

TITLE
DEPOSIT TOKENIZATION SYSTEM

RELATED APPLICATIONS

[0001] This application claims priority to Greek Patent Application No. 20230100784, titled DEPOSIT TOKENIZATION SYSTEM, filed September 29, 2023, and claims priority to U.S. Provisional Application No. 63/549,640, titled BANK TOKENS AND THEIR INTERBANK TRANSFERS, filed February 5, 2024, the disclosures of which are incorporated by reference in their entirety.

TECHNICAL FIELD

[0002] The following disclosure relates generally to a deposit tokenization (DT) system that can provide commercial banks and their clients with a platform to mint, burn, and/or transfer token deposits within a particular commercial bank or between different commercial banks.

BACKGROUND

[0003] The evolution of digital currencies has prompted a demand for tokenizing fiat currencies that run on blockchains. Types of digital currencies that have emerged include cryptocurrencies, stablecoins, central bank digital currencies (CBDCs), and tokenized bank deposits. Despite the demand for tokenizing fiat currencies, there are still numerous questions related to how such tokenization will be implemented. For example, issues may arise related to the tokenization of fiat currencies if banks or countries pursue different versions of a tokenized deposit product. Accordingly, there exists a need for devices, systems, and methods that enable interoperability between banks related to the tokenization of fiat currencies.

SUMMARY

[0004] According to one aspect, the present disclosure provides a method for preforming a token deposit transaction. The method can include receiving a request from a first client to perform a token deposit transaction for a bank asset or a bank liability. The method can further include identifying a second client in the token deposit transaction and determining an address associated with the second client. The method can further include verifying fiat reserves for the bank asset or the bank liability and minting a token deposit for the bank asset or the bank liability to a first blockchain. The method can further include initiating a reserve contract to hold a central bank digital currency to securitize the token deposit

transaction. The method can further include transferring the token deposit to the address associated with the second client.

[0005] According to another aspect, the present disclosure provides a computing system. The computing system can include a deposit tokenization service and an interoperability service. The deposit tokenization service can receive, from a first client, a request to transfer a token deposit to a second client. The token deposit can correspond to a bank asset or a bank liability. The token deposit can be recorded on a first blockchain. The deposit tokenization service can further identify a second blockchain for the second client. The first blockchain can be different from the second blockchain. The interoperability service can transfer the token deposit from the first blockchain to the second blockchain.

[0006] According to yet another aspect, the present disclosure provides a method for performing a token deposit transaction. The method can include hosting, by a first commercial bank system, a node of a first blockchain and sending, by the first commercial bank system via an application programming interface (API), a first request to a payment network system. The first request can be for minting a token deposit to the first blockchain. The token deposit can correspond to a bank asset or a bank liability. The method can further include updating, by the first commercial bank system, the first blockchain based on the payment network system minting the token deposit. The method can further include sending, by the first commercial bank system via the API, a second request to a payment network system. The second request can be for transferring the token deposit to a second commercial bank system associated with a second blockchain. The payment network system can transfer the token deposit by burning the token deposit from the first blockchain and minting the token deposit to the second blockchain. The method can further include updating, by the first commercial bank system, the first blockchain based on the payment network system burning the token deposit.

BRIEF DESCRIPTION OF THE DRAWINGS

[0007] In the description, for purposes of explanation and not limitation, specific details are set forth, such as particular aspects, procedures, techniques, etc., to provide a thorough understanding of the present technology. However, it will be apparent to one skilled in the art that the present technology may be practiced in other aspects that depart from these specific details.

[0008] The accompanying drawings, where like reference numerals refer to identical or functionally similar elements throughout the separate views, together with the detailed

description below, are incorporated in and form part of the specification, and serve to further illustrate aspects of concepts that include the claimed disclosure and explain various principles and advantages of those aspects.

[0009] The systems and methods disclosed herein have been represented where appropriate by conventional symbols in the drawings, showing only those specific details that are pertinent to understanding the various aspects of the present disclosure so as not to obscure the disclosure with details that will be readily apparent to those of ordinary skill in the art having the benefit of the description herein.

[0010] FIG. 1 illustrates a block diagram for a deposit tokenization (DT) system that provides a blockchain infrastructure, according to at least one aspect of the present disclosure.

[0011] FIG. 2 illustrates a block diagram for another DT system that provides a blockchain infrastructure and that can enable a commercial bank to test the DT system, according to at least one aspect of the present disclosure.

[0012] FIG. 3 illustrates a block diagram for another DT system that provides a blockchain infrastructure and that can enable a commercial bank and a client of the commercial bank to test the DT system, according to at least one aspect of the present disclosure.

[0013] FIG. 4 illustrates a block diagram for a DT system that provides a blockchain infrastructure for tokenized deposits of a first commercial bank and a second commercial bank, according to at least one aspect of the present disclosure.

[0014] FIG. 5 illustrates a block diagram for a DT system with a commercial bank hosting a blockchain node and a wallet, according to at least one aspect of the present disclosure.

[0015] FIG. 6 illustrates a block diagram for a DT system that provides a blockchain infrastructure for transferring tokenized deposits between a first commercial bank and a second commercial bank, according to at least one aspect of the present disclosure.

[0016] FIG. 7 illustrates a block diagram for a DT system that provides a blockchain infrastructure, according to at least one aspect of the present disclosure.

[0017] FIG. 8 illustrates a flowchart for a wrapped token deposit, according to at least one aspect of the present disclosure.

[0018] FIG. 9 illustrates a block diagram for a DT system comprising a consortium-permissioned chain, according to at least one aspect of the present disclosure.

[0019] FIG. 10 illustrates a block diagram for a DT system comprising a consortium-permissioned chain and multiple commercial banks employing different blockchains, according to at least one aspect of the present disclosure.

[0020] FIG. 11 illustrates a block diagram of a network architecture for a DT system, according to at least one aspect of the present disclosure.

[0021] FIG. 12 illustrates a communication channel flow diagram for an onboard service within the network architecture of FIG. 11, according to at least one aspect of the present disclosure.

[0022] FIG. 13 illustrates a communication channel flow diagram for an infrastructure manager within the network architecture of FIG. 11, according to at least one aspect of the present disclosure.

[0023] FIG. 14 illustrates a communication channel flow diagram for a custody service within the network architecture of FIG. 11, according to at least one aspect of the present disclosure.

[0024] FIG. 15 illustrates a communication channel flow diagram for a universal payment channel (UPC) on-chain within the network architecture of FIG. 11, according to at least one aspect of the present disclosure.

[0025] FIG. 16 illustrates a communication channel flow diagram for a universal payment channel (UPC) off-chain within the network architecture of FIG. 11, according to at least one aspect of the present disclosure.

[0026] FIG. 17 illustrates a communication channel flow diagram for a universal payment bridge (UPB) within the network architecture of FIG. 11, according to at least one aspect of the present disclosure.

[0027] FIG. 18 illustrates a communication channel flow diagram for a tokenization manager within the network architecture of FIG. 11, according to at least one aspect of the present disclosure.

[0028] FIG. 19 illustrates a process that can be implemented using a DT system to add digital tokens to a blockchain, according to at least one aspect of the present disclosure.

[0029] FIG. 20 illustrates a process that can be implemented using a DT system for an issuing bank to approve another bank to hold and use tokens of the issuing bank, according to at least one aspect of the present disclosure.

[0030] FIG. 21 illustrates a process that can be implemented using a DT system to transfer tokens from a wallet of a first bank to a wallet of a second bank, according to at least one aspect of the present disclosure.

[0031] FIG. 22 illustrates a process that can be implemented using a DT system for purchasing tokenized assets using tokenized deposits that an investor holds with a bank, according to at least one aspect of the present disclosure.

[0032] FIG. 23 illustrates a process that can be implemented using a DT system for off-ramping tokens of an issuing bank in exchange for fiat currency, according to at least one aspect of the present disclosure.

[0033] FIG. 24 illustrates a process that can be implemented using a DT system for a first bank and a second bank to transact based on tokens of an issuing bank, where the second bank may not be approved by the issuing bank to hold tokens of the issuing bank, according to at least one aspect of the present disclosure.

[0034] FIG. 25 illustrates a process that can be implemented using a DT to facilitate a cross-border transaction between a first bank and a second bank, according to at least one aspect of the present disclosure.

[0035] FIG. 26 is a block diagram of a computer apparatus with data processing subsystems or components, according to at least one aspect of the present disclosure.

[0036] FIG. 27 is a diagrammatic representation of an example system that includes a host machine within which a set of instructions to perform any one or more of the methodologies discussed herein may be executed, according to at least one aspect of the present disclosure.

[0037] FIG. 28 illustrates a logic flow diagram of a tokenized deposit being transferred between two contracts deployed on the same blockchain, according to at least one aspect of the present disclosure.

DESCRIPTION

[0038] The following disclosure may provide exemplary systems, devices, and methods for conducting a financial transaction and related activities. Although reference may be made to such financial transactions in the examples provided below, aspects are not so limited. That is, the systems, methods, and apparatuses may be utilized for any suitable purpose.

[0039] Before discussing specific embodiments, aspects, or examples, some descriptions of terms used herein are provided below.

[0040] A “central bank” may refer to an institution that issues issue public money in the form of banknotes and coins. A central bank may also issue reserves for wholesale money movement and lending between commercial banks.

[0041] A “central bank digital currency” or “CBDC” may refer to a tokenized version of the reserves issued by a central bank.

[0042] A “client” may refer to a commercial bank or a client of the commercial bank, such as an individual or a company with a deposit account at a commercial bank.

[0043] A “commercial bank” may refer to any type of depository institution, such as a bank, thrift, or credit union. A commercial bank may issue money in the form of deposit accounts (e.g., a promise to pay a client legal tender that is counted as a liability of the commercial bank). As used herein, a “bank” can refer to a commercial bank, including any type of depository institution.

[0044] A “demand deposit account” or “DDA” may refer to a deposit account issued by a commercial bank and recorded to a blockchain. A “tokenized bank deposit” or a “deposit token” can refer to money represented in a demand deposit account.

[0045] Reference to “a device,” “a server,” “a processor,” and/or the like, as used herein, may refer to a device, server, or processor that is recited as performing a previous step or function, a different server or processor, and/or a combination of servers and/or processors. For example, as used in the specification and the claims, a first server or a first processor that is recited as performing a first step or a first function may refer to the same or different server or the same or different processor recited as performing a second step or a second function.

[0046] “Distributed Ledger Technology (DLT)” may refer to a permissioned blockchain accessible only by a consortium and maintained by a consortium-approved set of validator

nodes running Byzantine Fault Tolerance (BFT) consensus. BFT is a consensus mechanism that works on resolving the Byzantine Generals' problem in a decentralized network.

[0047] A "minter" can be an entity authorized to mint new deposit tokens.

[0048] A "party" can be any entity of a system identifiable with an identity (e.g., a public key). For example, a bank, a minter, and a client each may be considered parties.

[0049] A "payment network" or "payment network system" may refer to an electronic payment system used to accept transactions, transmit transactions, process transactions, or otherwise facilitate transactions. In some aspects, the payment network may transfer information and funds among issuers, acquirers, merchants, and payment device users. One illustrative non-limiting example of a payment network is VisaNet, which is operated by Visa, Inc. In some aspects, a "payment network" or "payment network system" may refer to an electronic system for facilitating and/or managing the minting, burning, and transfer of deposit tokens, for example, between commercial banks.

[0050] As used herein, the term "system" may refer to one or more computing devices or combinations of computing devices (e.g., processors, servers, client devices, software applications, components of such, and/or the like).

[0051] A "TokenContract," a "token contract," or a "smart contract" may refer to a smart contract deployed for a commercial bank to maintain tokenized deposits of its clients.

[0052] Various commercial banks may have interest in deploying in-house deposit tokenization solutions. These solutions can enable the banks to move fiat funds globally through demand deposit accounts and newly created blockchain deposit accounts (BDAs). In some aspects, commercial banks have interest in deploying in-house deposit tokenization solutions as a response to pressure and opportunities brought by CBDC and stablecoins.

[0053] However, many of the in-house deposit tokenization solutions deployed by commercial banks have been fragmented and siloed. Therefore, a lack of congruity and interoperability can exist between various deposit tokenization solutions being deployed across different commercial banks. For example, with commercial banks deploying different tokenization solutions with different standards and different protocols, various challenges can exist related to transferring deposit tokens from one commercial bank to another.

[0054] Moreover, there can exist a significant barrier to entry for various commercial banks to develop, test, and implement deposit tokenization solutions. For example, in order for commercial banks to be able to transfer deposit tokens with other banks, the commercial

banks will likely need to develop the ability to tokenize assets and liabilities on a blockchain. However, there can exist a significant cost of investment related to developing such technologies.

[0055] The present disclosure provides various devices, systems, and methods that can enable commercial banks and/or clients of commercial banks to “mint,” “burn,” and “transfer” assets and liabilities via tokenized transactions. The tokenized transactions may be executed across client accounts of a commercial bank (e.g., intrabank transactions) and/or with client accounts of other commercial banks (e.g., interbank transactions) to transfer commercial bank assets and liabilities.

[0056] The devices, systems, and methods disclosed herein can employ a deposit tokenization (DT) system for executing tokenized transactions. The DT system can define standard rules and/or protocols for commercial banks and/or clients of commercial banks to adhere to for executing tokenized transactions. The DT system can include tools to assist commercial banks and/or clients in accessing and operating within the DT system and can further include operational roles for a service provider (e.g., a payment network system managing the DT system) to perform during the execution of the DT system. The DT system can provide a blockchain infrastructure for executing tokenized transactions.

[0057] In various aspects, the DT system can include a payment network system. The payment network system may be hosted by a payment network system server (e.g., a service provider server). The payment network system can include or otherwise execute tokenization services (e.g., via a DT sandbox) for assisting commercial banks and/or clients of the commercial banks in accessing and operating within the DT system. For example, the payment network system can include or otherwise facilitate the execution of custom smart contracts and customizable token management functions to enable commercial banks and/or clients to issue tokenized deposits. The tokenization services included in the payment network system are sometimes referred to herein as deposit tokenization (DT) services. As discussed further herein, the DT services can include a tokenization provider, a smart contract library, a proof-of-reserve (e.g., proof-of-reserve generator), and a multi-chain manager.

[0058] The tokenization provider (e.g., of the DT services) can serve as an overall management service of DT system and can enable commercial banks and/or clients of the commercial banks to deploy a smart contract, such as a smart contract generated from smart contract features stored in the smart contract library. The tokenization provider can set security parameters and permission levels on a minting contract. Furthermore, in some

aspects, the tokenization provider can attest to proof-of-reserve (e.g., fiat currency) for a token to be minted, for example, by interaction with a third party. The tokenization provider can include or otherwise access a directory service that enables commercial banks and/or clients of the commercial banks to identify other commercial banks and/or clients based on human-readable names. The directory service may be used for both interbank and intrabank transfers and may track all public addresses and other identifying information related to commercial banks and clients of commercial banks established in the DT system.

[0059] The smart contract library (e.g., of the DT services) can be utilized for deploying and/or generating smart contracts for use in the DT system. For example, by interacting with the payment network system via an application program interface (API), a commercial bank can select and deploy service provider-developed (e.g., payment network system-developed) smart contracts and features stored in the smart contract library, thereby customizing the tokenized deposits executable by the smart contracts. The smart contracts stored in the smart contract library can include configurable parameters for enabling customization. The configurability of the parameters may be controlled such that smart contracts generated therefrom comply with various regulatory requirements. The payment network system can therefore provide, via the tokenization provider and the smart contract library of the DT services, a smart contract system for defining protocols for tokenization. In various aspects, by generating smart contracts via the DT services, the payment network system can be used to establish entities for deposit tokenization, mint deposit tokens, audit fiat reserves for the underlying token, and deploy token contracts with privacy-preserving token functionality.

