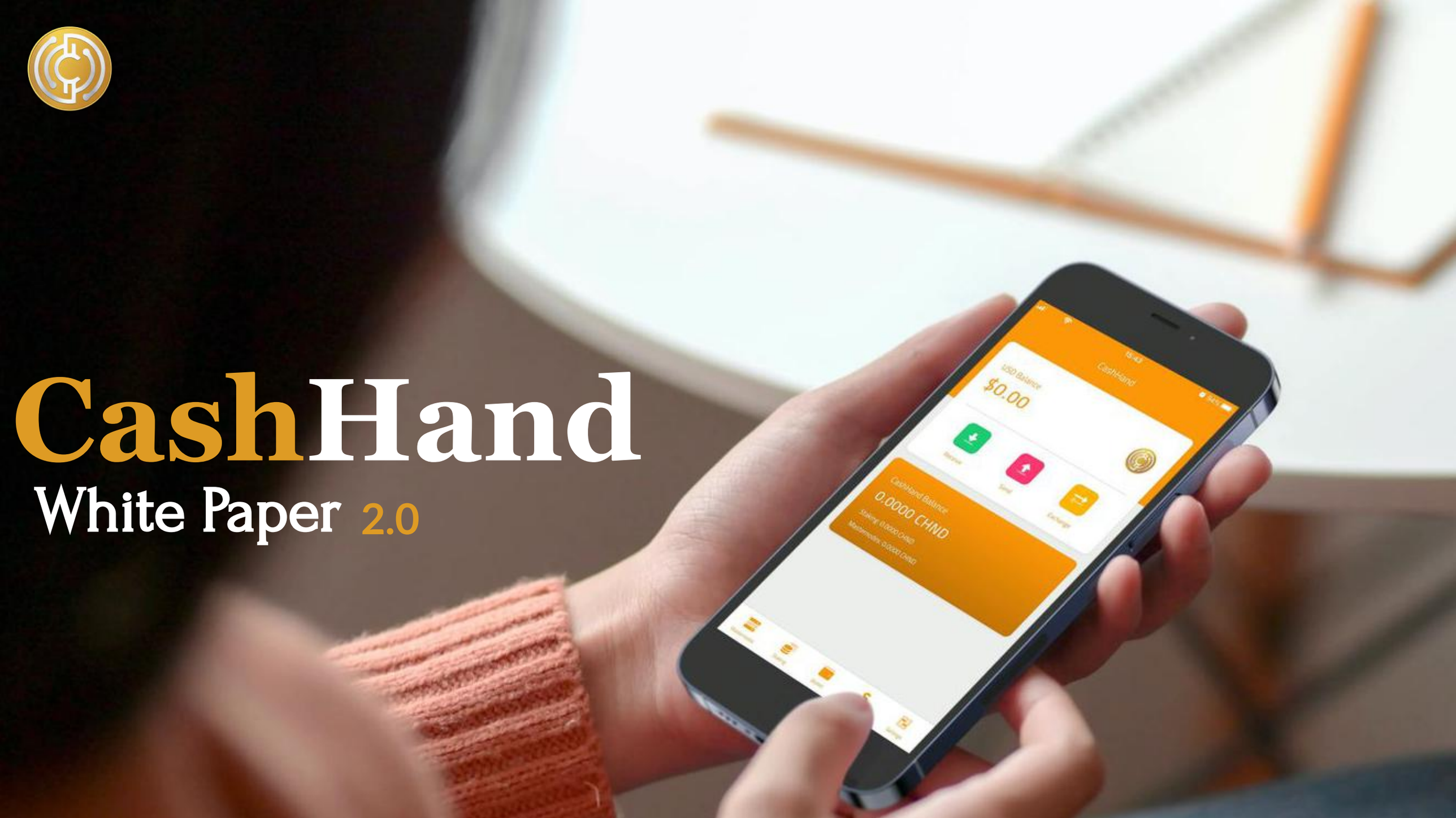
CashHand

White Paper 2.0



CashHand

CashHand

Introduction

Cryptocurrencies brought a wide unprecedented opportunities for entrepreneurs and investors with great potential for economic growth.

CashHand came to create a payment opportunity for small cryptocurrency purchases.

The cryptocurrency industry has given rise to a new market and thereby CashHand; was created to make changes on the market, so that the world can take advantage of several opportunities that the crypto market has to offer.

CashHand

What is CashHand?

CashHand is a digital,decentralized currency that does not need a third party to function.
Is means that you do not depend on banks, large corporations or governments to move your money.
With CashHand, the money is really yours.
CashHand is based on a nextremely secure decentralized network called Blockchain.

