ADMAS UNIVERSITY SCHOOL OF POST GRADUATE STUDIES

DEPARTMENT OF COMPUTER SCIENCE - SECTION 2

OBJECT ORIENTED SYSTEM DESIGN

CROSS-BORDER MONEY TRANSFER USING BLOCKCHAIN

System Design Document (SDD)

Project Members:

Name	ID No.
Seifu Birega	PGMGC/8062/20
Yordanos Woldu	PGMGC/8053/20
Mekonnen Ayalew	PGMGC/3118/19
Simeon Gebre Yohans	PGMGC/8067/20
Habtamu Birhan Godana	PGMGC/8023/20

Submitted to: Abdi Mulatu (Ass. Professor)

FEBRUARY 28, 2021

Table of Contents

1. Introduction

1.1. Overview

The blockchain is a private / public, trusted and shared ledger, based on a ***peer-to-peer network*** where no one controls it, instead it is maintained by thousands of participants, making it a shared ledger. Since the blockchain is available to all the participants, it is a public ledger. The information that is recorded on the blockchain cannot be manipulated without going unnoticed, making it a trusted ledger. These elements allow information transfer on the blockchain without any intermediaries. Consequently, the role played by financial institutions as trusted third party mitigating the risk of a transaction is being questioned. Blockchain and its benefits are interesting for the payments industry, as it promises to facilitate fast, secure, low-cost international payment processing services through the use of encrypted distributed ledgers that enable trusted verification of transactions in real time without needing intermediaries. [1]

1.2. Purpose of the System Design

The goal of the system design in this document specifically is to model the system in detail by including the missing components of the RAD and clarifying the most important components of the RAD document using a better reference model. This document will be an input for the implementation and testing of the CBMT; but some of the most important figures in the RAD are also included because they are input for the design and can show the current and the proposed systems clearly.

2. System Design Model

2.1. Overview of the Current System – High Level

As clearly specified in the RAD, the infrastructure of global payments moves money from payment system to payment system through a series of internal book transfers across financial institutions. Most international transactions are executed through the *Society for Worldwide Interbank Financial Telecommunications (SWIFT)* network, a *cooperative society founded by seven international banks* which operate a global network *to facilitate the messaging of financial transfers as shown in the figure below.*

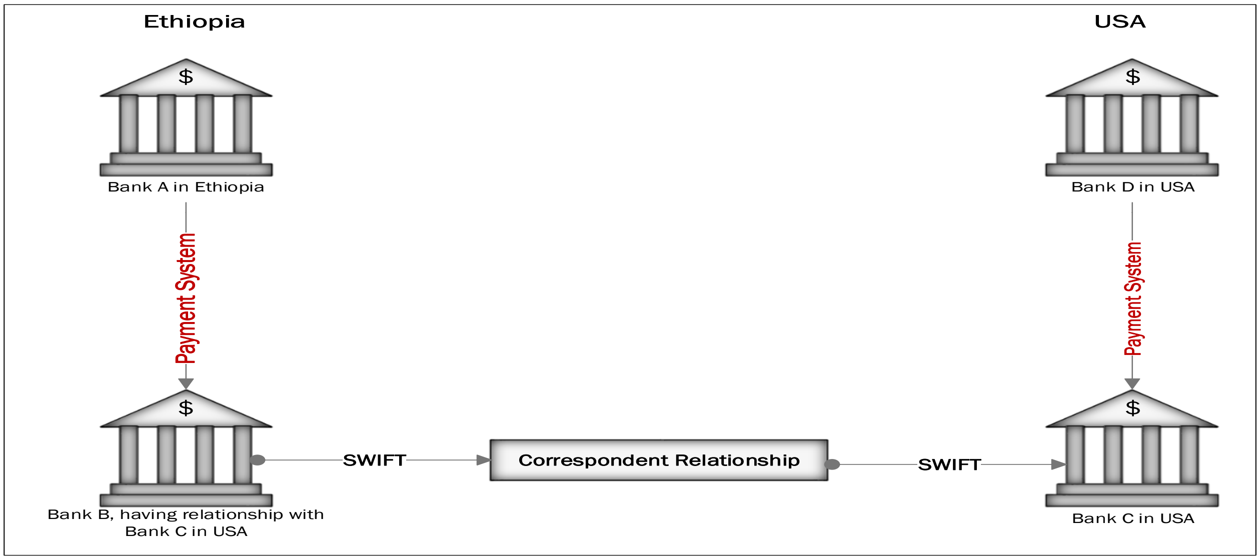


Fig 1: The traditional cross-border payment system (Bank A to Bank D)

2.2. Proposed System Design

2.2.1. Overview

The transactions in blockchain are *cryptographically signed to prevent tampering, making it a secured and trusted platform*. Additionally, an *auditable trail of all the transactions is maintained in the ledger*. *These features remove the necessity of a trusted middle man or intermediary banks financial institutions rely on for global payments.* The use of intermediary adds to the cost and causes a delay in payment processing. The next figure shows Blockchain eliminating the intermediaries

