November 2023

DIGITAL CLIENT IDENTITY AND MANAGEMENT USING BLOCKCHAIN

Avishek Prasad
VISA

Arian Sani
VISA

Si Ying Ong
Visa

Nashita Guntaguli
VISA

Sripad Rampally
VISA

DIGITAL CLIENT IDENTITY AND MANAGEMENT USING BLOCKCHAIN

VISA

INVENTORS:

1. **Avishek Prasad**
2. **Arian Sani**
3. **Si Ying Ong**
4. **Nashita Guntaguli**
5. **Sripad Rampally**

TECHNICAL FIELD

[0001] The present subject matter is, in general, related to securing a client identity, and particularly, to techniques for managing digital client identity using blockchain.

BACKGROUND

[0002] Client onboarding is a process of integrating new clients into an organization or a business by addressing their questions and concerns with the goal of establishing a strong and productive working relationship. The organization ensures that the new clients have a comprehensive understanding of all the services available to them. The client onboarding is a crucial part or one of the most important functions for any organization or business as the client onboarding directly affects client's experience with the organizations and subsequently, it affects organization's profitability. Further, the client onboarding process is an important step in establishing a long-term relationship between the clients and the organization.

[0003] The client onboarding is performed in different areas or domains. For instance, the clients can be onboarded into payment network domains to conduct their business with multiple stakeholders e.g., merchants, banks, issuers, etc. In general, a payment network is an association of member banks that facilitates payment transactions between the merchants and the issuers.

[0004] In conventional techniques, the process of client onboarding into the payment networks may involve multiple internal parties/departments which in turn makes the onboarding process lengthy and time consuming. In other words, the process of onboarding clients into the payment networks is a cumbersome process and it may take a lot of processing time ranging from two months to 12 months (a year) because of background checks. The reason for such lengthy processing time is multiple onboarding applications/entities work independently i.e., they do not share information and even if they work interdependently, they may share redundant information. Also, manual effort may be required to trace lifecycle of the information because of the various entities involved in the onboarding process. Further, the client onboarding process may demand highly secure infrastructure to secure sensitive information of clients and with many entities involved in a payment system, maintaining data consistency across the process is hard.

[0005] Due to the above-mentioned reasons, the onboard process negatively impacts client's go-to market strategy or service implementation strategies which are time sensitive to current

market conditions. Such onboarding process may negatively impact multiple stakeholders such as payment gateways, a card network, payment processors, issuers, merchants etc. During the process of client onboarding, customers may face difficulty in using services provided by the clients thereby degrading customer experience and eventually impacting the client's business.

[0006] The information disclosed in the background section of the disclosure is only for enhancement of understanding of the general background of the invention and should not be taken as an acknowledgement or any form of suggestion that this information forms the prior art already known to a person skilled in the art.

BRIEF DESCRIPTION OF THE DRAWINGS

[0007] The accompanying drawings, which are incorporated in and constitute a part of this disclosure, illustrate exemplary embodiments and, together with the description, explain the disclosed principles. In the figures, the left-most digit(s) of a reference number identifies the figure in which the reference number first appears. The same numbers are used throughout the figures to reference like features and components. Some embodiments of device or system and/or methods in accordance with embodiments of the present subject matter are now described, by way of example only, and with reference to the accompanying figures, in which:

[0008] **Figure 1** shows a schematic representation showing a sequence of steps involved in onboarding clients into a payment network, in accordance with some embodiments consistent with the present disclosure.

[0009] **Figure 2** shows an exemplary flow chart illustrating method steps for onboarding client into the payment network using permissioned blockchain, in accordance with some embodiments consistent with the present disclosure.

[0010] **Figure 3** illustrates a distributed ledger showing details of all transactions performed to onboard a client, in accordance with some embodiments consistent with the present disclosure.

[0011] **Figure 4** illustrates an implementation of the proposed technique in an Enterprise Ethereum (permissioned) blockchain network for a financial application, in accordance with some embodiments consistent with the present disclosure.

[0012] **Figure 5** illustrates a block diagram of a use case for onboarding clients in a Token Service (TS) system, in accordance with some embodiments consistent with the present disclosure.

[0013] The figures depict embodiments of the disclosure for purposes of illustration only. One skilled in the art will readily recognize from the following description that alternative embodiments of the structures and methods illustrated herein may be employed without departing from the principles of the disclosure described herein.

DESCRIPTION OF THE DISCLOSURE

[0014] 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.

[0015] 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.

[0016] 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.

[0017] 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.

[0018] The terms "including", "comprising", “having” and variations thereof mean "including but not limited to", unless expressly specified otherwise. The terms “client”, “user”, and “customer” have been used interchangeably throughout the disclosure. In the present disclosure, the term “node”, “App node”, and “party”.

