Integration of Blockchain with Energy Smart Grid

Project report submitted to
Visvesvaraya National Institute of Technology, Nagpur
in partial fulfillment of the requirements for the award of the degree

Bachelor of Technology
in
Computer Science and Engineering

by

Karan Sujan (BT19CSE048)   **Kaushal Lodd (BT19CSE052)**

Abhishek Wagh (BT19CSE127)

under the guidance of

Dr. Anshul Agarwal

Department of Computer Science and Engineering
Visvesvaraya National Institute of Technology
Nagpur 440 010 (India)

2023

Integration of Blockchain with Energy Smart Grid

Project report submitted to
Visvesvaraya National Institute of Technology, Nagpur
in partial fulfillment of the requirements for the award of
the degree

Bachelor of Technology
in
Computer Science and Engineering

by

Karan Sujan (BT19CSE048) **Kaushal Lodd (BT19CSE052)**

Abhishek Wagh (BT19CSE127)

under the guidance of

Dr. Anshul Agarwal

Department of Computer Science and Engineering
Visvesvaraya National Institute of Technology
Nagpur 440 010 (India)

2023

Department of Computer Science and Engineering
Visvesvaraya National Institute of Technology, Nagpur

DECLARATION

We, Karan Sujan, Kaushal Lodd, Abhishek Wagh hereby declare that this project work titled “Integration of Blockchain with Energy Smart Grid” is carried out by us in the Department of Computer Science and Engineering of Visvesvaraya National Institute of Technology, Nagpur. The work is original and has not been submitted earlier whole or in part for the award of any degree/diploma at this or any other Institution / University.

Name and signature
Date:

CERTIFICATE

This to certify that the project titled "Integration of Blockchain with Energy Smart Grid", submitted by **Karan Sujan, Kaushal Lodd and Abhishek Wagh** in partial fulfillment of the requirements for the award of the degree of **Bachelor of Technology in Computer Science and Engineering**, VNIT Nagpur. The work is comprehensive, complete and fit for final evaluation.

Dr. Anshul Agarwal
Assistant Professor, Computer
Science and Engineering, VNIT,
Nagpur Dept., VNIT, Nagpur

Head, Department of Computer Science and Engineering
VNIT, Nagpur
Date:

ACKNOWLEDGEMENT

We would like to express our sincere gratitude to our guide, Dr. Anshul Agarwal, for giving us the opportunity to work on this research project, guiding us thoughtfully throughout this project and giving us directions and for providing the necessary resources whenever required by us. We would like to thank all the teaching and non-teaching faculty of the Computer Science and Engineering Department for supporting us and providing necessary facilities at all times. We would like to acknowledge the contribution of those who have constantly supported and encouraged us which helped in the successful completion of our project.

ABSTRACT

Energy is produced at power plants and then transferred to users through a complicated system of transmission and distribution lines under the traditional power grid's centralized arrangement. In order to lessen reliance on fossil fuels and advance sustainability, the idea of a smart grid was proposed. It allows for the integration of renewable energy sources such as solar and wind power into the grid.

However, the current smart grid's centralized system is having trouble incorporating new connections, particularly from modest producers like household rooftop solar panel systems. This is because incorporating new connections can be difficult and expensive because the current infrastructure is built to accommodate large-scale power output from centralized power plants.

We provide a solution to this problem that includes changing the smart grid's topology to a decentralized one. The solution suggested uses a smart contract on the Ethereum blockchain to establish a peer-to-peer energy trading mechanism between suppliers and consumers. Through this system, producers also referred to as prosumers can sell the green energy they generate directly to consumers, eschewing the conventional centralized power grid. This strategy encourages sustainability while allowing for a more effective and economical way to integrate renewable energy technologies into the grid.

This solution encourages the use of renewable energy sources and aids in the shift to a sustainable energy future by making it possible to sell green energy through a reliable P2P network. Additionally, this strategy gives consumers more control over their energy use and gives them the option to choose the source of their energy. Overall, the suggested approach is a big step in the direction of a more sustainable and effective energy system

LIST OF FIGURES

LIST OF TABLES

TABLE OF CONTENTS

1. INTRODUCTION

The conventional centralized energy systems that depend on massive power plants and long-distance transmission lines have encountered significant difficulties in recent decades. High costs, carbon emissions, environmental degradation, single point failures, a lack of stakeholder transparency, and energy crises are some of these challenges. A more sustainable energy system that takes these issues into account has drawn increased attention in recent years.

The idea of a smart grid has been suggested as one solution. A smart grid [1] is an enhanced energy system that enables two-way energy and information exchange by leveraging contemporary technology like sensors, communication networks, and control systems. A smart grid aims to increase the energy system's efficiency, dependability, and sustainability while also optimizing the distribution, management, and integration of renewable energy technology.

But smart grids themselves are fraught with difficulties. The difficulty of gaining access to dispersed and scalable energy supplies on a big scale is one of the main obstacles. This is because integrating small-scale renewable energy technology is not a good fit for the current centralized energy system, which is built to manage large-scale electricity generation from centralized power plants.

The middlemen in charge of energy transactions present another difficulty for smart grids. These middlemen may cause a lack of confidence among parties and raise issues with privacy and security. Customers could be hesitant to divulge their energy usage information to outside companies, for instance, because doing so might jeopardize their privacy.

There is a need for creative methods to deal with these issues. The use of blockchain technology [7], which can offer a decentralized and secure platform for energy transactions, is one possible option. Energy transactions can be carried out without the use of middlemen in a transparent and reliable manner utilizing blockchain technology.

In conclusion, the idea of a smart grid presents a viable option for creating an energy system that is more sustainable. The access to distributed and scalable energy supplies and the middlemen engaged in energy transactions, however, create important obstacles. The use of blockchain technology may be able to address these issues and aid in the development of a more effective, dependable, and sustainable energy infrastructure

2. LITERATURE SURVEY

A consensus of digital data that is copied, shared, and synchronized across various sites, nations, and institutions is referred to as a distributed ledger technology, or DLT. Every member of the network has access to these files, and any modifications made to the ledger are immediately replicated to every member. Data security is aided by cryptographic signatures and keys.

A collection of blocks make up the distributed ledger known as "blockchain technology," which is composed of various records of data or transactional information. Each block in the chain of blocks references the cryptographic hash of the information in the block before it. The chain is continuously extended with newly created blocks at regular intervals, and it is copied across the network's participants. A date, nonce, a hash tree known as a Merkle tree [9], smart contract [10] scripts, and more could also be included in each block. The integrity of the block's content may be verified, thanks to the hash and Merkle tree. The blockchain stores and uses smart contracts, computer programmes that carry out an agreement's terms and conditions. The smart contract can be automatically carried out in accordance with the script when a condition is satisfied or an event occurs. The most widely used blockchain-based smart contract platform is Ethereum [8].

Based on a variety of factors, the blockchain concept can be classified into multiple groups. Permissioned versus permissionless blockchain is one of the categories. Anyone can join the network and add new blocks in a permissionless blockchain, but only authorized nodes are allowed to do so in a permissioned blockchain. Blockchains can also be divided into public and private categories. While private blockchains only allow trusted and predefined users to validate and publish new blocks, and other users can only read the data in the blocks, public blockchains are completely decentralized and permissionless. Blockchain transactions can also be performed on-chain or off-chain. Off-chain transactions take place outside of the blockchain network and are growing in popularity because of their low cost and immediate execution, whereas on-chain transactions are accessible to all users and confirmed by participants. On-chain transactions can identify a user's identity by transaction patterns, but off-chain transactions offer more anonymity and security because information is not made available to the public.

A key element of blockchain technology is the consensus process, which makes sure that new blocks can be added to the network. By enabling a group of validators or miners to agree on whether a block is genuine or not, this system ensures trust in the network. Since anyone can participate in public blockchain consensus without needing the confidence of other network nodes, the consensus mechanisms in public and private blockchains are different. As a result of the smaller number of miners and validators involved in private blockchains, which enables faster network throughput, the consensus processes in private blockchains are comparably easier than those in public blockchains.

The Proof of Work (PoW) [7], Proof of Stake (PoS) [11], Delegated Proof of Stake (DPoS) [13], Leased Proof of Stake (LPoS) [12], Proof of Activity (PoAc) [14], Proof of Burn (PoB) [15], and Proof of Inclusion (PoI) [9] are some of the more well-known consensus procedures. The PoW consensus process, initially used in Bitcoin [9], is a computationally difficult puzzle that must be solved in order to add a new block to the blockchain. In PoS, a variation of PoW, validators rather than miners validate blocks, and a node's likelihood of validating the subsequent block is inversely correlated with its stakes and asset holdings. Only a selected few delegates can create and validate blocks in DPoS, a variation of PoS. With LPoS, nodes can lease their own assets to increase chances that they will act as validators and decrease the chances that a small number of nodes will control the blockchain network.

In contrast to PoB, which enables validators to create new blocks and earn rewards by sacrificing their own assets or coins, PoAc is a consensus mechanism that combines PoW and PoS. Without comparing the full blockchain, PoI entails leveraging the Merkle tree root in the block to verify specific data.

