Design and Implementation of Blockchain Based Central Bank Digital Currency

- Master Thesis Defense -

Sajan Maharjan

Supervisor: Prof. James Won-Ki Hong

Dept. of CSE, DPNM Lab., POSTECH, Korea
thesajan@postech.ac.kr

2020. 06. 30

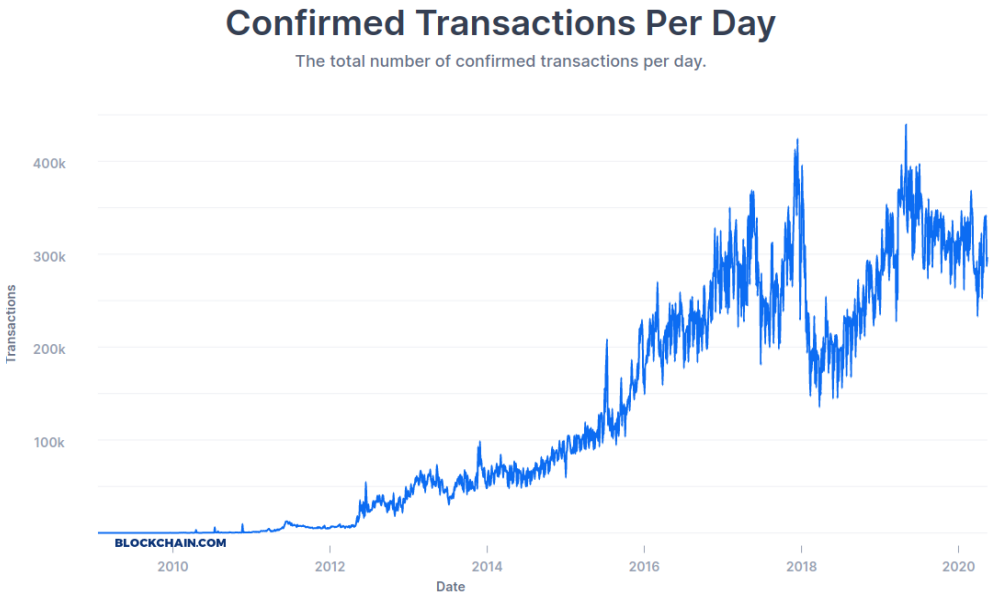
Table of Contents

- Introduction
- Background & Related Work
- Requirements & Design
- Implementation
- Evaluation
- Conclusion

Introduction

Introduction

- **Rise of Blockchain and Cryptocurrencies**
  - Gradual increase in the price of Bitcoin, Ethereum and other cryptocurrencies
  - Increasing number of transactions using cryptocurrencies
  - Concern of banks & governments over regulation & valuation of cryptocurrencies



https://www.blockchain.com/charts/n-transactions

https://etherscan.io/chart/tx

Introduction

- **Inherent Problems of Fiat Cash**
  - Expensive to produce and circulate
    - Bank of Korea reports it costs 20KRW to produce coins worth 10KRW[1]
    - 20M KRW worth of bills is subject to damage and 60B KRW is invested in re-issuance annually[1]
    - The US Federal Reserve spends $827M for printing bills and engraving coins[2]
  - Difficulty in Cash Movement
  - Lack of Transparency
  - Storage and Safety Requirements
- **Domination of private PSPs and e-money**

[1] https://www.blockmedia.co.kr/archives/114942
[2] https://www.federalreserve.gov/foia/files/2020currency.pdf

Introduction

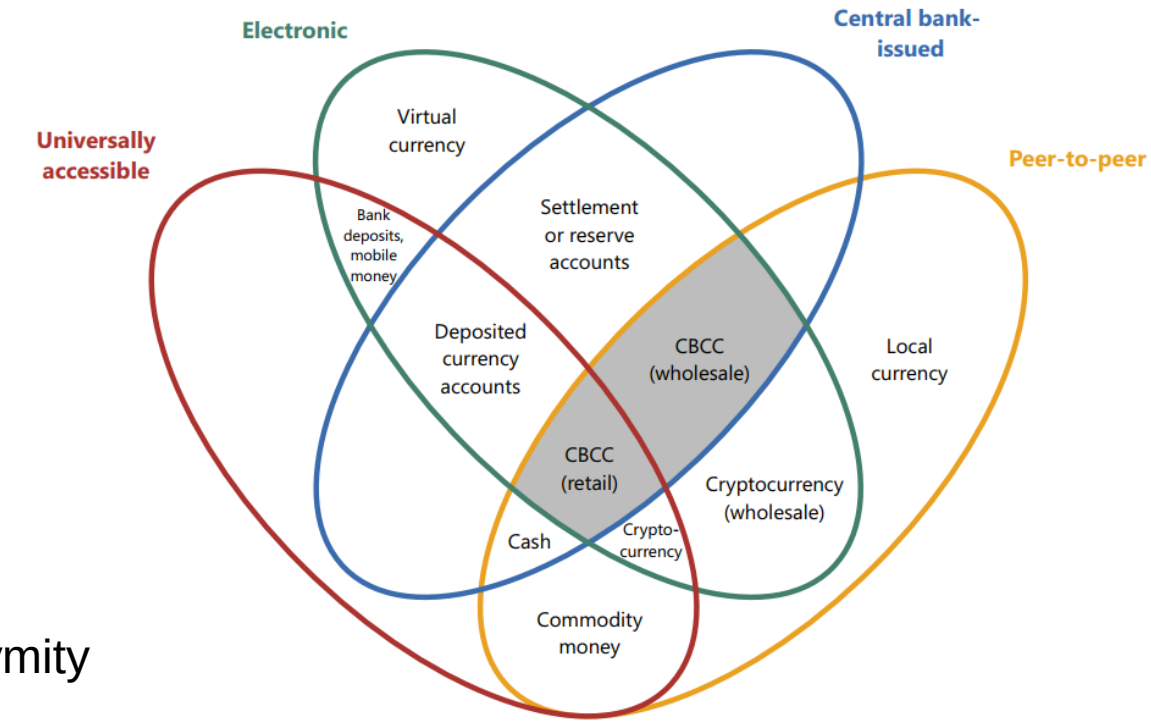
- **Motivation**
  - Banks and governments desire control and authority over the payments market
  - Maintain monetary sovereignty & discourage use of non-regulated cryptocurrencies
  - Curb and limit illegal activities on cryptocurrencies
  - Devise an effective alternative to fiat cash

- **Research Goals**
  - Study and explore different CBDC initiatives taken by banks around the world
  - Design and implement effective blockchain based CBDC models
  - Compare different blockchain platforms for CBDC implementation

Background & Related Work

Background

- **Central Bank Digital Currency (CBDC)**
  - Digital currency issued by a central bank
  - Money flower[3] visually depicts major forms of money and defines CBDC
    - Central bank issued electronic form of money
    - Peer-to-peer transfer
    - Accessibility
  - Classification of CBDC
    - Retail vs Wholesale CBDC
    - Direct vs Indirect CBDC vs Hybrid CBDC
    - Cross-border CBDC vs Domestic CBDC
    - Account-based CBDC vs Token-based CBDC
  - CBDC Design Considerations
    - Geographical, Political & Technological factors
    - Availability; Custody; Transaction Limits; Anonymity



[3] https://www.bis.org/publ/qtrpdf/r_qt1709.pdf

Background

Central Bank Digital Currency (CBDC)

- Retail vs Wholesale CBDC

Features	Retail CBDC	Wholesale CBDC
Accessibility	General Public	Banks and Financial Institutions
Use Case	Daily payments	Interbank settlements
Value of transactions	Small	Large

- Direct CBDC vs Indirect CBDC vs Hybrid CBDC

Features	Direct CBDC	Indirect CBDC	Hybrid CBDC
Presence of intermediary	No	Yes	Yes
Claim on Central Bank	Yes	No	Yes
Division of Responsibilities	No	Yes	Yes
Deferred Settlements	No	Yes	Yes

Related Work

- **Discussion Papers**
  - *Central Bank Digital Currencies* by Bank for International Settlement
  - *The Rise of Digital Money* by International Monetary Fund → Hybrid CBDC
  - *Central Bank Digital Currency Policy Maker Toolkit* by World Economic Forum
- **Technical Papers**
  - *Centrally Banked Cryptocurrencies* → RSCoin
    - Central authority mints digital currency and verified entities (commercial banks) validates transactions
    - Use of sharding
    - Overcomes the issues with Bitcoin
  - *PRCash: Fast, Private and Regulated Transactions for Digital Currencies*
    - Extension of RSCoin
    - Use of Homomorphic Encryption and Zero-Knowledge Proofs to hide transaction details
    - Focuses on user anonymity and privacy

Related Work

- **Banking Initiatives**
  - *Project Jasper* by Bank of Canada
  - *Project Ubin* by Monetary Authority of Singapore
  - *Project Khokha* by South African Reserve Bank
  - *Project Bakong* by National Bank of Cambodia
- **Industrial Initiatives**
  - *Digital Fiat Currency* by VISA
    - Patent registered at USPTO
    - Replace existing fiat currency for digital cash
    - Incorporates private transaction processing networks, central banks, commercial banks & end-users
  - *Libra by* Libra Association
    - Promote financial inclusion
    - Provide a global, open, instant and low-cost payments network
    - Permissioned network with strong regulations and compliance

Main Contribution

- **Blockchain-based Retail CBDC Implementation**
  - Most initiatives taken by central banks have been focused at wholesale level
  - Prototype implementation for CBDC offering at retail level

- **Proposed indirect model mimics general banking environment**
  - Offering of banking services via smart contracts executed on top of blockchain
  - Mapping of real-world user identities with account addresses
  - Regulation over services (blacklisting and whitelisting)

- **Comparison between popular blockchain platforms for CBDC**
  - Direct Model has been implemented using Hyperledger Iroha
  - Indirect Model has been implemented using Quorum