CashHand

Proof of Staking

Understanding Proof of Stake (PoS)

The proof of stake was created as an alternative to the proof of work (PoW), to tackle inherent issues in the latter. When a transaction is initiated, the transaction data is fitted into a block with a maximum capacity of 1 megabyte, and then duplicated across multiple computers or nodes on the network. The nodes are the administrative body of the blockchain and verify the legitimacy of the transactions in each block. To carry out the verification step, the nodes or miners would need to solve a computational puzzle, known as the proof of work problem. The first miner to decrypt each block transaction problem gets rewarded with coin. Once a block of transactions has been verified, it is added to the blockchain, a public transparent ledger.

Key Takeaways

With Proof of Stake (POS), CashHand miners can mine or validate block transactions based on the amount of CashHand miner holds.
Proof of Stake (POS) was created as an alternative to Proof of Work (POW), which is the original consensus algorithm in Blockchain technology, used to confirm transactions and add new blocks to the chain.
Proof of Work (POW) requires huge amounts of energy, with miners needing to sell their coins to ultimately foot the bill; Proof of Stake (PoS) gives mining power based on the percentage of coins held by a miner.
Proof of Stake (POS) is seen as less risky in terms of the potential for miners to attack the network, as it structures compensation in a way that makes an attack less advantageous for the miner.

CashHand

Masternode

In addition to the traditional Proof of Staking (PoS) rewards for mining CashHand, users are also rewarded for running and maintaining special servers called masternodes. Masternodes are used to power InstantSend and the governance system. Users are rewarded for running masternodes; 85% of the block reward is

allocated to pay the masternode network.
Masternodes allow the following services:

InstantSend

allows for almost instant transactions. CashHand transactions are fully confirmed in two seconds.

ChainLocks

which protects the blockchain against 51% mining attacks by signing blocks as they are mined.

Masternode owners must own 200 CHND, which they prove by signing a message included in a special transaction recorded on the blockchain. CashHand can be moved or spent at any time, but this will cause the masternode to leave the queue and stop earning rewards.
Masternodes cost money and effort to host, so they receive a percentage of the block's reward as an incentive. As only one masternode is paid in each block, the frequency of payment may vary, as well as the amount of CashHand paid.

CashHand

Cold Staking

This document describes an implementation of "Cold Staking" for the CASHHAND.
With the proposed system, the private keys of the coins being staked are no longer required to be held in a online (hot) node. The owner of the coins
can safely store the keys offline (for example with a hardware or paper wallet), hence the term "cold".

https://github.com/CashHand-Project/Cashhand-Wiki/blob/master/Cold-Staking.mediawiki

Background and Motivation

Traditionally, in a Proof-of-Stake consensus algorithm, block producers are required to keep the private keys to the staked coins in online nodes.
The reason is twofold. First, whenever a valid kernel input is found, the corresponding UTXO is used as input for the coinstake transaction and
therefore its private key is needed to produce the transaction signature.
Second, after the block is assembled, it must be signed with the same private key.

Even if the wallet software has a password protection which enables the use of private keys "only for staking", the wallet still needs to be unencrypted, which leaves it prone to multiple kinds of attacks on compromised systems.
Large token holders might find the reward for staking not worth the risk described above, resulting in less participation in the block-creation
process, which lowers the overall security of the network (as it is proportional to the number of coins being staked).

With "Cold Staking", block producers are still required to keep a node online, but the private keys for the staked coins can be safely stored offline.
This is achieved by signing a special "contract" transaction which transfers the coin's staking rights, without transferring the coin's ownership.

Definitions

The following naming convention is used to define the two actors in the system:

1 The coin-owner (or delegator) is the user/wallet having the private keys required to spend the coins (e.g. a hardware wallet).

2 The cold-staker (or delegate) is the block producer, who has a node always online, with the private keys required exclusively to stake the coins. "Cold" here alludes to the fact that he does not own the staked coins, but he's only staking on someone else's behalf.

3 The stake delegation is the submission by the coin-owner of the special contract transaction, which transfers the staking rights to the cold-staker.

4 The delegated balance is displayed in the coin-owner's wallet and represents the total amount of wallet's funds that have been delegated to cold-stakers. It is a part of the total balance of the coin-owner, since it's spendable at any time.

5 The cold balance is displayed in the cold-staker's wallet and represents the sum of the coins received with stake delegation. It is NOT part of the total balance of the wallet since it's not spendable.

Specification

The cold staking features are obtained with the introduction in the scripting language of a new opcode, OP_CHECKCOLDSTAKEVERIFY, and the definition of a new standard transaction type using it, named P2CS (Pay-To-Cold-Staking).

