Designing A Blockchain Based Smart Contract Model: A Case of Real Estate Industry.

Alex Kibet
alexriongosha@gmail.com /akibet@laikipia.ac.ke
Laikipia University, P.O Box 1100-20300
Nyahuru, Kenya

***Abstract*— Real estate is fundamental to everything we do; we need real estate property to live in, work in, and for leisure. It is not only an important commercial and industrial resource but is also home to many national and international investment funds. Recent research proves that, today, more people live in cities than in rural areas, and urbanization continues to grow around the world. This increases the magnitude of financial transactions and transparency challenges. Therefore, a technology to manage such transaction records is needed. This paper explored blockchain technology to aid in reducing such challenges. Blockchain is a new and emerging technology with the potential for implementation in various industries. Previous research in blockchain technology has concentrated on its potential application in digital currency. In this paper, the researcher endeavored to design a blockchain based smart contract model for management of real estate property that would address the weaknesses of the existing management models and potentially reduce the housing cost by elimination middlemen in the management process.**

***Index Terms* — Blockchain, Smart Contract, Real Estate**

I. INTRODUCTION

Real estate management involves putting in place procedures to ensure the smooth running of a real estate property (Cooke *et al*., 2018). In tenant-focused property management, firms are committed to creating a safe and enjoyable experience for those in their communities. They do this by offering the highest quality properties at competitive pricing and by forging sincere relationships with their tenants. Precisely, their goal is to create positive experiences for tenants. The major areas of responsibility include; Administration and Risk Management, Tenant and Occupancy. However, in the digital world, "Real estate management" involves the use of technology to manage real estate property. The real estate transactions records are saved and managed in a standard format by assurances needed to complete transactions (Ullah *et al.,* 2018).

Most governments raise income through tax to enable them to finance development projects that are meant to improve the economy of a given country or the region. In most third world countries house rental taxation has been classified as economic. In developing nations, statistics indicate that less than half of property owners and developers comply with rental income tax requirements (Sirr *et al*, 2017). Regardless of sanctions like penalties, armed monitoring, routine audits and fines, lack of documentation to proof exact amount of rental income to base on when computing payable tax is a limitation to revenue collections (Karanja, 2015). According to Mwangi (2014), one of the main tax evasion reasons in personal income generating is lack of correct reference information such as rent income details, which tend to lead taxpayers to evade tax. It's therefore important to put in place a mechanism that will help in ensuring that all landlords provide correct information to compute taxes (Ayuba, Saad &Ariffin, 2016).

Blockchain is a new and emerging technology on the market. It is an information technology with multiple classes of application i.e. any form of asset registry, exchange or inventory (Swan, 2015). Blockchain is a decentralized transaction technology first developed for Bitcoin cryptocurrency (Yli-Huumo *et al*, 2016), but every area of economics, finance, and money could be included in the technology. The technology uses ledgers (a distributed database architecture) to register any kind of information (Pinna & Ruttenberg, 2016) and has the potential to shorten the transaction time in any kind of transaction and make systems more transparent and reliable.

A. Statement of the Research Problem

The current management strategy employed by real estate owners depends on a number of intermediaries, including brokers, agents and banking service providers. The process of renting a house involves a great deal of bureaucracy and estate intermediary fees. When a property is let, the agreed terms such as rental duration and monthly payments due are laid out in tenancy agreements; a predominantly paper-based and hand-signed form of contract. This has led to slow, expensive, and brittle transactions with transparency and integrity issues. Problems with the current system involve the excess of paper documents required, lengthy bank transfer times, and high estate agent fees. Above all, the current management strategy has a potential of transaction records manipulation such that the government tax authorities might not have clear or true transaction records for auditing and filing taxes. This warrants or permits tax evasion or under filling of tax as well as having false information for government tax administrations, and tax advisory organizations for making tax-related decisions.