Requirements & Design

Requirements

- **General Non-Functional Requirements of a CBDC System**
  - Payment system should be simple and easy to use
  - Near-immediate transaction finality
  - High-availability
  - API access to third-party applications
  - Stable, non-volatile digital currency pegged at a specific rate to fiat cash
- **Functional & Non-Functional Requirements**
  - Central Banks
    - Initialize or setup blockchain
    - Issue digital currency
    - Create wallet addresses for commercial banks (in indirect model) and for users (in direct model)
    - Distribute digital currency to commercial banks (in indirect model) and to users (in direct model)
    - View transactions
    - Burn digital currency when required
    - Deploy multiple nodes if needed

Requirements

- **Functional & Non-Functional Requirements**
  - Commercial Banks and Intermediaries
    - Create wallet addresses for users
    - Distribute digital currency to users
    - View transactions
    - Offer additional financial services on top of CBDC platform
    - Maintain a local node of the blockchain
  - End-Users
    - Access individual wallet to view transactions and balance
    - Conduct P2P transfers
    - Burn digital currency owned by it
    - Ensure the safety of private keys of corresponding wallet addresses
    - Provide necessary information or documents to banks

Design

- **Design of Direct CBDC Model**

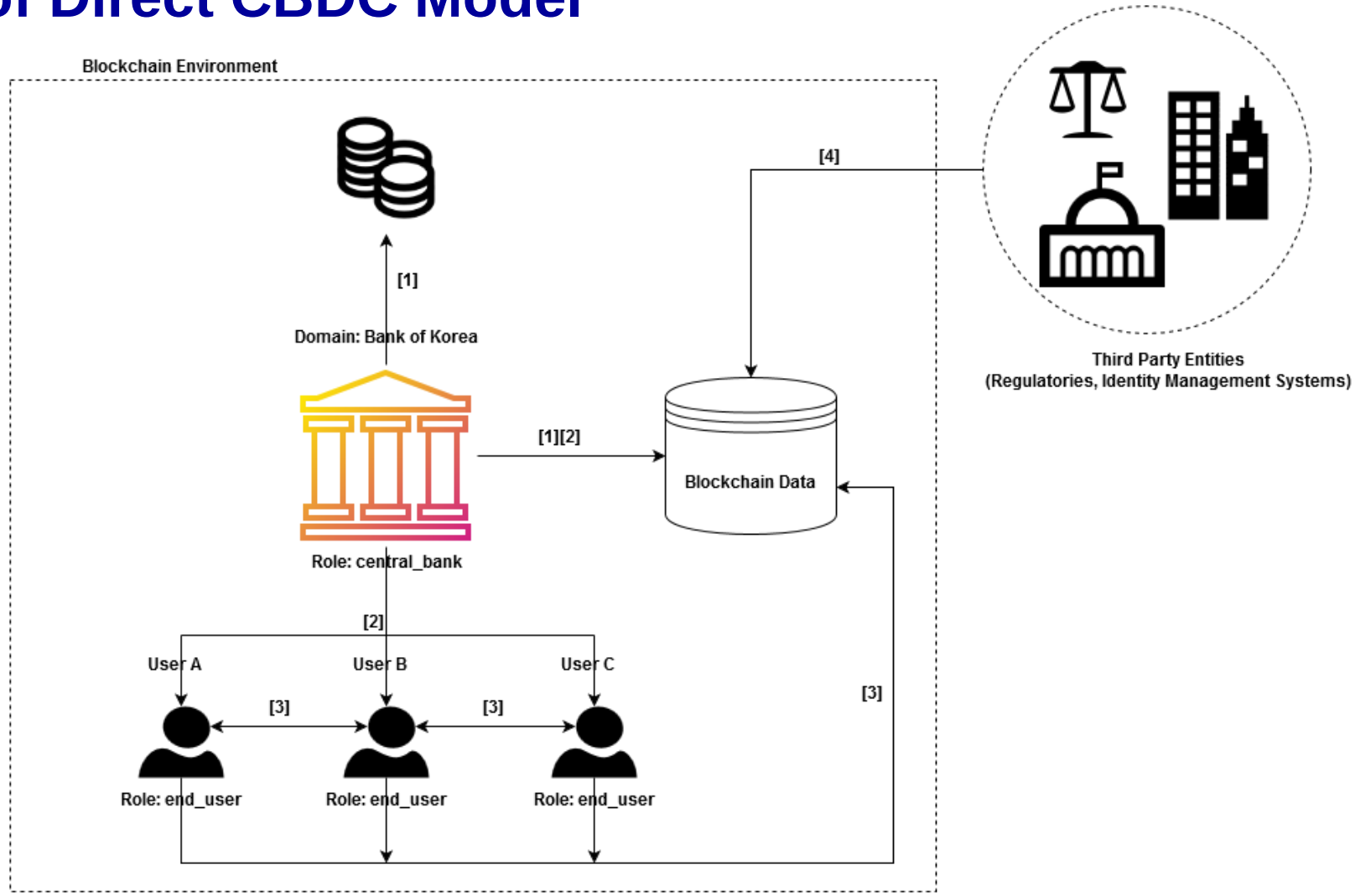


Fig. Proposed design for blockchain-based direct CBDC model

Design

- **Design of Direct CBDC Model**

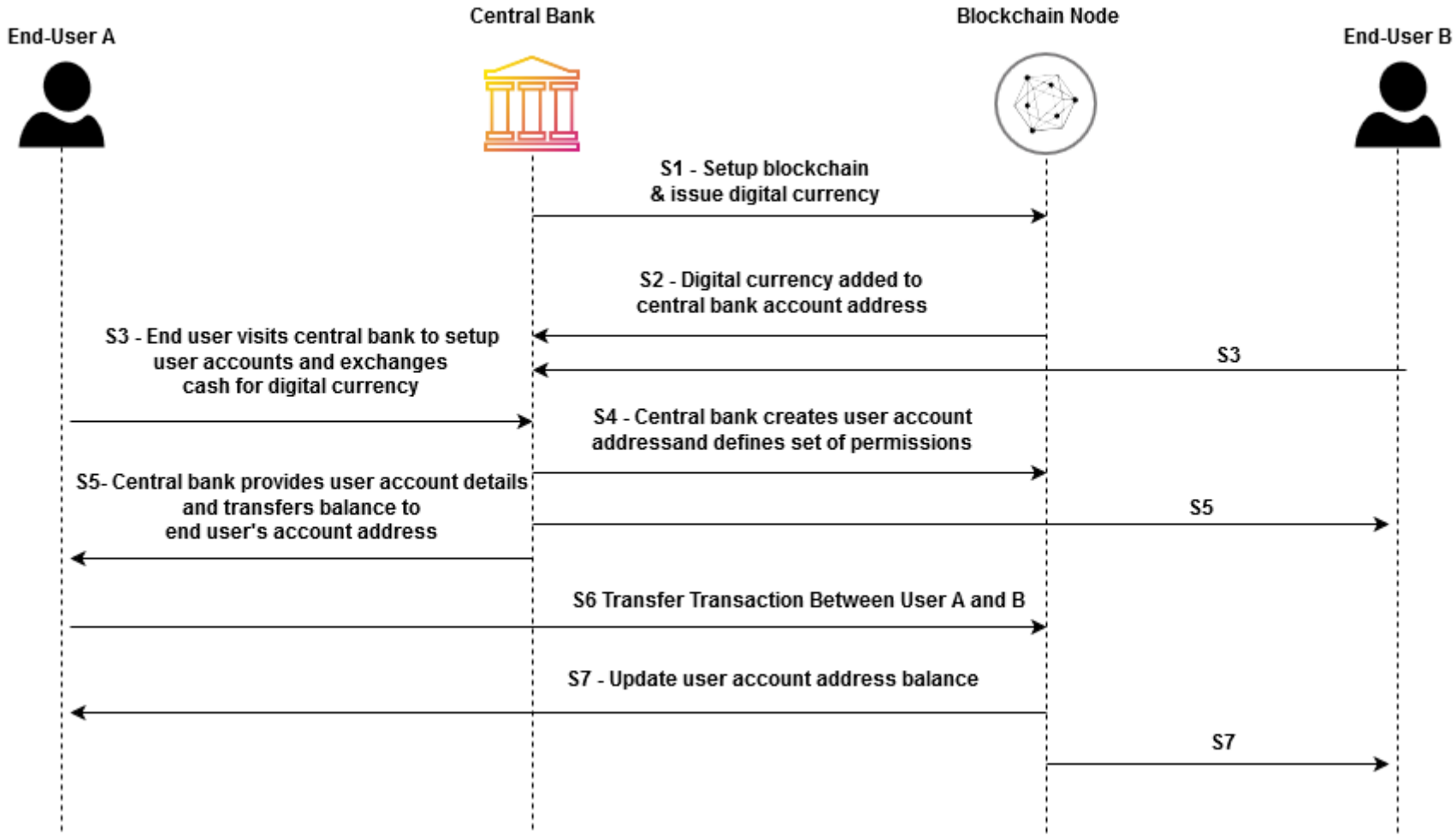


Fig. Sequence diagram depicting process flow in the proposed direct CBDC model

Design

- **Design of Indirect CBDC Model**

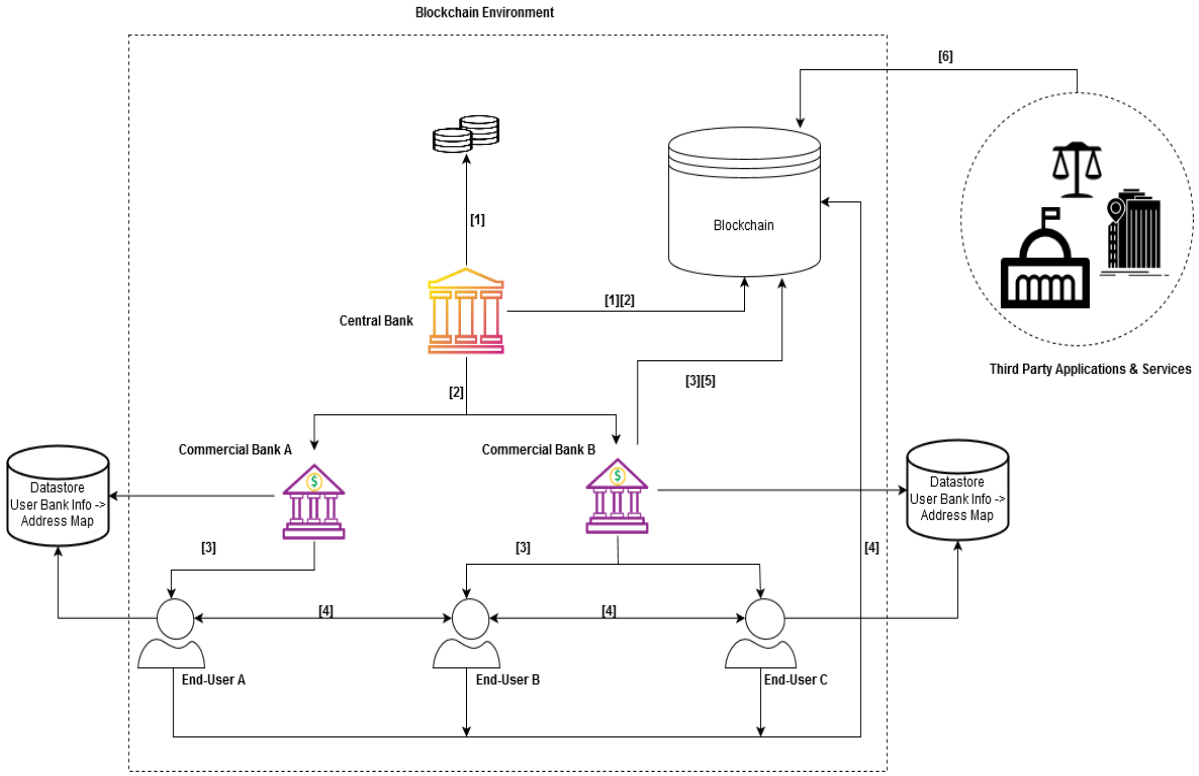


Fig. Proposed design for blockchain-based indirect CBDC model

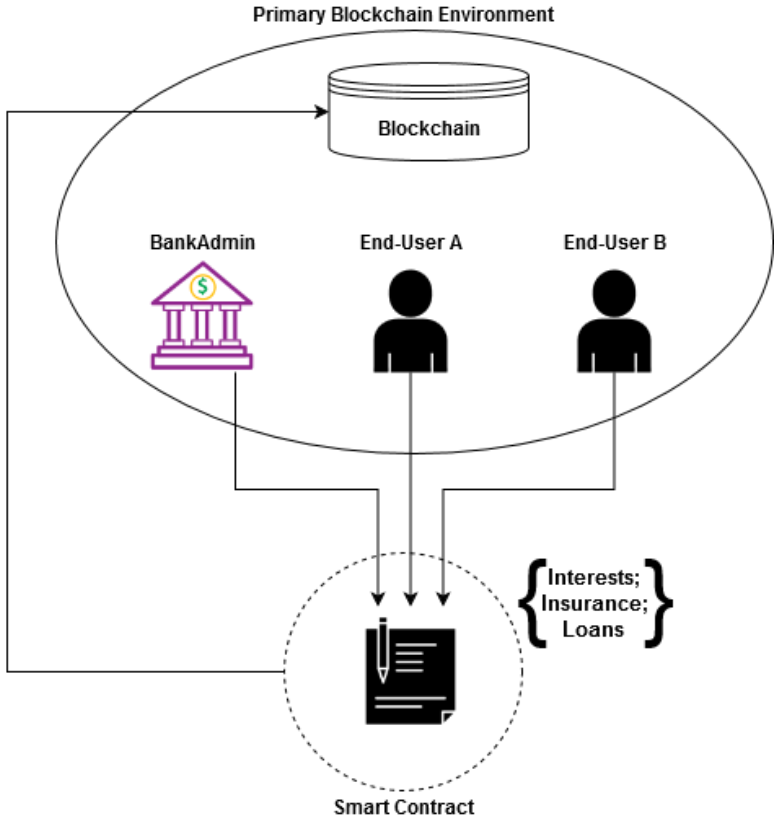


Fig. Offering of financial services via smart contracts

Design

- **Design of Indirect CBDC Model**

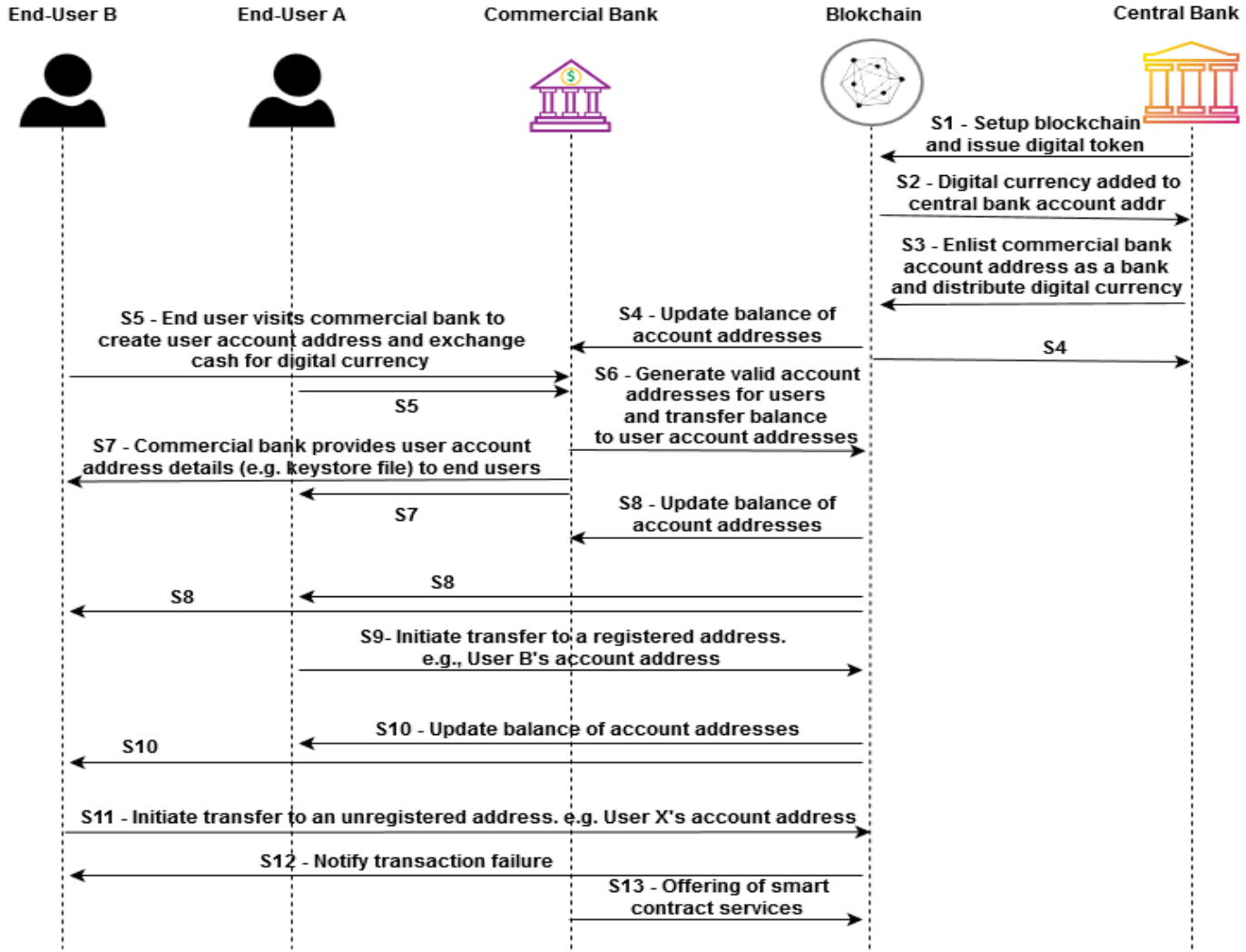


Fig. Sequence diagram depicting process flow in the proposed indirect CBDC model

Implementation

Implementation

Implementation of Direct CBDC Model

- Use of Hyperledger Iroha blockchain platform
  - Permissioned blockchain environment
  - Use Cases: digital asset and identities management, payment systems, logistics, etc.
  - Built-in commands and queries. e.g., create asset, create user, create role, view asset, etc.
- Custom definition of roles are attached to users

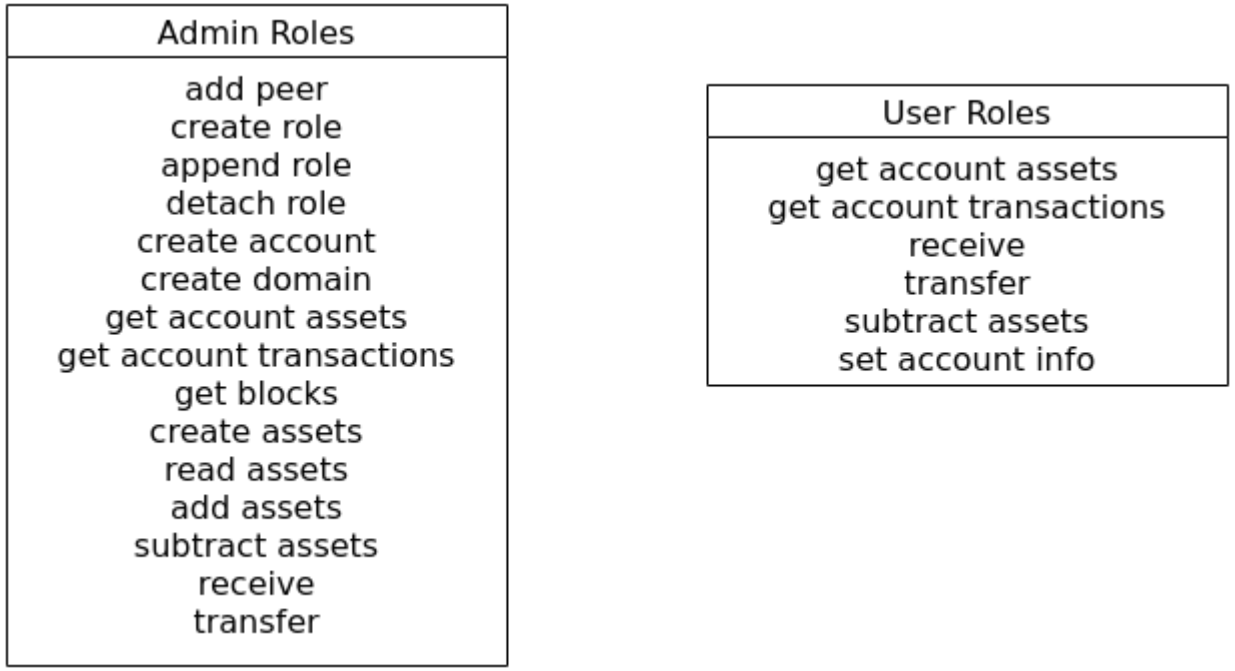


Fig. Custom defined set of roles created for end-user and central bank administrator

