Key Takeaways:

1. The proposed solution introduces a payment system based on accrual accounting, reflecting revenue and expenses before actual payment transfers.
2. Advantages of the presented solution include convenience, instant access to payment revenues, regular debentures, and significantly lower transaction fees.
3. The lazy evaluation approach is employed to efficiently process a large number of transactions and minimize computing power usage.
4. The solution utilizes smart contracts on Ethereum or Ethereum-compatible networks to implement the new token standard.
5. The implementation of the solution follows the currently developing IEEE standard for recurring transactions on distributed ledger technologies.

Automatic recurring payments have become a critical revenue stream for businesses in almost every sector, providing a reliable incremental cash flow to support business processes. Some real-world use cases are displayed in the following figure.

However, customer payments made on centralized platforms are subject to data breaches, and automated payments can easily be disrupted due to insufficient funds, expired debit or credit cards, and delays imposed by centralized banks. High transaction fees can take a bite out of payments, amounting to a significant reduction in revenues.

For consumers, the convenience of automatic payments is offset by risks to sensitive data, making them vulnerable to identity theft and other mischief, and subjecting them to unauthorized sale of data to advertisers and other entities.

While decentralized smart-contract payment systems do exist, they rely on a cash-based accounting system that is inadequate for executing ongoing recurring payments.

Learn how accrual-based accounting can offer a solution that eliminates or significantly reduces the drawbacks of existing decentralized payment systems via smart contracts on the blockchain/Web 3.0 networks.

Challenges of Recurring Payments

Subscription-based business models have been thriving globally for years, as exemplified by familiar centralized platforms such as streaming services, mobile service providers, and a variety of other platforms with automated recurring monthly fees.

However, in decentralized public networks, a system for ongoing recurring payments has not been well thought out. Crypto wallet users currently face challenges when it comes to remitting and receiving recurring payments – they lack convenient, fast, and efficient payment methods, and they often involve transaction fees.

But imagine if your recurring decentralized payments were made automatically, with minimal transaction fees, saving you time and money. We present a solution that has the potential to significantly impact decentralized transactions.

Recurring transactions can be defined in terms of payment frequency (weekly, biweekly, monthly, etc.), fixed amounts, the prolongation of payments, the number of persons involved (individual, family, corporate subscriptions), a list of provided services, termination rules, and other conditions.

However, problems arise when a customer does not have enough available funds to make a timely payment. In that case, we need to take into account the possibility of:

- debt formation – a necessary procedure to keep payments up-to-date and to charge late fees for late payments;
- debt repayment;
- early termination.

The Status Quo vs Our Approach

Current options for decentralized transactions

Let’s recall that a smart contract

is a self-executing contract with the terms of the agreement directly written into code. It is typically built on a blockchain platform and allows for the automation of transactions and agreements without the need for intermediaries.

Smart contracts are executed by nodes. The process of running a smart contract involves the following steps:

1. Choosing a suitable blockchain platform that supports smart contracts.
2. Writing code for the smart contract using a programming language that is compatible with the chosen blockchain platform.
3. Compiling the contract using the appropriate compiler for the chosen programming language. This step generates the bytecode that can be executed on the blockchain.
4. Deploying the compiled smart contract onto the blockchain. This process involves creating a transaction that contains the bytecode and sending it to the network for inclusion in a block.
5. Interacting with the contract: once the smart contract is deployed, it becomes a part of the blockchain and can be accessed and interacted with by users. Users can send transactions to the contract, triggering the execution of the predefined code.
6. Executing the contract. When a transaction is sent to the smart contract, the code within the contract is executed on the blockchain nodes. The contract's logic is automatically enforced, and the contract performs the actions specified in the code.
7. Verifying and validating the contract. The decentralized nature of blockchain ensures that every node in the network verifies and validates the execution of the smart contract.

The problem with popular smart contract platforms is that they are not designed to accept ongoing recurring payments. Decentralized public networks use a cash-based accounting system for payments, whose main advantages are ease of tracking and accurate reflection of account balances. The cash method of accounting records transactions whenever cash is received or paid. This approach is typical for small businesses with a relatively small turnover of funds and a small customer base.

There currently exist two types of payment solutions:

- Custodial solutions – these solutions depend on the crypto-wallet owner depositing funds into a smart contract account, making it possible for the funds to be freely used without the consent of their true owner. This method creates an opportunity to establish a system for regular or on-demand payments.
- Non-custodial solutions – here, the customer sends a batch of transactions to the supplier, who stores them off-chain. For each billing period, the supplier selects and submits a single transaction to the Ethereum blockchain or an EVM-compatible network. A smart contract verifies the transaction's validity and initiates the payment. While this approach does not reduce transaction fees, it offers the convenience of fully automated crypto wallets.

Our proposed solution

An alternative to a cash-based accounting system is the accrual system. Accrual accounting has long been used by traditional financial institutions, but it has not yet been used in decentralized networks. We propose an accrual-based system that operates in a non-custodial manner. In our proposed system, revenue and expenses appear whenever a product or service is delivered to a customer, but before the payment amount is actually transferred. Our proposed smart contract on Ethereum or Ethereum-compatible networks provides a new token standard.

