P2P Energy Trading with Smart Contracts using Solidity

Submitted in partial fulfilment for the award of degree of Bachelor of Engineering in Electrical Engineering

by

Sneha Gargade - 17

Priti Jadhav - 21

Shubham Pabalkar - 38

Malhar Mhatre - 82

Under the Guidance of

Prof. Garima Gurjar Unhale

(Assistant Professor ELEC Dept)

ATHARVA
COLLEGE OF ENGINEERING

Atharva College of Engineering

Malad-Marve Road, Charkop Naka, Malad (W), Mumbai-95

Department of Electrical Engineering

University of Mumbai

Year 2021-22

CERTIFICATE

This is to certify that Ms. Sneha Gargade, Ms. Priti Jadhav, Mr. Shubham Pabalkar and Mr. Malhar Mhatre of Electrical Department, has submitted the Project Black Book on 'P2P Energy Trading with smart contracts using solidity' and is accepted and examined for the partial fulfilment of Bachelor of Electrical Engineering Degree by the University of Mumbai.

Prof. Garima Gurjar Unhale
Assistant Professor (ELEC. Dept)

External Examiner

Prof. Garima Gurjar Unhale
HOD of ELEC. Dept

Dr. Shrikant Kallurkar
Principal, ACE

DECLARATION

We declare that this written submission represents our ideas in our own words and where others' ideas or words have been included, we have adequately cited and referenced the original sources. We also declare that we have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in our submission. We understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

Date:

(Signature)

ACKNOWLEDGEMENTS

We would like to express sincere gratitude to Principal Dr. Shrikant Kallurkar, Atharva College of Engineering for giving permission to work on the project in the industry and for providing the use of college infrastructure and facilities.

We are thankful to our project guide Prof. Garima Gurjar, HOD in Electrical Engineering Department for taking time from her busy schedule to provide us with a great deal of help, support and guide us to work diligently at every aspect of our project. Her views have always been equitable, striking a perfect balance between encouragement and constructive criticism. Her constructive tips and suggestions helped in our project. We have benefited a lot from her immense knowledge and experience.

We thank the project coordinators Prof. Ksphira Pandey, Prof. Pragya Jain and Prof. Kritika Bansal, Prof. Anu Shukla, lab assistants and staff members of Electrical department who have provided us various facilities and have guided us whenever required. Finally we would like to thank our parents for encouraging and friends for constantly supporting the efforts.

Sneha Kumar Gargade

Priti Raghunath Jadhav

Shubham Shankar Pabalkar

Malhar Makarand Mhatre

ABSTRACT

The proliferation of distributed energy resources (DERs) and, as a result, the emergence of the prosumer stem from the depletion of current energy resources and environmental issues. Renewable energy sources, particularly small scale rooftop solar photovoltaic systems will play a significant role in future energy portfolios of both developed and emerging economies.

Existing centralised energy trading systems rely on a central authority to keep track of transactions, which have numerous problems such as record tampering or record modifications. Also, due to the presence of a third party, the prosumer will not receive the complete revenue.

For energy trading between consumers and prosumers in a community, we propose a Smart contract based Energy trading system which will provide trust, security, and transparency.

This will enable the residents/small businesses to buy electricity at a cheaper rate than grid and sell their excess energy produce to their neighbours.

The goal of this strategy is to assist the community by lowering the cost of living and promote renewable energy generation.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

Abbreviations	Meaning
P2P	Peer to Peer
CDA	Continuous Double Auction
IoT	Internet of Things
IDE	Integrated Development Environment
MIT	Massachusetts Institute of Technology
GUI	Graphical User Interface
API	Application Program Interface
MATLAB	MATrix LABoratory
HTTP	HyperText Transfer Protocol
MQTT	Message Queue Telemetry Transport
C2C	Customer to Customer
B2C	Business to Consumer
Wh	Watt-hour
ETH	Ethereum
INR	Indian Rupee

Chapter 1
Introduction

Overwhelming adoption of Renewable-energy resources by the common masses is required for safeguarding the environment from pollution and combating climate change. The prosumer plays an important role in this process. A prosumer is an individual who consumes as well as produces electricity and sells the excess to either the grid or to a neighbour. In recent times, peer-to-peer (P2P) energy trading is gaining momentum as a new vista of research that is viewed as a possible way for enabling prosumers to sell energy to neighbours. P2P energy trading is necessary for making renewable-energy sources popular and gaining mass market adoption. Blockchain technology is sparking considerable interest among researchers for making P2P energy trading successful. Blockchains provides secure tamper-proof records of transactions that are recorded in distributed ledgers that are immutable.

1.1 Need of P2P network

Energy management systems have been changing from centralised management systems to distributed energy management systems for better energy efficiency. There can be two types of users, users who have surplus energy generations and users who lack energy generations compared to their demands. In this project, to alleviate such imbalance of energy generation between distributed users, a peer to peer (P2P) based energy trading platform is proposed. Specifically, blockchain is one of emerging solutions in which transactions can be made reliably to achieve P2P transactions without any centralised broker intervention. The energy generated from distributed users at the proposed platform can be traded by utilising the characteristics of decentralised application.

1.2 Benefits of using a P2P network

1. The system uses renewable resources that a consumer has access to, which can be availed locally at cheaper cost by other consumers instead of buying it at higher cost from electric companies.
2. Blockchain makes energy trading easier and application deployment will facilitate easy access to monitoring consumptions and sales.
3. The system idea proposed here provides various opportunities to both consumers as well as the prosumers. The prosumer can earn money by selling excess electricity generated on the solar site which increases employment while the consumer uses the solar energy which indirectly benefits nature as it is a clean renewable source of energy.
4. The system being easy to use with one time installation and setup and very less maintenance cost for both prosumers and consumers make it perfectly suitable for all sizes of applications.
5. The failure of one computer won't disrupt the rest of the system and the expense of having and maintaining a server is not required.

1.3 Applications of P2P networks

1. P2P Energy Trading.
2. Power Distribution Systems.
3. Smart Energy Meters.
4. Contactless Payments.

1.4 Blockchain

A blockchain is a list of records, called blocks, cryptographically linked together. Every block contains the cryptographic hash of the previous block, a timestamp, and transaction data (generally represented as a Merkle tree). The timestamp proves that when the block was published the transaction data existed. As each subsequent block contains information about the block previous in it, they form a chain, with each additional block referencing the ones before it. Therefore, blockchains are impervious to modifications of their stored data, because data once recorded in any given block, cannot be altered retroactively without altering all subsequent blocks.

Blockchains are usually managed by a peer-to-peer network as a publicly distributed ledger, where nodes collectively adhere to a protocol to communicate and validate new blocks. Blockchain records are not unalterable but forks are possible. Blockchains are secure by design and exemplify a distributed computing system which has a high Byzantine fault tolerance.

1.5 Benefits of using a Blockchain

1. **Enhanced security** : By creating a record that can't be altered and is encrypted end-to-end, blockchain helps prevent fraud and unauthorised activity. Privacy issues can also be addressed by encrypting personal data and using permissions to manage access. Since information is stored across a network of computers rather than a single server, making it difficult for the data to be compromised by hackers.
2. **Greater transparency** : Without blockchain, each organisation has to keep a separate database. Transactions and data are recorded identically in multiple locations due to the use of the distributed ledger thus enabling all network participants with permissioned access to see the same information at the same time, providing full transparency. All transactions are time and date stamped and are immutably recorded, which enables members to view the entire transaction history essentially eliminating any opportunity for fraud.
3. **Instant traceability** : Blockchain creates an audit trail that documents the provenance of an asset at every step on its journey, making it possible to share data about provenance directly with customers. Any weaknesses in the supply chain can be exposed by the traceability data.
4. **Increased efficiency and speed** : By streamlining traditional paper-heavy processes which are time-consuming, prone to human error, and often require third-party mediation with blockchain, transactions can be completed faster and more efficiently. Storing the documentation on the blockchain along with transaction details, eliminates the need to exchange paper and reconcile multiple ledgers, so clearing and settlement can be done much faster.
5. **Automation** : "Smart contracts" can be used to automate transactions which increase efficiency and further speed the process. Once

pre-specified conditions are met, the next step in the transaction process is automatically triggered. Smart contracts reduce human intervention as well as reliance on third parties for verification of when the terms of a contract have been met.

1.6 Applications of Blockchain

1. Cryptocurrencies
2. Smart contracts
3. Financial services
4. Energy trading
5. Anti-counterfeiting
6. Digital Voting
7. Supply Chain Management
8. Data sharing
9. Copyright and royalty protection

Chapter 2

Literature Review

Table 1: Literature Review

Sr. No.	Name	Description
1	NRGcoin: Virtual currency for trading of renewable energy in smart grids.	Proposed an NRGcoin model, also called a novel decentralised digital currency. The model allows locally generated renewable energy from prosumers to be sold using the digital currency organised by the market paradigm of buyers and sellers of green energy in a smart grid.
2	A novel electricity transaction mode of microgrids based on blockchain and continuous double auction.	Blockchain-based Continuous Double Auction (CDA) is used with various bid combinations that may have distinct initial conditions, but there is an inadequate flexibility to change the bid quantity during the CDA bid process.
3	Consortium blockchain for secure energy trading in the industrial internet of things.	The authors showcased a secure credit-based payment system with reduced wait times for transaction confirmation of the energy chain in permissioned blockchain-based Industrial Internet of Things (IIoTs) which makes electricity trading faster and makes the responses more frequent. The authors also used an optimal pricing mechanism by exploring the idea of stackelberg game theory to optimise bank utility credit-based loans.
4	Blockchain-based Fully Peer-to-Peer Energy Trading Strategies for Residential Energy Systems	In this paper, two strategies for implementing bilateral trading coefficients on a blockchain-based P2P energy market are proposed. Besides providing benefits, such as the indication of preferred trading partners and product differentiation, it is shown that the proposed platform can provide reduced costs for households as well as reduced overall energy imports from the main grid, increasing efficiency and potentially reducing strain on the grid. The results showed that both ST1 and ST2 strategies provide these benefits. However, the paper recommends ST2 since it enables more households to participate in P2P energy trading during more hours of the day, while at the same time considering the technical efficiency in the network.

