Article

Blockchain Implementation Method for Interoperability between CBDCs

Hyunjun Jung [ORCID] and Dongwon Jeong *[ORCID]

Department of Software Convergence Engineering, Kunsan National University, Gunsan 54150, Korea; junghj85@kunsan.ac.kr
* Correspondence: djeong@kunsan.ac.kr; Tel.: +82-(063)-469-8912

Abstract: Central Bank Digital Currency (CBDC) is a digital currency issued by a central bank. Motivated by the financial crisis and prospect of a cashless society, countries are researching CBDC. Recently, global consideration has been given to paying basic income to avoid consumer sentiment shrinkage and recession due to epidemics. CBDC is coming into the spotlight as the way to manage the public finance policy of nations comprehensively. CBDC is studied by many countries. The bank of the Bahamas released Sand Dollar. Each country's central bank should consider the situation in which CBDCs are exchanged. The transaction of the CDDB is open data. Transaction registers CBDC exchange information of the central bank in the blockchain. Open data on currency exchange between countries will provide information on the flow of money between countries. This paper proposes a blockchain system and management method based on the ISO/IEC 11179 metadata registry for exchange between CBDCs that records transactions between registered CBDCs. Each country's CBDC will have a different implementation and time of publication. We implement the blockchain system and experiment with the operation method, measuring the block generation time of blockchains using the proposed method.

Keywords: CBDC; CBDC exchange; ISO/IEC 11179; CBDC interoperability blockchain system

1. Introduction

Currently, we use computers and mobile phones to make online remittances and e-commerce transactions on networks such as the Internet, and we exchange data in online transactions. Exchange data has different security requirements depending on the situation. For example, messengers have low integrity requirements for data transmission. However, online transactions require high-integrity data to be used in authentication and transaction authentication. We rely on trusted third-party intermediaries for data integrity and pay high brokerage costs. In online transactions, brokerage costs limit daily transactions at low prices because they raise transaction costs, and non-reversible deals are theoretically impossible. Bitcoin proposed an electronic cash system directly trades without intermediaries between individuals based on cryptographic proofs instead of third-party trust [1–3].

Since Bitcoin was released to the world, many cryptocurrencies have been released. Many cryptocurrencies have been proposed to deal with the shortcomings of Bitcoin or to solve many global problems. Ethereum is a platform that supports smart contracts and decentralized applications (DApps) for monetary functions [4–6]. Smart contracts can express the contents of the contract in a language (Solidity) and automatically execute the contract when the predetermined conditions are satisfied. They can automate by combining insurance, real estate, and copyright. Ripple is a cryptocurrency to support real-time money transfers among banks around the world [7]. International remittances typically use SWIFT (Society for Worldwide Interbank Financial Telecommunications). SWIFT takes two to three hours to complete a trade, and different banks have different fees. Ripple aims to efficiently and immediately process remittances anywhere in the world to replace SWIFT.

Additionally, Ripple wants to reduce the cost of remittance by using cryptocurrency. As such, different cryptocurrencies have been proposed to solve real-world problems.

Countries should prepare for cashless societies to avoid problems. A cashless society is a society in which cash has decreased due to the activation of non-cash payment methods such as credit cards and mobile payment methods. Countries entering cashless societies (Sweden, New Zealand, UK) have common problems with weakening cash access, banking and consumption activity of vulnerable groups, and weakening public currency distribution systems. Central banks study the Central Bank Digital Currency (CBDC) in preparation for a cashless society [8,9].

Facebook announced the cryptocurrency Diem (formerly known as Libra) in June 2019 [10]. Diem has announced plans to use a single, universal digital currency. Diem guarantees its cryptocurrency through the Diem Reserve, which consists of real-asset collateral. The Diem Reserve provides value assurance by organizing funds into a collection of gold, dollars, euros, bank deposits, and short-term government securities. Diem can be used globally by Facebook's 2.3 billion users (as of January 2019). However, governments—such as that of the United States and European countries—have expressed concern, and the US House of Representatives proposed a law to stop Diem. This objection to Diem hinged on the assertion that the pace of Diem infringes upon the sovereignty of a government's monetary system and can devastate the current global financial system.

In December 2019, the central bank of Bahamas began the Sand Dollar project. Sand Dollar (a digital version of the Bahamian dollar) was officially launched in October 2020 through authorized financial installations (AFIs). The People's Bank of China announced its high-priority CBDC research and development in 2020. China wants a globally accepted digital currency using the yuan-based CBDC [11]. As the largest trading country, China occupies more than 10% of the world's trade [12] and allows payments in RMB for trade settlements. China intends to replace the dollar-key currency by increasing the rate of international payments in yuan. The Chinese government will use the yuan CBDC to open the financial market and increase international exchange. As a result, China's global influence will expand over the long term. We should consider the environment of using CBDCs made in each country rather than using a credit card for overseas payment.

The Bank for International Settlements predicts that 20% of the world's population will use a CBDC within three years [13]. It furthermore announced that emerging countries are rapidly developing CBDCs and developed countries are cautious. Eventually, central banks will develop CBDCs and conduct trade accordingly. This paper proposes a transaction method between CBDCs. CBDC transactions are different from existing cryptocurrency transactions in that the former have clear accountability and guarantee an exchange equivalent to real money. However, the exchange of CBDCs requires cooperation between central banks, which should keep records of currency movements between countries. This paper proposes a blockchain system based on the ISO/IEC 11179 metadata registry for exchange between CBDCs. Specifically, we propose a CBDC design based on a metadata registry for consistent management and sharing of CBDCs and a blockchain operation method for exchange between CBDCs.

Section 2 of this paper introduces related research on the cryptocurrency exchange. Section 3 proposes a CBDC management method based on ISO/IEC 11179 and a blockchain operation method. Section 4 shows how to implement the proposed CBDC blockchain system in Bitcoin and Ethereum. Section 5 compares the block generation time of blockchain with the proposed method and compares the proposed method with the features of cryptocurrency exchanges. Finally, Section 6 concludes this paper.

2. Related Work

Cross-border payments are highly inefficient, and blockchain technology can play a role in improving them. One approach may be through the interoperability of CBDCs. To date, only one central bank has issued CBDCs, so interoperability between multiple CBDCs is conceptual envisaging.

The Bank of the Bahamas has launched Sand Dollar, the world's first CBDCThe Bahamas have more mobile phones than bank accounts. Sand Dollar is designed for financial integration. Customers can pay with Sand Dollar to any vendor with a digital wallet certified by the Central Bank of the Bahamas [14]. The National Bank of Cambodia started the Bakong system. The Bakong system will be issued in digital currency by Cambodian partner organizations, not by the central bank. Anyone with a phone number under their name in Cambodia can use the Bakong system and pay Cambodian riel or dollars [15].

