payment solution has been identified heavily throughout the development of the blockchain technology, but no truly decentralized and risk reduction solutions have been presented until this work.

III. System Architecture

This section illustrates the interactive flows between the three roles of participants and the corresponding technical implementations of the paper, in the format of code interface and pseudo-codes.

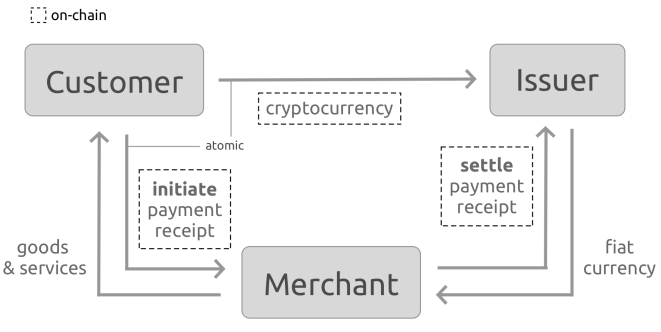

Fig. 1. Relationships of Merchant, Customer, and Issuer

Figure 1 depicts the interactions between the three major roles: the Customer, the Merchant, and the Issuer. Particularly,

there are two categories of actions in figure 1. When a payment action takes place, the Customer initiates a payment to the Merchant, which simultaneously completes a cryptocurrency transfer to the Issuer, and generate an event on the blockchain recording the payment details like a receipt with a unique hash identifier. This entire action is atomic and only one transaction is needed, which inherently brings the benefit of gas saving and race condition prevention. Once the transaction receipt is generated on the blockchain, the Merchant can release the goods and services to the Customer. This operation has a latency that depends on the choice of network and its current status. When the Merchant approaches the Issuer to conduct a settlement of the payments, the Merchant provides a pile of historical payment receipts associated with herself. The Issuer pays back the Merchant in fiat currency with the value equivalent to the sum calculated from all the payment receipts and requests the Merchant to settle the payments by their receipts, converting the receipts to be invalid for double-redemption. The Merchant and the Issuer need to reach a consensus upon the exchange rate for the next receipt collection period, which will then be used for calculating the corresponding amount of fiat currency for paying back the Merchant as the next period ends.

A. ERC Standard Proposal

In this work, ERC1586 [25] is proposed to address a practical and standardized smart contract interface that is fully compatible with ERC20 [26]. Three kinds of roles are defined and explained as follows:

1) Roles:

- **Issuer:** The Issuer is the creator and owner of the cryptocurrency. The Issuer decides a predetermined exchange rate with the Merchant, which will be adopted when the Merchant wants to execute a settlement of payment receipts.
- **Merchant:** The Merchant provides goods and services to Customer and requests cryptocurrency payment in return. The Merchant receives no cryptocurrency on the spot but the receipt recorded on the blockchain instead. At a regular basis, the Merchant can settle the collection of receipts at a predefined exchange rate with the Issuer.
- **Customer:** The Customer possesses cryptocurrencies and initiates payments to a Merchant in exchange of goods and services. The actual cryptocurrency flows to the cryptocurrency Issuer, and a payment receipt is generated as a record of the payment. It is kept by the Merchant for the purpose of settling the payments periodically.

2) Smart Contract Interface: A smart contract interface, named $RiskRedistributable$, is defined in the following section. Participants are required to register their roles as either $Customer$ or $Merchant$ before the commencement of the payment operations. The Role $Issuer$ stands for the cryptocurrency issuer.

In the smart contract interface, there are four different key-value $mappings$ declared to record information about the role of participants, the payment status, and the corresponding Merchant of a payment receipt, respectively. A nonce is included in the smart contract in order to preclude the possibility of replay attacks, initiating a payment that has already been initiated before. The nonce is incremental in accordance with the number of accumulated payments. The nonce is included in the $paymentHash$ of the payment receipt.

```
contract RiskRedistributable {
  enum Role {Customer, Merchant, Issuer}

  mapping (address => Role) public addressToRole;
  mapping (bytes32 => bool) public paymentSettled;
  mapping (bytes32 => bool) public paymentInitiated;
  mapping (bytes32 => address) public
      paymentHashToMerchant;

  uint256 nonce;

  event PaymentInitiated(
    address customer,
    address merchant,
    uint256 value,
    uint256 timeStamp,
    bytes32 paymentHash,
    uint256 nonce
  );

  event PaymentSettled(bytes32 paymentHash);

  function pay (address _merchant, uint256 _value)
      public onlyCustomer;

  function settle (bytes32 _paymentHash) public
      onlyMerchant paymentHasNotBeenSettled(
      _paymentHash) paymentHasBeenInitiated(
      _paymentHash);
}
```

Two functions are defined to enable the payment and settlement. A pay function is called by a Customer when paying to a Merchant, which is described in Algorithm 1. An Immutable event $PaymentInitiated$ that records every detail of the payment in the blockchain is emitted simultaneously. A unique $paymentHash$is generated in this payment receipt as the identifier of a payment. It is worth noting that the cryptocurrency is transferred directly to the Issuer instead of the Merchant since that to avoid exchange rate fluctuation risk, the Merchant should possess no cryptocurrency.