5	Implementation of Blockchain based P2P Energy Trading Platform	In this paper, they designed and implemented a blockchain based P2P energy trading platform. The authors provided detailed information of how they have constructed the proposed platform, which includes both hardware platform and software platform. Through the demonstration of the process of energy trading via web-page, they reveal that the proposed platform guarantees reliable P2P energy transactions between distant users without centralised broker intervention. This informative contribution towards implementation of blockchain based energy trading provides an incentive for P2P energy trading in future power systems.
6	Joulin: Blockchain-based P2P Energy Trading Using Smart Contracts	Proposed and developed Joulin, a blockchain based P2P energy trading system using smart contracts, that provides an always-available, transparent and easily accessible marketplace. By utilising the Ethereum blockchain for removing the dependency on third party central payment processors, the marketplace enables transactions between peers and utilises an efficient matching service. With a low transaction costs marketplace and flexible architecture, Joulin could be easily adapted to suit different needs.
7	A blockchain-based decentralized energy management in a P2P trading system	Three smart contracts are designed to implement this local energy market. The main smart contract is responsible for the registration of the members and storing all the necessary data related to all transactions, P2P smart contract is responsible to manage the local trading of the market and the P2G smart contract manages the prosumer to grid electricity transactions. The simulations are carried out to check the performance of the proposed system. Simulation results depict that the objectives, cost and PAR reduction, are successfully achieved.
8	Blockchain Based Energy Trading	Blockchain technology promises decentralisation, trust, security and also privacy in the renewable energy trading business. Hyperledger model helps in developing a per-missioned blockchain network where all nodes are authorised. In this paper, a localised Peer-to-Peer energy trading model is proposed which is based on a promising consortium blockchain using Hyperledger fabric.
9	Electricity market design for the prosumer era	The authors concluded that designing electricity markets for the prosumer era could maximise residential and commercial energy efficiency efforts, democratise demand-response and prepare society for ubiquitous distributed clean energy technologies. However, they stated that this can be achieved only if proponents are able to recognize and support prosumer markets differentiated by services, role and function, and anticipate a series of compelling caveats and complexities.

10	A Bidding-Based Peer-to-Peer Energy Transaction Model Considering the Green Energy Preference in Virtual Energy Community	The authors tested a wide range of scenarios and found that the benefit of prosumers occurred in every scenario, and green energy preferences had a significant impact on those benefits.
11	Blockchain-enabled Peer-to-Peer energy trading.	This paper focuses on Peer-to-Peer energy trading and proposes a blockchain scalability solution which is empirically modelled using data collected in a trial case study to improve scalability compared to base layer models without compromising on security and decentralisation.
12	Smarter City: Smart Energy Grid based on Blockchain Technology.	The authors demonstrated that Blockchain will play an important role in facilitating communications, transactions and security among the stakeholders that are involved in a Smart Energy Grid. This paper has dealt with the proposal of an innovative high technology-based architecture in the Smart Environment Pillar of the Smart City evolutionary process. In particular, the improvement of the Quality of Life (QoL) and the enhancement of the Quality of Services (QoS) for the citizens of a Smarter City has been addressed by this paper.
13	A Blockchain-Enabled Decentralized Energy Trading Mechanism for Islanded Networked Microgrids	A two-layer blockchain-based energy trading algorithm for a group of isolated, yet interconnected microgrids was proposed. A preconditioned smart contract-based energy trading in layer one, and a novel two-phase blockchain-based contract settlement protocol is developed in the second layer. It was found that the proposed electricity trading mechanism can efficiently promote energy trading between isolated networked microgrids to assure system reliability when the grid backup is unavailable, and for all peers in the islanded network it offers a price fairness negotiation mechanism.
14	A Blockchain-Based Load Balancing in Decentralized Hybrid P2P Energy Trading Market in Smart Grid	In this paper, three smart contracts for efficient P2P and P2G energy trading in a smart energy market were used. A main smart contract was developed to control all the operations of energy trading in the local energy market. A P2P smart contract was responsible for the whole trading mechanism of the local energy market. P2G smart contract was used when market participants wanted to buy energy from the main grid when they are power deficit and prosumers sell back the surplus energy after fulfilling their energy demand and selling it to the power deficit neighbours.

15	A novel decentralized platform for peer-to-peer energy trading market with blockchain technology.	The solution proposed consists of a market layer and a blockchain layer. They developed a novel decentralised market clearing method called DACO that provides a near-optimal market solution (in terms of maximum social welfare) within a limited number of stages.
16	Smart contracts in energy systems: A systematic review of fundamental approaches and implementations	This paper concludes that there is clearly considerable potential for the use of smart contracts in the energy sector with a wide range of opportunities, from distributed control of energy assets, to automation of billing and market mechanisms, such as market clearing and settlement functions. The ongoing trend towards more transactive, peer-peer and community energy systems, in which energy prosumers are increasingly empowered to take control of their own energy supply can be a key driver for smart contract adoption.

Chapter 3
Concept and Methodology

3.1 Concept

1. Initially we considered a virtual power monitoring system in the power circuit which generated random values for energy for the client.
2. Then for every cycle we generate random values and log those values as the current energy generation/consumption.
3. After some predefined period of time we calculate the average power usage for that period and multiply with time and the rate.
4. The prosumers pool their generation capacity together forming a generation pool.
5. The consumers pool their consumption capacity together forming a consumption pool.
6. The distribution of energy is done in the ratio of the energy requested/supplied.
7. Finally, after each period, the consumers request energy and the calculated tokens transferred from the customer to the prosumer's address before starting a new period.
8. This process repeats in a loop for multiple customers and prosumers forming a smart grid.
9. Cloud Storage is used to store and transfer data for consumption and cost incurred for each customer.
10. An android app is built using kodular in order to remotely monitor current consumption and cost incurred.

3.2 Setting up the blockchain

A local private distributed ledger on an Ethereum based blockchain is set up using Ganache which by default creates a blockchain with ten accounts already configured, with 100 ethers each. Then, running a smart contract on this blockchain requires first compiling the smart contract and then uploading the bytecode to the Ethereum Virtual Machine through one account.

The steps to set up and deploy a smart contract are as follows:

1. Configuration of a local blockchain with nodes (virtual machines) and accounts, using Ganache.
2. Develop a smart contract in a given language (Solidity).
3. Smart contract code has to be compiled using the language compiler.
4. Deploy the compiled code (byte code) to the blockchain using either Python or Javascript Web3 libraries.
5. Interact with the contract (and the blockchain) through python or javascript commands that are sent to the address of the smart contract via a node of the local blockchain.

3.3 Pseudo code for sending data to the smart contracts and thingspeak:

- ➢ Create a web3 object by linking it to ganache blockchain running locally
- ➢ Get accounts on the blockchain
- ➢ Get Real time ETH to INR data from Alpha vantage
- ➢ Get Transaction hash of the contract creation
- ➢ Get Contract ABI
- ➢ Create a contract instance using contract address and contract ABI
- ➢ Initiate the contract with some ethereum from the admin account
- ➢ Start Loop for n cycles:
    - ○ Generate random values for energy production/consumption for all prosumers and consumers
    - ○ Find the total energy production and total energy consumption
    - ○ Get the rate using the required function
    - ○ Send Data to the smart contract
    - ○ Collect data from the smart contract and blockchain
    - ○ Send data to thingspeak
- ➢ Withdraw the remaining balance from the smart contract to the admin account
- ➢ Combine the data collected to a single list
- ➢ Write the created list to a .csv file