Implementation

- **Implementation of Direct CBDC Model**

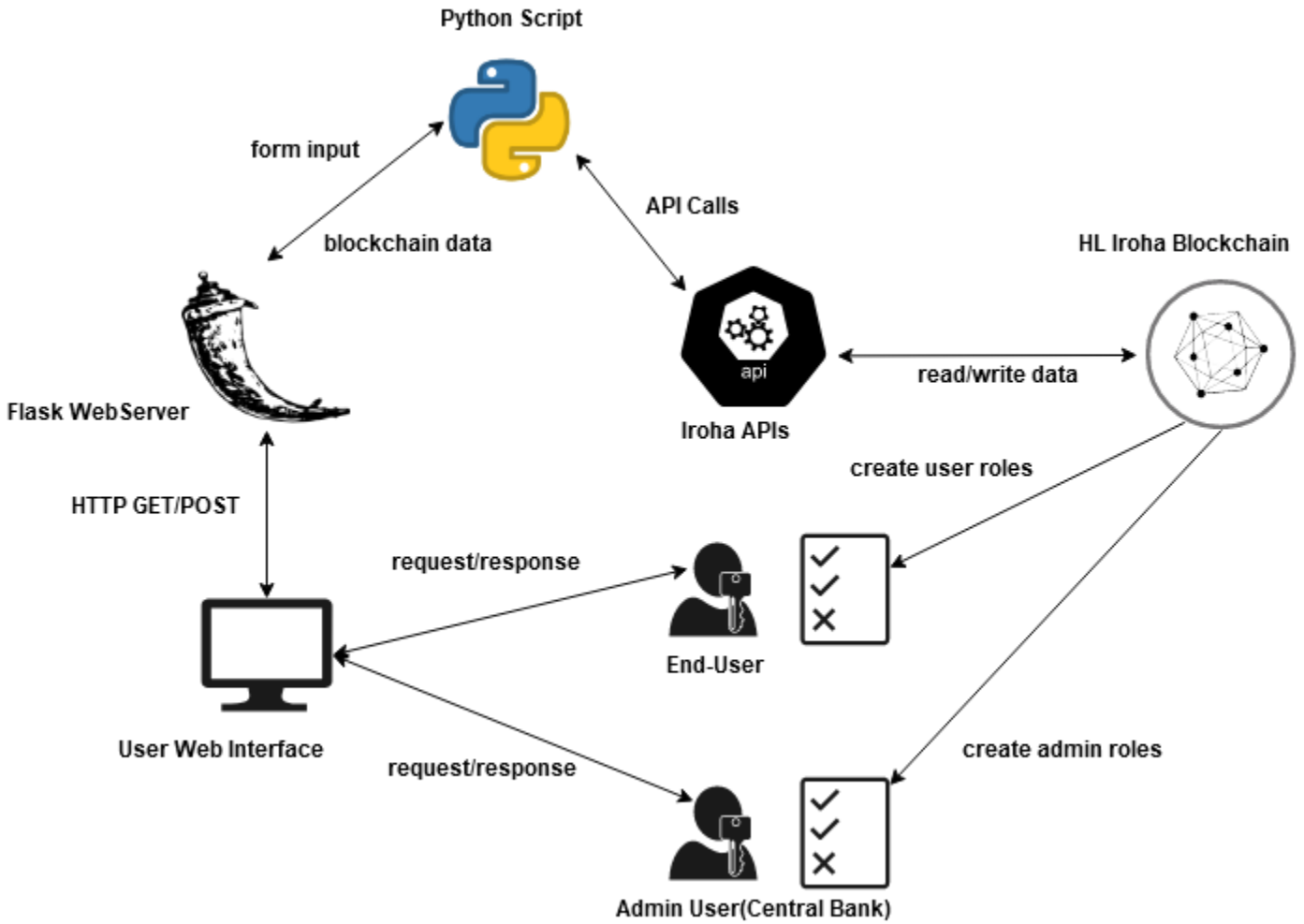


Fig. Implementation details of direct CBDC model using Hyperledger Iroha

Implementation

- **Implementation of Indirect CBDC Model**
  - Use of Quorum Blockchain Platform
    - Permissioned blockchain network
    - Support for “private” transactions
    - Smart contract execution functionality
  - Deployed a token smart contract containing additional financial services e.g., deposit service, mapping of address to real-world identity, controllable actions, etc.

```
struct Deposit {
        uint256 id;
        address payable depositor;
        address payable provider;
        uint256 rate;
        uint256 timeperiod;
        uint256 principal;
        uint256 maxDepositAmt;
        uint256 endtime;
        uint256 starttime;
        uint256 interest;
        uint256 amt;
        bool isAvailable;
        bool notWithdrawn;
}
mapping(uint256=>Deposit) public deposits;
```

```
struct AccountInfo {
        address userAddress;
        address creatorAddress;
        string name;
        string identification;
        uint256 idnum;
}

mapping(uint256=>AccountInfo) public accountDetailsById;
mapping(address=>AccountInfo) public accountDetailsByAddr;
```

```
mapping(address=>bool) public validAddresses;
mapping(address=>bool) public validBankAddresses;

function addUserAddressInfo(...) public bool (returns success) {
    require(validBankAddresses[msg.sender] == true);
    ...
}

function transfer(address _to, uint256 _val) public bool (returns
    success) {
    require(validAddresses[_to] == true);
    ...
}

function makeDeposit(...) public bool (returns success) {
    require(validAddresses[msg.sender] == true);
    ...
}
```

Fig. Smart contract code snippet for deposit object, account address mapping and control of token transfers

Implementation

Implementation of Indirect CBDC Model

- Deposit service implemented using smart contract

Algorithm 1 offerDeposit

Parameters: $rate, timeperiod, maxDepositAmt, msg.sender$

Require: $SenderBalance \geq maxDepositAmt$

Stage 1: Initialize Deposit

$depositCount \Leftarrow depositCount + 1$

$deposits[depositCount].id \Leftarrow depositCount$

$deposits[depositCount].provider \Leftarrow msg.sender$

$deposits[depositCount].rate \Leftarrow rate$

$deposits[depositCount].timeperiod \Leftarrow timeperiod$

$deposits[depositCount].maxDepositAmt \Leftarrow maxDepositAmt$

$deposits[depositCount].isAvailable \Leftarrow true$

Algorithm 2 makeDeposit

Parameters: $id, principal, msg.sender$

Require: $SenderBalance \geq principal$

$deposits[id].maxDepositAmt \geq principal$

$deposits[id].isAvailable = true$

$validAddresses[msg.sender] == true$

Stage 1: Update deposit

$deposits[id].depositor \Leftarrow msg.sender$

$deposits[id].principal \Leftarrow principal$

$deposits[id].starttime \Leftarrow now()$

$deposits[id].endtime \Leftarrow now() + timeperiod$

$deposits[id].notWithdrawn \Leftarrow true$

$deposits[id].isAvailable \Leftarrow false$

Stage 2: Transfer balance

$balanceOf[depositor] -= principal$

$balanceOf[provider] += principal$

Stage 3: Calculate interest and amount

$deposits[id].interest \Leftarrow principal * timeperiod * rate/100$

$deposits[id].amt \Leftarrow principal + deposits[id].interest$

Algorithm 3 withdrawDeposit

Parameters: $id, msg.sender$

Require: $now() \geq deposits[id].endtime$

$msg.sender = deposits[id].depositor$

$deposits[id].notWithdrawn = true$

Stage 1: Update balance

$balanceOf[depositor] += amt$

$balanceOf[provider] -= amt$

Stage 2: Update notWithdrawn flag

$deposits[id].notWithdrawn = false$

Fig. Algorithm for implementing deposit service between commercial bank and end-user using CBDC token