B. General Objective

The primary concern of the study is to come up with a model design that would reduce operational costs while managing a rental house and consequently lowering the rental fee. This would be realized by reducing the current great deal of bureaucracy around real estate property management and to allow Immutable and Autonomous payment of a rental fee in

a transparent, conflict-free way while avoiding the services of middlemen. The design of the model would be guided by the following objectives:-

i) To examine Existing Models Used In Real Estate Property Management
ii) To design a blockchain based smart contract model for real estate

II. SURVEY OF LITERATURE

A. Existing Models Used In Real Estate Property Management

This section presents results of objective one of this study, which involved exploration of the weakness of existing models used in real estate property management. The Existing real estate management models reviewed in this study included E-Resident, SPENN, Deed-coin and The Bit-property. E-resident Platform was designed by KCB Bank Kenya Limited in partnership with E-Resident Limited and has been publicized to offer a host of benefits such as security, convenience and cut collection cost for landlords and agents. The platform also makes tracking of payments easier since the money is sent from any of the money transfer platforms straight to the bank. This model also designed to ensure customer's right to privacy and right to access information with lawful restrictions therein as provided for in the Constitution of Kenya. SPENN is a mobile-based platform for banking application that was designed to enable Rwandans to make a range of financial transactions. The application allows smartphone users to transact and pay for goods and services. Deed-coin platform allows searching for a real estate agent. Bit-property is a decentralized real estate platform powered by Ethereum smart contracts that aim at giving users a seamless, quick, and low-cost way to own and trade real estate revenue streams.

B. Weaknesses of the Existing Models

To appreciate the importance of technology product, there needs to be a reason for the technology to exist. Therefore, to justify the need for a new blockchain smart contract for real estate, the existing real-world alternative needs to exhibit inefficiencies in their use cases. The set criteria for the existing models evaluation include; transaction transparency and immutability, decentralization (distributed ledger), presence intermediaries, implementation of rights and obligations, potential to reduce overall rental fee, fractions in claiming deposits conflicts over late payment, possibility of unexplained rent increments and possibility to enforce lease agreement. The table below presents the key of assessment criteria:

Table 4 Key of Assessment Criteria

S/N	Assessment Criteria	Traceability
i)	Transparency	A.
ii)	Immutability	B.
iii)	Decentralization	C.
iv)	Presence Intermediaries	D.
v)	Implementation of Rights and Obligations	E.
vi)	Potential To Reduce Overall Rental Fee	F.
vii)	Fractions in Claiming Deposits	G.
viii)	Conflicts Over Late Payment	H.
ix)	Possibility of Unexplained Rent Increments	I.
x)	Possibility To Enforce Lease Agreement	J.

Table 5 Existing Models Assessment

Model	A	B	C	D	E	F	G	H	I	J
E-Resident	✗	✗	✗	✓	✓	✗	✓	✓	✓	✓
Bit-property	✓	✓	✓	✗	✓	✗	✗	✗	✗	✓
SPENN	✓	✓	✓	✓	✗	✗	✓	✓	✓	✗
DEED'S	✓	✓	✓	✗	✓	✓	✗	✓	✓	✗

The E-Resident platform in Kenya is a web-based platform that focuses only on creating convenience to tenants who will not have to deal with agents or queue at the bank to make rental payments. It does not allow transaction transparency, immutability or decentralization (distributed ledger). Its operation also encourages the need for trusted intermediaries such as the banks and agents thus increasing the rental fee. Furthermore, the model does not prevent the cases of unexplained rent increment or allow enforcement of lease agreement, rights and obligations. The Bit-property smart contract platform aside from being decentralized smart contract platform that aims at giving users a seamless, quick, and low-cost way to own and trade real estate revenue streams, it does not address issues surrounding renting a property.

SPENN is a blockchain mobile-based platform for banking application that enables users to make a range of financial transactions. This model does not address property management requirements such agreements**,** to authorize occupants' rights or obligations. DEED'S platform focuses on reducing challenges in finding real estate agent and the decision on the best commission. It does not deal with real estate property management.

III. DESIGNING A BLOCKCHAIN BASED SMART CONTRACT MODELS

In this section, the design of the b**l**ockchain based smart contract model for real estate is presented. This addresses the attainment of research objective two of the study. First, blockchain overview and smart contracts are described

followed by the three different user archetypes are described along with user stories in order to provide functional specifications for the model and the model objectives. Further descriptions of the model such as quality attributes and how to set the system of smart contracts up with a blockchain are also given. Thereafter a schematic of the prototypical interactions on the blockchain is shown along with the interactions between the smart contracts.