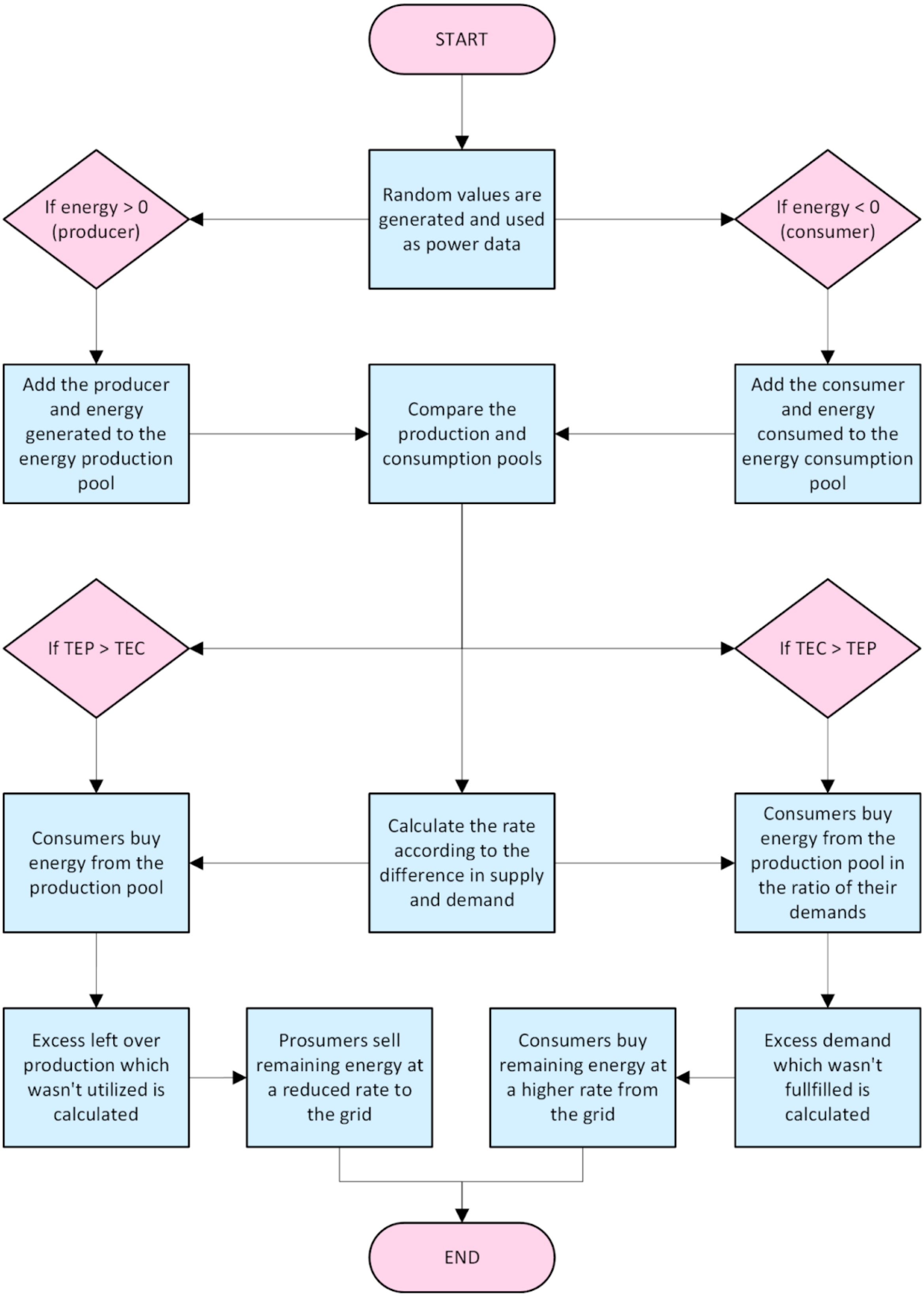

Figure 1: Algorithm Flowchart

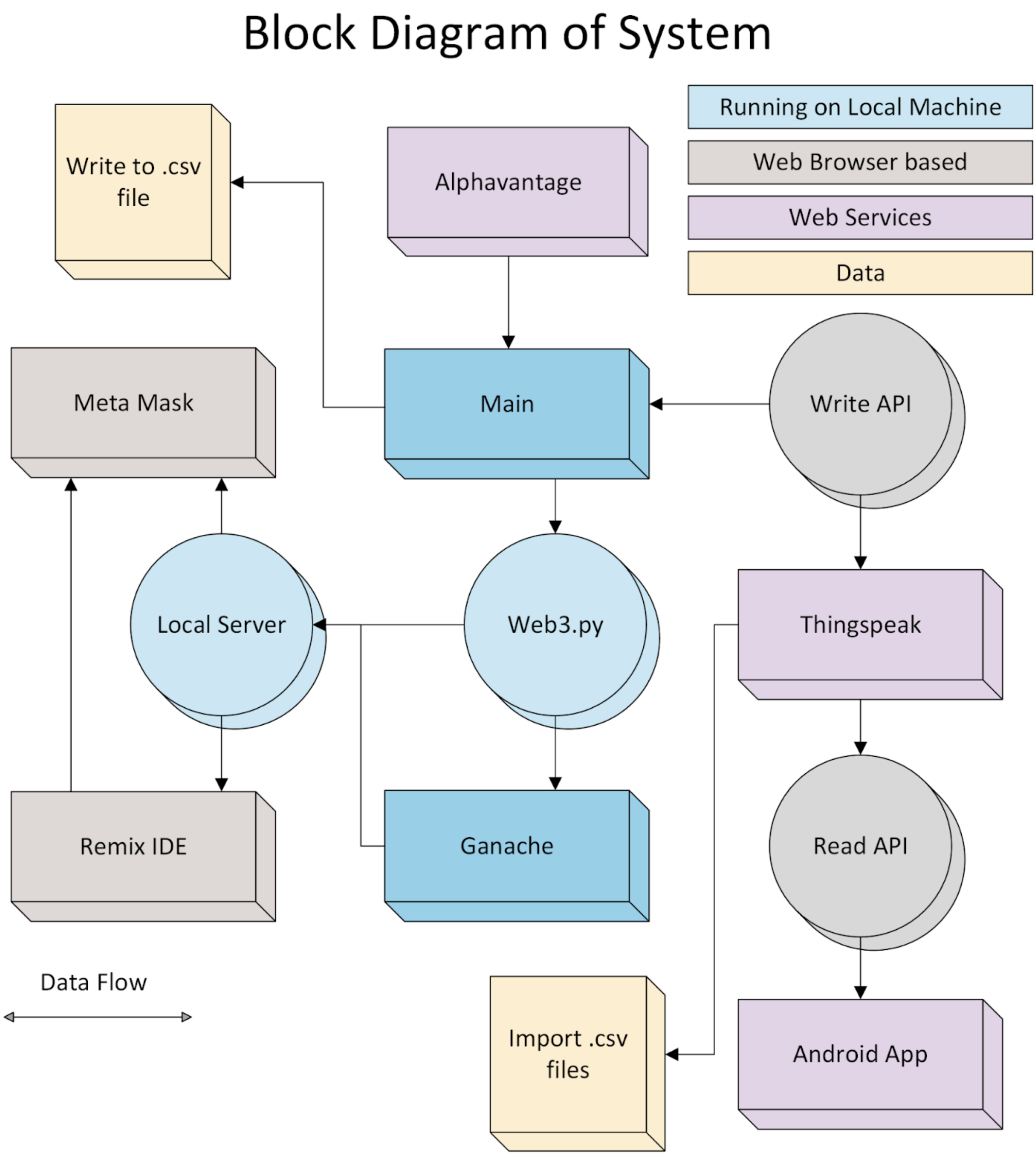

Figure 2: Block diagram of the system

Chapter 4
Software Used

4.1 Ganache

Ganache is a tool that allows users to create a local Ethereum DLT with 10 accounts in which a smart contract can be deployed.

Ganache is part of the Truffle Suite ecosystem which is a development environment, asset pipeline, and testing framework using the Ethereum Virtual Machine.

Figure 3: Ganache Logo

Ganache is a high-end development tool which is used to run local blockchain for both Ethereum and Corda dApp development and it is helpful in all parts of the development process. The local chain allows the development, deployment and testing of the projects and smart contracts in a deterministic and safe environment.

4.2 Remix Ethereum IDE

Remix IDE (Integrated Development Environment) which is a popular browser-based IDE for developing, compiling and deploying solidity smart contracts.

Figure 4: Remix IDE Logo

4.3 Metamask

Metamask is a browser extension which is used to manage Ethereum wallets and deploy Distributed Applications.

It is a software cryptocurrency wallet which users can access through a browser extension or a mobile app and be used to interact with the Ethereum blockchain and decentralised applications.

Figure 5: Metamask Logo

MetaMask allows users to store and manage account keys, send and receive Ethereum-based cryptocurrencies and tokens, broadcast transactions, and securely connect to decentralised applications through a compatible web browser or the mobile app's built-in browser. Developers achieve a connection between Metamask and their decentralised applications by using a JavaScript plugin such as Web3js or Ethers to define interactions between Metamask and Smart Contracts.

4.4 Spyder

Figure 6: Spyder

Spyder is an open-source cross-platform integrated development environment (IDE) released under the MIT licence and used for scientific programming in the Python language. Many prominent packages in the scientific Python stack, including SciPy, NumPy, Matplotlib, IPython, pandas, SymPy and Cython, as well as other open-source software are integrated with spyder.

First- and third-party plugins can be used for spyder with included support for interactive tools for data inspection and embeds Python-specific code quality assurance and introspection instruments, such as Pyflakes, Pylint and Rope.

4.5 Kodular

Kodular is an open-source online suite for mobile app development. It has an innovative component and block design which provides a free drag-and-drop Android app creator without coding, based on MIT App Inventor.

