electronics

MDPI

Article

Verification Plan Using Neural Algorithm Blockchain Smart Contract for Secure P2P Real Estate Transactions

Jun-Ho Huh [1] and Seong-Kyu Kim [2,*]

[1] Department of Data Informatics, Korea Maritime and Ocean University, Busan 49112, Korea; 72networks@kmou.ac.kr

[2] Department of Information Technology, Sungkyunkwan University, Seoul 03063, Korea

* Correspondence: guitara7@skku.edu

Received: 21 May 2020; Accepted: 23 June 2020; Published: 26 June 2020

Abstract: Blockchain and artificial intelligence are the most important keywords in the Fourth Industrial Revolution. This study sought to apply these core technologies to future validated algorithms that make real estate transactions secure to come up with an encryption algorithm. In addition, the real estate transaction is being paid a large fee by the middlemen, the real estate agent. Furthermore and recently, P2P (peer-to-peer) real estate exchange is used a lot. However, these P2P real estate exchanges also have problems that have not been identified by each other between landlords and tenants. In particular, a research model was established to compare and verify the PBFT (practical Byzantine fault tolerance) algorithm of Hyperledger through the blockchain agreement process. Subsequently, a process for verifying the real estate contract was established. Through VM (virtual machine) research methodology for the verification of blockchain real estate contracts, ElGamal communication was provided to prove quantum cryptography. We also automated lightweight encryption test verification tools and blockchain smart contract VM (virtual machine) models using artificial intelligence. Verification was performed through a reservation server and a monitoring server using a test verification tool for network-based lightweight security IoT (Internet of things) GW (gateway). It presents important ECP (elastic curve program) and elastic curve Qu-Vanstone (ECQV) models among the main functions of the blockchain smart contract, and it is equipped with quantum-based encryption algorithm. In addition, the necessary UML (unified modeling language) source code and performance data were calculated according to the actual experimental environment, and the average value for blockchain for administrative or government authorized assets—4000 TPS (transaction per second) were tested. In the future, we want to use this technology for real estate transactions.

Keywords: real estate; real estate blockchain; real estate transactions; blockchain; blockchain consensus algorithm; smart contract; ethereum

1. Introduction

Blockchain and artificial intelligence are the most suitable technologies for future real estate transactions. As such, this study sought to incorporate verification encryption algorithms to make real estate transactions necessary for future transactions secure. In this study, Blockchain was intended to be applied as a way of solving problems of real estate transaction services. In the case of previously applied Blockchain technology, it is characterized by its safety against disclosure, tampering, or false information, and it is appropriate to apply to an environment that deals with low-frequency data [1–3]. We also found common ground, i.e., the subject is easy to apply when it is able to define a relationship

between different positions. As such, the real estate transaction market is an environment where in sales information should be made public, there should be no false information, and infrequent data occur with an average of millions to tens of millions of occurrences per day globally. In addition, since sales transactions take place between tenants, renters, and real estate agents, each relationship can be defined, making it suitable for the application of Blockchain technology [4–7]. Accordingly, in this study, we proposed a plan to implement a service model that can share information on the sales of the real estate market with Blockchain and link transactions to form a mutual monitoring system between real estate properties through which a real estate agent's malicious registration of false sale is prevented. Moreover, by preventing the occurrence of false sale due to omission, the market's credibility is enhanced. The rest of this paper is organized as follows: Section 2 presents the technology related to this paper particularly Blockchain smart contract and artificial intelligence deep learning; Section 3 explains the research plan, presents a methodology for real estate transactions, and proposes the necessary skills such as credentials, key pairs for credentials, and verifier information; Section 4 describes the application and operating principles of the proposed method and the virtual machine (VM) certification method to measure the performance of the real estate contract in terms of how the technology is operated; Section 5 analyzes the characteristics of Blockchain and describes its strengths and weaknesses and possibilities; finally, Section 6 aims to present a Blockchain artificial intelligence-based technology model in conclusion so that real estate transactions can be made and discuss ways to revitalize the market including the possibility of future development of Blockchain technology.

2. Related Research

Related research has shown that the existing real estate market has been around for a long time all over the world, and trading is still active today. Nonetheless, all the real estate transactions so far have been done by middle arbitrators who collect a lot of fees when notarization is skipped. There are many websites that can do P2P (peer-to-peer) transactions between individuals, but the reality is that they are legally limited. This is because it would be difficult to conduct real estate transactions securely without a real estate agent. This study began with the idea that applying smart contracts to real estate transactions where these problems exist can solve these problems by using artificial intelligence Blockchain smart contracts in a non-face-to-face manner rather than directly between the parties to the contract. This study began a few years ago; in this section, we will look at the relevant research. First is smart contract technology. Smart contracts can completely break away from existing repeater-oriented transactions. Second is Blockchain agreement technology, which verifies contracts made in this manner. The similarity is measured using artificial intelligence algorithms prior to an agreement. We will also look at the future prospects of real estate [8–10].

2.1. Real Estate Market

The real estate market for smart contracts has already been discussed extensively. Nonetheless, it is true that specifics are lacking as to how to apply and how Blockchain technology is mounted. The existing system, which normally registers one property, displays all the physical and legal, especially physical, properties related to this property [11–14]. In smart assets, these attributes can still be represented on Blockchain tokens. For now, physical characteristics such as address, GPS (global positioning system) coordinates, land area, building floor area, building year, and legal characteristics of ownership and utility rights can be recorded on one or a common token. Likewise, whenever a legal act such as sale or collateral or mortgage occurs, such transactions are stored and tracked in real time in a block chain token. When an asset is first linked to a Blockchain token, the Blockchain records the owner of the asset as an "initial" owner. If this asset is sold, the original owner's transaction output reads "Consumption (sent)"; at the same time, the transaction output of the new owner is created. A third party wishing to determine who owns the asset simply needs to look at the transaction history of the asset. The transaction details go all the way from the original owner to the current owner whose transaction output has not yet been "consumed". The current owner's transaction output is "open",

meaning only the current owner is the party who can "consume" his/her ownership through the act of trading. The current owner is a Blockchain to prove his/her ownership.

The corresponding address of the award can be made public through a private key. If a smart asset is operated by a public Blockchain, the transaction details of the smart asset can be queried through the Internet via a program called "Block Explorer". Since Blockchain's smart asset management system does not specify an individual as a natural person but only as a private key, however, the aforementioned legal problems may arise if a private key is lost or hacked. Real estate as a smart asset consists of various official documents that must be issued by a registered office, a notary office, or a county clerk in the United States, or various kinds of documents that must be issued by a court registration office and a front-line administrative office in the case of Korea.

Official documents can be replaced with digital signatures. This is possible because Blockchain is a decentralized ledger system, and digital signatures can function as notary documents of real estate anywhere in the world beyond borders. Therefore, real estate as a smart asset can be traded to anyone around the world through the certification of networks around the world without the verification of the authority of governments and courts and registration of new authority. As stated in the tactic, the global expansion of these trading areas can dramatically overcome the locality that could not be avoided by the existing real estate market through linkage with the global financial system. Few countries like Korea have achieved the computer-aided registration of national real estate registration; even the US and developed countries do not have a nationwide real estate registration network, which requires a lot of information costs for real estate transactions across the county border. In particular, a real estate agent as a smart self-employed can have more significance for developing countries where the gap between the registration system and actual ownership is not resolved in a chronic manner compared to developed countries, where legal protection for real estate registration is comprehensive (see the global real estate outlook in Figure 1) [15].

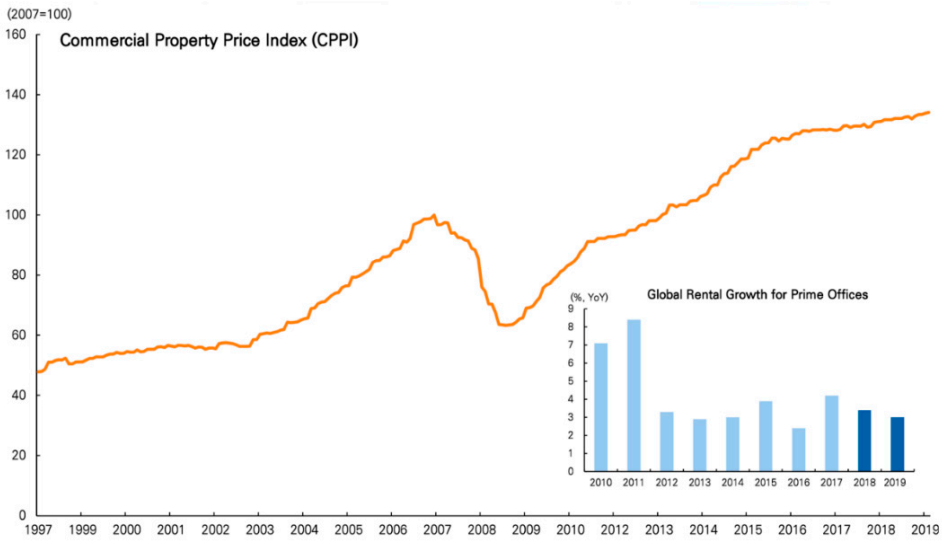

Figure 1. Global real estate index.

Real estate is traded worldwide through the new real estate investment trusts (REIT). REIT refers to a real estate investment company in the form of a corporation that collects funds from a large number of investors and allocates to investors the profits generated by investing in real estate, development projects, and securities [16,17]. These real estate transactions are a very necessary part of Blockchain smart contracts because they are funded by unspecified individuals.

In addition, REITs' profits worldwide are reaching record highs in 2019. The US's Nareit FFO (funds from operations) recorded $16.5 billion, up 13 percent compared to 3 years ago. Various properties are also flowing into REITs, which are showing stable transportation profit (see Figure 2).

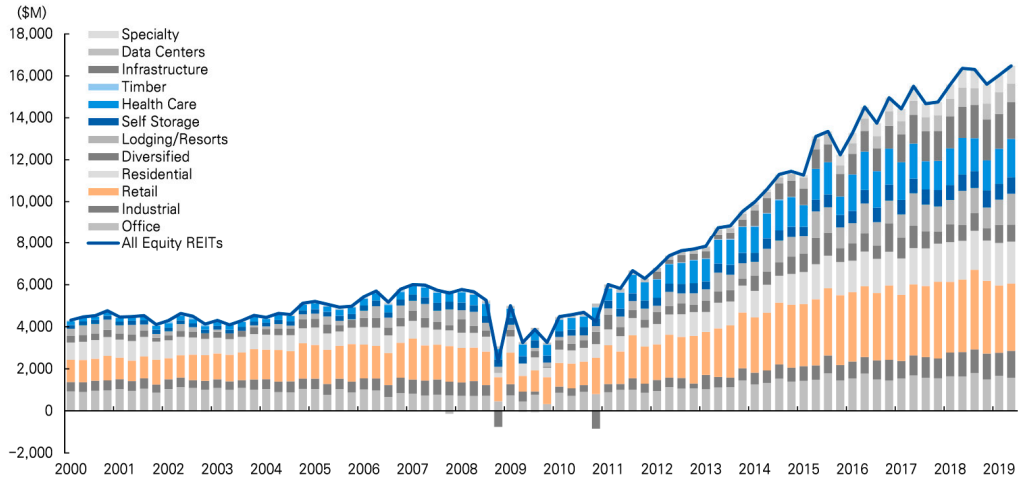

Figure 2. Global REITs (real estate investment trusts).