Implementation

- **Implementation of Indirect CBDC Model**

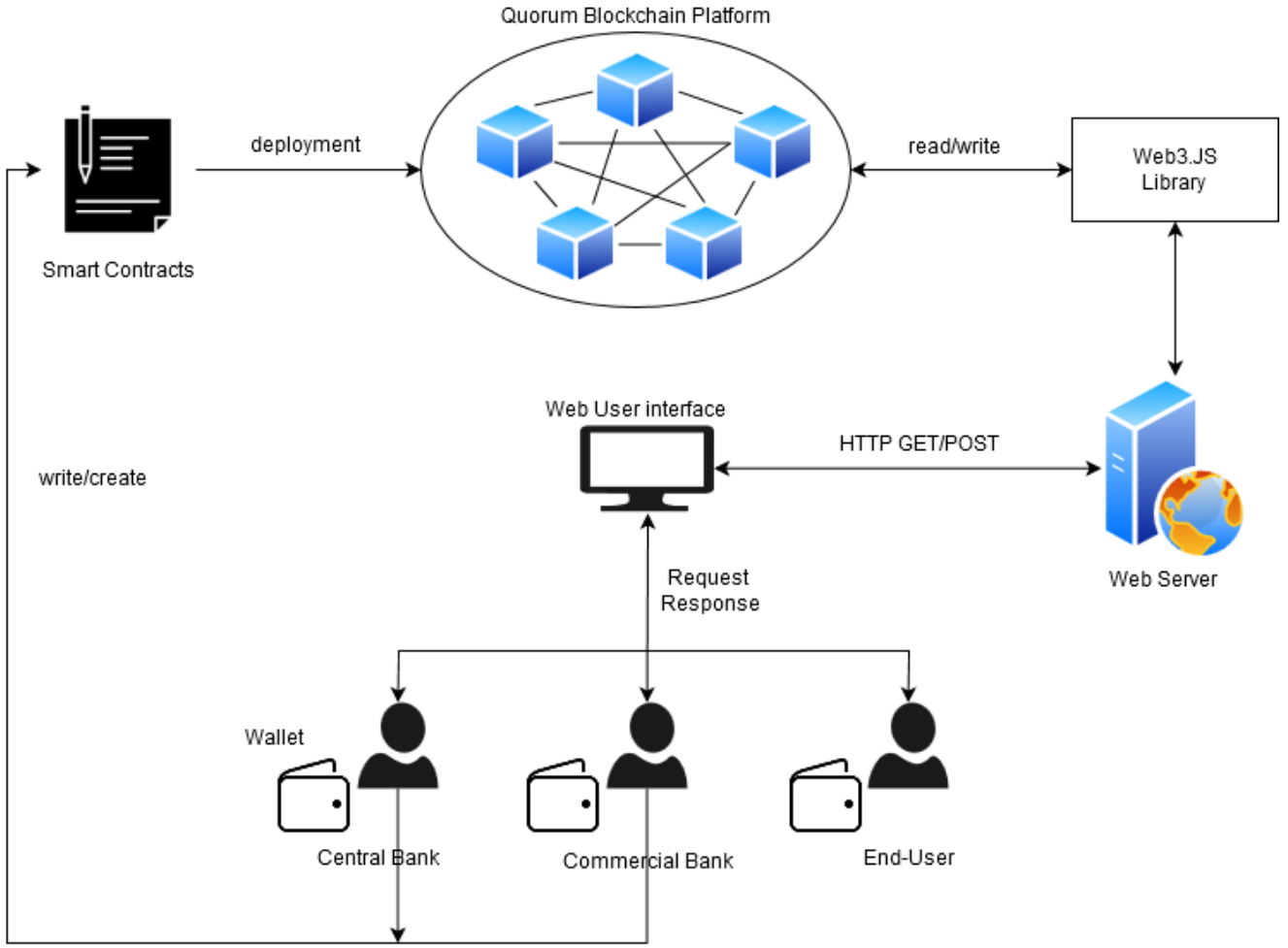


Fig. Implementation details of indirect CBDC model using Quorum blockchain

Evaluation

Evaluation

- **Quorum and Hyperledger Iroha Use Cases**
  - Project Ubin (Retail CBDC)
  - Project Khokha (Wholesale CBDC)
  - Project Bakong (Wholesale CBDC)
- **Quorum vs Hyperledger Iroha**

Features	Hyperledger Iroha	Quorum
Consensus Mechanism	YAC	I-BFT and RAFT
Support for Private Transactions	No	Yes
Smart Contract Execution	No	Yes
Built-In Commands and Queries	Yes	No
Native Cryptocurrency	No	Yes (Ether)
Decimal Precision Support	Yes	No
Metamask Compatibility	No	Yes

Evaluation

- **Pros**
  - Retail CBDC implementation that fits well with current banking environment
  - Offering of financial services by commercial banks
  - Mapping of account addresses to real-world identities + control of actions
  - No use of external datastore
- **Cons**
  - Not absolutely relevant to any central bank initiative
  - “BOKCoin” bears no relation with Bank of Korea CBDC initiative
  - Improvements in UI possible
  - Throughput analysis not performed

Conclusion

Conclusion

- **Summary**
  - Explained the concepts of CBDC
  - Designed and implemented blockchain-based retail CBDC models
    - Direct model implemented using Hyperledger Iroha blockchain
    - Indirect model implemented using Quorum blockchain
  - Offering of financial services via smart contracts on top of CBDC platform
- **Future Work**
  - Mobile application development to encourage ubiquitous use
  - Scalability measure
  - Use of enhanced “privacy” features of Quorum to hide transaction details
  - Testing against known smart contract vulnerabilities