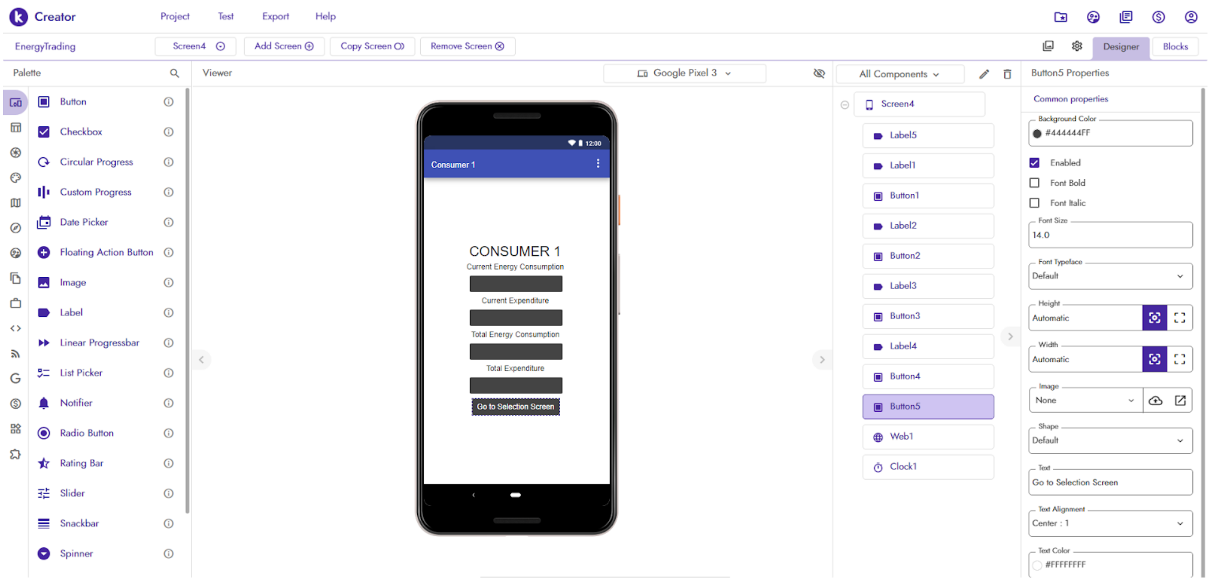

Figure 7: Kodular Designer screen

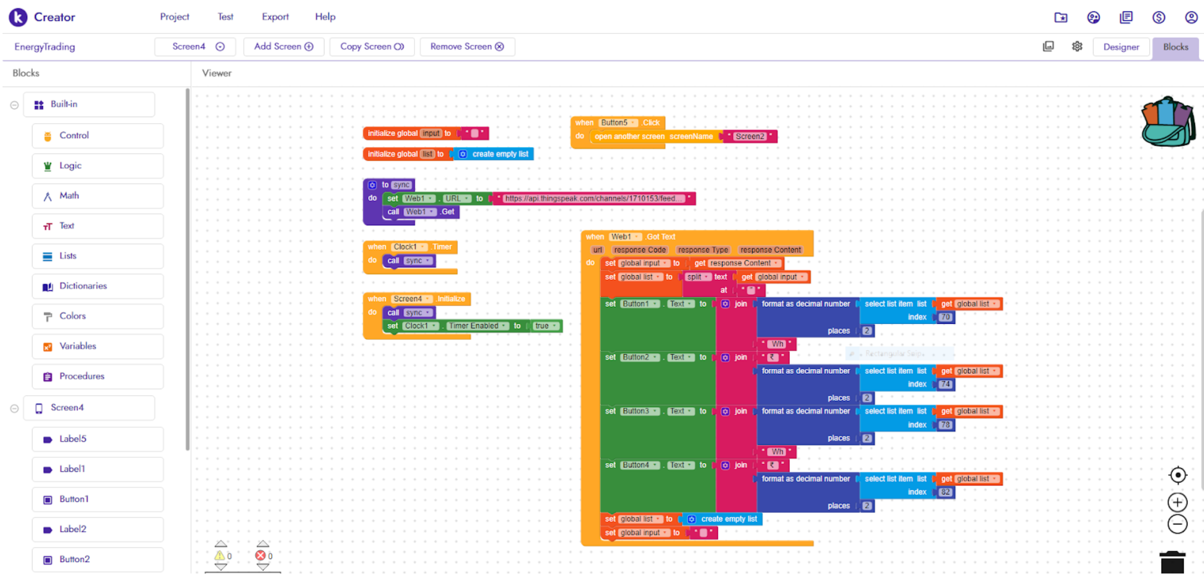

Figure 8: Kodular Blocks screen

4.6 Thingspeak

ThingSpeak is an open-source Internet of Things (IoT) application and API enabling the creation of location tracking applications, sensor logging applications and a social network of things with status updates. It stores and retrieve data from things using the HTTP and MQTT protocol over the Internet or via a Local Area Network ThingSpeak and it has integrated support from the numerical computing software MATLAB from MathWorks, allowing users to analyse and visualise uploaded data using Matlab.

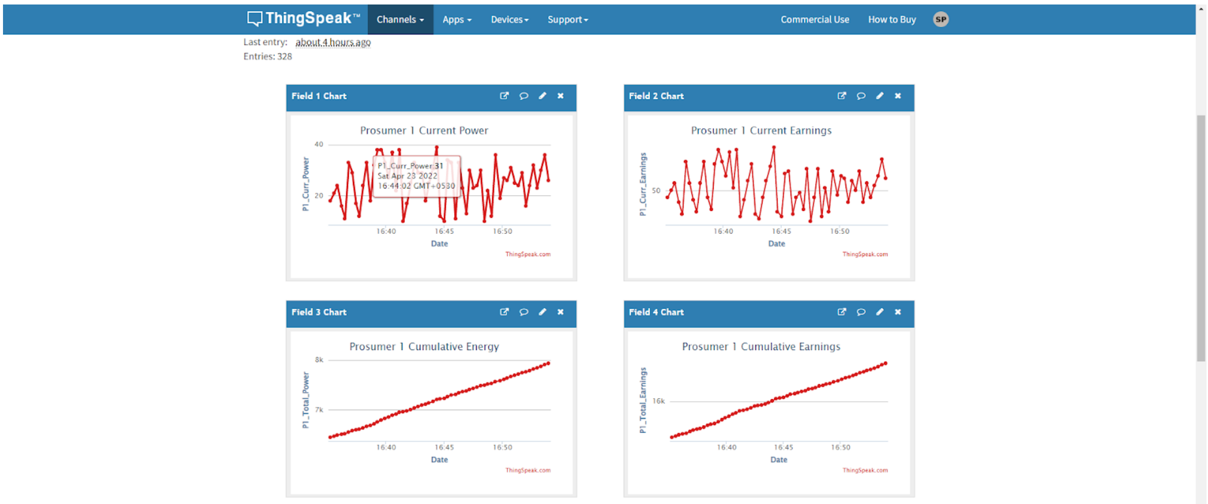

Figure 9: Thingspeak private channel view

4.7 Python Libraries

Web3.py

Web3.py is a Python library for interacting with Ethereum which is usually found in decentralised apps (dapps) for sending transactions, interacting with smart contracts, reading block data, and a variety of other use cases.

The original API was derived from the Web3.js Javascript API, but has since evolved to suit the needs and creature comforts of Python developers.

Random

Almost all module functions in the random module depend on the basic function random(), which generates a random float uniformly in the semi-open range [0.0, 1.0). The Mersenne Twister which is one of the most extensively tested random number generators in existence. is used as the core generator. It produces 53-bit precision floats with a period of 2^{19937}-1 with the underlying implementation in C is both fast and thread safe.

It is used for generating random values to be used as production and consumption data

Time

This module provides various time-related functions. Python time module allows to work with time in Python which allows functionality like getting the current time, pausing the Program from executing, etc.

It is used for setting the cycle time as required.

Requests

The requests module allows the user to send HTTP requests using Python which returns a Response Object with all the response data (content, encoding, status, etc).

It is used for getting the real time ETH to INR conversion rate as well as sending data to thingspeak.

Math

The math module allows the user to perform mathematical tasks on numbers and provides access to the mathematical functions defined by the C standard.

It is used in getting the rate for prosumer to consumer energy trade.

CSV

The CSV (Comma Separated Values) format is the most common import and export format for spreadsheets and databases.The csv module implements classes to read and write tabular data in CSV format.

It is used to write the data generated to a file.

Chapter 5

Simulation

The Main python program was run for 50 cycles with a cycle time period of 30 seconds with the blockchain running on Ganache. The Smart contract was compiled with remix and it was deployed on the local blockchain with the help of metamask. The data was sent to the thingspeak cloud and from the cloud it was sent to the android app for monitoring purposes.