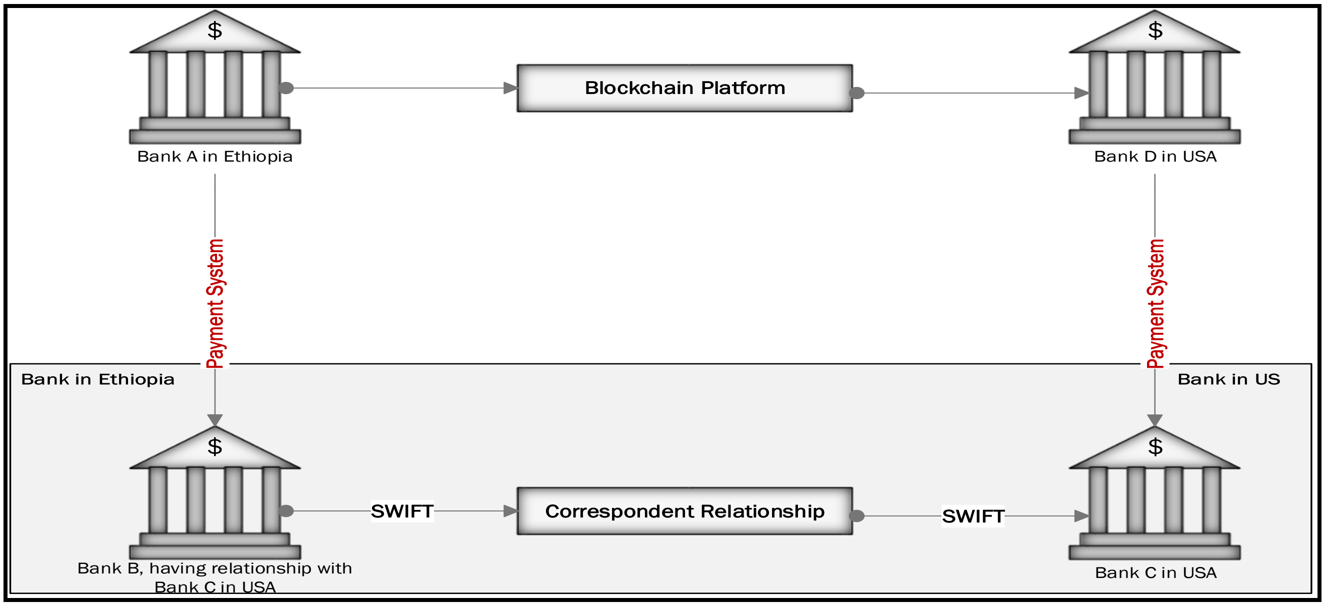


Fig 2: Money transfer from Bank A to Bank D eliminating the intermediaries

In this section of the SDD we focus on the application schemas, database schemas and user interface specifications for the solution we proposed in the RAD known as the CBMTS (Cross-Border Money Transfer System) mainly the business logic layer.

Because of its nature the distributed ledger technology uses distributed computing architecture; it is therefore, we proposed a layered architectural pattern known as Distributed Data and Application which mainly focuses on the Application / Business Logic Layer, Data Manipulation Layer and the Data Layer (Distributed Ledger). The next figure which is used in the RAD will give as a view of the major components participated in the system

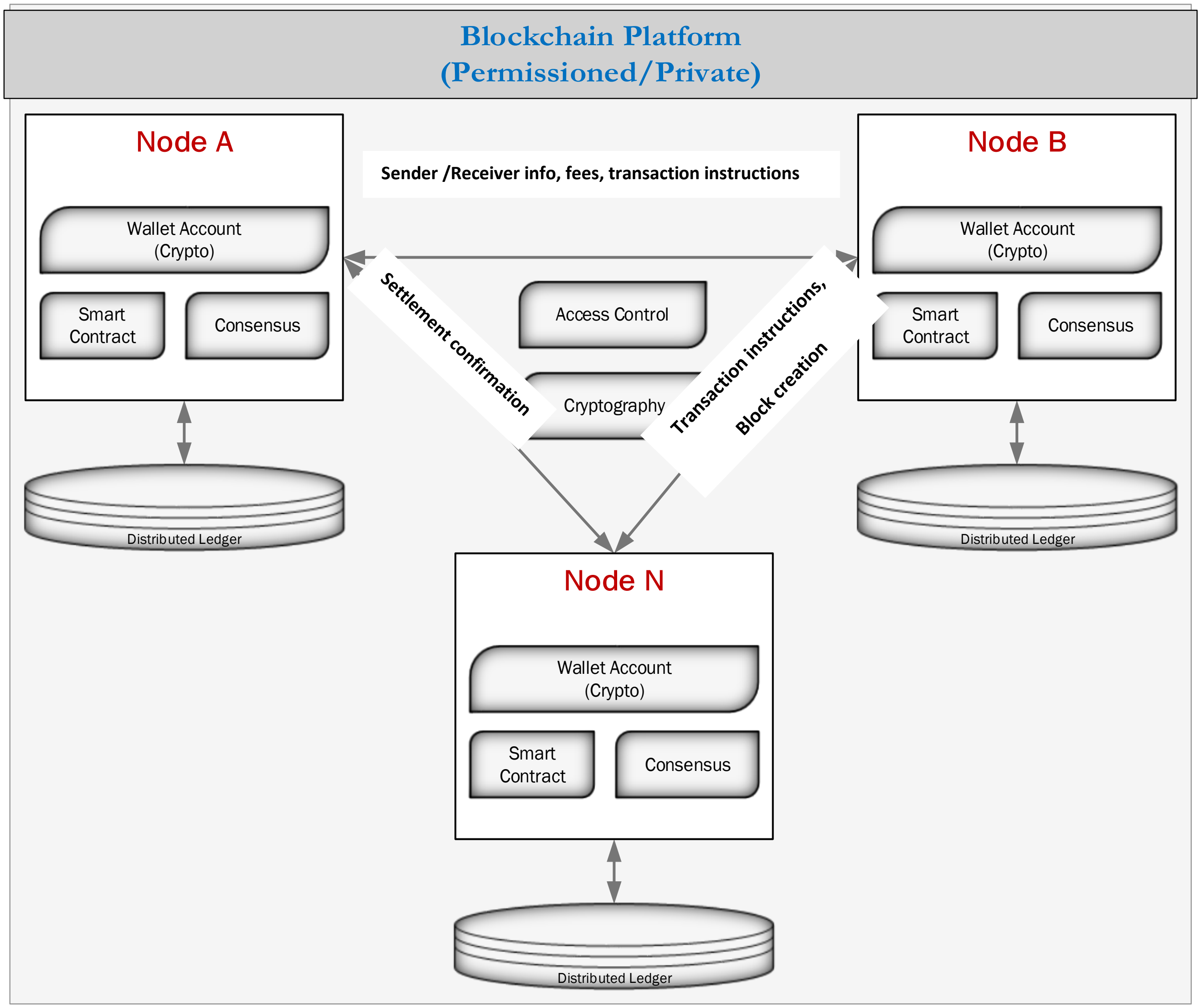


Fig 3: A Permissioned Blockchain Platform-Proposed

2.2.2. The Application Architecture

The CBMTS is a decentralized application(DApp) that runs on a blockchain network and enables direct interaction between consumers and providers, for example, connecting buyers /payer/ sender and sellers /payee / receiver in a decentralized marketplace. Similar to the centralized application architecture, a DApp usually involves a decentralized backend that runs on the blockchain network and a centralized frontend that allows end users to access their wallets and make a transaction. The below diagram shows the differentiation between centralized and decentralized applications:

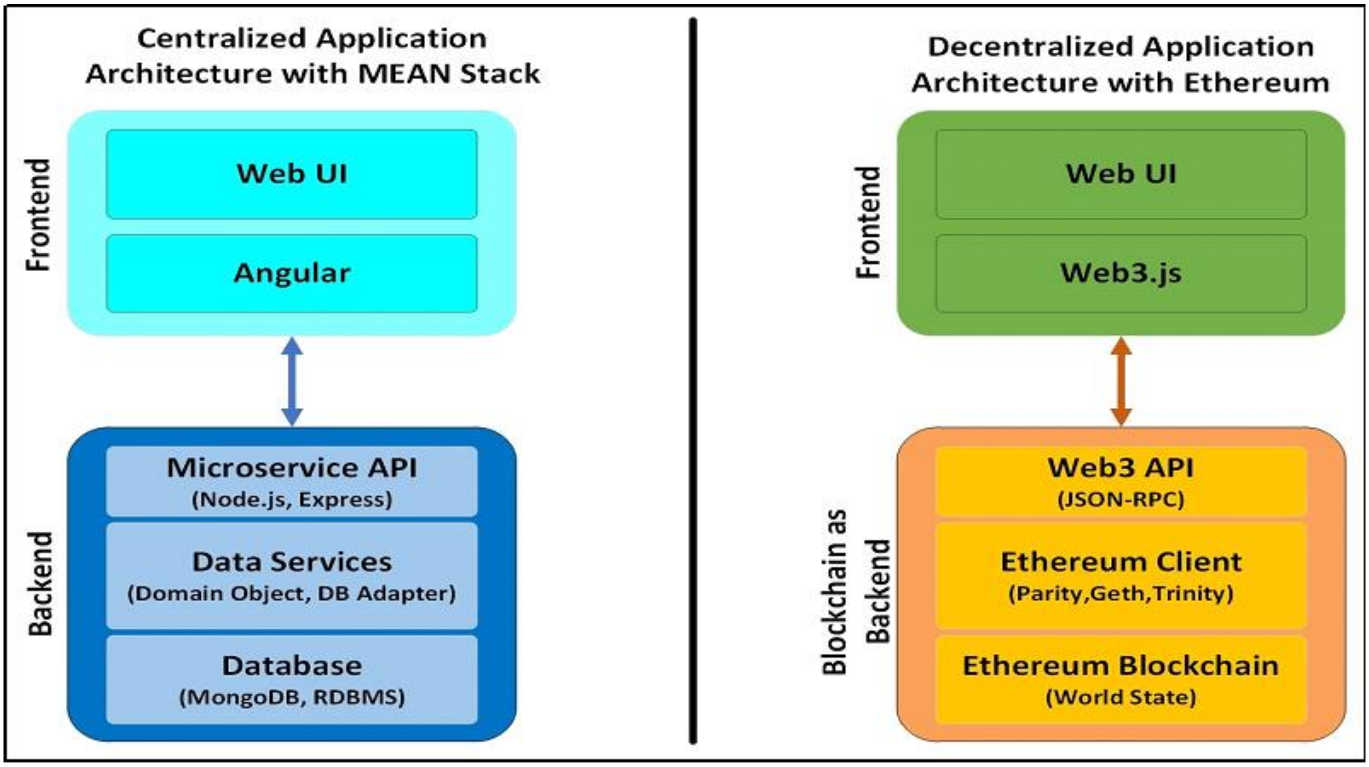


Fig 4: Decentralized Application Architecture-Right: Ethereum

2.2.3. The Application and Business Logic Layer

2.2.3.1. Overview

Because of the complexity, make ease and implementation challenges in a duration of three weeks the RAD document on section 4 modelled the system as simple as possible, but we strongly believe that we have to separate the **Transaction/Block** object specified in the class diagram to be normalized and the data should be stored separately for the **Transaction** *processing* and ***Block*** creation processes identified in the how a blockchain works (section 1.4) and the functional requirements section (section 3.4) including all the major components in the RAD shown in ***Fig 1***, ***Fig 2, Fig 3*** and ***Fig 4***.

2.2.3.2. The Business Logic Layer

As clearly specified in the functional requirements section 3.4 of the **RAD,** this section of the ***SDD*** document in general focuses on the business logic layer of the ***CBMTS*** and describes the following ***Six*** major components of the system

1. **Actors and roles**: In CBMT system context, each ***node*** is an actor in the network, but actors perform different operations. Based on that, two ***roles*** are defined:

   Client - user of the technology that wants to send or receive bitcoins. To do so, a transaction is created, signed and broadcasted to the network.

   Miner is an actor who mines data; this means validation of transactions, generating blocks and submitting them to the network, to be included in the blockchain. When submitting a valid block, miner is rewarded with bitcoins (for the work done on the proof-of-work algorithm).

2. **Services**: There are two main services; transaction creation service allows the client to create a new transaction and send it to the Bitcoin network (to transfer bitcoins to another user). The Miner will use data mining services to be able to mine data.

3. **Processes**: There are 4 sets of major processes: *Network discovery process, Transaction creation process, Block validation process and Mining process.*

   The next figures consecutively show the Network discovery, Transaction creation and Block validation and the mining processes.

2.2.3.3. The Network Discovery Process

Finding known CBMTS peers (Figure 5) - how a new node can find known peers IP addresses to connect to the network. New node will either get a known IP from unknown 3rd source, another person for example, or it can query the DNS seed list for known IPs. Once node has IPs, it can connect to a network.

Handshake process - once connection with the known IP address is established, the new node has to transmit a version message to the existing node, in order to verify that they are running the same version of the software. The existing node can either choose to respond to the request or not. In the positive case, existing node will reply with its version message. and a message is sent to acknowledge that the peer is willing to connect.