A. Blockchain Overview

Blockchain is technology originally developed as the accounting method for the virtual currency Bitcoin (Eskandari *et al.*, 2018). Blockchain uses distributed ledger technology (DLT) and is appearing in a variety of commercial applications today (Lipton, 2018). Currently, the technology is primarily used to verify transactions, within digital currencies though it is possible to digitize code and insert practically any document into the blockchain. Doing so creates an indelible record that cannot be changed. Furthermore, the record's authenticity can be verified by the entire community using the blockchain instead of a single centralized authority. The sustained high level of robust security demonstrated by public crypto-currencies has shown to the world that this new wave of blockchain technologies can provide efficiencies and intangible technological benefits very similar to what the internet has done (Freund & Stanko, 2018).

Within the blockchain is a block that is the 'current' part of a blockchain, which records some or all of the recent transactions (Ozercan *et al.*, 2018). Once completed, a block goes into the blockchain as a permanent database. Each time a block gets completed, a new one is generated. There is a countless number of such blocks in the blockchain, connected to each other (like links in a chain) in proper linear, chronological order. Every block contains a hash of the previous block. The blockchain has complete information about different user addresses and their balances right from the genesis block to the most recently completed block. Blockchain transaction works broadly as follows:

i) **Transaction Initiation**: One party (the sender) creates a transaction and transmits it to the network.
ii) **Transaction Authentication:** The nodes (computers and users) of the peer network receive the message and authenticate its validity by decrypting the digital signature.
iii) **Block Creation:** Pending transactions are put together in an updated version of the ledger, called a block, by one of the nodes in the network.
iv) **Block Validation:** The validator nodes of the network receive the proposed block and work to validate it through an iterative process that requires consensus from the majority of the network.
v) **Block Chaining:** If all transactions are validated, the new block is "chained" into the blockchain, and the new current state of the ledger is broadcast to the network.

B. Smart Contracts

Smart contracts are self-executing programs with terms of the agreement between two or more parties being directly written into lines of code that runs on the blockchain network (Atzei, 2017). The code and the agreements contained therein exist across a distributed, decentralized blockchain network (Luu et al, 2016). Kosba and others (2016), argues that contracts permit trusted transactions and agreements to be carried out among different, anonymous parties without the need for a central trusted party, legal system, or external enforcement mechanism. They render transactions traceable, transparent, and irreversible. Smart contracts help to exchange currency, possessions, stocks, lease of assets or whatever of value in a clear, conflict-free manner while avoiding the services of a middleman. As shown in the following figures, the difference between traditional contracts and the present smart contract can be seen. The traditional middlemen is replaced by auto-executable contract code.

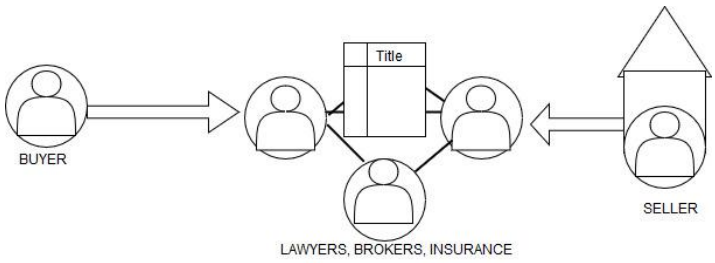

Figure 1 a Simple visualization of traditional contract having intermediaries (CB Insights, 2017)

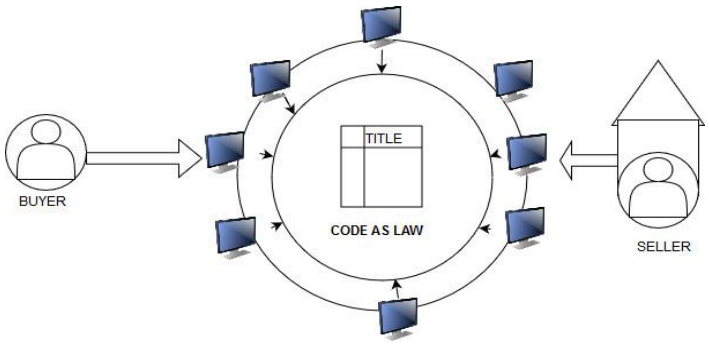

Figure 2 an example of working smart contract (CB Insights, 2017)

