IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

Agricultural Insurance on Blockchain

[1] M Kashif, [2] Vivek T A, [3] U Sohini Sahukar, [4] M S Nikhil

[1] BE Undergraduate, [2] BE Undergraduate, [3] BE Undergraduate, [4] BE Undergraduate
[1] Computer Science,
[1] Cambridge Institute of Technology, Bengaluru, India

Abstract: Farmer suicide rates have been on the rise since the 90s and a major reason for this is the inability to pay back loans due to failure of crops. Also, the climate inherently not being under the control of the farmer makes the situation even worse. So this is where, Bulwark, a blockchain based insurance platform gives them the financial security they need. Bulwark aims to provide a decentralized insurance protocol that is simple, secure, quick and accurate. We offer to absorb risk in exchange for premium payments in case of disasters, particularly droughts and floods. Our policies are designed to be simple and we use accurate weather data to detect catastrophes and process claims efficiently. The farmers get to choose their own coverage amount and frequency at which they wish to pay their premiums, according to their needs. Being built on Ethereum, the policies and transactions are immutable that makes it difficult to be hacked and no human intervention is needed to process transactions resulting in zero human error. And since it's the second most valued digital currency, all major wallets offer support. So distributing the risk by spreading the insurer base to a wide enough area where climatic changes in one location doesn't affect the other would be the future business model. At the end, our goal is to provide farmers an efficient and an anti-fraudulent way to secure their harvest no matter what. Even if we get to save one farmer's life with Bulwark, that's an absolute win for us.

