Full WHITEPAPER

SARA

New generation reviewing platform based on the blockchain technology

Contents

Abstract

The blockchain is an incorruptible digital ledger of economic transactions that can be programmed to record not just financial transactions but virtually everything of value."

– Don & Alex Tapscott, authors Blockchain Revolution (2016).

Based on the essentials of Blockchain Technology, We, the team SARA decided to design a platform in the necessitous field such as paper publication, review industry, supply chain management, healthcare and education services on (say) SARA network.

In recent years, many organizations have sprung up which publish journals submitted to the conferences organized by them. Such prestige system is a complex socio-economic system perpetuated by journals and researchers themselves which is not transparent. The traditional system is self-reinforcing and is very difficult to remove. Hence there is a need of a new reputation ecosystem which can assure the credibility of the papers published and gain the trust. The system aims at creating Decentralised Autonomous Organisation (DAO) which encourages peer review and creates its own reputation ecosystem to provide an alternative to the current prestige system that dominates academic publishing with detrimental consequences. Information is stored on the Ethereum blockchain to allow version control of documents and provide redundancy and resiliency to the information in the network.

The review platform of the SARA network implements the Blockchain technology in order to ensure feedback legitimacy and as a backbone for the economic model which ensures that all feedback is genuine and tamper proof.

During the development of the peer review platforms, SARA network noticed the importance of the review or audit activities of the processes or the standards in Traceability system so that we have decided to utilize SARA platform in Supply Chain Management which mainly focuses on Traceability.

SARA tokens shall be the native token of the system, with the purpose of tokenizing the review industry as well as the Supply Chain Management.

Introduction

To improve editorial standards, it is essential to understand the current status quo and obstacles facing journal editors and others in the peer-review and publishing process.

- Dr. Lorraine Ferris, President, World, Association of Medical Editors (WAME)

SARA Paper Review is a decentralised publication/conference system. In recent years, many organizations have sprung up which publish journals submitted to the conferences organized by them. Such prestige system is a complex socio-economic system perpetuated by journals and researchers themselves by rewarding publication in prestigious journals and punishing a lack thereof. It is self-reinforcing and is very difficult to remove. Hence there is a need of a new reputation ecosystem which can assure the credibility of the papers published and gain the trust of the people who will be referring such papers for their research. The system allows reputation to be accrued by users and uploaded academic papers by creating a reputation ecosystem that can be drilled down into the show the number of papers uploaded and their quality, number of citations the paper receives, number of reviews performed and their quality, number of decisions participated in and what decision was made. The system aims at creating Decentralised Autonomous Organisation which encourages peer review and creates its own reputation ecosystem to provide an alternative to the current prestige system that dominates academic publishing with detrimental consequences. Information is stored on the Ethereum blockchain to allow version control of documents and provide redundancy and resiliency to the information in the network.

Introduction

A Review site is a website on which reviews can be posted about people, businesses, products, or services.

-Wiki

pedia

Blockdoor Review Platform implements Blockchain technology in order to ensure feedback legitimacy and as a backbone for the economic model which ensures that all feedbacks are legitimate and genuine. Reviews received are saved in the Blockchain, which guarantees that those comments can't be edited at a later date. The internal token of the system, called SAT, helps motivate participants to take their reviews and all related actions seriously.

Review means the evaluation and description of the user's experience upon the use of a product or service. Although reviews and reviewers play a very important role in stimulating demand for products and services and building image for service providers, in practice the review results are often manipulated by Corporations, large companies through marketing techniques, or even a direct cooperation with the review system providers, making it difficult for users to trust the results of product reviews. The solution to this problem is to build Blockdoor review Platform for reviewing on Blockchain, utilizing blockchain's immutability to produce the best possible transparency and create conditions for reviewers to benefit from providing quality reviews, as well as easily and directly interact with users and providers of products or services that are currently trustless.

These problems can be divided into following groups:

How can participants be so sure that reviews are not being edited or deleted?

When users post a review, they trust that the platform will not change, edit or delete it. The problem faced is a problem of trusting third party.

Introduction

How can a company be so sure that reviews are being written by genuine clients and not competitors?

Doubtlessly, reviews on the internet are important and can impact the image of a company. Besides the obvious function of feedback, reviews affect the popularity of the business among new and potential clients. This eventually attracts competitors who will try to tarnish the reputation of this rival. Meanwhile , companies have no way of knowing who writes the comments/reviews.

Economical problems can be described as follows :

User Motivation:

Writing a quality review is not a simple task. It requires time and effort , it involves detailed descriptions of the pros and cons of a product and a conclusion . In the light of this , the following question arises: “What does the author a review receive as a reward for their efforts?” Of course , there are people who do not need to be motivated to write a review , and yet the majority of users would prefer to be compensated for their time.

Review Credibility:

If there is no incentive to write reviews , could it be that some parts of reviews are created by means of artificial intelligence? Nowadays it is not impossible to use artificial intelligence to generate several hundred of short texts that would look like they were written by a human , or simply just buy reviews.

