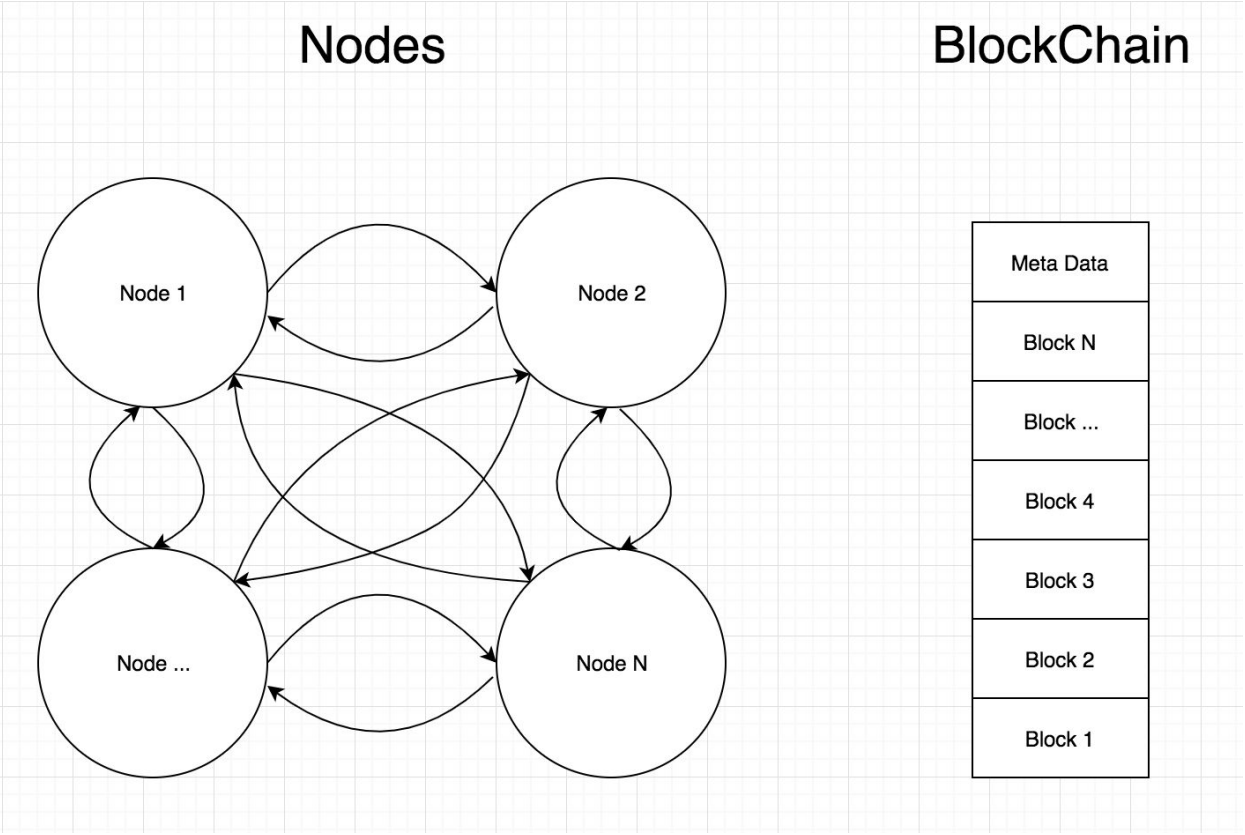
5.1 Introduction

This Software Requirements Specification (SRS) documents the requirements for the decentralized database system, called **debased**. To this end, debased will rely on blockchain technology and proof of stake to guarantee the system can meet the three guarantees of the CAP theorem under normal operation, favoring accessibility in the case of a system partition. From a user perspective, the debased system functions similarly to current enterprise grade relational database systems. The user will interact with the system using modified SQL calls. These calls will be encrypted and sent to the



debased system; the system will verify the commands either fulfilling the request, updating the blockchain and returning pertinent information, or rejecting the command if the transaction is invalid.

Figure 5.1: High level diagram of technology used in debased

5.2 CSCI Component Breakdown

The debased system is divided into three main subsystems, or sections. The first section, Peer-to-peer Network, controls the communication between nodes. The second section, the blockchain, stores the transaction history of the system, both data and

currency. The third section, proof of stake, allows the system to come to consensus, creating a uniform blockchain and incentivizing nodes for their input in the process.