C. Defining Model Functional Requirements

The design outlined in this study is limited to three archetypical users: Admin, Tenant, and Landlord. In order to describe the various functional requirements that users have on the application, user stories were written and are shown in the Table below. The different user types are thereafter defined more in detail. An effort was made to simplify the

user stories and requirements to the bare minimum, while still keeping the PoC at a viable level of usability and security.

Table 1 User stories defining functional requirements and guiding the development of the model PoC Source: (Researcher, 2019)

As a . .	I want/need to . . .	Traceability
admin	Identify myself in a cryptographically secure manner upon accessing Tenant information on the blockchain, so that no unauthorized entities can access it.	1.1
	Administrate (setup and maintaining accounts) the system.	1.2
Landlord	Be able to see records of my tenants so that I can know the number and compute expected pay and when to expect payment.	2.1
	Verify the tenant-account identity to know who has paid and who has not.	2.2
	Be able to see how much rent has been collected, so that I can plan for service, bills, and further investment.	2.3
	Set and Adjust the rent level to attract tenants to the property.	2.4
	Be able to Collect Rent from smart contract account each month and strictly enforcing late fees.	2.5
	Identify myself in a cryptographically secure manner upon accessing landlord information on the blockchain, so that no unauthorized entities can access it.	2.6
Tenant	Be able to create an account through registration.	3.1
	Identify myself in a cryptographically secure manner upon accessing my personal information on the blockchain.	3.2
	Be able to see and update my personal information upon registration.	3.3
	Be able to share information on my account with Landlord and Admin on the blockchain.	3.4
	Be able to pay my rental fee using any form of cryptocurrency. bills for gas, electricity, and telephone if this was agreed with the landlord	3.5
	Be able to check my account balance (ether wallet or token wallet).	3.6
	Be able to see my current payment status and expected date of the next payment.	3.7

The above user stories can be summarized in the table below to give the model functional overview.

Table 2 Model Functional Overview Source: (Researcher, 2019)

The Model Functionality	How to realize
1. User registration	All users are required to register with the model prior to accessing any of the model functionality. This is to capture the user personal information and authentication details.
2. User authentication and security	The proposed model ensured that all users are registered before allowing them to access any of the system functions. The registered persons are required to log in using their account addresses and passwords.
3. User de-registration	Anytime a tenancy contract is terminated, the tenant details are removed from an active account to a repository.
4. Rental fee payment	The model allows rental payment by use of Ethereum. This PoC uses a created crypto tokens to allow unlimited testing.
5.Transaction records management	The model allows recording of the transaction on the blockchain
6. Report generation	The model enables users to generate reports regarding transactions.

1) Users Overview and Interactions with Blockchain

The participating individuals in the blockchain based model for real estate management model are composed of; Tenants, admin, Landlord, and hosting service providers (EVM, Github, and Metamask).

1) **Tenants** are assumed to be private persons or group that rents and occupies a house, an office, or the like, from another (landlord) for a period of time; lessee.

2) **The landlord** is assumed to be a person or group who owns a house, apartment, condominium or real estate which is rented or leased to an individual or business, who is called a tenant (also a lessee or renter).

3) **System administrator, or sysadmin,** is a person who is responsible for the upkeep, configuration, and reliable operation of computer systems; especially multi-user computers, such as servers. The system administrator seeks to ensure that the uptime, performance, resources, and security of the computers they manage meet the needs of the users, without exceeding a set budget when doing so.