The best consensus method to choose will rely on the particular requirements of the blockchain application and has its own advantages and downsides. Due to PoW's low throughput, high latency, and high energy consumption, for instance, it may be less suitable for applications that demand high throughput.

The objective of this modernization is to lessen reliance on fossil fuel-based generation by integrating and using additional distributed energy resources (DERs) and renewable energy sources. By utilizing independent distributed renewable energy providers, the smart grid connects producers and consumers.

An improved version of the smart grid technology has recently been developed as the EI (Energy Internet) [3] idea. The EI integrates information, energy, and economics to create the next generation of smart grids using internet technology. A fully autonomous and intelligent energy network can be created by facilitating the smooth integration of a variety of clean and renewable energies into the grid and by increasing interactions between the various components of the power grid. Sharing information and energy in a manner like that of online data sharing is the main concept behind the EI.

The centralized topology [2] of the existing smart grid design, which depends on centralized or intermediary companies for energy generation, transportation, and delivery networks, and markets, is a major flaw. The smart grid's components interact and communicate with centralized organizations that are able to monitor, gather, and process data and provide all components with the proper control signals. In order to supply energy to the end consumers through the distribution network, the energy transmission is typically done across a long-distance network. The current design of the smart grid raises some questions about scalability, expandability, heavy computational and communicational burdens, availability attacks, and the inability to control future power systems that will consist of a large number of components due to the penetration of renewable energies and the continuously expanding number of elements.

To include more dynamic, intelligent, and proactive features, the trend in smart grid technology is heading towards a decentralized system. In order to boost the dynamic interactions between all components of smart grid systems, the grid infrastructure is also moving towards a completely automated network with decentralized topologies. The EI's connectivity and accessibility raise the degree of the smart grid system's economical, efficient, and dependable operation.

The modernized electrical infrastructure known as the "smart grid" makes use of information technology to increase the efficiency, dependability, and sustainability of the distribution of electricity. Electric vehicles (EVs), distributed energy resources (DERs) [2], prosumers, and cyber-physical systems (CPSs), which demand high levels of security, privacy, and trust, are just a few of the many components that it incorporates.

The ability to convert the smart grid system from a centralized to a decentralized network is provided by blockchain technology. It eliminates the need for a centralized middleman to keep a safe, transparent, and unchangeable record of transactions.

The smart grid can benefit from the use of blockchain technology in a number of ways, including building trust through consensus mechanisms, maintaining confidentiality, integrity, authentication, authorisation, and non-repudiation, as well as privacy through cryptographic methods. Smart contracts, which are self-executing agreements that can automate transactions without requiring human input or centralized authorization, can also be supported in their deployment.

The blockchain network's decentralized and robust design makes it suited for facilitating the integration of numerous smart grid components, allowing for increased interaction and efficiency between them. Additionally, it offers transparency, auditability, and fault tolerance, which can raise the smart grid system's level of security, privacy, and trust.

3. PROBLEM STATEMENT

Title: Integration of the Blockchain with Energy Smart Grid

Description: Our problem statement is described as follows:

1. Our objective is to develop a system that enables a decentralized energy market, allowing energy producers and consumers to conduct direct business without the assistance of middlemen like utility companies. By putting this system into place, we hope to encourage energy democratization by giving people and communities more control over their energy decisions and a more fair opportunity to participate in the energy market. Using blockchain technology as a basis, this strategy entails building a peer-to-peer energy transaction system before expanding it to include dynamic pricing depending on current supply and demand factors. The system's decentralized structure offers a variety of possible advantages, including improved efficiency, lower costs, and more open energy trades.

2. In conclusion, the system we want to create will have a lot of benefits, like decentralization, transparency, efficiency, adaptability, resilience, and the incorporation of renewable energy. By empowering individuals, encouraging sustainability, and expediting energy transactions, these advantages have the potential to completely transform the energy sector.

3.1 Aim

The goal of this project is to create a software application that will enable direct transactions between energy providers and customers using blockchain technology. The majority of energy transactions currently take place through middlemen like utility providers, which can make things more difficult and expensive. This initiative attempts to foster decentralization [2] in the energy sector, which can result in energy democratization, by doing away with the necessity for intermediaries. As a result, people and communities will be able to actively engage in the energy market and have more control over their energy choices. Overall, the objective is to establish a more just and efficient energy system.

1. Transparency will be achieved by keeping track of all transactions in a transparent, immutable distributed ledger. As a result, energy exchanges can be done in a fair and transparent manner with a transparent and auditable record of all transactions.
2. Automate energy trading operations, such as billing, payments, and settlement, to enable more effective, optimized energy transactions while lowering administrative costs and transaction fees.
3. To lower the risk of fraud, cyberattacks, and data manipulation while enhancing the security of energy transactions and data. This could contribute to building a more durable energy market infrastructure.
4. Allowing the tracking and trading of renewable energy certificates (RECs) or tokens that reflect energy produced from renewable sources will help to facilitate the integration of renewable energy sources, like solar and wind, into the energy market. This may encourage the use of sustainable and clean energy sources.
5. To streamline processes and automate energy transactions, settlements, and grid operations, thereby lowering the need for middlemen. This may result in enhanced transactional accuracy, cost savings, and increased productivity.

3.2 Challenges

The process of creating a solution for our problem statement is fraught with difficulties.

1. Technical Difficulty: It may be difficult to develop a P2P energy trading network utilizing blockchain technology because it is still in its early stages. It demands expertise in blockchain development, smart contract programming, and integration with current energy infrastructure, such as grids and meters.

2. Energy Market Dynamics: Market structures, rules, and participants in the energy sector are well-established. They are also intricate and subject to strict regulation. The market dynamics could change as a result of blockchain-based P2P energy trading, which could lead to regulatory challenges, resistance from current energy

market participants, and ambiguity surrounding pricing, tariffs, and grid management.

3. Privacy and Security: Energy data, such as consumption patterns and transactional data, can be delicate and necessitate rigorous privacy and security measures. It can be challenging to maintain transparency and traceability on a blockchain while protecting data security and privacy. Addressing the potential for fraud, cyberattacks, and other security risks is also required in order to foster confidence among participants in the P2P energy trading ecosystem.

4. Adoption and User Experience: Since blockchain-based P2P energy trade is a novel concept, user adoption may be slow. Creating user-friendly interfaces, educating and rewarding participants, and fostering trust among consumers, prosumers (energy producers/consumers), and other stakeholders are crucial for widespread adoption. The P2P energy trading ecosystem needs an easy-to-use interface to attract and retain users.

5. Grid Integration and Infrastructure Limitations: It can be difficult to integrate P2P energy trading with the current energy networks and infrastructure. In order to prevent any technical problems and guarantee the reliable and effective operation of the energy system, grid management, grid stability, and energy balance must all be carefully taken into account.

6. Economic Viability and Business Models: It can be difficult to create sustainable business models for blockchain-based P2P energy trade. It takes significant thought and collaboration between stakeholders to decide on equitable pricing, revenue sharing, and monetization strategies that yet ensure economic viability for all participants.

3.3 Approach

The first stage of the process for creating a decentralized energy system was to use blockchain technology to build a simple peer-to-peer transaction system. With the help of this system, energy producers and customers would be able to do direct business without the aid of middlemen like utility corporations.

Extending the system to include dynamic pricing is the next stage of development. Dynamic pricing is the process of determining prices for goods or services depending on the supply and demand conditions in the market at the time. This indicates that in the context of energy, various variables, including the availability of energy resources and the degree of customer demand, would affect energy prices.

The objective is to develop a more effective and affordable energy market by integrating dynamic pricing into the energy system. The ability to make educated choices about when and how to use energy based on current prices will provide consumers more control over their energy options. A more balanced and sustainable energy market may result from producers being motivated to modify their output in response to market demand. In general, this strategy seeks to advance energy democratization and build a more open and just energy system.

- **Frontend**: A user-friendly interactive interface for users to interact with the blockchain network and smart contracts to participate in energy trading.
- **Backend**:
    - **User Database**: Responsible for storing and maintaining crucial user information like personal details and crypto wallet address, to participate in energy trading and perform transactions in the blockchain.
    - **Smart Contracts**: Self-executing contracts that automatically enforce the terms and conditions of the energy trade, such as registration, pricing, matching and trading, without the need for intermediaries.
    - **Scripts**: Responsible for smart contract deployment to the blockchain, testing smart contracts, interacting with smart contracts and integrating with blockchain APIs