[0019] The present disclosure is related to client onboarding process using blockchain technology. Specifically, the present disclosure relates to techniques of onboarding clients into a decentralized blockchain network. In a non-limiting embodiment, a method that uses permissioned blockchain for onboarding the clients is proposed. The method includes creating a single secured digital print of a client which may be accessed by permissioned nodes on the blockchain network, in response to a request from a client for onboarding. Further, the method includes reusing and modifying the created single secured digital print record when the client onboards to multiple applications. The permissioned blockchain network may comprise a single source of truth and may avoid maintaining multiple copies of client at different applications.

[0020] In the present disclosure, each transaction may comprise a secure multiparty computation (sMPC) protocol. The sMPC may work by enabling different parties with their own private data to carry out a joint computation without need for revealing their private inputs to each other. As a result, the different parties may learn from combined data without sharing individual data with each other. None of the party may learn anything about the other party’s data, instead each party may learn a result of the computation and each party may take decision based on the learned result. In sMPC, data is encrypted, broken up and distributed across players (or “secretly shared”) during the computation, which makes it safe against quantum attacks. Thus, data is quantum secured in sMPC. Further, each transaction may go through a zero-knowledge proof, that is, one party may prove possession of certain information without revealing that information. As a result, the security and scalability of the transactions may be enhanced since the proof may be stored instead of the payload.

[0021] **Figure 1** shows a schematic representation 100 showing a sequence of steps involved in onboarding clients into a payment network, in accordance with some embodiments consistent with the present disclosure.

[0022] At **step 1** (S1), a user may initiate a client onboarding process using a decentralized application user interface (Dapp UI) application **105** installed in the user device (not shown in

Figure 1) or using a Dapp UI web application by accessing a web browser. The user device may include, without limitation, a smart phone, a laptop, a desktop, a computer, and the like. The user **103** may be a customer, a client, a patron, a vendor, and/or a merchant who wants to be onboarded into the payment network. In a non-limiting embodiment, the decentralized application user interface (Dapp UI) may be an application that runs on a blockchain or peer-to-peer (P2P) network of computers instead of a single computer. In other words, the Dapp UI may comprise open source applications that are not controlled by a single centralized entity. For example, the Dapp UI may include, without limitation, a web application, a mobile application, and the like. The web application is an application software which may be accessed using a web browser. In the Dapp UI, partners, merchants, partner may be onboarded and the partner to merchant relations may be created.

[0023] According to **Figure 1**, the client may initiate onboarding into a payment network, to conduct their business with multiple stakeholders. For example, the multiple stakeholders may include, without limitation, payment gateways, a card network, payment processors, Issuers, merchants, and the like. Specifically, the user may create an account, login to the Dapp UI **105** using unique login credentials, and subsequently, may enter configuration data. As an example, the configuration data may be created or generated by client themselves. Further, the Dapp UI **105** may require the user **103** to sign the transaction request using their private key to add the transaction to the blockchain or to query the blockchain. In other words, a configuration data may be forwarded to the specific app ledger in the block chain network.

[0024] At **step 2** (S2), in response to the user's initiation of onboarding and signing the transaction request, the Dapp UI **105** may send a transaction request to the blockchain network or distributed blockchain network. For instance, the configuration data of the client with transaction request may be forwarded to the specific app ledger in the blockchain network. The blockchain network may validate the signature of the client. Specifically, the transaction request from the Dapp UI **105** may be sent to an Application Programming Interface (API). As an example, the transaction request may include, without limitation, a transaction payload (Tx payload) and a recipient address. For example, the transaction payload may be an information extracted from a message or string provided to the API. The transaction pay load may include, without limitation, a configuration data, a customer or client information, an order details and the like.

[0025] At **step 3** (S3), the blockchain network may generate a symmetric key by creating the Tx payload. Further, the Tx payload may be encrypted with symmetric key using 'symmetric key algorithms' or 'secret key algorithms"(for example, a zero knowledge encryption algorithm). After encrypting the Tx payload with symmetric key, the blockchain network may hash the encrypted Tx payload and subsequently, encrypt the symmetric key with public keys of the parties. The hash may be a cryptographic by-product of a hash algorithm (for example, Ethereum using KECCAK-256 algorithm) which may take an input string of any length, for e.g., numbers, alphabets, etc., and transforms it into a fixed length. This fixed length may be varied based on the hash function which is being used. Subsequently, the API may share the encrypted symmetric key with public key to a client manager for storing. In other words, the API may share a digitally signed Tx payload to the client manager.

[0026] At **step 4** (S4), the client manager may receive digital signature or encrypted Tx payload information from the API and store the Tx payload information. The client manager may validate the received signature. Subsequently, the client manager may publish the Tx payload message in the network. Further, the encrypted Tx payload information may be sent to the Dapp UI through API.