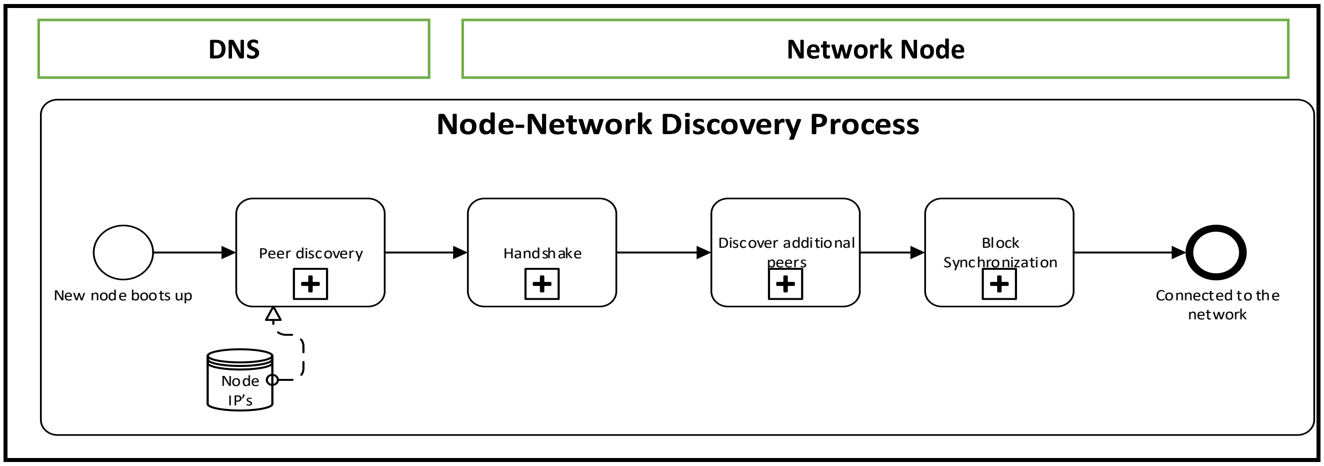


Fig 5: The Network Discovery Process

- Discover additional peers - once the new node receives the confirmation, it sends its IP address for other nodes to be propagated around the network and/or it can ask the neighbouring node for IP addresses of other nodes, to establish further connections.
- Block synchronization - when first connecting to the blockchain and before the new node can start validating transactions and newly mined blocks, it has to download and validate the entire blockchain from the very first block. The new node will download the longest chain based on what its neighbours have (in the handshake process, nodes exchange information about the latest block). When connecting to the network next time, node will synchronise all the blocks that have been added since last connection.

2.2.3.4. Transaction Creation process

The next diagram shown in Fig 6. describes creating transactions and transmitting them to the CBMTS network. Process begins with the Client specifying the amount of money to send and the receivers CBMTS address. Then the transaction is constructed and the client will sign the transaction with a private key. After signing, the transaction is transmitted to the neighbouring nodes, who will validate the transaction. If the transaction is valid, they will propagate the transaction to other neighbouring nodes who repeat the validation process. Additionally, node will keep the transaction in the memory (called unverified transaction pool). Otherwise, if the transaction is invalid, the neighbouring node will not propagate it forward.

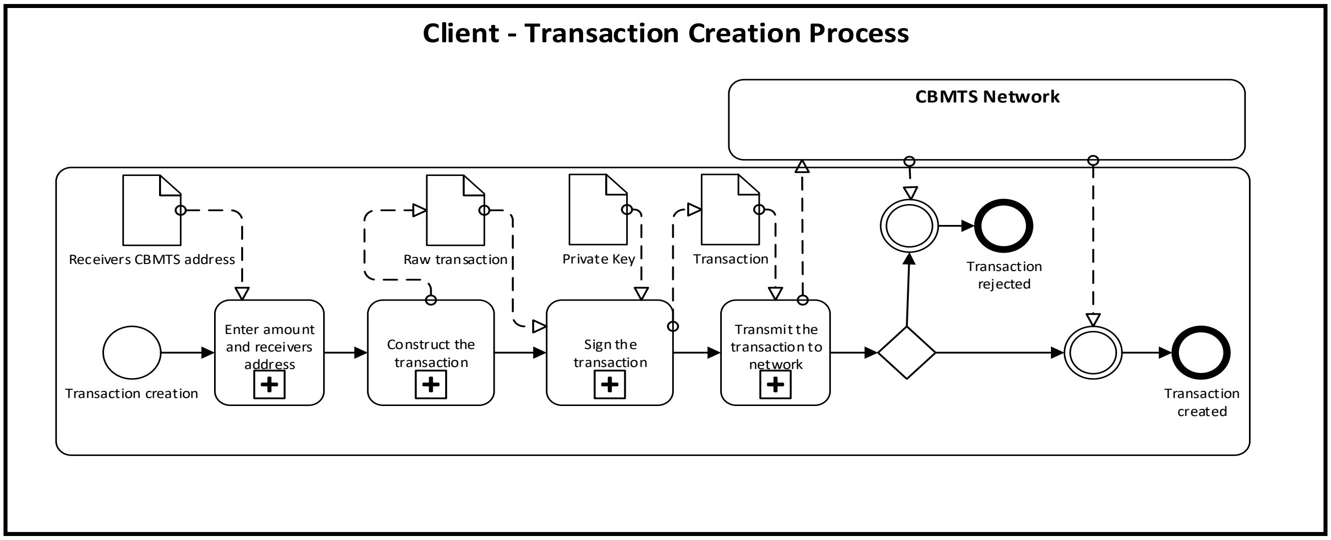


Fig 6: The Transaction Creation Process

2.2.3.5. The Block Validation and Mining process

Figure 6 explains the generation of a new block and submitting it to the network to be included in the global blockchain. New block is triggered when the previous block has been mined, broadcasted to the network and valid. Miner first creates a new block from a template, collects unverified transactions, adds the hash value from the previous block and a generation transaction, which will reward the miner on creating a successful block. Then the race for calculating the proof-of-work starts. Whichever miner is the first to find the solution and broadcast the new block to the network (to be added to the global blockchain), will receive the reward for that block. Once a new block is broadcasted on the network, other miners will stop mining the current block and begin working on a new one.