Bank for International Settings (BIS) Innovation Hub works on The Multiple CBDC (mCBDC) Bridge Project [16]. mCBDC Bridge is a multi-currency cross-border project that explores the capabilities of distributed ledger technology (DLT) and studies the application of CBDC to strengthen financial infrastructure. The participating authorities of the project are the Hong Kong Monetary Authority, the Bank of Thailand, the Digital Currency Institute of the People's Bank of China, and the Central Bank of the United Arab Emirates. This project explores how CBDC can be used to address the challenges of cross-country funding delivery, including inefficiency, high cost, low transparency, and compliance.

The BIS conceptually proposes three models of compatible CBDC systems (model 1), interlinked CBDC systems (model 2), and a single system for mCBDC (model 3) [17]. Compatible standards allow payment systems to reduce friction and barriers to cross-border monetary services. Model 1 is probably very similar to a traditional payment system. CBDC, like payment methods, can reduce the operational responsibility of participating in multiple systems through common technical standards such as message format, encryption technology, data requirements, and user interfaces. Linking payment systems are complex tasks and often require compatibility measures. Model 2 would simply not work together to bring the system together. Two forms can be taken for implementation of the model: (i) a shared technology interface (ii) a common deletion mechanism (a distributed shared account or a more centralized universal payment agent or system). Finally, a multinational payment system with CBDCs is possible. Model 3 potentially allows for more operational capabilities and efficiency, but increases governance and control hurdles. Through the shared network, participants can make payments across borders through deposit receipts linked to CBDCs held in the domestic system. However, a single mCBDC system creates policy problems for the central bank. For example, the scope of CBDC issuance for monetary policy and financial stability and payment policy.

3. The Proposed Method

3.1. CBDC Management Method Based on ISO/IEC 11179

The ISO/IEC 11179 metadata registry is an international standard for data interoperability. Interoperability is a property in which systems can be used interchangeably with other systems of the same or heterogeneous type. Data interoperability can be understood and exchanged without any additional work by other systems or repositories.

A computer does not know when to distinguish between specific data with equivocal meanings. For example, when the word "Apple" is given to a computer, the computer does not distinguish between "Apple" as in the fruit and the Apple Corporation. As the data for representing other data, metadata is used to distinguish the meaning of data, and the metadata structure the means for interpretation of the data. Metadata expression languages include Resource Description Framework (RDF) [18] and Web Ontology Language (OWL) [19].

The ISO/IEC 11179 Metadata Registry (MDR) is a metamodel that manages metadata through registration and authentication of metadata [20]. The MDR supports data element creation, registration, and management to share information between systems; it structurally supports the sharing, expression, and identification of the meaning of data and can involve data interoperability using the data stored.

We manage the CBDC to the data elements of the MDR. An efficient exchange of CBDCs should ensure that the sender and receiver interpret the same CBDC information.

CBDC information sharing requires the expression of a standardized CBDC structure. Central banks will build CBDCs with their technology and guarantee its transaction. In this section, we discuss how to exchange and receive transactions between CBDCs. MDR can finely express the data and represent it to the Concept layer and Representation Layer. This may involve representing such data value as the range of the domain. In other words, we can identify the data in semantic form using the MDR. While the integrity of data stored in the blockchain is difficult to maintain, as in a database, MDR can help ensure the integrity of the CBDC.

The MDR manages data using data elements. Figure 1 presents an overview of the metadata registry. The data elements of the MDR represent the data as the Concept layer and Presentation layer. The concept layer expresses data and domain information with the Data Element Concept and Conceptual Domain. The Representation layer implies a data element and value domain that expresses the data value format and measurement units. The MDR can obtain the conceptual classification of data and expression information for data values. The "*" and "1" of the Data Element are the relationships of each element. A single Value Domain can relate to many Data Elements.
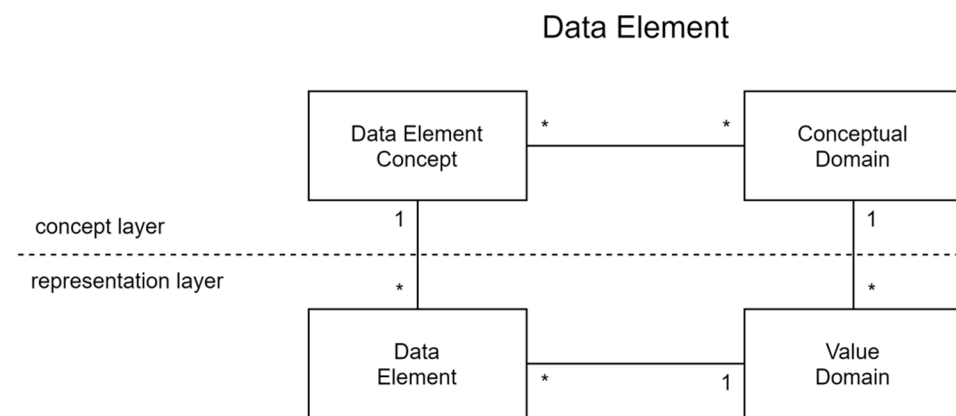
Figure 1. ISO/IEC 11179 Metadata Registry—Data Element.

Figure 2 shows the data element structure. Data elements are represented by Object Class, Property, and Representation. Object classes provide a structure for representing objects in data and represent concepts that contain data. For example, the object class of the Korea CDBC is Digital Currency. Property is a common feature that can distinguish object classes. For example, the Bank for International Settlements (BIS) distinguishes the digital currency as a CBDC, Non-CBDC Token, and Non-CBDC Account. The Data Element is expressed as a combination of value domain, data type, and unit of measurement. Figure 3 is a CBDC example of the data element concept; it can express an object class as digital currency and property as a central bank.