A $settle$ function is executed by a Merchant to settle a payment given a unique $paymentHash$, which is delineated as Algorithm 2. The $paymentHash$ will be invalid for re-settling afterward. By the settle action, the Merchant obtains an equivalent amount of fiat currency from the Issuer, with the exchange rate determined by each pair of Merchant and Issuer. The exchange rate is defaulted to be stable but can vary on a regular basis, as long as both the Merchant and the Issuer agree. The settlement is designed to take place less frequently. The frequency is also jointly decided by each pair of Merchant and Issuer. An example is to perform the settlement on a monthly basis.

Several safety requirements that have been addressed in the smart contract interface are presented as follows. In the

Algorithm 1 Initiate Payment

```
function PAY(merchant, val)
    if sender != CUSTOMER then return
    end if
    TRANSFER(Issuer, value)
    nonce ← nonce + 1
    hash ← HASH(sender, merchant, val, time, nonce)
    if PaymentAlreadyInitiated(hash) then return
    end if
    if PaymentAlreadySettled(hash) then return
    end if
    PaymentAlreadyInitiated(customer) ← true
    forMerchant(hash) ← merchant
    Generate PaymentInitiated Event
end function
```

experimental code base[5], it is verified that the accessibility of the pay function is restricted to the $Customer$, whereas the accessibility of $settle$ is limited to the Merchant who shall receive the payment as recorded on the receipt. Moreover, a verification is enforced on both pay and $settle$ function to ensure a unique payment record can be initiated only once, and so does the settlement. This verification guarantees that no payment action can be replayed or double spend to benefit the offenders by any mean.

Algorithm 2 Settle Payment

```
function SETTLE(hash)
    if sender != MERCHANT then return
    end if
    if PaymentAlreadySettled(hash) then return
    end if
    if PaymentNotInitiated(hash) then return
    end if
    if forMerchant(hash) != sender then return
    end if
    PaymentAlreadySettled(customer) ← true
    Generate PaymentSettled Event
end function
```

IV. Experiments

To testify the architecture and its resulted efficacy, five experiments including the confirmation latency and the gas cost have been conducted.

A. Environment Configuration

The experiments are conducted on Amazon Web Service EC2 instances to provide a replicable examination on our implementation. collection of t2.large instances with the bandwidth of up to 500 Mbps and two virtual CPU cores are adopted as the environmental settings. A Ganache [27] private blockchain [6] is used for all the experiments as a simulation to the Ethereum main network with higher throughput and TPS, Transaction Per Second, to expedite the experimental process.

In addition to system-wide settings, all the experiments are repeatedly tested for at least 10 trials to produce a more credible result. The average value and the standard deviation of all trails for each experiment are shown in the below figures. The thin vertical lines represent the span of standard deviation, and the solid black dots stand for the average value.

B. Limitations of the Testing Environment

Although experiments conducted on a private blockchain results in a non-replicable outcome on the main Ethereum network or other testing networks, it is sufficient for observing associations between influential factors on the smart contract setup rather than on the testing environment.

C. Measurements

1) Experiment 1: The Payment Latency Test on Various Numbers of Merchants: As declared in Section III, the payment function is triggered when a Customer initiates a payment to a Merchant. The aim of this experiment is to measure the dependency between the latency of payment function execution and the number of Merchants involved, considering the scenario with individual Customer. In Figure 2, the x-axis denotes the total numbers of Merchants participated, varying from 10 to 100, and the y-axis indicates the latency at the scale of a millisecond.

From Figure 2, it is conspicuous that the latency of the payment function is positively proportional to the total number of Merchants, displaying a linear relationship. The slope of the regression line of the average data points is around 36, implying that the expected growth rate in the latency delay per Merchant added is approximately 36 milliseconds. This indicates that up to 3.6 seconds are required when the number of Merchants scale to 100, which leads to a conclusion that there can be at least about 30 transactions being initiated and finalized on the blockchain per second. However, most of the established payment solution nowadays offered very high throughput for payment confirmation, for instance, Visa can process up to 24 thousand transactions per second. The throughput demonstrated in this experiment addresses an imminent demand for a solution to renovate the efficiency factor of the system, named Early Confirmation, which is outlined in Section VI-B.