When the new block is submitted to the network by the miner, it is not added to the blockchain right away. Nodes in the network will know when a new block has been mined, and then each of them will validate the block according to the consensus rules (see Figure 7).

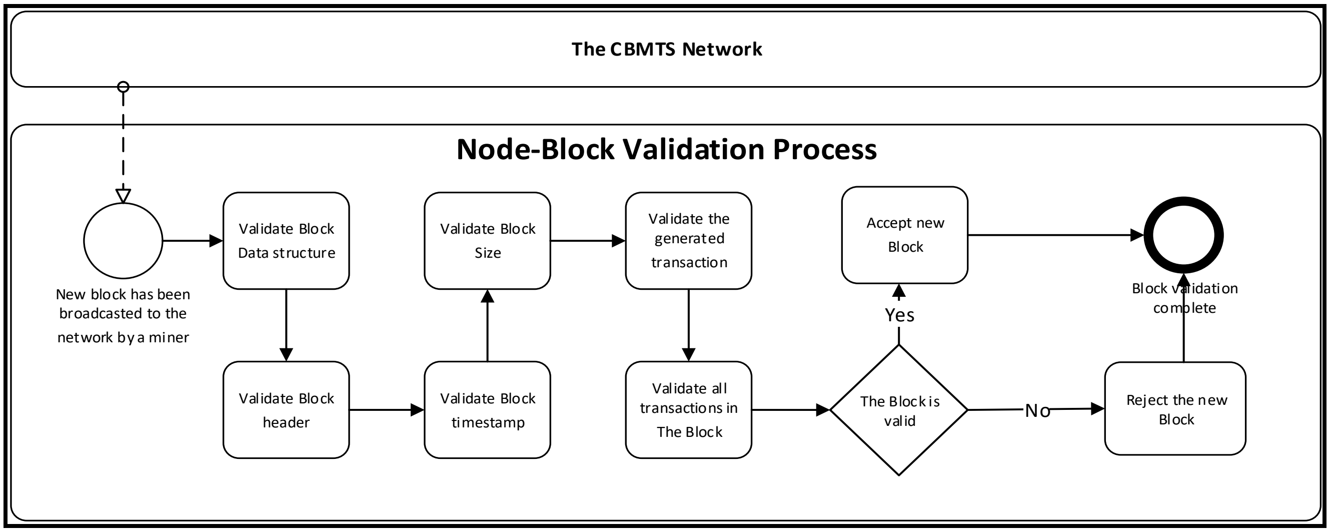

The CBMTS Network
Node-Block Validation Process
New block has been broadcasted to the network by a miner
Validate Block Data structure
Validate Block header
Validate Block timestamp
Validate Block Size
Validate the generated transaction
Validate all transactions in The Block
The Block is valid
Yes
Accept new Block
No
Reject the new Block
Block validation complete

Fig 7: Block validation / Mining Process

If the block is valid, they will add the new block to their local blockchain. This independent validation ensures the consensus in the network, because everyone is validating based on the same rules. The consensus includes

- Block data structure validation
- Block Header validation
- Block timestamp validation
- Block size validation
- First (and only the first) transactions is a generation transaction, that will reward the miner
- Validation of all the transactions included in the block.

Stream Lining Payment Protocol

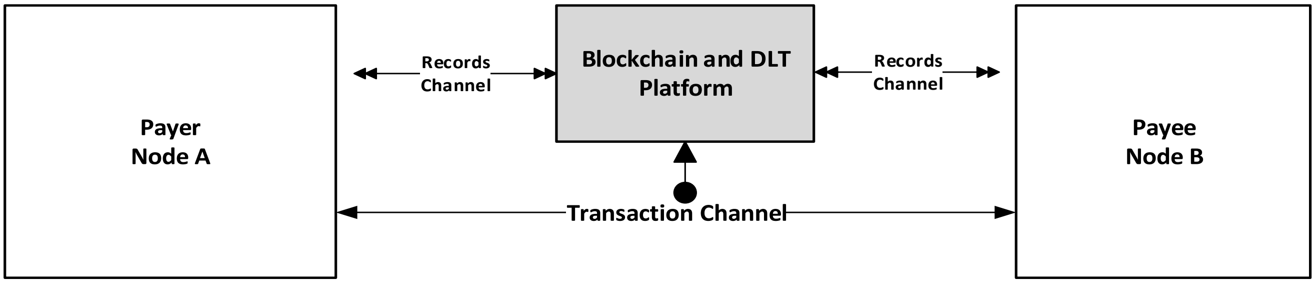


Fig 8: Stream Lining Payment Protocol

2.3. Design Goals and Constrains

The goal of the system design in this document specifically is to model the system in detail by including the missing components of the RAD and clarifying the most important components of the RAD document using a better reference model. Moreover, there is a limitation in covering all the layers of the application architecture in detail.

This design document focuses on the Business Logic Layer of the CBMTS. The Database Design/Data Model section shows a high level of objects participated in the CBMTS.

2.4. Subsystem Decomposition

The system contains few functionalities with related features and is not decomposed for the time being. For example, functionalities like ***Smart Contracts*** are not included as part of the system, the system will be decomposed in to further subsystems.

2.5. Deployment Design / Hardware and Software

Even though smart contracts are not the scope of this document, as shown in the decentralized application architecture in fig 4 the next figure shows the deployment diagram for Blockchain based systems to show the hardware software interactions