[0027] At **step 5** (S5), the encrypted Tx payload is sent to an application network. For example, the Dapp UI **105** may send the encrypted Tx payload in the form of new block to the application network. The application network may comprise one or more application nodes in the blockchain network. The one or more application nodes in the application network may fetch new block from the network for validating private Tx (or private key). For example, the application network may comprise Application node 1, Application node 2, and Application node 3. However, the present disclosure is not limited thereto and in general, the application network may comprise 'n' number of application nodes. Each of the application node may include a smart contract and a shared ledger. The smart contract (also known as crypto contract) may be a computer program which directly and/or automatically controls the transfer of digital assets between parties. The shared ledger (also known as distributed ledger) may be used to record transaction in a ledger in multiple places at the same time.

[0028] At **step 6** (S6), each of the application node from the one or more application nodes in the application network may receive or fetch new block from network for validation and the new block may contain private Tx payload (or private key). Further, each of the application node may request Tx payload response from the client manager as shown in **step 7** (S7). The

Tx payload response may include the encrypted symmetric key and the encrypted Tx payload. Further, if the requestor (for example, any one of App node 1, App node 2, or App node 3) is known party to the client Tx, then that particular requestor (or App node) may receive the Tx response from the client manager. For example, consider all the App nodes, e.g., App node 1, App node 2, and App node 3 receive the client Tx payload from the Dapp UI **105** and each of the App node may request for Tx response data from the client manager. Consider that the App node 2 is a known party to the client Tx. Since, the App node 2 is the known party to the client Tx, the App node 2 may receive client Tx response (i.e., the encrypted symmetric key and the encrypted Tx payload). In other words, when the transaction request is provided to the blockchain, the blockchain with smart contract may run the query through Multi-Party Computation (MPC) and Zero Knowledge Proof (ZKP). Further, if the configuration data or the encrypted Tx payload matches with the proof in the blockchain, then the transaction response may be sent to that particular App node.

[0029] At **step 8** (S8), the app node that received the Tx payload may decrypt the symmetric key and decrypt the Tx payload and, subsequently, store the client configuration. For example, from the above in **step 7**, the APP node 2 may already comprise a private key. The App node 2 may use the received private key to decrypt the symmetric key and the client Tx payload. Subsequently, the App node 2 may store the client configuration information. At **step 9** (S9), in a non-limiting embodiment, the updates or information in steps 4-8, may be stored in the shared ledger of the respective application nodes. As a result, the client may be onboarded, and their digital identity may be created using zero knowledge proof and multiparty computation which may make the entire process faster compared to traditional approaches.

[0030] **Figure 2** shows an exemplary flow chart illustrating method steps for onboarding clients into the payment network using permissioned blockchain, in accordance with some embodiments of the present disclosure.

[0031] At block **202**, initially, a permissioned blockchain network may be created. Specifically, the permissioned block chain network may be created by involving components of a service as nodes or entities. The client's information or transaction data may be onboarded to the permissioned block chain network. For example, the components of the permissioned blockchain may include multiple nodes connected to each other.

[0032] At block **204**, changes in components of the permissioned blockchain network may be recorded as a single transaction and subsequently, the transactions may be collected in a transaction pool. The transaction pool, also known as mempool may be a data structure containing set of transactions which are not mined but are validated by miners. As an example, the miners may add transaction data to global public ledger of past transactions. In the permissioned blockchain network, the blocks may be secured by the miners and all the blocks are connected to each other and may form a chain structure.

[0033] After collecting the transactions in the transaction pool, set of transactions may be obtained by various application nodes from the transaction pool for mining as indicated in block **206**. For example, the mining may be a process of adding transactions into the distributed ledger by solving a complex computational incentive problem in the network. In a non-limiting embodiment, the application node that may successfully mine the transactions are allowed to add the block into the network or in the shared ledger. The addition of the block may be passed through the consensus among all the other application nodes. After the addition of the block to the network or the shared ledger, the miner may receive rewards. As an example, the rewards for miners may include, without limitation, new coins created with addition of blocks, transaction fees from all the transactions included in the block.

[0034] At block **208**, the miners may encrypt the data using the private key and may create a signature. Subsequently, the signature may be decrypted using the public key of the miner. In an embodiment, the blockchain components may read data from the ledger and may add data into the ledger. However, the data may not be modified. As a result, a solid audit trail may be created for any record changes in the components of the permissioned blockchain network. In the permissioned blockchain network, a transparent environment may be created. As a result, any node within the network may easily view any changes in the other nodes. Further, any changes in the one node may need to be reflected in the other nodes. Hence, each application node is employed with smart contract to automate data movement and availability.