2.2. Blockchain Smart Contracts for Real Estate

Blockchain-based smart contracts for real estate basically have two databases: a Blockchain database where all transaction logs are stored, and a smart contract's status. The smart contract here is an application that can change its status; the status of the smart contract is a variable used by that application, and the input value to change it is included in the transaction. Smart contracts have two interfaces: one for transactions and one for queries. The interface through transactions is stored in the transaction database and is an approach that changes the status of smart contracts. Queries are operations that read the status of smart contracts without recording in the transaction database. Transactions execute write, delete, and modify, and queries only run queries through reads [18–20].

In addition, the types of Blockchain are divided into public, private, and consortium Blockchains (see Table 1).

Table 1. Type of Blockchain.

	Public Blockchain	**Private Blockchain**	**Consortium Blockchain**
Management Subject	All Participants	Managed by central institution	Participants in the Consortium
Network Participating Condition	Non	Managed by central institution	Non or managed by selected institution
Transaction Speed	Slow	Quick	Quick
Identification	Anonymous	Identifiable	Identifiable
Transaction Proof	Proof of work Algorithm, Transaction verifier cannot be known in advance	Transaction verification is made by central institution.	Transaction verifier is known through certification. Transaction verification and block

Therefore, since smart contracts share all the data with each other, they cannot be manipulated even if a particular user tries to manipulate the execution results of smart contracts. Blockchain can also ensure the integrity of smart contracts in a way that guarantees the integrity of all transactions. It can also reduce the execution costs and the likelihood of disputes in the contract by automatically implementing the contract if the conditions are met. In the process of programming the everyday language, however, costs of the contract writing phase may arise, and programming error problems may occur during this process [21–24]. Blockchain technology is behind Bitcoin development and is a key part. Blockchain is a ledger-based tamper-proof technology that allows for various use cases in various applications. It generally represents a database that is maintained and controlled continuously, taking into account the increasing number of nodes and the set of data samples collected. The main

elements of a node are the transactions made by the participants and the record blocks of those transactions. Here, the recorder block checks whether the transaction history is maintained in the correct order. This does not allow any manipulation of available data. When it is necessary to keep recorded data sequentially, the need for chain access arises. The transactions thus maintained are shared with the network of participating nodes. This eliminates the concept of a central server by identifying each node participating in the transaction sharing process using encryption. This allows secure authentication [25–30].

2.2.1. Save Records

The seller encodes the details of uploading the product, makes a transaction, and sends it to the Blockchain. When a product registration transaction occurs, all nodes in the network share a product registration transaction, create a block, and broadcast the block. Each node that receives the block adds the block to the end of its own Blockchain and applies the transactions stored in that block to synchronize its own smart contract database. This process allows all nodes in the Blockchain to share a smart contract status database.

2.2.2. Product Inquiry

Buyers look up products on the Blockchain network. Writing to smart contracts generates transactions, but reading values that were already stored does not [31–34]. Query information need not be synchronized with the Blockchain; it can respond immediately regardless of the timing of the block synchronization as it only needs to look up the stored state values within the smart contract database without having to change any data in the Blockchain.

2.2.3. Contract Fulfillment

When a buyer sends a product purchase transaction, the transaction is shared and synchronized with the Blockchain network. Product buyers are registered in the smart contract database on all nodes, and money is sent to the sellers. The ownership of the registered content is then transferred to the buyer (see Figure 3).

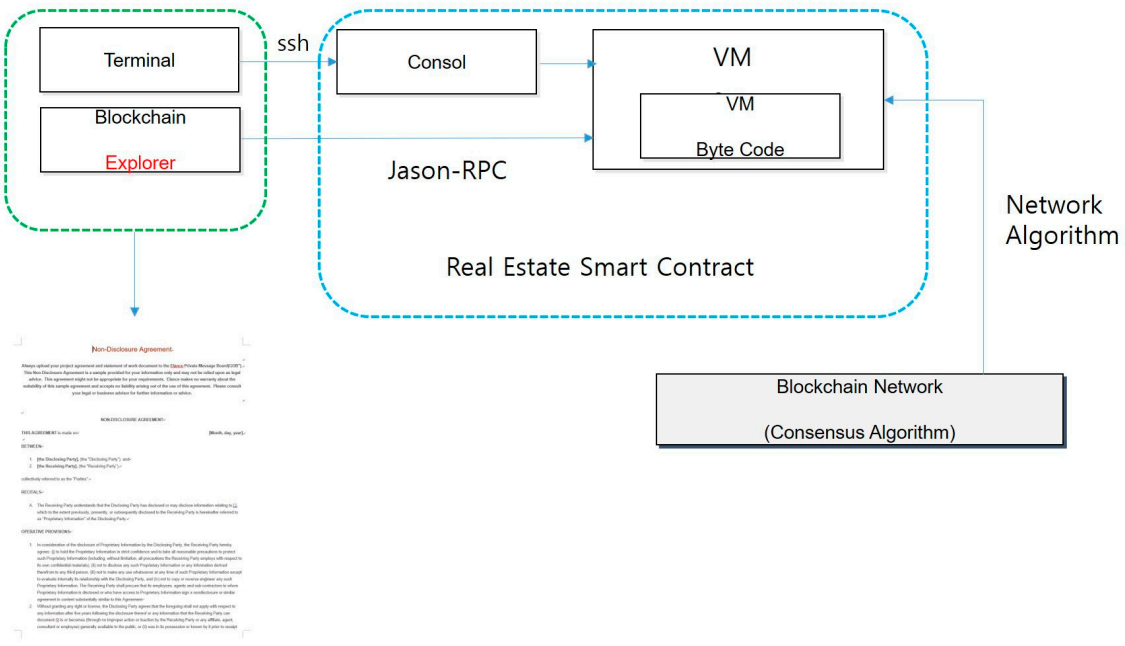

Figure 3. Smart contract for real estate.

2.3. Deep Belief Network

Verification of ledger data is very important in real estate transactions. An in-depth trust neural network is used to compare these real estate ledger data. The dep belief network (DBN) is an in-depth neural network based on algorithm-based, unsupervised learning methods for verification of real estate ledger and contracts [35–38]. A graph-generated model is used in machine learning, which means an in-depth neural network consisting of multiple layers of potential variables in deep learning. There is a link between layers for this ledger data analysis, but none between units within the hierarchy. The deep belief network can be used for prior learning due to its nature as a generating model; after learning the initial weight through prior learning, it can be used for unadjusted weights through reverse waves or other discriminating algorithms. These characteristics are very useful when training data are small, in which case the impact of the initial weighting on the resulting model is greater [39–42]. The initial value of the pre-learned weighting is closer to the optimum weighting than the initial value of the weighting set arbitrarily, which enables the performance and speed improvement of the unadjusted phase. Each layer of deep trust neural network usually takes the form of a restricted Boltzmann machine (RBM). DBN is taught in a non-instructed manner by layer, where each layer usually takes the form of an RBM. An explanation on how to train DBN while stacking up RBMs is provided below [43–45]. As an energy-based generation model, RBM is a unidirectional dichotomy graph consisting of visible units and hidden units. Connection exists only between visible and hidden units. Once the RBM is trained, other RBMs are stacked on top of it, forming a multi-tier model. Every time an RBM is stacked, the top layer of an already trained RBM is used as the input of a new RBM. Using this input, the new RBM is trained, and the process is repeated until as many layers as desired are built. Using this iterative algorithm, we use only deep trust nerves for future real estate transactions.

2.4. Security and Privacy

A Blockchain is a distributed book in which several entities record and verify transactions together. In order for the majority to verify the transaction details, the transaction details must be made public by default. Here, the deal means 'when and how much tokens I sent to whom'. Although where and who sent them on the block chain are recorded, individuals appear in the form of complex addresses, not specific. At this level, however, privacy is not completely protected. This is because there are ways to link addresses to actual individuals by analyzing the disclosed transactions. Recently, even companies that analyze the details of transactions in the crypto market and identify actual owners have emerged and provide them to law enforcement agencies or banks. Given this, it seems that the complex address of the crypto cannot completely conceal its identity. Through the block chain, it is revealed to everyone when, what, and how much we paid for the purchase. If the identity of the block chain is revealed, there is a risk that such sensitive personal information will be disclosed. To solve the privacy problem of smart contracts described above, various projects are creating privacy-preservation smart contracts. Privacy smart contracts are technologies that guarantees the privacy of smart contracts. Then, in order to provide privacy to users on the block chain but also to support programming features that enable smart contracts to operate, the company must develop a Blockchain privacy technology to meet certain conditions.

3. Neural Algorithm Blockchain Smart Contract for Secure P2P Real Estate Transactions

This study first diagnosed the problems in real estate transactions. As a design model of artificial intelligence, the Blockchain algorithm was presented to overcome these problems. In addition, this research was designed for establishing research models and important ledger data image correction and Blockchain smart contract for block verification [46,47]. The deep belief network algorithm was also applied for the comparative verification of the director of the required real estate contract. Therefore, the data consistency was verified by the correction image verification comparison algorithm, and the research intangible was then established to compare and verify the practical Byzantine fault tolerance

(PBFT) algorithm of Hyperledger through the Blockchain agreement process. Subsequently, a process for verifying the real estate contract was established (see Figure 4).

