dBlind – Blockchain Simulation with (PoET)

By Lex 

dBlind Simulation

1

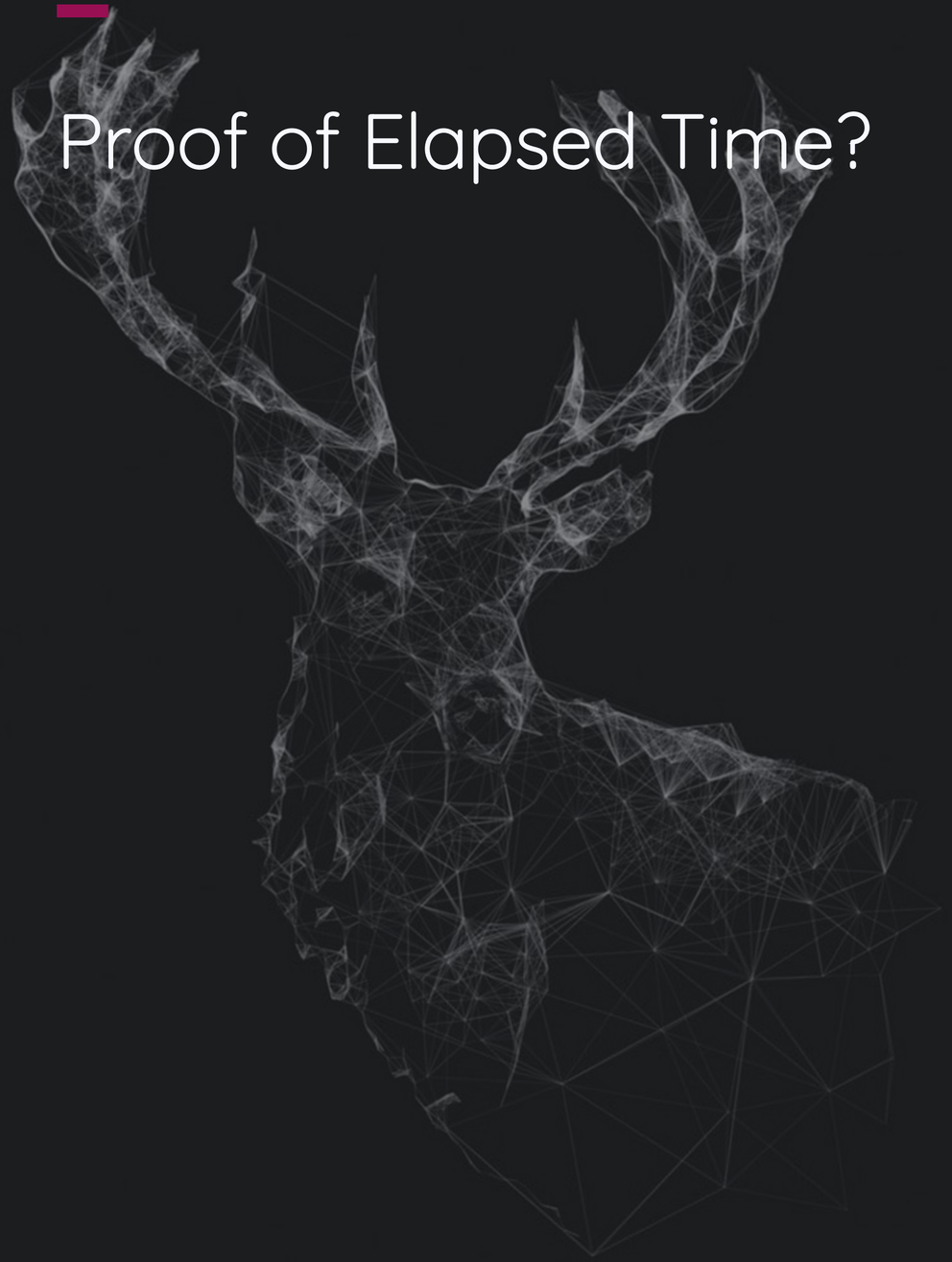
dBlind Simulation is a private blockchain application with the consensus algorithm PoET to simulate the banking sector.

2

The simulation will be based on the financial sector because blockchain technology has many potentials for financial industries. Moreover, it can eliminate the threat or the risk of fraud in all areas of banking.

For instance, most fraud cases are double-spending, or data can be tampered with or stolen to alter a particular situation or customers' data explosion into public.