Advantages, especially for enterprises and state authorities as well as social interaction purposes, include:

- convenience and ease of use;
- instant access to payment revenues;
- regular debentures and recurring payments;
- significantly lower transaction fees.

Our High-Level Design

Conventional blockchain financial information flow

Here is how financial information is typically processed on the blockchain:

1. Financial and other transactions flow from users to nodes on the network, forming a common transaction pool.

2. Following a specific consensus protocol, transactions from the pool end up in blockchain blocks. In effect, the blockchain acts as a payment transaction ledger.
3. Each node receives a new block and sequentially applies transactions from the block to its version of the blockchain state.
4. The blockchain state records the amount of funds in the accounts of all users.

In blockchain transactions, nodes perform calculations and remember the results only when transactions from the next block are applied. Until there is a new block, the nodes do nothing with the blockchain state. Put simply, the blockchain state is a ledger storing the number of tokens in the accounts of all users.

How our solution differs

Our proposed approach to making regular payments does not change the blockchain scheme described above. As before, the amounts in the accounts are updated only when transactions are processed. Even if a regular payment is made once every second, the system does not update the account amounts with the same frequency.

For example: If Alice pays Bob one token per second, the system does not process the payments every second. The status of both Alice's and Bob's accounts are recalculated all at once (so-called lazy evaluation

or call-by-need method), for a given period, at the moment when any transaction from Bob or Alice requires knowledge of their account status.

In other words, to process regular payments, our solution adds a recursive algorithm for finding and accounting for all regular payments associated with the participating accounts, at the moment of transaction processing. All linked regular payments are applied in their correct chronological order. This coincides with the concept of lazy computation in programming – the value of a variable is not computed until the variable is used.

Activity Diagram

[

Image2

589×826 111 KB

](https://ethresear.ch/uploads/default/original/2X/d/d98593b3774f8abb6e9c19293051e966347ddf46.png)

Payment Logic Design

Unlike unsecured transfer transactions that frequently result in payment delays, our proposed solution enables regular payments without freezing funds. Payments that are nearing their due date, but for which there are insufficient available funds, are called "short-term payment commitments," which are sent to the debt queue. When funds are insufficient, suppliers can decide whether to extend credit or terminate the contract. To terminate, they must send a termination transaction, to prevent a default to granting credit.

In the same way, customers may send a transaction to terminate their subscription and stop future payments.

To implement this system, certain elements must be considered:

- the final balance after completed payments;
- a provision for partial payments;
- a procedure for late payments.

The table represents an example where transactions on a customer's account gradually pay off her debt. A capital letter indicates the payee and an asterisk* marks the separable transaction – in this case, it is the first in the queue. When funds appear in the account, the account is reviewed and the oldest debt is paid first. Subsequent debts remain in the account queue and are paid when funds appear, from oldest to newest, until all are paid in full.

Debt queue

Funds added to Alice's account

Alice's account status

B*(20), C(100), B(1), C(2)

5

0

B*(15), C(100), B(1), C(2)

17

1

C(100), C(2)

50

49

C(100)

50

99

C(100)

10

9

Let’s look at the first row. Alice adds 5 tokens to her account, and they are applied to her oldest debt: B*(20). In the second row, we see that Alice is able to pay off her first debt, and also the third one. At that time, she cannot pay off the second debt, so it remains the same. In subsequent rows, we see that Alice is able to pay off all her debts over time.

Implementation and Future Work

The Recurring Transactions on the Distributed Ledger Technologies (DLTs) Working Group for the development of atoken standard (project P3228) has been approved by the IEEE Computer Society/Blockchain and Distributed Ledgers (C/BDL) Standards Committee. The purpose of this standard is to provide a resource for implementing blockchain and distributed ledger-based recurring payment methods in relevant industries, including public services, banking, finance, insurance, real estate, commercial payments, payrolls, and online services.

Our proposed payment and token accounting solution is implemented as a smart contract on the Ethereum network, or on any system with a compatible virtual machine. Two smart contract versions were developed for backward compatibility with ERC-20 and ERC-777 standards, respectively. The tokens issued on their basis have successfully demonstrated the declared properties.

Our idea was presented at the 2022 IEEE 1st Global Emerging Technology Blockchain Forum: Blockchain & Beyond (iGETblockchain), in Irvine, CA, USA.

Future functionality of our solution may expand to include:

- suspension of subscriptions;
- accrual of late fees on debts;
- recalculation of payments based on fiat exchange rates;
- creation of “oracles” to enable changes in the payment amount during the subscription period;
- confirmation of work performed;
- creation of web applications.

Decentralized solutions like the one we propose can expand the scope of opportunities and services provided to users of cryptocurrency platforms, without sacrificing important advantages such as transparency and security.

Learn more about

[recurring payment contracts

](https://github.com/waterfall-network/recurring-payment-contract).