4)

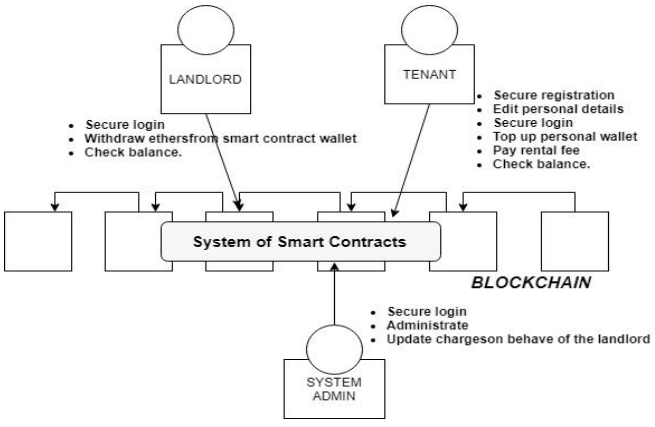

Figure 3 Overview of different users and their interactions with the blockchain and the model as a system of smart contracts
Source: (Researcher, 2019)

The requirements are described in a less formal way in Figure 1, where the different users are shown interacting with the blockchain, on which the smart contracts reside. In bullets next to them are the actions they need to be able to perform. There are some requirements that apply to the general system and not just to one user specifically. Some of them are described in part by the user stories, but for the sake of exhaustiveness and application to users not in the system, they are explicitly written below.

R1. It must be impossible for a non-registered person to join the model network.

R2. Only those permitted to should be allowed to connect to the network.

R3. There must be immutable traceability built into the model, where it is possible to see:

i) All rental fee transactions captured in the blockchain (in form of transaction Blocks).
ii) Account balance before and after the transaction.
iii) Old transactions.

Immutable traceability means that there must be a history of changes made to the rental fee, payment and that it must be made very difficult, if not impossible, to alter it post ex.

R4. Smart contracts must be exchangeable without the need to remove the entire model or change addresses to contracts with which humans interact.

The design of the final PoC was based on the requirements and user stories mentioned above. It would also be guided by the core objective of this research.

D. Model Design Constructs

The real estate model can be subdivided to modules. The sections are organized according to functions to be achieved. The model functional sections are as listed below;

i) **The smart Contract -** To write an Ethereum smart contract Solidity programming language was used. It's a general-purpose programming language that uses a class (*contract*) and methods that define it. Solidity main purpose is to send and receive digital tokens as well as storing states.

ii) **Ethereum wallets-** To fully demonstrate the research objectives, three types of wallets on Ethereum were created: for the tenant, for the landlord, and for the contract respectively.

iii) **Dapp** - The front end users are using Dapp to connect with the blockchain via the Smart Contract (Front End → Smart Contract → Blockchain). Truffle development framework for Ethereum was used.

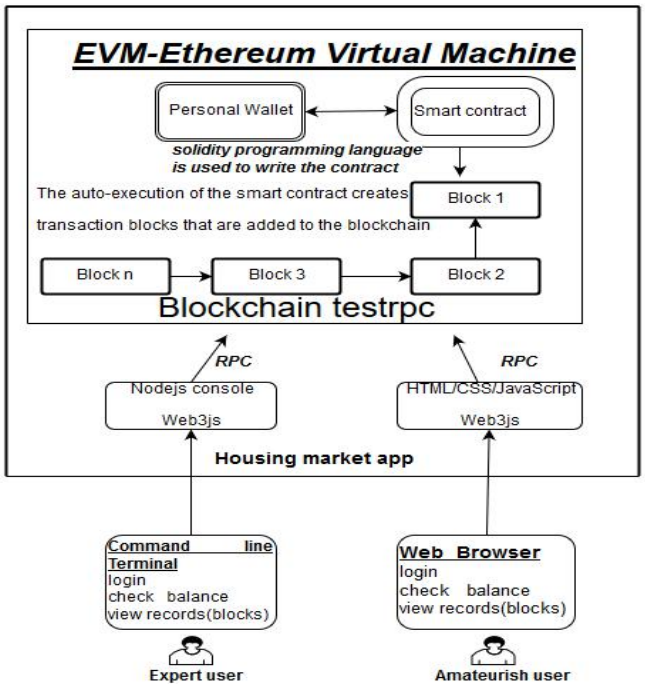

Figure 4 Model Development Source: (Researcher, 2019)