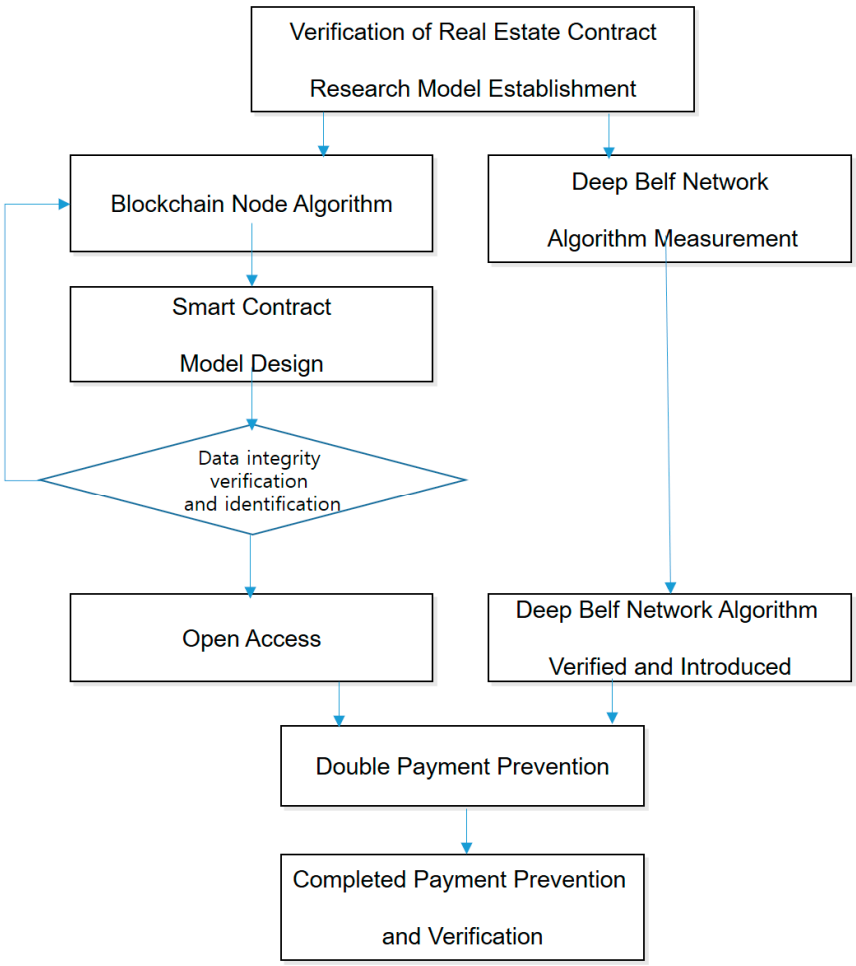

Figure 4. Research plan for real estate.

3.1. VM Study Methodology for Blockchain Real Estate Contract Validation

3.1.1. Study Methodology

The purpose of this study is to use a more secure encryption algorithm to make smart contracts in Blockchain. These encryption algorithms are implemented based on the ElGamal encryption system. A useful ElGamal commutation is also provided for the certificate of quantum knowledge. This applies a short signature using the binding properties of an elliptical curve using BLS (Bone–Lynn–Schacham). It also uses the technology of a more stable elastic curve Qu-Vanstone (ECQV) system to apply certificate-based technology. These smart contours allow uploading a script format created after encryption to inside the VM, enabling working independently from the programmer. It also implements encryption library APIs for the verification of real estate contracts so that they can be performed on their own embedded form so as not to have problems with the dependencies required for implementing encryption. This study consists of a property-based methodology to validate these real estate contracts and a holder-hold creating proof that all nodes on the Blockchain have delivered evidence to the issuer, verifier to validate the evidence and to the issuer. The holder may have a certificate, followed by proof of the following inputs:

1. Credentials
2. Key pair for credentials

3. Verifier information
4. Credentials result
5. Verifier information result

The evidence is structured in the form of JSON, and anyone can enter it as shown below to verify the evidence (see Figure 5).

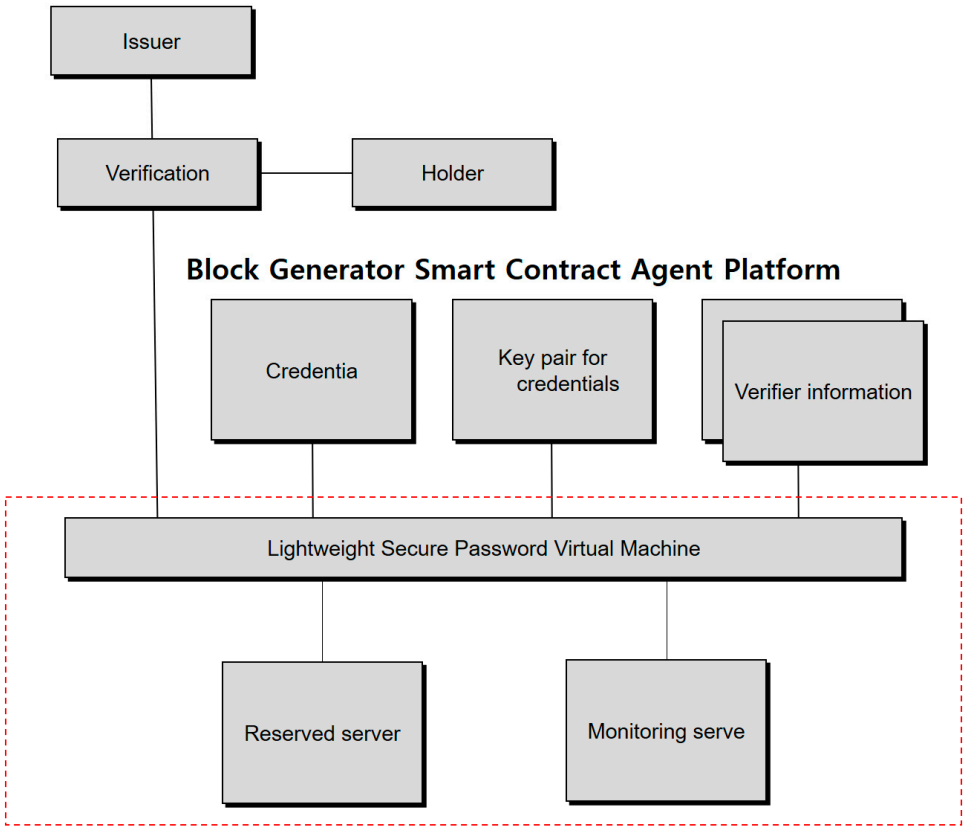

Figure 5. Block generator smart contract agent platform.

3.1.2. Lightweight Secure Password Virtual Machine

VM for smart contracts under Blockchain has a basic four-bit memory manager for lightweight secure password virtual machines; it should be designed to include basic applications and advanced scripting languages easily, including terminal (Android, iOS) and Linux as well as library components for Windows. There is no separate call API (application programming interface), and it should be provided in a single call format in the form of running and returning scripts; the way the host program calls them should be configured to be very simple. The ability to copy results within the memory buffer should be provided for error messages and debugging. The call should proceed with information in JSON format without running the script. Bindings of various languages are provided. Both x86 and ARM (advanced RISC machine) architectures should be provided as targets. JavaScript and Webassembly compilation using LLVM (low level virtual machine)-based emscripten SDK (software development kit) shall be provided. The dynamic library should be provided in the Android architecture. The iOS architecture provides a static library.

3.1.3. Test Verification Tool for Lightweight Security IoT Gateway (GW)

Web services should be provided to create Contracts. Web-style file upload function should be supported. Asymmetric AES-GCM (advanced encryption standard-galois/counter mode) testing is possible within a browser.

3.2. Blockchain Smart Contract VM Model

It should run in different execution environments, giving data integrity and consistency in cryptographic transformation in managing Blockchain. Through the cryptographic transformation, each transformation must achieve the same result. Therefore, implementation is necessary so that there is no cryptographic function that produces different results for different platforms and languages. End-to-end passwords should be enforced in peer-to-peer communications for Blockchain.

The parser should be advanced to interpret and execute complex arithmetic operations of ECP modules (elliptic curve primitives) securely.

Similarity to the scripting method used by software cryptographers should be maintained. New encryption models will be implemented in advanced models and supported in template form. The ElGamal encryption system is implemented, and ElGamal communication, which is useful for young knowledge proof, is provided. It also applies a short signature using the binding properties of an elliptical curve using BLS (Bone–Lynn–Schacham). Certificate-based technology should be provided using the technology of the elliptic curve Qu-Vanstone (ECQV) system.

It provides the ability to upload the format of a script generated by a cryptographer inside the VM to perform tasks independently from the programmer, so that systems can be embedded and lightweight on their own; this way, there is no problem with the cryptographic library API and the dependencies required to implement encryption. As a step toward upgrading attribute-based qualification, it consists of issuer, which is used by a number of IoT GWs with lightweight crypto virtual machines, verifier, and holder, creating evidence that all nodes on the Blockchain have been delivered to the issuer, including evidence that they have requested a certificate.

3.2.1. Test Verification Tool for Network-Based Lightweight Security IoT GW (Terminal Mounted SW)

It provides web services to create network-based Contracts. It also supports a network-based, web-based file upload function and enables asymmetric AES-GCM testing within a browser.

3.2.2. Reserved Server (Server-Mounted S/W)

It provides functions for adjusting and controlling the values of sensors such as adjusting the temperature and humidity according to the schedule functions. It can also be controlled for a certain period of time, and it has the ability to control when the user is present and not to control when the user is not present, and vice versa. The alarm function is supported according to the schedule setting.

3.2.3. Monitoring Server (Server-Mounted S/W)

It provides a function that stores the collected sensor data by linking it with GW, a device equipped with a lightweight rock-protection virtual machine linked to a number of sensor terminals. It monitors the status of the cryptographic virtual machine and the status of the terminal GW. It also provides a dashboard that visualizes statistics and status of data linked to a terminal GW (see Figure 6).

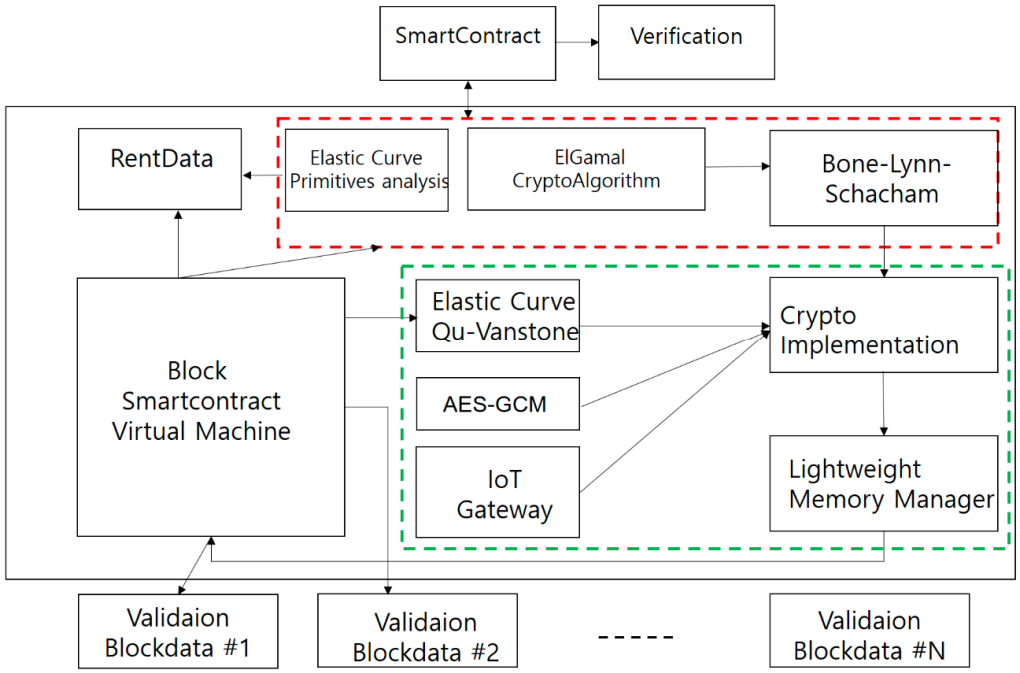

Figure 6. Smart contract architecture for real estate.