2) Experiment 2: The Payment Latency Test on Various Numbers of Customers: Different from Experiment 1, this experiment focuses on measuring the effect of the number of Customers involved on the payment function latency. In Figure 3, the x-axis represents the total number of Customers participated from 10 to 100, and the y-axis denotes the latency.

It is an anticipated behavior that latency growth trend of the payment function is nearly identical for the case of increasing the number of merchants and Customers. The reason behind

[5]Experimental code base: https://github.com/BlockChain-UROP/risk-redistribution-payment-dappcon

[6]a tool from Truffle Suite to configure and initiate a local blockchain for Ethereum development

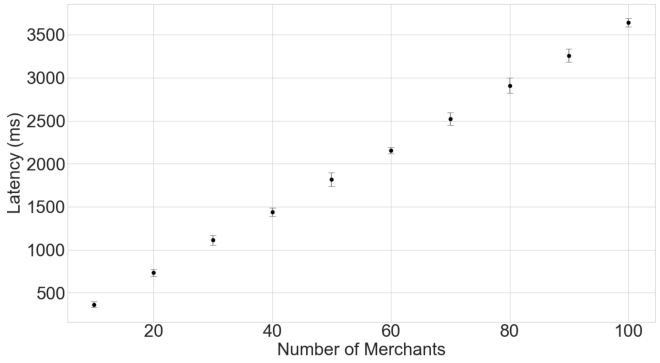

Fig. 2. Experiment 1: The Payment Latency Test on Various Numbers of Merchants

this outcome is as simple as the inherent independent characteristic of the payment function. Regardless of how many times the payment function has been initiated, who is the function caller, and to whom is the payment directed to, the procedure of the standard payment remains unchanged.

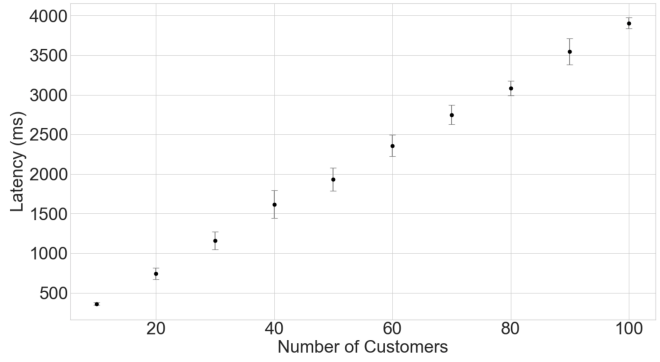

Fig. 3. Experiment 2: The Payment Latency Test on Various Numbers of Customers

3) Experiment 3: The Settle Latency Test on Various Numbers of Merchants: Section III defines the settle function to be enforced by the Merchants when they have a goal to redeem their payment in fiat currencies from the Issuer. This experiment has a target of examining the settle function execution latency, in accordance with the number of Customers in total. In Figure 4, the x-axis represents the total number of Merchants participated from 2 to 20, and the y-axis denotes the latency.

It is self-evident in this experiment that the standard deviation gradually surges as more and more Merchants trigger the settle function simultaneously. This is due to the fact that the settlement of the fund requires disparate Merchants to first retrieve a list of past generated payment event from their Customers. This procedure demands to query the blockchain's record, and in this experiment, the time for claiming the list of payment events is considered in the total latency measured. Higher volatility may be introduced as these events are fetched using the same machine at almost the same time for the convenience of conducting the experiments. Such unpredictable factor can be eliminated by regularly updating the local event record with the associated blockchain when all the Merchants are in segregated environments. In so doing, a stabilized standard deviation can be achieved.

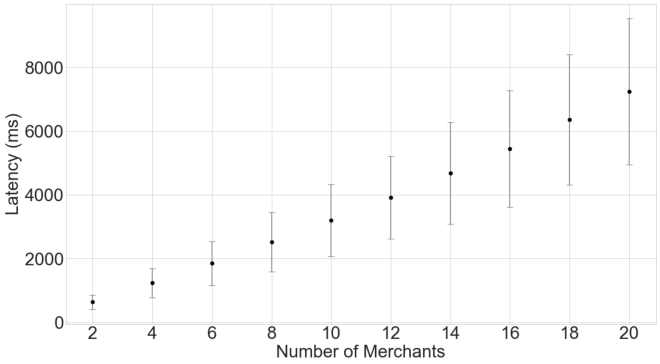

Fig. 4. Experiment 3: The Settle Latency Test on Various Numbers of Merchants

4) Experiment 4: The Settle Latency Test on Various Numbers of Customers: Similar to what Experiment 1 is to Experiment 2, this experiment inspects the level of incremental latency to the settle function as the number of Customers rises, given that only one merchant involved. In Figure 5, the x-axis represents the total number of Customers participated from 2 to 20, and the y-axis denotes the latency.