The data value of the MDR is expressed using a conceptual domain and a value domain. Figure 4 shows the relationship between the Conceptual Domain and the Value Domain. The MDR expresses the data in the descriptive or enumerated form. The descriptive form expresses the data values in the Described Conceptual Domain and Described Value Domain. The enumerated form expresses the meaning of the data in the Enumerated Conceptual Domain, Value meaning, Permissible value, and Enumerated Value Domain. The CBDC can represent data using enumerations. For example, Bitcoin expresses BTC and Satoshi units and a range of data values. Ethereum can set ETHER, WEI units, and expressive ranges.
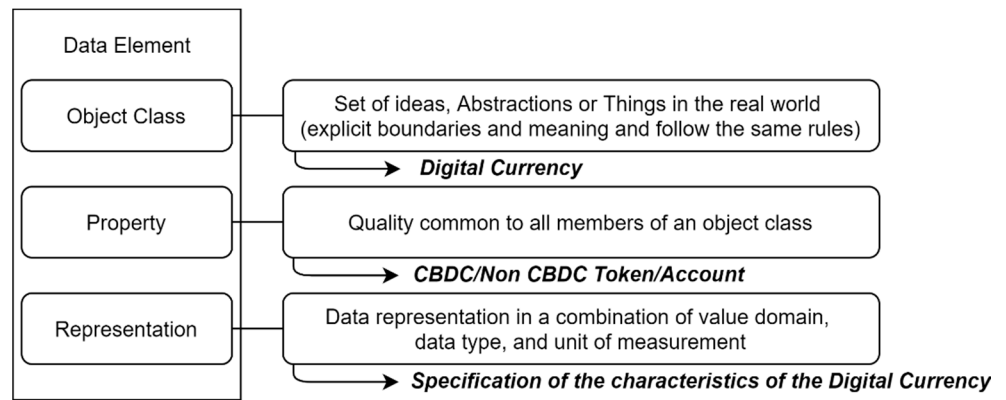
Figure 2. Data Element Structure of MDR.
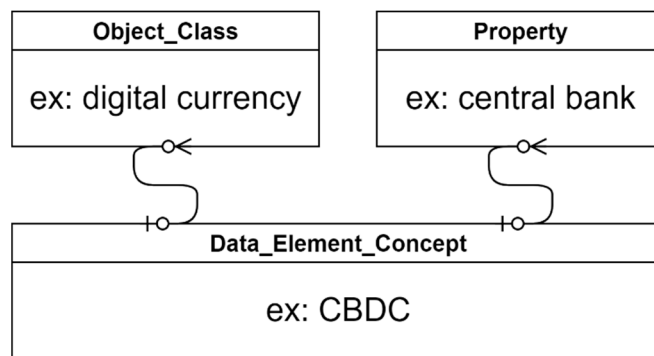
Figure 3. Data Element Concept example of MDR.
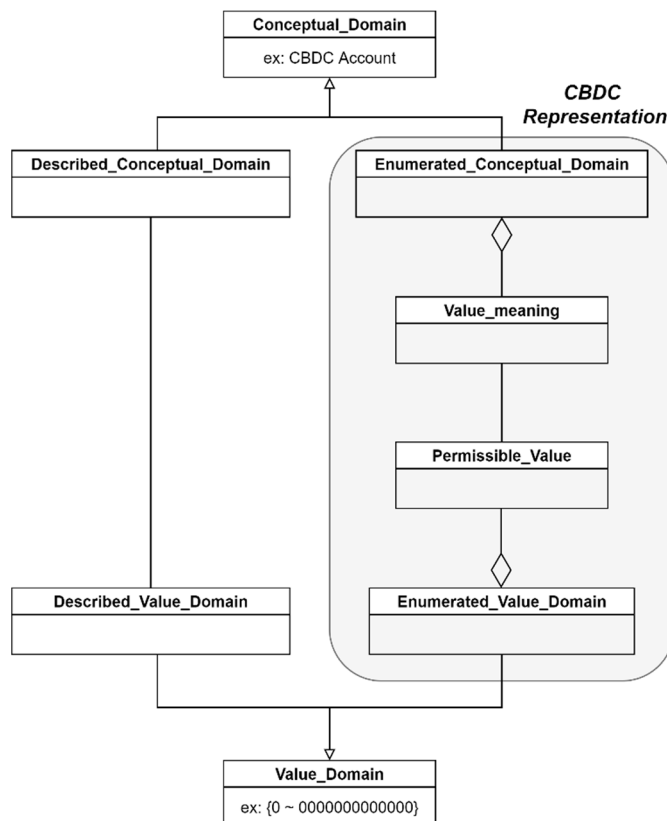
Figure 4. Conceptual Domain and Value domain example of MDR.

The central bank can register and manage CBDCs in the MDR. ISO/IEC 11179 Part 6 Registration proposes a data element registration method in MDR. Figure 5 shows the CBDC registration process. Central banks organize groups for CBDC transactions. According to the registration procedure, the central bank group conducts the Submitter, Steward, and Control Committee agrees on identification, submission, checking, and confirmation, and can register and identify the CBDC in the MDR. Moreover, central banks can keep the CBDC registered in the MDR updated.
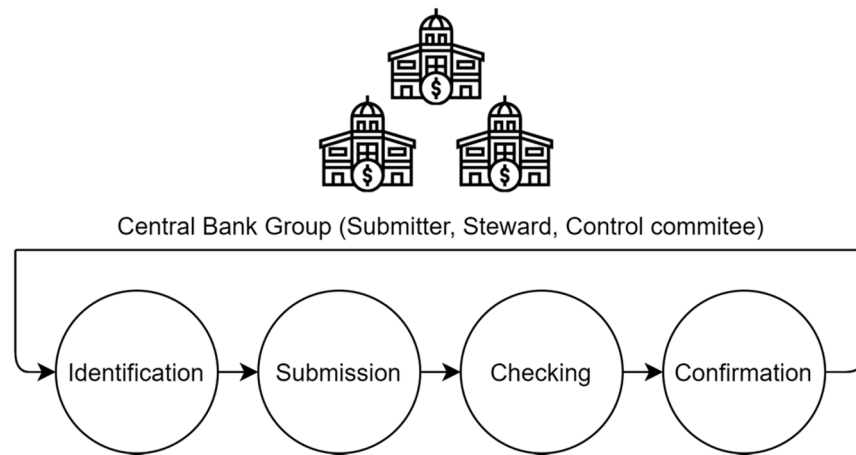
Figure 5. Registration procedure of CBDC.

3.2. Blockchain System for Exchange between CBDCs

This section describes the MDR-based blockchain system for exchange between CBDCs. Figure 6 is an overview of the MDR-based blockchain model. In the blockchain system, the central bank group registered in the MDR participates as a node. The blockchain contains transactions between CBDCs in the transaction, and. The nodes of the blockchain group determine the blockchain consensus. The transaction consists of the sender's amount and CBDC address (e.g., A kr_CBDC) and the receiver's Amount and CBDC address (e.g., B ch_CBDC). The MDR helps to validate the CBDC of the transaction.
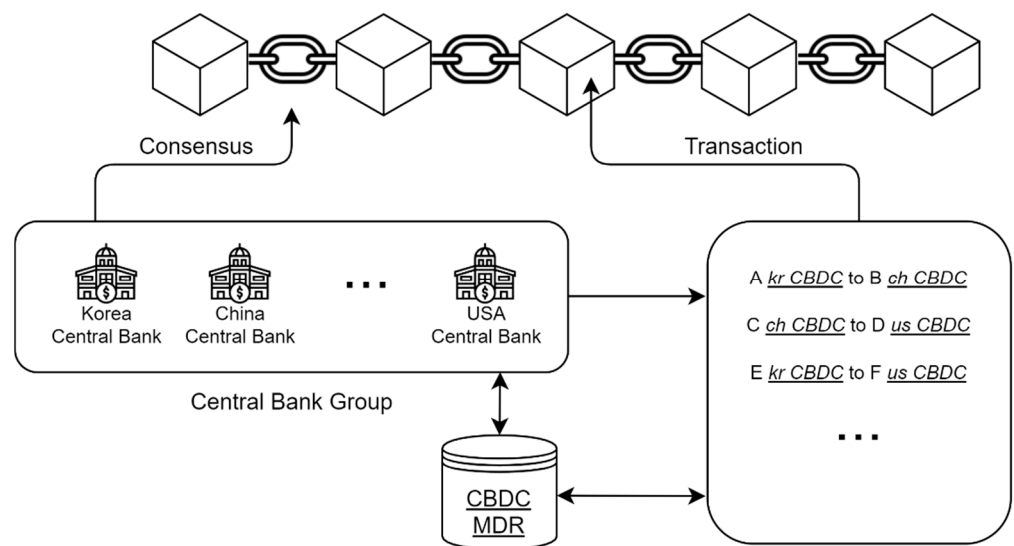
Figure 6. MDR-based CBDC blockchain model.