3.3. Blockchain Smart Contract Key Features

3.3.1. Lightweight Secure Password Language Interpreter

The main functions are as follows:

1. Ability to perform elastic curve primitives analysis.
2. New encryption model template functionality.
3. ElGamal commission features.
4. Bone–Lynn–Schacham (BLS) coupling for short signature.
5. Elastic curve Qu-Vanstone (ECQV) certificate-based technology.
6. Ability to upload script formats inside the VM.
7. Encryption library APIs and crypto implementation non-subjectivity features.
8. Property-based qualification technology.

3.3.2. Lightweight Secure Password Virtual Machine

1. Lightweight memory manager (4-byte default).
2. Single call form function.
3. Copy results within memory buffer.
4. JSON format information call function.
5. Bindings of various languages (minimum of three types or more).
6. Provides deployment of x86 or ARM architecture.
7. JavaScript/Webassembly compilation features.
8. Android/iOS library offerings.

3.3.3. Test Verification for Lightweight Secure IoT GW

1. Web service features to create Contracts.
2. Web-style file upload feature.
3. Asymmetric AES-GCM test function.

3.3.4. One Lightweight Secure IoT GW SYSTEM

1. Ability to upload and mount lightweight secure virtual machines.
2. Test command tool for testing lightweight secure virtual machines.
3. Secure operation check function in conjunction with various sensors.

3.4. Quantum Cryptography Method

3.4.1. Quantum Proof-Based Algorithm

It uses quantum encryption algorithms, and the circuit you want to create in the quantum encryption should be arithmetic. Circuit values in finite fields take the elements inside as input values, and output values are also elements of Blockchain. A circuit consists of a gate and a wire, using two numbers as input values to add or multiply the numbers and subsequently outputting the results through the output wire. The wire is responsible for conveying values inside and outside the gate, and the circuit shall satisfy the following:

1. Each new program requires a new setup, which in turn creates a new key.
2. Memory access or repeat count of repeat statements cannot vary depending on the program's input value.
3. Allowing data dependence on memory access requires heavy work, such as verification of the muffler pass.
4. Supports any program, but the circuit size increases inefficiently relative to the program size.

Existing studies show that as the program grows in size, the number of gates per cycle continues to increase. In this study, however, the number of gates per cycle should not change.

In the expression above, $X_{(l-1)}H$, $XH_{(l-1)}$ and $X_{(l-1)}W$, $XW_{(l-1)}$ mean the height and width, respectively, of the image printed in the l-1th composite layer. To solve the equation, first, calculate $lX(l)_{,m,n}$, $W(l)_{k,i,jm}$, and $X_{k,m,n}(l)/iW_{k,i,j}(l)$ as follows:

1. $\partial X(l)_{k,m,n} \partial W(l)_{k,i,j} = \partial \partial W\ (l)_{k,i,j}\{W(l)_{k,i,j} f(X_{(l-1)k,(i+m),(j+n)})\}$
2. $\partial X(l)_{k,m,n} \partial W(l)_{k,i,j} = f(X_{(l-1)k,(i+m),(j+n)})$

3.4.2. Quantum Proof-Based Algorithm

The circuit generator is an encryption algorithm representing the output of the circuit; since C relies only on the value rather than the program or main input value, when combined with the circuit verification system, the parameters of the verification system also become universal. In this case, all programs can be validated in one key generation, and keys that fit a given calculation range can then be selected. This reduces the cost of generating keys per program. In addition, when a particular byte code is executed, all nodes in the Ethereum network run the same byte code one by one to validate the transaction. At this time, if the virtual machine is equipped with a generic computational verification technology, the virtual machine need not execute transactions; if it performs verification of the knowledgeable evidence, however, it will know if the transaction is the correct one.

3.4.3. Virtual Machine Calculator

The virtual machine calculator should determine which part of the virtual machine's execution process needs to be modified to apply quantum cryptography authentication technology. In order for transactions to run, these byte codes are decomposed into Opcodes that are stacked and executed one by one. Gas costs are deducted for virtual machine execution before the Opcode executes.

3.4.4. Quantum Proof-Based Architecture

Quantum-encrypting smart contracts can perform various functions, such as ordinary computers, through transactions if smart contracts are supported. Anyone can access the transaction once it

is recorded in the Blockchain. If anonymity is required, the results of the performance should be communicated only to authorized persons without external exposure. Proof of quantum encryption can be used for this purpose. In general, quantum cryptographic evidence is bought and applied when a question is given to verify that the answer to that question is correct.

3.5. Deep Learning Smart Contract

Deep learning Blockchain-based smart contracts basically have two databases: a block chain database where all transaction logs are stored and a database that stores the state of smart contracts. In real estate transactions, smart contracts can be called applications that can change their status, and the status of smart contracts can be called variables used by those applications, and input to change them is included in transactions. Smart contract reveals two interfaces: one is a transaction, and one is a query. The interface through transactions is an approach that is stored in the transaction database and changes the state of smart contracts. A query is an operation that reads the status of a smart contract without recording in the transaction database. Transactions execute write, delete, and modify, and queries only execute queries through reads.

For example, 'product trading' works in a smart contract system as follows:

Blockchain deep learning-based record storage: Real estate sellers code content to upload products, create transactions, and send them to the block chain. When a product registration transaction occurs, all nodes in the network share the product registration transaction, create the block, and then broadcast the block. Each node that receives the block adds the block to the end of its own bloc chain and applies the transactions stored in that block to synchronize its own smart contract database. This process allows nodes in all Blockchains to share a smart contract status database.

Inquiry of products based on deep learning of block chain: buyer checks products on the block chain network. Writing to smart contracts generates transactions, but reading values that have already been stored does not cause a transaction. Query information does not need to be synchronized to the block chain and can respond immediately, regardless of the timing of the block sync, because only the stored state values in the smart contract database need to be viewed without having no need to change any data in the Blockchain.

Implementing a deep learning-based contract for the block chain: When a real estate buyer sends a product purchase transaction, he shares the transaction and synchronizes it to the block chain network. Registers the purchaser of goods in the smart contract database on all nodes and sends money to the seller. The ownership of the registered property is then transferred to the buyer.

As such, Blockchain deep learning-based smart contracts share all data with each other, so even if a particular user tries to manipulate the execution results of smart contracts, they cannot be manipulated. The integrity of smart contracts can also be ensured in such a way that the block chain ensures the integrity of all transactions. In addition, the execution cost of the contract and the possibility of dispute can be reduced by automatically implementing the contract if conditions are met. However, the cost of the contract preparation phase may be incurred in programming the daily language, and there is a possibility of programming error in this process.

4. Program Class-Diagram and Source Code

4.1. Program Class-Diagram

1. Measure the memory usage of lightweight rock-dissolving virtual machines
2. Establish an environment for monitoring memory usage and measure memory usage
3. Binary size for lightweight rockfish virtual machines
4. Build a build environment, create a binary, and measure its size
5. Speed of blockification of lightweight rockfish virtual machines
6. Establish a debugging environment and measure the speed of cryptography in virtual machines
7. Property-based identity assurance speed

8. After establishing an environment for issuer, holder, and verifier, the holder is created
9. Request to the issuer and measure the speed of identity assurance up to the stage where the verifier verifies it

Among class diagrams, RentBlock is sent to a smart contract intelligence agent, with the e-bidder accepted as filter, parameter sharing, and spatial arrangement (see Figure 7).

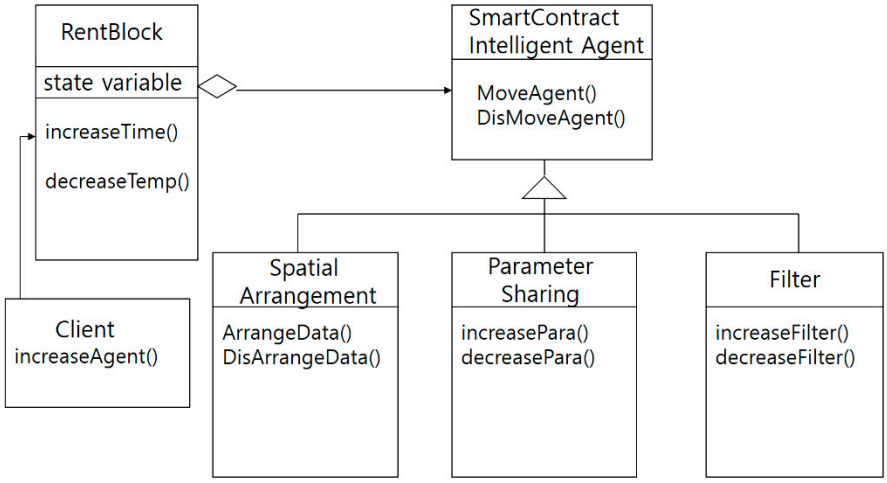

Figure 7. Unified modelling language (UML, class diagram) for real estate.

4.2. Program Source Code

Reveals the source code of the rental block chain. It also sharpens the import of smart contract as real estate and sends style ="transaction" source code. Thus, these smart contract.java codes are opened (see Figure 8).

Smart Contract.Java
import Smart Contract as Real Estate RealEstate.set(style="transaction") tips = RealEstate.load_dataset("tips") g = Real Estate.jointplot("total_bill", "tip", data=tips, kind="reg", truncate=False, xlim=(0, 60), ylim=(0, 12), color="m", height=7)

```
V[0,0,0] = np.sum(X[:5,:5,:]  * W0) + b0 # np.dot(X[:5,:5,:],W0) + b0
V[1,0,0] = np.sum(X[2:7,:5,:] * W0) + b0
V[2,0,0] = np.sum(X[4:9,:5,:] * W0) + b0
V[3,0,0] = np.sum(X[6:11,:5,:]* W0) + b0
```

Figure 8. Smart contract source code for real estate.

The Transaction.java file gives a smart contract as real estate and stores different variable values. After that, it is drawn as a random function (see Figure 9).