It is conspicuous that the latency sum of the entire settlement process, as more and more Customers engage in, leads to a considerable amount of expected waiting time. This explains the impossibility of the settlement to be in real time since that only the settle function can be only be executed twice per second. Consequently, this is the main reason behind this paper's mechanism to settle the payment only once per certain period that is comparably longer, at the discretion of the dual parties, to provide a more hazard-free user experience.

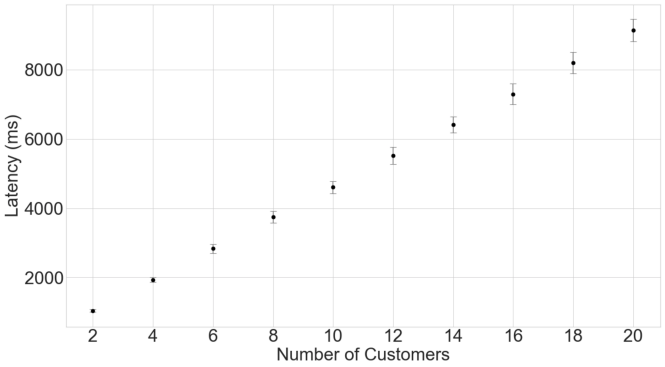

Fig. 5. Experiment 4: The Settle Latency Test on Various Numbers of Customers

5) Experiment 5: Gas Usage of Payment, Settle, and Transfer: Figure 6 further demonstrate how the cost of this design architecture looks like in the real world scenario, the gas usage of the payment and settle function are measured, with the

transfer function's gas cost presented as the control group. For simplicity, the amount of cryptocurrency transferred here is set to one unit.

From the figure, we can see that for payment, settle, and the classic ERC20 transfer function, it takes approximately 150000, 70000, and 5000 amount of gas respectively. By the time of writing, the gas price suggested from the ETH Gas Station [28] that yields a reasonable confirmation latency of 30 seconds on the main Ethereum network is 30 GWei [7]. Under this setting, table I demonstrates the transaction fee, inferred from the gas cost, in Ether as well as in fiat currencies that is needed when issuing the three functions.

TABLE I
ETH GAS STATION GAS COST AND TRANSACTION COST ESTIMATION

	payment	settle	transfer
Gas cost	151642	69496	50146
Transaction fee (ETH)	0.0045493	0.0020849	0.0015044
Transaction fee (USD)	$0.54137	$0.2481	$0.17902

It is inevitable to tell that problems may arise for the payment procedure due to its obligation of being fast enough when confirming payments, which leads to its high transaction cost as the required gas price scales up in exchange for a desirable transaction confirmation latency. Thus, section VI-A proposes a design extension of this work that allows Customers to delegate the payment function execution right to the Issuer. Therefore, the burden of the increased payment cost will be delivered to the Issuer, and lessen the costs for the general Customers.

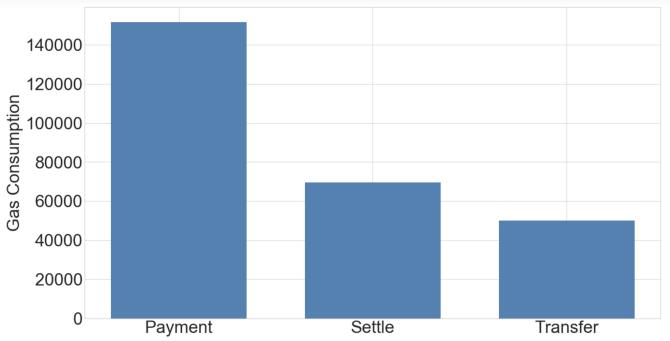

Fig. 6. Experiment 5: Gas Usage of Payment, Settle, and Transfer

V. RISK BEHAVIOR MODEL

This system enables the risk redistribution among Customers, Merchants, and the Issuer. To examine the economic correctness of the mechanism, consider a model with these three parties, who are all assumed to be risk-averse. Each of them has a utility function as Equation 1:

$$u(x) := 1 - e^{-\alpha x} \tag{1}$$

[7]Wei is the base unit of Ether in the Ethereum blockchain. While GWei stands for Gigawei, 1 Ether = 10^9 GWei.

which is known as the Constant Absolute Risk Aversion(CARA) utility function with the degree of risk aversion α, as described in Equation 2.

$$\text{degree of risk aversion} = \frac{u''(x)}{u'(x)} = -\frac{-\alpha^2 e^{-\alpha x}}{\alpha e^{-\alpha x}} = \alpha \tag{2}$$

Assume that the Merchants' activities are relatively small and therefore the token price is not affected by the Merchants' actions. Instead, the cryptocurrency price is affected by external environments and for simplicity, it can be assumed that the cryptocurrency price is a simple random variable with the same distribution over time. Each Customer bought the cryptocurrency and spend it on one of the Merchants later. Since everyone is spending their money at a different time, the payoffs are different depending on the current market price of the cryptocurrency, and whether the Merchants adjust the nominal price accordingly. The are three basic systems:

- Proposed System: The Issuer charges the Customer a fixed price denominated in cryptocurrency, and the Merchant get a fixed amount of fiat money from the Issuer. In this system, the Issuer bears all the risk.
- Alternative System 1: Merchants dynamically change the price of the product denominated in cryptocurrency according to the market price in the exchange. Then the Merchants sell the cryptocurrency to the market for fiat money. In this system, the Customers bear all the risk.
- Alternative System 2: Merchants charge a fixed price denominated in cryptocurrency and sell the cryptocurrency to the market for fiat money. In this system, the Merchants bear all the risk.

Since the cryptocurrency price is uncertain, the payoffs under different scenarios are random and not directly comparable. The model can convert the uncertain payoff x into a risk-adjusted fiat money value by calculating the certainty equivalent $CE(x)$, as shown in Equation 3:

$$CE(x) = u^{-1}(E[u_(x)]) = \frac{-log(E(e^{-\alpha x}))}{\alpha} \tag{3}$$

The payoffs are summarized in table II.

TABLE II
PAYOFFS UNDER DIFFERENT SYSTEMS

Proposed System	
Customers	$V - P + D_n$
Merchants	$(P - C) + D_m$
Issuer	$CE(B - D + \sum^{M} \sum^{N_m} (P + \delta_p))$
Alternative 1	
Customers	$CE(V - (P + \delta_p) + D_n)$
Merchants	$(P - C) + D_m$
Issuer	$B - D$
Alternative 2	
Customers	$V - P + D_n$
Merchants	$CE(\sum^{N_m} (P + \delta_p - C) + D_m)$
Issuer	$B - D$

V is the value of the product. C is the production cost of the product. B is the benefit the Issuer can obtain from

supporting the cryptocurrency payment system. The benefit can be promotion, seigniorage or any potential benefits. The Issuer can distribute the gain to Customers and Merchants via D_n and D_m respectively where $D = \sum^M D_m - \sum^N D_n$. D can be viewed as some off-chain benefits provided to Customers and Merchants. It is assumed that $V > C$ and B is sufficiently large so that the Merchant has the incentive to produce the product and the Issuer has the incentive to bolster the payment system. Since the model focuses on evaluating the risk redistribution, these assumptions ensure that the model will not run into degenerated cases where the product or the pay system itself is not worth running, therefore every parties' optimal choice is to produce nothing.

Since

$$\sum^{N_m} CE(\delta_p) < CE(\sum^{N_m} \delta_p) \tag{4}$$

and

$$\sum^{M} CE(\sum^{N_m} \delta_p) < CE(\sum^{M} \sum^{N_m} \delta_p) \tag{5}$$

, the total risk-adjusted payoffs is the highest in the proposed system, which is the scenario put forward in this paper. In other words, if the system is changed from alternate system one/two to the proposed system, the Issuer can reduce d_m and d_n to a certain extent that the Merchants and Customers are indifferent before and after the change, but the Issuer's payoff will still increase.

The risk reduction is driven by the fact that the whole platform has more transactions than any individual Merchant by construction. The random fluctuation of cryptocurrency prices has higher variance than the aggregate price deviations ($\sum^M \sum^{N_m} \delta_p$)). This is true even if the degree of risk aversion is the same for the Customers, the Merchants, and the Issuer. This risk reduction benefit is even higher if the Issuer is less risk averse than the Merchants and the Customers.

The analysis above assumes that there is no gas consumption. In reality, the Customer has to pay the gas via native cryptocurrency. This paper also proposes an extensive design which allows Customers to delegate the payment function execution right to the Issuer. Since the gas cost is affected by the price of the native cryptocurrency, there is an exchange rate risk associated with it as well. Allowing the Issuer to aggregate the gas cost and bear this risk can increase the payoff of the Issuer in the same manner as the aggregated exchange rate risk of the product price. This extensive design will be discussed in section VI-A.

The proposed system suggests that the cryptocurrency payment should head directly to the Issuer. One may suggest a similar system where the cryptocurrency payment is collected by the Merchant and the Issuer guarantee a fixed exchange rate when the Merchant wants to convert the cryptocurrency into fiat money. Under this system, the payoffs are illustrated in table III.

Intuitively, the Merchant can request the Issuer to exchange for fiat money only when the exchange rate is favorable. Therefore, the Merchant will sell the cryptocurrency in an exchange if the cryptocurrency price is high and use the guaranteed rate provided by the Issuer when the cryptocurrency price is low. If D_m is kept constant, the Merchants' payoff will increase but the Issuer's payoff will decrease. More importantly, the risk-adjusted Merchants' benefit is smaller than the Issuer's loss. Thus, alternative system 3 is less efficient than the proposed system.