Proof of Elapsed Time?



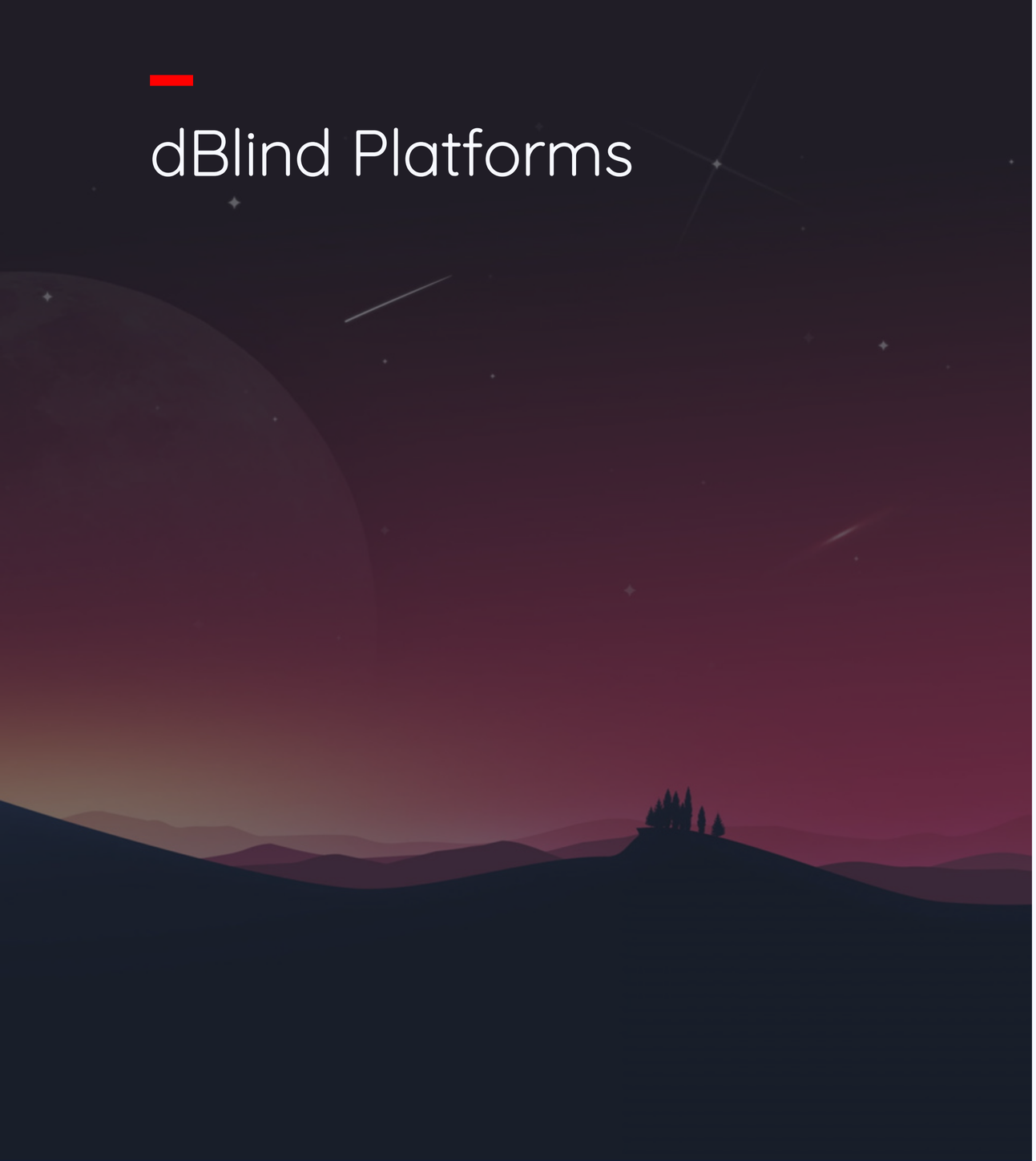
1

Proof of Elapsed Time is a common consensus algorithm used on the blockchain network. 👉

2

As each participant node in the blockchain network will receive a waiting time and with its randomly time duration, the very first one that has completed the given waiting time will be able to add the new block.

