ISSN No:-2456-2165

A Decentralized Economy Using Blockchain

Alwin K Scaria
Dept. of Computer Science and Engineering
St. Joseph's College of Engineering and Technology,
Palai, Kottayam, India

Teena Tomy
Dept. of Computer Science and Engineering
St. Joseph's College of Engineering and Technology,
Palai, Kottayam, India

Shinu Shaju
Dept. of Computer Science and Engineering
St. Joseph's College of Engineering and Technology,
Palai, Kottayam, India

Prince Abraham
Dept. of Computer Science and Engineering
St. Joseph's College of Engineering and Technology,
Palai, Kottayam, India

Abstract:- The introduction of blockchain technology has revolutionized the concept of decentralization, which is testing our preconceived notions about present social, political, and economic institutions. The rapid growth of this technology aids in unifying the world. The industrial age's centralized economic model is largely designed to generate profit for its proprietors.

An economy can be built around a community that provides services by tokenizing assets. The token represents the value of the service provided in that economy. Instead of being determined, measured, and administered for them by a central power, a decentralized economy allows communities to define what they value. Thus, shifting the locus of power outside to the individuals who keep these economies running.

Keywords: - *Tokenization, Bloc Network, Explorer*

I. INTRODUCTION

During the modern era, centralized institutions channel and structure the world's economic forces, defining, managing, and controlling how value is represented and moves within an economy.

This implies we have to trust those institutions, and in doing so, we are handing up a vast amount of power to monopolistic institutions that are prone to central-point failure. The primary goal of the industrial age's centralized economic model is to make money for its owners.

The introduction of blockchain technology has ushered in a new era of decentralization. The rapid growth of technology has begun to tear down national barriers, revealing glimpses of a better, alternate future. However, the technology is still in its youth and has flaws in terms of performance, convenience of use, and service quality. But the possibilities are limitless. The technology has already proven its power in the crypto market and soon enough it'll be able to capture most of the markets.

II. BACKGROUND

A blockchain is a digital ledger of transactions that is copied and disseminated across the network of computer systems that make up the blockchain. Every block on the blockchain contains a collection of transactions, and whenever a new transaction occurs on the blockchain, each participant's ledger is updated. DLT stands for Distributed Ledger Technology, which is a decentralized database managed by a group of individuals (DLT).

Blockchain is a distributed ledger system in which transactions are recorded using a hash, which is a non-changeable cryptographic signature.

Decentralization in blockchain refers to the transfer of control and decision-making from a centralized entity (person, organization, or group thereof) to a distributed network. Decentralized networks aim to limit the amount of trust that participants must place in one another and to prevent them from exerting power or control over one another in ways that harm the network's performance.

The concept of decentralization is not new. Three basic network designs are often examined when developing a technology solution: centralized, distributed, and decentralized. While decentralized networks are frequently used in blockchain technology, a blockchain application cannot simply be classified as decentralized or not. Rather, decentralization should be applied to all components of a blockchain program on a sliding scale. Greater and fairer service can be accomplished by decentralizing resource management and access in an application. Decentralization has some drawbacks, such as lower transaction throughput, but the benefits of enhanced stability and service levels outweigh the drawbacks.

III. OBJECTIVE AND SCOPE

This project aims to build a decentralized economy around micro-communities where individuals can have a peer-to-peer exchange of assets using tokens. The transactions are

carried out with the use of an ERC-20 token, which represents the value of the assets being traded. The people who create value in this system are compensated with tokens. As the network expands, the tokens gain in value for all of its holders. This contributes to the development of a long-term and equitable economic network in which profits are dispersed fairly.

We're proposing a solution that allows for a value-based trade network. Different simultaneously running on the blockchain network. Each service may function as its para chain, with its own set of rules and tokens. Through the relay chain, the services communicate with one another and exchange data.

Currently, the world is being run by centralized organizations. Imagine a self-regulating network of organizations that do not require a third party to trust each other. This is what Bloc Network is all about, a decentralized blockchain network that facilitates trustless transactions and smart contracts between organizations and individuals. It aids in creating different-sized micro-communities and assuring interoperability.

