Epic Cash - Exchange Implementation

Integrating Epic Cash which is MimbleWimble based presents a higher level of difficulty than other non-private coins. However, with careful planning and utilizing our guidance resources, an integration can be achieved in a relatively short amount of time.

Other Exchanges like Bitmart, ViteX, ChangeNOW, and TradeOgre have overcome the technical challenges and provided a trading platform based on the MimbleWimble protocols and procedures so it is not unachievable.

The following is an outline of tech resources required to create the framework necessary to support Account Deposits and Withdrawals.

Our core code is written in Rust and binaries are typically run under Linux. For purposes of standardization and clarity we will provide examples that are linux based. Create user 'epic' and place all binary, service, shell script, and required files in the epic home folder. You will need to run the following in support of the custom code you will create to process transactions via API:

- Epic Node
- Epicbox Server
- Epic Wallet
    - Owner Api
    - http listener for Deposit
    - epicbox listener for Withdraw

Epic Node and Wallets can be run in a linux screen session to facilitate monitoring and control (using 'screen -r <screen-name> to attach and <ctrl>A-D to detach) from shell scripts or just as a service. Epicbox runs just from a service monitored with journalctl. Example service and script files are included for each. Wallets will share the toml and log files in .epic/main as well as the default wallet db and transaction history.

Our main open source code repository is located at Github. Documentation for Exchanges is here.

It is recommended that you run your own Epic Node to post finalized transactions to the BlockChain. The code is located here.

It is also recommended that you run your own Epicbox Server which will forward transaction slates to and from the Exchange Epic Wallet and the Account holder's Epic Wallet. The code is located here. Follow the Installation instructions in the repo pdf.

You will run an Epic Wallet with the owner_api option to process local API calls and create send transactions to pass through your local epicbox server to the Account holder's wallet.

You will run an Epic Wallet with listen -m http to process incoming Deposits after the transaction has gone through an nginx proxy and the Account information part of the custom URL you will provide on the Deposit web page has been parsed and processed.

You will run an Epic Wallet with listen -m epicbox to subscribe to your epicbox server in order to send transaction slates from the Exchange Epic Wallet to the Account holder's Epic Wallet as well as accept the Account holder's Receive Confirmation slates for Finalization and Posting to the BlockChain.