Lab Report: Python-based Smart Contract

Authors: Muhammad Hamza & Emmanuel
Course: Blockchain and Cryptocurrency
Lecturer/Instructor: Dr. Shahbaz Siddiqui

1. Introduction

This lab task demonstrates the setup of a permissioned MultiChain network that maintains a global ledger recording only transaction summaries, while each node (wallet address) retains its private ledger of relevant transactions. The objective is to show how to configure a MultiChain environment, create wallet addresses, issue and transfer an asset, and record transaction summaries in a dedicated stream for audit purposes.

2. Objectives

- **Set Up a MultiChain Network:**
  Create a blockchain environment with MultiChain CLI commands.

- **Wallet Management:**
  Create and manage wallet addresses.

- **Asset Management:**
  Issue an asset and perform transactions between addresses.

- **Global Ledger Implementation:**
  Use MultiChain streams to maintain a global ledger that records transaction summaries.

- **Private Ledger Maintenance:**
  Filter transactions from the global ledger to display each address's private transaction history.

- **Automation via Python:**
  Develop a Python script that automates the entire process and interacts with the MultiChain RPC API.

3. Methodology

3.1 MultiChain Environment Setup

Chain Creation:

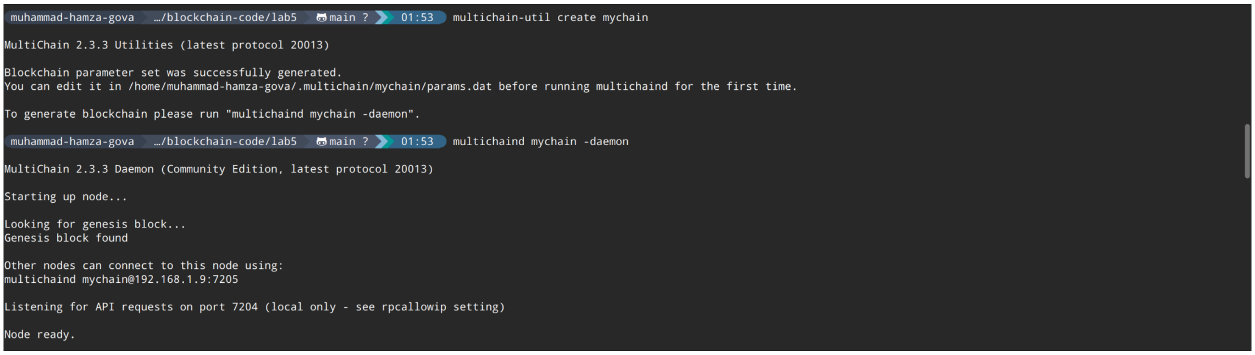
The chain is created using:

```
multichain-util create mychain
```