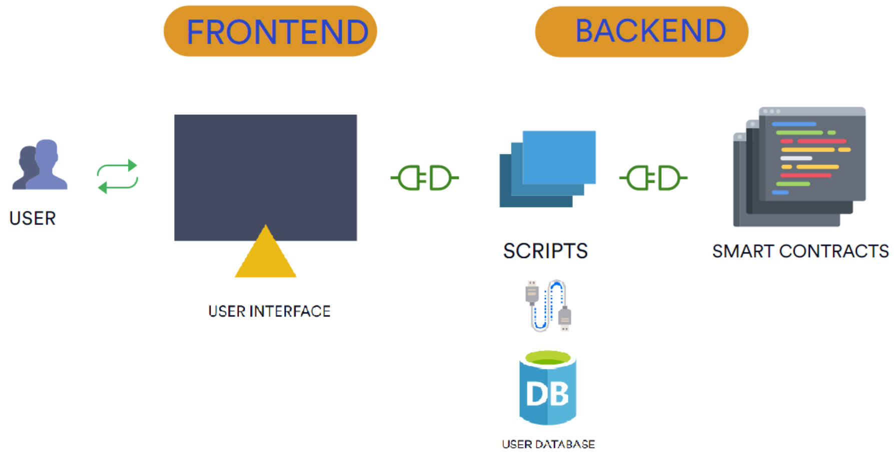


Fig 3.1A. Application architecture

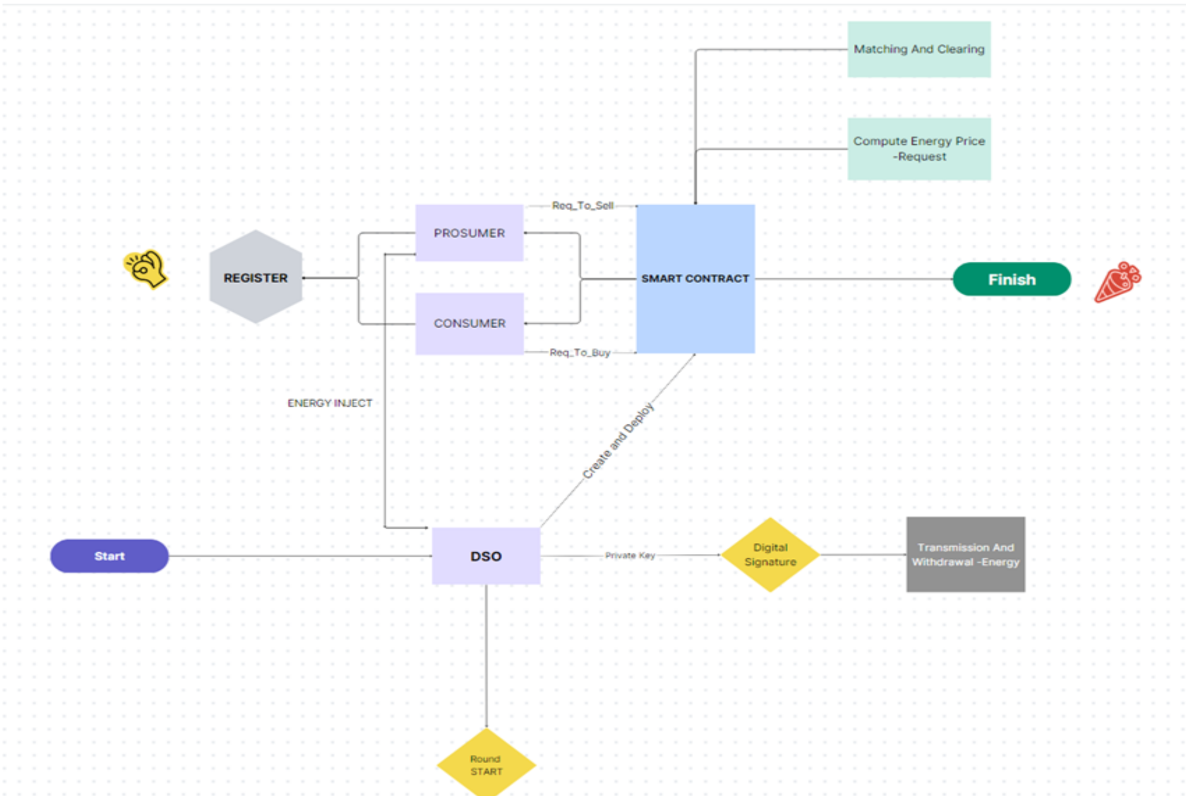


Fig 3.1B. Workflow

Stages of Software Development:

1. **Design and Planning**: Design and Planning: During this stage, the goals, specifications, and features of the application are laid out. We identify the types of smart contracts required, such as those for trading and for private cryptocurrency, as well as the necessary data structures and variables, and we set up the business logic that controls the behavior of energy traders.
2. **Development of Smart Contracts**: During this stage, the application's smart contract code is developed in Solidity. We use Hardhat, a developing environment, to compile, test, and debug the code.
3. **Front-end Development**: In this stage, we use ethers.js to interact with the smart contract code and the Ethereum] blockchain network. We develop the front-end user interface of the application, using HTML, CSS, and JavaScript frameworks such as React.
4. **Testing**: In this stage, we test and debug the smart contract's functionality and behavior using various testing techniques, such as unit testing, integration testing, and functional testing.
5. **Deployment**: Once testing is complete, we deploy the application onto the blockchain network using Alchemy, a node provider. We carefully choose the blockchain network and node we will use to deploy the contract and set appropriate gas fees for the transaction.
6. **Ongoing Maintenance**: After deployment, we continue to monitor the applications performance and usage, and make updates or changes as necessary. This may involve adding new features, fixing bugs, or optimizing performance to ensure a smooth user experience.

4. METHODOLOGY

4.1 The Dynamic Pricing Equation

The primary aim of our dynamic pricing algorithm is to achieve supply-demand equilibrium among prosumers and consumers within a microgrid. For instance, when the overall demand within a trading round exceeds the total supply, the algorithm raises the price of energy units to discourage further demand. The dynamic pricing feature enables our energy trading system to achieve balance, whereby total supply corresponds to total demand within a microgrid. Let's use the symbols ES(t) and ED(t) from equation (1) and (2) to signify total supply and total demand at the beginning of the trading round t.

- $ES(t) = \sum_{i=1}^{np} Si(t), \; (Si(t) \geq 0)$ (1)
- $ED(t) = \sum_{j=1}^{nc} Dj(t) \; (Dj(t) \geq 0)$ (2)

where np and nc are the respective numbers of prosumers and consumers and Si(t) and Dj(t) are the supply of prosumer i and consumer j, respectively. According to Equations (3) and (4), R(t) and D(t) stand for the ratio and the difference between total demand and supply, respectively.

- $R(t) = ED(t)/ES(t)$ (3)
- $D(t) = ED(t) - ES(t)$ (4)

Dynamic real-time pricing has been proposed by Chekired, Khoukhi, and Mouftah [16] utilizing R(t) and D(t) as .

- $p(t) = \{tan^{-1}(e^{D(t)}) + tan^{-1}(R(t))^{10}\} + P_{min}$ (5)

4.2 Proposed Dynamic Pricing Algorithm

Our dynamic pricing algorithm is represented in the figure below.

- $P_{balance}$:- price when total demand is equal to total supply *(R(t)=0)* .
- P_{con} :- to determine the ranges from price($P_{balance}$ - P_{con} .$P_{balance}$+ P_{con}).
- Price *p(t)* : - ranges from $P_{balance}$ - P_{con} to $P_{balance}$+ P_{con}
- $\lim_{R(t) \to 0} tan^{-1}(lnR(t)^{k}) = -\frac{\Pi}{2}$
- $\lim_{R(t) \to \infty} tan^{-1}((ln(R(t))^{k}) = \frac{\Pi}{2}$

The price at time t (p(t)) is set to balance if the total supply and demand are equal, which indicates that R(t) is equal to 1. We can choose the balance price, as well as the minimum and maximum prices, using this strategy. We may ease the burden on producers and consumers by establishing a single price each trading round, enabling them to respond to price fluctuations by altering supply or demand in accordance with their own predetermined norms. The supply/demand function against price can be visualized as a line, curve, or step function to reflect these rules. As demonstrated in Figure 4.1 for k = 3, 5, and 7, the exponent "k" in Equation (6) can be utilized to regulate the pricing curve for the DSO.

In Figure 4.1, Different values of k (3, 5, and 7) are used to illustrate the relationship between p(t) and R(t). The graph demonstrates that p(t) converges to the maximum price of 130 when R(t) is high and is near to the minimum price of 70 when R(t) is extremely small. The price p(t), which represents the P $_{balance}$, is set to 100 when R(t) is equal to 1. In a log scale, the price fluctuates symmetrically against R(t), and the value of k affects how steeply the price curve slopes. As a result, the suggested price model enables DSO to choose $P_{balance}$ and P_{con} with ease.