TABLE III
PAYOFFS UNDER ALTERNATIVE SYSTEM 3

Alternative 3	
Customers	$V - P + D_n$
Merchants	$CE(\sum^{N_m}(P + (1(\delta_p \geq 0)\delta_p) - C) + D_m)$
Issuer	$CE(B - D + \sum^M \sum^{N_m}(P + 1(\delta_p < 0)\delta_p)))$

Finally, the assumption that the cryptocurrency price is purely random and it has the same distribution over time mean that there is neither an upward nor a downward trend. This makes sense if the cryptocurrency is maintained as a stable coin[8]. In the above analysis, it is assumed that the purchasing activities are small and will not affect the cryptocurrency price. In case these activities are large and frequent enough to affect the cryptocurrency market, these activities could help the Issuer to stabilize the cryptocurrency price. It is because the products are relatively cheaper when the cryptocurrency price is low. More Customer will tend to buy the cryptocurrency and use it to pay for the good. This increase in demand for the currency will act as an automatic stabilizer of the currency price.

In case the cryptocurrency is not a stable coin, there could be an upward or downward trend of the cryptocurrency price between the times when the Issuer redetermines the exchange rate with the Merchant. The system is not designed to hedge this kind of risk, but the payment system and the principles discussed above still work, the difference is that there is an extra risk added to the payoff of the Issuer.

VI. DESIGN EXTENSION

A. Transaction Fee in Non-Native Cryptocurrencies

In common cases, it is compulsory for a Customer to possess a sufficient amount of native cryptocurrency for the purpose of sending transactions on the underlying blockchain, for instance, a Customer needs to acquire some Ethers to be eligible to create transactions on Ethereum. The mandatory prerequisite is a need to guarantee that the network is not subjected to *DDoS*, namely Distributed Denial-of-Service attack, by any of the sources which abuses the bandwidth resources, leading to a temporary or permanent block of the entire network access. However, this prerequisite raises the educative curve for a novice Customer to adapt to the use of cryptocurrencies and curtail their incentives.

Delegate transfer is a special transferal mechanism, one possible implementation is proposed as ERC865 [29] in the Ethereum community, that allows function execution without

[8]Stable coin refers to a cryptocurrency with a fixed exchange rate to fiat money

explicitly paying gas for the transaction fee. The name delegate transfer comes from that mitigating the gas cost of the transfer function is one of its most common use cases. A proxy serves as a delegator to pay the transaction fee in the representation of the Customers who are paying the cryptocurrencies. The Customer signs a raw transaction with their local private key that is kept secretly to themselves and submits the signed transaction to the proxy. The proxy leverages the signature to validate the identity and eligibility of the Customer and broadcasts the signed transaction, resigned by the proxy, to the network attached with the transaction fee in Ether to pay for the gas. This design does not require Customers to own Ethers on the Ethereum blockchain, therefore significantly lessens the difficulty for the Customer to accommodate to the cryptocurrency payment.

To enhance user experiences and expand the cryptocurrency payment adoption, this paper also actively considers employing delegate transfer on the Customer side, reducing troubles for the Customers without owning Ethers. The Issuer is obliged to serve as the proxy and pay for any signed transaction from Customers. The system is then optimized to have the Customers directly submit their signed transactions to one or multiple proxies ran by the Issuers, shortening the latency for the payment function initiated by the Customers, as signing the raw transaction is considered to have nearly zero latency. A cache of the state machine can be applied, imitating the structure of a sidechain at the Issuer side: on one hand the gas is delegatedly paid; and on the other hand, the cache can bolster the solution of an early-confirmation mechanism that shortens the latency of the confirmation for each payment. The early-confirmation mechanism is in great demand, due to the fact that most of the blockchain who has a latent decentralized consensus are prone to have a long waiting time for the confirmations, impeding the prevalent adoption of cryptocurrency payment. The details of this mechanism are to be discussed in the next section, Section VI-B.

Nonetheless, though promising as the ERC865 implementation may sound, it has one critical limitation that is often neglected by most adopters – the implementation is obligated to be inherited in the cryptocurrency smart contract *before* the issuance of the currency, namely the irreversible deployment of the smart contract onto the blockchain. This implies that the cryptocurrencies issued without inheriting ERC865 in advance have no possibility of benefiting from the adopter-friendly delegate transfer. As such, it is judicious to conclude that a *pluggable* delegate transfer, independent from when to be inherited, is an indispensable future work to achieve the final goal of a practical cryptocurrency payment architecture of this paper.