Figure 7 shows the transaction registration process in the proposed blockchain system. Transactions between CBDCs execute the generation of the block. Smart Contracts are available to ensure the stability of transactions. A Smart Contract is created as a move

request between CBDCs. The Smart Contract waits for the block generation of the transaction, running with block creation. The sending CBDC burns the sending amount, and the receiving CBDC mints the received amount. Eventually, each central bank recognizes the CDBC as a burn and mint transaction.
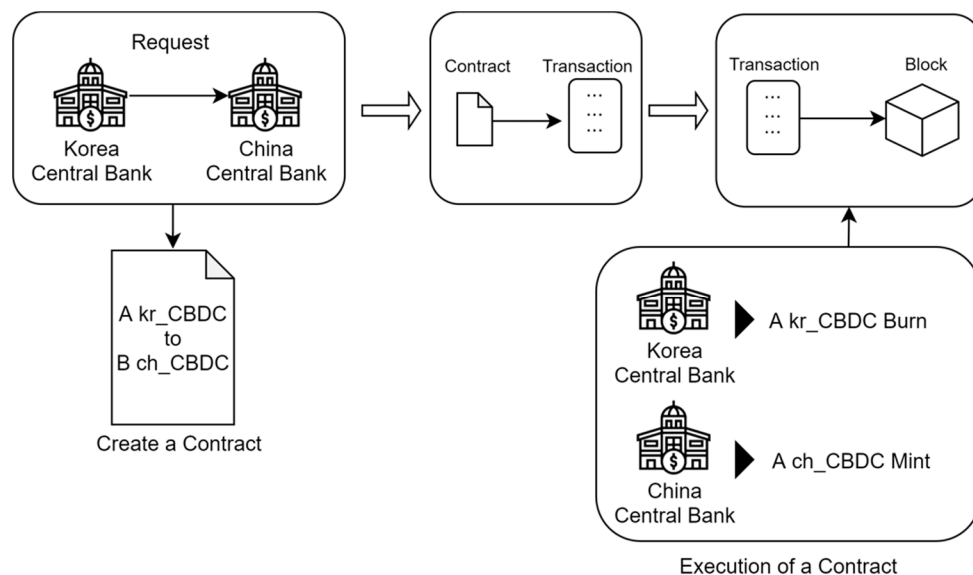
Figure 7. Registration procedure of CBDC.

4. Experiment

In this section, we experiment to verify the proposed blockchain system for exchange between CBDCs. The experiment stores transactions between CBDCs based on MDR in a blockchain (Bitcoin, Ethereum). In the experiment, the blockchain system stores the "A kr_CBDC to B ch_CBDC" transaction in the block. The experiment covers the part on how to record CBDC transactions on the blockchain.

Bitcoin stores arbitrary data using the OP_RETURN script. OP_RETURN can record up to 80 bytes and receive support from Bitcoin Core 0.9 version [21]. This experiment stores CBDC transactions in the OP_RETURN script. Figure 8 shows the creation of an account in a test environment and ten blocks, in which we send the transaction to the account.

```
jun@ubuntu:~/bitcoin_ECC$ bitcoin-cli getnewaddress
37KvDZ3vt5jHrpcD8cJE2kfX2BaDS7BRdP

jun@ubuntu:~/bitcoin_ECC$ bitcoin-cli generatetoaddress 10 37KvDZ3vt5jHrpcD8cJE2kfX2BaDS7BRdP
[
  "5790956a674c8b612f64f2422a9af12f068b81eb3ba838c42d6ff03da1f2b9bf",
  "170456cd42027edd50d6af55ccc285d765457dcf75afcb3e161489054d0622fc",
  "cb8f36284da5c87b5cf1e1dc864c704f3bc0b3c78c1b1bc2f8e79b3e0ad1b913",
  "148987ae6d83fcec210ee2e244c2ef7a7b2a2b17bb23c2b5656ce1f3cbcbf4fb",
  "e1f78815a73680e4ceb8bac9dde57843139201afc904e3fd9669decf27cc4d92",
  "70cc7566b81016b217dfd9fe25f9047540c459490f48fd837695e7dd3f89f69e",
  "cf467a14db4357de3e985ad15228168c4dc1d460d377e17d9cbc95eb287881f5",
  "d94f90889d9bdd0107e5773494c03b7559366ad75015691c5df58be631f51284",
  "4684342d71bef9ce36c45f9fc15cb03e283e4cc8c89ac2112cbfe11f91870497",
  "3cd300d05a73d3b853fb5df917f3896194c2327534930ec0e6ca5961d287931d"
]

jun@ubuntu:~/bitcoin_ECC$ bitcoin-cli getblockchaininfo
{
  "chain": "main",
  "blocks": 10,
  "headers": 10,
  "bestblockhash": "3cd300d05a73d3b853fb5df917f3896194c2327534930ec0e6ca5961d287931d",
  "difficulty": 4.523059468369196e+74,
  "mediantime": 1585536723,
  "verificationprogress": 1,
  "initialblockdownload": false,
  "chainwork": "0000000000000000000000000000000000000000000000000000000000573f8",
```

Figure 8. Create account and block in bitcoin.

Bitcoin stores OP_RETURN as Hex. Figure 9 shows the conversion of the value to Hex to be stored. The data "30 kr_CBDC to 40 ch_CBDC" is stored in the block as Hex "3330206b725f4342444320746f2034302063685f43424443."

Input Text

30 kr_CBDC to 40 ch_CBDC

Convert

Hex Output

3330206b725f4342444320746f2034302063685f43424443

Figure 9. Convert input data to Hex.