Transaction.Java

```
import Smart Contract as Real Estate
import seaborn as transaction
import matplotlib.pyplot as plt

transaction.set(style ="dark")
rs = np.random.RandomState(46)

# Set up the matplotlib figure
f, axes = plt.subplots(3, 6, figsize = (4, 3), sharex = True, sharey = True)

# Rotate the starting point around the cubehelix hue circle
for ax, s in zip(axes.flat, np.linspace(0, 3, 10)):

    # Create a cubehelix colormap to use with kdeplot
    cmap = sns.cubehelix_palette(start = s, light = 1, as_cmap = True)

    # Generate and plot a random bivariate dataset
    x, y = rs.randn(2, 50)
    sns.kdeplot(x, y, cmap = cmap, shade = True, cut = 5, ax = ax)
    ax.set(xlim = (-3, 3), ylim = (-3, 3))

f.tight_layout()
```

```
INPUT: [224x224x3]        memory:  224*224*3=150K   weights: 0
CONV3-64: [224x224x64]  memory:  224*224*64=3.2M   weights: (3*3*3)*64 = 1,728
CONV3-64: [224x224x64]  memory:  224*224*64=3.2M   weights: (3*3*64)*64 = 36,864
POOL2: [112x112x64]  memory:  112*112*64=800K   weights: 0
CONV3-128: [112x112x128]  memory:  112*112*128=1.6M   weights: (3*3*64)*128 = 73,728
CONV3-128: [112x112x128]  memory:  112*112*128=1.6M   weights: (3*3*128)*128 = 147,456
POOL2: [56x56x128]  memory:  56*56*128=400K   weights: 0
CONV3-256: [56x56x256]  memory:  56*56*256=800K   weights: (3*3*128)*256 = 294,912
CONV3-256: [56x56x256]  memory:  56*56*256=800K   weights: (3*3*256)*256 = 589,824
CONV3-256: [56x56x256]  memory:  56*56*256=800K   weights: (3*3*256)*256 = 589,824
POOL2: [28x28x256]  memory:  28*28*256=200K   weights: 0
CONV3-512: [28x28x512]  memory:  28*28*512=400K   weights: (3*3*256)*512 = 1,179,648
CONV3-512: [28x28x512]  memory:  28*28*512=400K   weights: (3*3*512)*512 = 2,359,296
CONV3-512: [28x28x512]  memory:  28*28*512=400K   weights: (3*3*512)*512 = 2,359,296
POOL2: [14x14x512]  memory:  14*14*512=100K   weights: 0
CONV3-512: [14x14x512]  memory:  14*14*512=100K   weights: (3*3*512)*512 = 2,359,296
CONV3-512: [14x14x512]  memory:  14*14*512=100K   weights: (3*3*512)*512 = 2,359,296
CONV3-512: [14x14x512]  memory:  14*14*512=100K   weights: (3*3*512)*512 = 2,359,296
POOL2: [7x7x512]  memory:  7*7*512=25K  weights: 0
FC: [1x1x4096]  memory:  4096  weights: 7*7*512*4096 = 102,760,448
FC: [1x1x4096]  memory:  4096  weights: 4096*4096 = 16,777,216
FC: [1x1x1000]  memory:  1000 weights: 4096*1000 = 4,096,000

TOTAL memory: 24M * 4 bytes ~= 93MB / image (only forward! ~*2 for bwd)
TOTAL params: 138M parameters
```

Figure 9. *Cont*.

RentBlock.Java

```
package com.rentBlock.blockchain;
import java.util.List;
public class Block {
    private int previousHash;
    private List<Transaction> transactions;
    public Block(int previousHash, List<Transaction> transactions) {
        this.previousHash = previousHash;
        this.transactions = transactions;
    }
    public int getPreviousHash() {
        return previousHash;
    }
    public void setPreviousHash(int previousHash) {
        this.previousHash = previousHash;
    }
    public List<Transaction> getTransactions() {
        return transactions;
    }
    public void setTransactions(List<Transaction> transactions) {
        this.transactions = transactions;
    }
    @Override
    public boolean equals(Object o) {
        if (this == o) return true;
        if (o == null || getClass() != o.getClass()) return false;
        Block = (Block) o;
        if (previousHash != block.previousHash) return false;
        return transactions != null ? transactions.equals(block.transactions)
: block.transactions == null;
    }
    @Override
    public int hashCode() {
        int result = previousHash;
        result = 31 * result + (transactions != null ? transactions.hashCode() : 0);
        return result;
    }
}
```

Figure 9. Smart contract calculation source code for real estate.

5. Experimental Method

5.1. Performance Measurement Evaluation Items

The experimental method of the first Blockchain smart contract consists of 250 nodes. The storage of these configured nodes and consensus are used as consensus algorithms. The necessary experimental methods are set up, and the performance information required for this time is shown.

1. Measure memory usage of lightweight rock-dissolving virtual machines
2. Establish an environment for monitoring memory usage and measure memory usage
3. Binary size for lightweight rockfish virtual machines
4. Build a build environment, create a binary, and measure its size
5. Speed of blockification of lightweight rockfish virtual machines
6. Establish a debugging environment and measure the speed of cryptography in virtual machines
7. Property-based identity assurance speed
8. After establishing an environment for issuer, holder, and verifier, the holder is created
9. Request to the issuer and measure the speed of identity assurance up to the stage where the verifier verifies it.

5.2. Performance Measurement Offering

To measure quantitative targets, quantitative experimental data are first discovered through quantitative target items, preparation for evaluation, evaluation methods, and configuration of test networks (see Table 2).

Table 2. Performance measurement offering.

Quantitative Objective Items	Preparing for Evaluation	Evaluation Method	Composition of Test Network
Preprocessing time	Network configuration using nodes with four preprocessing.	Perform pretreatment for a total of 10 times and derive the average.	Blockchain—four nodes. One load test node.
Percentage of real-time throughput of preprocessing code	Network configuration using nodes with four preprocessing capabilities. Network configuration using nodes that exclude four preprocessing functions.	Load zoom (60 min) using uniform remittance transactions across all nodes. Average speed is obtained according to the procedure above. Measure a total of 10 repetitions and derive the average.	Blockchain—eight nodes. Two load test nodes
Real-time transaction performance	Configure your network using four nodes.	Load zoom (60 min) using uniform remittance transactions across all nodes. Average speed is obtained according to the procedure above. Measure a total of 10 repetitions and derive the average.	Blockchain—16 nodes. One load test node

5.3. Performance Measurement Data

The experimental data were verified by calculating the number of smart contract users using rental Blockchain. For the completeness of functional suitability, the Blockchain system, which is fully functional, evaluated whether all specified functions and scenarios were implemented and whether the functions and scenarios implemented in the Blockchain system worked normally for correct function implementation accuracy. An assessment was done on the algorithms used to search for peers/nodes that are appropriate for peer/node search algorithms. In order for a node to become part of the network, it must be connected to another node, but it has been validated against algorithms (such as near peer-first searches) that search for other nodes/peer within the network without a central server. An evaluation of whether the gas and gas limits are properly set was carried out for the adequacy of the gas and gas limit settings. If the transaction was set too low as a user fee paid to the network for transactions and used as a reward for the nodes that maintain the network, the transaction may not be processed; if it was too high, the user set out to pay an excessive user fee, and an assessment was done on whether the policy of keeping and backing up private keys was appropriate in keeping with the personal key retention/backup policy. In addition, an assessment of the number of transactions

completed in a given time period, which is the transaction throughput, was performed to experiment on the Blockchain capacity. Marked as transaction per second per second, the total completed transactions were divided by the total time required in seconds. The query throughput is an evaluation of the number of query operations completed during the specified time period. Marked by read per second, the experiment was conducted by dividing the total lookup operation by the total number of seconds, and the processing performance scalability considered how to improve processing performance as well as the resulting tradeoff. It defined methods to improve processing performance, such as block size, block mining difficulty, gas limit, and transaction size, and tested the impact of each application on the overall quality.

Here (see Figure 10), when the average of 250 Blockchain smart contract nodes was paid 10 times, 3500 TPS was shown, with the results of experimenting 100 times steadily taking about 4000 TPS (see Figure 11). This experiment showed stable performance by breaking away from the performance of the existing Blockchain.

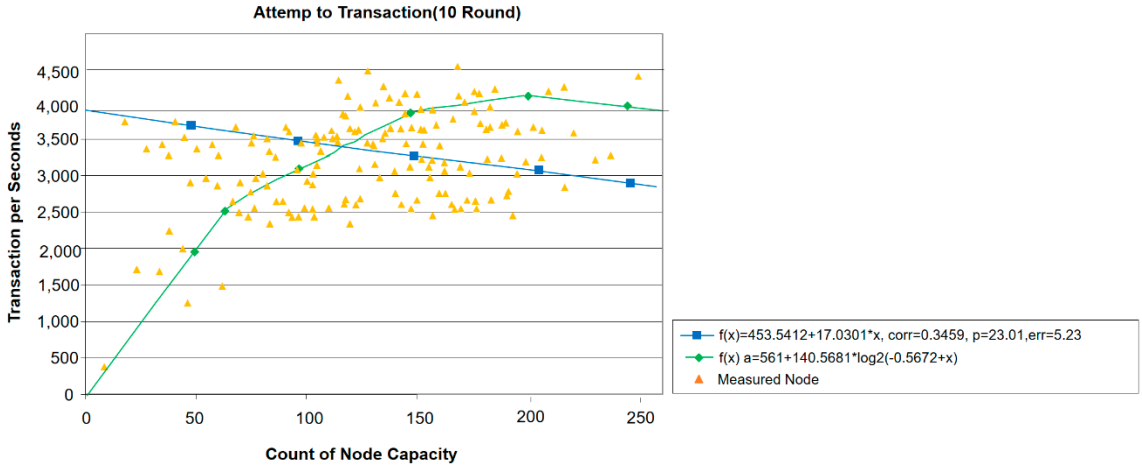

Figure 10. Experimental method (average value of Experiment 10 Round).

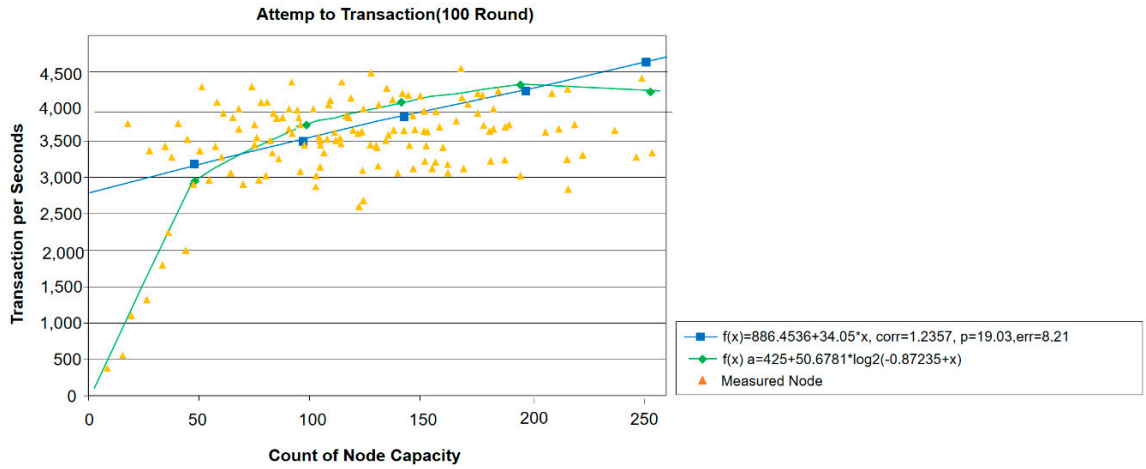

Figure 11. Experimental method (average value of Experiment 100 Round).