B. Early Payment Confirmation with Sidechain

Illustrated in Figure 7, the design of early confirmation simulates a sidechain along with the main blockchain to cache the states of the transactions on a machine, which is the proxy in this case. When the Customer submits a signed transaction to the cache proxy, the proxy will delegate the transaction to the network if the signature is presented valid, and simultaneously apply the transaction to its state machine rather than halting for the main blockchain to confirm the transaction. A quick response of validity and result can be sent to the Customer because the cache stands as the Proof-of-Authority side chain that is centralized yet efficient. If the main blockchain returns a confirmation, the quick response is finalized, otherwise, a dispute will take place at the proxy, by updating the Customer with the latest transaction result. Owing to the fact that the dispute is very unlikely to happen, as long as the proxy is not compromised, the latency remains inconsequential and the correctness is impregnable, backed by the double confirmation of the two chains.

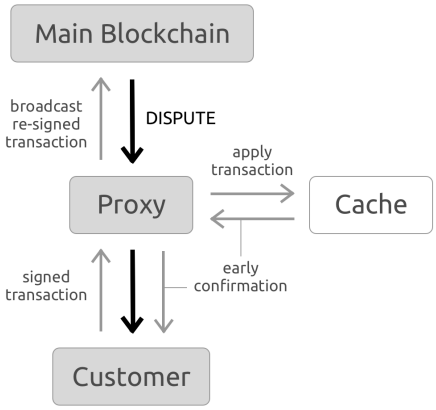

Fig. 7. Early Confirmation with Proxy and Cache

C. Accurate Real Market Exchange Rate with Decentralized Oracle

When the Merchant and Issuer wish to reach their agreement on the cryptocurrency exchange rate on blockchain automatically, a source of the real market exchange rate is in need. Since that the blockchain is an enclosed space without an easy entry for external data to synchronize with the internal states, a special type of smart contract is necessary, named the Oracle smart contract. The Oracle smart contract is dedicated to bridge the data integrity across the external knowledge with the blockchain, and bring in updates of the extrinsic data, such as the price of the cryptocurrency, the temperature, or any data that varies along time and requires a regular update.

However, this approach encompasses an old problem that has long been debated upon, the Oracle Problem. The Oracle Problem is the challenge that describes a trade-off between utilizing a single Oracle contract and multiple Oracle contract. Single Oracle contract can expedite the process of absorbing external data, whereas, there is a possibility that it represents a single-point-of-failure, which can be compromised, misled, and itself malfunctioned. The entire logic within the blockchain space can collapse due to an incorrect input of information from the imperil single Oracle. As an alternative, the system can choose to rely on multiple sources of information, particularly, multiple Oracle contracts. It reduces the chance

for a majority of the Oracles to act faulty, but at the same time largely increase the latency for all the Oracles to deal with inconsistency between them by reaching on a consensus.

A scheme of decentralized Oracle leveraging ERC860 [30], the hierarchical relationships between smart contracts [31], can be considered as a practical solution to this conundrum. The proposed architecture is composed of two types of smart contracts: Custodian and Client. While Client smart contracts vote on the external data they possess, a specific Custodian smart contract aggregates the voting results to finalize a state consensus shared across all parties.

In the case of a practical cryptocurrency payment architecture, the Client smart contracts collect the market exchange rate of the target cryptocurrency, and the Custodian smart contract assembles all the provided data periodically to reach an accurate fiat and cryptocurrency exchange rate used to define the fee for Customers that intend to delegate the payment function to the Issuer.

VII. Conclusion

This paper addresses a solution to the problem of the real world market adoption of cryptocurrencies, which was knotty owing to the unpredictable fluctuation of the cryptocurrency prices. The paper proposes the ERC1586 standard to formulate an interactive flow that removes the risks from the merchants, which stabilizes the exchange rate for both the customers and the merchants, making the solution favorable for adoptions. A proposed Ethereum smart contract interface has been evaluated to testify its effectiveness of risk reduction in actual environments. The results indicate that even though both the initiation and the settlement actions of the ERC1586 might entail additional efforts and costs, those costs are comparatively minor when the disadvantages of risk bearing are considered. Despite that the current experiments and implementations are laid on Ethereum blockchain, the concept is migratable to other blockchain platforms with trivial efforts.

A heuristic economy model is constructed under the scenario that a cryptocurrency has its price oscillating around a mean. The model composes the utility functions respectively for the customers, the merchants, and the cryptocurrency issuers, and derives a total sum of the certainty equivalent payoff. The model leverages the certainty equivalent to demonstrates that the larger the base of the transactions, the less the risk can degrade the payoff of the participant. Therefore, a conclusion is effectively deduced that allocating all the risks to the cryptocurrency issuer can significantly improve the overall utility for all the involved parties. The economy model is technically described in the previously proposed ERC standard.