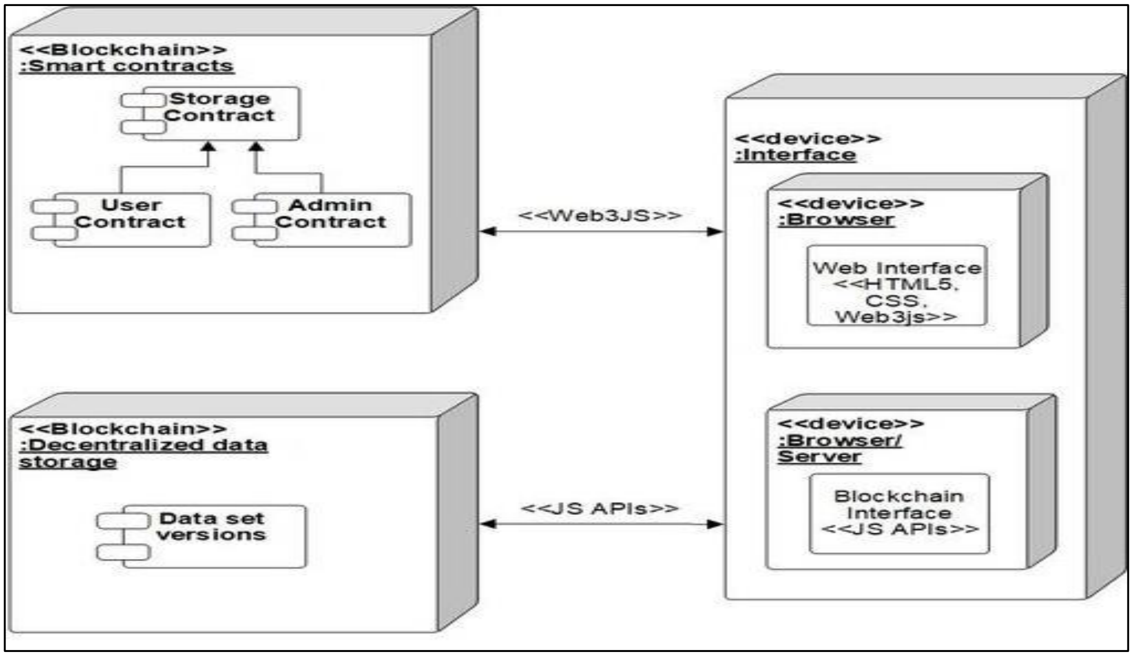


Fig 9: Deployment diagram for the proposed Blockchain CBMTS

2.6. Database Design

2.6.1. Overview

A database design document shows the relationship between the objects in a system including the stored data in each object. Even though modeling the database i seems a little bit challenging compared to other systems we try to model the system using DFD of the structural modeling to show high level data flows and data attributes with the next two figures shown below

2.6.2. Data Flow Diagram

The next figure shows high level flow of data when a payer sends money to a payee in different location avoiding the correspondent banks using the Blockchain distributed ledger technology

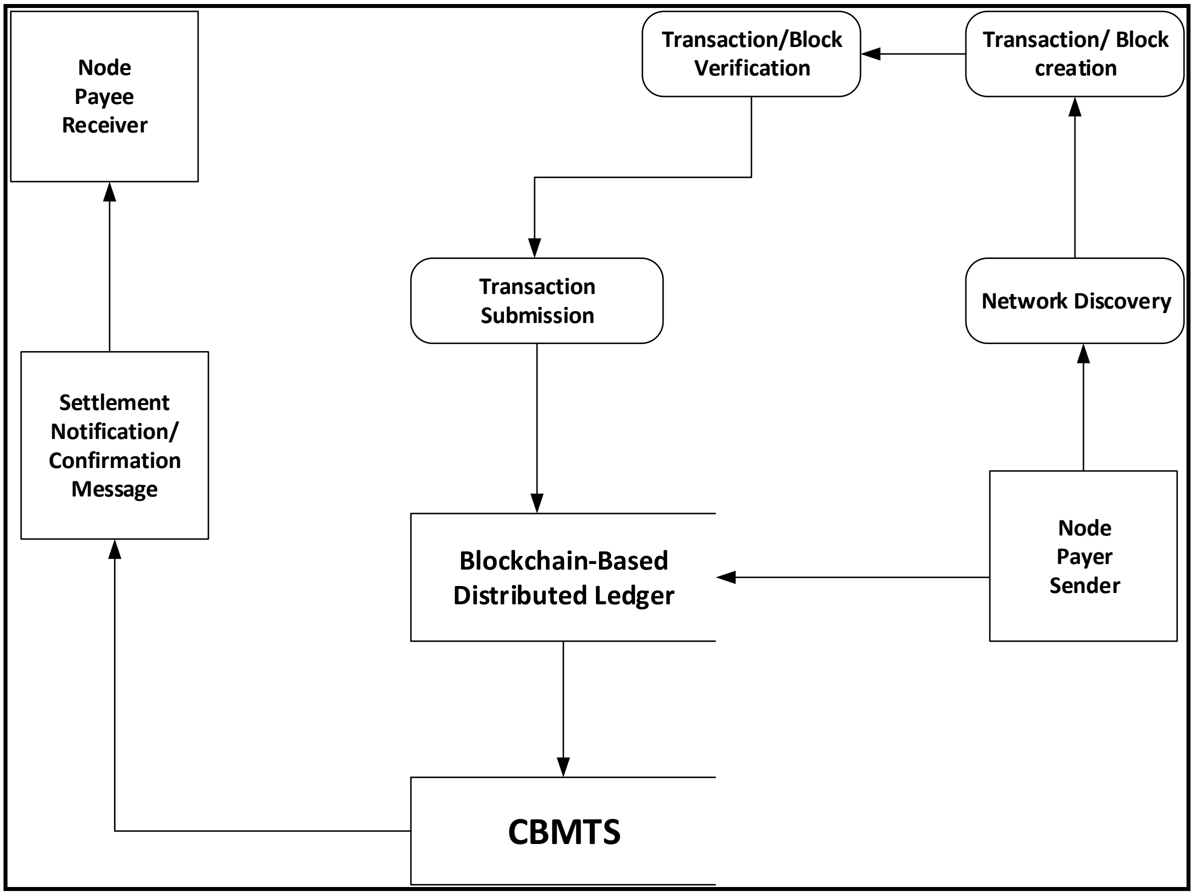

Node
Payee
Receiver
Transaction/Block Verification
Transaction/ Block creation
Transaction Submission
Network Discovery
Settlement Notification/ Confirmation Message
Blockchain-Based Distributed Ledger
Node
Payer
Sender
CBMTS