The experimental data were verified by calculating the number of smart contract users using rental Blockchain. For the completeness of functional suitability, the Blockchain system, which is fully functional, evaluated whether all specified functions and scenarios were implemented and whether the functions and scenarios implemented in the Blockchain system worked normally for correct function implementation accuracy. An assessment was done on the algorithms used to search for peers/nodes that are appropriate for peer/node search algorithms. In order for a node to become part.

6. Discussion

6.1. Discussion

This study verified the safest method for real estate transactions using the verification method of automated artificial intelligence Blockchain. Looking at the world's trends in real estate transactions, we could see that there were many more bankruptcy deals going on. Still, it is true that, despite the heavy price of such real estate transactions, centralized real estate agents wield a lot of influence. Nonetheless, it is based on artificial intelligence Blockchain with more transparency and stability while slightly reducing the role of these certified brokers by verifying the research methodology and preferring trade by automatically lowering central fees. Therefore, the values for credentials, key pair for credit, and verifier information are obtained. We designed the architecture model lightweight secure password virtual machine, a test verification tool for lightweight security IoT GW, and developed a smart country virtual model. Nonetheless, various design models were still presented to apply artificial intelligence Blockchain. The verification tool for network-based lightweight security IoT GW, etc., was tested. Despite providing such good performance data and verifiable data, however, the company still uses low Blockchain automation performance and quantum cryptography to overcome the problems and introduce a model of smart contract-based real estate Blockchain. This Blockchain conducted research on safe transactions in the future. In the future, we will apply the Blockchain for transparent trading of energy by applying it to energy, solar energy, carbon emission rights, etc. We will also apply the Blockchain smart contract technology for transparent trading of trade products. There are also many problems in real estate transactions for the vulnerability of cyber physics systems. The study proposes a system in which computing and the physical world are organically fused through networking in a real estate transaction network where all real estate information in a cyber-physics system is connected to each other and exchanged information, and objects are communicated to each other and controlled automatically and intelligently. It should also be developed in a future-oriented and developmental form of the existing embedded system. In addition, the real estate transaction system does not move separately, but studies a system that improves the results by controlling them as a group. Therefore, we study zero-defect intelligence systems that not only prevent various accidents in advance, but also ensure high reliability.

6.2. Smart Contract-Based Real Estate Blockchain

Smart contract Sivan real estate transaction Blockchain has established an actual model. This screen shows the main screen that can be used by sellers and consumers for real estate transactions using mobiles, the Internet, etc. (see Figure 12).

The next screen is for the seller to upload the real estate with his or her credit information, etc. for real estate transactions. It can feature video information (see Supplementary Materials), image information, and text information. It also shows a personalized screen to make the necessary transactions (see Figure 13).

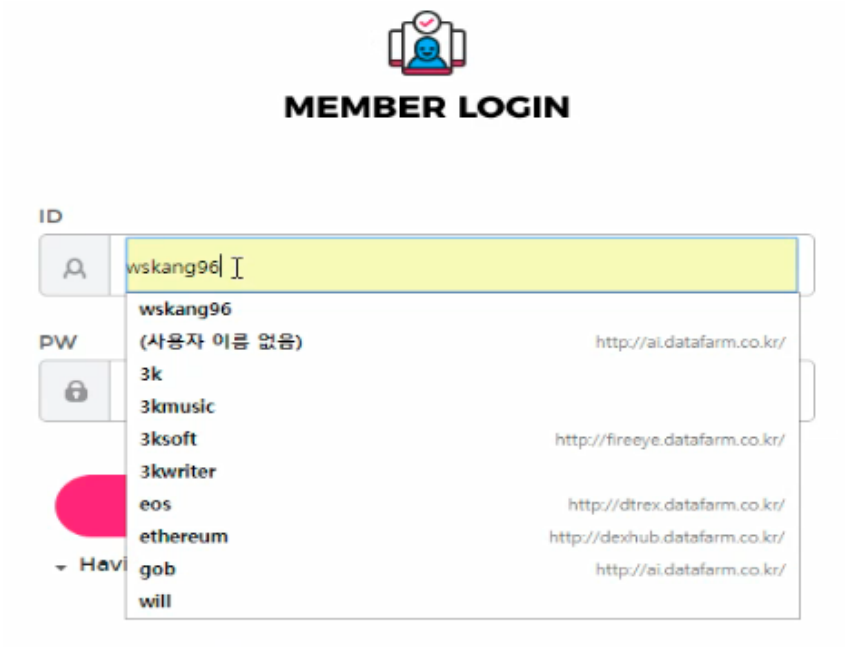

Figure 12. Main screen of Blockchain in smart contract real estate transactions.

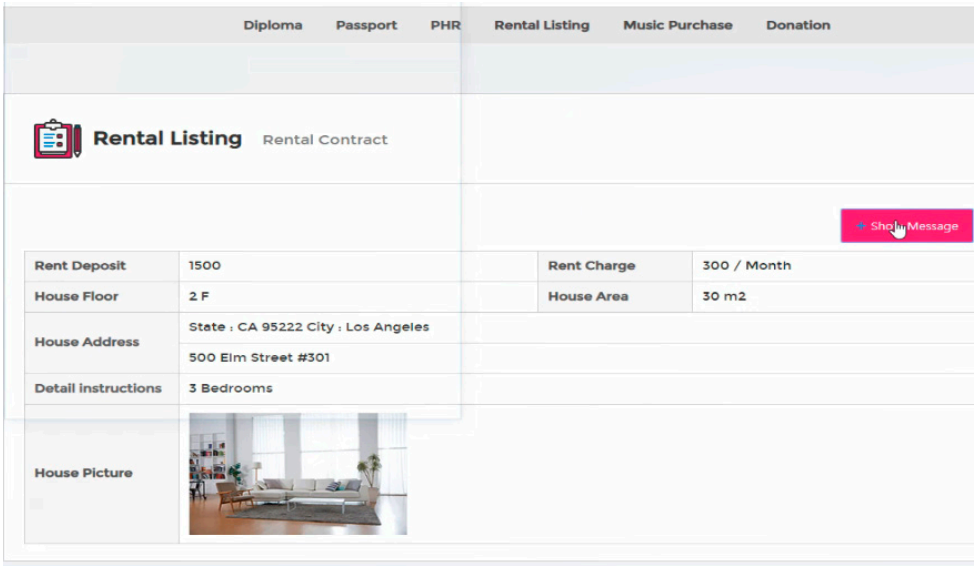

Figure 13. Smart contract real estate transaction Blockchain transaction screen.

There is a significant downside to real estate transactions online, however. When we chat with each other through a chat program, all of the conversations are saved to Blockchain engines and Blockchain nodes. The stored Blockchain comes with a "wife's seal function" (see Figure 14).

The next screen shows an actual smart contract between real estate traders and sellers. These smart contracts include each other's personal information and stored initial data through pre-certification. The data stored on the node is stored in a contract file that shows the initial data values (see Figure 15).

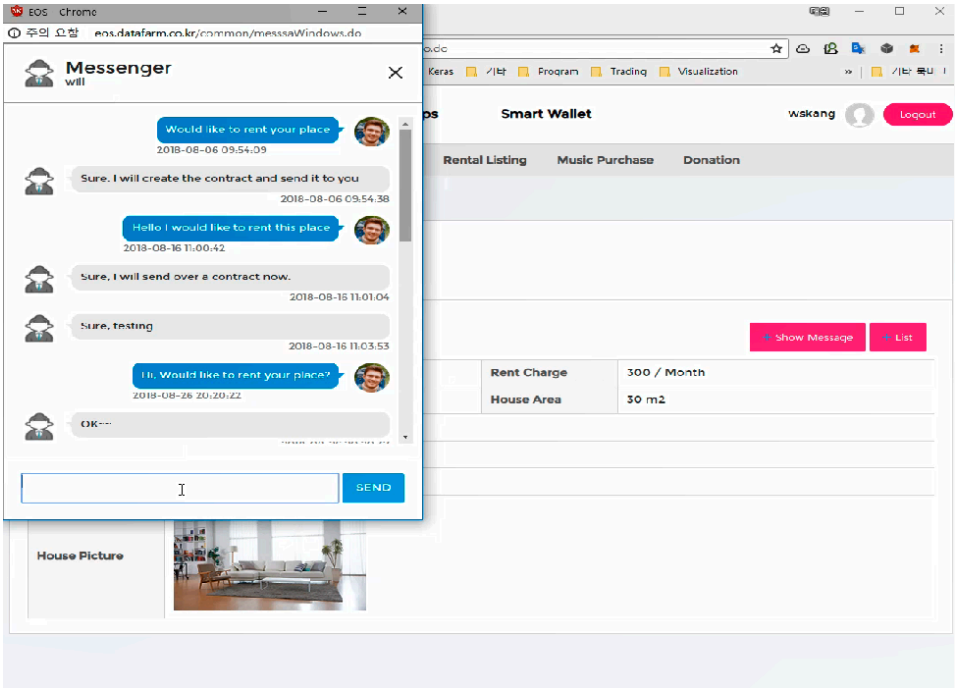

Figure 14. Chat screen between parties of the Blockchain contract in real estate transaction.