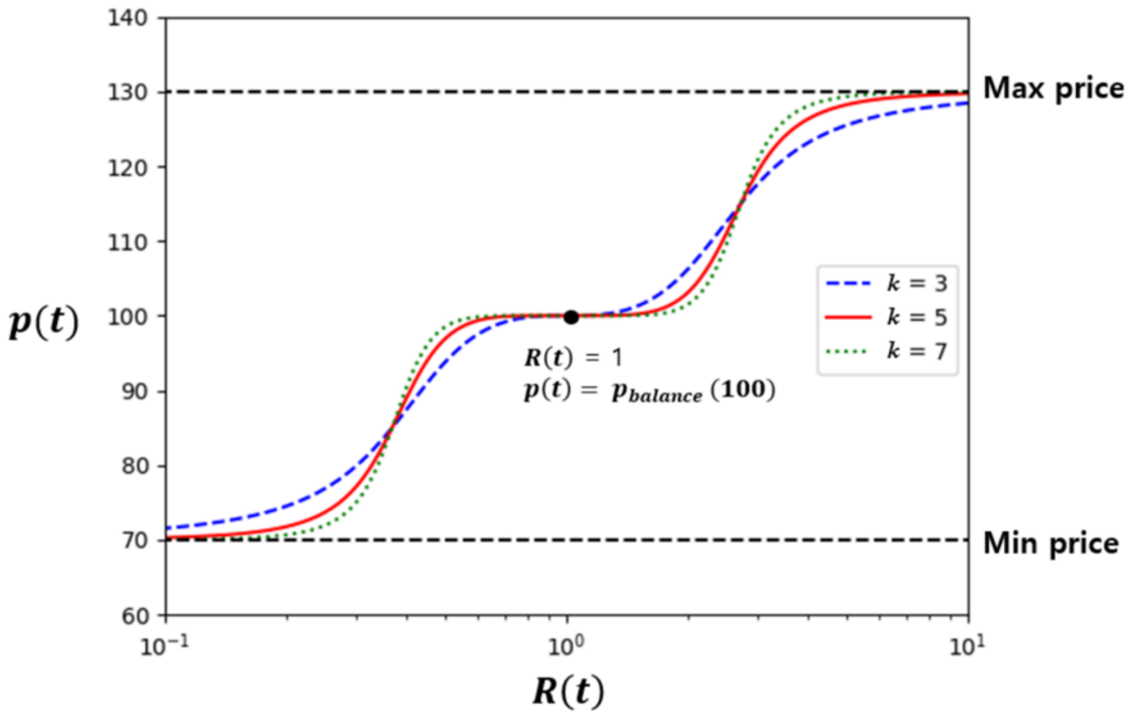


Figure 4.1 Relation between Price and Demand-Supply

4.3 State Diagram Representation and Solidity Program

The study makes the assumption that there are three different types of nodes participating in the energy trading blockchain within a microgrid: prosumers, consumers, and the DSO in order to depict the trading system in a state diagram.Consumers buy energy from prosumers based on their own needs and modify their energy consumption in response to price changes, whereas prosumers adapt their energy supply in response to price changes. Equations (1) through (3) and (6) are used by the smart contract to determine the energy price for each trading period (such as an hour or a day).

The DSO is in charge of energy transmission and the smart contract for energy trading and serves as the blockchain network's operator or manager. It also modifies the exponent "k" in

Equation (6) to regulate the rate at which the system approaches equilibrium in addition to creating, enhancing, and distributing the smart contract.For creating smart contracts on Ethereum, the programming language Solidity is frequently used. The paper uses Solidity's enum type, which lets programmers specify a list of permitted members, to build states. Five stages of energy are defined here for each prosumer or consumer, and they are "register," "injected," "board," "match," and "purchased" (as can be seen on line 8 of Algorithm 1). Figures 4.2 and 4.3 show, respectively, how the situation of prosumers and consumers has changed.

- $p(t) = \frac{2}{\Pi}(P_{con}).tan^{-1}((ln(R(t))^{k} + P_{balance}$ (6)

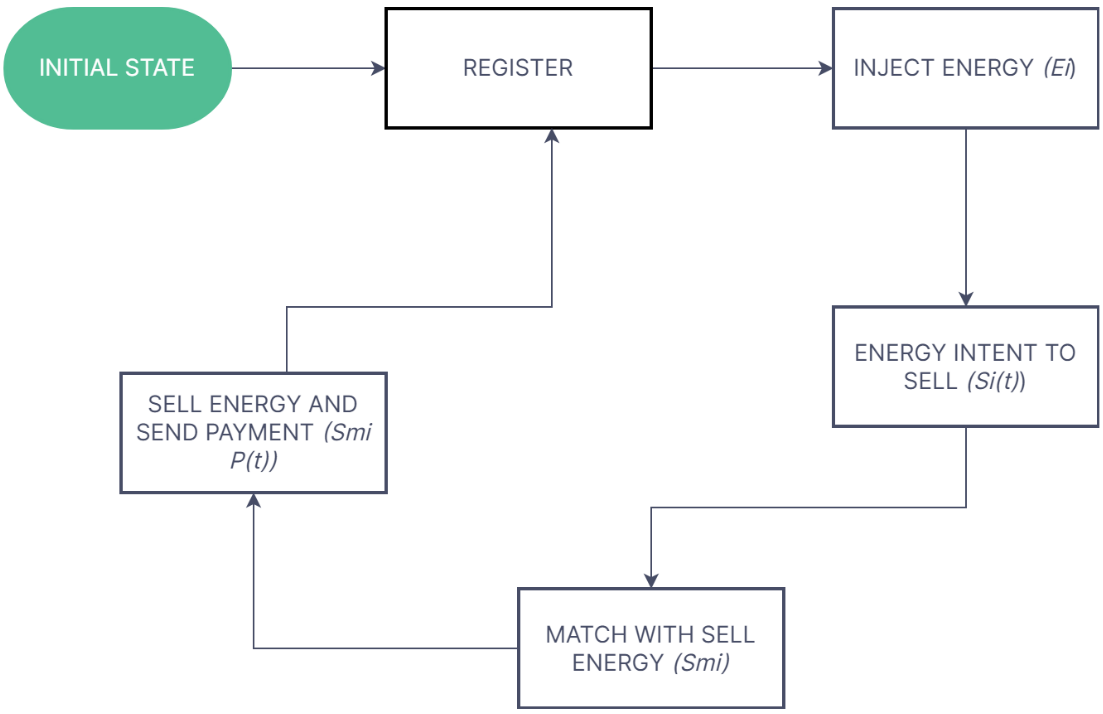


Figure 4.2 State Diagram -Prosumer

Table 4.1 State Diagram -Prosumer

Pi(t)	t=0	t=1	t=2	t=3	t=4
Ii(t)	0	*Ei*	*Ei-Si(t)*	*Ei-Smi*	*Ei-Smi*
Bi(t)	0	0	*Si(t)*	0	0
Mi(t)	0	0	0	*Smi*	0

- *Ii* :- Injected Energy
- *Bi*: - Board Energy
- *Mi* :- Match Energy
- $q = ED(t)/ ES(t)$
- $ES(t) = \sum_{n=1}^{np} Si(t)$ $\quad ED(t) = \sum_{n=1}^{nc} Dj(t)$
- $Smi = Si(t)$ if $(S < D)$, otherwise $q.Si(t)$

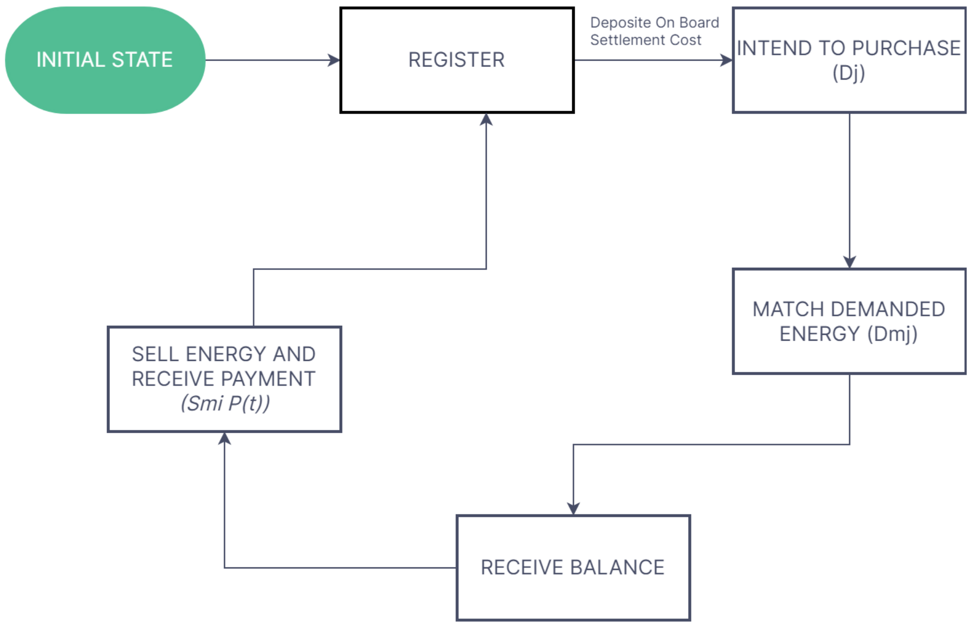


Figure 4.3 State Diagram- Consumer

Table 4.2 State Diagram - Consumer

Cj(t)	t=0	t=1	t=2	t=3	t=4
Bj(t)	0	*Dj(t)*	0	0	0
Mj(t)	0	0	*Dmj*	0	0
Pj(t)	0	0	0	*Pj*	0

- *Bj* :- Board Energy
- *Mj* :- Match Energy
- *Pj* :- Purchased Energy
- $q = ED(t)/ ES(t)$
- $ES(t) = \sum_{n=1}^{np} Si(t)$
- $ED(t) = \sum_{n=1}^{nc} Dj(t)$
- $Dmj = Dj(t)$ if $(S \geq D)$, Otherwise $Dj(t)/q$

Individuals must first register with the private Ethereum[5] blockchain connected to the microgrid in order to engage in energy trading. Sending a transaction that executes the "Register" function, which is in charge of initializing state vectors representing quantities of energy during energy trading, is required in order to accomplish this. These state vectors, which are specified by the structure EnergyOwnership, include data on the energy injected, available for purchase, traded, and paid for by prosumers and consumers. The INJECTENERGY process, which involves sending an encrypted message to the DSO with the amount of injected energy and a signature from the prosumer's private key, is used by prosumers to inject surplus energy into the microgrid. In order to update the status vector for the prosumer with the injected energy, the DSO checks the signature and sends a confirming transaction to the smart contract. Hackers cannot alter the state vector since only the DSO can verify the injection or delivery of energy. By using the "RequestSell" function, which modifies their state vector to reflect the quantity of energy on board available for sale, prosumers can also sell energy.The status vector changes once more when the matching phase starts to represent the amounts of matched energy (sold) and mismatched energy (returned to injected). The prosumer is compensated for the energy that was sold and is permitted to re-inject excess energy.Customers can purchase energy by utilizing the "RequestBuy" function and the "Transfer" function to deposit the desired amount of energy into the smart contract. When the matching phase starts, their state vector changes once more to reflect the amount of energy that has been matched. This happens after their state vector has been updated to reflect the amount of energy on board for purchase. The consumer acquires ownership of the energy that was purchased as well as a reimbursement for the energy that was not purchased.