dBlind Platforms



1

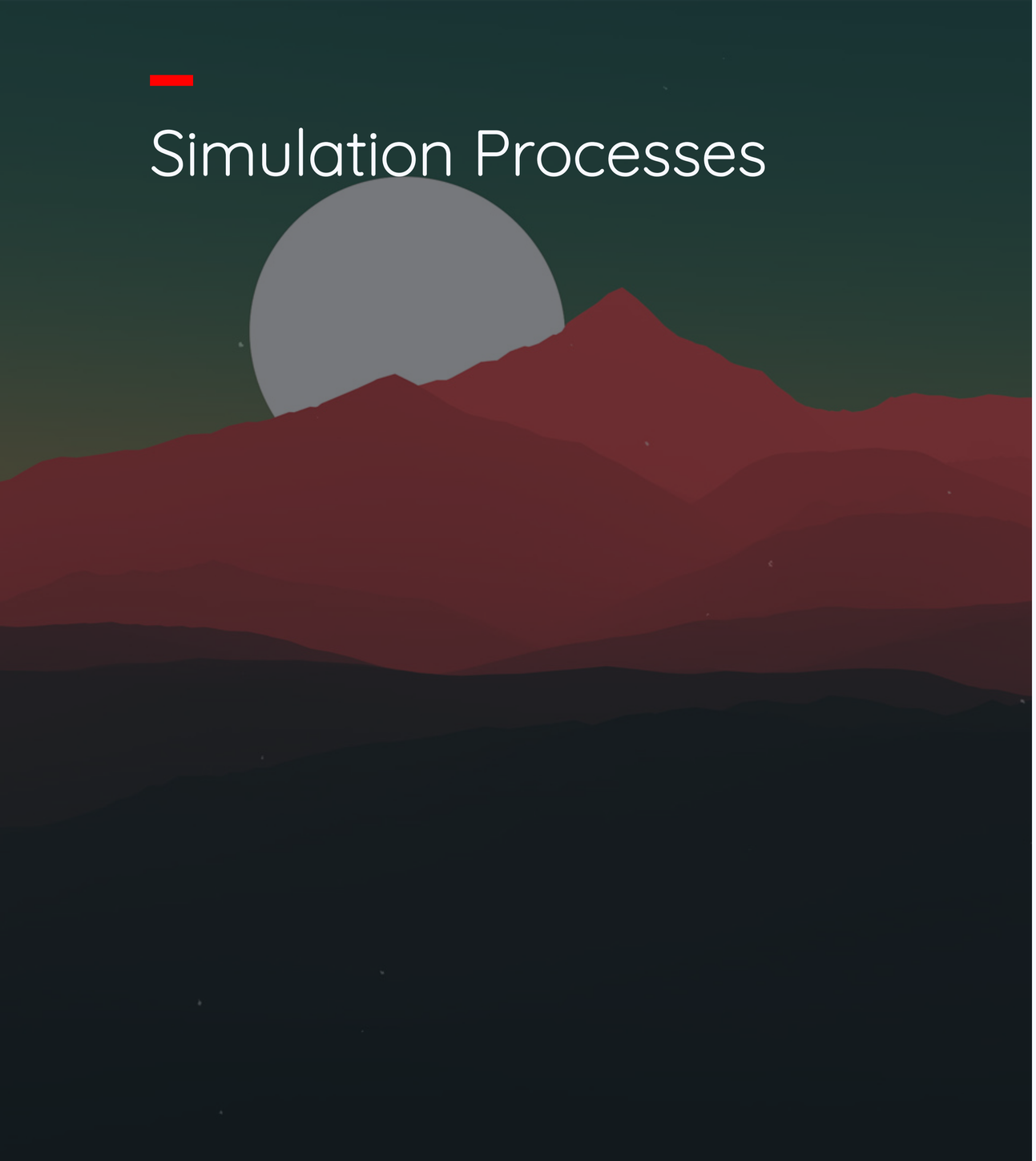
dBlind App: is written in TypeScript using ReactJS for Client-Side Application.

2

dBlind Cloud: is written in TypeScript using NodeJS 12 as the Cloud Functions to be a middleware or a network protocol to prevent deleting, editing or invaliding transactions into the database. Additionally, it is the one who fetches transactions from pools and generates random times into each node to add blocks.

3

dBlind Service (Node): is written in TypeScript using NextJS as a web service to facilitate as a node to participate in the private blockchain simulation.