and then started in daemon mode:

```
multichaind mychain -daemon
```

- *[Screenshot 1: Output of chain creation and daemon startup]*



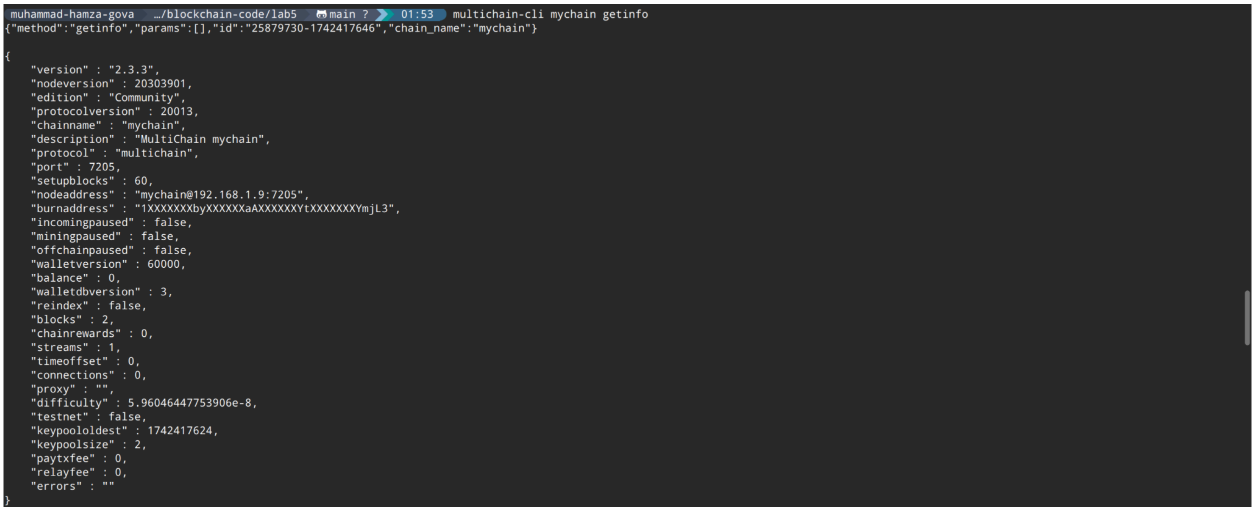
```
muhammad-hamza-gova  …/blockchain-code/lab5  main ?  01:53  multichain-util create mychain

MultiChain 2.3.3 Utilities (latest protocol 20013)

Blockchain parameter set was successfully generated.
You can edit it in /home/muhammad-hamza-gova/.multichain/mychain/params.dat before running multichaind for the first time.

To generate blockchain please run "multichaind mychain -daemon".

muhammad-hamza-gova  …/blockchain-code/lab5  main ?  01:53  multichaind mychain -daemon

MultiChain 2.3.3 Daemon (Community Edition, latest protocol 20013)

Starting up node...

Looking for genesis block...
Genesis block found

Other nodes can connect to this node using:
multichaind mychain@192.168.1.9:7205

Listening for API requests on port 7204 (local only - see rpcallowip setting)

Node ready.
```

Chain Verification:

Basic chain information is obtained using:

```
multichain-cli mychain getinfo
```

- *[Screenshot 2: getinfo output]*



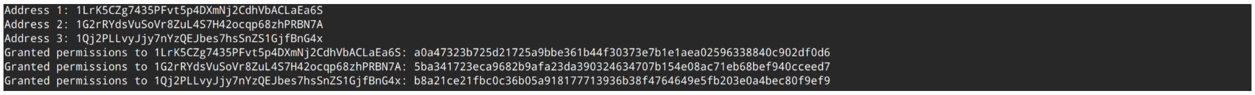
```
muhammad-hamza-gova  …/blockchain-code/lab5  main ?  01:53  multichain-cli mychain getinfo
{"method":"getinfo","params":[],"id":"25879730-1742417646","chain_name":"mychain"}

{
    "version" : "2.3.3",
    "nodeversion" : 20303901,
    "edition" : "Community",
    "protocolversion" : 20013,
    "chainname" : "mychain",
    "description" : "MultiChain mychain",
    "protocol" : "multichain",
    "port" : 7205,
    "setupblocks" : 60,
    "nodeaddress" : "mychain@192.168.1.9:7205",
    "burnaddress" : "1XXXXXXXbyXXXXXXaAXXXXXXXYtXXXXXXXYmjL3",
    "incomingpaused" : false,
    "miningpaused" : false,
    "offchainpaused" : false,
    "walletversion" : 60000,
    "balance" : 0,
    "walletdbversion" : 3,
    "reindex" : false,
    "blocks" : 2,
    "chainrewards" : 0,
    "streams" : 1,
    "timeoffset" : 0,
    "connections" : 0,
    "proxy" : "",
    "difficulty" : 5.96046447753906e-8,
    "testnet" : false,
    "keypoololdest" : 1742417624,
    "keypoolsize" : 2,
    "paytxfee" : 0,
    "relayfee" : 0,
    "errors" : ""
}
```

3.2 Wallet Address and Permission Management

- **Address Generation:**
  Three new wallet addresses are generated using the MultiChain RPC command `getnewaddress`.