Solidity-based smart contracts are executed on the Ethereum blockchain. Each node (prosumers, consumers, and the DSO) has its own Ethereum Virtual Machine (EVM), which is used to execute smart contracts. Each node makes use of the hardhat development environment and the Node.js and Ether.js APIs. The tools for editing, compiling, debugging, and deploying smart contracts and decentralized applications (dApps) on an Ethereum blockchain are part of the hardhat development environment.

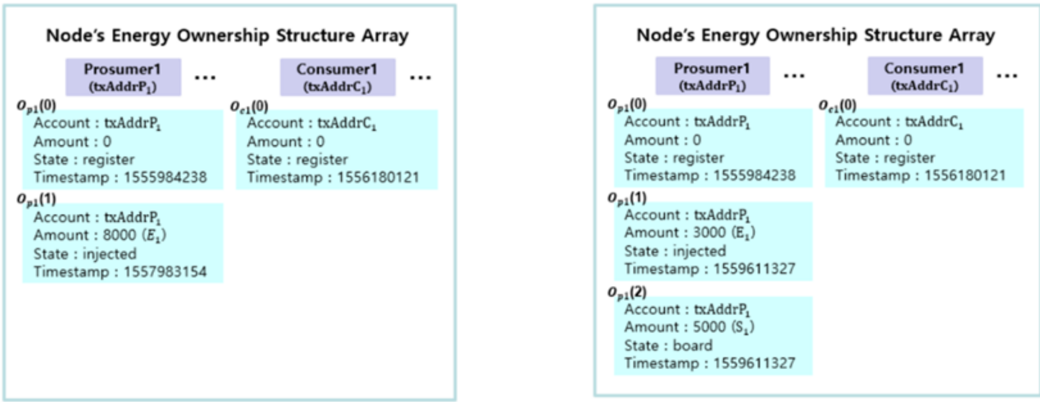
Three groups of users make up the energy trading blockchain network: prosumers, consumers, and a DSO. Prosumers produce their own energy and can choose to sell any surplus to other users of the microgrid. The DSO serves as the network's operator and is in charge of overseeing the distribution of energy among all users. The DSO also manages the overall operation of the blockchain network and develops and updates the smart contracts used for the trade of electricity.A smart contract account is formed once the smart contract code has been broadcast on the blockchain. This account has a special address known as the smcAddr that serves as its identifier. Sending authorized transactions to the smcAddr enables any participating node in the microgrid to communicate with the smart contract. When the corresponding transaction is accepted by the network, the smart contract's functions that specify the guidelines for trading energy are automatically carried out.

4.4 Prevention of Tampering of Transactions

A private key is used to create digital signatures, which are used to secure transactions in a blockchain network. The only person who has access to the private key and can create a digital signature for a transaction is the account owner. The digital signature is contained in the transaction data when a node delivers a transaction to the network. The sender's identity is verified by the use of the signature, which also prevents non-repudiation—the ability for the sender to retract their sending of the transaction.A new block is added to the blockchain when a transaction has been confirmed to be genuine. The contents in the block can't be altered or tampered with since the block contains a mathematical proof, called a cryptographic hash. As a result, the transaction becomes immutable, which means that once it is recorded on the blockchain, it cannot be changed or reversed without the network's approval.

4.5 Prevention of Double Sale of Energy

The system maintains a record of the energy's states through a smart contract that specifies which nodes are qualified to change which states under which circumstances, preventing the sale of energy more than once. Each participant has a state vector that corresponds to its own energy. This state vector is stored inside the smart contract and can only be altered by the qualified node(s) or the smart contract itself. For instance, only the DSO, after physically verifying the amount of energy injected with its smart meter, can update the amount of energy injected by a prosumer using their private key. The prosumer must use their private key to update the smart contract with a "Request_to_Sell" transaction that contains a defined amount of energy that is equal to or less than the injected energy.



(**a**) Prosumer 1 injects the energy ($E_1 = 8000$) after "register".

(**b**) Prosumer 1 puts the supply ($S_1 = 5000$) on board for sale (intends to sell).

Figure 4.4 Node's Energy Ownership Structure Array-Initial state

4.6 Implementation of State Diagram

The state diagrams in Figures 4.2 and 4.3 were implemented in Solidity using the enum data type. We can specify a group of components for the energy state vector using this data type. For each player in our example, we have defined five different energy states: "register," "injected," "board," "match," and "purchased." These five states make up the enum data type, which is specified on line 8 of algorithm 1.

A new prosumer or customer must register themselves on the blockchain network by sending a "Register" transaction to the smart contract. This establishes the participant's state and zeroes the energy values in each state vector, including "Ii(0), Bi(0), Mi(0)" for prosumer i and "Bj(0), Mj(0), Pj(0)" for consumer j. A prosumer i injects extra energy Ei through the INJECTENERGY process, changing its state to Pi(1) and setting Ii(1) to Ei. The prosumer must make a "RequestSell" transaction if they choose to sell the specified quantity Si(t) within the current round of trade. Prosumer i now has the state Pi(2), energy on board for sale Bi(2) is set to Si, and Ii(2) is set to Ei - Si(t). The state of prosumer i changes to Pi(3) when the DSO starts the matching phase by running the "Matching" function. The energy that is really matched for sale, Smi, is set as Mi(3), and it must be less than or equal to Si(t).

Smi's value is calculated Ii(3) is set to Ei - Smi if Si(t) - Smi is greater than zero. Prosumer i receives payment of Smip(t) from the smart contract after the matching is finished and the DSO executes the "Trade" function, changing its state to Pi(4). The prosumer now has two options: inject the remaining energy or offer it for sale. Consumer j must make a "RequestBuy" transaction to express interest in purchasing energy D(t). Once the transaction has been validated, the customer must send a transaction that uses the "Transfer" function to deposit the settlement cost Dj(t)pmax to the smart contract address.

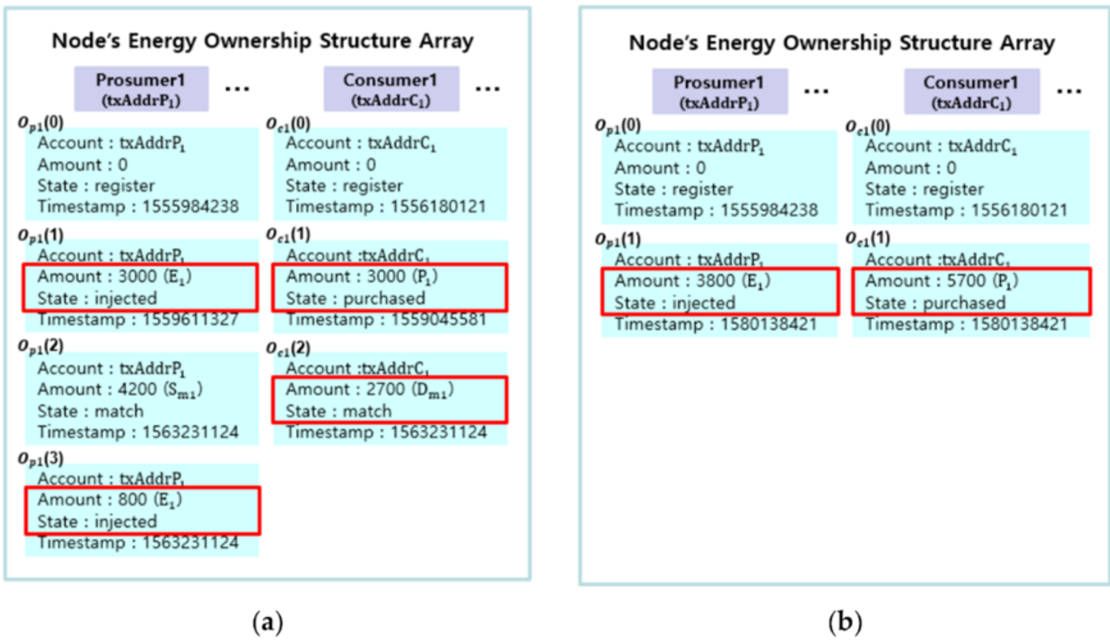
As a result, the consumer's state changes to Cj(1), and Bj(1) is set to Dj. When the "Matching" function is used to start the matching phase, the consumer's state is changed to Cj(2) and the computed energy in the state vector Mj(2) is set to Dmj. The consumer receives Dj(t)Pmax - Dmjp(t) as change from the smart contract, and its state changes to Cj(3) when the DSO performs the "Trade" function and the matching is finished. The amount of energy is then assigned to Dmj and the consumer receives the ownership it has acquired. Finally, the consumer can use this ownership to obtain energy from the DSO, and its state changes to Cj(4).