E. Overview of Smart Contracts Model Design

Every non-trivial Dapp will require more than one contract to work well. According to beck and others, there is no way to write a secure and scalable smart contract back-end without distributing the data and logic over multiple contracts (Beck et al, 2016). Therefore, the proposed architecture of the system of smart contracts is based on the design principle of having different types of contracts to perform different classes of tasks. To classify the contracts, a model called "The Five Types Model" is used (Monax, 2017), although not all of the five models are actually used in this proposed model. This model divides contracts into: Database contracts, Controller contracts, Contract managing contracts, Application logic contracts and Utility contracts (monax, 2017) . To manage the model through its entire life cycle, from setting up a structure to drawing up detailed plans, to executing and completing the project the following overview of the system of smart contracts for the Real Estate Management PoC figure was of help. It describes what the system is and what it does. Although, based on visualization technique by (Brown, 2016) not all smart contracts are included in the PoC for sake of clarity and relevance.

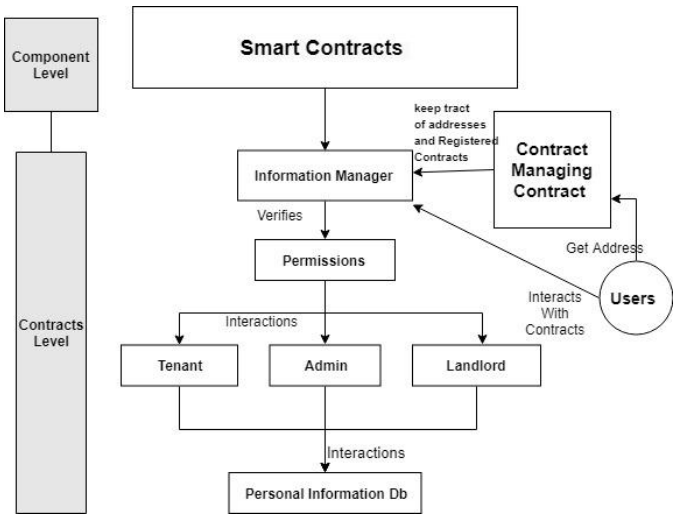

Figure 5 Overview of smart contract design Source: (Researcher, 2019)

F. The Model Starting-Up Activities

To start bare truffle project the "truffle init" is ran from the command line in the project directory. Once this operation is completed, the project structure is created with the following items: Contracts Directory for Solidity contracts, migrations Directory for scriptable deployment files, Test Directory for test files for testing your application and contracts, and truffle.config.js a Truffle configuration file. The sequence of activities is as shown in the figure below where the validators are configured, the blockchain is started and transactions can begin.

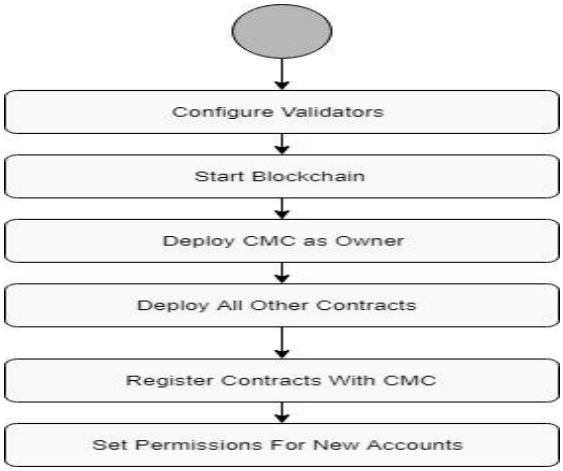

Figure 6 Diagram of how the starting-up activities for the model work Source: (Researcher, 2019)

G. Truffle framework installation

The PoC model isn't complete with just the smart contracts. There is also a need for blockchain, as well as a structure for handling keys. Additionally, there needs to be a type of either distributed or centralized consensus established, on who is allowed to join the network as a Tenant or Landlord.Truffle platform was used due to its efficient consensus mechanism compared to the available alternative (parity). Thus issues around appropriate consensus mechanism are solved. Truffle is a development environment, testing framework and asset pipeline for blockchains using the Ethereum Virtual Machine (EVM), aiming to make life as a developer easier (Ahmad, 2017). It has become one of the most widely used IDEs and framework in the Ethereum community. Developers use it to build and deploy Dapp for testing purposes with many features that make it more attractive to users with a Web 3.0 development background. It has Built-in smart contract compilation, linking, deployment and binary management with Mocha and Chai Automated contract testing. It is also known to be Configurable build pipeline with support for custom build processes and Scriptable deployment & migrations framework. The following steps describe how to install and run truffle.