Granting Permissions:
Each address is granted the required `send,receive` permissions via:
`multichain-cli mychain grant <address> send,receive`

- *[Screenshot 3: Output showing permission grants]*



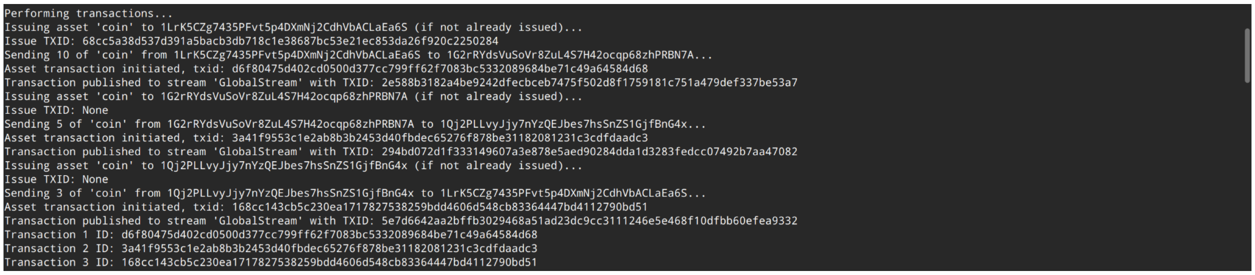
```
Address 1: 1LrK5CZg7435PFvt5p4DXmNj2CdhVbACLaEa6S
Address 2: 1G2rRYdsVuSoVr8ZuL4S7H42ocqp68zhPRBN7A
Address 3: 1Qj2PLLvyJjy7nYzQEJbes7hsSnZS1GjfBnG4x
Granted permissions to 1LrK5CZg7435PFvt5p4DXmNj2CdhVbACLaEa6S: a0a47323b725d21725a9bbe361b44f30373e7b1e1aea02596338840c902df0d6
Granted permissions to 1G2rRYdsVuSoVr8ZuL4S7H42ocqp68zhPRBN7A: 5ba341723eca9682b9afa23da390324634707b154e08ac71eb68bef940cceed7
Granted permissions to 1Qj2PLLvyJjy7nYzQEJbes7hsSnZS1GjfBnG4x: b8a21ce21fbc0c36b05a918177713936b38f4764649e5fb203e0a4bec80f9ef9
```

3.3 Global Ledger Creation and Transaction Processing

- **Stream Creation:**
  A dedicated stream named "GlobalStream" is created for recording transaction summaries. This stream is later subscribed to by all nodes.

- **Asset Issuance and Transactions:**
  For demonstration, an asset named "coin" is issued to the sender address and then transferred between the addresses using `sendassetfrom`. Each transaction's summary (including TXID, sender, receiver, amount, timestamp) is published to the stream.