[0060] The proof-of-reserve (e.g., of the DT services) may be employed for minting tokenized bank deposits. For example, depending on a country's capital and liquidity requirements for commercial banks on deposit accounts, the proof-of-reserve of the DT system can provide a service to attest to the bank's reserves held on a designated ledger. In the case of a country issuing a wholesale CBDC (wCBDC), the proof-of-reserve can provide an attestation service that reviews the amount of wCBDC held by a commercial bank against the amount of tokenized deposits. In some aspects, the proof-of-reserve can also be set to "zero" to not conduct any proof-of-reserve verification for the minting process.

[0061] The multi-chain manager (e.g., of the DT services) can be employed to enable a multi-chain DT system. For example, the payment network system can be blockchain-agnostic, such that the DT system can be deployed across a selection of blockchains. Commercial banks may choose from a range of blockchains to decide where to tokenize their assets and liabilities. It is also possible that a commercial bank may tokenize on several

different blockchains. The multi-chain manager can be configured to track and manage funds across different blockchains.

[0062] The DT system can include payment services (e.g., deposit tokenization interoperability services). The payment services can form an infrastructure layer to enable transfer of deposit tokens between different commercial banks across different blockchains. In some aspects, the payment services may be hosted by the payment network system (e.g., hosted by the same server). The payment services can include a universal payment bridge (UPB) and/or a universal payment channel (UPC). For example, the UPC can be configured to create payment channels, open and/or close payment channels, manage protocols for transactions, deposit and/or withdraw funds, and/or otherwise manage payment channels between blockchains, serving as a “universal adaptor” among blockchains. The UPB can be configured as a payment bridge.

[0063] In various aspects, the payment network system of the DT system can establish commercial banks within the DT system. For example, the DT services provided by the payment network system can establish commercial banks as deposit tokenization owners and enable the commercial banks to deploy smart contracts. Via the DT services, the payment network system can mint deposit tokens though (e.g., based on instructions from) minter(s). In some aspects, the minter(s) can include a commercial bank or a consortium of bank-authorized minters. The underlying assets (e.g., fiat currency reserves) for the minted deposit tokens may be verified by a third-party auditor. For example, the third-party auditor may verify the presence of fiat reserves for the token deposits by interacting with the tokenization provider and/or the proof-of-reserve of the DT services. Based on instructions from the minter(s), the payment network system can deploy a smart contract to execute tokenization of deposits. In some aspects, the payment network system employs privacy-preserving smart contracts for the tokenization (e.g., similar to the standard defined by Ethereum Request for Comment 20 (ERC-20)) with encrypted balances and read/write policies.

[0064] In various aspects, the payment network system and/or the DT system may be configured to: (a) mint tokens based on user’s permissions, wherein the token minting is secured with multi-signature security features; (b) burn tokens based on user’s permissions, wherein the token burning is secured with multi-signature security features, and wherein the token is taken out of circulation by sending the token to an address or wallet that can only receive tokens; (c) transfer tokens that allow any user to send an encrypted amount of funds to another user; (d) execute privacy-preserving features and policies to prevent data leakage while still allowing for anti-money laundering (AML) and counter-terrorism financing (CTF)

monitoring to occur; and/or (e) deploy a reserve contract to hold reserves used to back the smart contract (TokenContract) issuance. In some aspects, the smart contract is a simplified version of a central bank digital currency (CBDC) contract for holding CBDC reserves on deposit of commercial banks. In various aspects, the smart contract may be a reserve contract that comprises parameters that can be adjusted based on a country's specific requirements, as set by a central bank.

[0065] The devices, systems, and methods disclosed herein can provide various technological benefits. For example, various DT systems disclosed herein can provide a global scheme and platform for any commercial bank to tokenize its assets and liabilities on the blockchain of its choice, while also enabling the commercial bank to transfer the assets and liabilities to a different bank that has its assets and liabilities tokenized on a different blockchain. Furthermore, various DT systems disclosed herein can provide general interoperability related to deposit tokenization. This interoperability can include token-to-token transactions (e.g., interbank transactions that require interoperability and/or settlement between blockchain networks), token-to-fiat transactions (e.g., interoperability of fiat rails and tokenized deposits), and token-to-smart-contract transactions (e.g., interaction of tokenized deposits with smart contracts).

[0066] As another example, blockchain technology can reach faster settlement speed and can reduce counterparty settlement risks through programmability. With tokenized bank deposits, corporations may save significant costs, especially in cross-border flows. Moreover, with money becoming programmable, there are new ways to pay and be paid that can be enabled through programmable smart contracts.

[0067] FIG. 1 illustrates a block diagram for a deposit tokenization (DT) system 100 that can provide a blockchain infrastructure for tokenized deposits, according to at least one aspect of the present disclosure. The DT system 100 can include a payment network system 102 and a commercial bank 140 (e.g., a commercial bank system). The commercial bank 140 can be configured to execute functions related to core banking 144, such as issuing money in the form of deposit accounts (e.g., functions for managing traditional deposit accounts in a non-tokenized form). Although the DT system 100 of FIG. 1 depicts one commercial bank 140, the DT system 100 can include a plurality of commercial banks 140 in communication with the payment network system 102.

[0068] In various aspects, the DT system 100 can be a version of a DT system that enables the commercial bank 140 to perform various testing and development related to generating tokenized deposits. For example, in one aspect, the DT system 100 can be

configured to operate as a non-production, non-value environment (e.g., for alpha-phase development). The payment network system 102 can provide an end-to-end solution that enables the commercial bank 140 to tokenize its liabilities via deposit tokens within the DT system 100.

[0069] The payment network system 102 can include a deposit token (DT) application program interface (API) 104. Further, the commercial bank 140 can include a software development kit (SDK) 142. In various aspects, the SDK 142 comprises a software package and/or software tools provided to the commercial bank 140 by the payment network system 102 (e.g., a service provider associated with the payment network system) for enabling the commercial bank 140 to perform testing and development in the DT system 100. The SDK 142 can communicate with and access services of the payment network system 102 by interacting with the DT API 104.

[0070] The payment network system 102 can further include deposit tokenization (DT) services 106. The DT services 106 can provide functionality to enable the commercial bank 140 to generate tokenized deposits. DT services 106 can include a tokenization provider 108, a smart contract library 110, a proof-of-reserve 112 (e.g., proof-of-reserve generator), and a multi-chain manager 114.

[0071] The tokenization provider 108 serves as an overall management service of DT system 100. For example, based on instructions from the commercial bank 140, the tokenization provider 108 can deploy a smart contract. The tokenization provider 108 can define security parameters and permission levels on a minting a smart contract. Furthermore, in some aspects, the tokenization provider 108 interacts with the proof-of-reserve 112 to obtain confirmation that a deposit token defined by the smart contract is backed by a reserve (e.g., fiat currency). The tokenization provider 108 can include or otherwise access a directory service of commercial banks 140 and/or clients of the commercial banks 140 established in the DT system 100. In some aspects, by accessing the directory service, the tokenization provider 108 can identify commercial banks 140 and/or clients of the commercial banks 140 based on human-readable names. The directory service may be used for both interbank and intrabank transfers and may keep track of public addresses and other identifying information related to the commercial banks 140 and clients of the commercial banks 140 established in the DT system 100.

[0072] The smart contract library 110 can be utilized for deploying and/or generating smart contracts. For example, by interacting with the payment network system 102 via the DT API 104, the commercial bank 140 can cause the tokenization provider 108 to select and

deploy smart contracts and features stored in the smart contract library 110. The smart contracts and features stored in the smart contract library 110 can include configurable parameters for enabling customization. The configurability of the parameters may be controlled by the payment network system 102 such that smart contracts generated based on the smart contract library 110 comply with various regulatory requirements.

[0073] In various aspects, the smart contract library 110 includes features to deploy smart contracts that perform various functions that enable tokenized transaction. For example, smart contracts generated based on the smart contract library 110 can (a) establish entities necessary for a tokenized deposit, (b) perform functions associated with tokenizing, and/or (c) deploy a reserve contract to hold reserves for backing the tokenized deposit.

[0074] Establishing entities necessary for a tokenized deposit can include establishing the owner of the tokenized deposit (e.g., the commercial bank deploying the smart contract), establishing the minter(s) of the tokenized deposit (e.g., the commercial bank deploying the smart contract, a consortium of bank-authorized minters), and/or establishing an auditor (e.g., a third party to attest to the validity of fiat reserves backing the tokenized deposit).

[0075] Performing functions associated with tokenizing can include minting deposit tokens based on user's permissions (e.g., secured with multi-signature security features), burning deposit tokens based on user's permissions (e.g., secured with multi-signature security features), transferring deposit tokens (e.g., allowing any user to send an encrypted amount of money to another user), and/or preventing data leakage related to the deposit tokens (e.g., via privacy preserving features and policies while still allowing for anti-money laundering and counter-terrorism financing monitoring to occur). The smart contracts generated based on the smart contract library 110 can include privacy-preserving token contracts (e.g., similar to the standard defined by Ethereum Request for Comment 20 (ERC-20)) with encrypted balances and read/write policies.

[0076] Deploying a reserve contract for holding reserves for backing the tokenized deposit can include deploying a reserve contract (e.g., simplified version of a CBDC contract) for holding CBDC reserves on deposits of the commercial bank 140. The reserve contract can include adjustable parameters based on a country's specific requirements (e.g., as set by a central bank).

[0077] The proof-of-reserve 112 (e.g., proof-of-reserve generator) can be configured to verify that the reserves associated with the commercial bank 140, held on a designated

ledger, are sufficient to back deposit tokens associated with the commercial bank 140. In the case of a country issuing a wholesale CBDC (wCBDC), the proof-of-reserve 112 can be configured to review the amount of wCBDC held by the commercial bank 140 against the amount of tokenized deposits. In some aspects, the proof-of-reserve 112 can be configured to not conduct any proof-of-reserve verification.

[0078] The multi-chain manager 114 enables the DT system 100 to be deployed across multiple blockchain systems. For example, the payment network system 102 can be blockchain-agnostic and can be configured to enable the commercial bank 140 (e.g., a plurality of different commercial banks 140) to execute tokenized deposits across multiple different blockchains. Thus, the multi-chain manager 114 can enable the commercial bank 140 to select its preferred blockchain for tokenizing liabilities. Further, the multi-chain manager 114 can enable the commercial bank 140 to tokenize liabilities on several different blockchains. The multi-chain manager 114 can be configured to track and manage funds across different blockchain payment networks.

[0079] The payment network system 102 can host a node (e.g., at least one node) of one or more than one blockchain(s) 116. The blockchain(s) 116 can provide ledger(s) to record tokenized deposits and transactions executed by the commercial banks 140. In some aspects, the payment network system 102 can host all the nodes needed to execute the blockchain(s) 116, such as, for example, in aspects where the DT system 100 is an alpha-phase version of a DT system. In some aspects, the infrastructure of the blockchain(s) 116 is a flexible blockchain node infrastructure configured for partial and/or incremental decentralization with a standard initial node setup and trusted governance authority by the payment network system 102. Hosting some or all of the nodes of the blockchain(s) 116 by the payment network system 102 can streamline onboarding and service access of the commercial bank 140 to the DT system 100. In various aspects, the blockchain(s) 116 can be configured with various privacy requirements for pseudonymity and basic confidentiality associated with tokenized deposits.

[0080] Access to any of the one or more than one blockchain(s) 116 may be private permissioned (access by the commercial bank 140, the payment network system 102, and a third-party auditor) or consortium permissioned (access by a set of commercial banks 140, the payment network system 102, and a third-party auditor). Additionally or alternatively, any of the one or more than one blockchain(s) 116 may be a public blockchain (e.g., Ethereum, Algorand).

[0081] The payment network system 102 can host or otherwise include one or more than one wallets on behalf of the commercial bank 140. The hosted wallets(s) can be configured to simulate the different types of accounts the commercial bank 140 typically employs. For example, the payment network system 102 can host a master wallet 118 and one or more than one client wallet 120 for the commercial bank 140. The master wallet 118 and one or more than one client wallet 120 may be granted different permission levels for utilizing the DT services 106. For example, some of the wallets 118, 120 (e.g., the master wallet 118) may be permissioned to mint new tokens whereas others (e.g., the client wallet 120) may not. The permission levels may be assigned based on whether the corresponding wallets 118, 120 are configured to interact with an external party (e.g., a different commercial bank 140). Deposit tokens may be assigned to (e.g., held by) the wallets 118, 120. Transactions to and from the wallets 118, 120 can be recorded to a ledger of one or more than one of the blockchain(s) 116.

[0082] The DT system can include payment services 130. The payment services can form an infrastructure layer to enable transfer of deposit tokens between different commercial banks 140 across different blockchains 116. In some aspects, the payment services 130 may be hosted by the payment network system 102 (e.g., hosted by the same server). In some aspects, the payment services 130 may be a separate system from the payment network system 102 (e.g., hosted by a different server or a different provisioned resource). In some aspects, the payment services 130 and the DT services 106 can be combined. The payment services 130 can include a universal payment bridge (UPB) 132 and a universal payment channel (UPC) 134. The payment services 130 are sometimes referred to herein as deposit token (DT) interoperability services (DT interoperability services).

[0083] The UPC 134 can be configured to create payment channels, open and/or close payment channels, manage protocols for transactions, deposit and/or withdraw funds, and/or otherwise manage payment channels (e.g., between commercial banks 140, between blockchains 116). In some aspects, the UPC 134 can provide an infrastructure layer and/or hub to create a network of blockchains 116, thereby connecting into other blockchain-based payment systems. In some aspects, the UPC 134 can be a “universal adaptor” among blockchains 116, enabling central banks, commercial banks 140, and clients of commercial banks 140 to exchange value by moving different forms of money (e.g., deposit tokens) between different blockchains.

[0084] The UPB 132 can be configured as a payment bridge. For example, the UPB 132 can be configured to generate and manage state proofs for transactions (e.g., proof of validation and proof of transmission across the blockchains 116).

[0085] In some aspects, the payment services 130 (DT interoperability services) can include an accumulator. The accumulator can be configured to ensure transaction concurrency across the blockchains 116. The accumulator may employ a Merkle tree-based accumulator scheme, including proof generation and accumulator updates.

[0086] In some aspects, the payment services 130 (DT interoperability services) can enable interbank transfer of deposit tokens according to the following scheme. A first commercial bank 140 can create deposit tokens recorded on a first blockchain 116 using the DT services 106. A second commercial bank 140 (e.g., a real commercial bank or a simulated commercial bank for testing purposes) can create deposit tokens on a second blockchain 116 using the DT services 106. The first commercial bank 140 can send one of the deposit tokens it created to a wallet 118, 120 associated with the second commercial bank 140, for example, by selecting, via the tokenization provider 108 the wallet 118, 120 address of the second commercial bank 140 using a directory service (e.g., human-readable names/identifiers of entities established on the DT system 100). The DT services 106 can interact with the payment services 130 (DT interoperability services) to transfer the deposit token from the first blockchain 116 to the second blockchain 116, into the wallet 118, 120 associated with the second commercial bank 140 by burning the deposit token of the first commercial bank 140 and minting a new deposit token for the second commercial bank 140.