Q&A

감사합니다

Appendix

- **User Interfaces for Direct CBDC implementation**

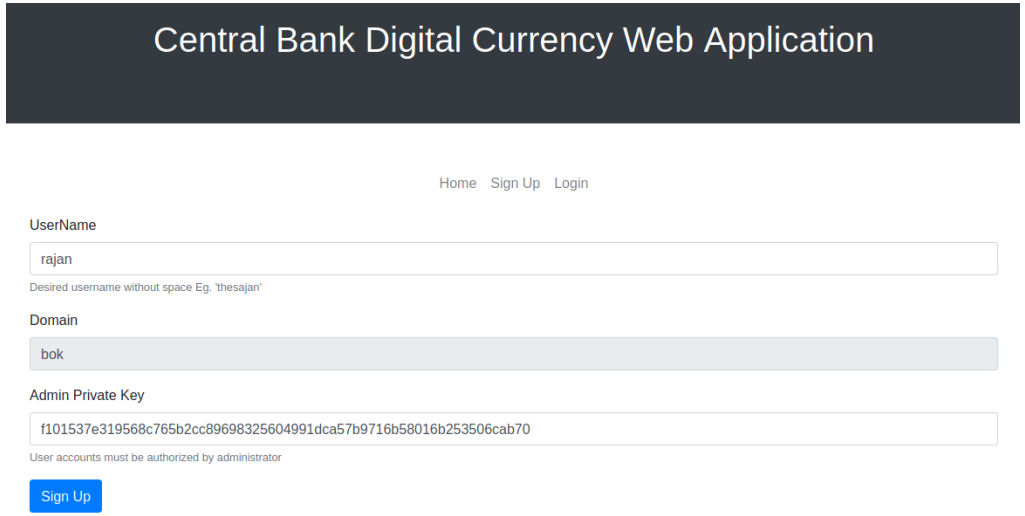


Fig. User account sign-up requires central bank authorization

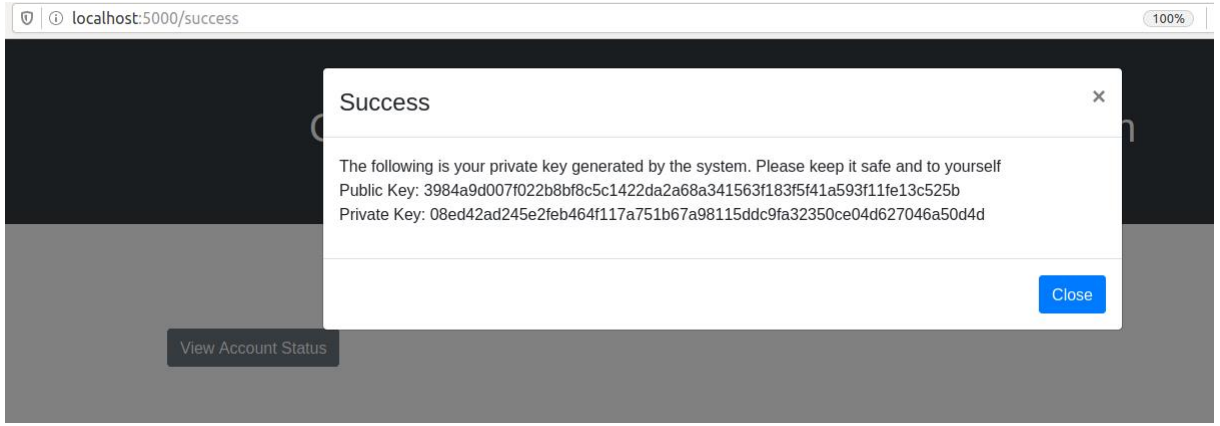


Fig. Private public key-pair generated for an end-user

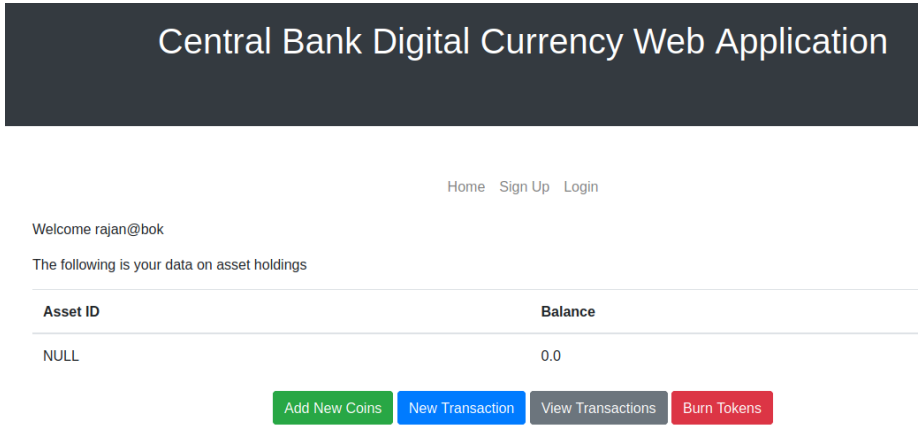


Fig. Newly created user has no initial balance

Appendix

- **User Interfaces for Direct CBDC implementation**

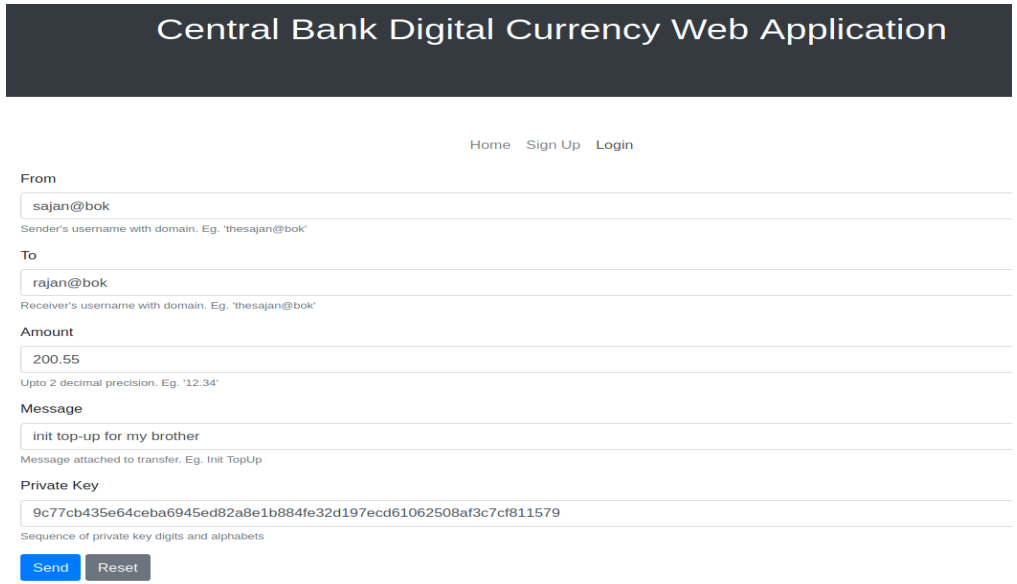


Fig. P2P Transfer between two users in the system

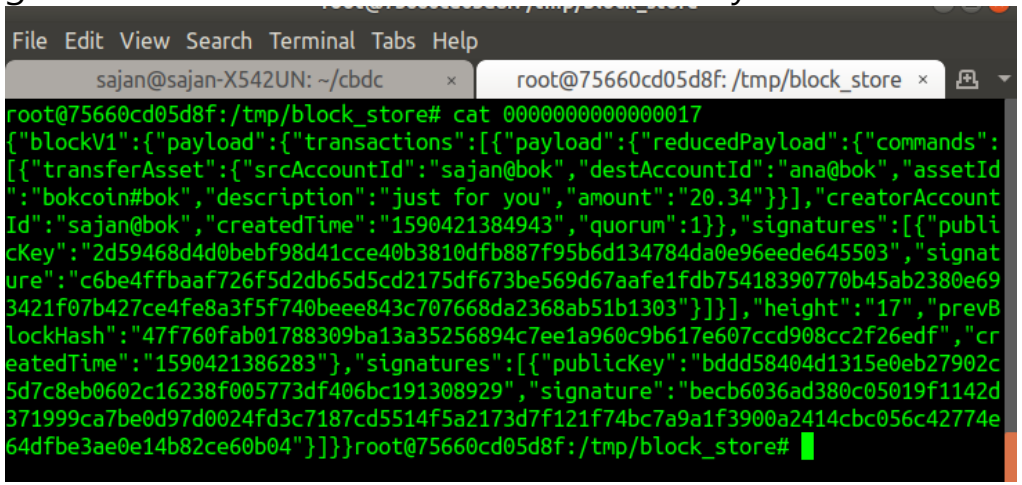


Fig. Transaction data stored in blockchain in JSON format

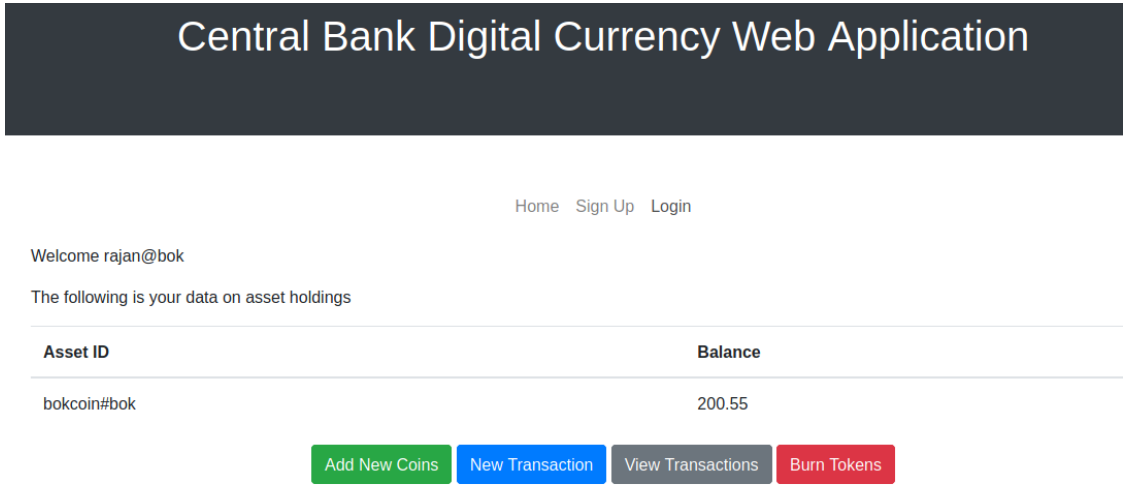