Fig 10: High Level Data Flow Diagram

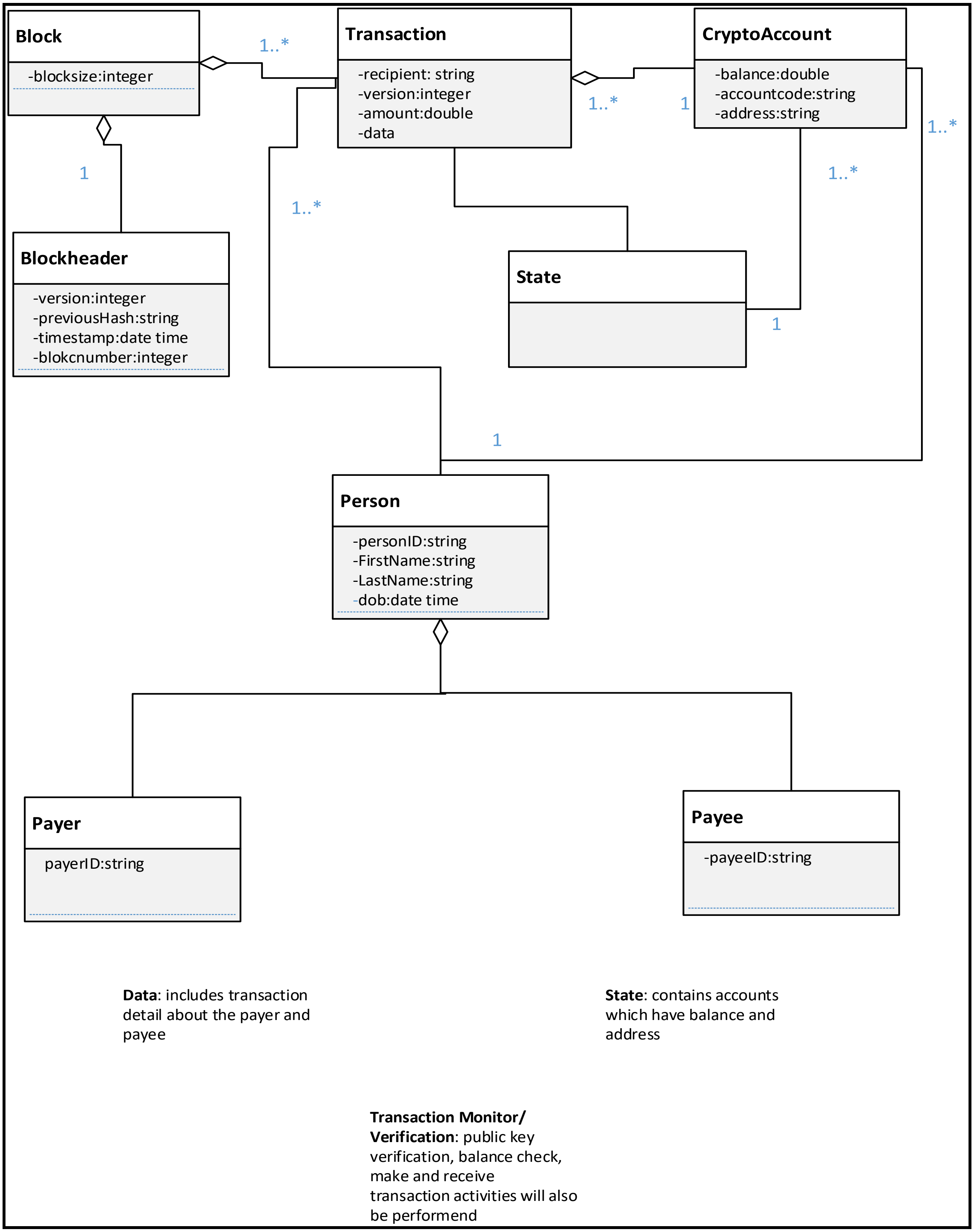
2.6.3. Data Model / Database Design

Because of the complexity, make ease and implementation challenges in a duration of three weeks the RAD document on section 4 modelled the system as simple as possible, but we strongly believe that we have to separate the **Transaction/Block** object specified in the class diagram to be normalized and the data should be stored separately for the **Transaction** *processing* and ***Block*** creation processes identified in the how a blockchain works (section 1.4) and the functional requirements section (section 3.4) including all the major components in the RAD shown in ***Fig 1***, ***Fig 2, Fig 3*** and ***Fig 4***.

Keeping the state of the blockchain is preferred opposed to choosing an Input Output type transaction logic. However, since smart contracts are powerful, keeping the state of the blockchain allows for more complex logic. State contains accounts, which have balance and address. In case of contracts, they also have to keep the executable code and storage specific to given account.

Accounts are linked to transactions - each transaction changes the state of the blockchain (balance changes, contracts function calls and others). As standard for every analysed platform, Block and Block Header are essential to blockchain technology. Block contains transactions, while the metadata (previous block's hash, timestamp, number) of the block is kept in the Block Header.

The next data model shows how we presented the CBMTS inspired from our RAD's class diagram and the Ethereum data model



Data: includes transaction detail about the payer and payee

State: contains accounts which have balance and address

Transaction Monitor/ Verification: public key verification, balance check, make and receive transaction activities will also be performend

Fig 11: CBMTS Data Model

2.7. Interface Design

The next diagram shows a sample User Interface for the CBMTS. The rest of the forms will be part of the implementation