Our platform implements several innovative technologies , most of which are based on the blockchain technology. Smart contracts let us publicly save thousands of reviews , thus making it impossible to delete or modify them. Our economic model involves the ETH and SAT tokens and helps motivate users to leave only genuine feedback. In addition , we use mathematical formulas for calculating rewards that make submitting fake reviews economically unprofitable.

Economical Aspects

This section explains the functioning of the SARA platform from an economic stand point. During the first stage of development our clients will be mainly the organisations related to publication/conferences who wish to use the SARA network, the companies who get registered with Blockdoor to have legitimate feedbacks and the clients connected to supply chain networks. Our secondary plan is to conduct ICOs where our clients will be mainly the companies that have successfully completed a successful ICO and their investors. The amounts raised can be huge taking into account the non-transparent legal status of many companies and almost complete lack of control from most governments.

Tokens(ETH and SAT)

Our project is different from most other Blockchain projects since two tokens are required for the system to function smoothly. ETH is the Ethereum token which is linked to our own created token SAT for internal mechanism vital to the functioning of the platform.

ETH is mainly the transaction fee for every transaction on Ethereum Blockchain whereas SAT token will be exclusively used inside the SARA network. This approach allows for a stable exchange rate. All interactions between platform, users, stakeholders and companies is based on this approach.

We understand that as more and more people use our platform, the demand of the SAT token will become higher than the supply and it’s price on exchanges will start to grow. The system design ensures that the currency rate of SAT should not be affected by the market.

To accomplish this, we have decided to make 1 SAT equals to 10 INR for now. It will be later decided on Crowdfunding.

Design of the SARA Platform

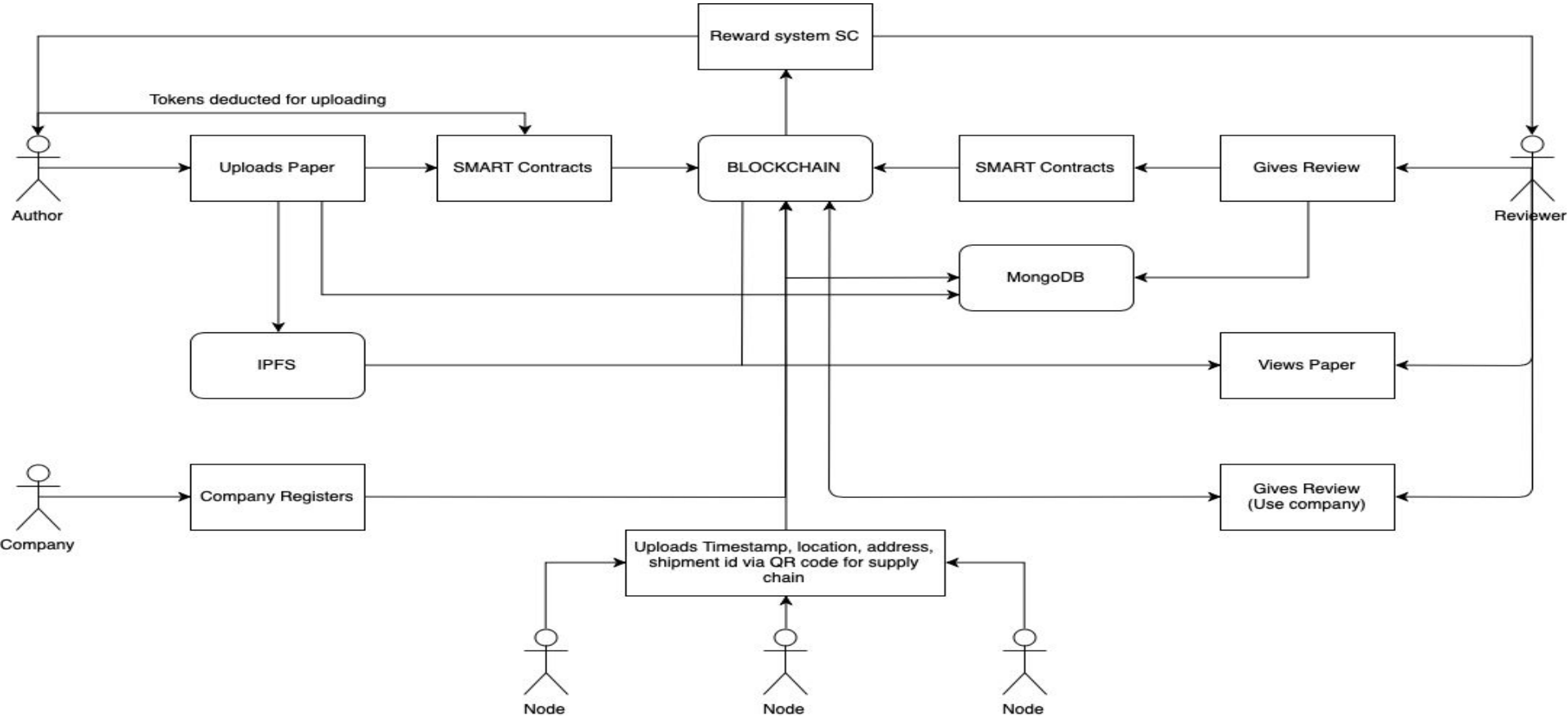
The main public Ethereum decentralized network, which uses the concept of smart contract, is chosen to serve as the decentralized settlement layer of the SARA Platform with SAT Token and provide transparency of transactions, submission of documents and reviews. SARA Platform introduces a hybrid architecture approach, bridging between public Ethereum chain and the high performance, scalable private side-blockchain transaction services for scalable
interactions with the SAT token. The key reasons for not using a purely public chain architecture are scalability and transaction fees. The current Ethereum network's capabilities do not allow us to launch and scale the service globally due to several issues, including:

- The Ethereum network currently is theoretically capable of handling about 10 transactions per second , which is definitely not enough for the scale of millions of users (or even hundreds of thousands of users).
- Ethereum's transaction confirmation times are significantly delayed - which affect the user experience.
- Every transaction on Ethereum blockchain is required to pay transaction fee in ETH which most likely will be costly and creating an adoption barrier for the average user.

This hybrid approach will help avoid network fees in transactions between users, avoid stress on the public network due to the large volume of transactions and improve the user experience in terms of responsiveness and latency.

This design is applicable for every type of review and SARA Supply Chain where the auditor plays the same role as the reviewer in SARA review and mainly responsible for auditing, checking the compliance of the processes and standards of Supply Chain system.

SARA Platform overview



Paper Review Process

Before the paper become searchable and downloadable through the client, a community vote must be carried out for the paper to be accepted into SARA. This is to prevent things other than academic papers and related documents being uploaded to SARA. A vote for the document to be accepted into SARA is launched at the time of submission, a deliberately general question that is a binary yes/no vote: "Do the documents represent academic research or not?". All users can see pending votes on documents through their client, can download the documents associated with the vote to assess them and cast their own vote.

Threshholds:

Three thresholds are required to be met for a vote to be carried. First, a reputation threshold will be set and for a vote to pass a number of users that collectively have more than the required reputation must vote yes for the documents to be added to SARA. Second, a user number threshold will also need to be met, to ensure a minimum number of users vote on the inclusion of any documents. This ensures it is harder for users who have legitimately amassed reputation to dictate what does or does not come into SARA. The third and final threshold will be a time, votes will remain active for a set duration allowing enough time for either a yes or a no vote to accrue. If both yes and no votes reach their reputation and user number thresholds within the time frame, the vote defaults to no and fails. All three thresholds, reputation, number of users and time elapsed must be met for a vote to succeed.

Storage in SARA:

If a vote passes the respective documents are accepted into SARA and will be returned in the client through search results. If the vote fails the documents will be deleted and the user notified by way of their client. Acceptance into SARA means a hash of the document and the document metadata are added to Ethereum and the SARA client will seed the document for download by other users. The hash can be checked against seeds to prevent tampering with documents and help demonstrate version control for papers. Due to current limitations with the amount of information able to be stored on a blockchain, the article itself is not stored on Ethereum rather the hash value associated with the documents get stored on Blockchain and documents itself get stored into IPFS and metadata goes into database (say) MongoDB. With advances in blockchain and computing it is hoped

Paper Review Process

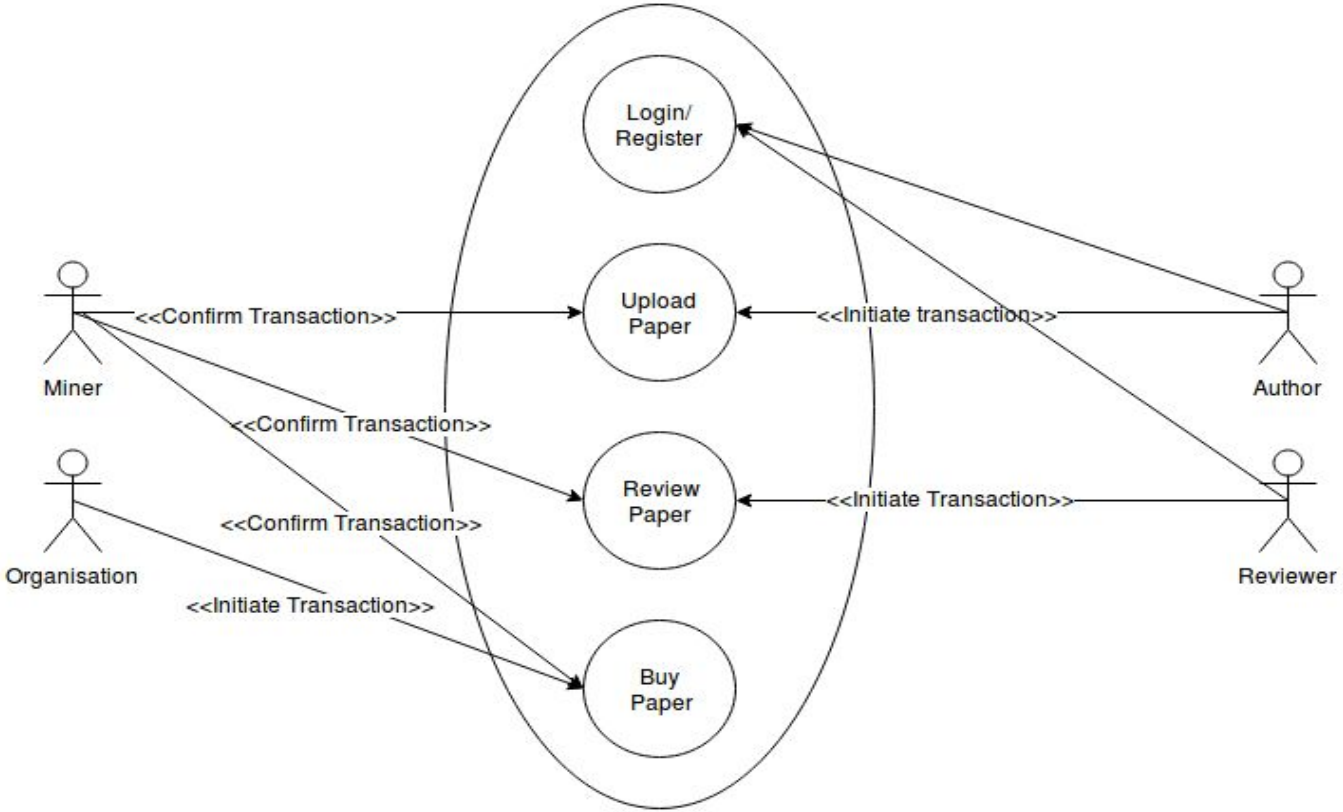
Voting reputation:

Voting will be incentivised through reputation and further reputation will be rewarded for voting with the majority, regardless of if the vote passed or failed or if the majority voted one way and the vote went the other way due to thresholds either being met or not. This means people will not be punished by the vote but they will be rewarded by following the crowd. The posed question, "Do the documents represent academic research or not?" is deliberately general and the crowd in most instances should pick the correct outcome, therefore deliberately trying to skew votes will be punished. To prevent a malicious actor downloading a number of clients and attempting to vote, only users with a set reputation level can vote. That reputation level can be attained by the solving of a one time captcha and/or any other method of proving a user is a human as the technology around bots continues to develop.

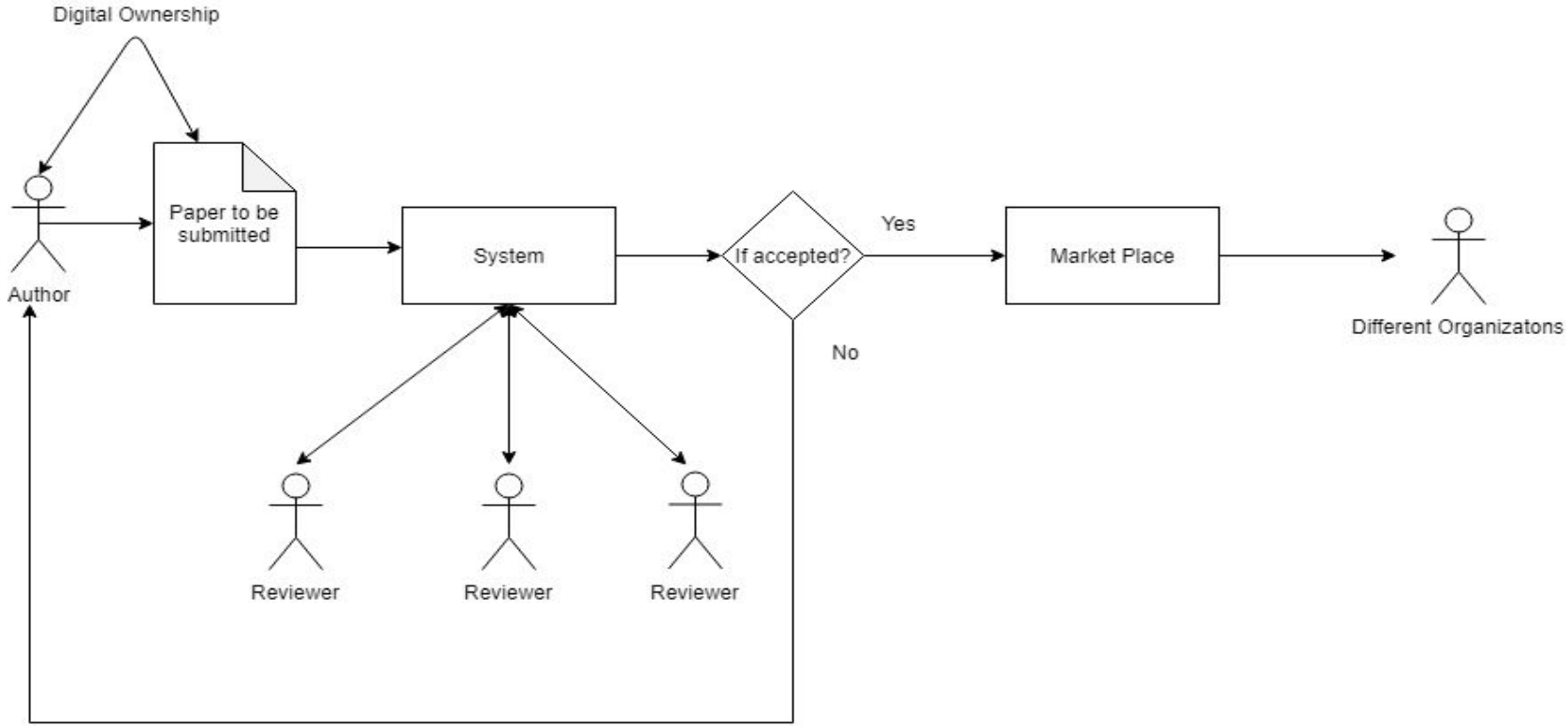
Cryptographic control:

One obvious vulnerability of SARA is that an attacker could upload a large number of articles in an attempt to overload the network. The mechanism that has been designed to mitigate such an attack is called a cryptographic control. A control is a cryptographic resource issued to users that allow them to upload documents to the network. It is abundant enough that a normal user would not notice the control, but places an upper limit on how many documents can be uploaded per user in a twenty four hour period. The control will be tied to interaction with the network, downloading articles, participating in votes, peer review etc., along with a natural amount that accrues over time that will decay if not used. The decay prevents an attacker registering 'sleeper' accounts to attack the network at some point in future. Currently, the SARA platform handles it by issuing a limited number of Ethereum tokens to each user internally to ensure that the users can enjoy the functionalities of the systems. However, this also acts as a barrier for a malicious user to carry out cyber attacks such as Denial of Service etc. on the platform.

Paper Review UML



Paper Review Design



Blockdoor Review Process

By being implemented on Blockchain technology, SARA Blockdoor review platform will address issues related to transparency and immutability of the score, as well as develop mechanisms to help users and experts earn profits based on system development and the quality of their reviews.

Below are preliminary procedures of the system:

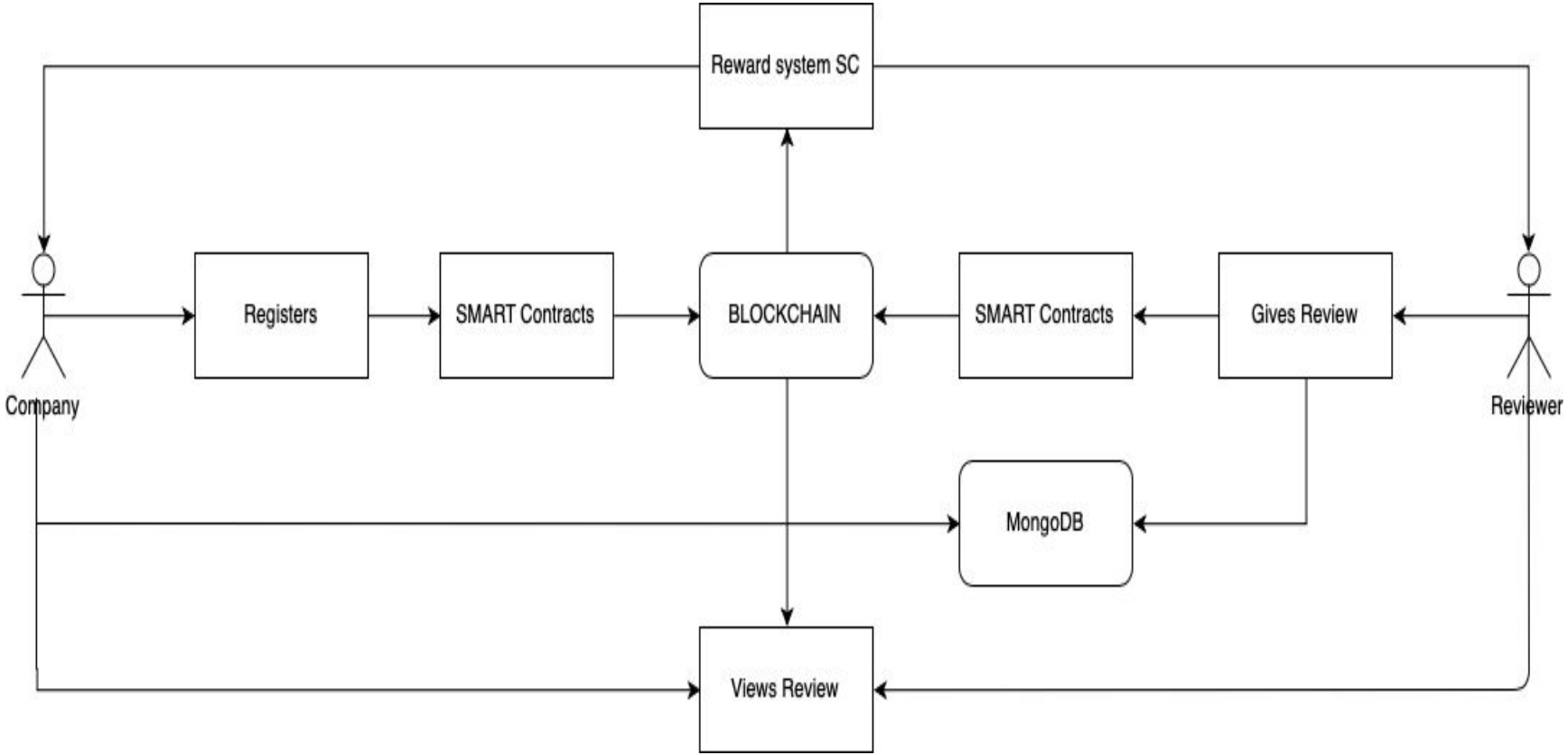
For Companies:
Companies willing to get reviewed get themselves registered. They may have to pay to get themselves registered to Blockdoor Review system.