  [Screenshot 4: Script output showing asset issuance and transaction details]



```
Performing transactions...
Issuing asset 'coin' to 1LrK5CZg7435PFvt5p4DXmNj2CdhVbACLaEa6S (if not already issued)...
Issue TXID: 68cc5a38d537d391a5bacb3db718c1e38687bc53e21ec853da26f920c2250284
Sending 10 of 'coin' from 1LrK5CZg7435PFvt5p4DXmNj2CdhVbACLaEa6S to 1G2rRYdsVuSoVr8ZuL4S7H42ocqp68zhPRBN7A...
Asset transaction initiated, txid: d6f80475d402cd0500d377cc799ff62f7083bc5332089684be71c49a64584d68
Transaction published to stream 'GlobalStream' with TXID: 2e588b3182a4be9242dfecbceb7475f502d8f1759181c751a479def337be53a7
Issuing asset 'coin' to 1G2rRYdsVuSoVr8ZuL4S7H42ocqp68zhPRBN7A (if not already issued)...
Issue TXID: None
Sending 5 of 'coin' from 1G2rRYdsVuSoVr8ZuL4S7H42ocqp68zhPRBN7A to 1Qj2PLLvyJjy7nYzQEJbes7hsSnZS1GjfBnG4x...
Asset transaction initiated, txid: 3a41f9553c1e2ab8b3b2453d40fbdec65276f878be31182081231c3cdfdaadc3
Transaction published to stream 'GlobalStream' with TXID: 294bd072d1f333149607a3e878e5aed90284dda1d3283fedcc07492b7aa47082
Issuing asset 'coin' to 1Qj2PLLvyJjy7nYzQEJbes7hsSnZS1GjfBnG4x (if not already issued)...
Issue TXID: None
Sending 3 of 'coin' from 1Qj2PLLvyJjy7nYzQEJbes7hsSnZS1GjfBnG4x to 1LrK5CZg7435PFvt5p4DXmNj2CdhVbACLaEa6S...
Asset transaction initiated, txid: 168cc143cb5c230ea1717827538259bdd4606d548cb83364447bd4112790bd51
Transaction published to stream 'GlobalStream' with TXID: 5e7d6642aa2bffb3029468a51ad23dc9cc3111246e5e468f10dfbb60efea9332
Transaction 1 ID: d6f80475d402cd0500d377cc799ff62f7083bc5332089684be71c49a64584d68
Transaction 2 ID: 3a41f9553c1e2ab8b3b2453d40fbdec65276f878be31182081231c3cdfdaadc3
Transaction 3 ID: 168cc143cb5c230ea1717827538259bdd4606d548cb83364447bd4112790bd51
```

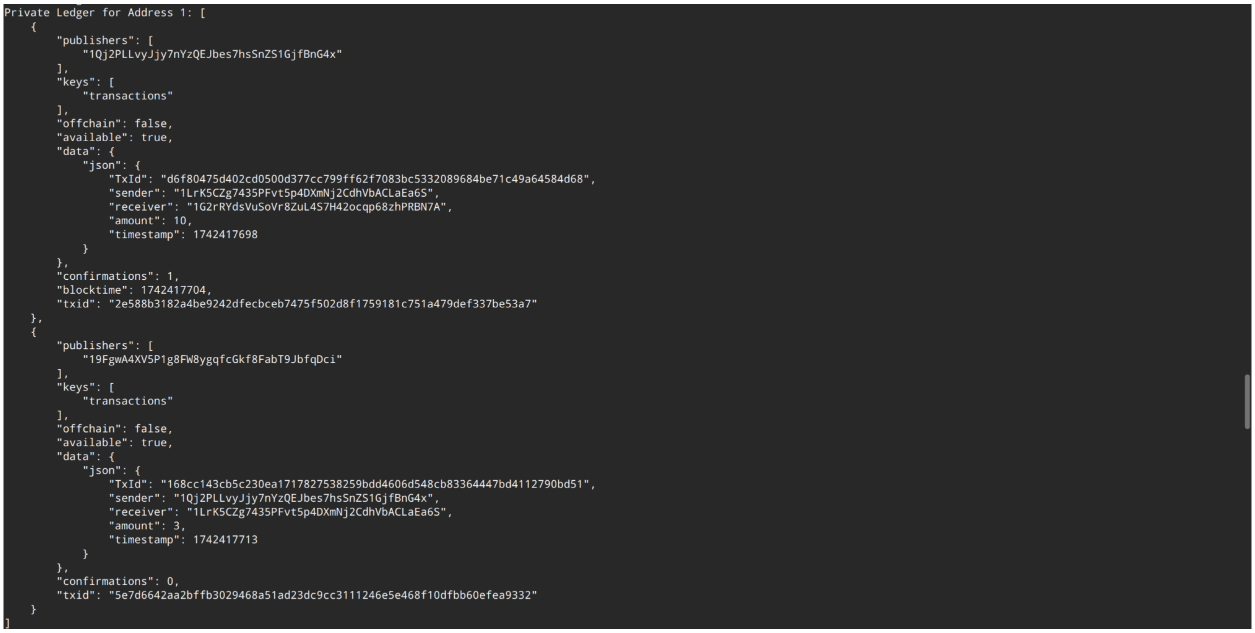
3.4 Ledger Query Functions

- **Private Ledger Extraction:**
  The global stream is queried and then filtered by address (either as sender or receiver)

