**Development Roadmap as a Teaching Tool**

Blockchain is a peer-to-peer technology where information is maintained collaboratively in a ledger or database in the format of transactions. The ledger consists of blocks and each block holds multiple transactions. As blockchain is decentralized, the ledger is shared among the group of participant nodes in the network and each node maintains a copy of those ledger and verifies the transactions to ensure the information stored in the blockchain ledger is immutable, verifiable and non-repudiable. The code-base we explored is a good starting point to understand how transactions (or stored information) are maintained at the protocol level to ensure the properties of immutability and verifiability of the information by using hash functions, digital signature and encryption mechanisms. But several key features of a full-fledged blockchain are missing to avoid the complexities at the beginning. In the next development of this code-base as an enhanced teaching tool, we can address the following three improvements.

• **Distributed Ledger:** In blockchain, the copy of the ledger should be maintained by all or a subset of participant nodes in the network. But in the simulation, only a single copy of the ledger is stored centrally by the **ScroogeCoin instance** as a dictionary where each element refers to a block in the chain. All the participants access the same copy of the ledger by iterating through the dictionary elements (or blocks). It’s easier in terms of implementation as we don’t have to worry about synchronizing multiple copies. But it doesn’t capture the features and challenges of the fully distributed architecture of blockchain. To emulate the distributed nature of blockchain, in the next development, separate replicas should be maintained by each participant node in the network. In order to achieve this, the copies of the **Blockchain class instance** need to be maintained and synchronized by the participant nodes. We can extend this to emulate a more realistic setting by spinning up a couple of virtual machines which will represent the participant nodes and each virtual machine will maintain a copy of the ledger.

**• Blockchain Consensus:** In blockchain, the transaction of assets or information to be stored in the ledger need to be verified and approved by all the participants. This process ensures that all the transactions are legitimate through collective agreement which refers to the consensus mechanism. In the simulation, the verification and approval of the transactions are controlled centrally by the **Scrooge instance** which doesn’t totally represent the distributed nature of blockchain. In the future development, consensus mechanisms can be integrated in the codebase for making the block finalization decisions. This can help to extend this tool to learn and experiment with popular consensus mechanisms (e.g. **proof-of-work, proof-of-stake**, etc).

**• Block Mining Fee (and Transaction Execution Cost):** The participants in the blockchain compete against each other with their resources (e.g. processing power, monetary stake, etc) to win the permission to write the next block in the ledger. These competing nodes are called the miners. When a miner is able to add a block to the ledger, it gets some **incentives** or transaction/mining fees which is the motivation for miners to participate in the distributed network to verify transactions. The mining fee comes from the transactions as the cost of executing, verifying, and storing those transactions. In the simulation, transaction or mining fee is not considered. The value of consumed coins and the value of output coins in a transaction are equal. To simulate a realistic blockchain, transaction and mining fees should be associated with transactions while proposing and adding a transaction to the ledger.