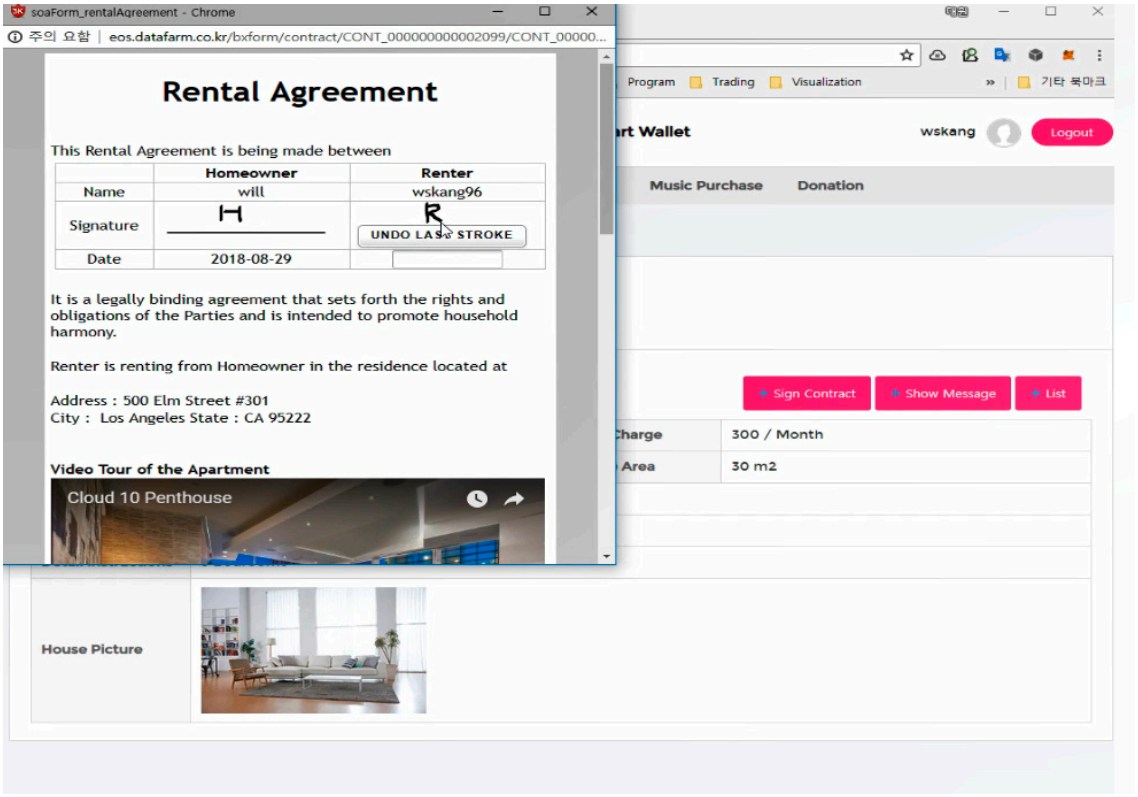

Figure 15. Blockchain smart contracts for real estate transactions.

It also shows a screen that actually deposits down payment to do real estate transactions. The deposit will be saved to the node of the Blockchain so that neither sellers nor buyers can take it. This is because it will be remitted to the seller only after a secure transaction (see Figure 16).

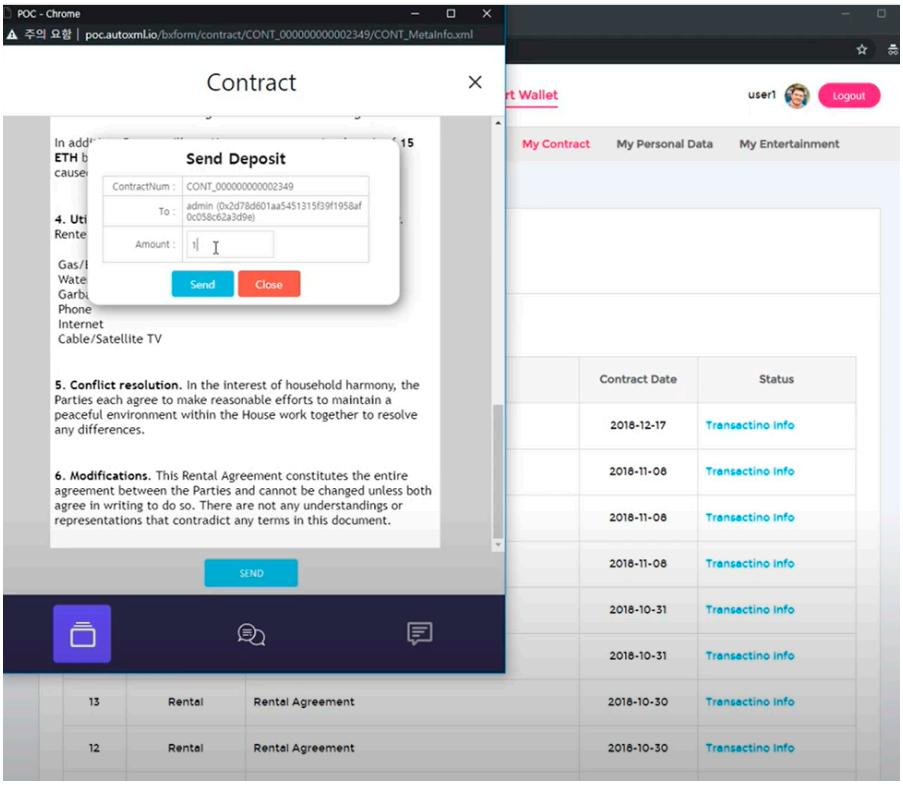

Figure 16. Blockchain deposit screen for real estate transaction.

Finally, it is mounted on a Blockchain engine called Ethe Sken, which can be seen stored on a Blockchain node that is no longer forged (see Figure 17).

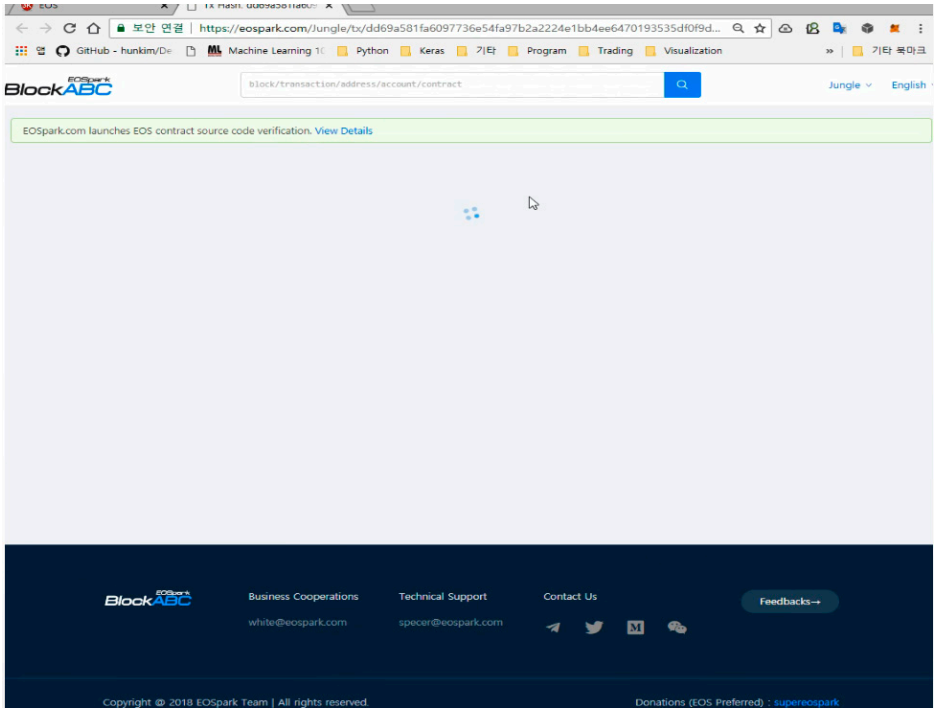

Figure 17. Real estate transaction Blockchain smart context contract results screen.

7. Conclusions

Real estate transactions have been on the rise every year, but for hundreds of years, they have only been done through real estate agents called real estate agents. Recently, an Internet page has emerged that allows personal transactions. Still, sellers and buyers have no trust in each other, so they distrust each other and fear transactions. Moreover, skipping notarization takes a lot of fees from real estate agents. There are many websites that can make P2P transactions between individuals, but there is a legal limit. This is because it is difficult to make real estate transactions safe without real estate agents. The study began with the idea that applying smart contracts to real estate transactions where such problems exist can solve these problems by using artificial intelligence block chain smart contracts as non-face-to-face rather than face-to-face meetings between contract parties.

The VM research methodology was studied, produced and presented for the verification of the real estate contract of the block chain. These attestation starts with credentials, key pairs of credentials, and verification information.

It also means that there is an artificial intelligence block chain in the existing block chain. In addition, within the methodology, testing and verification tools for lightweight network-based, light-weight security IOT GW (on-terminal-mounted SW), lightweight security encryption virtual machines and lightweight security IOT GW (on-server-mounted SW) were designed, including reserved servers (S/W) and monitoring servers (on-server-mounted S/W).

Smart contract VM model of the block chain: ElGamal Commission encryption algorithm was also used for Liang secure cryptographic language interpreters. It also used quantum cryptography algorithms that are needed.

In addition, the required UML source code has been released. The experimental method of the first block chain smart contract consists of 250 nodes. The storage and consensus of these configured nodes was used as a consensus algorithm. The necessary testing methods were also set to show the performance information required for this test.

Finally, 250 block chain nodes were formed. The network structure takes the form of a mesh.

After that, the performance test was conducted 10 to 100 times to show the average value. About 4000 TPS performance data were presented. Actual design-to-application technologies and scenarios have been introduced.

In the future, I want to contribute to the use of the safest block chain service in real estate transactions. Further, instead of eliminating the basic role of a licensed realtor, this paper aims to cooperate with the artificial intelligence block chain and the authorized relay company and overcome them using the block chain technology.

Supplementary Materials: The following are available online at http://www.mdpi.com/2079-9292/9/6/1052/s1, Video S1: the seller to upload the real estate with his or her credit information.

Author Contributions: Conceptualization, J.-H.H. and S.-K.K.; data curation, J.-H.H.; formal analysis, S.-K.K.; funding acquisition, J.-H.H.; investigation, J.-H.H. and S.-K.K.; methodology, J.-H.H.; project administration, J.-H.H. and S.-K.K.; resources, J.-H.H. and S.-K.K.; software, J.-H.H. and S.-K.K.; supervision, S.-K.K.; validation, S.-K.K.; visualization, J.-H.H. and S.-K.K.; writing—original draft, J.-H.H. and S.-K.K.; writing—review and editing, J.-H.H. and S.-K.K. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korean Government (MSIT) (No.2017R1C1B5077157).

