August 2022

AUTOPAYMENTS VIA ACCOUNT ABSTRACTION

ANDREW BEAMS
VISA

RANJIT KUMARESAN
VISA

MOHAMMAD MOHSEN MINAEI BIDGOLI
VISA

MAHDI ZAMANI
VISA

SRINIVASAN RAGHURAMAN
VISA

See next page for additional authors

Inventor(s)
ANDREW BEAMS, RANJIT KUMARESAN, MOHAMMAD MOHSEN MINAEI BIDGOLI, MAHDI ZAMANI, SRINIVASAN RAGHURAMAN, and WANYUN GU

TITLE: “AUTOPAYMENTS VIA ACCOUNT ABSTRACTION”

VISA

ANDREW BEAMS

RANJIT KUMARESAN

MOHAMMAD MOHSEN MINAEI BIDGOLI

MAHDI ZAMANI

SRINIVASAN RAGHURAMAN

WANYUN GU

TECHNICAL FIELD

[0001] This disclosure relates generally to the field of automatic payment processing. More particularly, the disclosure simplifies the user's ability to make autopayments without sharing the private key.

BACKGROUND

[0002] Generally, autopayments are the payments that may be sent to one of a user's billers from the user bank account or credit card account. The user can authorize an automatic bill payment to be made using your debit card, credit card, savings account or money market account. The amount due for the payment is collected automatically by the biller according to your payment schedule. Automated payments can be used to pay different types of bills. For example, the user may be able to use automatic bill payments to pay to mortgage, utility bills, cell phone bill, streaming subscriptions, credit card bills, auto loan payment and the like. The user may inform the bank or credit union how much to pay and when to pay in each month. The bank then authorizes that amount to be deducted from the selected account of the user each month and transferred to the company that the user needs to pay. Thus, the above-mentioned scenario is a straightforward approach in net banking application. However, autopayments on blockchain platform is different in which the protocol that may allow block chain users to enrol in autopayment services that may pull and push money to and from merchants account.

[0003] One of the existing technologies disclose cryptocurrency exchange or a Digital Currency Exchange (DCE) that may be a business that allows customers to trade cryptocurrencies or digital currencies for other assets, such as conventional money or other digital currencies. A private key is also used in cryptocurrency transactions in order to show ownership of a blockchain address. For instance, when the user makes a purchase, the cryptocurrency acquired is automatically stored in his/her exchange-hosted wallet, which is typically custodial which means the exchange has control of user private keys. The private key is an extremely large number that is used in cryptography, similar to a password. The Private keys are used to create digital signatures that can easily be verified, without revealing the private key. The custodial wallets can be defined as a wallet in which the private keys are held by a third party. In other words, the third party has

full control over user's funds while the user only has to give permission to send or receive payments. Due to which the security level is low in custodial wallets unless the authoritative party implements strong security measures. For instance, if user wants to make payment to a merchant, the user has to call a transfer function on the third-party server using the user private key. Upon receiving the user request for transferring, the third-party server will push the amount from user account to the merchant account as shown in **FIG.1A** (Prior art). In another example, if the user has a custodial wallet, the private key of the user is managed by the third-party wallet provider. In this way, the user can schedule the payments in advance and thus can rely on the custodial wallet to make recurring payments on behalf of the user.

[0004] Another technology discloses a non-custodial wallet that allows the user to own and control the private keys. This gives the user full access to the user funds. The non-custodial wallets give the user complete charge of digital assets. For instance, if the user wants to make the payment to the merchant, then user can login to his/her account and use the private key associated with the user to push the payment to the merchant account as shown in **FIG.1B** (prior art). However, if the merchant makes a request to the user, the merchant may not be sure if the user completes the requested transaction. This scenario is difficult when the user is not online, and merchant is not aware whether the user make the requested transaction. Thus, autopayment industry is exploring the method in which the non-custodial wallets in which the users and the private key associated with the user are not involved while making the transaction to the merchant.

SUMMARY

According to some non-limiting embodiments, the present disclosure discloses autopayment processing that simplifies the user's ability to make autopayments without making use of the private key associated with the user. The objective of the present disclosure is to make autopayments to the merchant without revealing the private key of user to any third-party server. However, without making use of user's private key, a smart contract can make an autopayment on behalf of the user to the merchant to whom the user wishes to make the payment. In other words, the smart contract will make the automatic payment to merchants associated with the user upon receiving the approval from the user. The approval here may be the user providing the details that is allowed for the smart contract to make the autopayments on behalf of the user. For instance, the

user may check list the transaction that the smart contract may process without making use of the user private key information. In other words, the user may allow the transaction, that the smart contract may process the payment on behalf of the user when there is a request from the merchant's side. As an example, when the user is off-line, and the merchant sends the request to the user for processing the payment. Since the user has allowed the requested payment prior, the smart contract will process the payment requested by the merchant in the absence of the user.