Fig. Update in balance after transaction execution

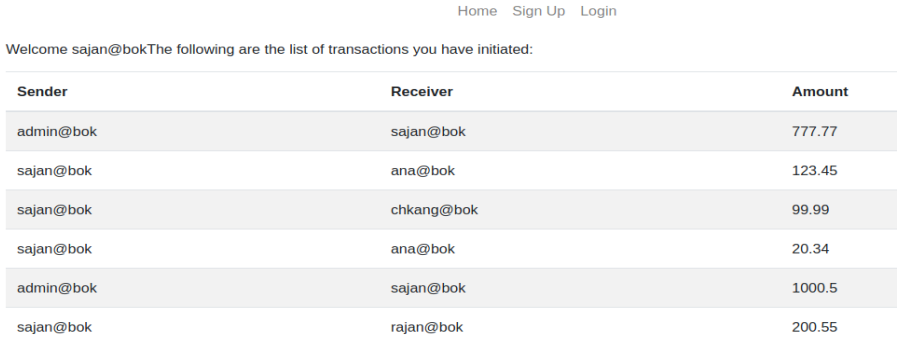


Fig. List of transactions corresponding to a user

Appendix

- **User Interfaces for Indirect CBDC implementation**

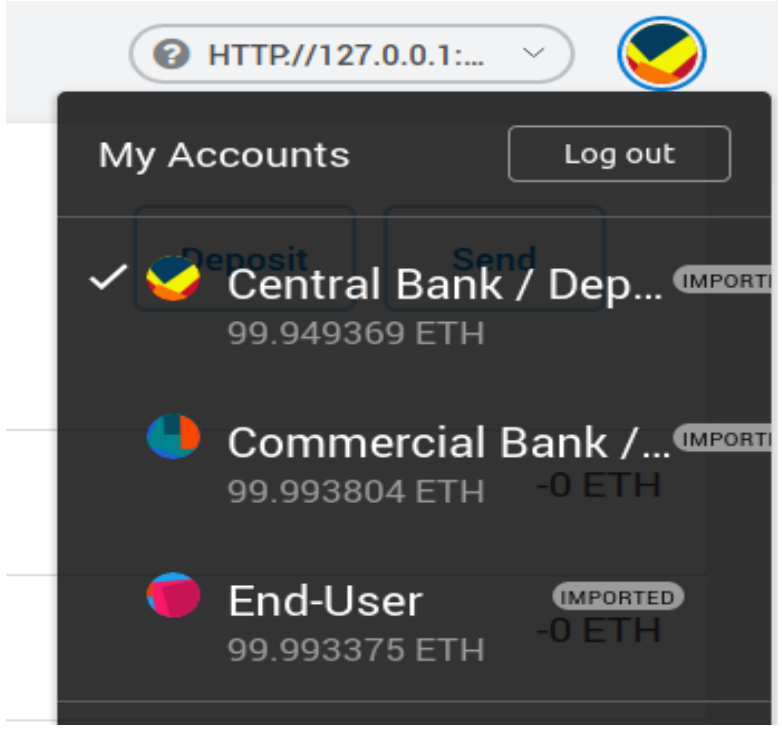


Fig. Example list of registered accounts

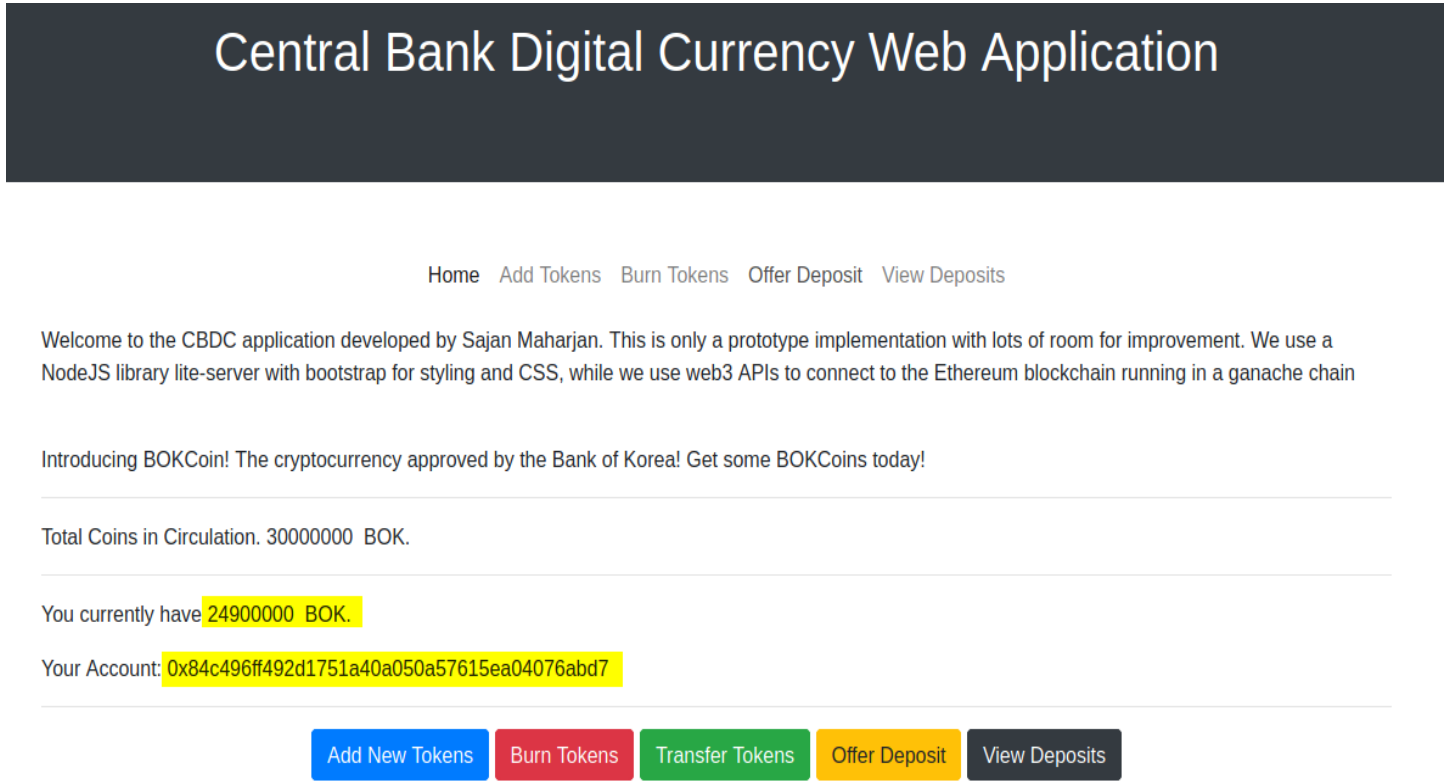


Fig. Landing page showing central bank's account balance

Appendix

- **User Interfaces for Indirect CBDC implementation**

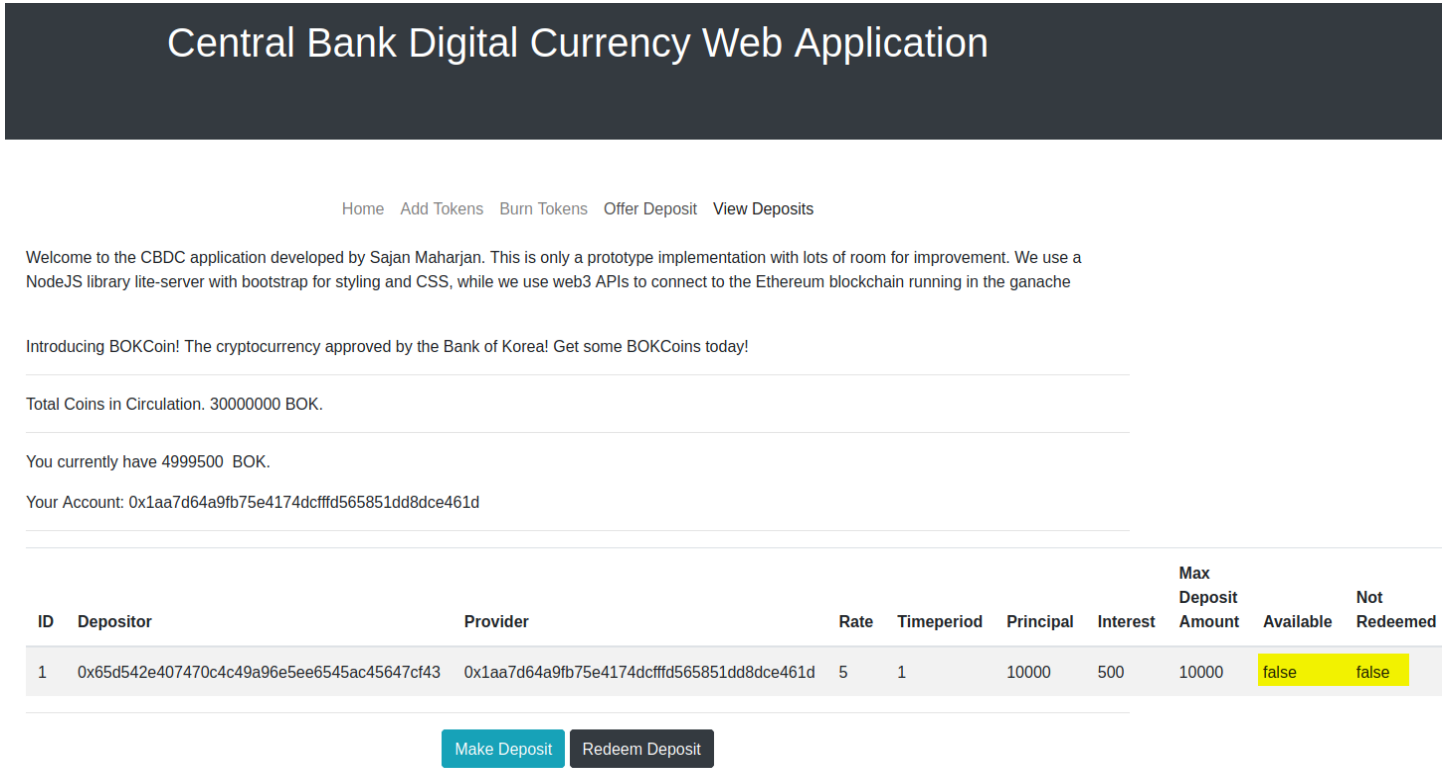


Fig. Initial List of Deposits

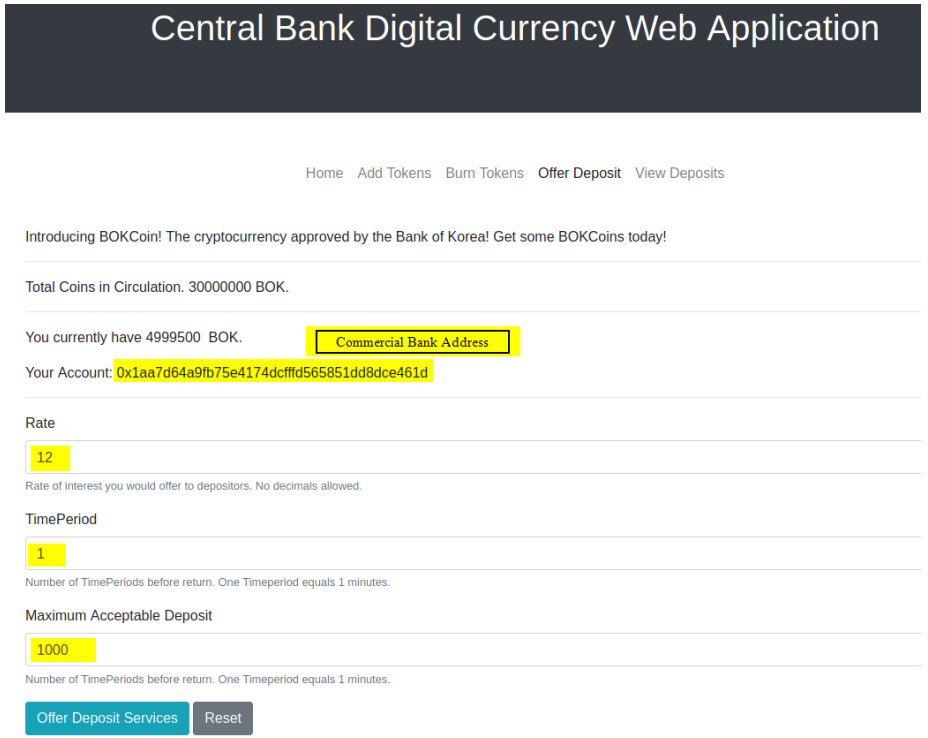


Fig. Offering of deposit by a commercial bank

Appendix

- **User Interfaces for Indirect CBDC implementation**