[0087] FIGS. 2–7 and 9–10 illustrate DT systems 200 (FIGS. 2–3), 400 (FIG. 4), 500 (FIG. 5), 600 (FIG. 6), 700 (FIG. 7), 900 (FIG. 9), and 1000 (FIG. 10). The DT systems 200, 400, 500, 600, 700, 900, and 1000 may be similar in many respects to the DT system 100 (FIG. 1). For example, in some aspects, the DT systems 200, 400, 500, 600, 700, 900, and 1000 can represent different embodiments, implementations, and/or versions of the DT system 100. Accordingly, any of the features, functions, and/or aspects described with respect to the DT system 100 can apply to like components of the DT systems 200, 400, 500, 600, 700, 900, and 1000. For the sake of brevity, some of the features, functions, and/or aspects are not repeated below. Likewise, any of the features, functions, and/or aspects described with respect to any of the DT systems 200, 400, 500, 600, 700, 900, and 1000 can apply to like components of the DT system 100.

[0088] FIG. 2 illustrates a block diagram for a deposit tokenization (DT) system 200 that provides a blockchain infrastructure for tokenized deposits, according to at least one aspect

of the present disclosure. As noted above, the DT system 200 can be similar to the DT system 100 (FIG. 1). For example, the payment network system 202 (e.g., and components thereof) can be similar to the payment network system 102, the commercial bank 240 (e.g., and components thereof) can be similar to the commercial bank 140, and the DT interoperability services 230 can be similar to the payment services 130 (DT interoperability services) (e.g., and components thereof). Although the DT system 200 of FIG. 2 depicts one commercial bank 240, the DT system 200 can include a plurality of commercial banks 240 in communication with the payment network system 202.

[0089] In various aspects, the DT system 200 can be a version of a DT system that enables the commercial bank 240 to perform various testing and development related to generating tokenized deposits. For example, in one aspect, the DT system 200 can be configured to operate as a non-production, non-value environment (e.g., for alpha-phase development). The payment network system 202 can provide an end-to-end solution that enables the commercial bank 240 to tokenize its liabilities via deposit tokens within the DT system 200.

[0090] The commercial bank 240 can include API testing tools 246. In various aspects, the API testing tools 246 can be similar to the SDK 142 (FIG. 1). The API testing tools 246 can comprise a software package and/or software tools provided to the commercial bank 240 by the payment network system 202 (e.g., a service provider associated with the payment network system 202) for enabling the commercial bank 240 to perform testing and development in the DT system 200. This testing can include simulating smart contract generation, deposit tokenization, creating wallets 218, 220, and transferring deposit tokens between wallets 218, 220 via interbank transfers (e.g., wallets hosted by additional payment network system instances 250) and/or intrabank transfers. The API testing tools 246 can communicate with and access services of the payment network system 202 by interacting with the DT API 204. The payment network system 202 may require a username and/or password for the commercial bank 240 to access the payment network system 202 via the DT API 204.

[0091] The payment network system 202 can further include deposit tokenization (DT) services 206. The DT services 206 can provide functionality to enable the commercial bank 240 to generate and transfer tokenized deposits. DT services 206 can be similar to the DT services 106 (FIG. 1). The DT services 206 can include a tokenization provider 208 (e.g., similar to the tokenization provider 108), a smart contract library 210 (e.g., similar to the smart contract library 110), a proof-of-reserve 212 (e.g., similar to the proof-of-reserve 112), and a multi-chain manager 214 (e.g., similar to the multi-chain manager 114).

[0092] The payment network system 202 can host a node (e.g., at least one node) of one or more than one blockchain(s) 216. The blockchain(s) 216 can provide ledger(s) to record tokenized deposits and transactions executed by the commercial banks 240. In some aspects, the payment network system 202 can host all the nodes needed to execute the blockchain(s) 216, such as, for example, in aspects where the DT system 200 is an alpha-phase version of a DT system. The blockchain(s) 216 can be similar to the blockchain(s) 116 (FIG. 1).

[0093] The payment network system 202 can host or otherwise include one or more than one wallets on behalf of the commercial bank 240. The hosted wallets(s) can be configured to simulate the different types of accounts the commercial bank 240 typically employs. For example, the payment network system 202 can host a master wallet 218 and one or more than one client wallet 220 for the commercial bank 240. The wallets 218, 220 can be similar to the wallets 118, 120 (FIG. 1). In various aspects, the wallets 218, 220 can be recorded to the blockchain(s) 216.

[0094] The DT system 200 can include deposit token (DT) interoperability services 230. The DT interoperability services 230 can form an infrastructure layer to enable transfer of deposit tokens between different commercial banks 240 across different blockchains 216. In some aspects, DT interoperability services 230 may be hosted by the payment network system 202 (e.g., hosted by the same server). In some aspects, the DT interoperability services 230 may be a separate system from the payment network system 202 (e.g., hosted by a different server or a different provisioned resource). The DT interoperability services 230 can include a universal payment bridge (UPB) 232 and a universal payment channel (UPC) 234. The DT interoperability services 230 can be similar to the payment services 130 (DT interoperability services), the UPB 232 can be similar to the UPB 132, and the UPC 234 can be similar to the UPC 134.

[0095] The DT system 200 can include one or more than one additional payment network system instance 250. For example, in some aspects, the payment network system 202 can be hosted by a payment network server (e.g., a distributed network of servers). The payment network server may have a resource provisioned to the payment network system 202 and additional resources respectively provisioned to one or more than one additional payment network system instance 250. In other aspects, the payment network system 202 and the one or more than one additional payment network system instance 250 can be hosted by different servers.

[0096] Each additional payment network system instance 250 can include some or all of the components of the payment network system 202. For example, each additional payment network system instance 250 can include a DT API 204, the DT services 206, and host one or more than one node of the blockchain(s) 216. In some aspects, each additional payment network system instance 250 may be provisioned to a different commercial bank 240 or consortiums of commercial banks 240 by a service provider of the additional payment network system instances.

[0097] FIG. 3 illustrates a block diagram for another embodiment of the DT system 200, according to at least one aspect of the present disclosure. The embodiment of the DT system 200 illustrated by FIG. 3 additionally includes a client 260 of the commercial bank 240. The client 260 of the commercial bank 240 can be, for example, a device or system associated with an individual or a company with a deposit account at the commercial bank 240.

[0098] The client 260 can include API testing tools 262. In various aspects, the API testing tools 262 can be similar to the API testing tools 246. For example, the API testing tools 262 can comprise a software package and/or software tools provided to the client 260 by the payment network system 202 (e.g., a service provider associated with the payment network system 202) and/or the commercial bank 240 for enabling the client 260 to perform testing and development in the DT system 200. This testing can include, for example, creating wallets 218, 220 and transferring deposit tokens between wallets 218, 220 via interbank transfers and/or intrabank transfers. The API testing tools 262 can communicate with and access services of the payment network system 202 by interacting with the DT API 204 via the API testing tools 246. The payment network system 202 and/or the commercial bank 240 may require a username and/or password for the client 260 to access the payment network system 202 via the DT API 204.

[0099] FIG. 4 illustrates a block diagram for a DT system 400 that provides a blockchain infrastructure for tokenized deposits of a first commercial bank (commercial bank A 440a) and a second commercial bank (commercial bank B 440b), according to at least one aspect of the present disclosure. The DT system 400 can be similar to the DT system 100 (FIG. 1). For example, the DT system 400 can represent an implementation of the DT system 100 including the commercial bank A 440a and the commercial bank B 440b.

[0100] The payment network system 402 (e.g., and components thereof) can be similar to the payment network system 102 (FIG. 1). For example, the payment network system 402 can include a DT API 404 (similar to DT API 104), deposit tokenization (DT) services 406

(similar to DT services 106), and blockchain(s) 416 (similar to blockchain(s) 116). The DT services 406 can include a tokenization provider, a smart contract library, a proof-of-reserve, and a multi-chain manager. The DT services 406 can further include payment services (DT interoperability services) (e.g., similar to payment services 130). For example the DT services 406 can include a universal payment bridge (UPB) 432 (similar to UPB 132) and a universal payment channel 434 (similar to UPC 134).

[0101] The commercial bank A 440a and the commercial bank B 440b can each be similar to the commercial bank 140 (FIG. 1). For example, commercial bank A 440a and the commercial bank B 440b can each include an SDK (SDK 442a, SDK 442b, similar to SDK 142) and core banking (core banking 444a, core banking 444b, similar to core banking 144).

[0102] The commercial bank A 440a and the commercial bank B 440b can each generate smart contracts for minting new deposit tokens using the DT services 406. The smart contracts and deposit tokens for each of the commercial bank A 440a and the commercial bank B 440b can be recorded to a private permissioned chain. For example, the commercial bank A 440a can generate a smart contract 456a for minting a new token deposit 454a that is recorded to the private permissioned chain 452a. As another example, the commercial bank B 440b can generate a smart contract 456b for minting a new token deposit 454b that is recorded to the private permissioned chain 452b. Each of the private permissioned chain 452a and the private permissioned chain 452b may be included in the blockchain(s) 416, with nodes hosted by the payment network system 402. The private permissioned chain 452a and the private permissioned chain 452b can be the same blockchain or different blockchains.

[0103] In some aspects, the commercial bank A 440a and the commercial bank B 440b can each include service provider credentials (e.g., service provider credentials 446a, service provider credentials 446b) for accessing services of the payment network system 402.

[0104] In some aspects, the commercial bank A 440a and the commercial bank B 440b can each include a custody provider (e.g., custody provider 448a, custody provider 448b) for storing and/or providing keys and credentials associated with consumer accounts (e.g., client accounts consumer accounts 450a, consumer accounts 450b), such as consumer accounts with wallets for holding deposit tokens.

[0105] FIG. 5 illustrates a block diagram for a DT system 500 with a commercial bank 540 hosting at least one a blockchain node 522 and wallets 518, 520, according to at least

one aspect of the present disclosure. In various aspects, the DT system 500 can be a version of a DT system that enables the commercial bank 540 to perform various testing and development related to generating tokenized deposits. For example, in one aspect, the DT system 500 can be configured to operate as a non-production, non-value environment (e.g., for beta-phase development) for development following development performed using the DT system 100 where the commercial bank 540 hosts more components of the DT system 500.

[0106] The DT system 500 can be similar in many respects to the DT system 100. The DT system 500 includes a payment network system 502 (similar to payment network system 102), which includes a DT API 504 (similar to DT API 104) and deposit tokenization (DT) services 506 (similar to DT services 106). The DT services 506 can include a tokenization provider 508 (similar to the tokenization provider 108), a smart contract library 510 (similar to smart contract library 110), a proof-of-reserve 512 (similar to proof-of-reserve 112), and a multi-chain manager 514 (similar to multi-chain manager 114).

[0107] The DT system 500 includes a commercial bank 540 (similar to the commercial bank 140). The commercial bank can include a software development kit (SDK) 542 (similar to SDK 142) and core banking 544 (similar to core banking 144).

[0108] The payment network system 502 can host a node (e.g., at least one node) of one or more than one blockchain(s) 516. In some aspects, the payment network system 502 does not host all of the nodes of the one or more than one blockchain(s) 516, such that the blockchain(s) 516 are at least partially decentralized. Furthermore, the commercial bank 540 further includes (e.g., hosts) at least one blockchain node 522 of the one or more than one blockchain(s) 516.

[0109] The commercial bank 540 further includes a master wallet 518 and one or more than one client wallet 520. In some aspects, the master wallet 518 and one or more than one client wallet 520 can be respectively similar to the master wallet 118 and the client wallet 120, except that the master wallet 518 and one or more than one client wallet 520 are hosted by the commercial bank 540 rather than the payment network system 502.

[0110] The DT system 500 includes payment services 530 (DT interoperability services). The payment services 530 can be similar to the payment services 130 (FIG. 1) and can include a universal payment bridge (UPB) 532 (similar to the UPB 132) and a universal payment channel (UPC) 534 (similar to the UPC 134).

[0111] FIG. 6 illustrates a block diagram for a DT system 600 that provides a blockchain infrastructure for transferring tokenized deposits between of a first commercial bank (commercial bank A 640a) and a second commercial bank (commercial bank B 640b), according to at least one aspect of the present disclosure. In some aspects, the DT system 600 can be similar to the DT system 400 (FIG. 4) and the DT system 500 (FIG. 5). For example, the DT system 600 can represent an implementation of the DT system 500 including the commercial bank A 540a and the commercial bank B 540b (e.g., analogous to how the DT system 400 can represent an implementation of the DT system 100 including the commercial bank A 440a and the commercial bank B 440b).

[0112] The payment network system 602 (e.g., and components thereof) can be similar to the payment network system 502 (FIG. 5). For example, the payment network system 602 can include a DT API 604 (similar to DT API 504), deposit tokenization (DT) services 606 (similar to DT services 506), and blockchain(s) 616 (similar to blockchain(s) 516). The DT services 606 can include a tokenization provider, a smart contract library, a proof-of-reserve, and a multi-chain manager. The DT services 606 can further include payment services (DT interoperability services) (e.g., similar to payment services 530). For example the DT services 606 can include a universal payment bridge (UPB) 632 (similar to UPB 532) and a universal payment channel 634 (similar to UPC 534).

[0113] The commercial bank A 440a and the commercial bank B 440b can each be similar to the commercial bank 140 (FIG. 1). For example, commercial bank A 440a and the commercial bank B 440b can each include an SDK (SDK 442a, SDK 442b, similar to SDK 142) and core banking (core banking 444a, core banking 444b, similar to core banking 144).

[0114] The commercial bank A 640a and the commercial bank B 640b can each generate smart contracts for minting new deposit tokens using the DT services 606. The smart contracts and deposit tokens for each of the commercial bank A 640a and the commercial bank B 640b can be recorded to a private permissioned chain. For example, the commercial bank A 640a can generate a smart contract 656a for minting a new token deposit 654a that is recorded to the private permissioned chain 652a. As another example, the commercial bank B 640b can generate a smart contract 656b for minting a new token deposit 654b that is recorded to the private permissioned chain 652b. Each of the private permissioned chain 652a and the private permissioned chain 652b may be included in the blockchain(s) 616, with nodes hosted by the payment network system 602, the commercial bank A 640a, and/or the commercial bank B 640b. The private permissioned chain 652a and the private permissioned chain 652b can be the same blockchain or different blockchains.

[0115] The DT system 600 can enable token deposits to be transferred from the commercial bank A 640a to the commercial bank B 640b and vice versa. For example, to transfer from the commercial bank A 640a to the commercial bank B 640b, the commercial bank A 640a may generate a smart contract 660a for transferring the token deposit 658a to the commercial bank B 640b. With the payment network system 602 acting as an intermediary, the smart contract 660a can cause the token deposit 658a to be burned at the private permissioned chain 652a and can cause a new token deposit to be minted at the private permissioned chain 652b, holding the funds associated with the token deposit 658a in escrow until the burning and minting is completed. As another example, the smart contract 660b can cause the token deposit 658b to be transferred to the commercial bank 640a via a similar process of burning at the private permissioned chain 652b and minting at the private permissioned chain 652a.

[0116] In some aspects, the commercial bank A 640a and the commercial bank B 640b can each include service provider credentials (e.g., service provider credentials 646a, service provider credentials 646b) for accessing the payment network system 602.

[0117] In some aspects, the commercial bank A 640a and the commercial bank B 640b can each include a custody provider (e.g., custody provider 648a, custody provider 648b) for storing and/or providing keys and credentials associated with consumer accounts (e.g., client accounts, consumer accounts 650a, consumer accounts 650b), such as consumer accounts with wallets for holding deposit tokens.