Truffle framework environment setup

To use truffle framework, the following discussed dependencies and software must be configured. These include NodeJS, Web3, Node packet manager NPM, and lite server.

1) **NodeJS v8.9.4 or later**

Node.js is an open-source, cross-platform JavaScript run-time environment that executes JavaScript code outside of a browser. Both the JavaScript and Node.js run on the V8 JavaScript runtime engine. This engine takes the JavaScript code and converts it into faster machine code. Node.js uses an event-driven, non-blocking I/O model that makes it lightweight and efficient.

2) **The node package manager (npm)**

npm is a package manager for the JavaScript programming language written entirely in JavaScript and was developed by Isaac Z. It is the default package manager for the JavaScript runtime environment Node.js. It consists of a command line client, also called npm, and an online database of public and paid-for private packages called the npm registry. The registry is accessed via the client, and the available packages can be browsed and searched via the npm website. The package manager and the registry are managed by npm.

3) **Lite server**

Lite-server is a lightweight development web server with support for Single Page Apps (SPAs)**.** Lite-server is a simple customized wrapper around Browser Sync to make it easy to serve Single Page Applications (SPAs). It is Lightweight development only node server that serves a web app, opens it in the browser, refreshes when html or JavaScript change, injects CSS changes using sockets and has a fallback page when a route is not found. Browser Sync is

super-fast lightweight development server. It serves the static content, detects changes, refreshes the browser, and offers many customizations. Lite-server uses Browser Sync and allows for configuration overrides via a local bs-config.json or bs-config.js file in a project.

4) **Metamask**

Metamask is a browser plugin that allows users to make Ethereum transactions through regular websites. It is a bridge that allows you to visit the distributed web of tomorrow in your browser (MetaMask, 2018). It allows running of an Ethereum Dapps on browser without running a full Ethereum node. This facilitates the adoption of Ethereum because it bridges the gap between the user interfaces for Ethereum, for example, Mist browsers or Dapps and the regular web browser such as Chrome or Firefox. According to metamask (2018), MetaMask injects a JavaScript library called web3.js into the namespace of each page your browser loads. Furthermore, MetaMask allows users to specify which Ethereum node to send these requests to. The ability to send requests to nodes outside of the user's computers is important because it means that people can use Ethereum without having to download a node consisting of 10+GB blockchain on to their computers. The MetaMask add-on can be installed in Chrome, Firefox, Opera, and the new Brave browser (Coin, 2018). To use MetaMask, this research installs it on the Chrome browser.

5) **Web3**

Web3.js is a collection of libraries which allow interaction with a local or remote Ethereum node, using an HTTP or RPC connection. It is for the Ethereum blockchain and smart contracts. The whisper protocol to communicate p2p and broadcast, swarm protocol, the decentralized file storage and has web3-utils helper functions for Dapp development. Web3.js talks to The Ethereum Blockchain with JSON RPC (Remote Procedure Call) protocol. Web3.js allows requests to an individual Ethereum node with JSON RPC in order to read and write data to the network as shown below.

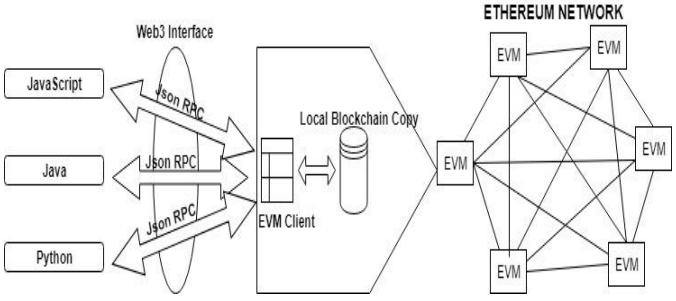

Figure 7 Interaction with a local or remote Ethereum node, using web3 interface *Source:* (Researcher, 2019)

IV. REFERENCES

Ahmad, A. (2017). *Integration of IoT devices via a blockchain-based decentralized application* (Master's thesis).