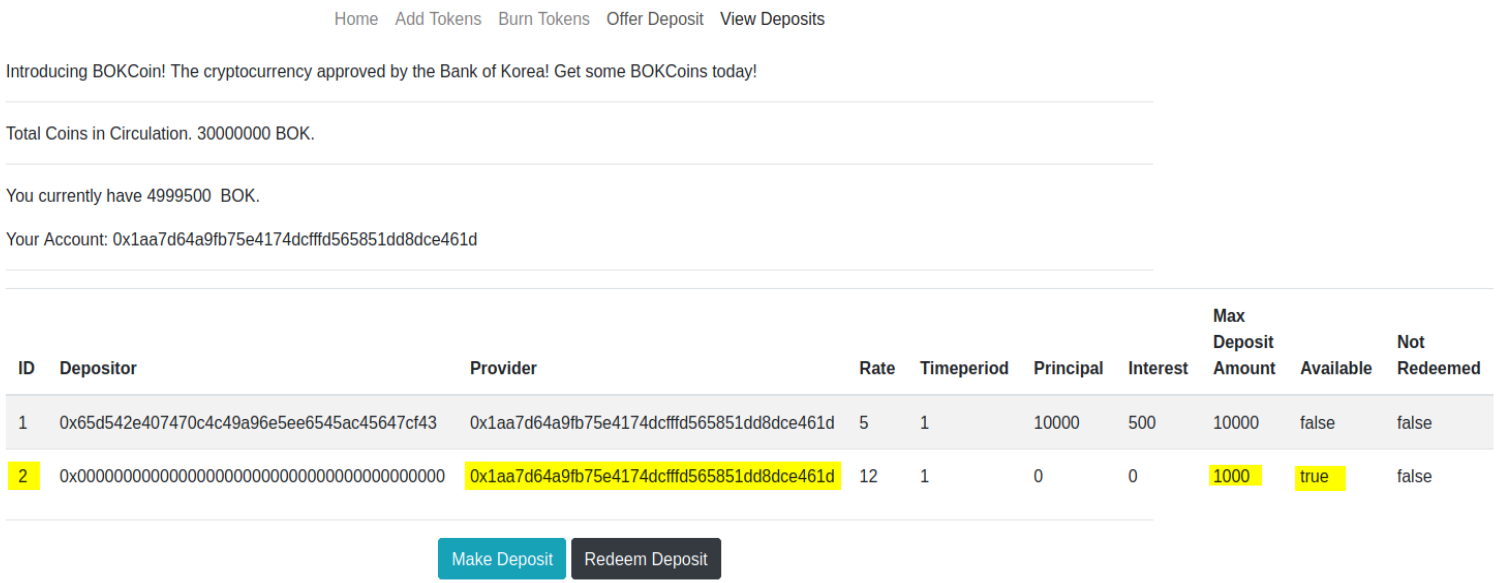


Fig. Deposit offered by bank is added to the list of deposits

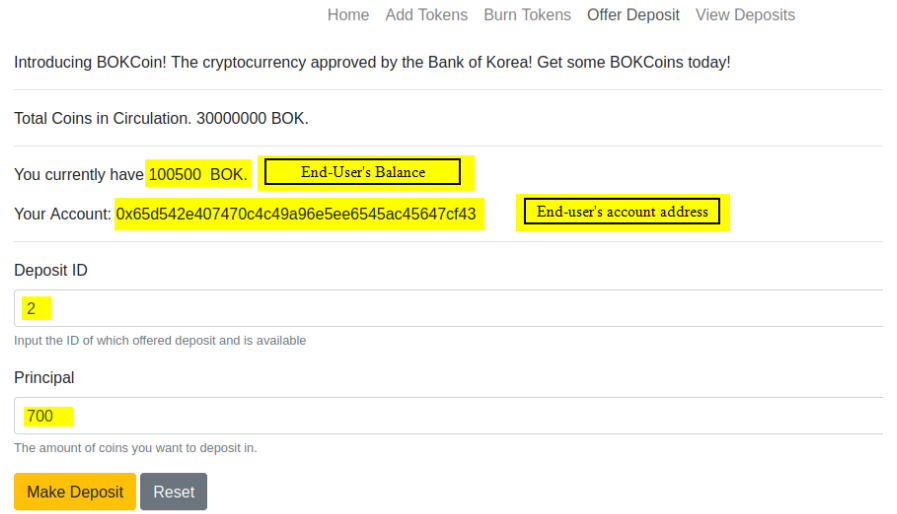


Fig. Interested user making a deposit to the available deposit

Appendix

- **User Interfaces for Indirect CBDC implementation**

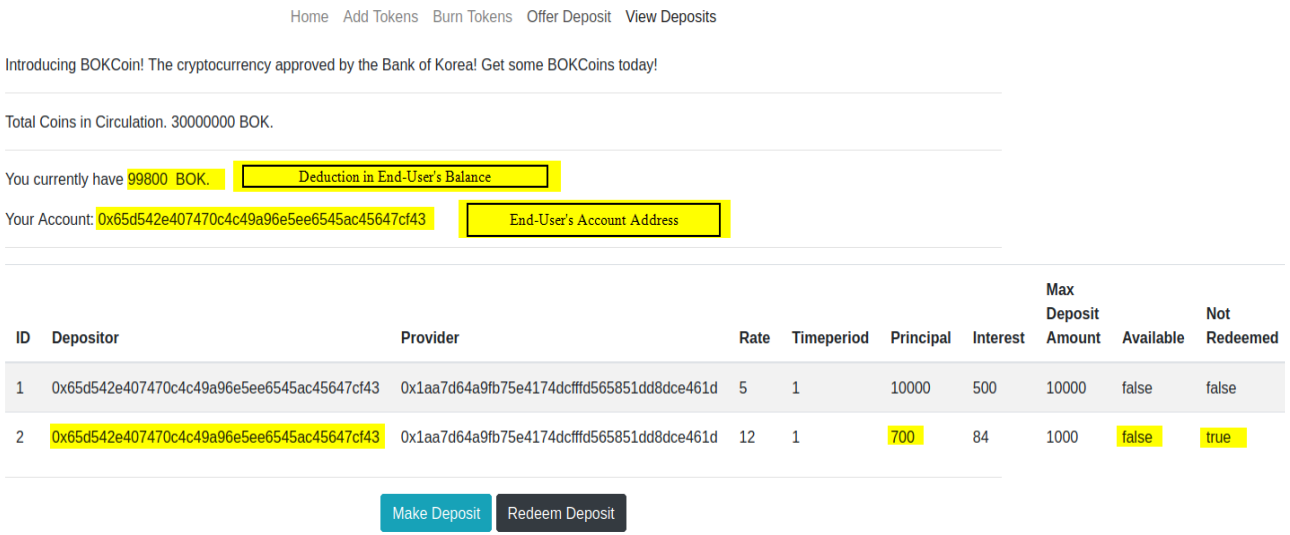


Fig. Deposit object's information is updated on making deposit

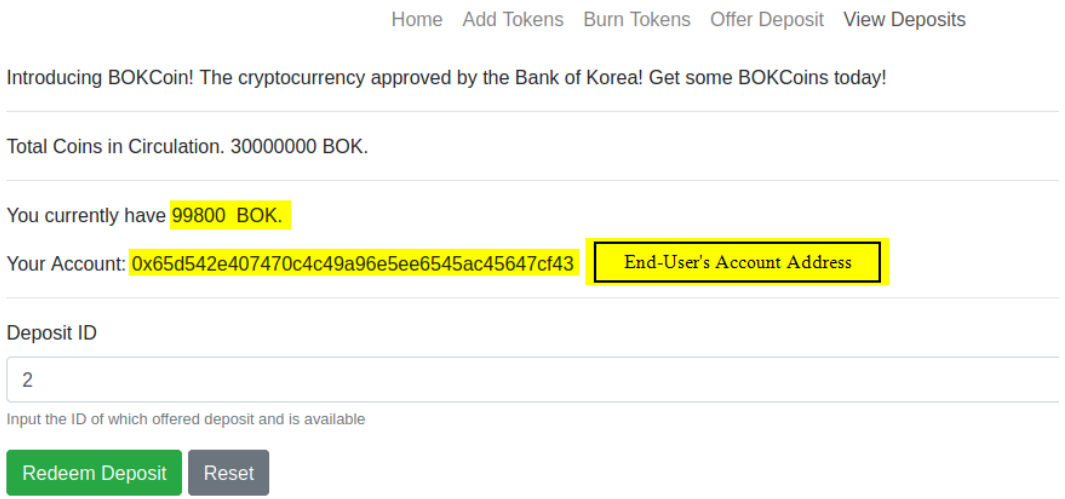


Fig. Withdrawl of user's deposit with interest on maturity

Appendix

- **User Interfaces for Indirect CBDC implementation**

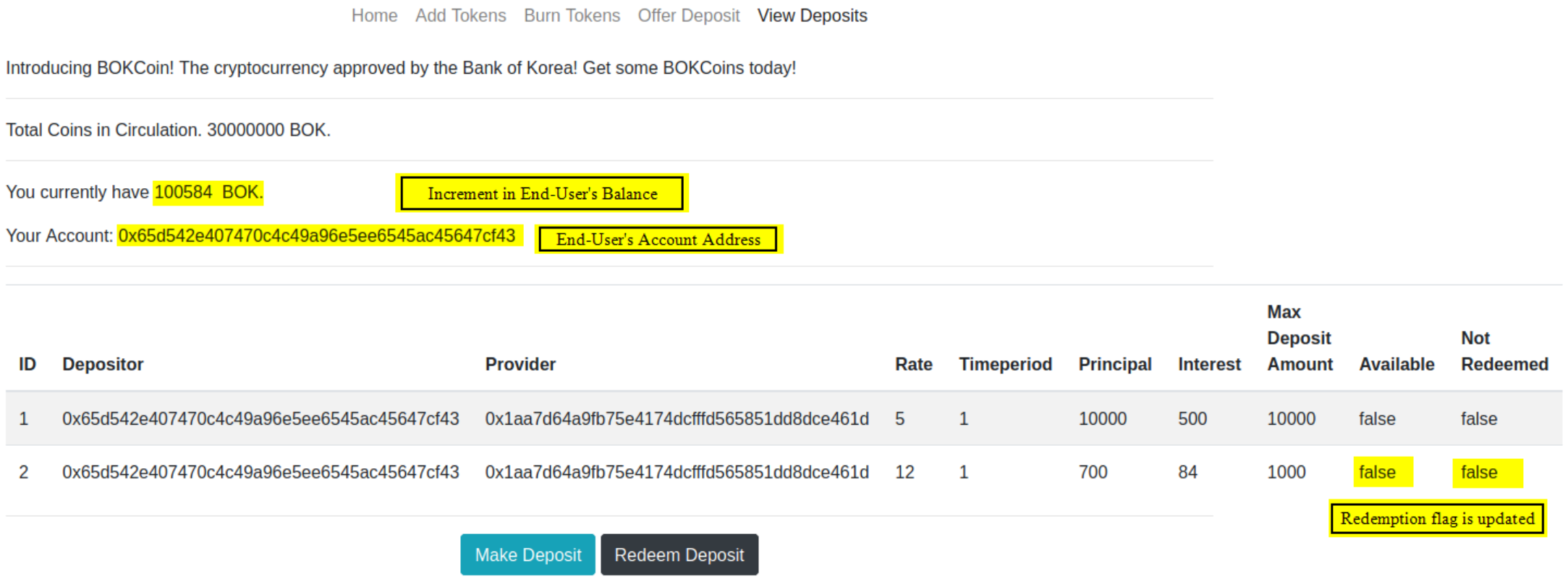


Fig. Change in user's balance and deposit object's flag updated