IV. LITERATURE SURVEY

A. Cosmos Network [1]

Cosmos is a network of independent parallel blockchains that are each powered by classical BFT consensus algorithms like Tendermint 1 [1].

The first blockchain in this network will be the Cosmos Hub. The Cosmos Hub connects to many other blockchains (or zones) via a novel inter-blockchain communication protocol. The Cosmos Hub tracks numerous token types and keeps a record of the total number of tokens in each connected zone. Tokens can be transferred from one zone to another securely and quickly without the need for a liquid exchange between zones because all inter-zone coin transfers go through the Cosmos Hub [1].

This architecture solves many problems that the blockchain space faces today, such as application interoperability, scalability, and seamless upgradability. For example, zones derived from Bitcoin, Go-Ethereum, CryptoNote, ZCash, or any blockchain system can be plugged into the Cosmos Hub. These zones allow Cosmos to scale infinitely to meet global transaction demand. Zones are also a great fit for a distributed exchange, which will be supported as well[1].

Cosmos is not just a single distributed ledger, and the Cosmos Hub isn't a walled garden or the center of its universe. We are designing a protocol for an open network of distributed ledgers that can serve as a new foundation for future financial systems, based on principles of cryptography, sound economics, consensus theory, transparency, and accountability [1].

B. ICON Foundation [2]

This paper outlines our vision and philosophy of the ICON Project and details on the supporting proprietary technologies that have been in development over the past few years. More importantly, we discuss actual use cases with dozens of reputable institutions that are already in the ICON Network. This demonstrates our progress beyond the initial concept stage and validates our team's strong execution ability; a major factor that differentiates ICON from the majority of blockchain projects today [2].

ICON is inspired by Gilles Deleuze and Felix Guattari's rhizome – "the world with no center point and the world where any point is a mere connection to other points." ICON is a connector of disparate cryptocurrencies with different blockchain governances, and a connector of the crypto world to our real world. ICON embraces the new and the unfamiliar, the principle of radical inclusion – accepting new ideas and decisions made by the new republic established by ever-changing crypto-to-real world connections [2].

C. Avalanche [3]

Avalanche is a high-performance, scalable, customizable, and secure blockchain platform. The overarching aim of Avalanche is to provide a unifying platform for the creation, transfer, and trade of digital assets [3].

Scalable Avalanche is designed to be massively scalable, robust, and efficient. The core consensus engine can support a global network of potentially hundreds of millions of internet-connected, low and high 25 powered devices that operate seamlessly, with low latencies and very high transactions per second [3].

Secure Avalanche is designed to be robust and achieve high security. Classical consensus protocols are designed to withstand up to f attackers, and fail when faced with an attacker of size f + 1 or larger, and Nakamoto consensus provides no security when 51% of the miners are Byzantine. In contrast, Avalanche provides a very strong guarantee of safety when the attacker is below a certain threshold, which 30 can be parametrized by the system designer, and it provides graceful degradation when the attacker exceeds this threshold. It can uphold safety (but not liveness) guarantees even when the attacker exceeds 51%. It is the first permission less system to provide such strong security guarantees [3].

Decentralized Avalanche is designed to provide unprecedented decentralization. This implies a commitment to multiple client implementations and no centralized control of any kind. The ecosystem is designed to avoid 35 divisions between classes of users with different interests. Crucially, there is no distinction between miners, developers, and users [3].

ISSN No:-2456-2165

V. MODULES

The system is divided into nine modules:
(1) Bloc Architecture
(2) Main Chain Network
(3) sub-chain Network
(4) Smart Contracts
(5) Network Roles
(6) Governance Model
(7) Consensus
(8) Bloc Token
(9) Wallet

1) ***Bloc Architecture:*** We will be creating our own blockchain network called the Bloc Network. As illustrated in Fig. 5.1.1, a single main chain houses several sub-chains forming micro-communities that interact based on shared value.