[0035] **Figure 3** illustrates a distributed ledger showing details of all transactions performed while onboarding a client, in accordance with some embodiments consistent with the present disclosure.

[0036] The illustrated distributed ledger may comprise three blocks linked together. For example, block 3, block 2 and block 1. The present disclosure may include ‘n’ number of blocks

in the distributed ledger. Each block may include a Nonce value, a current hash value, a previous hash value, and transaction details. The Nonce value may be created only once while creating a new block or validating a transaction, The previous hash value may be the hash of a previous block header and point to the current hash value of the previous block. The data in the blocks cannot be changed. Further, in the distributed ledger, new blocks cannot be inserted between any two blocks. Further, if any attempt is made to change data in the blocks, all the participants in the blockchain network may be alerted immediately. Each block has transaction details. As an example, in **Figure 3**, the transaction details may provide information about creating client profile. The transaction steps for creating the client profile may include [TR0] – [TR11]. The transaction steps [TR0]-[TR11] in **Figure 3**, may depict the transactions for onboarding a client in the blockchain. All the transactions are made available in the distributed ledger and can never be modified.

[0037] For instance, the NONCE value for each of the block may be for example, 'PJXX01'. The NONCE may be a value or a number which can be used once in the transaction and may be used to verify transactions and to perform security checks. According to **Figure 3**, in block 1, the client profile data may be initiated in a transaction step '0' [TR0], and subsequently, the client may be authenticated using the public encryption key in the transaction step '1'[TR1]. Further, the client information may be validated successfully in the transaction step 2 [TR2], and initial client profile information may be updated in the transaction step 3 [TR3].

[0038] Additionally, in block 2, initial client profile information is provided with initial client id in the transaction step '4' [TR4], for example api-key, and all the client terms are updated in the transaction step '5'[TR5]. Further, the client may be initiated and issuer for the client may be mapped in the transaction step '6' [TR6]. Furthermore, the client profile information may be updated in the transaction step '7' [TR7].

[0039] Finally, in block 3, bin configuration information may be updated in the transaction step '8' [TR8], and card art information may be updated in the transaction step '9' [TR9]. Subsequently, transaction configuration may be performed in the transaction step '10' [TR10]. Finally, the client profile information may be created successfully in the transaction step '11' [TR11].

[0040] **Figure 4** illustrates an implementation of the proposed technique in an Enterprise Ethereum (permissioned) blockchain network for financial application, in accordance with

some embodiments consistent with the present disclosure. The present disclosure is not limited thereto. The present disclosure may be implemented on any of the block chain network. The Enterprise Ethereum is a permissioned blockchain or a decentralized blockchain that may run on a decentralized computer known as Ethereum Virtual Machine (EVM). In the Ethereum decentralized blockchain network, each node may hold a copy of EVM, that is, any interactions (for example, transactions) may be verified so that each node may update their copy. Further, the Enterprise Ethereum may restrict who can establish a node on and write to the ledger. In other words, the implementer has authority over which parties write to the blockchain, install approval system, and participate in the consensus process. In the deployment phase of Enterprise Ethereum, the data may be authentic. Further, use of shared distributed ledger may provide traceability and guarantee transparency of the data distribution process.

[0041] **Figure 5** illustrates a block diagram of a use case for onboarding clients in a Token Service (TS) system. The client onboarding may be initiated using the Dapp UI. The permissioned block chain may include one or more different components such as different applications (for example, an application 1, an application 2, application 3 etc.) and client manager, and all the components may be integrated or interconnected with each other in the permissioned blockchain network. Further, the Dapp UI may send the transaction request or client information to the permissioned blockchain. The components in the permissioned blockchain may store client information or client identity information. According to the present disclosure, if the one component is updated, then the update may be automatically replicated across different components using blockchain concept. Further, a IP world may orchestrate all the transaction in the permissioned block chain.

ADVANTAGES OF THE PRESENT DISCLOSURE

[0042] In the present disclosure, the process of client onboarding into blockchain is made very simple and seamless. As a result, the present disclosure may provide ability to centralize all client documentation, allow changes to be made faster and customer support representative to be informed on every step of the process.

[0043] The present disclosure may implement zero knowledge proof and multiparty computation. As a result, the permissioned blockchain may accelerate the whole process of onboarding. Consequently, the existing client and the new client may be onboarded efficiently.

[0044] The present disclosure may provide a digital distributed ledger to track the changes done to client information by the users or client. As a result, the present disclosure may ensure that changes in the blockchain network are digitally signed.

[0045] In the present disclosure, the onboarding in the permissioned blockchain may eliminate the overhead for onboarding new customers in a secure and trusted manner.