4.7 Energy Ownership Structure

To depict the energy states of both prosumers and consumers, we choose to utilize arrays. A distinct prosumer or consumer is represented by each array, which has an initial element set to "register" upon registration. In the arrays Opi and Ocj, which include account, amount, status, and timestamp, for prosumer i and consumer j, respectively, the energy ownership structure is specified in lines 9–19. Following the trading procedures, Figure 4.5 shows the changes in energy ownership for prosumer 1 and consumer 1. As seen in Figure 4.5, the smart contract keeps track of these modifications.

All prosumers and consumers must have an element with the "register" state as the first element in their array in order to take part in energy trading.When a smart contract function is carried out through a transaction, subsequent elements are added to, modified, or deleted in accordance with the trading protocols. Figure 4.5a shows how the DSO adds an element to the array with the "injected" state whenever prosumer 1 injects energy. Figure 4.5b demonstrates how an element with the "board" state is added when prosumer 1 requests to sell energy, while the amount of the "injected" element is decreased in proportion. The "board" element is destroyed when the "Matching" function is used, as seen in Figure 4.5, while elements with the "match" state are added to both the prosumer's and consumer's arrays. An element with the "injected" state for unmatched energy is appended .

Finally, using the "Trade" function results in the deletion of elements with the "match" state and the addition of an element with the "purchased" state, as shown in Figure 4.5. Additionally, there are multiple array elements that have elements with the "injected" or



"purchased" states, as seen in Figure 4.5.On the arrays Opi and Ocj, aggregation is carried out.

Figure 4.5 Node's Energy Ownership Structure Array-Intermediate state

4.8 Matching between Prosumers and Consumers

Prosumers and consumers place orders to supply and demand energy, respectively, during aggregation. It is possible that the total balance of the sell and purchase transactions does not match. Equations (7) and (8), where ES(t) and ED(t) stand for the total supply and demand, respectively, and Si(t) and Dj(t) stand for the requested supply and demand by prosumer i and consumer j, are used to calculate this ratio.

- $ES(t) = \sum_{i=1}^{np} Si(t), \quad ED(t) = \sum_{j=1}^{nc} Dj(t) \quad Si(t), Dj(t) \geq 0$ (7)
- $q = ED(t)/ES(t)$ (8)

Np and nc, respectively, stand for the number of prosumers and consumers. The actual amount of energy that consumer j purchased and sold during the matching process is indicated by Smi and Dmj, respectively, as stated in Equations (9) and (10). The values for Smi and Dmj depend on whether ES(t) is larger than or equal to ED(t), and it is vital to remember that the sum of all Smi is equal to the sum of all Dmj.

- $Smi = q.Si(t)$, $if\ ES(t) \geq ED(t)\ or\ Si(t)$, *Otherwise* *(9)*
- *Dmj = Di(t) , if ES(t)* $\geq ED(t)\ or\ Dj(t)/q$, $Otherwise$ *(10)*

4.9 Settlement

In order to facilitate the settlement between sellers (prosumers) and buyers (consumers), we have chosen to use a smart contract as an escrow. The secure execution of agreed-upon procedures without the involvement of a third party is made possible by this blockchain-based technology. Figure 10 depicts the payment procedure between prosumers and consumers in the energy trade.

The DSO sends a transaction to run the "Roundstart" function to start the aggregation phase. All prosumers and consumers then send transactions to perform the "RequestSell" or "RequestBuy" functions to transmit requests to sell or buy energy. Consumer j uses the "Transfer" function, which makes use of ERC-20-compliant tokens, to deposit Dj(t)pmax, the maximum amount of payment (real price p(t) pmax), to the smart contract address smcAddr.

Consumer i receives the payment Smip(t), and consumer j receives the change Dj(t)pmax Dmjp(t), by changing the respective balances on the smart contract when the matching process is over and the DSO performs the "Trade" function. The consumer's array is then supplemented with an element that has the state of "purchased" to represent energy purchases.

5. RESULT AND DISCUSSION

Figure 5.1 illustrates how "k" in Equation (6) affects the convergence to equilibrium (R(t)=1) under the assumption that consumers i and j react to price changes. Figure 5.1 shows p(t) vs. R(t) for three distinct values of "k": (a) k = 3, (b) k = 5, and (c) k = 7 using Equation (6) with

$P_{balance}$ = 150 and P_{con} = 100. The range when p(t) falls within $P_{balance}$ 1% is referred to as the convergence area.

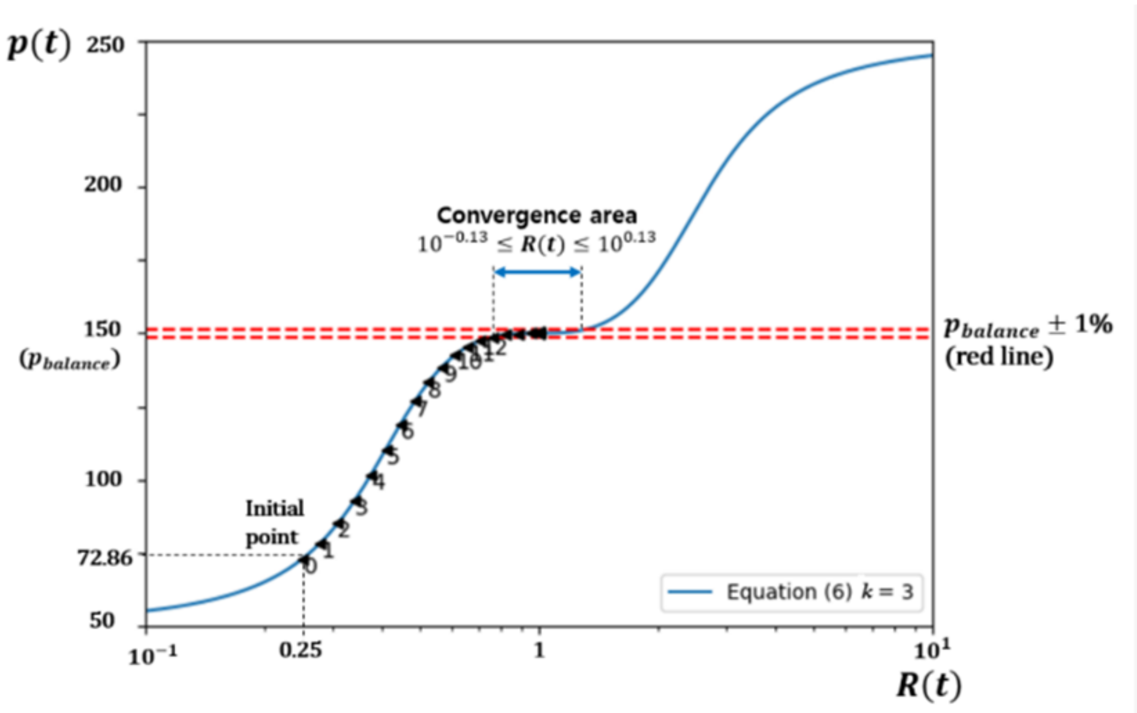


Figure 5.1(a) Price(p(t)) vs.Demand-Supply ratio(R(t)) for k = 3

The convergence speed is the slowest and requires the most rounds (12) to reach the convergence area in Figure 5.1 (a), where the smallest value of "k" (k=3) is employed. However, the convergence region, with a range of $10^{-0.13} \leq$ R(t)$\leq 10^{0.13}$, is the closest to equilibrium. The convergence speed, on the other hand, is the fastest in Figure 5.1(c), where the greatest value of "k" (k=7) is utilized, and it only requires the fewest cycles (8) to reach the convergence area. The convergence region, on the other hand, is the most out of equilibrium, having a range of $10^{-0.25} \leq R(t) \leq 10^{0.25}$.