The ETH to INR rate at the time of performance was
1 ETH = 211537.49634700 INR

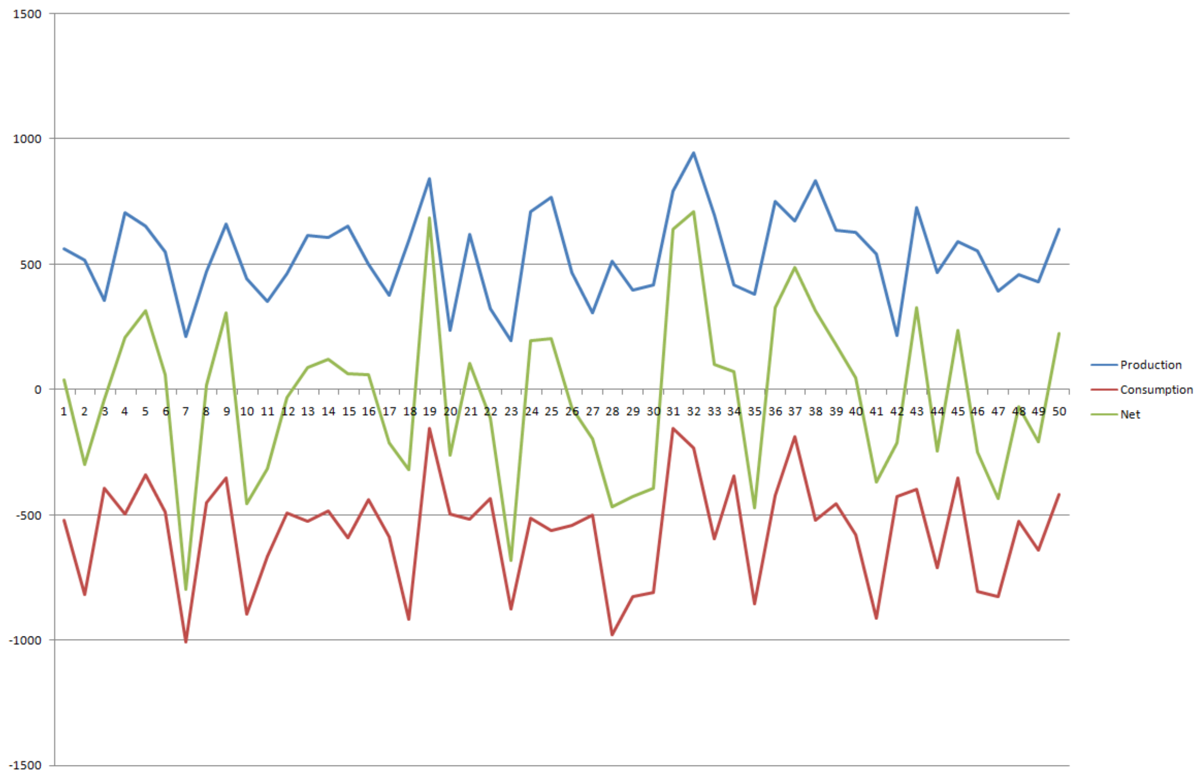

Figure 10: Total Production by producers and Total Consumption by consumers

5.1 Calculating the rate for Prosumer-Consumer Transactions:

A function whose graphical representation is the orange line in figure 12 which is based on hyperbolic tangent is used for determining the rate. It is modified to suit the average difference in production and consumption. This ensures that if the demand is higher than the supply, the difference will be on the positive x-axis, thus the rate will be higher and vice versa for when demand is lower than supply. When supply matches the demand, the rate will be 50 gwei.

Figure 11: The function for determining the rate per Wh is modelled after the Orange Curve

The dotted lines in the figure indicate the rate for grid transactions. The grid rate is considered fixed for the purposes of this project. The dotted line at y=40 is the rate at which the grid buys electricity from prosumers and the dotted line at y=60 is the rate at which the grid sells power to the consumers.

The Prosumer-consumer rate never exceeds beyond the range set by the dotted lines and approaches it asymptotically. Thus even if the difference is high, the rate will never increase or decrease beyond the rate set by the grid. This ensures that the participants never lose money by participating in P2P energy trading.

The orange function approaches the grid rate rapidly so the transaction rate is effectively at the grid rate at ±600 difference. Another function based on the inverse tangent (purple line) can be used to vary the rate slower than the hyperbolic tangent function.

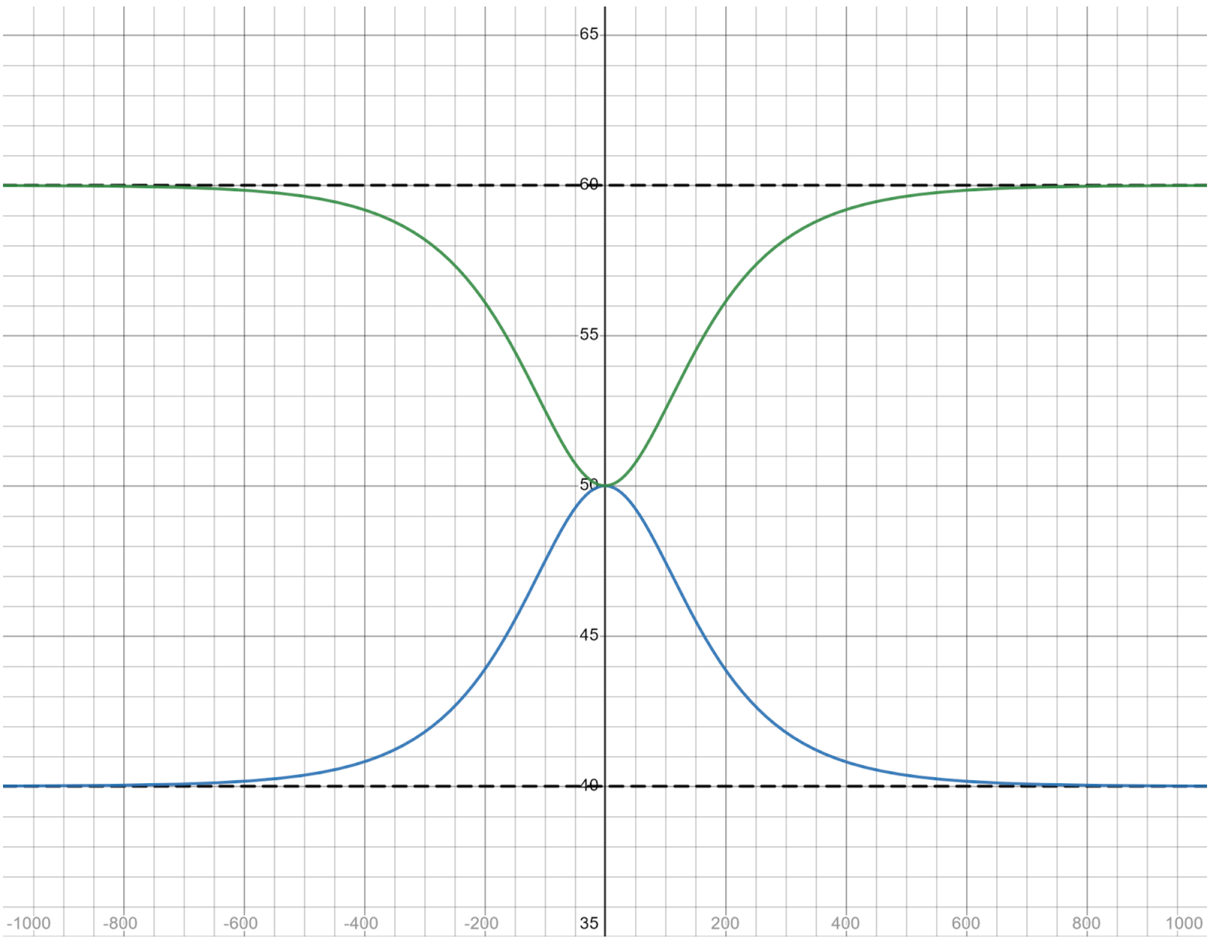

Figure 12: Other functions

Another approach to designing the functions is to incentivize participants to match the supply/demand. The green function shown in figure 13 encourages consumers to match the production, since a mismatch will result in higher rates for them. The blue function encourages prosumers to match the demand, as an imbalance will result in lower rate. Care must be taken to ensure the other party does not benefit from intentionally creating an imbalance.

From the production (Blue line figure 11) and Consumption (Red line figure 11) data, the difference (Green line figure 11) is calculated. The difference is used as an input to the orange function (Figure 12) and the output of this function is plotted in figure 14.

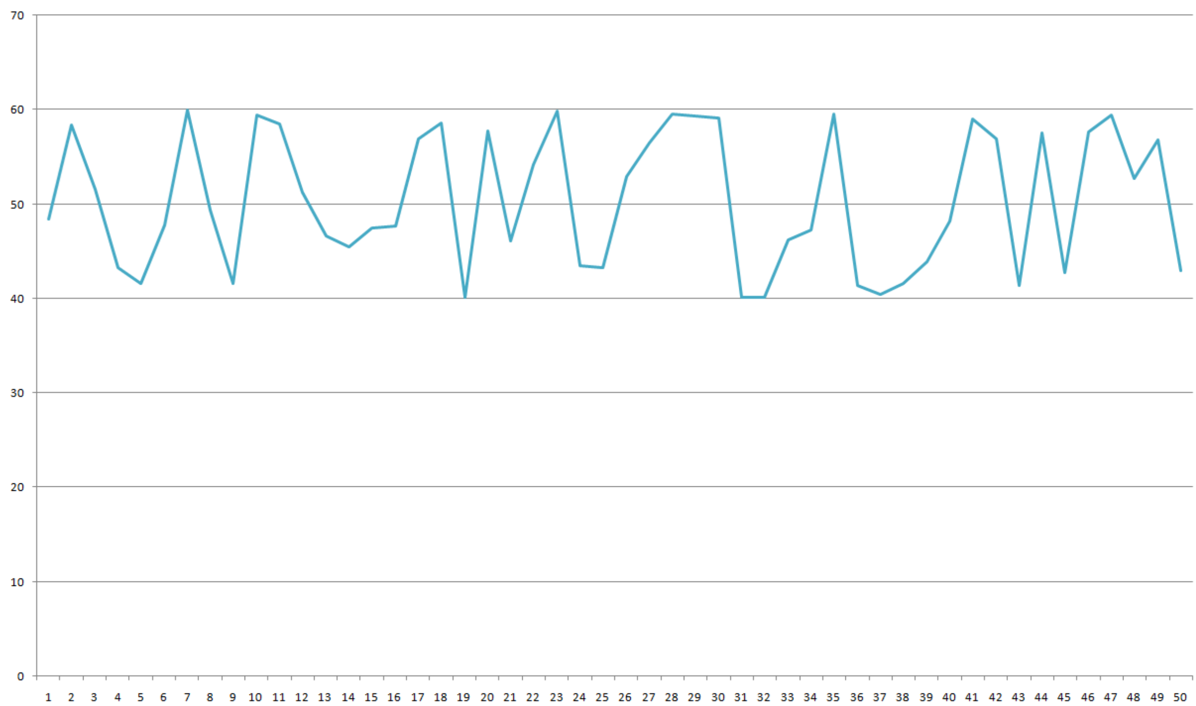

Figure 13: Prosumer Consumer transaction rate which is the output from the function