[0046] The present disclosure may take only header information using zero knowledge proof. As a result, the time consuming for onboarding process is efficiently reduced.

[0047] The present disclosure may provide Multi-Party Computation (sMPC) for each transaction where the nodes carry out joint computation without revealing their private inputs to each other. As a result, each transaction goes through zero knowledge proof. Consequently, this makes the transactions highly secure and scalable.

[0048] In the present disclosure, the immutable nature of the blockchain network makes it hard to mutilate the client information. As re result, the present disclosure may avoid maintaining multiple copies of the client at different applications.

[0049] In the present disclosure, the shared ledger or distributed ledger may provide public verifiability of its overall state without leaking information about the state individual participant.

[0050] The present disclosure may consolidate all client configurations that are currently silo’d and may easily trace their lifecycle.

[0051] The present disclosure may provide secure and decentralized blockchain platform for the client onboarding process. As a result, the present disclosure may provide high degree of security.

[0052] The proposed model may be a low cost business model and may be easily deployed on any blockchain platform.

[0053] The present disclosure may employ smart contracts. As a result, manual interventions are avoided.

[0054] Finally, the language used in the specification has been principally selected for readability and instructional purposes, and it may not have been selected to delineate or circumscribe the inventive subject matter. Accordingly, the disclosure of the embodiments of the disclosure is intended to be illustrative, but not limiting, of the scope of the disclosure.

[0055] With respect to the use of substantially any plural and/or singular terms herein, those having skill in the art can translate from the plural to the singular and/or from the singular to the plural as is appropriate to the context and/or application. The various singular/plural permutations may be expressly set forth herein for sake of clarity.

DIGITAL CLIENT IDENTITY AND MANAGEMENT USING BLOCKCHAIN

ABSTRACT

The present disclosure relates to implementing client onboarding process using a blockchain network. Specifically, the present disclosure relates to techniques of onboarding clients in a decentralized blockchain network. A method that uses permissioned blockchain is proposed for onboarding clients. The method includes creating a single secured digital print of a client by the permissioned blockchain when the client initiates onboarding process and allowing only permissioned nodes on the decentralized network to access the created digital print. Further, the method discloses using and modifying the created single digital print record when the client onboards to multiple applications. The blockchain network is immutable, so the client information cannot be mutilated. Also, the technique avoids maintaining multiple copies of the client at different applications.

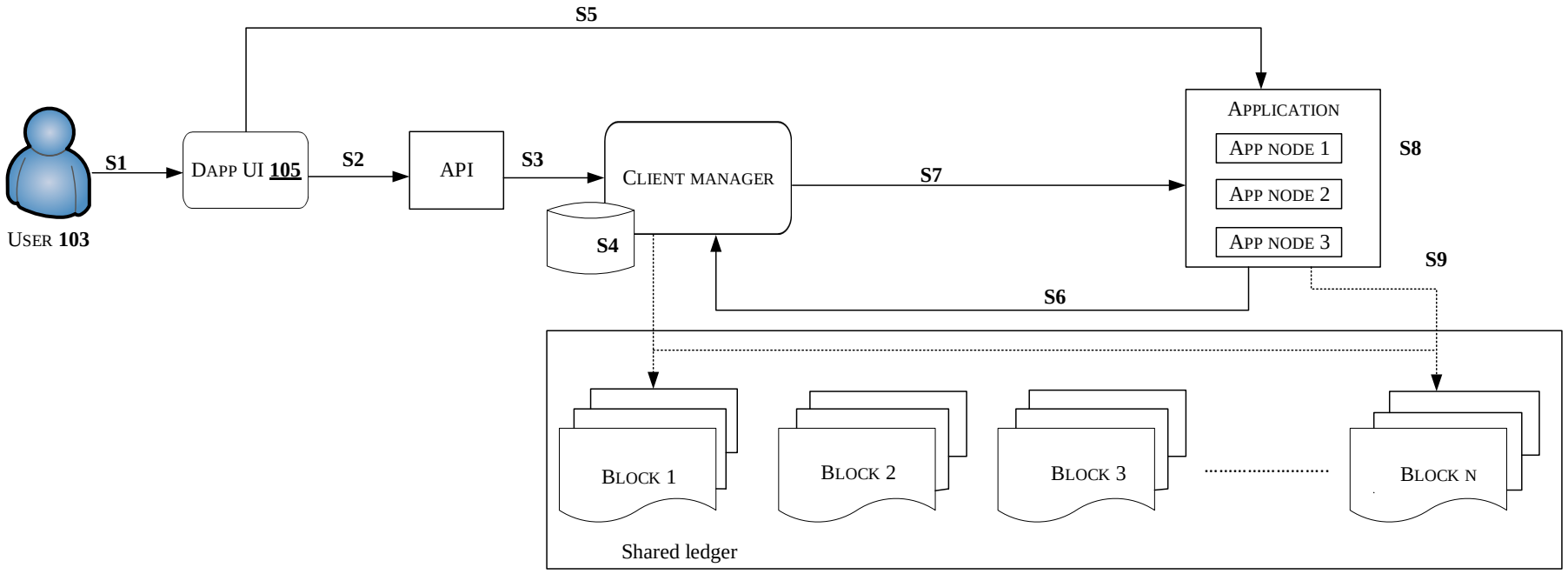
S5
USER 103
S1
DAPP UI 105
S2
API
S3
CLIENT MANAGER
S4
S7
APPLICATION
APP NODE 1
APP NODE 2
APP NODE 3
S8
S9
S6
BLOCK 1
BLOCK 2
BLOCK 3
BLOCK N
Shared ledger