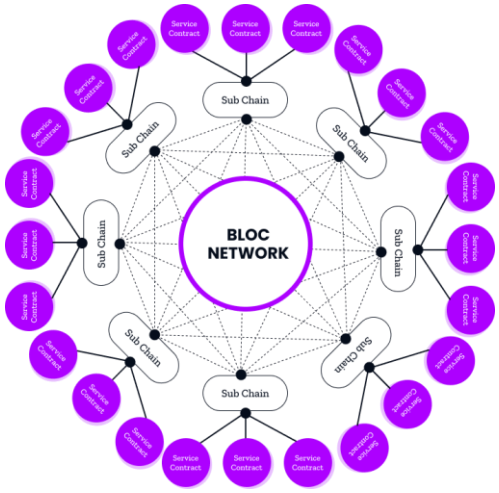

Fig. 5.1.1. Bloc Architecture

Sub-chains communicate with one another to share services and values. Self-evolved algorithms and indigenous value systems regulate sub-chains.

2) ***Main Chain Network:*** It's a decentralized network of parallel blockchains that operate independently of one another. The main chain network is made up of an arbitrary number of sub-chain networks or micro-communities. It displays all sub-chains, as well as the events, transactions, and accounts that occur within each one. Users must first register on the Main Chain before being able to use the platform.

The main chain network is a private blockchain built on the Avalanche platform. The network consists of three chains bundled in one node. Platform, Asset, and Contract chains. The primary network is used to run contracts and retrieve information for the explorer. When we're ready to invite public validators to join the network, we'll transfer it to a subnet.

3) ***Sub-chain Network:*** Interoperability is ensured on the sub-chain level. sub-chains are separate from the main chain. They are their own micro-communities within the main chain network. Members of the main chain network can request access to join a sub-chain network.

Members who have access to a sub-chain may view and participate in all of the events that take place inside of it. A member can sign up either as a service provider or as a regular member.

4) ***Smart Contracts:*** There are mainly three contracts. One is an ERC20 Contract for tokenization. An NFT Contract for the asset transactions on the blockchain network and a Governance Contract for the smooth sailing of the network.

5) ***Network Roles:*** There are mainly four roles. Members: Members can create new sub-chains, request membership and propose changes to existing sub-chains, and can use the services deployed on the network.
Service Providers: They can deploy services and propose changes to existing sub-chains.
Governance Members: Proposals and membership requests are accepted or rejected by the means of voting.
Network Owner: They are the account holder(s) who created the genesis block.

6) ***Governance Model:*** Every member can create and own a subnet. Any member can propose the creation of a new subnet. As a result, they are the default subnet owner. They can stake their tokens on the members who will be governance members once the subnet has enough members. The members who hold 40% of all votes after staking would be designated governance members. They have the authority to accept or reject an offer.

A member can start a sub-chain and make a suggestion. Members who may develop proposals to provide services are known as service providers. In order to deliver their services, they must first be accepted into a certain sub-chain. In all the sub-chains, the network owner is automatically considered a governance member.

7) ***Consensus:*** We're currently testing a consensus that broadcasts each transaction as a separate block. There is byzantine fault-tolerance performed here, but no specific miners are engaged.

8) ***Bloc Token:*** This is the bloc network's native currency. Every service relies on this token to deliver services and transactions to all of its users. Whenever a member interacts with a network service. For such services, they are either taxed or given bloc tokens. These tokens can also be used to pay for other services on the same network.

9) ***Wallet:*** An interface for storing and transferring tokens. It's handled in-platform. The tokens stored in these wallets are

used for in-network services. It can be used to trade tokens in the future.

VI. TECHNOLOGIES USED

A. Ethereum Virtual Machine

The EVM's physical instantiation can't be described in the same way that one might point to a cloud or an ocean wave, but it does exist as one single entity maintained by thousands of connected computers running an Ethereum client [4].

The Ethereum protocol itself exists solely for the purpose of keeping the continuous, uninterrupted, and immutable operation of this special state machine; It's the environment in which all Ethereum accounts and smart contracts live. At any given block in the chain, Ethereum has one and only one 'canonical' state, and the EVM is what defines the rules for computing a new valid state from block to block [4].

B. Avalanche Blockchain

Avalanche is an open-source platform for deploying decentralized applications and enterprise blockchain installations in a unified, highly scalable environment. Avalanche is the first decentralized smart contracts platform designed for global finance, with near-instant transaction completion. Because Solidity works out-of-the-box, Ethereum developers can easily build on Avalanche.