Simulation Processes

1

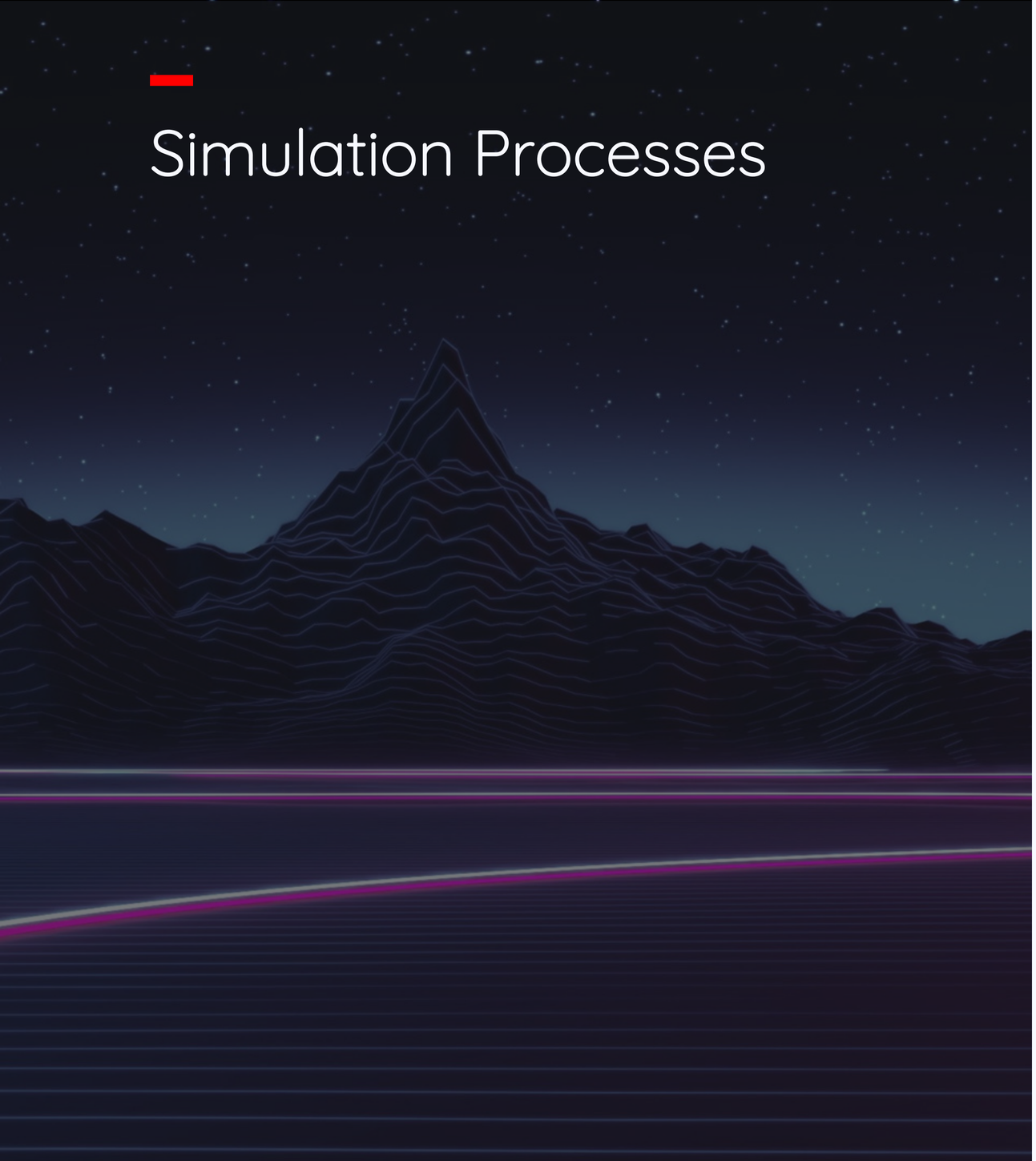
Customers can make a payment to another customer. Then all transactions will be added into the network pools. 🚀

2

Cloud Functions, also known as the network protocol, will get the earliest transaction within the pool and generate the transaction into the queue. Then, it takes all nodes within the network to continue the work by giving them generated random times. 😈

3

Web Service facilitated as the node will listen in real-time and pick the very first queue as its task. The very node with the smallest amount of time will complete the adding blocks and distribute them across the network. ✌🏿