Figure 10 shows creating a raw transaction that stores the OP_RETURN value converted to a Hex value. We choose "txid" to make the transaction and create a raw transaction. Additionally, Then, we search the account's UTXO (Unspent transaction output) with the "listunspent" command. Raw transactions are created using the "createrawtransaction" command. The raw transaction is created using the searched txid and Hex.

```
jun@ubuntu:~/bitcoin_ECC$ bitcoin-cli listunspent 0 999999 '["37KvDZ3vt5jHrpcD8cJE2kfX2BaDS7BRdP"]'
[
  {
    "txid": "d176430b7aebd4c02f75681002bc8a1222d3f8d3fbf61c525ed4f75abc72980f",
    "vout": 0,
    "address": "37KvDZ3vt5jHrpcD8cJE2kfX2BaDS7BRdP",
    "label": "",
    "redeemScript": "001418e09509ff9a64016e29d1340433f7582c643b95",
    "scriptPubKey": "a9143dd34179ee139fa6838969ed0169a0bf5c6ece2d87",
    "amount": 50.00000000,
    "confirmations": 5,
    "spendable": true,
    "solvable": true,
    "desc": "sh(wpkh([78178039/0'/0'/0']036c4d3a384e3595b0de67ad8b22db530b2ded96324eabcb3b56d3cbb38279c621))#pnyljreg",
    "safe": true
  },

jun@ubuntu:~/bitcoin_ECC$ bitcoin-cli createrawtransaction '[{"txid":"d176430b7aebd4c02f7568100
2bc8a1222d3f8d3fbf61c525ed4f75abc72980f","vout":0}]' '{"data":"3330206b725f4342444320746f203430
2063685f43424443","37KvDZ3vt5jHrpcD8cJE2kfX2BaDS7BRdP":49.9995}'
02000000010f9872bc5af7d45e521cf6fbd3f8d322128abc021068752fc0d4eb7a0b4376d10000000000ffffffff020
0000000000000001a6a183330206b725f4342444320746f2034302063685f43424443b02e052a0100000017a9143dd3
4179ee139fa6838969ed0169a0bf5c6ece2d8700000000
```

Figure 10. Create raw transaction in bitcoin.

To send the transaction, it must be signed using a private key. Figure 11 shows the transmission transaction. The private key can be obtained using the "dumpprivkey" command, and we use the private key to sign the raw transaction. The signed transaction is sent to the mempool using the "sendrawtransaction" command. If the transmission was successful, a transaction hash is an output.

```
jun@ubuntu:~/bitcoin_ECC$ bitcoin-cli dumpprivkey 37KvDZ3vt5jHrpcD8cJE2kfX2BaDS7BRdP
L1aoMzEuGEZFYvU7QGGwd4jFwGygaNA2DMhHwQek1WkQi5oJNgNV

jun@ubuntu:~/bitcoin_ECC$ bitcoin-cli signrawtransactionwithkey "02000000010f9872bc5af7d45e521c
f6fbd3f8d322128abc021068752fc0d4eb7a0b4376d10000000000ffffffff02000000000000000001a6a183330206b7
25f4342444320746f2034302063685f43424443b02e052a0100000017a9143dd34179ee139fa6838969ed0169a0bf5c
6ece2d8700000000" '["L1aoMzEuGEZFYvU7QGGwd4jFwGygaNA2DMhHwQek1WkQi5oJNgNV "]'
{
  "hex": "020000000001010f9872bc5af7d45e521cf6fbd3f8d322128abc021068752fc0d4eb7a0b4376d10000000
01716001418e09509ff9a64016e29d1340433f7582c643b95ffffffff02000000000000000001a6a183330206b725f43
42444320746f2034302063685f43424443b02e052a0100000017a9143dd34179ee139fa6838969ed0169a0bf5c6ece2
d870247304402204eb6601b4c9a189de5486149h44e529f00adbf24255aeb59e4ffeda53d7516ef02204318b8b8cdd5
2fa0e988f8b1e7ff12f2046af2808c3a983a45b286632acc34ed0121036c4d3a384e3595b0de67ad8b22db530b2ded9
6324eabcb3b56d3cbb38279c62100000000",
  "complete": true
}

jun@ubuntu:~/bitcoin_ECC$ bitcoin-cli sendrawtransaction 020000000001010f9872bc5af7d45e521cf6fbd3f
8d322128abc021068752fc0d4eb7a0b4376d1000000000716001418e09509ff9a64016e29d1340433f7582c643b95fffff
fff02000000000000000001a6a183330206b725f4342444320746f2034302063685f43424443b02e052a0100000017a9143
dd34179ee139fa6838969ed0169a0bf5c6ece2d870247304402204eb6601b4c9a189de5486149h44e529f00adbf24255ae
b59e4ffeda53d7516ef02204318b8b8cdd52fa0e988f8b1e7ff12f2046af2808c3a983a45b286632acc34ed0121036c4d3
a384e3595b0de67ad8b22db530b2ded96324eabcb3b56d3cbb38279c62100000000
6905b4fc7ea3a567729b237b99e6038952df5d231f442a3e5ae35a591db3eec9
```

Figure 11. Sign and Send Raw Transaction in bitcoin.

Figure 12 shows that the insertion of the transaction was sent to mempool. The 11th block is generated using the "generatetoaddress" command. Then, we check the transaction hash using the "getblock" command. Figure 13 shows the decryption of the insert transaction. We can confirm the data inserted into the OP_RETURN value.

```
jun@ubuntu:~/bitcoin_ECC$ bitcoin-cli generatetoaddress 1 37KvDZ3vt5jHrpcD8cJE2kfX2BaDS7BRdP
[
  "ada85616743b95df2abf048278bfa85fae9dd28af67ff58c8053097c46ff6a50"
]
jun@ubuntu:~/bitcoin_ECC$ bitcoin-cli getblock ada85616743b95df2abf048278bfa85fae9dd28af67ff58c8053097c46ff6a50
{
  "hash": "ada85616743b95df2abf048278bfa85fae9dd28af67ff58c8053097c46ff6a50",
  "confirmations": 1,
  "strippedsize": 355,
  "size": 500,
  "weight": 1565,
  "height": 11,
  "version": 536870912,
  "versionHex": "20000000",
  "merkleroot": "73bc6da5ceefa7cd7e3c0860c58ce73c9639d85b5e283a73f6ab24af51b0c8cb",
  "tx": [
    "80d14881b4a8e5a3cf30fd23aa9b741c644a806f8e59a584d13510882550d232",
    "6905b4fc7ea3a567729b237b99e6038952df5d231f442a3e5ae35a591db3eec9"
  ],
  "time": 1585546924,
  "mediantime": 1585536723,
  "nonce": 19,
  "bits": "00000001",
  "difficulty": 4.523059468369196e+74,
  "chainwork": "000000000000000000000000000000000000000000000000000000000005f2e0",
  "nTx": 2,
  "previousblockhash": "3cd300d05a73d3b853fb5df917f3896194c2327534930ec0e6ca5961d287931d"
}
```