For Merchants:
Merchants are the parties providing the products or services who wish them to be reviewed. To join the system, a merchant is required to register with Blockdoor review about their fields of business and may have to pay for registration or product listing fee(s).

For Experts:
Experts mean those users who are qualified and will review products and services on the system.

For Common Users:
Common users mean those who are using the Blockdoor review system to search the information on products and its reviews as well as write their own review and assessment. Common users can become Helpers through the above-mentioned options. Common users may be distributed system revenue for their review and assessment work, depending on configuration.

Blockdoor Review Design



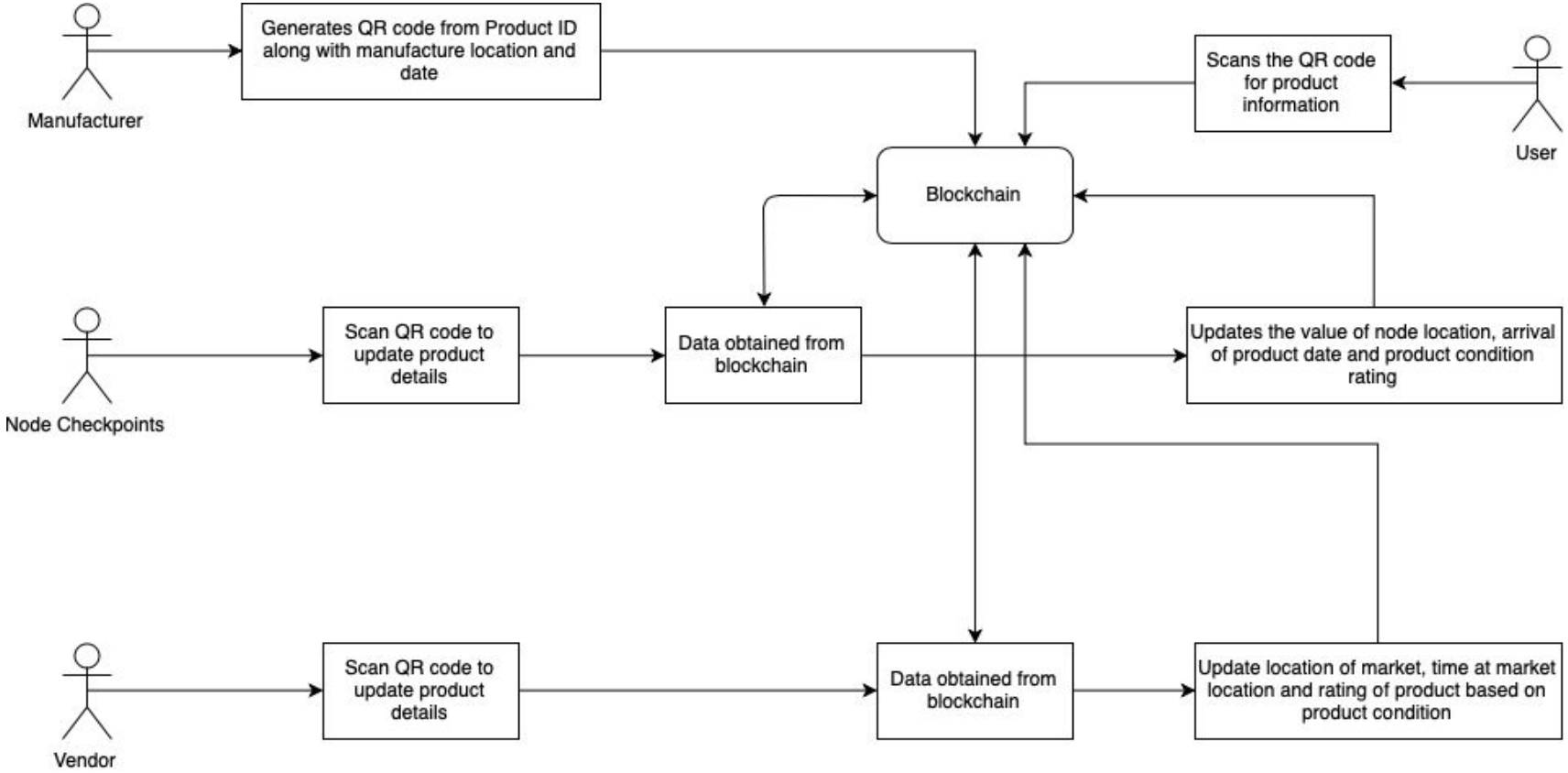
SCM Process

Blockchain has the frameworks to make up a robust system that takes into account all the building blocks and structure of the supply chain. The other advantage that comes with blockchain is the fact that it is capable of tracing all the history of specific entities that make up the rest of the food chain. Blockchain in solving traceability can bring together the four pillars; data to trace, traceability tools, product routing, and product identification, to make up a robust traceability system based on the blockchain.