[0118] FIG. 7 illustrates a block diagram for a deposit tokenization (DT) system 700 that provides a blockchain infrastructure, according to at least one aspect of the present disclosure. In various aspects, the DT system 700 can be a production environment version of a DT system for generating and transferring tokenized deposits. In various aspects, the DT system 700 can be a testing environment version of a DT system for generating and transferring tokenized deposits.

[0119] The DT system 700 can be similar to any of the DT systems described herein, such as, for example, the DT system 500 (FIG. 5). The DT system 700 includes a payment network system 702 (similar to any of the payment network systems described herein) and a commercial bank 740 (similar to any of the commercial banks described herein). The payment network system includes a DT API 704 (e.g., similar any of the DT APIs described herein), deposit tokenization (DT) services 706 (similar to any of the DT services described herein), and a blockchain(s) node infrastructure 716 hosting nodes for one or more than one blockchains. The DT services 706 includes a universal payment bridge (UPB) 732, a

universal payment channel (UPC) 734, a tokenization provider 708, a proof-of-reserve 712, a smart contract library 710, and a multi-chain manager 714, which can be similar to any of the UPBs, UPCs, tokenization providers, proof-of-reserves, smart contract libraries, and multi-chain managers described herein.

[0120] The commercial bank 740 can include an SDK 742, core banking 744, service provider credentials 746, a custody provider 748, and consumer accounts 750, which can be similar to any of the SDKs, core banking, service provider credentials, custody providers, and consumer accounts described herein. The commercial bank 740 can host at least one blockchain node 752 of the DT system 700. The commercial bank 740 can generate smart contracts for executing and/or transferring tokenized deposits, for example, as described herein with respect to any of the DT systems 100, 200, 400, 500, 600. For example, FIG. 7 illustrates the commercial bank 740 with a smart contract 756 for minting a token deposit 754, which can be recorded to a blockchain of the DT system 700 via the at least one blockchain node 752.

[0121] According to various aspects of the present disclosure, the DT systems disclosed herein may employ consortium-permissioned blockchains and/or public blockchains for deposit tokenization.

[0122] According to various aspects of the present disclosure, the DT systems disclosed herein may employ multiple different blockchains for deposit tokenization (e.g., different commercial banks using different blockchains).

[0123] In some aspects, the use of consortium-permissioned blockchains and/or public blockchains for deposit tokenization and/or the use of multiple different blockchains for deposit tokenization may be enabled via a wrapped token deposit scheme. In one aspect, according to a wrapped token deposit scheme, a commercial bank generates a deposit token that is recorded to a private permissioned blockchain that is specific to that commercial bank. Further, a smart contract associated with the deposit token can cause the deposit token to be held in escrow on the private permissioned blockchain and cause the minting of a corresponding stable coin on a consortium-permissioned blockchain and/or public blockchains (e.g., via a universal payment channel (UPC) and/or a universal payment bridge (UPB)). Thus, the wrapped token deposit scheme can enable access to and liquidity of the deposit token across multiple different blockchains.

[0124] FIG. 8 illustrates a flowchart for wrapped token deposits 800, according to at least one aspect of the present disclosure. To carry out a first one of the wrapped token

deposits 800, a first commercial bank mints a token deposit 858a (e.g., using a payment network system) to a private permissioned blockchain specific to the first commercial bank. A smart contract 860a associated with the token deposit 858a causes the token deposit 858a to be escrowed at the first commercial bank's private permissioned blockchain and also causes (e.g., via a UPC and/or a UPB) the minting of a corresponding stable coin 872a on a consortium-permissioned blockchain and/or a public blockchain.

[0125] Still referring to FIG. 8, to carry out a second one of the wrapped token deposits 800, a second commercial bank mints a token deposit 858b (e.g., using a payment network system) to a private permissioned blockchain specific to the second commercial bank. A smart contract 860b associated with the token deposit 858b causes the token deposit 858b to be escrowed at the second commercial bank's private permissioned blockchain and also causes (e.g., via a UPC and/or a UPB) the minting of a corresponding stable coin 872b on a consortium-permissioned blockchain and/or a public blockchain. Thus, the token deposit 858a and the token deposit 858b, respectively existing on different private permissioned blockchains, can be recorded to the same consortium-permissioned blockchain and/or public blockchain.

[0126] FIG. 9 illustrates a block diagram for a network architecture of a DT system 900 comprising a consortium permissioned chain, according to at least one aspect of the present disclosure. The DT system 900 can be similar to any of the DT systems described herein, such as, for example the DT system 700 (FIG. 7). The DT system 900 includes a payment network system 902 (similar to any of the payment network systems described herein), a commercial bank A 940a, and a commercial bank B 940b (each similar to any of the commercial banks described herein). The payment network system 900 includes a DT API 904 (e.g., similar to any of the DT APIs described herein), deposit tokenization (DT) services 906 (similar to any of the DT services described herein), and a blockchain(s) node infrastructure 916 hosting nodes for one or more than one blockchains. The DT services 906 includes a universal payment bridge (UPB) 932, a universal payment channel (UPC) 934, and can further include a tokenization provider, a proof-of-reserve, a smart contract library, and a multi-chain manager, which can be similar to any of the UPBs, UPCs, tokenization providers, proof-of-reserves, smart contract libraries, and multi-chain managers described herein.

[0127] The commercial bank A 940a and the commercial bank B 940b can each include an SDK (SDK 942a, SDK 942b), core banking (core banking 944a, core banking 944b), service provider credentials (service provider credentials 946a, service provider credentials 946b), a custody provider (custody provider 948a, custody provider 948b), and consumer

accounts (consumer accounts 950a, consumer accounts 950b), which can be similar to any of the SDKs, core banking, service provider credentials, custody providers, and consumer accounts described herein.

[0128] The commercial bank A 940a can host at least one blockchain node of a private permissioned chain 952a of the DT system 900. The commercial bank A 940a can generate smart contracts for executing and/or transferring tokenized deposits, for example, as described herein with respect to any of the DT systems 100, 200, 400, 500, 600, 700. For example, FIG. 9 illustrates the commercial bank A 940a with a smart contract 956a for minting a token deposit 954a, which can be recorded to the private permissioned chain 952a.

[0129] The commercial bank B 940b can host at least one blockchain node of a private permissioned chain 952b of the DT system 900. The commercial bank B 940b can generate smart contracts for executing and/or transferring tokenized deposits, for example, as described herein with respect to any of the DT systems 100, 200, 400, 500, 600, 700. For example, FIG. 9 illustrates the commercial bank B 940b with a smart contract 956b for minting a token deposit 954b, which can be recorded to the private permissioned chain 952b.

[0130] The DT system 900 further includes a consortium permissioned chain 970. Deposit tokens of both the commercial bank A 940a and the commercial bank B 940b can also be represented on the consortium permissioned chain 970. For example, using a wrapped token deposit scheme (e.g., similar to the wrapped token deposits 800 of FIG. 8), commercial bank A 940a can employ a smart contract 960a to cause a token deposit 958a to be escrowed on the private permissioned chain 952a and cause, via the UPB 932 and/or the UPC 934, a corresponding stable coin 972a to be minted to the consortium permissioned chain 970. As another example, using a wrapped token deposit scheme, commercial bank B 940b can employ a smart contract 960b to cause a token deposit 958b to be escrowed on the private permissioned chain 952b and cause, via the UPB 932 and/or the UPC 934, a corresponding stable coin 972b to be minted to the consortium permissioned chain 970. The deposit tokens of both the commercial bank A 940a and the commercial bank B 940b can be recorded as stable coins on the consortium permissioned chain 970.

[0131] In some aspects, the consortium permissioned chain 970 may also record central bank digital currencies (CBDC) 976. The CBDC 976 may act as a reserve backing deposit tokens of the commercial bank A 940a and/or the commercial bank B 940b. In some aspects, the consortium permissioned chain 970 may also record treasury bonds 974.

[0132] FIG. 10 illustrates a block diagram for a network architecture of a DT system 1000 comprising a consortium permissioned chain 970 and multiple commercial banks employing different public blockchains 1080a, 1080b, according to at least one aspect of the present disclosure. The DT system 1000 is similar to the DT system 900 (FIG. 9), with like reference numerals representing the same or similar elements.

[0133] The DT system 1000 further includes a public blockchain A 1080a and a public blockchain B 1080b. The public blockchain A 1080a and/or the public blockchain B 1080b can include blockchain systems, such as, for example, Ethereum and/or Algorand. Deposit tokens of commercial bank A 940a can be represented on public blockchain A 1080a and deposit tokens of commercial bank B 940b can be represented on public blockchain B 1080b. For example, using a wrapped token deposit scheme (e.g., similar to the wrapped token deposits 800 of FIG. 8, similar to the wrapped token deposit scheme for recording deposit tokens to the consortium permissioned chain 970), the commercial bank A 940a can employ a smart contract 960a to cause a token deposit 958a to be escrowed on the private permissioned chain 952a and cause, via the UPB 932 and/or the UPC 934, a corresponding stable coin 1072a to be minted to the public blockchain A 1080a. As another example, using a wrapped token deposit scheme, commercial bank B 940b can employ a smart contract 960b to cause a token deposit 958b to be escrowed on the private permissioned chain 952b and cause, via the UPB 932 and/or the UPC 934, a corresponding stable coin 1072b to be minted to the public blockchain B 1080b. Thus, the deposit tokens of both the commercial bank A 940a and the commercial bank B 940b, respectively, can be recorded as different stable coins on different public blockchains while also being recorded as stable coins on the same consortium permissioned chain 970.

[0134] FIG. 11 illustrates a block diagram of the network architecture 1100 for a DT system, according to at least one aspect of the present disclosure. The network architecture 1100 can be applied to any of the DT systems disclosed herein. The network architecture 1100 can include external components 1182 that access services 1106 via a gateway 1188. The network architecture 1100 further includes an underlying infrastructure 1198 that can be accessed by the services 1106 when carrying out various functions. The network architecture 1100 may be hosted by a service provider server (e.g., a payment network system server), which, in some aspects, may be implemented as a plurality of servers providing a distributed network architecture 1100. Various elements of the network architecture 1100 may be provisioned resources of a distributed network.

[0135] The external components 1182 may be used by external devices to access functions provided by the services 1106. For example, external components 1182 may be

hosted by a commercial bank system and/or by a client system. The external components can include a sandbox software development kit and/or application program interface (sandbox SDK/API) 1142, a commercial bank application 1184, and/or a block explorer 1186. The sandbox SDK/API 1142 can be similar to any of the SDKs (e.g., SDK 142, 442a, 442, 542, 642a, 642b, 742, 942a, 942b) described herein. The commercial bank application 1184 can include any type of application hosted by a commercial bank system and/or provided by a commercial bank system to a client device for receiving user instructions to perform transactions related to tokenized deposits. The block explorer 1186 may be used to retrieve information (e.g., history) related to blockchain transactions executed across the DT system.

[0136] The services 1106 can include an onboard 1190 service, a custody 1192 service, an infrastructure manager 1194, a blockchain transaction manager 1196, tokenization manager 1108, a universal payment channel (UPC) on-chain 1134a, a UPC off-chain 1134b, a universal payment bridge (UPB) 1132, and/or an accumulator 1136.

[0137] The onboard 1190 service can provide functions related to onboarding and/or establishing commercial banks and/or clients of commercial banks to the DT system. For example, the onboard 1190 service can be used to assign a username and/or password for a commercial bank to access the services 1106. The onboard 1190 service can enable a commercial bank to provision its own blockchain infrastructure and deploy it within the DT system. The onboard 1190 service can be configured to receive instructions from a commercial bank to enroll in one or more of the services 1106, such as, for example, the tokenization manager 1108, the universal payment channel (UPC) on-chain 1134a, the UPC off-chain 1134b, and/or the universal payment bridge (UPB) 1132. The onboard 1190 server can enable a commercial bank to deploy different smart contracts for various assets and applications.

[0138] The infrastructure manager 1194 can provision blockchain nodes to a commercial bank and provide governance over the commercial bank within the DT system. The infrastructure manager 1194 can deploy various smart contracts and/or contract features from a smart contract library (e.g., ERC-20 standards, non-fungible tokens (NFTs), claw back functions, freeze functions, encrypt functions, proof-of-reserve functions). The infrastructure manager 1194 can provide commercial banks access to a block explorer (e.g., block explorer 1186) for accessing the blockchain node infrastructure of the DT system. The infrastructure manager 1194 can post-process data and implement status updates.

[0139] The blockchain transaction manager 1196 can be employed to communicate with multiple blockchain providers. The blockchain transaction manager 1196 can create transactions including crafting transaction data, managing gas, managing nonces, and signing transactions. The blockchain transaction manager 1196 can send transactions to a blockchain node. The blockchain transaction manager 1196 can post-process mined transactions. The blockchain transaction manager 1196 can perform nonce assignment, thereby internally enabling maximum throughput for a single account. In some aspects, the blockchain transaction manager 1196 can be similar to the multi-chain managers disclosed herein (e.g., multi-chain manager 114).

[0140] The custody 1192 service can manage custody of private keys for internal and external clients (e.g., commercial banks). The custody 1192 service can manage custody for the payment network system (e.g., as a master key), commercial banks, and clients of the commercial bank. The custody 1192 service can provide credentials for clients of commercial banks, such as credentials for accessing the payment network system, wallet discovery credentials, and/or client-specific identifiers. The custody 1192 service can sign transactions. In various aspects, the custody 1192 service can be similar to the custody providers disclosed herein (e.g., custody provider 948a, 948b).

[0141] The tokenization manager 1108 can enable token deposit services according to a standard established by the payment network system. The tokenization manager 1108 can cause the minting and burning of deposit tokens. The tokenization manager 1108 can send and receive funds and messages. The tokenization manager 1108 can cause deposit tokens to be recorded with encryption and pseudonymity. The tokenization manager 1108 can implement claw back and account freeze functions. The tokenization manager 1108 can validate identity for consent, Know Your Customer (KYC) (e.g., obtaining information about a customer and verifying their identity), and Anti-Money Laundering (AML) purposes. The tokenization manager 1108 can be similar to any of the tokenization providers disclosed herein (e.g., tokenization provider 108).

[0142] The UPC on-chain 1134a can create payment channels, open and/or close payment channels, and deposit and/or withdraw funds across blockchains. The UPC off-chain 1134b can manage protocols for transactions across blockchains, manage UPC promises, and/or manage UPC receipts. The UPC on-chain 1134a and UPC off-chain 1134b can be similar to any of the UPCs disclosed herein (e.g., UPC 134).

[0143] The UPB 1132 can be configured as a payment bridge. For example, the UPB 1132 can be configured to generate and manage state proofs for transactions (e.g., proof of

validation and proof of transmission across the blockchains). The UPB 1132 can be similar to any of the UPBs disclosed herein (e.g., UPB 132).

[0144] The accumulator 1136 can be configured to ensure transaction concurrency across the blockchains. The accumulator 1136 may employ a Merkle tree-based accumulator scheme, including proof generation and accumulator updates.

[0145] The databases 1191 can store various information accessible by the services 1106. For example, the databases can store profiles generated by the onboard 1190 service (e.g., profile library 2004).

[0146] The blockchain nodes 1193 can represent one or more than one blockchain nodes for one or more than one blockchain networks of the DT system.

[0147] The encrypted file storage 1195 can store various encrypted information accessible by the services 1106.