5.3 Functional Requirements by CSC

5.3.1 Peer-to-peer Network

The following requirements are levied on the Peer-to-Peer Network Subsystem of the debased project.

5.3.1.1 Messaging Protocol

5.3.1.1.1 The subsystem shall encrypt messages
5.3.1.1.2 The subsystem shall send messages
5.3.1.1.3 The subsystem shall receive messages
5.3.1.1.4 The subsystem shall decrypt messages

5.3.1.2 Peer specs

5.3.1.2.1 The subsystem shall structure the message payloads for dispersing information

5.3.2 Blockchain

5.3.2.1 Block construction

5.3.2.1.1 The subsystem shall group transactions into blocks
5.3.2.1.2 The subsystem shall link the created block to the previous block on the chain
5.3.2.1.3 The subsystem shall number the created block with the previous block number plus 1

5.3.2.2 Metadata

5.3.2.2.1 The subsystem shall contain the current block number
5.3.2.2.2 The subsystem shall contain addresses to each block in the blockchain
5.3.2.2.3 The subsystem shall contain permissions for each account
5.3.2.2.4 The subsystem shall contain the current balance for each Account

5.3.3 Proof of Stake

5.3.3.1 Voting System

5.3.3.1.1 The subsystem shall randomly select a node to present its version of the added blocks
5.3.3.1.2 The subsystem shall determine the validity of the presented blocks
5.3.3.1.3 The subsystem shall determine its confidence in its position

5.3.3.1.4 The subsystem shall determine how much currency to wager
5.3.3.1.5 The subsystem shall place a vote on the presented block
5.3.3.1.6 The subsystem shall receive votes from other nodes
5.3.3.1.7 The subsystem shall place more votes based on all information available

5.3.3.2 Consensus

5.3.3.2.1 The subsystem shall determine if the votes are converging
5.3.3.2.2 The subsystem shall restart the voting system if the vote fails
5.3.3.2.3 The subsystem shall update the blockchain if the vote passes

5.3.3.3 Payout

5.3.3.3.1 The subsystem shall distribute coins among the other nodes based on vote and consensus reached
5.3.3.3.2 The subsystem shall add the determined payouts to the accounts in the metadata

5.3.3.4 Confidence in Peers

5.3.3.4.1 The subsystem shall determine if data received from a node is malicious
5.3.3.4.2 The subsystem shall set a time to check the blockchain in the future
5.3.3.4.3 The subsystem shall lower confidence in nodes found to give false information
5.3.3.4.4 The subsystem shall start a vote to remove a node if confidence in a node drops below the threshold

5.4 Performance Requirements

5.4.1 Query a row

5.4.1.1 The user will be able to send a query a row in less than 1 second.

5.4.1.2 The user will be able to receive a response from a query in less than 1 second.

5.4.2 Create a table

5.4.2.1 The user will be able to create a table in less than 2 seconds.

5.4.2.2 The user will be able to receive a response in less than 1 second after the creation of a table.

5.4.3 Delete a table

5.4.3.1 The user will be able to delete a table in less than 2 seconds.

5.4.3.2 The user will be able to receive a response in less than 1 second after the creation of a table.

5.4.4 Modify a row

5.4.3.1 The user will be able to modify a row in less than 1 second.

5.4.3.2 The user will be able to receive a response in less than 1 second.

5.4.5 Launch a node

5.4.3.1 The user will be able to launch a node in less than 5 seconds.

5.4.3.2 Once launched, the node will be able to connect to other nodes affiliated with the decentralized database.

5.4.3.3. The nodes will communicate via a home build peer-to-peer system.

5.5 Project Environment Requirements

5.5.1 Hardware requirement for development, deployment, and execution.

Category	Requirement
Processor	Intel Pentium Gold G5500 or Better
RAM	4 GB
Hard Drive Space	32 GB
Display	Any

5.5.2 Software requirement for development, deployment, and execution.

Category	Requirement
CLI	Bash
Operating System	masOS X10.14 or later; Ubuntu 18.04.1 or later
Text Editor	Any