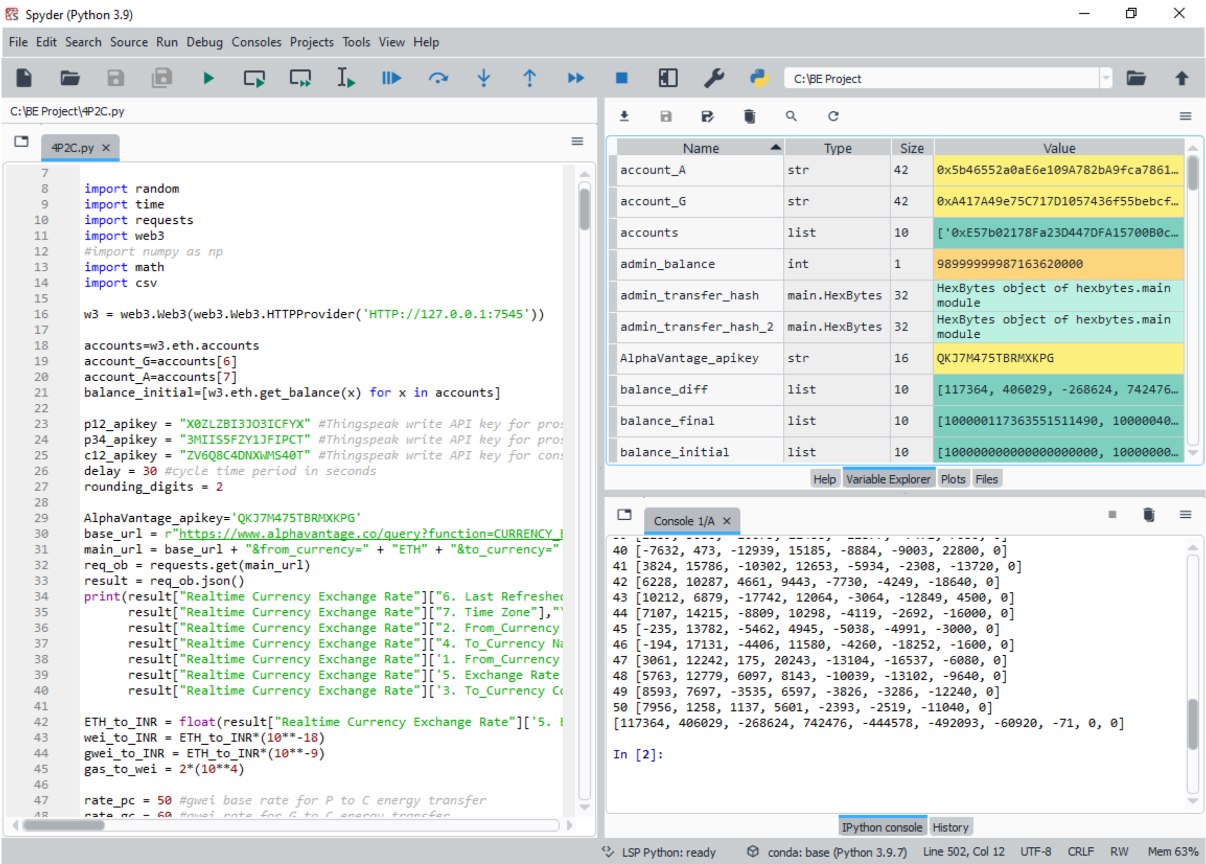

Figure 14: Spyder window after 50 cycles were completed

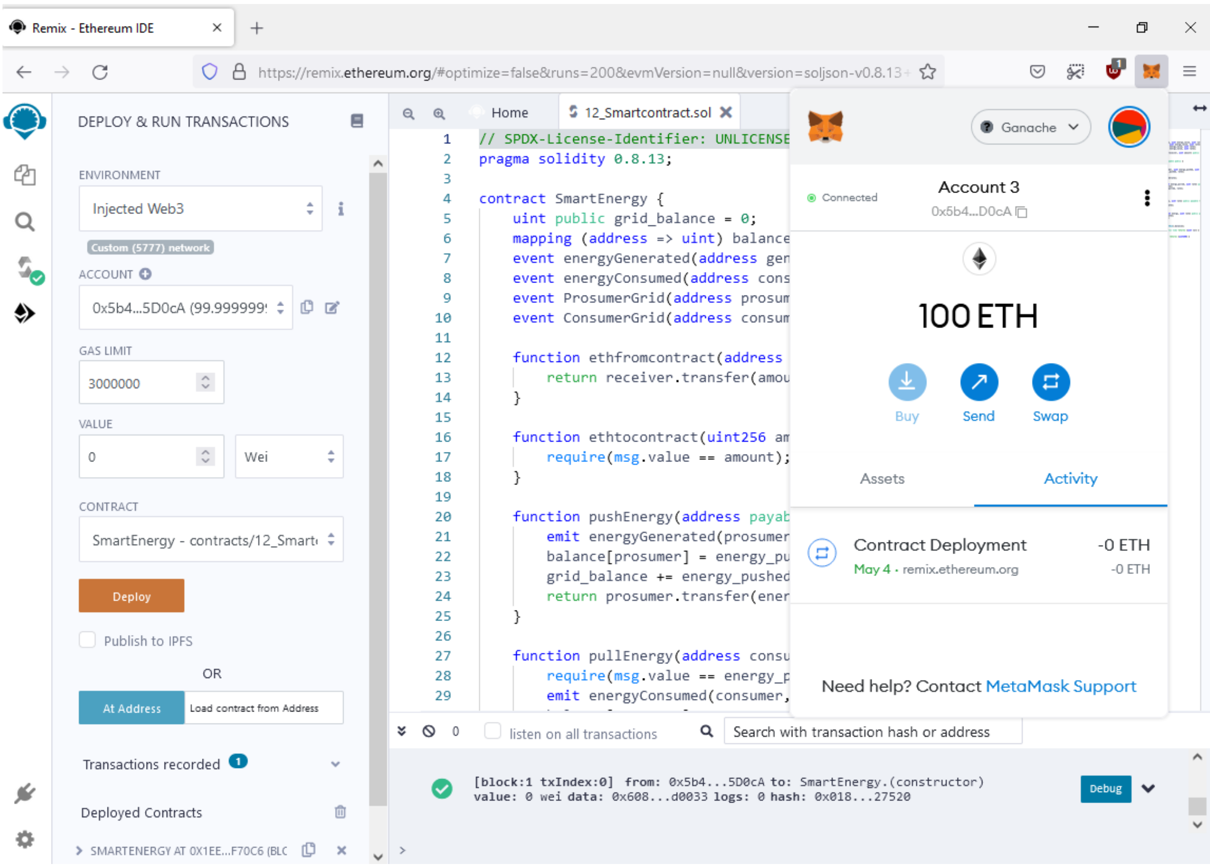

Figure 15: Remix IDE with Metamask showing the contract deployment

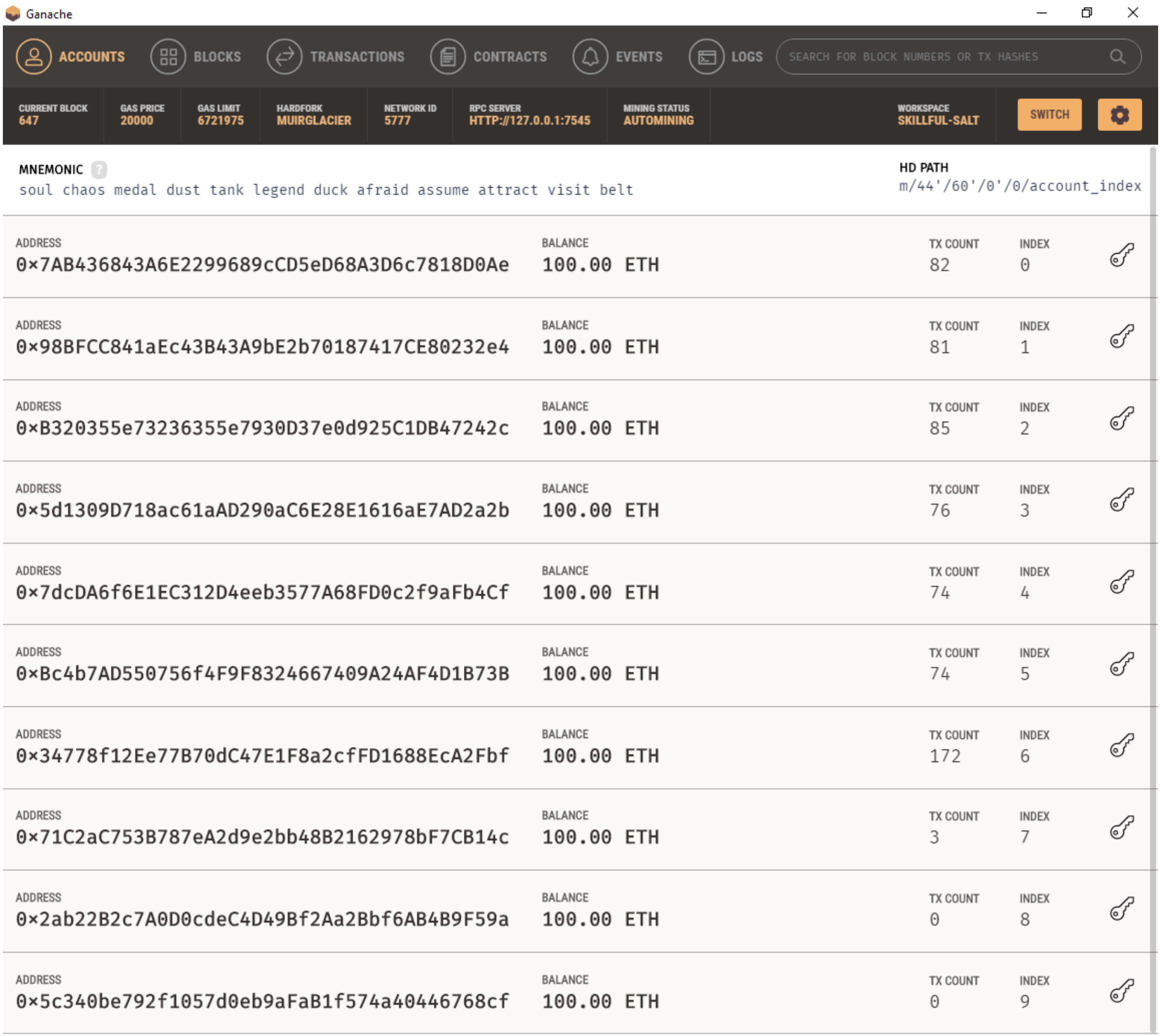

Figure 16: Ganache accounts showing the transaction count after 50 cycles

Table 2: Accounts and their users

Account Index	User
0	Prosumer 1
1	Prosumer 2
2	Prosumer 3
3	Prosumer 4
4	Consumer 1
5	Consumer 2
6	Grid
7	Admin

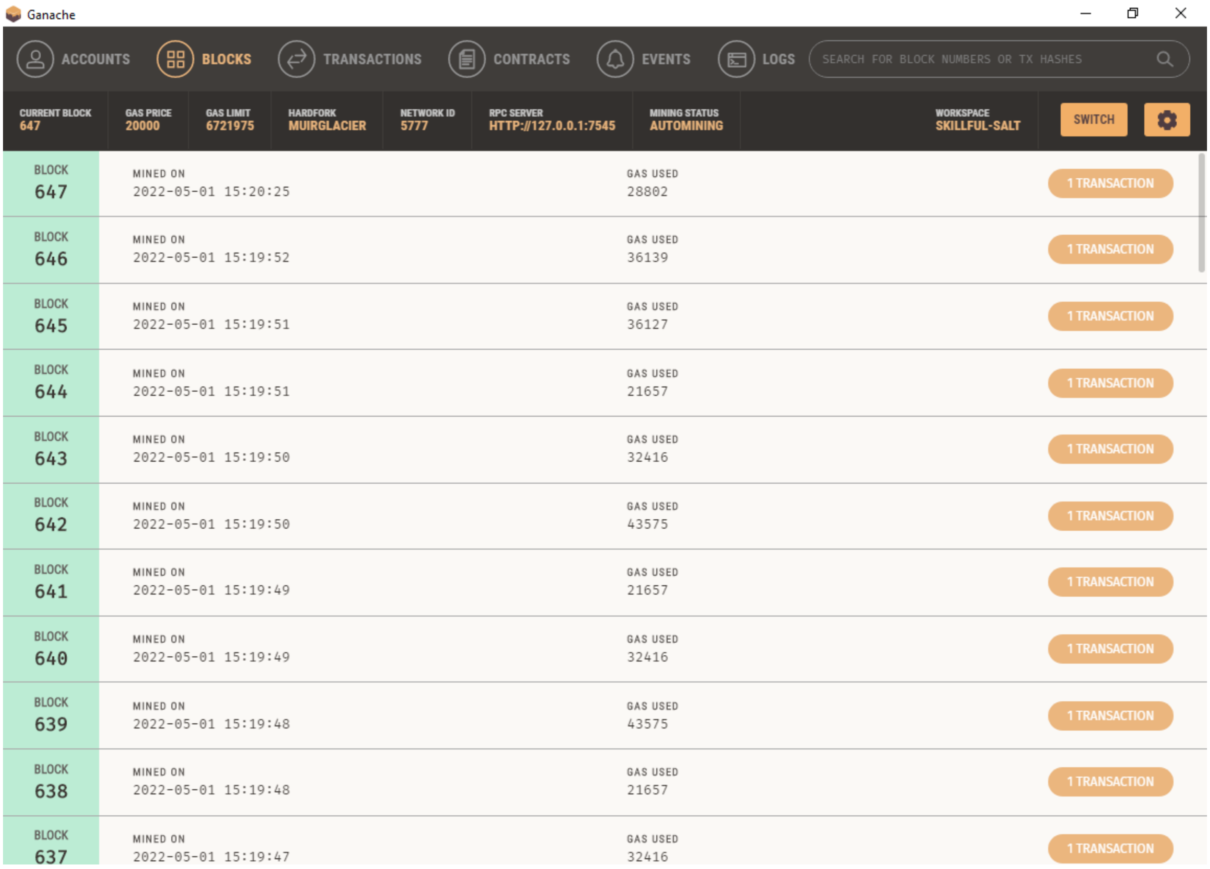

Figure 17: Blocks mined after 50 cycles

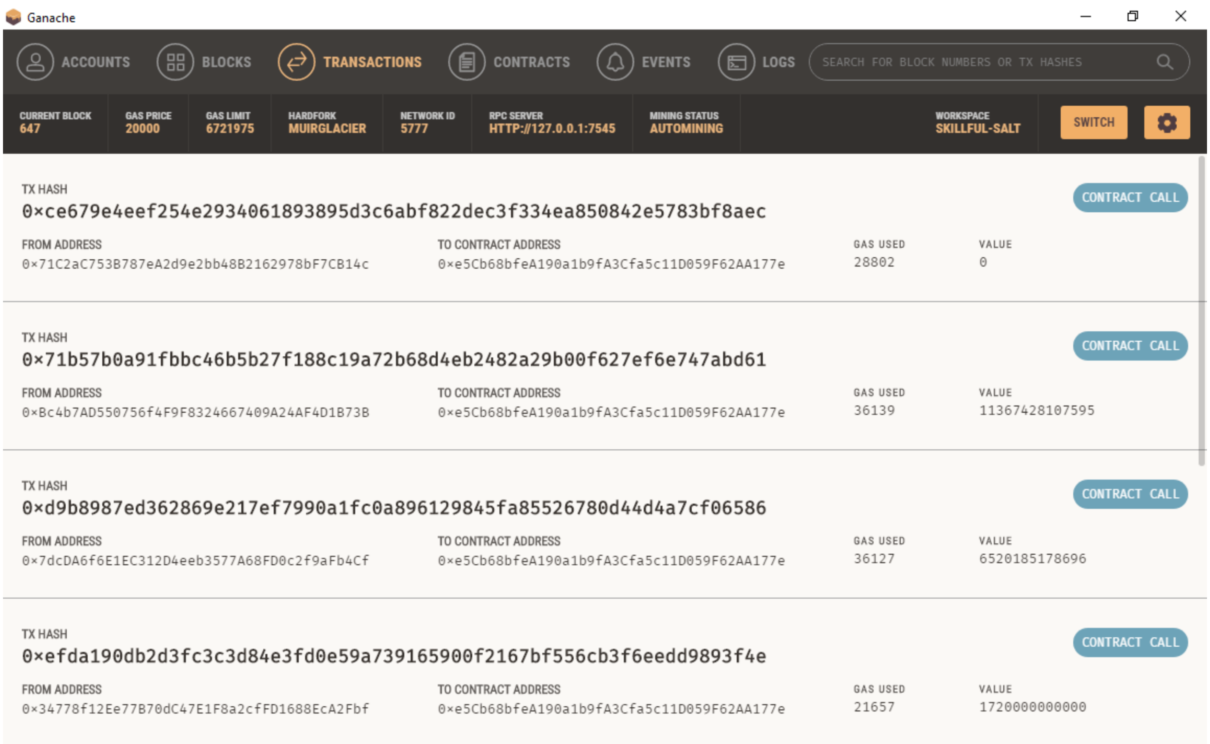

Figure 18: Transactions on the blockchain

Figure 19: Thingspeak channel for Prosumers 1 and 2

Figure 20: Thingspeak channel for Prosumers 3 and 4

Figure 21: Thingspeak channel for Consumers 1 and 2

Chapter 6
Simulation Results

6.1 For Prosumers and Consumers:

Table 2: Results

Parameters	Total Energy	Energy Traded P2P	Gas Used for P2P Transactions	P2P Earnings	Without P2P Earnings	Savings / Extra Revenue
Units	Wh	Wh		Rupees	Rupees	Rupees
Prosumer 1	2397	1938.47	3015345	22.7142	14.0204	8.6938
Prosumer 2	6844	4794.91	3049937	65.5408	54.8853	10.6555
Prosumer 3	-5018	-3190.31	2948626	-63.4119	-70.6537	7.2418
Prosumer 4	13728	11366.74	3047558	144.0438	116.1598	27.884
Consumer 1	-8449	-6670.33	2420750	-91.8784	-107.237	15.3582
Consumer 2	-10718	-8239.43	2375870	-118.43	-136.035	17.605