A P2CS script is defined as follows:

```
OP_DUP OP_HASH160 OP_ROT
OP_IF
OP_CHECKCOLDSTAKEVERIFY <HASH160(stakerPubKey)>
OP_ELSE
<HASH160(ownerPubKey)>
OP_ENDIF
OP_EQUALVERIFY OP_CHECKSIG
```

And the corresponding scriptSig is defined as
<signature> OP_TRUE <stakerPubKey> if used (by the cold-staker) in a coinstake transaction.
<signature> OP_FALSE <ownerPubKey> if used (by the coin-owner) to spend the coins in a regular transaction (voiding the stake delegation contract).
When the coin-owner spends a P2CS output, the OP_ELSE branch is selected (due to the inclusion of OP_FALSE on the stack), and the script behaves like a normal P2PKH transaction.
When, instead, the P2CS output is used as coinstake input by a cold-staker, then the OP_IF branch is selected and the transaction must pass the checks defined by OP_CHECKCOLDSTAKEVERIFY:
The transaction must be a coinstake transaction
All inputs must have the same scriptSig
All outputs, except the first one (coinstake marker) and the last one (masternode payout), must have the same scriptPubKey and it must correspond to the prevtx scriptPubKey spent by the input scriptSigs.
In other words, the only way for the cold-staker to spend a P2CS UTXO, is to send it to an identical P2CS contract in a coinstake transaction.
A new address type, "staking address", is defined providing a different Base58Check encoding version. Staking addresses begin with the letter S on main net and W on test net.
They are generated inside the cold-staker wallet and communicated to the coin-owner.
The coin-owner, then, creates a P2CS output embedding one of his addresses (as HASH160(ownerPubKey)) and the received staking address (as HASH160(stakerPubKey)), and sends the stake delegation transaction.
The cold-staker wallet recognizes the P2CS output script in the the stake delegation transaction, and starts to stake the coins.
The coin-owner sees the change reflected in his "delegated balance", while the coin-staker sees the amount in his "cold balance".

Validation Changes

The signature of Proof-of-Stake blocks is traditionally checked against the public key obtained by the second coinstake output. For this purpose P2PKH UTXOs are converted to P2PK when staking. This is not possible in this system because the coin-owner does not know the cold-staker public key when sending the stake delegation. He only knows the staking address, from which he can derive only the HASH160 of the cold-staker's public key.

Furthermore, the (cold) staker cannot change this script as before, as the rules defined above force it to be equal to the input script.

However, the cold-staker signs the same P2CS script and publishes his key in the input scriptSig. Therefore block validators can easily check the block signature, obtaining the staker public key from the coinstake input, rather than the output.

UI Flow

Given the specification above, the basic interaction with the system can be summarized as follows:

1)cold-staker creates a staking address (special address starting with 'S') stakerAddr.

2) coin-owner creates a normal receiving address ownerAddr.

3) coin-owner selects a number of coins (or choses a CHND amount letting the wallet pick the coins) and creates the special "stake delegation" transaction, using ownerAddr and stakerAddr, and broadcasts it to the network.

4) cold-staker adds ownerAddr to the whitelist of delegators in his wallet (or simply "accepts" the delegation tx in the GUI).

From now on cold-staker stakes the coins sent in the stake delegation on behalf of coin-owner, until the latter voids the contract, spending the coins.

Notes:

At any point in time, as long as the UTXO is mature, coin-owner can spend CHND in ownerAddr (provided he has the corresponding private key). ownerAddr doesn't necessarily need to be in the same wallet from which the stake delegation transaction was sent. It can be any address (e.g. a paper wallet).

coin-owner can re-use ownerAddr for multiple delegation transactions, even to different cold stakers (with different stakerAddr addresses). Also stakerAddr can be used for multiple delegation transactions from different coin-owners (with different ownerAddr addresses).

UI Requirements

The command line interface has been enriched with the functions required to access the feature.

method	description	used by
`getnewstakingaddress`	create staking addresses	cold-staker
`delegatestake`	create stake delegation transactions	coin-owner
`getcoldstakingbalance`	get the sum of unspent P2CS utxos involving this wallet's staking addresses	cold-staker
`getdelegatedbalance`	get the sum of unspent P2CS utxos involving this wallet's owner addresses	coin-owner
`delegatoradd`	whitelist a delegator (owner) address	cold-staker
`delegatorremove`	undo the whitelisting of a delegator address	cold-staker
`listdelegators`	show the delegator's whitelist	cold-staker
`liststakingaddresses`	show the wallet staking adddresses	cold-staker
`listcoldutxos`	show the P2CS utxos belonging to the wallet	both

The GUI will have to provide the same functionality in an easy-to-use ergonomic way.

Particular attention must be applied in highlighting the visual difference of cold-staking transactions in order to prevent users from mistaking a particular stake delegation transaction for a "regular" transaction, and accepting it as a payment (as already mentioned, for example, the "cold balance" must be kept separate from the "total balance" and it should be clear that it's not part of it).