Figure 12. Insert transaction to block in bitcoin.

```
jun@ubuntu:~/bitcoin_ECC$ bitcoin-cli decoderawtransaction 020000000001010f9872bc5af7d45e521cf6fbd3f8d322128abc0210687
52fc0d4eb7a0b4376d1000000001716001418e09509ff9a64016e29d1340433f7582c643b95ffffffff0200000000000000001a6a183330206b725
f4342444320746f2034302063685f43424443b02e052a0100000017a9143dd34179ee139fa6838969ed0169a0bf5c6ece2d870247304402204eb66
01b4c9a189de5486149b44e529f00adbf24255aeb59e4ffeda53d7516ef02204318b8b8cdd52fa0e988f8b1e7ff12f2046af2808c3a983a45b2866
32acc34ed0121036c4d3a384e3595b0de67ad8b22db530b2ded96324eabcb3b56d3cbb38279c62100000000
{
  "txid": "6905b4fc7ea3a567729b237b99e6038952df5d231f442a3e5ae35a591db3eec9",
  "hash": "a62330421b8bc91d2e3dcdedeec5334c9c11c844cc89f90043b1d49c3314cd55",
  "version": 2,
  "size": 250,
  "vsize": 169,
  "weight": 673,
  "locktime": 0,
  "vin": [
    {
      "txid": "d176430b7aebd4c02f75681002bc8a1222d3f8d3fbf61c525ed4f75abc72980f",
      "vout": 0,
      "scriptSig": {
        "asm": "001418e09509ff9a64016e29d1340433f7582c643b95",
        "hex": "16001418e09509ff9a64016e29d1340433f7582c643b95"
      },
      "txinwitness": [
        "304402204eb6601b4c9a189de5486149b44e529f00adbf24255aeb59e4ffeda53d7516ef02204318b8b8cdd52fa0e988f8b1e7ff12f20
46af2808c3a983a45b286632acc34ed01",
        "036c4d3a384e3595b0de67ad8b22db530b2ded96324eabcb3b56d3cbb38279c621"
      ],
      "sequence": 4294967295
    }
  ],
  "vout": [
    {
      "value": 0.00000000,
      "n": 0,
      "scriptPubKey": {
        "asm": "OP_RETURN 3330206b725f4342444320746f2034302063685f43424443",
        "hex": "6a183330206b725f4342444320746f2034302063685f43424443",
        "type": "nulldata"
      }
    },
    {
      "value": 49.99950000,
      "n": 1,
      "scriptPubKey": {
        "asm": "OP_HASH160 3dd34179ee139fa6838969ed0169a0bf5c6ece2d OP_EQUAL",
        "hex": "a9143dd34179ee139fa6838969ed0169a0bf5c6ece2d87",
        "reqSigs": 1,
        "type": "scripthash",
        "addresses": [
          "37KvDZ3vt5jHrpcD8cJE2kfX2BaDS7BRdP"
        ]
      }
    }
  ]
}
```

Figure 13. Decode raw transaction in bitcoin.

Ethereum stores arbitrary data using the input of the transaction. Figure 14 shows how to store Hex values in transactions in Ethereum. The input of Ethereum transactions serves the same purpose as Bitcoin's OP_RETURN. We created a transaction that stores the Hex value using the "eth.sendTransaction" command. Ethereum sends the generated transaction to the transaction pool.

```
> eth.sendTransaction({from:eth.accounts[0],to:eth.accounts[1], data:"0x3330206b725f4342444320746f2034302063685f43424443"
, value:'1000000000'})
INFO [04-23|11:44:34.617] Submitted transaction                    fullhash=0xe063d9657a5bdc2653e20a7e5372c4f9f62a8703180
c6b548a051216172c2fe9 recipient=0xC5b7635714F5d39AE6E2713398Dd167E707f2859
"0xe063d9657a5bdc2653e20a7e5372c4f9f62a8703180c6b548a051216172c2fe9"
```

Figure 14. Transaction generation in Ethereum.

Figure 15 shows the transactions stored in the transaction pool of Ethereum. A Transaction pool is a storage place for creating the block. We added two transactions to the transaction pool. The first transaction is a transaction between the general accounts, and. The second transaction is a transaction that Hex data stored in the input item.

```
> eth.pendingTransactions
[{
    blockHash: null,
    blockNumber: null,
    from: "0x1fe23f72d1308f672caf22dc78a7d04b2d8052a1",
    gas: 21000,
    gasPrice: 1000000000,
    hash: "0x8c05ae76fae8a70287b93e1706d9eace0d91bc46566be9d17d964442fe69c46f",
    input: "0x",
    nonce: 0,
    r: "0x5a560a497ed19959d500dadfc2eb30efe3cec289f34ed167076e024910b38aa8",
    s: "0x517a45aa82d4d54fab1a28bc56e76203d7ba46a2f39b7c40e0ed70b13a715cfd",
    to: "0xc5b7635714f5d39ae6e2713398dd167e707f2859",
    transactionIndex: null,
    v: "0x6096",
    value: 5000000000000000000
}, {
    blockHash: null,
    blockNumber: null,
    from: "0x1fe23f72d1308f672caf22dc78a7d04b2d8052a1",
    gas: 22632,
    gasPrice: 1000000000,
    hash: "0xe063d9657a5bdc2653e20a7e5372c4f9f62a8703180c6b548a051216172c2fe9",
    input: "0x3330206b725f4342444320746f2034302063685f43424443",
    nonce: 1,
    r: "0x13f0c68216c68fb28644d0f3ccba1a214950fee7dd129ecd87ce836a46c077a3",
    s: "0x15cbe8c599292c02a623f9ac39110617ab9d6e80be455bfaaba3e26818450090",
    to: "0xc5b7635714f5d39ae6e2713398dd167e707f2859",
    transactionIndex: null,
    v: "0x6095",
    value: 1000000000
}]
```

Figure 15. List of transactions stored in the transaction pool in Ethereum.

Figure 16 shows the contents of the block generation in Ethereum. Our generated transactions are stored in block 4. Figure 17 shows the contents of the transaction in the block. The Hex value was stored in the input item of the transaction.

```
> eth.getBlock(4)
{
  difficulty: 32488,
  extraData: "0xd8830109038467657468886776f312e31302e34856c696e7578",
  gasLimit: 4718380,
  gasUsed: 43632,
  hash: "0xdfc19923783ab8a4ce0806d6e915fe3ca14b3f478f84b497ebff2bd9fd747b76",
  logsBloom: "0x000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000000000000000000000
000000000000000000",
  miner: "0x1fe23f72d1308f672caf22dc78a7d04b2d8052a1",
  mixHash: "0x2cd38be1f272856b0a5a10562ee69005e79cbc9db3ff127df0368b913bd21229",
  nonce: "0x695a92d635901dde",
  number: 4,
  parentHash: "0x9d154d8d8b44b6965a7dc6b0b4d372abcaeb41ba3bec46ff6365fdb10c8762f5",
  receiptsRoot: "0x6482207898df5cd1a953c3e9ee505f2aa4282bdd14fad5c5004fe88114c860ef2",
  sha3Uncles: "0x1dcc4de8dec75d7aab85b567b6ccd41ad312451b948a7413f0a142fd40d49347",
  size: 780,
  stateRoot: "0xd262d0c2073d05dacf3da5244784f215beac962dce11d54456f1d4451d434834",
  timestamp: 1587610173,
  totalDifficulty: 658301,
  transactions: ["0x8c05ae76fae8a70287b93e1706d9eace0d91bc46566be9d17d964442fe69c46f"
, "0xe063d9657a5bdc2653e20a7e5372c4f9f62a8703180c6b548a051216172c2fe9"],
  transactionsRoot: "0xced4836c8a9fbecd5d53d1fee23e691261e9a5fa9a78d60f6a815e54b1654e
5b",
  uncles: []
}
```