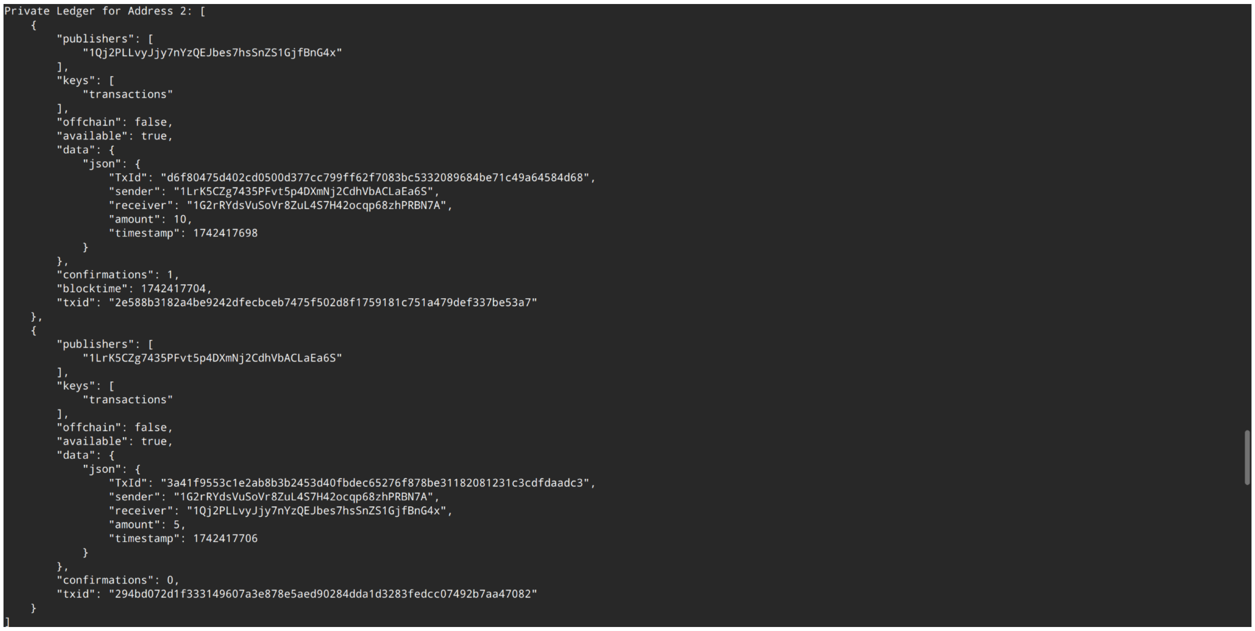
to simulate a private ledger.

- **Global Ledger Overview:**
  A function retrieves all transaction summaries from the global ledger stream.