A key difference between Avalanche and other decentralized networks is the consensus protocol. Over time, people have come to a false understanding that blockchains have to be slow and not scalable. The Avalanche protocol employs a novel approach to the consensus to achieve its strong safety guarantees, quick finality, and high throughput without compromising decentralization.

C. ERC721

ERC721 enables the use of a standard API for NFTs within smart contracts. This standard offers the fundamental functionality for tracking and transferring NFTs.

After the examination of individual ownership and transaction of NFTs, as well as consignment to third-party brokers/wallets/auctioneers ("operators"). NFTs can denote ownership of either digital or physical assets. After evaluating a wide range of assets, and it's assured that anyone can come up with many more:

Physical property — houses, unique artwork
Virtual collectables — unique pictures of kittens, collectable cards
"Negative value" assets — loans, burdens, and other responsibilities

In general, all houses are distinct and no two kittens are alike. NFTs are distinguishable and you must track the ownership of each one separately [5].

D. Truffle

A world-class development environment, testing framework, and asset pipeline for blockchains based on the Ethereum Virtual Machine (EVM), with the goal of making life easier for developers. You get the following with Truffle.

Smart contract compilation, linking, deployment, and binary management are all built-in. Contract testing that is automated for speedy development. Framework for deployment and migrations that is scriptable and flexible. For deployment to any number of public and private networks, network administration is required. Using the ERC190 standard, package management with EthPM and NPM. Direct contract communication via an interactive console. Configurable build process with tight integration support. Script runner that runs scripts outside of the Truffle environment.

E. Solidity

Solidity is an object-oriented, high-level language for implementing smart contracts. Smart contracts are programs that govern the behavior of accounts within the Ethereum state [6].

Solidity is a curly-bracket language. It is inspired by C++, Python, and JavaScript, and was created with the Ethereum Virtual Machine in mind (EVM). In the linguistic impacts section, you may learn more about the languages that have influenced Solidity.

Solidity is statically typed, supports inheritance, libraries, and complex user-defined types among other features [6].

F. InterPlanetary File System

The InterPlanetary File System or IPFS is a distributed system for storing and accessing files, websites, applications, and data. It's a peer-to-peer network that offers decentralization therefore it's not subjected to central point failure. It supports a resilient internet. IPFS files aren't ideally censored and they're always accessed from the nearest source, speeding things up.

IPFS is a peer-to-peer storage network. Content is accessible through peers located anywhere in the world, that might relay information, store it, or do both. IPFS knows how to find what you ask for using its content address rather than its location.

There are three fundamental principles to understanding IPFS: Unique identification via content addressing. Content linking via directed acyclic graphs (DAGs). Content discovery

via distributed hash tables (DHTs). These principles build upon each other to enable the IPFS ecosystem.

G. web3.js

The web3.js library is a collection of modules that contain functionality for the ethereum ecosystem. web3-eth is for the ethereum blockchain and smart contracts. web3-shh is for the whisper protocol, to communicate p2p and broadcast. web3-bzz is for the swarm protocol, the decentralized file storage. web3-utils contains useful helper functions for Dapp developers.

H. ReactJS

ReactJS is a user interface development library written in JavaScript. React makes creating interactive UIs a breeze. Create basic views for each state of your project, and React will update and render the appropriate components as your data changes.

Declarative views improve the predictability and debuggability of code. Compose encapsulated components that handle their own state to create complicated user interfaces.

You can simply transmit rich data through your app and keep state off of the DOM because component functionality is written in JavaScript rather than templates.

I. NodeJS

As an asynchronous event-driven JavaScript runtime, Node.js is designed to build scalable network applications. Many connections can be handled concurrently. Upon each connection, the callback is fired, but if there is no work to be done, Node.js will sleep.

This is in contrast to the more typical concurrency approach of today, which uses OS threads. Thread-based networking is wasteful and complex to implement. Furthermore, because Node.js has no locks, users do not have to worry about the process becoming deadlocked. Because almost no Node.js function does I/O directly, the workflow never comes to a halt. Because nothing blocks, Node.js is an excellent choice for building scalable systems.