[0005] While some people store large amounts of crypto on exchange accounts, many feel more comfortable with a non-custodial wallet, which eliminates a third-party between user and his cryptocurrency. The present research work provides an advantage of using such non-custodial crypto wallets that may give a user complete control of their private keys and funds. The present disclosure enables a user to create an allowed list, through which the user can pre-authorize transactions with the selected payees such as merchant, another user and the like. Therefore, even when the user is not available to make a certain recurring transaction, autopayments from the user account to the requested merchant account occur if the merchant is listed in the users allowed list. The allowed list acts as a substitute for authorization via a private key, due to which user can make a timely or recurring or scheduled payment even when he is offline or in a non-reachable location. Hence, the present disclosure eliminates the use of private key associated with the user, while using the non-custodial wallets for payments that are listed in the allowed list.

[0006] These and other features and characteristics of the present invention, as well as the methods of operation and functions of the related elements of structures and the combination of parts and economies of manufacture, will become more apparent upon consideration of the following description and the appended claims with reference to the accompanying drawings, all of which form a part of this specification, wherein like reference numerals designate corresponding parts in the various figures. It is to be expressly understood, however, that the drawings are for the purpose of illustration and description only and are not intended as a definition of the limits of the invention. As used in the specification and the claims, the singular form of "a," "an," and "the" include plural referents unless the context clearly dictates otherwise.

BRIEF DESCRIPTION OF THE DRAWINGS AND APPENDICES

[0007] Additional advantages and details of non-limiting embodiments are explained in greater detail below with reference to the exemplary embodiments that are illustrated in the accompanying schematic figures, in which:

[0008] FIG. 1A [prior art] discloses an exemplary architecture of autopayments using custodial wallets.

[0009] FIG. 1B [prior art] discloses an exemplary architecture of autopayments using non-custodial wallets.

[0010] FIG. 2A discloses an exemplary architecture of account abstraction representation disclosing stage 1 process according to some principles of the present disclosure;

[0011] FIG.2B discloses an exemplary architecture of processing the requested merchant's transaction (stage 2 of account abstraction), according to some principles of the present disclosure; and

[0012] FIG.3 shows a flowchart that illustrating a method of processing the autopayments using account abstraction, in accordance with some embodiments of the present disclosure.

DESCRIPTION OF THE DISCLOSURE

[0013] In the present document, the word "exemplary" is used herein to mean "serving as an example, instance, or illustration." Any embodiment or implementation of the present subject matter described herein as "exemplary" is not necessarily to be construed as preferred or advantageous over other embodiments.

[0014] While the disclosure is susceptible to various modifications and alternative forms, specific embodiment thereof has been shown by way of example in the drawings and will be described in detail below. It should be understood, however that it is not intended to limit the disclosure to the particular forms disclosed, but on the contrary, the disclosure is to cover all modifications, equivalents, and alternative falling within the spirit and the scope of the disclosure.

[0015] The terms “comprises”, “comprising”, or any other variations thereof, are intended to cover a non-exclusive inclusion, such that a setup, device or method that comprises a list of components or steps does not include only those components or steps but may include other components or steps not expressly listed or inherent to such setup or device or method. In other words, one or more elements in a device or system or apparatus proceeded by “comprises… a” does not, without more constraints, preclude the existence of other elements or additional elements in the device or system or apparatus.

[0016] The terms "an embodiment", "embodiment", "embodiments", "the embodiment", "the embodiments", "one or more embodiments", "some embodiments", and "one embodiment" mean "one or more (but not all) embodiments of the invention(s)" unless expressly specified otherwise.

[0017] The terms "including", "comprising", “having” and variations thereof mean "including but not limited to", unless expressly specified otherwise.

[0018] As used herein, the terms “communication” and “communicate” may refer to the reception, receipt, transmission, transfer, provision, and/or the like of information (e.g., data, signals, messages, instructions, commands, and/or the like). For one unit (e.g., a device, a system, a component of a device or system, combinations thereof, and/or the like) to be in communication with another unit means that the one unit is able to directly or indirectly receive information from and/or transmit information to the other unit. This may refer to a direct or indirect connection (e.g., a direct communication connection, an indirect communication connection, and/or the like) that is wired and/or wireless in nature. Additionally, two units may be in communication with each other even though the information transmitted may be modified, processed, relayed, and/or routed between the first and second unit. For example, a first unit may be in communication with a second unit even though the first unit passively receives information and does not actively transmit information to the second unit. As another example, a first unit may be in communication with a second unit if at least one intermediary unit (e.g., a third unit located between the first unit and the second unit) processes information received from the first unit and communicates the processed information to the second unit. In some non-limiting embodiments, a message may refer to a network packet (e.g., a data packet and/or the like) that includes data. It will be appreciated that numerous other arrangements are possible.