[0148] FIG. 12 illustrates a communication channel flow diagram 1200 for the onboard 1190 service within the network architecture 1100 of FIG. 11, according to at least one aspect of the present disclosure. As illustrated by FIG. 12, for onboarding and/or establishing a commercial bank to the DT system, the commercial bank may communicate with the onboard 1190 service via the gateway 1188. Based on the communication, the onboard 1190 service may update and/or store various information (e.g., profile library 2004) to the database 1191. The onboard 1190 service may communicate with the infrastructure manager 1194 via an event bus 2002, which may further communicate via an event bus 2002 with the blockchain transaction manager 1196 to establish blockchain(s) and/or smart contract features for the commercial bank (e.g., blockchain library/smart contract library 2010). The onboard 1190 service may communicate with the infrastructure manager 1194 via an event bus 2002, which may further communicate via an event bus 2002 with the custody 1192 service to establish and store various keys for the commercial bank (e.g., stored in profile library 2004, encrypted file storage 1195).

[0149] FIG. 13 illustrates a communication channel flow diagram 1300 for the infrastructure manager 1194 within the network architecture 1100 of FIG. 11, according to at least one aspect of the present disclosure. As illustrated by FIG. 13, a commercial bank may communicate via the gateway 1188 with the infrastructure manager 1194, which may communicate with the blockchain transaction manager 1196 and the custody 1192 service, for performing various functions related to provisioning blockchain nodes, providing

governance, deploying smart contracts, providing block-explorer access, and post-processing data.

[0150] FIG. 14 illustrates a communication channel flow diagram 1400 for the custody 1192 service within the network architecture 1100 of FIG. 11, according to at least one aspect of the present disclosure. Communication may occur directly between the gateway 1188 and the custody 1192 service, for example, for executing various functions related to accessing various private keys and credentials, as well as other custody management functions.

[0151] FIG. 15 illustrates a communication channel flow diagram 1500 for the universal payment channel (UPC) on-chain 1134a within the network architecture 1100 of FIG. 11, according to at least one aspect of the present disclosure. As illustrated by FIG. 15, a commercial bank may communicate via the gateway 1188 with the UPC on-chain 1134a, which may communicate with the blockchain transaction manager 1196 and the custody 1192 service, for performing various functions related to channel management, creating payment channels, opening and closing payment channels, and deposit and withdrawal of funds (e.g., to provide interoperability across different blockchains).

[0152] FIG. 16 illustrates a communication channel flow diagram 1600 for the universal payment channel (UPC) off-chain 1134b within the network architecture 1100 of FIG. 11, according to at least one aspect of the present disclosure. As illustrated by FIG. 16, a commercial bank may communicate via the gateway 1188 with the UPC off-chain 1134b, which may communicate with the accumulator 1136 and the custody 1192 service via an event bus 2002, for performing various functions related to protocol transaction management, UPC promises, and UPC receipt management.

[0153] FIG. 17 illustrates a communication channel flow diagram 1700 for the universal payment bridge (UPB) 1132 within the network architecture 1100 of FIG. 11, according to at least one aspect of the present disclosure. As illustrated by FIG. 17, a commercial bank may communicate via the gateway 1188 with the UPB 1132, which may communicate with the blockchain transaction manager 1196 and the custody 1192 service via the event bus 2002, for performing various functions related to providing an on-chain payment bridge with state proof creation and management, cross-chain proof validation, and cross-chain transmission.

[0154] FIG. 18 illustrates a communication channel flow diagram 1800 for the tokenization manager 1108 within the network architecture 1100 of FIG. 11, according to at least one aspect of the present disclosure. As illustrated by FIG. 18, a commercial bank may

communicate via the gateway 1188 with tokenization manager 1108, which may communicate with the blockchain transaction manager 1196 and the custody 1192 service via the event bus 2002, for performing various functions related to tokenized deposits, including minting, burning, transferring funds, sending fund messages, encrypting aspects of the tokenized deposits, freezing accounts, performing claw backs, and various identify validation functions.

[0155] FIGS. 19-25 illustrate example processes implementable by various DT systems. Any of the aspects of the processes disclosed with respect to FIGS. 19-25 may be applied to or otherwise implemented by any of the DT systems disclosed herein. Furthermore, any aspects of the various DT systems disclosed herein may be applied to or otherwise included in any of the DT systems disclosed with respect to FIGS. 19-25.

[0156] Various aspects of the systems and processes disclosed herein (e.g., such as those disclosed with respect to FIGS. 19-25) can enable entities (e.g., banks) to transact based on tokens issued by an issuing bank. A public or private blockchain may be used to manage the tokens. The issuing bank can control the entities' access to the tokens.

[0157] The issuing bank may approve an entity (e.g., a smaller bank that does not implement its own blockchain-based tokens) to hold and transact with the tokens managed by the issuing bank's blockchain. Another entity (e.g., another smaller bank) may also be approved by the issuing bank hold and transact with the tokens managed by the issuing bank's blockchain. The tokens controlled by the issuing bank can serve as a form of settlement for transactions between the entities.

[0158] The systems and processes described herein can provide various technological improvements in the field of tokenized deposits. For example, as explained further herein, enabling the use of an issuing-bank-controlled token as a form of settlement may serve to facilitate use of bank-specific tokenized deposits across different banks.

[0159] A bank token may be a tokenized deposit (e.g., a tokenized version of a bank deposit). The tokenized deposit can be issued by a bank to its customers and can represent money that the customers have deposited with the bank. Receiving the customers' money corresponding tokenized deposits can benefit the bank similar to traditional bank deposits (e.g., the bank can loan out the money received from the customer to other customers based on a fractional reserve-based model).

[0160] A tokenized deposit may be specific to a particular bank. Thus, there may be a need to settle when a first customer of a first bank transacts with a second customer of a

second bank using a tokenized deposit of the first bank (e.g., when the first customer sends money to the second customer). Various approaches disclosed herein employ a central bank digital currency (CBDC) as a type of currency that can be used for settlement of tokenized deposit transactions between the first bank and the second bank.

[0161] However, in some cases, an alternative to CBDC-based settlement of tokenized deposit transactions may be desirable. For example, the first bank and the second bank noted above may be in a country where no CBDC exists. Various systems and methods described herein can enable settlement of tokenized deposit transactions using an issuing bank's tokens (e.g., where the issuing bank's tokens may act similar to a quasi-CBDC).

[0162] FIG. 19 illustrates a process 1980 implemented by a DT system 1900, according to at least one aspect of the present disclosure. The DT system 1900 includes a payment network system 1902, an issuing bank 1940 (e.g., an issuing bank system, a commercial bank system), and a blockchain 1916. The payment network system 1902 can include APIs 1904, tokenization services 1906, a smart contract library 1910, and a blockchain manager 1914, for example, similar to various other payment network systems disclosed herein. The process 1980 can be implemented to enable the issuing bank 1940 to add digital tokens 1952 to the blockchain 1916 (e.g., bring cash on-chain to an external network) based on fiat currency held by the issuing bank 1940. The issuing bank 1940 can create a smart token contract and interact with the token contract via the APIs 1904 of the payment network system 1902 to mint, burn, and transfer the tokens 1952.

[0163] According to the process 1980, the issuing bank 1940 transfers 1982 fiat deposits 1950 into a reserve account. The reserve account can serve to back the tokens 1952 of the issuing bank 1940 that are recorded on the blockchain 1916. Based on establishing the reserve account including the fiat deposits 1950, the issuing bank 1940 submits 1984 a mint request to the payment network system 1902. The payment network system 1902 mints 1986 the digital tokens 1952 on the blockchain 1916 (e.g., a blockchain-based ledger).

[0164] FIG. 20 illustrates a process 2080 implemented by a DT system 2000, according to at least one aspect of the present disclosure. The DT system 2000 includes a payment network system 2002, an issuing bank 2040 (e.g., a commercial bank), and a bank 2060. The process 2080 can be used by the issuing bank 2040 for approving the bank 2060 to hold and use tokens of the issuing bank 2040.

[0165] According to the process 2080, the issuing bank 2040 verifies 2082 the bank 2060. The verification of the bank 2060 by the issuing bank 2040 may include a know your

business (KYB) verification. The verification of the bank 2060 may further include generating a credential, for example, attesting to the KYB verification.

[0166] According to the process 2080, the issuing bank 2040, via the payment network system 2002, generates 2084 a token wallet for the bank 2060 and associates the credential with the wallet. The credential may be checked any time a payment is made to or from the wallet. The wallet can correspond to a designated space on a blockchain that is associated with bank 2060 and the tokens it holds with bank 2040.

[0167] According to the process 2080, the bank 2060 can then request 2086 that the payment network system 2002 convert deposits of the bank 2060 to tokens of the issuing bank 2040. Based on the request 2086, calling a token contract 2054 of the issuing bank 2040, the payment network system 2020 can atomically mint 2088a tokens of the issuing bank 2040 into the wallet of the bank 2060 and transfer 2088b the corresponding deposits of the bank 2060 to the reserve account of the issuing bank 2040. Thus, the bank 2060 may now hold tokens issued by the issuing bank 2040 in its wallet with the issuing bank 2040.

[0168] As used herein, to perform two or more actions “atomically” can mean to perform the actions as an indivisible set of operations such that either all actions occur or none of the actions occur.

[0169] FIG. 21 illustrates a process 2180 implemented by a DT system 2100, according to at least one aspect of the present disclosure. The DT system 2100 includes a payment network system 2102, an issuing bank 2140, a first bank 2160a, and a second bank 2160b. A first business 2162a is a client of the first bank 2160a and a second business 2162b is a client of the second bank 2160b. Both the first bank 2160a and the second bank 2160b are approved by the issuing bank 2140 to hold tokens of the issuing bank 2140. The issuing bank 2140 has a token contract 2154 deployed on the payment network system 2102. Both the first bank 2160a and the second bank 2160b have token contracts 2156 deployed on the payment network system 2102. The process 2180 can be used to transfer tokens from the wallet of the first bank 2160a to the wallet of the second bank 2160b based on a transaction between the first business 2162a and the second business 2162b.

[0170] According to the process 2180, the first business 2162a requests 2182 that the first bank 2160a cause money to be moved to the second business 2162b. The first bank 2160a can receive and send this request to the payment network system 2102. The payment network system 2102 validates 2184 the request by employing the token contract 2154 of the issuing bank 2140 to validate the credentials of the first bank 2160a and the second

bank 2160b. The payment network system 2102 atomically: (i) calls 2186a the token contract 2156 of the first bank 2160a to burn tokenized deposits from the wallet of the first business 2162a; (ii) transfers 2186b settlement tokens from the wallet of the first bank 2160a to the wallet of the second bank 2160b; and (iii) calls 2186c the token contract 2156 of the second bank 2160b to mint tokenized deposits into wallet of the second business 2162b. The tokenized deposits minted into wallet of the second business 2162b are backed by the settlement tokens transferred to the wallet of the second bank 2160b.

[0171] FIG. 22 illustrates a process 2280 implemented by a DT system 2200, according to at least one aspect of the present disclosure. The DT system 2200 includes a payment network system 2202, an issuing system 2242, a bank 2262, an investor 2262, and a tokenized asset issuer 2284. In one aspect, the issuing system 2242 can be an issuing bank that issues tokens usable for settlement, similar to various examples above. In another aspect, the issuing system 2242 can be a central bank that issues a CBDC for settlement. The tokenized asset issuer 2264 can issue tokens backed by various types of assets that can be different from fiat currency (e.g., securities, bonds, real estate, gold, or any other type of asset representing value). The investor 2262 is a client of the bank 2260. In some aspects, the payment network system 2202 can facilitate interactions across a first ledger corresponding to tokenized deposits corresponding to the bank 2262, a second ledger corresponding tokens issued by the issuing system 2242 (e.g., a CBDC ledger), and a third ledger corresponding to tokenized assets issued by the tokenized asset issuer 2264. The process 2280 can be used to facilitate the purchase of tokenized assets by the investor 2262 from the tokenized asset issuer 2264 using tokenized deposits that the investor 2262 holds with the bank 2260, where tokens of the issuing system 2242 (e.g., CBDC, issuing bank tokens) are used for settlement. The process 2280 can improve over processes related to traditional type of asset investments, for example by improving the liquidity of the backing assets, enabling the investor 2262 to have fractional ownership over a tokenized asset, and/or delaying the time required for settlement (delivery of asset vs. payment).

[0172] According to the process 2280, the investor 2262 requests 2282 that the bank 2260 purchase tokenized assets on behalf of the investor 2262. The bank 2262 can receive and send this request to the payment network system 2202. The payment network system 2202 atomically swaps 2284 deposit tokens of the bank 2260 with tokenized assets of the tokenized asset issuer 2264. To do this, the payment network system 2202 may validate credentials of the bank 2260 and the tokenized asset issuer 2264 using a delivery vs. payment function 2250 (e.g., a smart contract). The payment network system 2202 can further settle the transaction using tokens of the issuing system 2262 (e.g., CBDC, issuing

bank tokens). The tokenized assets are delivered 2286 to a wallet of the investor 2262. In some aspects, that payment network system 2252 can provide tokenized asset custody 2252, for example, corresponding to the wallet.

[0173] FIG. 23 illustrates a process 2380 implemented by a DT system 2300, according to at least one aspect of the present disclosure. The DT system 2300 includes a payment network system 2302, an issuing bank 2340, and a first bank 2360a and/or a second bank 2360b. Both the first bank 2360a and the second bank 2360b are approved by the issuing bank 2340 to hold tokens of the issuing bank 2340. The issuing bank 2340 has a token contract 2354 deployed on the payment network system 2302. The process 2380 can be used by the second bank 2360b to off-ramp (e.g., exchange) tokens of the issuing bank 2340 that the second bank 2360b holds, for example, for fiat currency. A similar process to the process 2380 may be implemented by the first bank 2360a to off-ramp tokens of the issuing bank 2340 that the first bank 2360a. In some aspects, the DT system 2300 may not include the first bank 2360a.

[0174] According to the process 2380, the second bank 2360b requests 2382 to the payment network system 2302 that tokens of the issuing bank 2340 held by the second bank 2360b be exchanged to fiat currency. The payment network system 2302 validates 2384 the request by applying the issuing bank token contract 2354 to verify the credentials of the second bank 2360b. The payment network system 2302 atomically burns 2386a the corresponding tokens of the issuing bank 2340 from the wallet of the second bank 2360b and causes 2386b a fund transfer from a settlement account of the issuing bank 2340 to a deposit account of the second bank 2360b.

[0175] FIG. 24 illustrates a process 2480 implemented by a DT system 2400, according to at least one aspect of the present disclosure. The DT system 2400 includes a payment network system 2402, an issuing bank 2440, a first bank 2460a, a second bank 2460b, and a central bank 2442. In some examples, the first bank 2260a is approved by the issuing bank 2240 to hold tokens of the issuing bank 2440 but the second bank 2460b is not approved by the issuing bank 2440 to hold tokens of the issuing bank 2440. The issuing bank 2440 has a token contract 2454 deployed on the payment network system 2402. The process 2480 can be used by the first bank 2460a and the second bank 2460b to transact in cases the second bank 2460b is not approved by the issuing bank 2440 to hold tokens of the issuing bank 2440.