Using Blockchain Technology, SARA Supply Chain can resolve the current problem with the increasing lack of trust from customer towards product quality:

- **Transparency:** Data stored on the Blockchain will be securely available to all parties in Supply Chain, like a customer, manufacturer, transporter, regulator. That means the information provided now can be trusted.
- **Reduce Fraud:** Because all data stored on the Blockchain is transparency to all nodes and in order to add new data to the blockchain, each node will require specific permission, as well as the consensus of the whole network, it's much harder (or impossible) for unauthorized party to manipulate the information on the Supply Chain..
- **Digitalize** paper-based process.
- **Reduce Human Error:** Based on Smart Contract, there are specific rules applied to each supply chain system, and those rules could run automatically without any human interference. So, the user cannot input "wrong" information on the Supply Chain. That will also maintain the integrity of data.
- **Review** provided at each node from the manufacturing to destination node is stored on Blockchain which is again tamper-proof, hence customer builds a trust on the system as they get detailed legitimate information of the products.

SCM Design


Manufacturer
Generates QR code from Product ID along with manufacture location and date
Scans the QR code for product information
User
Blockchain
Node Checkpoints
Scan QR code to update product details
Data obtained from blockchain
Updates the value of node location, arrival of product date and product condition rating
Vendor
Scan QR code to update product details
Data obtained from blockchain
Update location of market, time at market location and rating of product based on product condition

Possible Issues

User may over populate the system by uploading many documents:

The SARA platform denies such behaviour as user can be allowed to upload only limited papers in fixed time frame.

Digital Ownership of documents:

Authors can digitally sign the papers using the private key.

Too much information needs to be stored on Blockchain:

Complete information is not mandatory to be stored on Blockchain. Part of few data could be stored in database and only critically important information to be stored on Blockchain.

Companies purchasing reviews:

This type of behaviour is not economically feasible in our system, since the review fee will grow continuously.

Companies rejecting all critical reviews:

Since all reviews and the company's rejection or approval pf them are visible to everyone on the system, companies with high rejection rates will raise suspicion.

Unauthentic user may try to change the information related to product:

Our system doesn't allow such behaviour to as for each change in the information on Blockchain, it get's the change verified from every other nodes on the SARA network (Consensus Mechanism)

Crowdfunding

The SARA crowdfunding and the corresponding process of token generation are handled by smart contracts which are based on Ethereum. After we initiate ICO conducting activities, our platform SARA would welcome users who wish to support project's development by sending cryptocurrencies or tokens to SARA wallets.

The goal of crowdfunding is securing funds for R&D and covering platform-related costs. We can further plan to divide funds among R&D, marketing, Team, infrastructure and other indirect.

SAT tokens are based on Ethereum platform and standard token interface ERC20. The source code of our token can be found in Appendix B.

Despite the fact that we suggest using the token with our software, ERC20 lets any investor work with the token using any Ethereum client (for example – Mist wallet or myetherwallet). Because ERC20 tokens are fungible, our investors can minimise the risk of losing funds in the situation that our software malfunctions.

Future Work

SARA platform is still in development phase. Poof of Concepts(POC) for the same is ready and the platform can be deployed and tested on local environment. We welcome Companies and Investors to fund the project so that we can deploy and test the application in real time.

The main mechanics for interaction between users and the platform are already in place as well as the mechanism for platform-Blockchain interaction on the level of SARA tokens. We are planning to undergo alpha testing and soon be announcing public alpha once our project gets proper funding and resources.

In addition to the main user interfaces for SARA paper review system, Blockdoor review system and an android application for SCM process, we plan to develop widgets which can be directly installed on companies' websites.

Later, after we have made sure the platform is operating smoothly, we plan to expand to markets with the stated use cases and other use cases under the flag of SARA.

Conclusion

This paper describes the realization of peer review system, Blockdoor review platform and SARA supply chain management. We believe that the SARA platform meets basic needs of clients and customers at great extent.

Thanks to it's module architecture and versatility, the system can be rebuilt and customised in order to accommodate every other type of review based platforms. In light of all of the above, we believe that SARA has the potential to become a groundbreaking project on the review market.

Appendix(A)

```
pragma solidity >=0.4.25 <0.6.0;

contract Migrations {
  address public owner;
  uint public last_completed_migration;

  modifier restricted() {
    if (msg.sender == owner) _;
  }

  constructor() public {
    owner = msg.sender;
  }

  function setCompleted(uint completed) public restricted {
    last_completed_migration = completed;
  }

  function upgrade(address new_address) public restricted {
    Migrations upgraded = Migrations(new_address);
    upgraded.setCompleted(last_completed_migration);
  }
}
```

Appendix(B)

```
pragma solidity ^0.5.0;

import "openzeppelin-solidity/contracts/token/ERC20/ERC20.sol";

contract SARAToken is ERC20 {

    constructor() public {
        _mint(msg.sender, INITIAL_SUPPLY);
    }

    string public name = "SARAToken";
    string public symbol = "SAT";
    uint8 public decimals = 2;
    uint public INITIAL_SUPPLY = 51000000;
}
```