To consider the optimization of the payment standard, several potential improvements have been discussed as the design extensions, including paying transaction fee in non-native cryptocurrencies, achieving early payment confirmation using sidechain, and being able to periodically update and having an accurate exchange rate of the real market with decentralized oracles. These improvements are regarded as possible future extensions of this paper that can be further integrated to form a more comprehensive ERC1586 standard.

References

[1] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," 2008.

[2] G. Wood, "Ethereum: A secure decentralised generalised transaction ledger," *Ethereum project yellow paper*, vol. 151, pp. 1–32, 2014.

[3] F. P. Hjalmarsson, G. K. Hreioarsson, M. Hamdaqa, and G. Hjalmtysson, "Blockchain-based e-voting system," in *2018 IEEE 11th International Conference on Cloud Computing (CLOUD)*, pp. 983–986, IEEE, 2018.

[4] "Eos.io technical white paper," *See https://github.com/EOSIO*, Accessed 2018-11-30.

[5] V. Buterin, "Chain interoperability," *R3 Research Paper*, 2016.

[6] L. Luu, V. Narayanan, C. Zheng, K. Baweja, S. Gilbert, and P. Saxena, "A secure sharding protocol for open blockchains," in *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, pp. 17–30, ACM, 2016.

[7] C. Decker and R. Wattenhofer, "A fast and scalable payment network with bitcoin duplex micropayment channels," in *Symposium on Self-Stabilizing Systems*, pp. 3–18, Springer, 2015.

[8] J. Poon and T. Dryja, "The bitcoin lightning network: Scalable off-chain instant payments," *See https://lightning. network/lightning-network-paper. pdf*, 2016.

[9] "Raiden network," *See http://raiden.network/*, Accessed 2018-11-10.

[10] J. Poon and V. Buterin, "Plasma: Scalable autonomous smart contracts," *White paper*, 2017.

[11] "Bitpay," *See https://bitpay.com*, Accessed 2018-11-26.

[12] "Coinbase," *See https://www.coinbase.com/merchants*, Accessed 2018-11-26.

[13] "Coinpayment store directory," *See https://www.coinpayments.net/store-directory–1*, Accessed 2018-11-26.

[14] "Tenx," *See https://tenx.tech/en/*, Accessed 2018-11-26.

[15] "Crypto.com," *See https://crypto.com/*, Accessed 2018-10-25.

[16] "Pundi x," *See https://pundix.com/*, Accessed 2018-11-02.

[17] "Tokencard," *See https://tokencard.io*, Accessed 2018-11-26.

[18] A. H. Dyhrberg, "Bitcoin, gold and the dollar–a garch volatility analysis," *Finance Research Letters*, vol. 16, pp. 85–92, 2016.

[19] E. Bouri, G. Azzi, and A. H. Dyhrberg, "On the return-volatility relationship in the bitcoin market around the price crash of 2013," 2016.

[20] C. Baek and M. Elbeck, "Bitcoins as an investment or speculative vehicle? a first look," *Applied Economics Letters*, vol. 22, no. 1, pp. 30–34, 2015.

[21] L. Guo and X. Li, "Risk analysis of cryptocurrency as an alternative asset class," in *Applied Quantitative Finance*, pp. 309–329, Springer, 2017.

[22] N. A. Bakar and S. Rosbi, "High volatility detection method using statistical process control for cryptocurrency exchange rate: A case study of bitcoin," 2017.

[23] K. Chuen, L. David, L. Guo, and Y. Wang, "Cryptocurrency: A new investment opportunity?," 2017.

[24] J. G. Ronca, J. B. Castinado, H. Dolan, T. E. Durbin, and R. H. Thomas, "Cryptocurrency risk detection system," Dec. 17 2015. US Patent App. 14/305,916.

[25] "Erc1586: Risk redistribution standard for stable cryptocurrency payment." https://github.com/ethereum/EIPs/issues/1586.

[26] "Erc20: A standard interface for tokens." https://github.com/ethereum/EIPs/blob/master/EIPS/eip-20.md.

[27] "Ganache," *See https://trufflframework.com/ganache*, Accessed 2018-11-29.

[28] "Eth gas station, transaction cost calculator," *See https://ethgasstation.info/calculatorTxV.php*, Accessed 2018-11-29.

[29] "Erc865: Pay transfers in tokens instead of gas in one transaction." https://github.com/ethereum/EIPs/issues/865.

[30] "Erc860: Custodian-client contract standard." https://github.com/ethereum/EIPs/issues/860.

[31] Y.-C. Hu, T.-T. Lee, D. Chatzopoulos, and P. Hui, "Hierarchical interactions between ethereum smart contracts across testnets," in *Proceedings of the 1st Workshop on Cryptocurrencies and Blockchains for Distributed Systems*, pp. 7–12, ACM, 2018.