[0019] As used herein, the term "merchant" may refer to an individual or entity that provides goods and/or services, or access to goods and/or services, to customers based on a transaction, such as a payment transaction. The term "merchant" or "merchant system" may also refer to one or more computer systems operated by or on behalf of a merchant, such as a server computer executing one or more software applications. A "point-of-sale (POS) system," as used herein, may refer to one or more computers and/or peripheral devices used by a merchant to engage in payment transactions with customers, including one or more card readers, near-field communication (NFC) receivers, RFID receivers, and/or other contactless transceivers or receivers, contact-based receivers, payment terminals, computers, servers, input devices, and/or other like devices that can be used to initiate a payment transaction.

[0020] As used herein, the term payment card may be (e.g., a credit or debit card), a gift card, a smartcard, smart media, a payroll card, a healthcare card, a wrist band, a machine-readable medium containing account information, a keychain device or fob, an RFID transponder, a retailer discount or loyalty card, a mobile device executing an electronic wallet application, a personal digital assistant, a security card, an access card, a wireless terminal, and/or a transponder, as examples.

[0021] As used herein, the term "computing device" may refer to one or more electronic devices that are configured to directly or indirectly communicate with or over one or more networks. A computing device may be a mobile or portable computing device, a desktop computer, a server, and/or the like. Furthermore, the term "computer" may refer to any computing device that includes the necessary components to receive, process, and output data, and normally includes a display, a processor, a memory, an input device, and a network interface. A "computing system" may include one or more computing devices or computers. An "application" or "Application Program Interface" (API) refers to computer code or other data sorted on a computer-readable medium that may be executed by a processor to facilitate the interaction between software components, such as a client-side front-end and/or server-side back-end for receiving data from the client. An "interface" refers to a generated display, such as one or more graphical user interfaces (GUIs) with which a user may interact, either directly or indirectly (e.g., through a keyboard, mouse, touchscreen, etc.). Further, multiple computers, e.g., servers, or other

computerized devices, such as an autonomous vehicle including a vehicle computing system, directly or indirectly communicating in the network environment may constitute a “system” or a “computing system”.

[0022] It will be apparent that systems and/or methods, described herein, can be implemented in different forms of hardware, software, or a combination of hardware and software. The actual specialized control hardware or software code used to implement these systems and/or methods is not limiting of the implementations. Thus, the operation and behavior of the systems and/or methods are described herein without reference to specific software code, it being understood that software and hardware can be designed to implement the systems and/or methods based on the description herein.

[0023] FIG. 2A discloses an exemplary architecture of performing autopayment using account abstraction representation disclosing stage 1 process according to some principles of the present disclosure.

[0024] FIG. 2A shows an exemplary architecture of autopayment process using account abstraction representation comprising a merchant and merchant’s website, a user wallet, a delegatable account smart contract, an autopayment smart contract and a token smart contract. The user wallet provides user with a digital solution for securely storing and managing blockchain assets and cryptocurrencies. The user wallet allows users to send, receive, and trade cryptocurrencies. Some cryptocurrency wallets may only provide support for a single cryptocurrency, many are multi-asset solutions, allowing users to hold multiple cryptocurrencies, including bitcoin, bitcoin cash, ethereum, and litecoin, among many others. These solutions ensure that the owner of the cryptocurrencies and blockchain assets is the only entity who can access the funds by requiring elaborate passwords and other security measures. The user can view or access cryptocurrency wallets from smartphones and computers. The cryptocurrency wallets of the user do not physically store the blockchain assets. Instead, the wallets store public and private keys. The public keys are segments of digital code that are attached to a decentralized blockchain, almost like a bank account number. Private keys are also pieces of digital code, but are unique to an individual’s cryptocurrency wallet, similar to an ATM PIN code. The private keys match and prove

ownership of public keys. The user uses their private keys to conduct all transactions with the cryptocurrency that they own.

[0025] In some embodiments, the merchant may be the biller associated with the user, who bills the user via the merchant website. For instance, consider that the user has to pay his/her car loan by 4th of every month. Thus, the merchant associated with the user who is meant to receive the payment of car loan may send the request to the user on 4th of every month. Thereafter, the user may be able to use automatic bill payment mechanism to transfer the amount to respective merchant accounts based on the payment requests received from respective merchants related to mortgage, utility bills, cell phone bill, streaming subscriptions, credit card bills, auto loan payment and the like.