Appendix(C1)

```
pragma solidity ^0.5.0;

contract MainContract {
        struct Submission {
    address index;
    uint status;
    address[] reviewers;
    uint[] marks;
    uint rating;
    uint cost;
  }
  mapping(string => Submission) private submissions;
  string[] public hashes;

  function newSubmission(address userAddress, string memory docHash) public {
    hashes.push(docHash);
    submissions[docHash].index = userAddress;
    submissions[docHash].status = 1;
  }

  function isOwner(address userAddress, string memory h) public view returns(bool) {
    return submissions[h].index == userAddress;
  }

function displayHash(uint num) public view returns(string memory) {
    return hashes[num];
  }

  function setStatus(string memory docHash, uint s) public {
    submissions[docHash].status = s;
  }

  function displayDocCount() public view returns(uint) {

  function Review(string memory docHash, address reviewer) public {
    require(!isReviewed(docHash), "Reviewing have been closed");
    require(!hasReviewed(docHash, reviewer), "You have already reviewed this submission");
    submissions[docHash].reviewers.push(reviewer);
  }

  function giveMarks(string memory docHash, uint m) public {
    submissions[docHash].marks.push(m);
  }

  function hasReviewed(string memory docHash, address addr) public view returns(bool) {
    for(uint i = 0;i < submissions[docHash].reviewers.length;i++)
      if(submissions[docHash].reviewers[i] == addr)
        return true;
    return false;
  }

  function isReviewed(string memory docHash) public view returns (bool) {
    return (submissions[docHash].reviewers.length >= 5);
  }

  function setRating(uint r, string memory docHash) public {
    require(r > 0 && r <= 10, "Invalid Rating");
    submissions[docHash].rating = r;
  }

  function getRating(string memory docHash) public view returns(uint) {
    require(isPresent(docHash), "Document hash is invalid");
    return submissions[docHash].rating;
  }
}
```

Appendix(C2)

```
pragma solidity ^0.5.0;

contract MainContract {

        // Review storage struct
        struct Review {
    address reviewer;
    address company;
    string review;
    uint rating;
  }

  // Review structs are mapped to hash values
  mapping(string => Review) private reviews;
  string[] public hashes;

  // Function to insert a review in mapping
  function newReview(string memory reviewHash, address userAddress, address c, string
memory rev, uint rate) public
  {
    hashes.push(reviewHash);
    reviews[reviewHash].reviewer = userAddress;
    reviews[reviewHash].company = c;
    reviews[reviewHash].review = rev;
    reviews[reviewHash].rating = rate;

  }

  // Function to return count of reviews
  function getReviewCount() public view returns(uint)
  {
    return hashes.length;
  }

function getReviewHash(uint num) public view returns(string memory)
  {
    return hashes[num];
  }

  // Function to get reviewer rating
  function getReviewRating(string memory reviewHash) public view returns(uint)
  {
    return reviews[reviewHash].rating;
  }

  // Function to get review creator
  function getReviewCreator(string memory reviewHash) public view returns(address)
  {
    return reviews[reviewHash].reviewer;
  }

  // Function to get review body
  function getReviewBody(string memory reviewHash) public view returns(string memory)
  {
    return reviews[reviewHash].review;
  }

  // Function get company hash of review
  function getReviewCompany(string memory reviewHash) public view returns(address)
  {
    return reviews[reviewHash].company;
  }

}
```

Appendix(C3)

```
pragma solidity >=0.4.25 <0.6.0;

contract product {
        struct Product {
                // address genesis;
                string id;
                uint time;
                uint time1;
                uint timeF;
                address[] checkpoints;
                string name;
                string comname;
                string location1;
                string locationF;
                uint review1;
                uint reviewF;
                uint price;
        }

        mapping(string => Product) private products;
        string[] public ids;

  function newProduct(string memory hashid, string memory pname, uint pprice, uint
timestamp) public returns (string memory){
    ids.push(hashid);
    products[hashid].name = pname;
    products[hashid].price = pprice;
    products[hashid].time = timestamp;
    return "success";
  }

 function updateProduct(string memory hashid, string memory comname, string memory location1, uint
review1, uint time1) public returns (string memory){
    products[hashid].comname = comname;
    products[hashid].location1 = location1;
    products[hashid].review1 = review1;
    products[hashid].time1 = time1;
    return "success";
  }

  function finalProduct(string memory hashid,  string memory locationF, uint reviewF, uint timeF) public
returns (string memory){
    products[hashid].locationF = locationF;
    products[hashid].reviewF = reviewF;
    products[hashid].timeF = timeF;
    return "success";
  }

        function getAll(string memory hashid) public view returns (string memory, uint, uint, string
memory, string memory, uint,
                                                                                                uint,
string memory, uint, uint) {
                Product memory prd = products[hashid];
                return (
                        prd.name, prd.price, prd.time, prd.comname, prd.location1,
prd.review1, prd.time1,
                        prd.locationF, prd.reviewF, prd.timeF
                );
        }
}
```