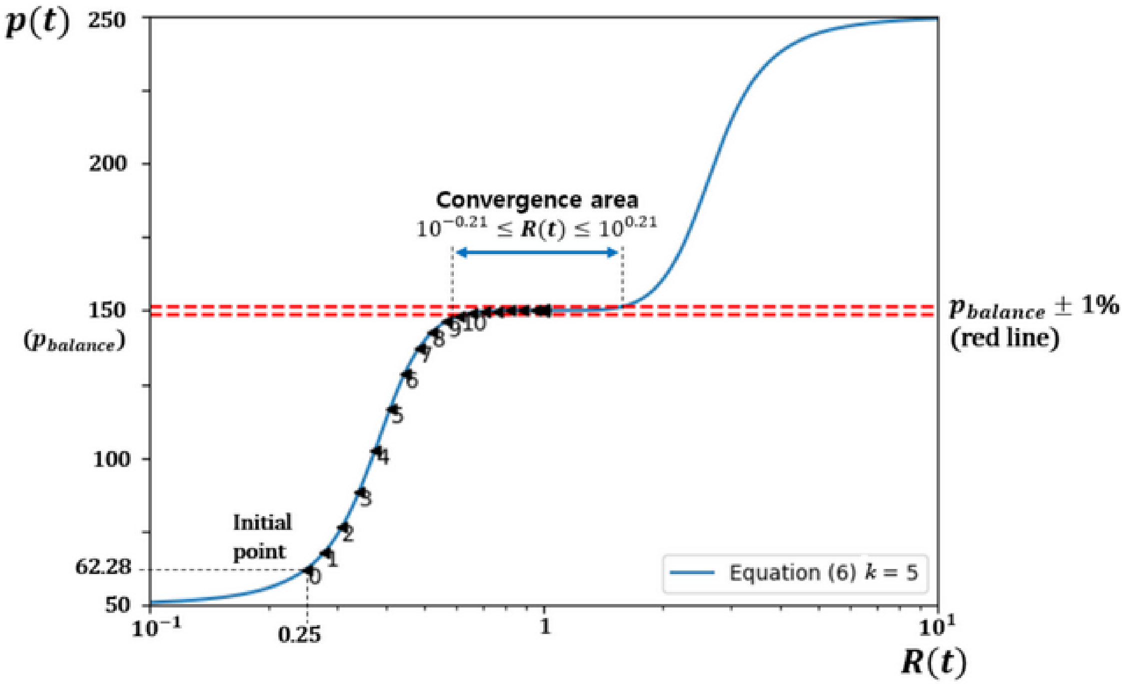


Figure 5.1(a) Price(p(t)) vs.Demand-Supply ratio(R(t)) for k = 5

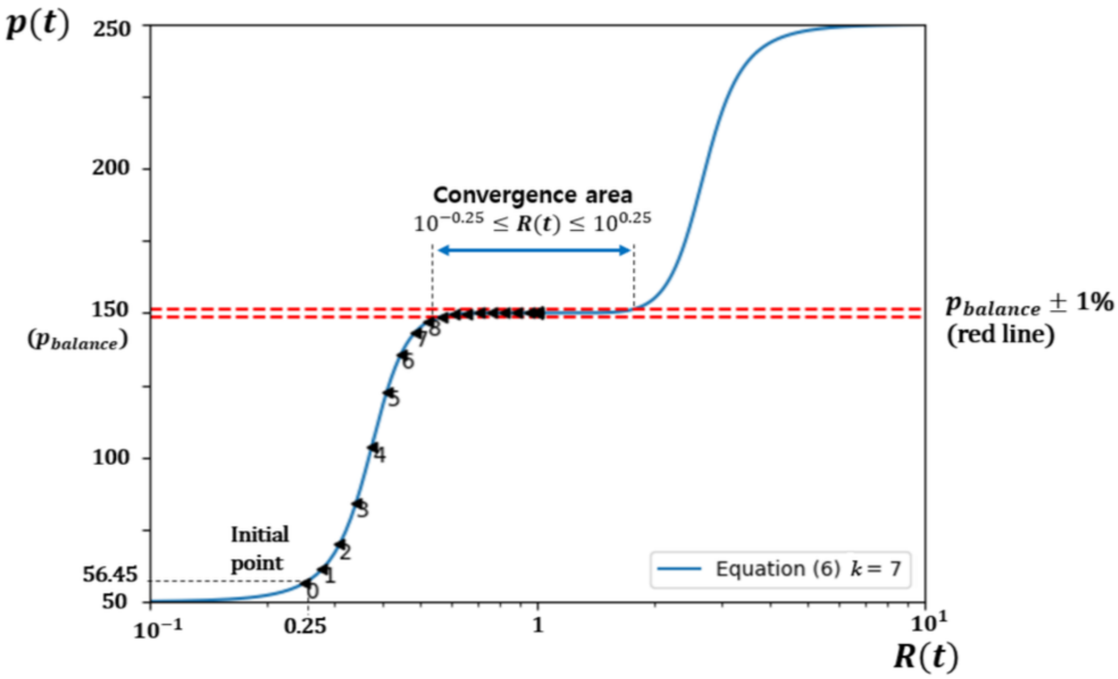


Figure 5.1(a) Price(p(t)) vs.Demand-Supply ratio(R(t)) for k = 7

The convergence pace is moderate in Figure 5.1 (b), where the medium value of "k" (k=5) is utilized, and it takes 10 rounds to reach the convergence area. The region of convergence is located halfway between the extremes shown in Figures 5.1(a) and (c). This indicates that our dynamic pricing approach enables the DSO to modify "k" in order to balance convergence speed and balancing precision (i.e., the narrowness of the convergence area) in accordance with its particular goals.

The Hardhat network was used in our study as the designated Ethereum[5] blockchain for carrying out testing. Private key accounts are available on the Hardhat network, and they were used to replicate the 11 nodes that would take part in the test Ethereum blockchain.

The Alchemy provider is used by each of the virtual prosumers, consumers, and DSO to communicate with the blockchain network. They are all represented in the system as nodes. These nodes use the Alchemy provider to deploy smart contracts, send transactions, and track events from smart contracts.

Our test scenario includes a DSO, 5 prosumers, 5 consumers (each with an account holding 5 ethers), and a smart contract on the Ethereum blockchain.

To perform smart contract activities or get notifications about events from the smart contract on the Ethereum blockchain, each participating node needs a unique smart contract address.

Figure 5.2 shows how Node.js is used by prosumers, customers, and the DSO to access the Ethereum blockchain. The Ethereum Virtual Machine is interacted with via the ethers.js library in Node.js. Each node can execute smart contract functions by sending transactions and monitoring events stated within the functions once a smart contract address has been formed, in accordance with the trading process. Through the use of Node.js, IoT devices can be connected to specific smart contract events, enabling automated energy transmission with the appropriate IoT control and obviating the need for human intervention.

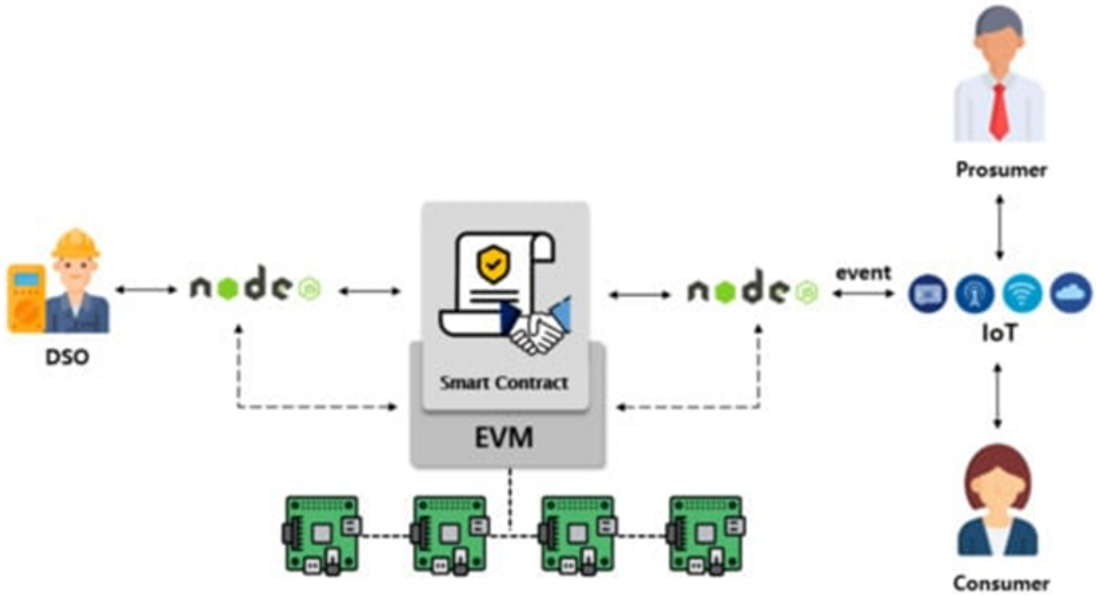


Figure 5.2 showcases our testbed used for conducting experiments on P2P energy trading on the Ethereum blockchain.The functions executed in a smart contract can generate events that are detectable by all participating nodes. These events are typically declared within the smart contract itself.

Figure 5.2 Testbed used for our experiments

Figure 5.2 illustrates the Demo Web page built using ReactJS in our testbed. The page displays the DSO, 5 prosumers and 5 consumers. The DSO has the ability to choose the "k" value, which determines the price curve displayed in the center-right section of the page. The DSO, prosumers and consumers can interact with each other and perform energy trading. The results of matches and trading are displayed on the bottom section of the page.

Further details about the experiment are provided below.

A prosumer injects energy into the Energy Storage System (ESS) of the DSO and sends a message to the DSO using a messaging protocol. The message contains the prosumer's account and the amount of injected energy. Upon receiving the message, the DSO physically checks the injected energy and then sends a transaction to execute the "inject" function. Once the transaction is executed, a new element is appended to the prosumer's ownership array.