[0026] In some embodiments, delegatable account smart contract is a smart contract that defines delegated signing rights, which needs to be triggered to successfully sign a message. In other words, the delegatable account smart contact is a smart contract that allows the user to allow third party smart contracts to execute transactions on the user's behalf. Particularly, the user may deploy delegatable account contract in his/her system in order to sign the rights.

[0027] In some embodiments, an autopayment smart contract may be programs stored on a blockchain that run when predetermined conditions are met. The merchant associated with the user or the merchant who request to process the transaction from user may deploy the autopayment smart contract in his/her system. When the merchant triggers the request to approve autopayments on user’s non-custodial wallet, the autopayment smart contract may process the transaction that may be allowed by the user prior. Hence, in the absence of the user, the autopayment smart contract can process the requested payment from the merchant. Further, the user can set few restrictions on his/her account such as, but not limited to, merchant cannot charge more than once in a month, can set limit to his transaction (as an example, charge must be less than $200), the user may have the rights to allow certain transaction and can put those transactions under allowed list. Based on the restriction set by the user, the smart contract may process the allowed or approved transaction and make the autopayment to the requested merchant.

[0028] In some embodiments, a token smart contract or also known as a token contract may be a special type of smart contract, which maps blockchain addresses to balances of value units or tokens. These software programs hold code, which specifies a set of functions and attributes of the value units, created and managed by the contract (as an example Ethereum).

[0029] In FIG.2A, the architecture discloses first stage of account abstraction representation of non-custodial wallet of the user. For instance, if the user has to pay his/her utility bills, home loan by 5^{th} of every month and the like. The user wishes to automate the payment process in such scenarios. In other words, the user wishes to use a smart contract to process the payment on his behalf, without sharing the private key of the user to a third party, as there is high risk of hacking the user's account due to the transfer of private keys via network. To initiate the autopayment to the merchants associated with the user, the user may firstly deploy the delegatable account in his/ her system. Secondly, the merchant who requests the autopayment from the user may deploy the autopayment smart contract in the merchant system. To enable autopayments, the user visits a page on the merchant's website, which triggers the request to approve the autopayments on the user's non-custodial wallet. Thereafter, the autopayment smart contract can process the requested payment made by the merchant without making use of private key associated with the user. The autopayment smart contract can make the payment on behalf of the user without making use of the private key associated with the user by looking at the allowed list that was made by the user. In other words, when the merchant requests the payment from the user, the smart contract may look into the allowed list of transactions approved by the user. If the merchant who has requested the payment is in the allowed list of the user, then the smart contract may proceed with settling the payment to the merchant's account without requiring the user's involvement and the private key of the user. The allowed list of the user may contain the list of merchants and transaction details of the merchant whom the user wishes to make the payment. The user can create the allowed list prior so that when the user is not online and there is a request from the merchant, the smart contract can make the payment on behalf of the user by just looking into the allowed list of the user. Further, the allowed list may be created based on certain restrictions as desired by the user. For instance, the user can set few restrictions on his/her account such as, but not limited to, a merchant cannot charge more than once in a month, can set limit to his transaction (as an example, charge must be less than $200), and the like. Based on the restriction set by the user, the smart contract may process

the allowed transaction and make the autopayment to the requested merchant. As an example, the merchant whose car loan is due on 5th of every month may trigger the request on the non-custodial wallet of the user. If the user is in offline state, the smart contract may process the requested transaction if the merchant details (merchant associated with car loan) is mentioned in allowed list. The user then later can see the transaction made by the smart contract to the merchants when the user is online. The merchant triggering the request and the process of processing the request is reflected in FIG.2B which is explained below.

[0030] FIG.2B discloses an exemplary architecture of processing the requested merchant's transaction (Stage 2 of account abstraction), according to some principles of the present disclosure;

[0031] FIG.2B discloses a scenario when the merchant requests the autopayment function, the smart contract initiates a push payment when the merchant details are in the user allowed list. For instance, consider the user has to pay his/her car loan every month, but if the merchant has requested for approval of deduction of car loan from the user when the user is offline, then the transaction may not be processed. Thus, the user creates an allow-list prior so that during the absence of the user, the smart contract can check if the requested merchant details is present in the allowed list of the user and proceed with the payment. Upon checking the allowed list of the user, the smart contract can make the requested payment automatically.

[0032] In the present disclosure, the delegatable accounts supports not only autopayments and pull payments in general, but also third-party account recovery services where multiple parties have to consent to initiate an account recovery. Third party asset managers with restrictions on the specific ERC tokens which they can manage and how they can trade the tokens.

[0033] FIG.3 shows a flowchart that illustrating a method of processing the autopayments using account abstraction, in accordance with some embodiments of the present disclosure.