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

P2P	Peer-to-Peer
APIs	Application Programming Interface
BLS	Bone–Lynn–Schacham
ECP	Elliptic Curve Primitives
ECQV	Elastic Curve Qu-Vanstone
GW	Gateway
IoT	Internet of Things
GPS	Global Positioning System
ARM	Advanced RISC Machine
LLVM	Low Level Virtual Machine
PBFT	Practical Byzantine Fault Tolerance
RBM	Restricted Boltzmann Machine
REITs	Real Estate Investment Trusts
SDK	Software Development Kit
TPS	Transaction Per Second
UML	Unified Modeling Language
VM	Virtual Machine

References

1. Hák, T.; Janoušková, S.; Moldan, B. Sustainable Development Goals: A need for relevant indicators. *Ecol. Indic.* **2016**, *60*, 565–573. [CrossRef]
2. Bizon, N. Efficiency and Sustainability of the Distributed Renewable Hybrid Power Systems Based on the Energy Internet, Blockchain Technology and Smart Contracts. Sustainability. Available online: https://www.mdpi.com/journal/sustainability/special_issues/Distributed_Renewable_Hybrid_Power_Systems (accessed on 6 May 2020).
3. Seong-Kyu, K.; Ung-Mo, K.; Jun-Ho, H. A study on improvement of blockchain application to overcome vulnerability of IoT multiplatform security. *Energies* **2019**, *12*, 402.
4. James, B.; Michael, C. How utilities are using blockchain to modernize the grid. *Harv. Bus. Rev.* **2017**, *23*, 1–8.
5. Griggs, D.; Stafford-Smith, M.; Gaffney, O.; Rockström, J.; Öhman, M.C.; Shyamsundar, P.; Noble, I. Policy: Sustainable development goals for people and planet. *Nature* **2013**, *495*, 305. [CrossRef]
6. Corea, F. AI and Blockchain. In *An Introduction to Data; Studies in Big Data*; Springer: Cham, Swizerland, 2019; Volume 50, pp. 69–76.
7. Chowdhury, M.J.; Ferdous, M.S.; Biswas, K.; Chowdhury, N.; Kayes, A.; Alazab, M.; Watters, P. A comparative analysis of distributed ledger technology platforms. *IEEE Access* **2019**, *7*, 167930–167943. [CrossRef]
8. Kuo, T.T.; Zavaleta, R.H.; Ohno-Machado, L. Comparison of blockchain platforms: A systematic review and healthcare examples. *J. Am. Med. Inform. Assoc.* **2019**, *26*, 462–478. [CrossRef]
9. Macdonald, M.; Liu-Thorrold, L.; Julien, R. The Blockchain: A Comparison of Platforms and Their Uses beyond Bitcoin; COMS4507-Advanced Computer and Network Security: 2017. Available online: https://www.researchgate.net/profile/Lisa_Liu-Thorrold/publication/313249614_The_Blockchain_A_Comparison_of_Platforms_and_Their_Uses_Beyond_Bitcoin/links/5894447baca27231daf63689/The-Blockchain-A-Comparison-of-Platforms-and-Their-Uses-Beyond-Bitcoin.pdf (accessed on 6 May 2020).
10. Beverly, Y.; Garcia-Molina, H. Pay: Micropayments for peer-to-peer systems. In Proceedings of the 10th ACM Conference on Computer and Communications Security, Washington, DC, USA, 27–31 October 2003; ACM: New York, NY, USA, 2003; pp. 300–310.
11. Lee, S.; Woo, H.; Shin, Y. Study on Personal Information Leak Detection Based on Machine Learning. *Adv. Sci. Lett.* **2017**, *23*, 12818–12821. [CrossRef]
12. Mnih, V.; Kavukcuoglu, K.; Silver, D.; Rusu, A.A.; Veness, J.; Bellemare, M.G.; Petersen, S. Human-level control through deep reinforcement learning. *Nature* **2015**, *518*, 529–533. [CrossRef]
13. Silver, D.; Schrittwieser, J.; Simonyan, K.; Antonoglou, I.; Huang, A.; Guez, A.; Chen, Y. Mastering the game of go without human knowledge. *Nature* **2017**, *550*, 354–359. [CrossRef]

14. DeepMind. AlphaGo Zero: Learning from the Scratch. Available online: https://deepmind.com/blog/alphago-zero-learning-scratch/ (accessed on 27 March 2020).
15. Bengio, Y.; Courville, A.; Vincent, P. Representation learning: A review and new perspectives. *IEEE Trans. Pattern Anal. Mach. Intell.* **2013**, *35*, 1798–1828. [CrossRef] [PubMed]
16. Wachsmuth, I.; Lenzen, M.; Knoblich, G. *Embodied Communication in Humans and Machines*; Oxford University Press: Oxford, UK, 2008.
17. Picard, R.W. Computer learning of subjectivity. *ACM Comput. Surv.* **1995**, *27*, 621–623. [CrossRef]
18. Charles, D. *The Expression of Emotions in Animals and Man*; Murray: London, UK, 1872; pp. 11–45.
19. Judith, E.; James, M.D.; Latane, B. Gregariousness in rats as a function of familiarity of environment. *J. Personal. Soc. Psychol.* **1969**, *11*, 107.
20. Sodhro, A.H.; Pirbhulal, S.; Hugo, V.; de Albuquerque, C. Artificial intelligence-driven mechanism for edge computing-based industrial applications. *IEEE Trans. Ind. Inform.* **2019**, *15*, 4235–4243. [CrossRef]
21. Wang, Z.; Peterson, J.L.; Rea, C.; Humphreys, D. Special issue on machine learning, data science, and artificial intelligence in plasma research. *IEEE Trans. Plasma Sci.* **2020**, *48*, 1–2. [CrossRef]
22. Wang, Y.; Kwong, S.; Leung, H.; Lu, J.; Smith, M.H.; Trajkovic, L. Brain-inspired systems: A transdisciplinary exploration on cognitive cybernetics, humanity, and systems science toward autonomous artificial intelligence. *IEEE Syst. Man Cybern. Mag.* **2020**, *6*, 6–13. [CrossRef]
23. Watkins, T. Cosmology of artificial intelligence project: Libraries, makerspaces, community and AI literacy. *ACM AI Matters* **2020**, *4*, 134–140. [CrossRef]
24. Seo, Y.-S.; Huh, J.-H. Automatic emotion-based music classification for supporting intelligent IoT applications. *Electronics* **2019**, *8*, 164. [CrossRef]
25. Manoj, N.; Kumara, P.; Mallickb, K. Blockchain technology for security issues and challenges in IoT. *Procedia Comput. Sci.* **2018**, *132*, 1815–1823.
26. Alfandi, O.; Otoum, S.; Jararweh, Y. Blockchain solution for IoT-based critical infrastructures: Byzantine fault tolerance. In Proceedings of the NOMS 2020–2020 IEEE/IFIP Network Operations and Management Symposium, Budapest, Hungary, 20–24 April 2020.
27. Aloqaily, M.; Boukerche, A.; Bouachir, O.; Khalid, F.; Jangsher, S. An energy trade framework using smart contracts: Overview and challenges. *IEEE Netw.* **2020**, 1–7. [CrossRef]
28. Tseng, L.; Wong, L.; Otoum, S.; Aloqaily, M.; Othman, J.B. Blockchain for managing heterogeneous internet of things: A perspective architecture. *IEEE Netw.* **2020**, *34*, 16–23. [CrossRef]
29. Akkaoui, R.; Hei, X.; Cheng, W. EdgeMediChain: A hybrid edge. Blockchain-based framework for health data exchang. *IEEE Access* **2020**. [CrossRef]
30. Liu, J.; Guo, S.; Shi, Y.; Feng, L.; Wang, C. Decentralized caching framework towards edge network based on blockchain. *IEEE Access* **2020**. [CrossRef]
31. Watson, D.; Tellegen, A. Toward a consensual structure of mood. *Psychol. Bull.* **1985**, *98*, 219–235. [CrossRef] [PubMed]
32. Huh, J.-H.; Kim, S.-K. The blockchain consensus algorithm for viable management of new and renewable energies. *Sustainability* **2019**, *11*, 3184. [CrossRef]
33. Burford, G.; Hoover, E.; Velasco, I.; Janoušková, S.; Jimenez, A.; Piggot, G.; Harder, M.K. Bringing the "missing pillar" into sustainable development goals: Towards intersubjective values-based indicators. *Sustainability* **2013**, *5*, 3035–3059. [CrossRef]
34. Gisbert, G. Policy: Base sustainable development goals on science. *Nature* **2012**, *491*, 35.
35. Wang, H.; He, D.; Ji, Y. Designated-verifier proof of assets for bitcoin exchange using elliptic curve cryptography. *Future Gener. Comput. Syst.* **2017**, *107*, 21–24. [CrossRef]
36. Löbbe, S.; Hackbarth, A. Chapter 15: *The Transformation of the German Electricity Sector and the Emergence of New Business Models in Distributed Energy Systems*; Elsevier: Amsterdam, The Netherlands, 2017; pp. 287–318.
37. Keesstra, S.D.; Bouma, J.; Wallinga, J.; Tittonell, P.; Smith, P.; Cerdà, A.; Bardgett, R.D. The significance of soils and soil science towards realization of the United Nations Sustainable Development Goals. *Soil* **2016**, *2*, 111–128. [CrossRef]
38. Chen, Y. Blockchain tokens and the potential democratization of entrepreneurship and innovation. *Bus. Horiz.* **2017**, *61*, 567–575. [CrossRef]
39. Kshetri, N. Blockchain's roles in meeting key supply chain management objectives. *Int. J. Inform. Manag.* **2018**, *39*, 80–82. [CrossRef]

40. Savelyev, A. Copyright in the blockchain era: Promises and challenges. *Comput. Law Secur. Rev.* **2018**, *34*, 550–561. [CrossRef]
41. Hajer, M.; Nilsson, M.; Raworth, K.; Bakker, P.; Berkhout, F.; De Boer, Y.; Kok, M. Beyond cockpit-ism: Four insights to enhance the transformative potential of the sustainable development goals. *Sustainability* **2015**, *7*, 1651–1660. [CrossRef]
42. Dorri, A.; Kanhere, S.S.; Jurdak, R.; Gauravaram, P. Blockchain for IoT security and privacy: The case study of a smart home. In Proceedings of the 2017 IEEE International Conference on Pervasive Computing and Communications Workshops (PerComWorkshops), Kona, HI, USA, 13–17 March 2017.
43. Imbault, F.; Swiatek, M.; De Beaufort, R.; Plana, R. The green blockchain: Managing decentralized energy production and consumption. In Proceedings of the 2017 IEEE International Conference on Environment and Electrical Engineering and 2017 IEEE Industrial and Commercial Power Systems Europe (EEEIC/I&CPS Europe), Milan, Italy, 6–9 June 2017; IEEE: Piscataway, NJ, USA, 2017; pp. 1–5.
44. Underwood, S. *Blockchain beyond Bitcoin. Communications of the ACM*; ACM: New York, NY, USA, 2016; Volume 59, pp. 15–17.
45. Wang, W.; Chang, Y.; Wang, H. An Application of the spatial autocorrelation method on the change of real estate prices in Taitung City. *ISPRS Int. J. Geo-Inf.* **2019**, *8*, 249. [CrossRef]
46. Jayantha, W.M.; Yung, E.H.K. Effect of Revitalisation of historic buildings on retail shop values in urban renewal: An empirical analysis. *Sustainability* **2018**, *10*, 1418. [CrossRef]
47. Bełej, M.; Figurska, M. 3D Modeling of Discontinuity in the spatial distribution of apartment prices using voronoi diagrams. *Remote Sens.* **2019**, *12*, 229. [CrossRef]

© 2020 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (http://creativecommons.org/licenses/by/4.0/).