Figure 16. Generated block contents in Ethereum.

```
> eth.getTransaction("0xe063d9657a5bdc2653e20a7e5372c4f9f62a8703180c6b548a051216172c2fe9")
{
  blockHash: "0xdfc19923783ab8a4ce0806d6e915fe3ca14b3f478f84b497ebff2bd9fd747b76",
  blockNumber: 4,
  from: "0x1fe23f72d1308f672caf22dc78a7d04b2d8052a1",
  gas: 22632,
  gasPrice: 1000000000,
  hash: "0xe063d9657a5bdc2653e20a7e5372c4f9f62a8703180c6b548a051216172c2fe9",
  input: "0x3330206b725f4342444320746f2034302063685f43424443",
  nonce: 1,
  r: "0x13f0c68216c68fb28644d0f3ccba1a214950fee7dd129ecd87ce836a46c077a3",
  s: "0x15cbe8c599292c02a623f9ac39110617ab9d6e80be455bfaaba3e26818450090",
  to: "0xc5b7635714f5d39ae6e2713398dd167e707f2859",
  transactionIndex: 1,
  v: "0x6095",
  value: 1000000000
}
```

Figure 17. Generated transaction contents in Ethereum.

5. Evaluation

5.1. Comparison of Block Generation Time between Blockchains

This section evaluates Ethereum and Bitcoin applied to the proposed system. The experimental environment is Ubuntu Server 18.04.2 (Bitcoin), linuxmint 19.3 (Ethereum), CPU-Intel® Core™ i5-8600 K CPU @ 3.6 GHz, Ram-64 G, SSD 500 G. In the experiment, blockchains use the ECCPoW Consensus Algorithm [22].

We apply the proposed method to ECCPoW cryptocurrency systems [23]. ECCPoW is a PoW method that uses error correction code. ECCPoW can be personalized by hard forking Bitcoin and Ethereum. ECCPoW has the effect of solving different problems every block for ASIC resistance. ECCPoW uses hamming weights and signs to control difficulty. We use the default generation time (bitcoin 180 s, Ethereum 60 s) of the block because we need to store additional CBDC information in the blockchain.

Figure 18 shows the block generation time of Bitcoin and Ethereum using the proposed method. Figure 18A is a block generation time graph of Bitcoin, and Figure 18B is a block generation time graph of Ethereum. Table 1 shows the target block generation time, difficult change time, average block generation time, and support of user data insertion in blockchains. Bitcoin's block generation time controls the difficulty of solving blocks for every 2016 blocks to occur every 10 min. The block generation time of Ethereum controls the difficulty with the goal of 10 to 20 s. We set the low difficulty of the block and created the block. For support of user data insertion, Bitcoin uses OP_RETURN, while Ethereum uses input. Bitcoin's average block generation is about 538 s per 400 blocks, while Ethereum's block generation time is approximately 4000 blocks per 1183 s, on average. As a result of the experiment, Bitcoin and Ethereum can be created using CBDC information. In the experiment, it is possible to create a stable block using a large number of nodes.
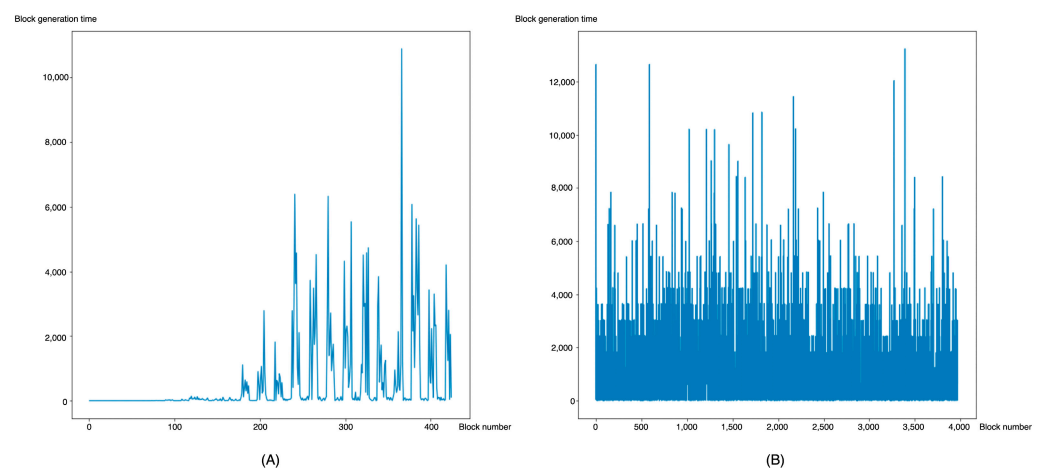
Figure 18. Block generation time in blockchains ((**A**) Bitcoin; (**B**) Ethereum).

Table 1. Block generation time result of each blockchain.

Blockchain Name	Target Block Generation Time	Difficult Change Time	Average Block Generation Time	Support of User Data Insertion
Bitcoin	The 180 s	30 min	538.4070588 s	OP_RETURN
Ethereum	The 60 s	Each block	1183.796214 s	input

5.2. Comparison of Proposed Methods and Exchanges

This section compares the proposed method with the cryptocurrency exchange. Cryptocurrency exchanges have both a centralized and decentralized exchange method. We compare the proposed method with the exchange's technical use of blockchain, Private Key rental, amount of fluidity, transaction liability, transaction target cryptocurrency, and cryptocurrency information management. Table 2 shows the comparison with the proposed method and the exchange.

Table 2. Comparison of proposed methods and related works.

Feature	Proposed Method	Centralization Exchange	Decentralized Exchange
Technical use of blockchain	Yes	No	Yes
Private Key rental	No	Yes	No
Amount of fluidity	Low	High	Low
Transaction liability	Central Banks	Exchange	Unknown or exchange
Transaction target cryptocurrency	CBDC	ALL cryptocurrency	Same Hash function cryptocurrency
Cryptocurrency information management	Yes (open)	Yes (close)	No

The technical use of blockchain uses the proposed method, while the decentralized exchange uses the blockchain to store transactions. Centralized exchanges use centralized storage instead of a blockchain. The use of blockchain technology can determine whether a transaction can proceed without an arbitrator. Decentralized exchanges disclose transaction details and technically guarantee trust. Centralized exchanges keep their transactions private and rely on trust in the exchanges.

Private key rental is not required by the proposed method and distributed exchange. Private key rental is closely related to security. The centralized exchange receives the private key and operates the exchange. In high volumes, centralized exchanges are a security risk. Cryptocurrency security incidents typically occur on exchanges rather than on blockchains.