[0034] As illustrated in **FIG. 3**, method **300a** includes one or more blocks illustrating a method for processing the autopayments using account abstraction. The method **300a** may be described in the general context of computer-executable instructions. Generally, computer-executable instructions can include routines, programs, objects, components, data structures, procedures, modules, and functions, which perform functions or implement abstract data types.

[0035] The order in which the method **300a** is described is not intended to be construed as a limitation, and any number of the described method blocks can be combined in any order to implement the method **300a**. Additionally, individual blocks may be deleted from the methods without departing from the scope of the subject matter described herein. Furthermore, the method **300a** can be implemented in any suitable hardware, software, firmware, or combination thereof.

[0036] At **block 301**, the method **300a** may include requesting the user to approve the autopayment on the user's non-custodial wallet. For instance, any merchant associated with the user may send the request on non-custodial wallet of the user to process the transaction.

[0037] At **block 303**, the method **300a** may include checking, by the smart contract, whether the details of the merchant who requested the payment are present in the allowed list of the user. If the merchant details are in the allowed list, the method proceeds to block **305** via **'YES'**. If the merchant details are not in allowed list, the method flow to block **307** via **'NO'**.

[0038] At **block 30**5, the method **300a** may include processing, by the smart contract, the requested payment from the user account to merchant's account. For instance, if the merchant's details found in the allowed list, then the smart contract will transfer the amount from user's amount to the merchant's account.

[0039] At **block 307**, the method **300a** may include stopping, by the smart contract, the transaction as the merchant's details are not present in the user's allowed list.

AUTOPAYMENTS VIA ACCOUNT ABSTRACTION

ABSTRACT

The present disclosure focuses to simplify the user's ability to make autopayments without making use of the private key associated with the user while using a non-custodial wallet. The present disclosure describes that without making use of user's private key, a smart contract can make an autopayment on behalf of the user to the merchant to whom the user wishes to make the payment. In other words, the smart contract will make the automatic payment to merchants associated with the user if the merchant's details are present in the allowed list of the user, else, the smart contract may reject the transaction.

FIG.2A and FIG.2B

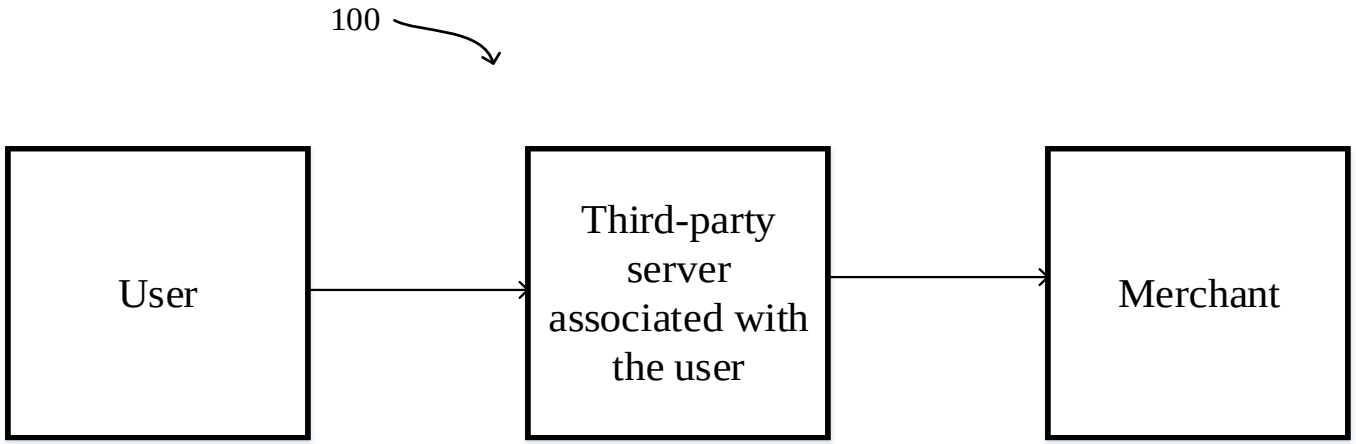

FIG.1A [Prior Art]