J. MongoDB

MongoDB is a general-purpose, document-based, distributed database built for modern application developers and the cloud era.

A record in MongoDB is a document, which is a data structure composed of field and value pairs. MongoDB documents are similar to JSON objects. The values of fields may include other documents, arrays, and arrays of documents.

Many computer languages provide native data types that correlate to documents (i.e. objects). The use of embedded documents and arrays reduces the requirement for costly joins. Fluent polymorphism is supported via dynamic schema.

K. ExpressJS

Express is a Node.js web application framework that offers a comprehensive range of functionality for both web and mobile apps.

Using a variety of HTTP utility methods and middleware, you can quickly and easily build a powerful API. Express adds a thin layer of basic web application functionality without obscuring the Node.js capabilities you already know and appreciate.

Express supports many popular frameworks.

L. HDWallet Provider

The Truffle HDWallet provider is a convenient and easy way to configure network connection to ethereum through infura.io (or any other compatible provider). Features like event filtering and transaction signing are an added bonus.

When deploying a contract, you eventually need to sign a transaction with an account (with a private key). This account management process is handled by the provider, and in this case by the HDWallet provider.

VII. IMPLEMENTATION

Smart contracts were written using Solidity for governance, asset, token control, and service management. These contracts can be deployed on any EVM-based platform. Then we created a genesis node using AvalancheGo and GCC modules on a virtual private server. Several other nodes on multiple virtual private servers were then added to this network, creating a private blockchain network i.e. the Bloc Network.

The network was configured using Postman and a custom NodeJS server over JSON-RPC. Deployed all the contracts onto this Bloc Network. Multiple services were deployed on the network. sub-chains were created on this network using the governance contract and tokenized the network using the token contract.

And an NFT marketplace was deployed onto the network which runs on the asset contract. Each transaction in the network is checked for Byzantine fault-tolerance consensus and broadcasted through the network. ERC-20 standard-based Bloc tokens are loaded into the in-platform wallet and are used throughout the network.

The frontend for the NFT marketplace called the bloc market was created using ReactJS. The backend was created using ExpressJS for the NodeJS server, MongoDB for database management, web3.js and truffle for smart contract interaction, and HDWallet for in-platform wallet functions.

IPFS is used to store token URIs which are to be used with ERC-721 NFT asset contracts.

A bloc explorer is also implemented which is used to display all the transactions carried out on the Bloc network in a user-friendly and comprehensive manner. Each and every block can be viewed and verified. Search can be used to verify transaction hash, block number, or a contract address. It can also be used to view the sub-chains, market cap, and the active validators of the network. Members can perform all governance functions through the explorer.

VIII. CONCLUSION

We have successfully tested and proved that value can be fairly equated among self-governing micro-communities without the involvement of centralized intervention. Our governance systems make sure that every member is given access to every service equally. Every member can propose new changes to the system and expect a fair evaluation. Thus, decentralizing the management gives access to resources in an application and helps to achieve greater and fairer services.

Smart contracts immediately address the issue of user trust. Users may also easily build and deploy decentralized apps on our network, which improves the user experience. In an era when bitcoin has become the global standard, the possibilities are limitless. Humanity is stepping into a new era of sustainability through blockchain. There are no centralized powers that can withstand the rise of decentralization and usher in a new era of industrial decentralization.

REFERENCES

[1]. “Cosmos: A Network of Distributed Ledgers Whitepaper” - Jae Kwon and Ethan Buchman.
[2]. “ICON Hyperconnect The World - ICON Whitepaper”.
[3]. “Avalanche Platform Whitepaper” - Kevin Sekniqi, Daniel Laine, Stephen Buttolph, and Emin Gun Sirer.
[4]. “Ethereum Development Document” - Ryan Cordell and Sam Richards.
[5]. William Entriken, Dieter Shirley, Jacob Evans, Nastassia Sachs, “EIP-721: ERC-721 Non-Fungible Token Standard,” Ethereum Improvement Proposals, no. 721, January 2018.
[6]. Solidity 0.8.6 Documentation 2021.