Simulation Processes

4

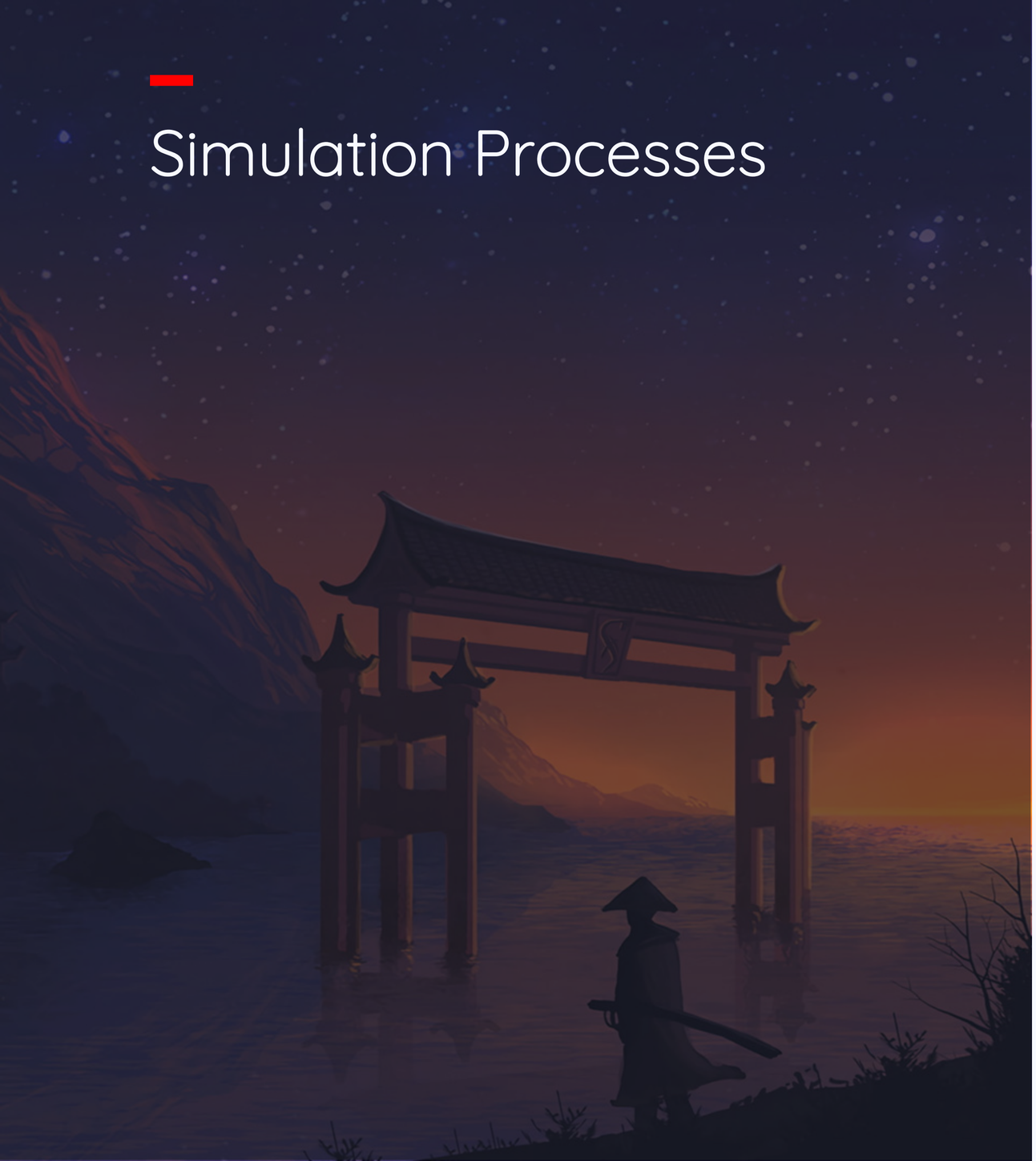
When a node has completed its task, it marks that specific queue as isCompleted, then other nodes while countdown for its task will halt everything and prepare to receive the next task within the network by listening to the next queue. 🤖

5

During the distribution, each node will check the received transactions themselves and only receive the validated transactions into their nodes. 🥶

6

If any node has missing blocks of transactions, it will reinstate itself by pulling the missing blocks of the transaction from the network. Moreover, verify those blocks by checking the block's hash value. 😈