Another safe-check is applied in the CLI when sending a stake delegation using an "owner address" which is not present in the wallet, since the lack of the corresponding private keys would result in loss of funds. To send a stake delegation using an "external" address as owner (which would be the use-case, for example, with paper wallets), the user has to force a particular argument to true. This can be streamlined in the graphical interface with a warning pop-up.
Lastly, the wallet adds a flag to the transaction when a P2CS script is being spent by the coin-owner. This is intended to be connected in the GUI with a warning message which alerts the user that the stake delegation is being voided and asks confirmation before proceeding.

CashHand

Masternode Mobile

With the CashHand Wallet app, you can run the Masternode from your mobile device wherever you are. This is the safest and easiest way to run Masternodes.

Wallet CashHand masternodes Easy and practical.

CashHand

Cold Staking Mobile

CashHand is the first cryptocurrency in the world to implement the Cold Staking system on Wallet mobile.

Earn or maintain your CashHand without compromising security, as encryption should be. Take your CashHand with you wherever you want.

CashHand

Specifications

CoinName: CashHand
ShortName: CHND
Algorithm: Quark
Block Type: Proof-of-Stake/Masternodes
Block Time: 60 seconds
Staking Reward: 15%
Masternodes Reward: 85%
MAXIMUM SUPPLY: 42,372,000

CashHand

Reward Structure

BLOCK START	BLOCK END	BLOCK VALUE AND REWARD	SUB TOTAL
0	1	180.000,00	180.000,00
2	250	1.00	249.00
251	131400	5.00	655.750,00
131401	394200	2.50	657.000,00
394201	525600	2.00	262.800,00
525601	1051200	1.00	525.600,00
1051201	2102400	0.50	525.600,00
2102401	107584900	0.40	42,372,000
Premine		**Masternode Reward**	**Staking Reward**
180.000,00		85%	15%

Documentation

What is the blockchain technology the token is based on? BTC or ETH or your own developed blockchain?

Cashhand is based on the Dash Source code which is also based on the original BTC source code. Cashhand has same development Application layer as bitcoin as they a similar API call list.

Official website:
https://cashhand.info/

Github address:
https://github.com/CashHand-Project/cashhand/releases

API document:
https://github.com/CashHand-Project/Cashhand-Wiki/tree/master/Developer-Documentation/APIs/JSON-RPC

How to integrate offline signature?
How to generate private/public key pairs and addresses using Golang?
How to signature from private key use Golang library?
How to broadcast the signature transcation?

Cashhand can be accessed with Golang just like Bitcoin and Dash using https://github.com/btcsuite/btcd/tree/master/rpcclient
and the Reference Documents on our github repository https://github.com/CashHand-Project/Cashhand-Wiki/tree/master/Developer-Documentation/APIs/JSON-RPC .

All possible function or operations, are available in the API Calls List document and some example operations on them are available in the API reference JSON-RPC document.

Other operational examples from online resources are available for Bitcoin that also works on Cashhand.
https://www.programmersought.com/article/2595883800/
https://www.thepolyglotdeveloper.com/2018/03/create-sign-bitcoin-transactions-golang/
https://medium.com/swlh/create-raw-bitcoin-transaction-and-sign-it-with-golang-96b5e10c30aa

Block-chain browser:
https://explorer.cashhand.info
https://chnd.nodesforall.com
https://blockbook.cashhand.info

Software/binary download address
https://github.com/CashHand-Project/cashhand/releases

It is advised to compile our source code on your system in order to avoid library inconsistencies.
https://github.com/CashHand-Project/cashhand/blob/master/doc/build-unix.md

For CentOS, 'apt-get' or 'apt' can be replace with 'yum'

How much CPU and Ram resources will the program consume?
RAM = 512 MB, Disk = 2BG

How much disk space is required to synchronize all blocks?
Minimum Storage Space for Cashhand Blockchain data requires 2GB of disk

Key operation API instructions:

How to get block height ?
How to create new address ?
How to transfer ?
How to get all transactions related to one wallet/account ?
How to get balance for one wallet/account ?
How to export（dump/backup）wallet ?
How to import wallet ?

All possible function or operations, are available in the API Calls List document and some example operations on them are available in the API reference JSON-RPC document.

https://github.com/CashHand-Project/Cashhand-Wiki/tree/master/Developer-Documentation/APIs/JSON-RPC

https://github.com/CashHand-Project/Cashhand-Wiki/blob/master/Developer-Documentation/APIs/JSON-RPC/API-Calls-List.mediawiki

https://github.com/CashHand-Project/Cashhand-Wiki/blob/master/Developer-Documentation/APIs/JSON-RPC/API-Reference-JSON-RPC.mediawiki

Conclusion

CashHand is more than just new code created by a developer. It is a social system that connects people. Making to morrow a better, more accessible world must be everyone'so bligation.

That is the purpose of CashHand.