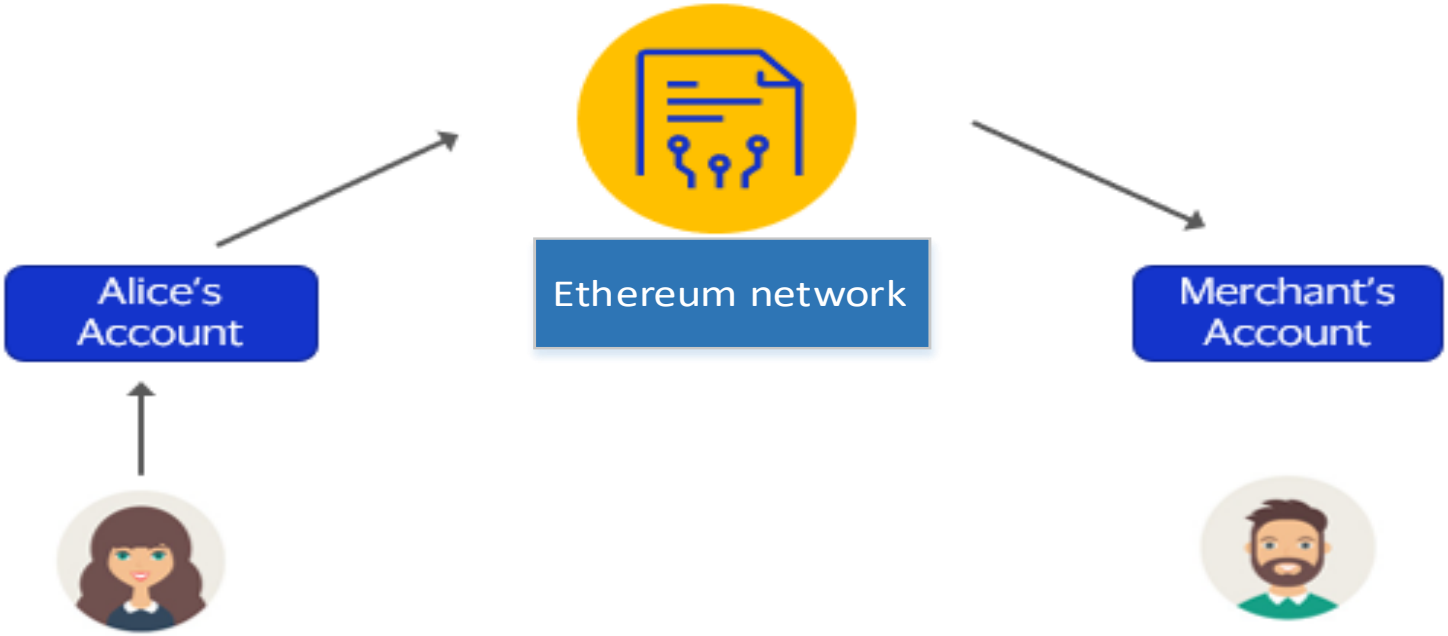

FIG.1B [Prior Art]

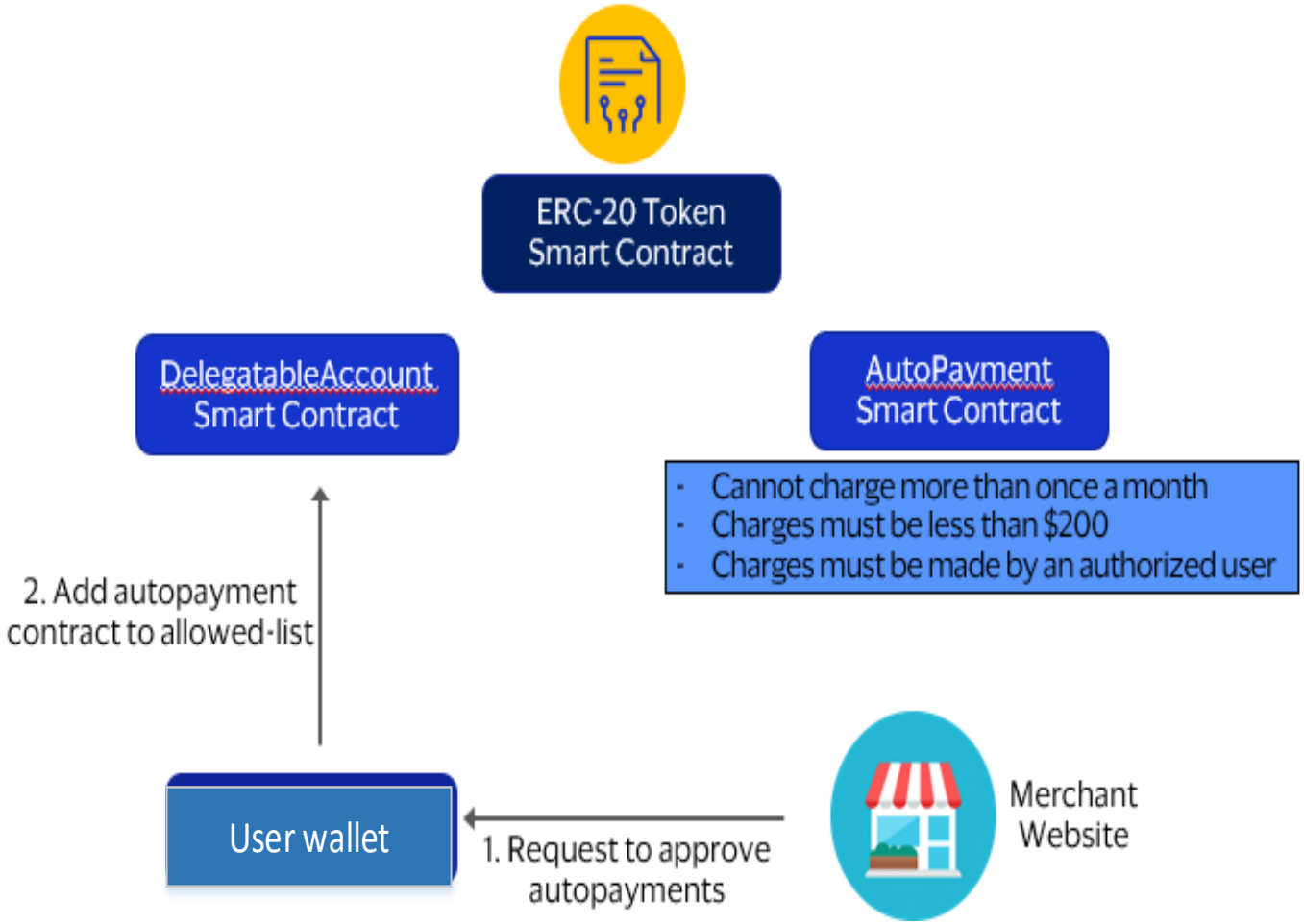
ERC-20 Token
Smart Contract
DelegatableAccount
Smart Contract
AutoPayment
Smart Contract
Cannot charge more than once a month
Charges must be less than $200
Charges must be made by an authorized user
2. Add autopayment
contract to allowed-list
User wallet
1. Request to approve
autopayments
Merchant
Website

