SHUBHROPARNO HALDER
SUPERVISOR : PC CHING
DEPT. OF ELECTRONIC ENGINEERING

BLOCKCHAIN BASED COLLATERAL MANAGEMENT SYSTEM

INTRODUCTION

SECURITY

EFFICIENCY

PERSONAL DEVELOPMENT

BLOCKCHAIN

COLLATERAL MANAGEMENT

METHODOLOGY

DEFINING OBJECTIVES

The first step was to define my objectives and scope of this project. The aim was to utilise the safety, efficiency and openness of blockchain and solve the persisting problems in collateral management. Another aim was to automate the process while maintaining a simple workflow. The final goal was to create a dynamic architecture to allow modifications to be made later.

UNDERSTANDING THE CONCEPTS

I utilised this stage to learn about the various features of blockchain in order to formulate methods to implement them. I also read in great detail about the shortcomings of collateral management to understand how Blockchain could be used to plug the gap. Finally, I improved my Python skills and decided what libraries should be used (Flask) to improve my code's performance.

BUILDING THE BLOCKCHAIN

This part included implementing.the mine function, new_transaction function, algorithm for calculating a valid proof, consensus algorithm and a method to check the validity of the blockchain. Finally, I created the APIs for external functions and various users to interact with the blockchain.

MODIFYING TO BUILD PLATFORM

The last stage included developing the collateral management platform on top of the Blockchain framework. This included writing new external methods to calculate collateral, accept contract details from the user, send money to different nodes' accounts and creating new APIs for the blockchain to interact with these external functions.

RESULTS

NODE 1 ENTERS INTO SWAP DEAL WITH NODE 2. NODE 1 ENTERS CONTRACT DETAILS

INFORMATION STORED AND AUTOMATIC COLLATERAL PAYMENTS ARE SCHEDULED

TRANSACTION DETAILS POSTED ON THE BLOCKCHAIN

REQUIRED COLLATERAL AMOUNT CALCULATED AND SENT TO RECIPIENT

BLOCKCHAIN UPDATED FOR ALL NODES TO VIEW PUBLICLY

DISCUSSIONS

SECURITY

Security being the primary concern, I checked the decentralisation by creating several nodes. To ensure immutability, I implemented the **valid_chain()** method which checks if the proof of work calculated is correct and if every block contains the correct hash of the previous block.

AUTOMATION

To ensure automation and maintain the high speed of the blockchain, I decided to create external functions. Now all steps including acceptance of contract details, its storage, scheduling transactions and collateral payments and deposition into respective nodes' account are separate steps. All these steps were consolidated and called by the **/info** API which ensured a smooth workflow.

SCALABLE

The aim was to make the architecture dynamic such that new financial products can easily be incorporated for which collateral can be calculated and posted. Therefore, the code reads a single parameter 'type' in the contract and calculates a collateral for that type of product. This means adding only another 'if' condition adds a new functionality.

CONCLUSIONS

WHILE THIS PROJECT LOOKED INTO THE POSSIBILITY OF COLLATERAL MANAGEMENT AS A POSSIBLE APPLICATION FOR BLOCKCHAIN FURTHER DEVELOPMENTS SUCH AS ADDING NEW DERIVATIVE TYPES, ENSURING REAL-TIME FLOW OF DATA AND DEVELOPING A SAFER USER INTERACTION PLATFORM COULD BE POSSIBLE.