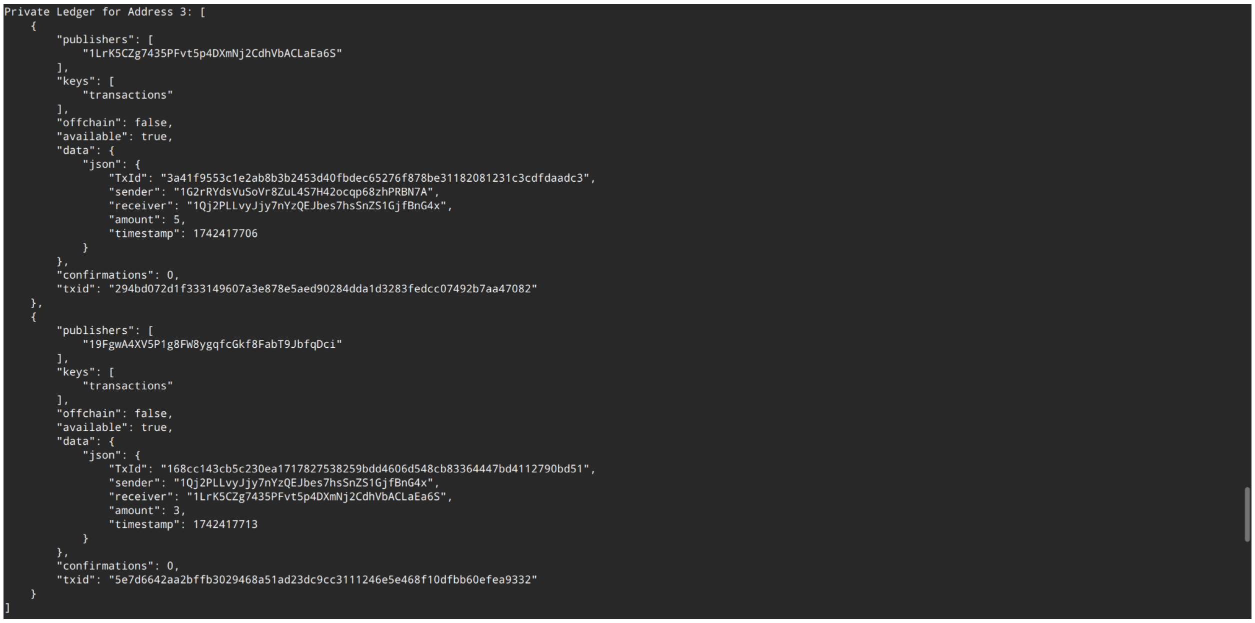
[Screenshot 5: Output of private and global ledger queries]