FIG.2A

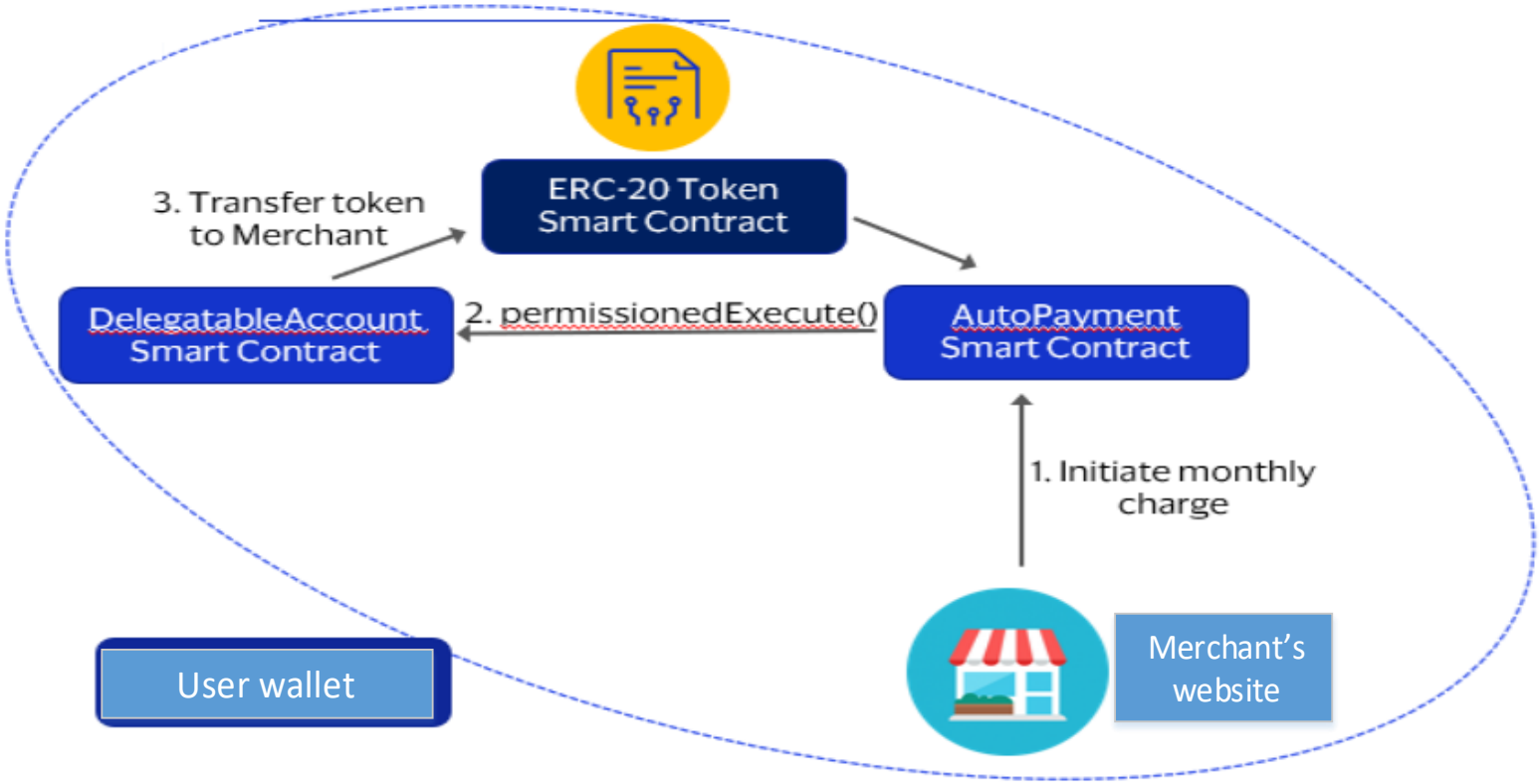
ERC-20 Token
Smart Contract
3. Transfer token
to Merchant
DelegatableAccount
Smart Contract
2. permissionedExecute()
AutoPayment
Smart Contract
1. Initiate monthly
charge
User wallet
Merchant's
website