The amount of fluidity has many centralized exchanges, with few proposal methods and decentralized exchanges. The amount and price of liquidity are challenging to establish a deal at the time you want. Nevertheless, the proposed method of transactions always requires the consensus of the central bank.

Transaction liabilities are all different. Transaction liability is a legal entity that guarantees cryptocurrency. If there is a problem with the transaction, it is responsible for solving the problem and compensation. Centralized exchanges are the responsibility of the exchange. In the event of a problem, the exchange guarantees a settlement. The responsibility for decentralized exchange is not clear, or the decentralized exchange is responsible. The proposed method is the responsibility of the central bank as a transaction between groups. A centralized exchange is responsible for the transaction. The decentralized exchange technically guarantees the reliability of the transaction. However, if there is a problem with the transaction, the liability may be unclear.

Transaction target cryptocurrency is as follows. The proposed method can deal with CDBC, decentralized exchanges with the same hash function cryptocurrency, and centralized exchanges with all cryptocurrencies. Decentralized exchanges are possible when they have the same hash function. Examples of this include ERC-20 Coin and Bitcoin Hard Fork Cryptocurrency.

Cryptocurrency information management is possible with the proposed method and centralized exchange. Decentralized exchanges can only trade cryptocurrencies that are initially set. The proposed method is managed by the central bank's public groups based on the MDR. The centralized exchange can further manage the cryptocurrency allowed by the exchange.

6. Conclusions

This paper proposes an inter-CBDC exchange system based on ISO/IEC 11179 in preparation for the introduction of CBDC by central banks around the world. CBDC would be difficult to implement and supply in one standard way, managed by a central bank or equivalent (for example, mCBDC Model 3 of BIS). This paper proposes a blockchain system that applies ISO/IEC 1179's approach to data interoperability. The system stores and manages CBDCs from central banks that want to be exchanged according to metadata registry elements.

CBDC has officially only released Sand Dollar. Therefore, we hard forked and applied blockchains (Bitcoin, Ethereum) that are conceptually functioning well at the moment. The experiment assumes that CBDC Data Elements generated by ISO/IEC 11179 is stored. We used space that allowed users of Bitcoin and Ethereum to use. The proposed theory could be applied to existing cryptocurrency systems. However, its application to existing cryptocurrency systems has structural limitations in which CBDC Data Element is added to free space. In practice, blockchain structures will be needed to efficiently manage CBDCs implemented in multiple ways.

In the face of the growing need for CBDCs by central banks, this paper presents a method of sharing among CBDCs. According to the BIS, more than 84% of central banks are involved in CBDC-related businesses, indicating that other CBDCs will be issued soon. Therefore, the need for interoperability will become more important for the success of CBDC.

References

1. Nakamoto, S. Bitcoin: A Peer-to-Peer Electronic Cash System. Manubot, November 2019. Available online: https://git.dhimmel.com/bitcoin-whitepaper/ (accessed on 30 April 2021).
2. Nofer, M.; Gomber, P.; Hinz, O.; Schiereck, D. Blockchain. *Bus. Inf. Syst. Eng.* **2017**, *59*, 183–187. [CrossRef]
3. Saberi, S.; Kouhizadeh, M.; Sarkis, J.; Shen, L. Blockchain technology and its relationships to sustainable supply chain management. *Int. J. Prod. Res.* **2019**, *57*. Available online: https://www.tandfonline.com/doi/abs/10.1080/00207543.2018.1533261 (accessed on 30 April 2021). [CrossRef]
4. Ethereum. Available online: https://ethereum.org/ (accessed on 12 May 2021).
5. Wood, G. ETHEREUM: A Secure Decentralised Generalised Sransaction Byzantium Version, Yellow Paper. 2014. Available online: ethereum.github.io/yellowpaper/paper.pdf (accessed on 30 April 2021).
6. Zhou, Q.; Huang, H.; Zheng, Z.; Bian, J. Solutions to Scalability of Blockchain: A Survey. *IEEE Access* **2020**, *8*, 16440–16455. [CrossRef]
7. Ripple. Available online: https://ripple.com/ (accessed on 12 May 2021).
8. Kumhof, M.; Noone, C. *Central Bank Digital Currencies—Design Principles and Balance Sheet Implications*; SSRN Scholarly Paper ID 3180713; Social Science Research Network: Rochester, NY, USA, 2018. [CrossRef]

9.   Central Bank Digital Currencies. March 2018. Available online: https://www.bis.org/cpmi/publ/d174.htm (accessed on 30 April 2021).
10.   White Paper Diem Association. Available online: https://www.diem.com/en-us/white-paper/ (accessed on 30 April 2021).
11.   First Look: China's Central Bank Digital Currency, Binance Research. 27 August 2019. Available online: https://research.binance.com/en/analysis/china-cbdc (accessed on 30 April 2021).
12.   kita. Available online: http://stat.kita.net/stat/world/trade/CtrImpExpList.screen (accessed on 12 May 2021).
13.   Boar, C.; Holden, H.; Wadsworth, A. *Impending Arrival—A Sequel to the Survey on Central Bank Digital Currency*; SSRN Scholarly Paper ID 3535896; Social Science Research Network: Rochester, NY, USA, 2020. Available online: https://papers.ssrn.com/abstract=3535896 (accessed on 30 April 2021).
14.   Sand Dollar. Available online: https://www.sanddollar.bs/ (accessed on 12 May 2021).
15.   Bakong. Available online: https://bakong.nbc.org.kh/en/ (accessed on 12 May 2021).
16.   Multiple CBDC(mCBDC) Bridge. Available online: https://www.bis.org/about/bisih/topics/cbdc/mcbdc_bridge.htm (accessed on 12 May 2021).
17.   Auer, R.; Haene, P.; Holden, H. Multi-CBDC Arrangements and the Future of Cross-Border Payments. no. 115, p. 23. Available online: bis.org/publ/bppdf/bispap115.htm (accessed on 12 May 2021).
18.   Resource Description Framework (RDF). Available online: https://www.w3.org/RDF/ (accessed on 12 May 2021).
19.   Web Ontology Language (OWL). Available online: ttps://www.w3.org/2001/sw/wiki/OWL (accessed on 12 May 2021).
20.   ISO/IEC 11179-1:2015. ISO. Available online: https://www.iso.org/cms/render/live/en/sites/isoorg/contents/data/standard/06/19/61932.html (accessed on 30 April 2021).
21.   Bitcoin—OP_RETURN. Available online: https://bitcoin.org/en/release/v0.9.0#opreturn-and-data-in-the-block-chain (accessed on 30 April 2021).
22.   ECCPoW. Available online: https://github.com/cryptoecc/bitcoin_ECC (accessed on 12 May 2021).
23.   Jung, H.; Lee, H.-N. ECCPoW: Error-Correction Code based Proof-of-Work for ASIC Resistance. *Symmetry* **2020**, *12*, 988. [CrossRef]