**Title**: **Project 2: Dash public blockchain and cross-chain interoperability part 1**

**Student Name:** Kenji Mah

**ASU ID:** 1219012444

**Abstract:**

In this project we will be using Dash, a public blockchain ecosystem to write and execute cryptocurrency-based financial transactions. We will be transferring funds using from one user to another in Dash’s testing blockchain.

**Keywords:**

Unspent Transaction Output (UTXO), Wallets, Pubkey script, tokens, address

**Introduction:**

The last couple of years have brought increase in popularity of blockchain technology with numerous projects being implemented by private and public entities. It is however an emerging technology and quickly evolving technology too. By creating a decentralized system, it removes the indulgence of central servers and provides peer-to-peer interaction. It can create a fully transparent and open to all database, which could bring transparency to the governance and elections.

**Terminology:**

Unspent Transaction Output (UTXO):

An output in a transaction which contains two fields: a value field for transferring zero or more duffs (denominations of dash value) and a pubkey script for indicating what conditions must be fulfilled for those duffs to be further spent.[[1]](#footnote-1)

Wallets:

Blockchain wallets store and manage keys used to interact with the blockchain. Because transactions need a digital signature to spend an output, which requires knowledge of a private key, blockchain wallets are one of the most important applications used with a blockchain. The secure storage and management of keys is crucial for most transactions.

Pubkey script:

A script included in outputs which sets the conditions that must be fulfilled for those duffs to be spent. Data for fulfilling the conditions can be provided in a signature script. Pubkey Scripts are called a scriptPubKey in code.[[2]](#footnote-2)

Tokens:

The ChainRider service uses web tokens for authentication. [[3]](#footnote-3) This is provided to us so we can access the test network.

Address:

A 20-byte hash formatted using base58check to produce either a P2PKH or P2SH Dash address. Currently the most common way users exchange payment information. [[4]](#footnote-4)

**Goal Description**:

**Phase I:** Writing a financial transaction to send some Dash between two wallets

1. Check if the sender wallet has enough UTXO to send.
2. Create a transaction from the sender to the receiver
3. Broadcast the transaction to the blockchain.

**Description of proposed solution:**

Our implementation of the solution will be coded in Node.js while using libraries such as DashCore and Got. Got is used for sending and receiving https request while DashCore is used for creating a Transaction object. We will be working with chainRider’s api for checking the wallets and sending the raw transaction to the blockchain.

Phase I:

Firstly, we must check if the sender has enough currency to send to the receiver. In our case we are trying to send 20,000 units over. We will be using the transaction by hash[[5]](#footnote-5) function from the chainrider api to receive all the UTXO’s that the sender has. Because there may be many UTXO’s with different denominations, we must total all of them to make sure there is enough to transfer. It is important to note that there are typically transactions fees that are taken in addition to the send amount, however because the testnet has an excessive amount of funds in it, we don’t have to worry about it for the application of this project.

After it is verified that the sender has enough funds to give, we must set up a transaction object because one of the parameters for the chainRider api for sending raw transactions[[6]](#footnote-6) takes in the serialized hexadecimal ASCII values of the raw transaction. Luckily, the DashCore library has a variety of functions for creating a raw transaction[[7]](#footnote-7) and serializing it. We will be using a simple transaction that requires the UTXO’s, receiving address, sending address, and privatekey as parameters. The function considers the fees required for the process, sending the change back to the owner, the Pubkey Scripts, and signing the transaction when the function is called so we do not have to worry about those variables.

The serialized transaction and the given token can then be used as parameters for sending the raw transaction with chainRider. The resulting request will output the Transaction Id that we can then use to verify that it was added to the blockchain through the auto grader.

**Issues Faced and Methods used to resolve them:**

There were not too many issues while implementing this project because of its simplicity, however there were some nuances that occurred occasionally. The api was not always working and would occasionally fail a request, however after some patience it would come back and perform normally. I did get stuck occasionally when formatting of the inputs of the parameters, however reading the documentation of the api’s and libraries, and reading error messages thoroughly help me progress through the problems efficiently.

**Conclusions**:

In conclusion, I learned how to use chainRider to send money from one user to another on the Dash blockchain network. I also learned more about http requests and how to utilize them in Node.js applications.

**Bibliography**:

<https://dashcore.readme.io/docs/core-guide-transactions>

<https://dashcore.readme.io/docs/core-additional-resources-glossary#pubkey-script>

<https://www.chainrider.io/docs/dash/#token-usage>

<https://dashcore.readme.io/docs/core-additional-resources-glossary#address>

<https://www.chainrider.io/docs/dash/#transaction-by-hash>

<https://www.chainrider.io/docs/dash/#send-raw-transaction>

<https://github.com/dashevo/dashcore-lib/blob/master/docs/transaction.md>

1. https://dashcore.readme.io/docs/core-guide-transactions [↑](#footnote-ref-1)
2. https://dashcore.readme.io/docs/core-additional-resources-glossary#pubkey-script [↑](#footnote-ref-2)
3. https://www.chainrider.io/docs/dash/#token-usage [↑](#footnote-ref-3)
4. https://dashcore.readme.io/docs/core-additional-resources-glossary#address [↑](#footnote-ref-4)
5. https://www.chainrider.io/docs/dash/#transaction-by-hash [↑](#footnote-ref-5)
6. https://www.chainrider.io/docs/dash/#send-raw-transaction [↑](#footnote-ref-6)
7. https://github.com/dashevo/dashcore-lib/blob/master/docs/transaction.md [↑](#footnote-ref-7)