Figure 1

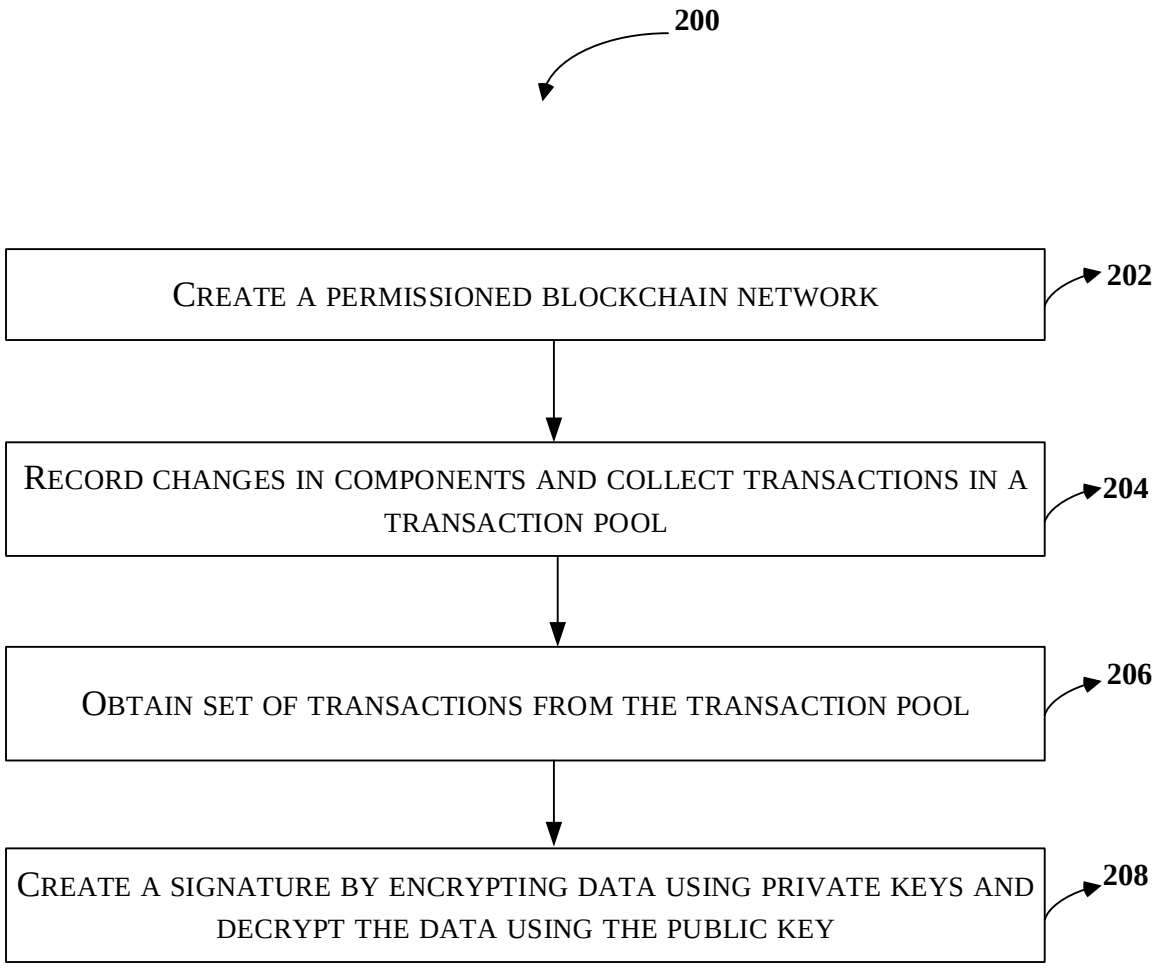
200
CREATE A PERMISSIONED BLOCKCHAIN NETWORK
202
RECORD CHANGES IN COMPONENTS AND COLLECT TRANSACTIONS IN A TRANSACTION POOL
204
OBTAIN SET OF TRANSACTIONS FROM THE TRANSACTION POOL
206
CREATE A SIGNATURE BY ENCRYPTING DATA USING PRIVATE KEYS AND DECRYPT THE DATA USING THE PUBLIC KEY
208