Simulation Processes

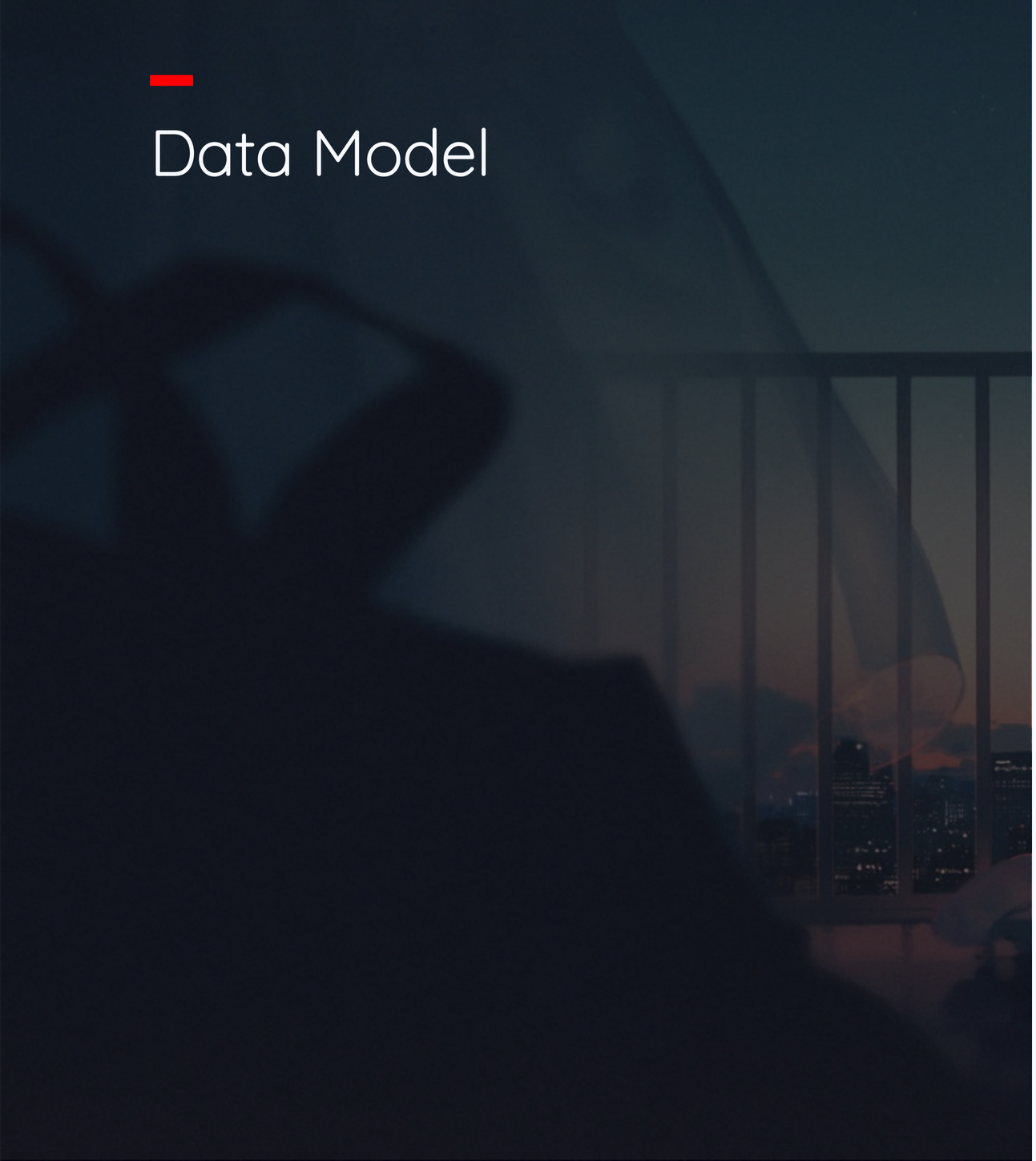
7

Since this is a private blockchain simulation, the database is centralized and decentralized storage.

Cloud Functions will handle all the procedures like preventing data tampered with and data deletion. Suppose any block or transaction information is being deleted. In that case, the Cloud Functions will reinstate the data back, or if the data is being modified, it will reinstate the previous one that matches the hash value of a block.

8

And each node that stores data as distribution will not take the modified block or transaction into itself. Moreover, it will delete invalid blocks automatically.