[0176] According to the process 2480, the first bank 2460b requests 2482 to transfer funds to the second bank 2460b based on tokens issued by the issuing bank 2440. The

payment network system 2402 validates 2484 the request by applying the issuing bank token contract 2454 to verify the credentials of the first bank 2460a the second bank 2460b. The payment network system 2402 atomically burns 2486a the corresponding tokens of the issuing bank 2440 from the wallet of the first bank 2460a and causes 2486b a transfer of funds from a settlement account of the issuing bank 2440 to an account of the second bank 2460b, for example, using CBDC from the central bank 2442. In some aspects, the DT system 2400 may include real time payment system in addition to or in place of the central bank 2442, and the real time payment system may be used to transfer funds from the settlement account of the issuing bank 2440 to the account of the second bank 2460b.

[0177] FIG. 25 illustrates a process 2580 implemented by a DT system 2500, according to at least one aspect of the present disclosure. The DT system 2500 includes a payment network system 2502, a first issuing bank 2540a, a second issuing bank 2540b, a first bank 2560a, a second bank 2560b, and a central bank 2442. A first business 2562a is a client of the first bank 2560a and a second business 2562b is a client of the second bank 2560b. The first bank 2560a is approved by the first issuing bank 2540a to hold tokens of the first issuing bank 2540a. The second bank 2560b is approved by the second issuing bank 2540b to hold tokens of the second issuing bank 2540b. For example, the first issuing bank 2540a may be associated with a first fiat currency (e.g., U.S. Dollars) and the second issuing bank 2540b may be associated with a second fiat currency (e.g., Euros). The process 2580 can involve cooperation of the first issuing bank 2540a and the second issuing bank 2540b to facilitate a cross-border transaction between the first bank 2560a and the second bank 2560b.

[0178] The issuing bank 2540a and the second issuing bank 2540b have token contracts 2554 deployed on the payment network system 2502. The first bank 2560a and the second bank 2560b have token contracts 2556 deployed on the payment network system 2502.

[0179] According to the process 2580, the first business 2562a (e.g., via the first bank 2560b) requests 2582 to transfer funds (e.g., U.S. Dollars) to the second business 2562b (e.g., in Europe) based on tokens issued by the first issuing bank 2540a. The payment network system 2502 validates 2584 the request by applying the first issuing bank and second issuing bank token contracts 2454 to verify the credentials of the first bank 2560a the second bank 2560b. The payment network system 2502 atomically: (i) call 2586a the token contract 2556 of the first bank 2560a to burn tokenized deposits from the wallet of the first business 2562a; (ii) transfer 2586b first settlement tokens (e.g., U.S. Dollar tokens) from the wallet of the first bank 2560a to a wallet of the second issuing bank 2540b on a first ledger (e.g., a U.S. Dollar ledger); (iii) mint 2586c second settlement tokens (e.g., Euro tokens) into

a wallet of the second bank 2560b on a second ledger (e.g., a Euro ledger); and (iii) call 2586d the token contract 2556 of the second bank 2560b to mint tokenized deposits into wallet of the second business 2562b. The tokenized deposits minted into wallet of the second business 2562b are backed by the second settlement tokens transferred to the wallet of the second bank 2560b.

[0180] Smart contracts generated and employed within the DT systems disclosed herein may be described using the following notation. $[\text{value}]_{\text{bank}}$ is the encrypted value under a bank's key (e.g., using a probabilistic, additively homomorphic encryption technique). r is the randomness used to construct $[\text{value}]_{\text{bank}}$. C is a smart contract address.

[0181] Smart contracts generated and employed within the DT systems disclosed herein may be described using the following fields. $[\text{totalLiabilities}]_{\text{bank}}$ is the sum of all client balances encrypted under the bank's key. $[\text{totalSupply}]_{\text{bank}}$ is the number of tokens in circulation encrypted under the bank's key. Balances is a dictionary in the form $\{\text{party} : ([\text{bal}_{\text{party}}]_{\text{bank}}, [\text{Outbal}_{\text{party}}]_{\text{bank}}, [\text{Inbal}_{\text{party}}]_{\text{bank}})\}$ maintaining the token balance of each party, where party denotes the identity of the party, $\text{bal}_{\text{party}}$ denotes its balance available for spending, and $\text{Outbal}_{\text{party}}$, $\text{Inbal}_{\text{party}}$ denote the pending outgoing and incoming balances, respectively, which are in a locked state. Minters is the set of allowed minters. Expiry is a hard-coded value that denotes the maximum time (or number of blocks) after which the amounts in pending states are reverted if not confirmed by the payment network system.

[0182] Smart contracts generated and employed within the DT systems disclosed herein may be configured to execute the following functions. $\text{prepareBurn}(\text{client}, [\text{amt}]_{\text{bank}}, \sigma_{\text{bank}})$ decreases the client's balance by $[\text{amt}]_{\text{bank}}$ and increases client's outgoing balance by the same amount. $\text{prepareMint}(\text{client}, [\text{amt}]_{\text{bank}}, \sigma_{\text{bank}})$ increases the client's incoming balance by $[\text{amt}]_{\text{bank}}$. $\text{commitBurn}(\text{client}, [\text{amt}]_{\text{bank}})$ decreases the client's outgoing balance by $[\text{amt}]_{\text{bank}}$. $\text{commitMint}(\text{client}, [\text{amt}]_{\text{bank}})$ increases the client's balance by $[\text{amt}]_{\text{bank}}$ and decreases clients incoming balance by the same amount. $\text{checkExpiry}()$ is a bookkeeping function, which reverts the above actions of prepareBurn and prepareMint if not confirmed by the respective functions within expiry. $\text{transferInternal}(\text{sender}, \text{receiver}, [\text{amt}]_{\text{bank}})$ transfers tokens within the same contract. $\text{totalSupply}(\text{bank})$ returns $[\text{totalSupply}]_{\text{bank}}$. $\text{balanceOf}(\text{party})$ returns $[\text{bal}_{\text{party}}]_{\text{bank}}$.

[0183] Additionally or alternatively, smart contracts generated and employed within the DT system may be configured to execute the following functions. $\text{mint}(\text{receiver}, [\text{amt}]_{\text{bank}})$ increases receiver's balance by $[\text{amt}]_{\text{bank}}$. $\text{burn}(\text{sender}, [\text{amt}]_{\text{bank}})$ decreases sender's balance by $[\text{amt}]_{\text{bank}}$. $\text{transferFrom}(\text{sender}, \text{receiver}, \text{amt})$ moves amt tokens from sender to receiver

via one of the following functions. $\text{transferInternal}(\text{sender}, \text{receiver}, [\text{amt}]_{\text{bank}})$ transfers tokens within the same contract. $\text{transferIntraChain}(\text{sender}, \text{receiver}, C, [\text{amt1}]_{\text{senderBank}}, [\text{amt2}]_{\text{receiverBank}}, \pi_1, \pi_2)$ transfers tokens between two contracts deployed on the same chain. The foregoing function can be invoked from the receiving C bank's contract by the sending commercial bank (e.g., FIG. 28). $\text{totalSupply}(\text{bank})$ returns $[\text{totalSupply}]_{\text{bank}}$. $\text{balanceOf}(\text{party})$ returns $[\text{bal}_{\text{party}}]_{\text{bank}}$. $\text{transfer}(\text{receiver}, [\text{amt}])$ moves $[\text{amt}]$ tokens from the caller's account to the receiver account. The foregoing function can be applicable if clients are allowed to directly control their deposits. $\text{approve}(\text{party}, [\text{amt}])$ sets $[\text{amt}]$ as the allowance of party over the caller's tokens to transfer. The foregoing function may be applicable if clients are in control of their tokens. $\text{allowance}(\text{party}, \text{party})$ returns the encrypted remaining number of tokens $[\text{amt}]$ that party will be allowed to spend on behalf of party through transferFrom . This is zero by default.

[0184] Additionally or alternatively, smart contracts generated and employed within the DT system may be configured to execute the following off-chain functions.

$\text{CreateMintTx}(\text{receiver}, \text{amt})$: (a) Encrypt amt into $[\text{amt}]_{\text{bank}}$; (b) Generate a NIZK proof π proving that $[\text{amt}]_{\text{bank}} > 0$; (c) Invoke $C.\text{mint}(\text{client}, [\text{amt}]_{\text{bank}}, \pi)$. $\text{CreateBurnTx}(\text{sender}, \text{amt})$:

(a) Encrypt amt into $[\text{amt}]_{\text{bank}}$; (b) Generate a NIZK proof π which proves that $\text{balanceOf}(\text{sender}) - [\text{amt}]_{\text{bank}} > 0$; (c) Invoke $C.\text{mint}(\text{client}, [\text{amt}]_{\text{bank}}, \pi)$.

$\text{CreateInternalTx}(\text{sender}, \text{receiver}, \text{amt})$: (a) Encrypt sender and receiver into $[\text{amt}]_{\text{bank}}$ and $[\text{amt}]_{\text{bank}}$ respectively; (b) Generate NIZK proofs, π_1, π_2 which prove that $\text{balanceOf}(\text{sender}) - [\text{amt}]_{\text{bank}} > 0$ and $[\text{amt}]_{\text{bank}} > 0$ respectively; (c) Invoke $C.\text{transferInternal}(\text{sender}, \text{receiver}, [\text{amt}]_{\text{bank}}, \pi_1, \pi_2)$.

$\text{sendCrossChainTXinit}(\text{sender}, \text{receiver}, \text{bank}_{\text{to}}, \text{amt}, \text{chainID})$: Invoked by sending Bank to send an outbound request to payment network system: (a) Encrypt amt into $[\text{amt}]_{\text{bank}}$; (b) Send $\text{Req} = (\text{sender}, \text{receiver}, \text{bank}_{\text{to}}, [\text{amt}]_{\text{bank_to}}, \text{chainID})$ and $\sigma_{\text{bank_from}}(\text{Req})$ to payment network system. $\text{sendCrossChainTXinitfwd}(\text{Req}, \sigma)$: Invoked by payment network system on receipt of transaction request from sending Bank $\text{bank}_{\text{from}}$: (a) Verify σ ; (b) Send $(\text{Req}, \text{bank}_{\text{from}}, \sigma)$ to bank_{to} .

$\text{rcvCrossChainTXinit}(\text{Req}, \text{bank}_{\text{from}}, \sigma)$: Invoked by bank_{to} on receipt of transaction request from the payment network system: (a) Parse Req into $(\text{sender}, \text{receiver}, \text{bank}_{\text{to}}, [\text{amt}]_{\text{bank_to}}, \text{chainID})$; (b) Decrypt $[\text{amt}]_{\text{bank_to}}$ to amt; (c) If σ verification or $[\text{amt}]_{\text{bank}}$ decryption fails or if receiver belongs to a blacklist, send $((\text{Req}, \text{NACK}), \sigma_{\text{bank_to}}((\text{Req}, \text{NACK})))$ to payment network system, else, send $(\text{ACK}, \sigma_{\text{bank_to}}, ((\text{Req}, \text{ACK})))$ to payment network system.

$\text{sendCrossChainTX}(\text{Req}, \sigma)$: (a) Generate NIZK π_1 for amt $\in \text{Req}$; (b) Invoke $\text{tx}_{\text{cross}} = C.\text{SendCrossChain}(\text{Req}, \pi, \sigma)$.

[0185] In various aspects, a smart contract may be executed according to the following functions, where an encrypted amount may be accompanied by range proofs to prevent overflow attacks:

1. mint(receiver, [amt]_{bank}, π):
 - (a) Require caller = bank or caller $\in$ minters;
 - (b) Verify π to ensure that [amt]_{bank} > 0;
 - (c) Add [amt]_{bank} to balanceOf(receiver);
 - (d) Add [amt]_{bank} to [totalSupply]_{bank};
2. burn(sender, [amt]_{bank}, π):
 - (a) Require caller = bank;
 - (b) Verify π to ensure that balanceOf(sender) $\geq$ [amt]_{bank};
 - (c) Subtract [amt]_{bank} from balanceOf(sender);
 - (d) Subtract [amt]_{bank} from [totalSupply]_{bank};
3. transferInternal(sender, receiver, [amt]_{bank}, π):
 - (a) Require caller = bank;
 - (b) Verify π to ensure that [amt]_{bank} > 0 and balanceOf(sender) $\geq$ [amt]_{bank};
 - (c) Subtract [amt]_{bank} from balanceOf(sender);
 - (d) Add [amt]_{bank} to balanceOf(receiver);
4. transferIntraChain(sender, receiver, C, [amt1]_{senderBank}, [amt2]_{receiverBank}, π_1 , π_2):
 - (a) Require caller = senderBank and C.owner = senderBank;
 - (b) Verify π_1 to ensure that amt1 = amt2 > 0;
 - (c) Invoke C.burn(sender, [amt1]_{senderBank}, π_2);
 - (d) Add [amt1]_{receiverBank} to balanceOf(receiver);
 - (e) Add [amt1]_{receiverBank} to [totalSupply]_{bank};
5. addMinter(party):
 - (a) Require caller to be bank;
 - (b) Add party to minters.
6. removeMinter(party):
 - (a) Require caller to be bank;
 - (b) Remove party from minters.

[0186] In various aspects, a contract can be invoked by the commercial bank; therefore, there may be no need for checking the status of the accounts for minting, burning, and transfers, and all of these functions can be performed offline. However, in situations where a contract is invoked by authorized third parties, a compliance contract $C_{\text{compliance}}$, which is controlled by the bank, will have control over the accounts. The contract may use the following verification check (e.g., for minting): $C_{\text{compliance}}.\text{checkAccount}(\text{receiver}, \text{"Mint"})$ is True.

[0187] Any of the deposit tokenization (DT) systems, and components thereof (e.g., payment network systems, commercial bank systems) may be deployed and executed by one or more servers or computer apparatuses, such as, for example, those shown in FIGS. 19–20 and described in the accompanying text.

[0188] FIG. 26 is a block diagram of a computer apparatus 3000 with data processing subsystems or components, according to at least one aspect of the present disclosure. The subsystems shown in FIG. 26 are interconnected via a system bus 3010. Additional subsystems such as a printer 3018, keyboard 3026, fixed disk 3028 (or other memory comprising computer-readable media), monitor 3022 (which is coupled to display adapter 3020), and others are shown. Peripherals and input/output (I/O) devices, which couple to an I/O controller 3012 (which can be a processor or other suitable controller), can be connected to the computer system by any number of means known in the art, such as a serial port 3024. For example, the serial port 3024 or external interface 3030 can be used to connect the computer apparatus to a wide-area network (WAN), such as the Internet, a mouse input device, or a scanner. The interconnection via system bus allows the central processor 3016 to communicate with each subsystem and to control the execution of instructions from system memory 3014 or the fixed disk 3028, as well as the exchange of information between subsystems. The system memory 3014 and/or the fixed disk 3028 may embody a computer-readable medium.

[0189] FIG. 27 is a diagrammatic representation of an example system 4000 that includes a host machine 4002 within which a set of instructions to perform any one or more of the methodologies discussed herein may be executed, according to at least one aspect of the present disclosure. In various aspects, the host machine 4002 operates as a stand-alone device or may be connected (e.g., networked) to other machines. In a networked deployment, the host machine 4002 may operate in the capacity of a server or a client machine in a server-client network environment, or as a peer machine in a peer-to-peer (or distributed) network environment. The host machine 4002 may be a computer or computing device; a personal computer (PC); a tablet PC; a set-top box (STB); a personal digital assistant (PDA); a cellular telephone; a portable music player (e.g., a portable hard drive audio device, such as an Moving Picture Experts Group Audio Layer 3 (MP3) player); a web appliance; a network router, switch, or bridge; or any machine capable of executing a set of instructions (sequential or otherwise) that specify actions to be taken by that machine. Further, while only a single machine is illustrated, the term “machine” shall also be taken to include any collection of machines that individually or jointly execute a set (or multiple sets) of instructions to perform any one or more of the methodologies discussed herein.