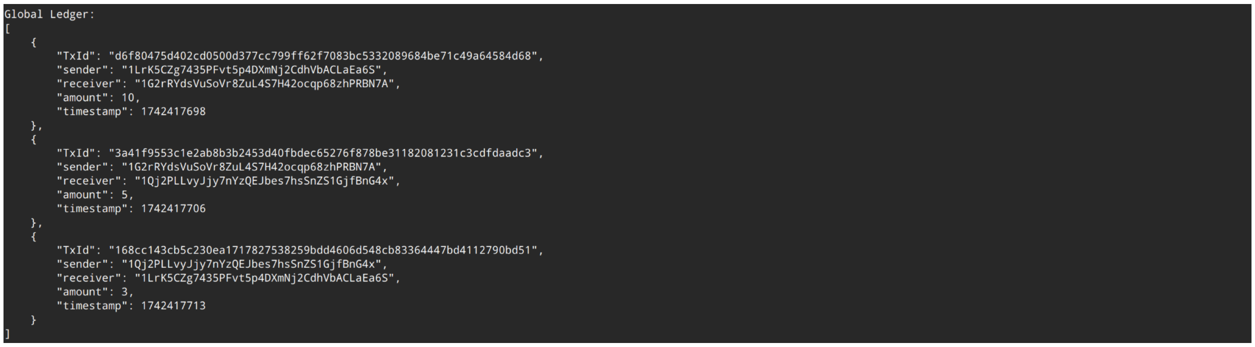
```
Private Ledger for Address 1: [
    {
        "publishers": [
            "1Qj2PLLvyJjy7nYzQEJbes7hsSnZS1GjfBnG4x"
        ],
        "keys": [
            "transactions"
        ],
        "offchain": false,
        "available": true,
        "data": {
            "json": {
                "TxId": "d6f80475d402cd0500d377cc799ff62f7083bc5332089684be71c49a64584d68",
                "sender": "1LrK5CZg7435PFvt5p4DXmNj2CdhVbACLaEa6S",
                "receiver": "1G2rRYdsVuSoVr8ZuL4S7H42ocqp68zhPRBN7A",
                "amount": 10,
                "timestamp": 1742417698
            }
        },
        "confirmations": 1,
        "blocktime": 1742417704,
        "txid": "2e588b3182a4be9242dfecbceb7475f502d8f1759181c751a479def337be53a7"
    },
    {
        "publishers": [
            "19FgwA4XV5P1g8FW8ygqfcGkf8FabT9JbfqDci"
        ],
        "keys": [
            "transactions"
        ],
        "offchain": false,
        "available": true,
        "data": {
            "json": {
                "TxId": "168cc143cb5c230ea1717827538259bdd4606d548cb83364447bd4112790bd51",
                "sender": "1Qj2PLLvyJjy7nYzQEJbes7hsSnZS1GjfBnG4x",
                "receiver": "1LrK5CZg7435PFvt5p4DXmNj2CdhVbACLaEa6S",
                "amount": 3,
                "timestamp": 1742417713
            }
        },
        "confirmations": 0,
        "txid": "5e7d6642aa2bffb3029468a51ad23dc9cc3111246e5e468f10dfbb60efea9332"
    }
]
```



```
Private Ledger for Address 2: [
    {
        "publishers": [
            "1Qj2PLLvyJjy7nYzQEJbes7hsSnZS1GjfBnG4x"
        ],
        "keys": [
            "transactions"
        ],
        "offchain": false,
        "available": true,
        "data": {
            "json": {
                "TxId": "d6f80475d402cd0500d377cc799ff62f7083bc5332089684be71c49a64584d68",
                "sender": "1LrK5CZg7435PFvt5p4DXmNj2CdhVbACLaEa6S",
                "receiver": "1G2rRYdsVuSoVr8ZuL4S7H42ocqp68zhPRBN7A",
                "amount": 10,
                "timestamp": 1742417698
            }
        },
        "confirmations": 1,
        "blocktime": 1742417704,
        "txid": "2e588b3182a4be9242dfecbceb7475f502d8f1759181c751a479def337be53a7"
    },
    {
        "publishers": [
            "1LrK5CZg7435PFvt5p4DXmNj2CdhVbACLaEa6S"
        ],
        "keys": [
            "transactions"
        ],
        "offchain": false,
        "available": true,
        "data": {
            "json": {
                "TxId": "3a41f9553c1e2ab8b3b2453d40fbdec65276f878be31182081231c3cdfdaadc3",
                "sender": "1G2rRYdsVuSoVr8ZuL4S7H42ocqp68zhPRBN7A",
                "receiver": "1Qj2PLLvyJjy7nYzQEJbes7hsSnZS1GjfBnG4x",
                "amount": 5,
                "timestamp": 1742417706
            }
        },
        "confirmations": 0,
        "txid": "294bd072d1f333149607a3e878e5aed90284dda1d3283fedcc07492b7aa47082"
    }
]
```



```
Private Ledger for Address 3: [
    {
        "publishers": [
            "1LrK5CZg7435PFvt5p4DXmNj2CdhVbACLaEa6S"
        ],
        "keys": [
            "transactions"
        ],
        "offchain": false,
        "available": true,
        "data": {
            "json": {
                "TxId": "3a41f9553c1e2ab8b3b2453d40fbdec65276f878be31182081231c3cdfdaadc3",
                "sender": "1G2rRYdsVuSoVr8ZuL4S7H42ocqp68zhPRBN7A",
                "receiver": "1Qj2PLLvyJjy7nYzQEJbes7hsSnZS1GjfBnG4x",
                "amount": 5,
                "timestamp": 1742417706
            }
        },
        "confirmations": 0,
        "txid": "294bd072d1f333149607a3e878e5aed90284dda1d3283fedcc07492b7aa47082"
    },
    {
        "publishers": [
            "19FgwA4XV5P1g8FW8ygqfcGkf8FabT9JbfqDci"
        ],
        "keys": [
            "transactions"
        ],
        "offchain": false,
        "available": true,
        "data": {
            "json": {
                "TxId": "168cc143cb5c230ea1717827538259bdd4606d548cb83364447bd4112790bd51",
                "sender": "1Qj2PLLvyJjy7nYzQEJbes7hsSnZS1GjfBnG4x",
                "receiver": "1LrK5CZg7435PFvt5p4DXmNj2CdhVbACLaEa6S",
                "amount": 3,
                "timestamp": 1742417713
            }
        },
        "confirmations": 0,
        "txid": "5e7d6642aa2bffb3029468a51ad23dc9cc3111246e5e468f10dfbb60efea9332"
    }
]
```



```
Global Ledger:
[
    {
        "TxId": "d6f80475d402cd0500d377cc799ff62f7083bc5332089684be71c49a64584d68",
        "sender": "1LrK5CZg7435PFvt5p4DXmNj2CdhVbACLaEa6S",
        "receiver": "1G2rRYdsVuSoVr8ZuL4S7H42ocqp68zhPRBN7A",
        "amount": 10,
        "timestamp": 1742417698
    },
    {
        "TxId": "3a41f9553c1e2ab8b3b2453d40fbdec65276f878be31182081231c3cdfdaadc3",
        "sender": "1G2rRYdsVuSoVr8ZuL4S7H42ocqp68zhPRBN7A",
        "receiver": "1Qj2PLLvyJjy7nYzQEJbes7hsSnZS1GjfBnG4x",
        "amount": 5,
        "timestamp": 1742417706
    },
    {
        "TxId": "168cc143cb5c230ea1717827538259bdd4606d548cb83364447bd4112790bd51",
        "sender": "1Qj2PLLvyJjy7nYzQEJbes7hsSnZS1GjfBnG4x",
        "receiver": "1LrK5CZg7435PFvt5p4DXmNj2CdhVbACLaEa6S",
        "amount": 3,
        "timestamp": 1742417713
    }
]
```

