Thomas Gordon

Toy Blockchain

A Simple distributed ledger, written in Haskell.

Each block will contain a unique id, a timestamp, and an owner.

Each transaction will be public, with an entire history of the chain shared by each node in the distributed network. This history is immutable.

The blockchain will be decentralized, meaning each node in the network
will contain a valid copy of the chain.

To maintain the integrity of the chain's history, i.e. it's structure, a simple consensus algorithm will be designed.

Blocks will be created, or mined, using a comparatively easy proof-of-work algorithm.
Ownership of blocks are not immutable. Any change in ownership is considered a transaction and recorded in the ledger. This transferral may only take place between two nodes. All nodes must have an id and be known.

The blockchain will have a minimal CLI which will allow mining blocks, or coin, and transferring ownership of coin.
Mining and transferring of coin requires a wallet.
Wallets provide a public-key encrypted, password protected, bill of sales.
Mining results in storing a private key corresponding to the mined block to the owners wallet.
The blockchain will check against a wallet prior to transferring ownership, which is strictly required.
Receiving coin will behave in the same manner as mining.

This toy blockchain is expected to behave as any distributed ledger should. That is, any changes made to the chain must be shared, and ownership of blocks must be protected.

Additional features:

- A Merkle tree will used to handle encryption.
- Transactional Notes.
- Expanded CLI.
- Visualization of the blockchain.
- GUI?