Index Terms **- Blockchain, Smart Contract, Insurance, Agriculture, Weather**

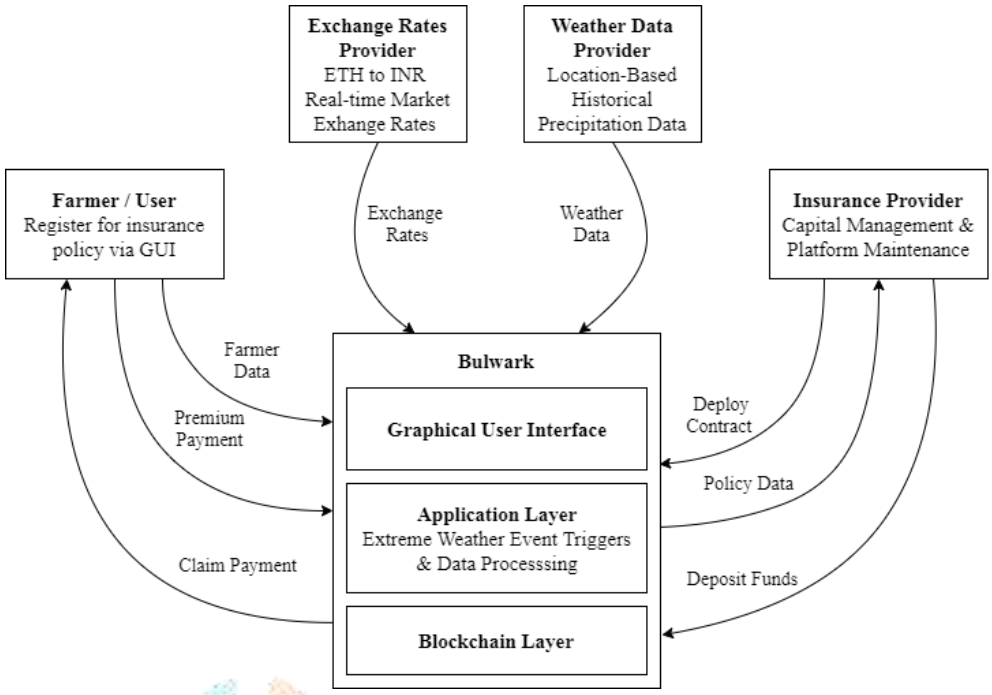
I. INTRODUCTION

Bulwark aims to provide a decentralized insurance protocol for farmers by utilizing blockchain technology and satellite weather data. Speed and precision are the most important factors we considered while building this. To achieve it, we chose the Ethereum network, because it is one of the most popular and stable blockchain networks that serves the ability to execute smart contracts autonomously. The platform is built on top of this, as a web application with MERN stack. Node JS serves as a backend framework, to facilitate REST API calls and React serves as a frontend GUI framework for the user to interact with the application. Ethereum serves as an abstract foundational layer for a Turing complete language that lets the usage of smart contracts. The protocol is designed to be simple, universal and modular. The network works on the digital currency Ether, that acts as an exchange channel for all transactions. Every time a transaction is initiated, the code in the smart contract is activated and executed that performs read and write operations within the internal storage autonomously. Each transaction is a signed package of data that is to be sent from an account. Each transaction is a package that consists of the recipient address, the signature of the sender, the amount in ether that is to be exchanged and extra data that is to be transmitted.

Our contribution in this paper is the design and implementation of a blockchain framework for agricultural insurance by creating smart contracts and a user interface application utilising precise location-based weather data.

II. PROPOSED MODEL

Our model is designed to make the entire insurance process simple and quick, yet secure. It mainly involves three layers. Blockchain layer is where the Ethereum virtual machine and the smart contract resides. The middle layer is the application layer handles all the calls to and from the blockchain layer. It also handles the several API calls that handle the processing of data. The calls range from weather data extraction from Meteosat and exchange rates of Ethereum. It also involves the pre-processing and validation of data before sending it to the blockchain. The top most layer is the layer that is visible to the user. All the transactions will be initiated from this layer.



Proposed System Architecture

III. Technologies and Frameworks Used

Solidity is a statistically typed object-oriented language to write smart contracts. It has complex user-defined data types and supports inheritance. Smart contract is a program that executes autonomously on the Ethereum virtual machine when a transaction is invoked. Web3 JS is a collection of libraries that allows interaction with local and remote nodes using WebSocket or HTTP. It is needed to connect to a node which sends the transaction to the Ethereum network through JSON RPC.

Truffle is a framework that supports custom deployments, library linking and complex Ethereum applications. It also provides the ability to write automated test for the contracts. In addition to this Ganache is used in a local environment to simulate actual Ethereum accounts. It also provides nodes that help in testing of decentralized applications.

Node JS is a backend framework that is used to execute JavaScript in server-side development. Since it is open sourced, supports all platforms and has an architecture that is able to handle asynchronous input and output, it is a framework of choice. Express is used to handle APIs that power the server-side application. It helps in making the application highly robust and scalable. Combined with MongoDB, a NoSQL database to handle user management for the user interface using JSON like documents.

React JS is a frontend JavaScript library that is used to build functional user interfaces. Our platform being a single page application, this is the best framework that suits the needs. Multiple JSX components for core functionalities are deployed using this framework. It also makes it easier to develop applications for mobile devices as well. Sass is a pre-processor scripting language for CSS. Along with HTML, it serves a perfect blend to provide a clean user interface, ranging from text formatting, button mapping and image placement.

JWT (JSON Web Token) is used for user authentication. It is an open standard RFC 7519 to securely transmit information between the user and the server. It supports Single Sign On with small overhead. The authentication information is digitally signed using HMAC and encrypted using keys with RSA.

IV. Core Processes

4.1 New Policy:

A farmer is allowed to sign up for a particular crop insurance at a targeted premium amount and suitable duration. Aadhar number and survey number is taken for identification of the user and the land. The location is read using the browser's built-in geo locating function, the user is also given an option to enter the location coordinates manually. This location is also stored in the smart contract for further processing. The user / farmer gets to choose their own coverage amount and interval at which they choose to pay, the premium would be calculated based on these two factors. All this information is stored in then stored on the chain. On successful sign up, the calculated premium amount is deducted from the farmer's account and if sign up is unsuccessful if then premium will not be paid.