Figure 2

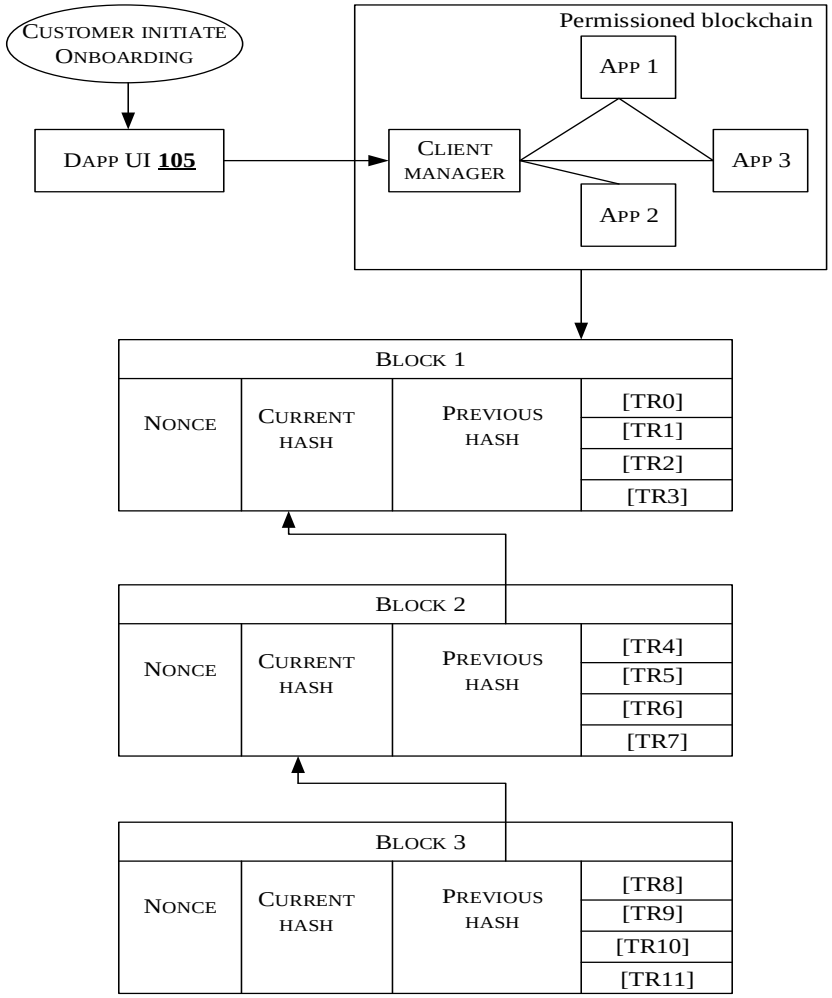
CUSTOMER INITIATE ONBOARDING
DAPP UI 105
Permissioned blockchain
APP 1
CLIENT MANAGER
APP 3
APP 2
BLOCK 1
NONCE
CURRENT HASH
PREVIOUS HASH
[TR0]
[TR1]
[TR2]
[TR3]
BLOCK 2
NONCE
CURRENT HASH
PREVIOUS HASH
[TR4]
[TR5]
[TR6]
[TR7]
BLOCK 3
NONCE
CURRENT HASH
PREVIOUS HASH
[TR8]
[TR9]
[TR10]
[TR11]

Figure 3

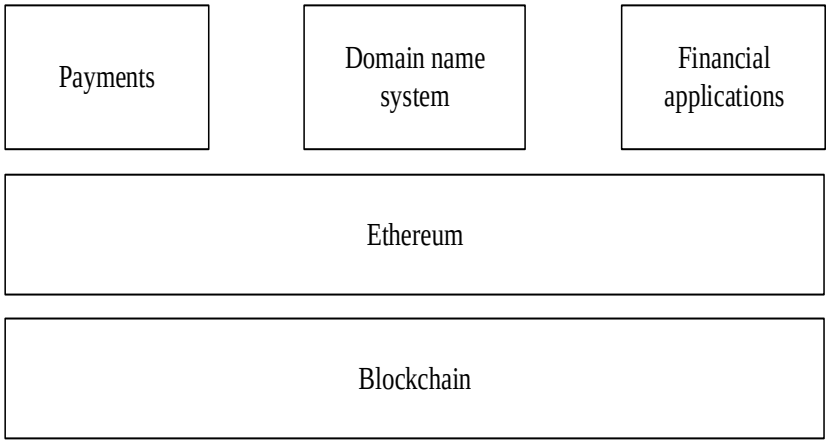
Payments
Domain name system
Financial applications
Ethereum
Blockchain