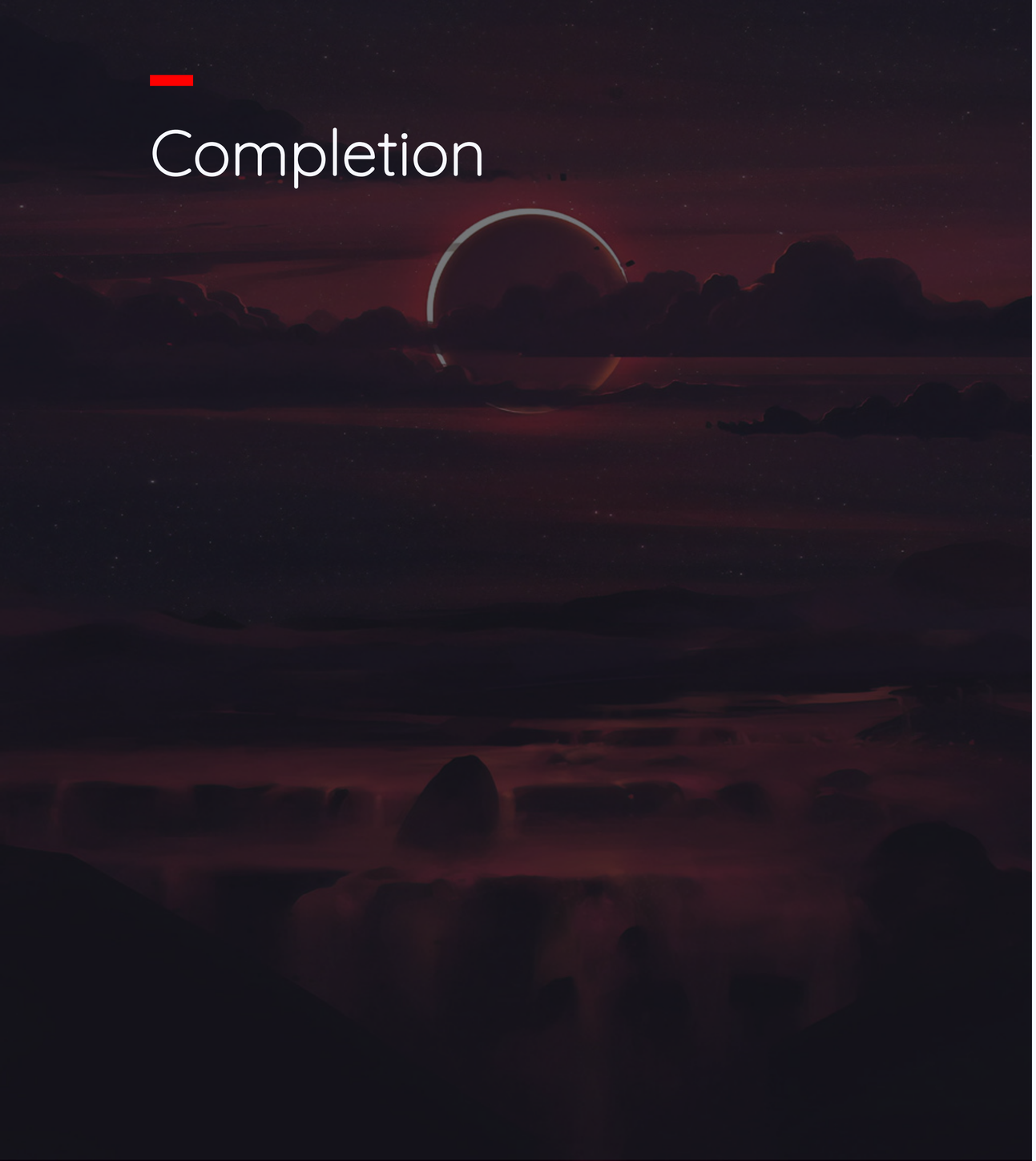
Data Model

Block

blockHash:
ownerAddress:
transactionAddress:
amount:
createdBy:
createdAt:
previsousHash:

Transaction

transactionHash:
senderAddress:
receiverAddress:
amount:
label:
message:
createdAt:



Completion

Partial peer-to-peer: centralized & decentralized storage, and all nodes will receive blocks of transactions from each other and pull from centralized data to check.

Distributed: when a node has a permissioned right to add a block into the network, it will get all nodes and distribute to them concurrently.

Cryptographically secured: all data models use hash value and hash value reference addresses.

Add-only: cannot be tampered with or deleted when hard manually deletion.

Consensus: this simulation uses Proof of Elapsed Time.

Features

1

Block data table visualization

2

Make payment to other customers via address


Simulation Demonstration