FIG.2B

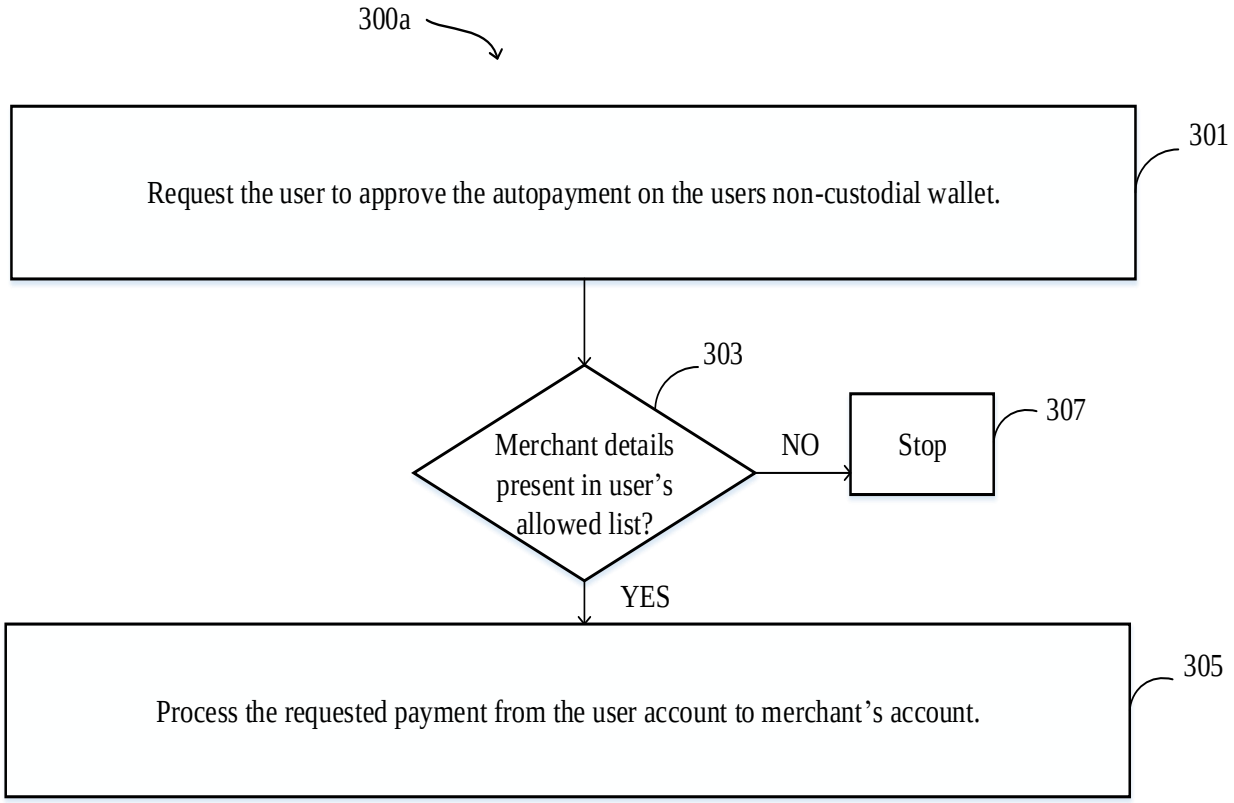
300a
Request the user to approve the autopayment on the users non-custodial wallet.
301
303
Merchant details present in user's allowed list?
NO
Stop
307
YES
Process the requested payment from the user account to merchant's account.
305

FIG.3