[0190] The example system 4000 includes the host machine 4002, running a host operating system (OS) 4004 on a processor or multiple processor(s)/processor core(s) 4006 (e.g., a central processing unit (CPU), a graphics processing unit (GPU), or both), and various memory nodes 4008. The host OS 4004 may include a hypervisor 4010 that is able to control the functions and/or communicate with a virtual machine (VM) 4012 running on machine-readable media. The VM 4012 also may include a virtual CPU or vCPU 4014. The memory nodes 4008 may be linked or pinned to virtual memory nodes or vNodes 4016. When the memory node 4008 is linked or pinned to a corresponding vNode 4016, then data may be mapped directly from the memory nodes 4008 to their corresponding vNodes 4016.

[0191] All the various components shown in host machine 4002 may be connected with and to each other, or communicate to each other via a bus (not shown) or via other coupling or communication channels or mechanisms. The host machine 4002 may further include a video display, audio device, or other peripherals 4018 (e.g., a liquid-crystal display (LCD), alphanumeric input device(s) including, e.g., a keyboard, a cursor control device, e.g., a mouse, a voice recognition or biometric verification unit, an external drive, a signal generation device, e.g., a speaker), a persistent storage device 4020 (also referred to as a disk drive unit), and a network interface device 4022. The host machine 4002 may further include a data encryption module (not shown) to encrypt data. The components provided in the host machine 4002 are those typically found in computer systems that may be suitable for use with aspects of the present disclosure and are intended to represent a broad category of such computer components that are known in the art. Thus, the example system 4000 can be a server, minicomputer, mainframe computer, or any other computer system. The computer may also include different bus configurations, networked platforms, multi-processor platforms, and the like. Various operating systems may be used, including UNIX, LINUX, WINDOWS, QNX ANDROID, IOS, CHROME, TIZEN, and other suitable operating systems.

[0192] The disk drive unit 4024 also may be a Solid-state Drive (SSD), a hard disk drive, (HDD) or other that includes a computer- or machine-readable medium on which is stored one or more sets of instructions and data structures (e.g., data/instructions 4026) embodying or utilizing any one or more of the methodologies or functions described herein. The data/instructions 4026 also may reside, completely or at least partially, within the main memory node 4008 and/or within the processor(s)/processor core(s) 4006 during execution thereof by the host machine 4002. The data/instructions 4026 may further be transmitted or received over a network 4028 via the network interface device 4022 utilizing any one of several well-known transfer protocols (e.g., Hyper Text Transfer Protocol (HTTP)).

[0193] The processor(s)/processor core(s) 4006 and memory nodes 4008 also may comprise machine-readable media. The term “computer-readable medium” or “machine-readable medium” should be taken to include a single medium or multiple medium (e.g., a centralized or distributed database and/or associated caches and servers) that store the one or more sets of instructions. The term “computer-readable medium” shall also be taken to include any medium that is capable of storing, encoding, or carrying a set of instructions for execution by the host machine 4002 and that causes the host machine 4002 to perform any one or more of the methodologies of the present application, or that is capable of storing, encoding, or carrying data structures utilized by or associated with such a set of instructions. The term “computer-readable medium” shall accordingly be taken to include, but not be limited to, solid-state memories, optical and magnetic media, and carrier wave signals. Such media may also include, without limitation, hard disks, floppy disks, flash memory cards, digital video discs, random access memory (RAM), read-only memory (ROM), and the like. The example aspects described herein may be implemented in an operating environment comprising software installed on a computer, in hardware, or in a combination of software and hardware.

[0194] One skilled in the art will recognize that Internet service may be configured to provide Internet access to one or more computing devices that are coupled to the Internet service and that the computing devices may include one or more processors, buses, memory devices, display devices, input/output devices, and the like. Furthermore, those skilled in the art may appreciate that the Internet service may be coupled to one or more databases, repositories, servers, and the like, which may be utilized to implement any of the various aspects of the disclosure as described herein.

[0195] The computer program instructions also may be loaded onto a computer, a server, other programmable data processing apparatus, or other devices to cause a series of operational steps to be performed on the computer, other programmable apparatus, or other devices to produce a computer-implemented process such that the instructions that execute on the computer or other programmable apparatus provide processes for implementing the functions/acts specified in the flowchart and/or block diagram block or blocks.

[0196] Suitable networks may include or interface with any one or more of, for instance, a local intranet; a PAN (Personal Area Network); a LAN (Local Area Network); a WAN (Wide Area Network); a MAN (Metropolitan Area Network); a virtual private network (VPN); a storage area network (SAN); a frame relay connection; an Advanced Intelligent Network (AIN) connection; a synchronous optical network (SONET) connection; a digital T1, T3, E1, or E3 line; a Digital Data Service (DDS) connection; a dDSL (Digital Subscriber Line)

connection; an Ethernet connection; an ISDN (Integrated Services Digital Network) line; a dial-up port, such as a V.90, V.34, or V.34bis analog modem connection; a cable modem; an ATM (Asynchronous Transfer Mode) connection; or an FDDI (Fiber Distributed Data Interface) or CDDI (Copper Distributed Data Interface) connection. Furthermore, communications may also include links to any of a variety of wireless networks, including WAP (Wireless Application Protocol), GPRS (General Packet Radio Service), GSM (Global System for Mobile Communication), CDMA (Code Division Multiple Access) or TDMA (Time Division Multiple Access), cellular phone networks, GPS (Global Positioning System), CDPD (cellular digital packet data), RIM (Research in Motion, Limited) duplex paging network, Bluetooth radio, or an Institute of Electrical and Electronics Engineers (IEEE) 802.11-based radio frequency network. The network can further include or interface with any one or more of an RS-232 serial connection, an IEEE-1394 (Firewire) connection, a Fibre Channel connection, an Infrared Data Association port, a SCSI (Small Computer Systems Interface) connection, a USB (Universal Serial Bus) connection, or other wired or wireless, digital or analog interface or connection, mesh, or Digi® networking.

[0197] In general, a cloud-based computing environment is a resource that typically combines the computational power of a large grouping of processors (such as within web servers) and/or that combines the storage capacity of a large grouping of computer memories or storage devices. Systems that provide cloud-based resources may be utilized exclusively by their owners, or such systems may be accessible to outside users who deploy applications within the computing infrastructure to obtain the benefit of large computational or storage resources.

[0198] The cloud is formed, for example, by a network of web servers that comprise a plurality of computing devices, such as the host machine 4002, with each server 4030 (or at least a plurality thereof) providing processor and/or storage resources. These servers manage workloads provided by multiple users (e.g., cloud resource customers or other users). Typically, each user places workload demands upon the cloud that vary in real time, sometimes dramatically. The nature and extent of these variations typically depends on the type of business associated with the user.

[0199] It is noteworthy that any hardware platform suitable for performing the processing described herein is suitable for use with the technology. The terms “computer-readable storage medium” and “computer-readable storage media” as used herein refer to any medium or media that participate in providing instructions to a CPU for execution. Such media can take many forms, including, but not limited to, non-volatile media, volatile media, and transmission media. Non-volatile media include, for example, optical or magnetic disks,

such as a fixed disk. Volatile media include dynamic memory, such as system RAM. Transmission media include coaxial cables, copper wire, and fiber optics, among others, including the wires that comprise one aspect of a bus. Transmission media can also take the form of acoustic or light waves, such as those generated during radio frequency and infrared (IR) data communications. Common forms of computer-readable media include, for example, a flexible disk, a hard disk, magnetic tape, any other magnetic medium, a CD-ROM disk, DVD, any other optical medium, any other physical medium with patterns of marks or holes, a RAM, a PROM, an EPROM, an EEPROM, a FLASH EPROM, any other memory chip or data exchange adapter, a carrier wave, or any other medium from which a computer can read.

[0200] Various forms of computer-readable media may be involved in carrying one or more sequences of one or more instructions to a CPU for execution. A bus carries the data to system RAM, from which a CPU retrieves and executes the instructions. The instructions received by system RAM can optionally be stored on a fixed disk either before or after execution by a CPU.

[0201] Computer program code for carrying out operations for aspects of the present technology may be written in any combination of one or more programming languages, including an object-oriented programming language such as Java, Smalltalk, C++, or the like and conventional procedural programming languages, such as the “C” programming language, Go, Python, or other programming languages, including assembly languages. The program code may execute entirely on the user’s computer, partly on the user’s computer, as a stand-alone software package, partly on the user’s computer and partly on a remote computer, or entirely on the remote computer or server. In the latter scenario, the remote computer may be connected to the user’s computer through any type of network, including a local area network (LAN) or a wide area network (WAN), or the connection may be made to an external computer (for example, through the Internet using an Internet Service Provider).

[0202] In various aspects, the commercial banks may use the same public parameters for the DT system, which include the core smart contract specifications, a specified additively homomorphic encryption scheme, and a non-interactive zero-knowledge (NIZK) proof system.

[0203] FIG. 28 illustrates a logic flow diagram of a method 2800 for transferring a tokenized deposit between two contracts deployed on the same blockchain, according to at least one aspect of the present disclosure. The tokenized deposit may be based on a first smart contract 2860a (TokenContract 1). According to the method, a sender 2840a (bank 1)

initiates 2802 an intra-chain transfer (e.g., `transferIntraChain(x)`) to a receiver 2840b (bank 2) according to a second smart contract 2860b (TokenContract 2). Based on initiating 2802 the intra-chain transfer, the tokenized deposit owned by the sender 2840a is burned 2804 (`burn(x)`) and a new tokenized deposit that will be owned by the receiver 2840b is minted 2806 (`mint(x)`).

[0204] “Account credentials” may include any information that identifies an account and allows a payment processor to verify that a device, person, or entity has permission to access the account. For example, account credentials may include an account identifier, a token (e.g., account identifier substitute), an expiration date, a cryptogram, a verification value, personal information associated with an account (e.g., address), an account alias, or any combination thereof. Account credentials may be static or dynamic, such that they change over time.

[0205] An “application,” “application programming interface,” or “API” may refer to computer code or other data stored 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.

[0206] “Authentication” is a process by which the credential of an endpoint (including, but not limited to, applications, people, devices, process, and systems) can be verified to ensure that the endpoint is who they are declared to be.

[0207] As used herein, the term “comprising” is not intended to be limiting, but may be a transitional term synonymous with “including,” “containing,” or “characterized by.” The term “comprising” may thereby be inclusive or open-ended and does not exclude additional, unrecited elements or method steps when used in a claim. For instance, in describing a method, “comprising” indicates that the claim is open-ended and allows for additional steps. In describing a device, “comprising” may mean that a named element(s) may be essential for an embodiment or aspect, but other elements may be added and still form a construct within the scope of a claim. In contrast, the transitional phrase “consisting of” excludes any element, step, or ingredient not specified in a claim. This is consistent with the use of the term throughout the specification.

[0208] A “credential” may be any suitable information that serves as reliable evidence of worth, ownership, identity, or authority. A credential may be a string of numbers, letters, or any other suitable characters that may be present or contained in any object or document that can serve as confirmation. Examples of credentials include value credentials,

identification cards, certified documents, access cards, passcodes, other login information, etc.

[0209] As used herein, an “electronic wallet,” “digital wallet,” or “mobile wallet” can store user profile information, payment information (including tokens), bank account information, and/or the like and can be used in a variety of transactions, including, but not limited to, e-commerce, social networks, money transfer/personal payments, mobile commerce, proximity payments, and/or the like for retail purchases, digital goods purchases, utility payments, purchasing games, transferring funds between clients, and/or the like.

[0210] A “key” may refer to a piece of information that is used in a cryptographic algorithm to transform input data into another representation. An exemplary encryption key may include a master derivation key (MDK) that may be used to generate a limited use key (LUK) that is provided to a computer device of a user. An LUK can be an encryption key that is intended for limited use (e.g., a limited number of transactions or a limited time period) and is not intended to be used for the lifetime of an account. The MDK may be used to generate and provision the token, as well as authenticate the token when used in authorization processing by validating static and variable transaction data.

[0211] Examples of the method according to various aspects of the present disclosure are provided below in the following numbered clauses. An aspect of the method may include any one or more than one, and any combination of, the numbered clauses described below.

[0212] Clause 1. A method for performing a token deposit transaction, comprising: receiving, by a computing system, a request from a first client to perform a token deposit transaction for a bank asset or a bank liability; identifying, by the computing system, a second client in the token deposit transaction; determining, by the computing system, an address associated with the second client; verifying, by the computing system, fiat reserves for the bank asset or the bank liability; minting, by the computing system, a token deposit for the bank asset or the bank liability to a first blockchain; initiating, by the computing system, a reserve contract to hold a central bank digital currency to securitize the token deposit transaction; and transferring, by the computing system, the token deposit to the address associated with the second client.

[0213] Clause 2. The method of Clause 1, further comprising burning, by the computing system, the token deposit based on a permission level of the first client or the second client.

[0214] Clause 3. The method of Clauses 1-2, wherein the first client and second client are commercial banks.

[0215] Clause 4. The method of Clauses 1-3, wherein the minting of the token deposit is based on a permission level of the first client or the second client and an encryption key of the first client or the second client.

[0216] Clause 5. The method of Clauses 1-4, wherein the address associated with the second client corresponds to a second blockchain, and wherein the second blockchain is different from the first blockchain.

[0217] Clause 6. The method of Clauses 1-5, wherein the token deposit is a wrapped token deposit that is transferrable over a universal payment channel or a universal payment bridge.

[0218] Clause 7. A computing system, comprising: a deposit tokenization service to: receive a request from a first client to transfer a token deposit to a second client, wherein the token deposit corresponds to a bank asset or liability, and wherein the token deposit is recorded on a first blockchain; and identify a second blockchain for the second client, wherein the first blockchain is different from the second blockchain; and an interoperability service to: transfer the token deposit from the first blockchain to the second blockchain.

[0219] Clause 8. The computing system of Clause 7, wherein interoperability service is to burn the token deposit from the first blockchain and mint the token deposit to the second blockchain to transfer the token deposit from the first blockchain to the second blockchain.

[0220] Clause 9. The computing system of any of Clauses 7-8, wherein the first client and the second client are commercial banks.

[0221] Clause 10. The computing system of any of Clauses 7-9, wherein the first blockchain and the second blockchain are public blockchains.

[0222] Clause 11. The computing system of any of Clauses 7-9, wherein the first blockchain and the second blockchain are private-permissioned blockchains, and wherein the computing system hosts at least one node of the first blockchain and at least one node of the second blockchain.

[0223] Clause 12. The computing system of any of Clauses 7-11, wherein the interoperability service comprises at least one of a universal payment channel and a universal payment bridge for transferring the token deposit from the first blockchain to the second blockchain.

[0224] Clause 13. The computing system of any of Clauses 7-12, wherein the deposit tokenization service is further to verify fiat reserves for the token deposit.

[0225] Clause 14. The computing system of any of Clauses 7-13, further comprising an application programming interface (API) for communicating with the first client.

[0226] Clause 15. The computing system of any of Clauses 7-14, further comprising a smart contract library, wherein the smart contract library comprises smart contract features selectable by the first client for generating a smart contract.

[0227] Clause 16. The computing system of Clause 15, wherein the deposit tokenization service is further to generate an escrow contract based on the smart contract library, wherein the escrow contract is for recording a stable coin corresponding to the token deposit on a third blockchain.

[0228] Clause 17. The computing system of any of Clauses 7-16, wherein the third blockchain is different from the first blockchain and the second blockchain, and wherein the third blockchain is a consortium-permissioned blockchain.