Figure 4

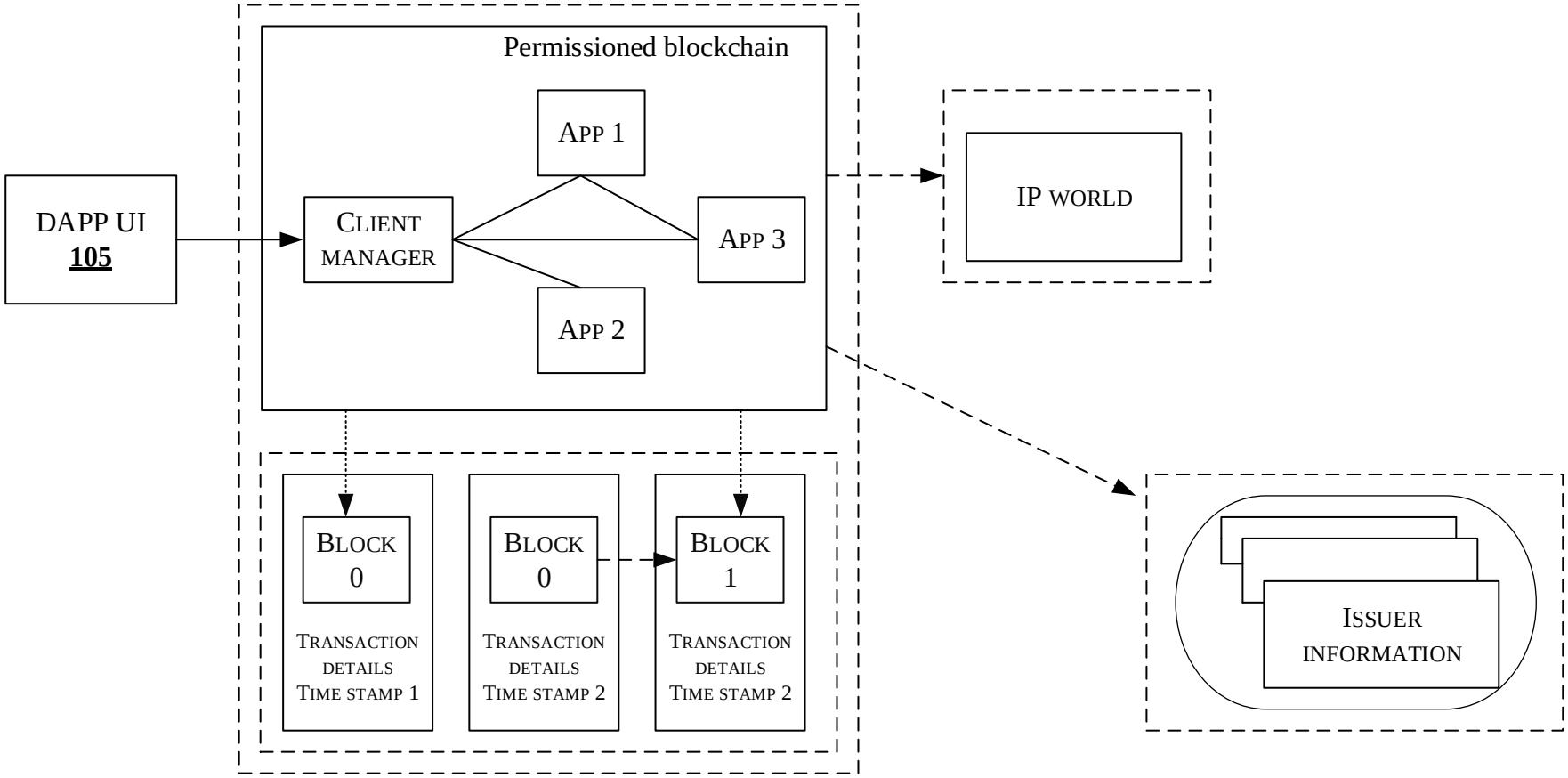
Permissioned blockchain
APP 1
DAPP UI
105
CLIENT MANAGER
APP 3
APP 2
IP WORLD
BLOCK 0
TRANSACTION DETAILS
TIME STAMP 1
BLOCK 0
TRANSACTION DETAILS
TIME STAMP 2
BLOCK 1
TRANSACTION DETAILS
TIME STAMP 2
ISSUER INFORMATION

Figure 5