Prosumers and consumers send transactions to make requests to sell or purchase energy. In the experiment, there are five prosumers and five consumers who present the supply (sell) and demand (purchase) of energy, as shown in Table 5.1. Upon confirmation of a consumer's valid transaction, the consumer deposits tokens to the smart contract address.

Table 5.1 Energy Injected by the Prosumers into the Energy storage system

Prosumer	Energy injected
1	157
2	980
3	530
4	126
5	700

The amount of tokens deposited is calculated as the consumer's demand multiplied by the maximum price of 50.

Table 5.2 prosumers and consumers requesting to sell and buy energy respectively

Prosumer	Supply (in kWh)	Consumer	Demand (in kWh)
1	48	1	380
2	520	2	115
3	256	3	220
4	126	4	87
5	350	5	168
Total	1300	Total	970

After the aggregation process, the DSO calculates the total supply (request to sell) and total demand (request to purchase), and computes the ratio q, which is the total demand divided by the total supply. For example, if the total supply is 1300 kWh and the total demand is 970 kWh, the ratio q would be 0.74. Since the total supply is greater than the total demand, each prosumer is able to sell q times the total supply as matched supply, while the remaining unmatched supply is injected back to itself. This is illustrated in the state diagram in Figure 4.2, and the result of the matching process is shown in Table 5.2. All matched supply and demand are appended to the ownership array as an element with the state of "match", while all unmatched supplies are appended as an element with the state of "injected" to their respective arrays.

Table 5.3 Outcomes of the matching process for both prosumers and consumers.

Matching Transaction	Prosumer	Consumer	Amount Matched (in kWh)	Unmatched Supply (in kWh)
1	1	1	35	13
2	2	1	345	175
3	2	2	39	136
4	3	2	76	180
5	3	3	113	67
6	4	3	93	33
7	5	3	14	336
8	5	4	87	249
9	5	5	158	91

The trading period employs a uniform price determined by Equation (6) with R(t) set to q = ED(t)/ES(t), assuming k = 3 and p(t) = 31 Tokens/kWh. When the DSO initiates the "Trade" function via a transaction, prosumers receive payment equivalent to p(t) multiplied by the matched supply, while consumers are refunded their deposit minus the matched demand multiplied by p(t). The outcomes are presented in Table 5.3

Table 5.4 presents the Ethereum gas consumption and associated monetary cost in US dollars for the major functions utilized in our implementation, assuming our system operates on the public Ethereum network. The gas consumption estimate serves as an indicator of the given complexity of our system, with the Deployment function executed only once at the beginning of the trading system, while other functions may be performed in each round.

Table 8 Tokens paid to prosumers and the refund provided to consumers.

Prosumer	Paid by Consumer (in NRGToken)	Consumer	Refund (in NRGToken)
1	1085	1	7220
2	11904	2	15435
3	5859	3	12180
4	2883	4	16303
5	8029	5	14102
Total	29760	Total	65240

Table 5.5 presents the Ethereum gas consumption and associated monetary cost in US dollars for the major functions utilized in our implementation, assuming our system operates on the public Ethereum network.The cost of 1 ether is approximately $1,844 as of 4th March 2023, with the average price of unit gas set at 33 Gwei (33 × 10−9 ether) based on Etherscan. The gas consumption estimate serves as an indicator of the given complexity of our system, with the Deployment function executed only once at the beginning of the trading system, while other functions may be performed in each round.

Table 5.5 estimated gas consumption for the various functions

Function	Gas Consumption	Gas Cost ($)
Deployment	4374878	717.17
Register	165920	27.20
Inject	124657	20.43
Round start	72047	11.81
Request sell	23837	3.90
Request buy	26680	4.37
Matching	394793	64.71
Transfer	72477	11.88

Table 5.6 presents the cumulative gas consumption over a specific time period in our implementation, including an estimate of the total gas consumption for np prosumers and nc consumers. It is worth noting that the gas consumption per user remains constant.

Table 5.6 Overall gas consumption for each trading period

Function	Np and Nc consumers and prosumers	5 consumers and 5 prosumers
Deployment	4374878	4374878
Register	(Np + Nc)*165920	1659200
Inject	Np*124657	623285
Round start	72047	72047
Request Sell	Np*23837	119185
Request Buy	Nc*26680	133400
Matching	394793	394793
Transfer	(Np + Nc)*72477	724770
Total	4841718 + Np*386911 + Nc*265077	8101658

In the present implementation, the limited number of states introduced has been exhaustively enumerated for all conceivable scenarios, and thus, we do not anticipate encountering any unforeseen disruptions. However, in the event of technical malfunction, the gas limit serves as a final option whereby we exclusively permit gas limits higher than the permissible transactions to prevent significant harm.

6. CONCLUSION

In order to enable direct transactions between energy customers and producers and do away with middlemen like utility corporations, the goal of this project is to create a standalone application that makes use of blockchain technology. We suggest a P2P energy trading system with a dynamic pricing model that is based on smart contracts. Due to the smart contract's storage on the blockchain, transaction records are secure and immutable, and traditional server-based P2P energy trading systems are free from the significant costs and overheads incurred by hacking or tampering.

The smart contract keeps track of the status of energy transactions to prevent double sales of energy. We also present a dynamic pricing model, which enables the Distribution System Operator (DSO) to balance supply and demand inside a microgrid by balancing convergence speed and exactness.

On a testbed with 5 prosumers, 5 consumers, and the DSO, we create personas to represent participants in energy trading and run virtual simulations of the market. Comparing our approach to a well-known blockchain-based energy trading system with a bidding-style, we can save up to 78% on gas.

6.1 Future scope

- Green Energy Component % check while injection and incentivize injection of green energy
- Deployment of the full Decentralized Application (dApp)
- Making the application more scalable to include more prosumers and consumer

7. REFERENCES

1. G. Dileep, “A survey on smart grid technologies and applications,” Renew. Energy, vol. 146, pp. 2589–2625, Feb. 2020.

2. M. B. Mollah et al., "Blockchain for Future Smart Grid: A Comprehensive Survey," in IEEE Internet of Things Journal, vol. 8, no. 1, pp. 18-43, 1 Jan.1, 2021, doi: 10.1109/JIOT.2020.2993601.

3. K. Wang et al., “A survey on energy Internet: Architecture, approach, and emerging technologies,” IEEE Syst. J., vol. 12, no. 3, pp. 2403–2416, Sep. 2018.

4. Song, Jae Geun, Eung seon Kang, Hyeon Woo Shin, and Ju Wook Jang. 2021. "A Smart Contract-Based P2P Energy Trading System with Dynamic Pricing on Ethereum Blockchain" *Sensors* 21, no. 6: 1985. https://doi.org/10.3390/s21061985

5. M. Mihaylov, S. Jurado, N. Avellana, K. Van Moffaert, I. M. de Abril, and A. Nowé, “NRGcoin: Virtual currency for trading of renewable energy in smart grids,” in Proc. 11th Int. Conf. Eur. Energy Market (EEM14), Krakow, Poland, 2014, pp. 1–6.

6. .Luo, Y.; Itaya, S.; Nakamura, S.; Davis, P. Autonomous Cooperative Energy Trading Between Prosumers for Microgrid Systems. In Proceedings of the 39th Annual IEEE Conference on Local Computer Networks Workshops, Edmonton, AB, Canada, 8–11 September 2014.

7. S. Nakamoto. Bitcoin: A Peer to Peer Electronic Cash System. Accessed: Nov. 2019. [Online]. Available: https://bitcoin.org/ bitcoin.pdf

8. G. Wood, “Ethereum: A secure decentralized generalized transaction ledger,” Ethereum, Zeughausgasse, Switzerland, Yellow Paper, vol. 151, pp. 1–32, 2014.

9. R. C. Merkle, “A digital signature based on a conventional encryption function,” in Proc. Conf. Theory Appl. Cryptograph. Techn., 1987, pp. 369–378.

10. N. Szabo. (1997). Smart Contracts: Building Blocks for Digital Markets. Accessed: Nov. 2019. [Online]

11. Proof of Stack. Accessed: Nov. 2019. [Online]. Available: https://en.bitcoin.it/wiki/Proof_of_Stake

12. Leased Proof-of-Stake (LPoS). Accessed: Nov. 2019. [Online]. Available: https://coinstelegram.com/2018/10/30/what-is-leased-proof- of-stake-lpos/

13. Delegated Proof of Stack (DPoS). Accessed: Nov. 2019. [Online]. Available: https://en.bitcoinwiki.org/wiki/DPoS

14. I. Bentov, C. Lee, A. Mizrahi, and M. Rosenfeld, “Proof of activity: Extending bitcoin’s proof of work via proof of stake,” ACM SIGMETRICS Perform. Eval. Rev., vol. 42, no. 3, pp. 34–37, Dec. 2014.

15. I. Stewart. Proof of Burn. Accessed: Nov. 2019. [Online]. Available: https://en.bitcoin.it/wiki/Proof_of_burn

16. Chekired, D.A.; Khoukhi, L.; Mouftah, H.T. Decentralized cloud-SDN architecture in smart grid: A dynamic pricing model. IEEE Trans. Ind. Inform. 2018, 14, 1220–1231.