4.1 Pay Premium:

The pay premium feature facilitates premium payment during sign up and subsequent premium payments of insured farmers only. Only the account address of the farmer needs to be furnished during subsequent premium payment, but all required documentation needs to be submitted at signup. The transfer of the amount from the farmer's account to the contract is the result of successful completion of the transaction. The contract fails if an uninsured person attempts to pay the premium and the premium amount will always be the same as during signup, neither the farmer can change that nor the insurance company or the owners of the smart contract.

4.2 Claim Processing:

When a user files a claim on the platform, the location that is stored in the contract when the user signed up is retrieved and all the weather data is accumulated at that location for the past 30 days, precipitation amount in that particular location and a standardised precipitation index is calculated dynamically. SPI serves as a very good measure as a drought index and the impact is then classified based on the severity ranging from 0 to 6 where 0 is a drought and 6 is a flood. This value is sent to the smart contract and it analyses the severity and processes the claim payment that is needed based on the value. If the claim is successful then they receive the coverage amount. If there is no impact detected by the climate, then the claim is rejected. This accounts for zero human error during the processing.

SPI Classification

CLASS	CONDITION	RANGE
0	EXTREMELY WET	[> = +2.00]
1	SEVERELY WET	[+1.50 ~ +1.99]
2	MODERATELY WET	[+1.00 ~ +1.49]
3	NEAR NORMAL	[-0.99 ~ +0.99]
4	MODERATE DROUGHT	[-1.00 ~ -1.49]
5	SEVERE DROUGHT	[-2.00 ~ -1.50]
6	EXTREME DROUGHT	[< = -2.00]

V. FUTURE SCOPE

Our current model can scale excellently by diversifying the insurer base by spreading the area of origin that is insured. The risk absorbed would be extremely high when the insured area is limited to a particular geographical location. Venturing into several geographical locations provides a way to diversify the risk and indeed support multiple places. The probability of a catastrophe occurring at several different areas far from each other is highly unlikely. This will ensure that there is proper capital availability when needed. This can be achieved by calculating a threshold of maximum number of insurers in one area. This threshold dynamically changes over time as the area covered by insurance increases or decreases. Dynamic calculation is the most important factor since the user base is always changing. The future scope of this paper also extends to insurance over other products that are already being insured in the traditional system. The crucial factor in determining the feasibility would be the dependency on human dependent processes involved in the system.

VI. CONCLUSION

With the rise of farmer suicide rates as a result of inability to pay back loans due to failure of crops, there is an urgent need for a new system that centres around the farmers need for financial safety. This paper introduces a blockchain based application, that aims to make the process of obtaining agricultural insurance seamlessly simple for a farmer. By developing the application on a blockchain, high security of transactions, inexpensive processing fee and lower transaction times are achieved. Above all it dynamically monitors the weather data which fosters automatic claim settlement alleviating fraud in the claim settlement process.

VII. ACKNOWLEDGEMENT

We express our sincere gratitude to our guide Prof. Priyadarshini M, for her guidance. We also wish to extend our deep gratitude to family and friends for all the support and encouragement in the successful completion of our project.

REFERENCES

[1] https://en.wikipedia.org/wiki/Blockchain
[2] https://101blockchains.com/history-of-blockchain-timeline/
[3] https://medium.com/@ChainTrade/10-advantages-of-using-smart-contracts-bc29c508691a
[4] https://www.nasdaq.com/articles/using-blockchain-track-assets-proof-ownership-2016-11-30
[5] https://skyrepublic.com/business-cases/insurance/
[6] https://climatepolicyinitiative.org/wp-content/uploads/2019/10/Blockchain_instrument-analysis.pdf
[7] https://coincentral.com/what-is-ethereum/
[8] https://en.wikipedia.org/wiki/Solidity
[9] https://www.trufflesuite.com/truffle
[10] https://www.trufflesuite.com/docs/ganache/overview
[11] https://www.trufflesuite.com/ganache
[12] https://www.mycryptopedia.com/what-is-web3-js-a-detailed-guide/
[13] https://en.wikipedia.org/wiki/Node.js
[14] https://en.wikipedia.org/wiki/React_(web_framework)