Atzei, N., Bartoletti, M., & Cimoli, T. (2017, April). A survey of attacks on ethereum smart contracts (sok). In *International Conference on Principles of Security and Trust* (pp. 164-186). Springer, Berlin, Heidelberg.

Beck, R., Czepluch, J. S., Lollike, N., & Malone, S. (2016, May).Blockchain-the Gateway to Trust-Free Cryptographic Transactions.In *ECIS* (p. ResearchPaper153).

Brown, S. (2016). Visualise, document and explore your software architecture – software architecture for developers. Leanpub. Retrieved from https ://leanpub .com/visualising-software-architecture

Cooke, H., Appel-Meulenbroek, R., & Arentze, T. (2019). Is the much discussed agility of corporate real estate visible in practice? An empirical study of the relationship between business metrics and surplus property. *International Journal of Strategic Property Management*, *23*(4), 227-243.

Eskandari, S., Clark, J., Barrera, D., & Stobert, E. (2018). A first look at the usability of bitcoin key management. *arXiv preprint arXiv:1802.04351*.

Freund, A., & Stanko, D. (2018). The wolf and the caribou: Coexistence of decentralized economies and competitive markets. *Journal of Risk and Financial Management*, *11*(2), 26.

Kosba, A., Miller, A., Shi, E., Wen, Z., & Papamanthou, C. (2016, May). Hawk: The blockchain model of cryptography and privacy-preserving smart contracts. In *2016 IEEE symposium on security and privacy (SP)* (pp. 839-858). IEEE.

Lipton, A. (2018). Blockchains and distributed ledgers in retrospective and perspective. *The Journal of Risk Finance*, *19*(1), 4-25.

Luu, L., Chu, D. H., Olickel, H., Saxena, P., & Hobor, A. (2016, October).*Making smart contracts smarter.In Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security* (pp. 254-269).ACM.

MetaMask. (2018, 7 6). MetaMask. Retrieved from MetaMask: https://metamask.io/

monax. (2017, 12 16). the five type model. Retrieved from solidity explainer: https://monax.io/learn/smart_contracts/.

Ozercan, H. I., Ileri, A. M., Ayday, E., & Alkan, C. (2018). Realizing the potential of blockchain technologies in genomics. *Genome research*, *28*(9), 1255-1263.

Pinna, A., & Ruttenberg, W. (2016). Distributed Ledger Technologies in Securities Post-Trading Revolution or Evolution?. *ECB Occasional Paper*, (172).

Ullah, F., Sepasgozar, S., & Wang, C. (2018). A Systematic Review of Smart Real Estate Technology: Drivers of, and Barriers to, the Use of Digital Disruptive Technologies and Online Platforms. *Sustainability*, 10(9), 3142.

Sirr, G., Garvey, J., & Gallagher, L. A. (2017). Bilateral investment treaties and foreign direct investment: Evidence of Asymmetric effects

on vertical and horizontal investments. *Development Policy Review*, *35*(1), 93-113.

Swan, M. (2015). *Blockchain: Blueprint for a new economy*. " O'Reilly Media, Inc.".

Yli-Huumo, J., Ko, D., Choi, S., Park, S., & Smolander, K. (2016). Where is current research on blockchain technology?—a systematic review. *PloS one*, *11*(10), e0163477.

Alex Kibet- is Computer Science lecturer with over four years' experience working in the ICT industry as a professional certification (HCIP, HCIA, CCNA and CCNP) instructor, University tutor, researcher and academy administrator. A researcher in information technology with extensive research experience in Blockchain for sustainability and a technology for Market disintermediation. Further research interests include: Cryptocurrency, smart contracts and currently working on Blockchain based Model for Real estate Markets.