3.5 Automation with Python

- **Script Overview:**
  A Python script (provided below) orchestrates all the above steps:
  - It connects to the MultiChain node.
  - Creates the global ledger stream.
  - Generates three wallet addresses.
  - Issues and transfers assets.
  - Publishes transaction summaries.
  - Queries and prints both private and global ledgers.

4. Experimental Setup

- **Software:**
    - MultiChain 2.3.3 (Community Edition)
    - Python 3.x
    - Custom Python library (`multichain.py`) for RPC integration
- **Configuration Details:**
    - RPC Host: 127.0.0.1
    - RPC Port: 7444
    - Chain Name: "mychain"
    - RPC Credentials:
        - Username: `multichainrpc`
        - Password: *[Provided in the script]*

5. Results

After executing the Python script, the following observations were made:

- **Chain Info and Wallet Addresses:**
  The script successfully retrieved and printed chain information and the existing wallet address, followed by the creation of three new addresses.

- **Stream Creation:**
  A stream named "GlobalStream" was created and subscribed to.

- **Asset Transactions:**
  Each asset transaction failed initially due to missing permissions on the destination addresses. Once permissions were granted, subsequent transactions succeeded.
  Note: In our final script, permissions are automatically granted for new addresses.

- **Ledger Outputs:**
  The global ledger correctly displays all transaction summaries, while each private ledger (filtered by address) shows only the transactions where the address was the sender or receiver.

6. Discussion

- **Permission Handling:**
  Granting `send,receive` permissions was crucial for successful asset transfers. Without these, transactions were rejected with error code -704.

- **Stream as a Global Ledger:**
  The use of a dedicated stream for recording transaction summaries provides an efficient audit trail while keeping full transaction data private to the respective nodes.

- **Automation via Python:**
  The script successfully automates the process. It also includes error handling for RPC failures and outputs the necessary details for verification.

- **Scalability:**
  Although the asset issuance is repeated in each transaction for demo purposes, in a production environment, assets would be issued once, and only transfer operations would occur thereafter.

7. Conclusion

This lab successfully demonstrates a MultiChain network configured to maintain both a global ledger of transaction summaries and private ledgers per wallet address. The integration of CLI commands for setup and a Python script for automation provides a robust framework for asset management and auditability on a permissioned blockchain. This approach enhances privacy and auditability in a distributed ledger environment.