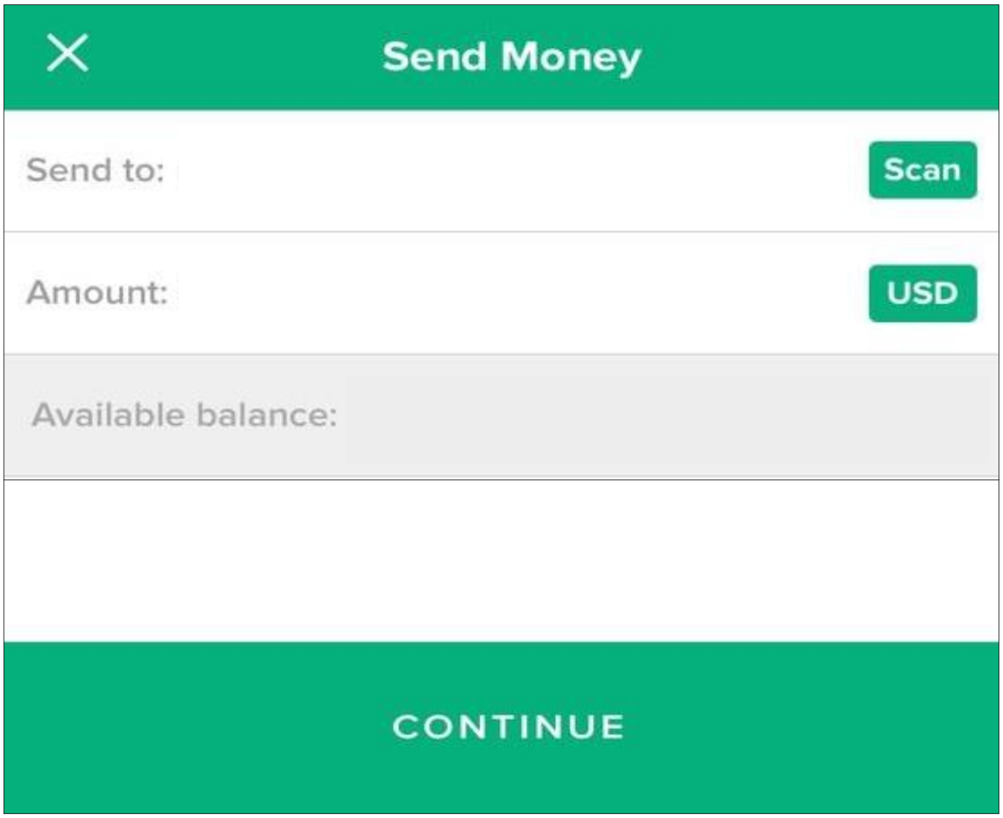


Fig 12: Sample UI form for Send money functionality

2.8. Access Control

We recommend Smart Contract-Based Access Control for systems like CBMTS based on Blockchain. A smart contract is a set of executable codes that automatically enforces agreements or terms and conditions between parties. The main benefit of utilizing a smart contract to access management in systems is that the policies of AC are enforced automatically by the smart contract, as well as offering high computing capability to reach numerous access management methods.

In our case Smart Contracts are not included in the system as clearly specified in the limitation of the proposed system.

It is therefore we don't discuss and include anything about access control and related issues in this design document.

3. Glossary

Cross-Border Money Transfer System (CBMTS) – The proposed system which transfers money where the payee and the transaction recipient are based in separate countries.

SWIFT - Society for Worldwide Interbank Financial Telecommunications.

Blocks – a collection of data containing multiple transactions over a given period of time on the blockchain network.

Chain – the cryptographic link which keeps blocks together using a 'hash' function.

Blockchain – a blockchain is a secure, global, immutable public ledger that records every transaction, chronologically, on the network.

DLT: Distributed Ledger Technology is a consensus of replicated, shared, and synchronized digital data geographically spread across multiple sites, countries, or institutions. Unlike with a distributed database, there is no central administrator.

Permissionless blockchain: anyone may join the network and read from the state stored, and write to the blockchain.

Public permissioned blockchain: a limited set of participants may write to the blockchain. Anyone may join the network and read the state.

Private permissioned blockchain: a limited set of participants may join the network, and write a new state. Only this set can read the state

Peer-to-peer Network – every node of the network is a client as well as server, holding identical copies of the application state

Cryptography – use of public key cryptography and cryptographies hash functions: essential for transparency and privacy.

Bitcoin – a Peer to Peer Electronic Cash System that would enable people to send it directly without it going in a financial institution. Blockchain is the technology that runs Bitcoin.

Coin: a cryptocurrency 'coin' is seen as a means of payment. The purpose of a coin is to act like money – to allow transactions of products and services to occur.

Nonce: is an abbreviation for "number only used once," which is a number added to a hashed—or encrypted—block in a blockchain that, when rehashed, meets the difficulty level restrictions. The nonce is the number that blockchain miners are solving for.

Decentralized: a system where no individual has ownership of the system and there is no central point of control. In the case of decentralized blockchains, the system is spread over the entire network of users. This makes it almost impossible to hack, tamper with or destroy.

Cryptocurrencies – the digital currencies that are secured using cryptography and built using blockchain technology.

Hash: the result of applying an algorithmic function to data in order to convert them into a random string of numbers and letters. This acts as a digital fingerprint of that data, allowing it to be locked in place within the blockchain.

Digital Signature: a digital code generated by public key encryption that is attached to an electronically transmitted document to verify its contents and the sender's identity

Public Address: the cryptographic hash of a public key. They act as email addresses that can be published anywhere, unlike private keys.

Private Key: a string of data that allows you to access the tokens in a specific wallet. They act as passwords that are kept hidden from anyone but the owner of the address.

Proof of Work: a consensus distribution algorithm that requires an active role in mining data blocks, often consuming resources, such as electricity. The more 'work' you do or the more computational power you provide; the more coins you are rewarded with.

Node: a copy of the ledger operated by a participant of the blockchain.

Miners: validate new transactions and record them on the global ledger (blockchain). Compete to solve a difficult mathematical problem based on a cryptographic hash algorithm.

DFD: Data Flow Diagram, high level flow of data in to the systems and out of the system

List of Figures

References

1. Blockchain technology and its effects on business models of global payment providers, Author: German Dolinski University of Twente
2. Blockchain Payment Systems, Elwood Research Series 09 December 2019
3. Blockchain Technology, Beyond Bitcoin, Michael Crosby, Google Nachiappan, Yahoo Pradhan Pattanayak, Yahoo Sanjeev Verma, Samsung Research America Vignesh Kalyanaraman, Fairchild Semiconductor
4. Blockchain for Global Payments, Oracle
5. Cross-Border Money Transfer Using Blockchain – Enabled by Big Data
6. Software Engineering Research for Blockchain Based Systems, Dr Mark Staples, Research Team Leader
7. Application Analysis on Blockchain Technology in Cross-border Payment
8. From Distributed Consensus Algorithm to the Blockchain Consensus Mechanism
9. Streamlining Cross Border Payment Processing with Blockchain, TATA Consultancy
10. A reference model for Blockchain-Based Distributed Ledger Technology, University of Tartu, Andreas Ellerbee
11. https://www.researchgate.net/figure/Deployment-diagram-of-proposed-architecture-Source-own-representation_fig9_328900688