[0229] Clause 18. The computing system of any of Clauses 7-16, wherein the third blockchain is different from the first blockchain and the second blockchain, and wherein the third blockchain is a public blockchain.

[0230] Clause 19. A method for performing a token deposit transaction, the method comprising: hosting, by a first commercial bank system, a node of a first blockchain; sending, by the first commercial bank system via an application programming interface (API), a first request to a payment network system, wherein the first request is for minting a token deposit to the first blockchain, wherein the token deposit corresponds to a bank asset or a bank liability; updating, by the first commercial bank system, the first blockchain based on the payment network system minting the token deposit; sending, by the first commercial bank system via the API, a second request to a payment network system, wherein the second request is for transferring the token deposit to a second commercial bank system associated with a second blockchain, wherein the payment network system transfers the token deposit by burning the token deposit from the first blockchain and minting the token deposit to the second blockchain; and updating, by the first commercial bank system, the first blockchain based on the payment network system burning the token deposit.

[0231] Clause 20. The method of Clause 19, wherein the first blockchain is a private-permissioned blockchain.

[0232] The foregoing detailed description has set forth various forms of the systems and/or processes via the use of block diagrams, flowcharts, and/or examples. Insofar as such block diagrams, flowcharts, and/or examples contain one or more functions and/or operations, it will be understood by those within the art that each function and/or operation within such block diagrams, flowcharts, and/or examples can be implemented, individually and/or collectively, by a wide range of hardware, software, firmware, or virtually any combination thereof. Those skilled in the art will recognize that some aspects of the forms disclosed herein, in whole or in part, can be equivalently implemented in integrated circuits, as one or more computer programs running on one or more computers (e.g., as one or more programs running on one or more computer systems), as one or more programs running on one or more processors (e.g., as one or more programs running on one or more microprocessors), as firmware, or as virtually any combination thereof, and that designing the circuitry and/or writing the code for the software and or firmware would be well within the skill of one of skill in the art in light of this disclosure. In addition, those skilled in the art will appreciate that the mechanisms of the subject matter described herein are capable of being distributed as one or more program products in a variety of forms, and that an illustrative form of the subject matter described herein applies regardless of the particular type of signal bearing medium used to actually carry out the distribution.

[0233] Instructions used to program logic to perform various disclosed aspects can be stored within a memory in the system, such as dynamic RAM (DRAM), cache, flash memory, or other storage. Furthermore, the instructions can be distributed via a network or by way of other computer readable media. Thus a machine-readable medium may include any mechanism for storing or transmitting information in a form readable by a machine (e.g., a computer), but is not limited to, floppy diskettes, optical disks, compact disc, read-only memory (CD-ROMs), and magneto-optical disks, read-only memory (ROMs), random access memory (RAM), erasable programmable read-only memory (EPROM), electrically erasable programmable read-only memory (EEPROM), magnetic or optical cards, flash memory, or a tangible, machine-readable storage used in the transmission of information over the Internet via electrical, optical, acoustical or other forms of propagated signals (e.g., carrier waves, infrared signals, digital signals, etc.). Accordingly, the non-transitory computer-readable medium includes any type of tangible machine-readable medium suitable for storing or transmitting electronic instructions or information in a form readable by a machine (e.g., a computer).

[0234] Any of the software components or functions described in this application, may be implemented as software code to be executed by a processor using any suitable

computer language such as, for example, Python, Java, C++ or Perl using, for example, conventional or object-oriented techniques. The software code may be stored as a series of instructions, or commands on a computer readable medium, such as RAM, ROM, a magnetic medium such as a hard-drive or a floppy disk, or an optical medium such as a CD-ROM. Any such computer readable medium may reside on or within a single computational apparatus, and may be present on or within different computational apparatuses within a system or network.

[0235] As used in any aspect herein, the term “logic” may refer to an app, software, firmware and/or circuitry configured to perform any of the aforementioned operations. Software may be embodied as a software package, code, instructions, instruction sets and/or data recorded on non-transitory computer readable storage medium. Firmware may be embodied as code, instructions or instruction sets and/or data that are hard-coded (e.g., nonvolatile) in memory devices.

[0236] As used in any aspect herein, the terms “component,” “system,” “module” and the like can refer to a computer-related entity, either hardware, a combination of hardware and software, software, or software in execution.

[0237] As used in any aspect herein, an “algorithm” refers to a self-consistent sequence of steps leading to a desired result, where a “step” refers to a manipulation of physical quantities and/or logic states which may, though need not necessarily, take the form of electrical or magnetic signals capable of being stored, transferred, combined, compared, and otherwise manipulated. It is common usage to refer to these signals as bits, values, elements, symbols, characters, terms, numbers, or the like. These and similar terms may be associated with the appropriate physical quantities and are merely convenient labels applied to these quantities and/or states.

[0238] A network may include a packet switched network. The communication devices may be capable of communicating with each other using a selected packet switched network communications protocol. One example communications protocol may include an Ethernet communications protocol which may be capable of permitting communication using a Transmission Control Protocol/Internet Protocol (TCP/IP). The Ethernet protocol may comply or be compatible with the Ethernet standard published by the Institute of Electrical and Electronics Engineers (IEEE) titled “IEEE 802.3 Standard”, published in December, 2008 and/or later versions of this standard. Alternatively or additionally, the communication devices may be capable of communicating with each other using an X.25 communications protocol. The X.25 communications protocol may comply or be compatible with a standard

promulgated by the International Telecommunication Union-Telecommunication Standardization Sector (ITU-T). Alternatively or additionally, the communication devices may be capable of communicating with each other using a frame relay communications protocol. The frame relay communications protocol may comply or be compatible with a standard promulgated by Consultative Committee for International Telegraph and Telephone (CCITT) and/or the American National Standards Institute (ANSI). Alternatively or additionally, the transceivers may be capable of communicating with each other using an Asynchronous Transfer Mode (ATM) communications protocol. The ATM communications protocol may comply or be compatible with an ATM standard published by the ATM Forum titled "ATM-MPLS Network Interworking 2.0" published August 2001, and/or later versions of this standard. Of course, different and/or after-developed connection-oriented network communication protocols are equally contemplated herein.

[0239] Unless specifically stated otherwise as apparent from the foregoing disclosure, it is appreciated that, throughout the present disclosure, discussions using terms such as "processing," "computing," "calculating," "determining," "displaying," or the like, refer to the action and processes of a computer system, or similar electronic computing device, that manipulates and transforms data represented as physical (electronic) quantities within the computer system's registers and memories into other data similarly represented as physical quantities within the computer system memories or registers or other such information storage, transmission or display devices.

[0240] One or more components may be referred to herein as "configured to," "configurable to," "operable/operative to," "adapted/adaptable," "able to," "conformable/conformed to," etc. Those skilled in the art will recognize that "configured to" can generally encompass active-state components and/or inactive-state components and/or standby-state components, unless context requires otherwise.

[0241] Those skilled in the art will recognize that, in general, terms used herein, and especially in the appended claims (e.g., bodies of the appended claims) are generally intended as "open" terms (e.g., the term "including" should be interpreted as "including but not limited to," the term "having" should be interpreted as "having at least," the term "includes" should be interpreted as "includes but is not limited to," etc.). It will be further understood by those within the art that if a specific number of an introduced claim recitation is intended, such an intent will be explicitly recited in the claim, and in the absence of such recitation no such intent is present. For example, as an aid to understanding, the following appended claims may contain usage of the introductory phrases "at least one" and "one or more" to introduce claim recitations. However, the use of such phrases should not be

construed to imply that the introduction of a claim recitation by the indefinite articles “a” or “an” limits any particular claim containing such introduced claim recitation to claims containing only one such recitation, even when the same claim includes the introductory phrases “one or more” or “at least one” and indefinite articles such as “a” or “an” (e.g., “a” and/or “an” should typically be interpreted to mean “at least one” or “one or more”); the same holds true for the use of definite articles used to introduce claim recitations.

[0242] In addition, even if a specific number of an introduced claim recitation is explicitly recited, those skilled in the art will recognize that such recitation should typically be interpreted to mean at least the recited number (e.g., the bare recitation of “two recitations,” without other modifiers, typically means at least two recitations, or two or more recitations). Furthermore, in those instances where a convention analogous to “at least one of A, B, and C, etc.” is used, in general such a construction is intended in the sense one having skill in the art would understand the convention (e.g., “a system having at least one of A, B, and C” would include but not be limited to systems that have A alone, B alone, C alone, A and B together, A and C together, B and C together, and/or A, B, and C together, etc.). In those instances where a convention analogous to “at least one of A, B, or C, etc.” is used, in general such a construction is intended in the sense one having skill in the art would understand the convention (e.g., “a system having at least one of A, B, or C” would include but not be limited to systems that have A alone, B alone, C alone, A and B together, A and C together, B and C together, and/or A, B, and C together, etc.). It will be further understood by those within the art that typically a disjunctive word and/or phrase presenting two or more alternative terms, whether in the description, claims, or drawings, should be understood to contemplate the possibilities of including one of the terms, either of the terms, or both terms unless context dictates otherwise. For example, the phrase “A or B” will be typically understood to include the possibilities of “A” or “B” or “A and B.”

[0243] With respect to the appended claims, those skilled in the art will appreciate that recited operations therein may generally be performed in any order. Also, although various operational flow diagrams are presented in a sequence(s), it should be understood that the various operations may be performed in other orders than those which are illustrated, or may be performed concurrently. Examples of such alternate orderings may include overlapping, interleaved, interrupted, reordered, incremental, preparatory, supplemental, simultaneous, reverse, or other variant orderings, unless context dictates otherwise. Furthermore, terms like “responsive to,” “related to,” or other past-tense adjectives are generally not intended to exclude such variants, unless context dictates otherwise.

[0244] It is worthy to note that any reference to “one aspect,” “an aspect,” “an exemplification,” “one exemplification,” and the like means that a particular feature, structure, or characteristic described in connection with the aspect is included in at least one aspect. Thus, appearances of the phrases “in one aspect,” “in an aspect,” “in an exemplification,” and “in one exemplification” in various places throughout the specification are not necessarily all referring to the same aspect. Furthermore, the particular features, structures or characteristics may be combined in any suitable manner in one or more aspects.

[0245] As used herein, the singular form of “a,” “an,” and “the” include the plural references unless the context clearly dictates otherwise.

[0246] Any patent application, patent, non-patent publication, or other disclosure material referred to in this specification and/or listed in any Application Data Sheet is incorporated by reference herein, to the extent that the incorporated materials is not inconsistent herewith. As such, and to the extent necessary, the disclosure as explicitly set forth herein supersedes any conflicting material incorporated herein by reference. Any material, or portion thereof, that is said to be incorporated by reference herein, but which conflicts with existing definitions, statements, or other disclosure material set forth herein will only be incorporated to the extent that no conflict arises between that incorporated material and the existing disclosure material. None is admitted to be prior art.

[0247] In summary, numerous benefits have been described which result from employing the concepts described herein. The foregoing description of the one or more forms has been presented for purposes of illustration and description. It is not intended to be exhaustive or limiting to the precise form disclosed. Modifications or variations are possible in light of the above teachings. The one or more forms were chosen and described in order to illustrate principles and practical application to thereby enable one of ordinary skill in the art to utilize the various forms and with various modifications as are suited to the particular use contemplated. It is intended that the claims submitted herewith define the overall scope.

CLAIMS

What is claimed is:

1. A method for performing a token deposit transaction, comprising:
receiving, by a computing system, a request from a first client to perform a token deposit transaction for a bank asset or a bank liability;
identifying, by the computing system, a second client in the token deposit transaction;
determining, by the computing system, an address associated with the second client;
verifying, by the computing system, fiat reserves for the bank asset or the bank liability;
minting, by the computing system, a token deposit for the bank asset or the bank liability to a first blockchain; and
transferring, by the computing system, the token deposit to the address associated with the second client.
2. The method of claim 1, further comprising initiating, by the computing system, a reserve contract to hold a central bank digital currency to securitize the token deposit transaction;
3. The method of claim 1, further comprising burning, by the computing system, the token deposit based on a permission level of the first client or the second client.
4. The method of claim 1, wherein transferring, by the computing system, the token deposit to the address associated with the second client comprises transferring a settlement token of an issuing bank from the first client to the second client, and wherein the first client and the second client are commercial banks.
5. The method of claim 1, wherein the minting of the token deposit is based on a permission level of the first client or the second client and an encryption key of the first client or the second client.
6. The method of claim 1, wherein the address associated with the second client corresponds to a second blockchain, and wherein the second blockchain is different from the first blockchain.

7. The method of claim 1, wherein the token deposit is a wrapped token deposit that is transferrable over a universal payment channel or a universal payment bridge.
8. A computing system, comprising:
a deposit tokenization service to:
receive, from a first client, a request to transfer a token deposit to a second client, wherein the token deposit corresponds to a bank asset or a bank liability, and wherein the token deposit is recorded on a first blockchain; and
identify a second blockchain for the second client, wherein the first blockchain is different from the second blockchain; and
an interoperability service to:
transfer the token deposit from the first blockchain to the second blockchain.
9. The computing system of claim 8, wherein the interoperability service is to burn the token deposit from the first blockchain and mint the token deposit to the second blockchain to transfer the token deposit from the first blockchain to the second blockchain.
10. The computing system of claim 8, wherein the first client and the second client are commercial banks, and wherein the interoperability service is to transfer a settlement token of an issuing bank from first blockchain to the second blockchain.
11. The computing system of claim 8, wherein the first blockchain and the second blockchain are public blockchains.
12. The computing system of claim 8, wherein the first blockchain and the second blockchain are private-permissioned blockchains, and wherein the computing system hosts at least one node of the first blockchain and at least one node of the second blockchain.
13. The computing system of claim 8, wherein the interoperability service comprises at least one of a universal payment channel and a universal payment bridge for transferring the token deposit from the first blockchain to the second blockchain.
14. The computing system of claim 8, wherein the deposit tokenization service is further to verify fiat reserves for the token deposit.
15. The computing system of claim 8, further comprising an application programming interface (API) for communicating with the first client.

16. The computing system of claim 8, further comprising a smart contract library, wherein the smart contract library comprises smart contract features selectable by the first client for generating a smart contract.

17. The computing system of claim 16, wherein the deposit tokenization service is further to generate an escrow contract based on the smart contract library, wherein the escrow contract is for recording a stable coin corresponding to the token deposit on a third blockchain.

18. The computing system of claim 17, wherein the third blockchain is different from the first blockchain and the second blockchain, and wherein the third blockchain is a consortium-permissioned blockchain.

19. The computing system of claim 17, wherein the third blockchain is different from the first blockchain and the second blockchain, and wherein the third blockchain is a public blockchain.

20. A method for performing a token deposit transaction, the method comprising:
hosting, by a first commercial bank system, a node of a first blockchain;
sending, by the first commercial bank system via an application programming interface (API), a first request to a payment network system, wherein the first request is for minting a token deposit to the first blockchain, wherein the token deposit corresponds to a bank asset or a bank liability;
updating, by the first commercial bank system, the first blockchain based on the payment network system minting the token deposit;
sending, by the first commercial bank system via the API, a second request to a payment network system, wherein the second request is for transferring the token deposit to a second commercial bank system associated with a second blockchain, wherein the payment network system transfers the token deposit by burning the token deposit from the first blockchain and minting the token deposit to the second blockchain; and
updating, by the first commercial bank system, the first blockchain based on the payment network system burning the token deposit.