Thus prosumers earned an average of ₹13.61 more while consumers saved an average of ₹16.48 compared to if they had sold/bought to/from the grid.

6.2 For Grid:

- **With P2P:**

  Energy = 1216 Wh (+ve as grid sold more than it bought)

  Gas Used = 4446128

  Total Revenue = ₹ 41.3326

- **Without P2P:**

  Energy bought from prosumers = -26810 (-ve as this is energy bought)

  Energy sold to consumers = 28026 (+ve as this is energy sold)

  Total Revenue = ₹ 128.8602

  Total Revenue Loss = - ₹ 87.5276

With P2P Energy trading the grid utility company managed to get a revenue of ₹41.33. If P2P Energy trading was not used, instead the utility company managed the buying and selling of power, the company would have had a revenue of ₹128.86. Thus P2P Energy trading resulted in a revenue loss of ₹87.52 for the company.

Chapter 7

Future Scope

Using a test network:

An Ethereum testnet is a collection of nodes that are used to test the Ethereum protocol to ensure that the protocol is working as expected.

Testnets are like mocks in that they are used to test the protocol in a controlled environment as writing smart contracts and deploying them on the mainnet is an expensive operation compared to writing tests and deploying them on the testnet.

Some commonly used testnets are: Rinkeby, Kovan, Ropsten and Görli.

The Project ran on a personal Ethereum blockchain running locally with auto mining enabled. Automining mines a new block for every transaction instead of mining a new block every 12 to 14 seconds like on the ethereum mainnet. So there is one transaction per block so gas limitations per block aren't an issue.

Currently the smart contract cannot be deployed on a test network as it consumes a large amount of gas every cycle which is inefficient, wasteful and expensive.

The smart contract needs optimizations before it can be deployed on a test network.

Using a different blockchain:

Using a permissioned blockchain with a different consensus mechanism which offers scalability, higher transaction throughput and lower block mining time.

Compensation for Voltage regulation and Power factor Improvements

Prosumers which supply reactive power or improve voltage stability can be added and compensation schemes can be developed accordingly.

Chapter 8
Conclusion

Prosumers and consumers both benefit from peer to peer trading, with the prosumer getting higher rates for their energy sold and the consumers getting a lower rate for their energy bought. The utility company which would have taken a cut otherwise is not required.

This system has a wide scope and it promotes the use of clean renewable energy sources. Smart grids can be deployed in both rural and urban areas to tackle various issues.

It can be deployed almost anywhere with sufficient DERs and it merely has an initial setup cost and requires little maintenance.

This system provides secure trustworthy transactions and enables prosumers to trade with consumers and pocket most of the revenue generated.

References

[1] Mihaylov, M.; Jurado, S.; Avellana, N.; Van Moffaert, K.; de Abril, I.M.; Nowé, A. NRGcoin: Virtual currency for trading of renewable energy in smart grids. In Proceedings of the 11th International conference on the European energy market (EEM14), Cracow, Poland, 28–30 May 2014; pp. 1–6.

[2] Wang, J.; Wang, Q.; Zhou, N.; Chi, Y. A novel electricity transaction mode of microgrids based on blockchain and continuous double auction. Energies 2017, 10, 1971.

[3] Li, Z.; Kang, J.; Yu, R.; Ye, D.; Deng, Q.; Zhang, Y. Consortium blockchain for secure energy trading in industrial internet of things. IEEE Trans. Ind. Inform. 2017, 14, 3690–3700.

[4]Tarek Alskaif, Jose Luis Crespo-Vazquez, Milos Sekuloski, Gijs van Leeuwen, Joao P. S. Catalao, "Blockchain-based Fully Peer-to-Peer Energy Trading Strategies for Residential Energy Systems", 2021 IEEE Transactions on Industrial Informatics, 1–1. doi:10.1109/tii.2021.3077008

[5]Subin Kwak, Joohyung Lee, "Implementation of Blockchain based P2P Energy Trading Platform", 2021 International Conference on Information Networking (ICOIN). doi:10.1109/icoin50884.2021.9333876

[6]Berrak Perk, Can Bayraktaroglu, Engin Deniz Dogu, Faizan Safdar Ali, Oznur Ozkasap, "Joulin: Blockchain-based P2P Energy Trading Using Smart Contracts", 2020 IEEE Symposium on Computers and Communications (ISCC). doi:10.1109/iscc50000.2020.9219696

[7]Rabiya Khalid, Nadeem Javaid, Sakeena Javaid, Muhammad Imran, Nidal Naseer, "A blockchain-based decentralized energy management in a P2P trading system", 2020 IEEE International Conference on Communications (ICC). doi:10.1109/icc40277.2020.9149062

[8]Ravi Kishore Kodali, Subbachary Yerroju, Borra Yatish Krishna Yogi, "Blockchain Based Energy Trading", 2018 IEEE Region 10 Conference. doi:10.1109/tencon.2018.8650447

[9]Parag, Y., Sovacool, B. Electricity market design for the prosumer era. Nat Energy 1, 16032 (2016). https://doi.org/10.1038/nenergy.2016.32

[10]D. -H. Park, J. -B. Park, K. Y. Lee, S. -Y. Son and J. H. Roh, "A Bidding-Based Peer-to-Peer Energy Transaction Model Considering the Green Energy Preference in Virtual Energy Community," in IEEE Access, vol. 9, pp. 87410-87419, 2021, doi: 10.1109/ACCESS.2021.3061767.

[11]Wongthongtham, Pornpit & Marrable, Daniel & Abu-Salih, Bilal & Liu, Xin & Morrison, Greg. (2021). Blockchain-enabled Peer-to-Peer energy trading. Computers & Electrical Engineering. 94. 107299. 10.1016/j.compeleceng.2021.107299.

[12]Alessandra, Pieroni & Scarpato, Noemi & Di Nunzio, Luca & Fallucchi, Francesca & Raso, Mario. (2018). Smarter City: Smart Energy Grid based on Blockchain Technology. International Journal on Advanced Science, Engineering and Information Technology. 8. 298. 10.18517/ijaseit.8.1.4954.

[13]T. M. Masaud, J. Warner and E. F. El-Saadany, "A Blockchain-Enabled Decentralized Energy Trading Mechanism for Islanded Networked Microgrids," in IEEE Access, vol. 8, pp. 211291-211302, 2020, doi: 10.1109/ACCESS.2020.3038824.

[14]R. Khalid, N. Javaid, A. Almogren, M. U. Javed, S. Javaid and M. Zuair, "A Blockchain-Based Load Balancing in Decentralized Hybrid P2P Energy Trading Market in Smart Grid," in IEEE Access, vol. 8, pp. 47047-47062, 2020, doi: 10.1109/ACCESS.2020.2979051. Same as 11

[15]Esmat, A., de Vos, M., Ghiassi-Farrokhfal, Y., Palensky, P., & Epema, D. (2021). A novel decentralized platform for peer-to-peer energy trading market with blockchain technology. Applied Energy, 282, 1-16. [116123]. https://doi.org/10.1016/j.apenergy.2020.116123

[16]Desen Kirli, Benoit Couraud, Valentin Robu, Marcelo Salgado-Bravo, Sonam Norbu, Merlinda Andoni, Ioannis Antonopoulos, Matias Negrete-Pincetic, David Flynn, Aristides Kiprakis, “Smart contracts in energy systems: A systematic review of fundamental approaches and implementations”, Renewable and Sustainable Energy Reviews, Volume 158, 2022, 112013, ISSN 1364-0321, https://doi.org/10.1016/j.rser.2021.112013.

[17]Zhipeng Jing, “A Distributed Energy Transaction Method Based on Blockchain” E3S Web Conf. Volume 267, 2021, 7th International Conference on Energy Science and Chemical Engineering (ICESCE 2021), https://doi.org/10.1051/e3sconf/202126701007

[18]Korkmaz, Ayşe & Kılıç, Esra & Türkay, Metehan & Çakmak, Ömer & Arslan, Taha & Erdoğan, Ulaş. (2021). A Blockchain Based P2P Energy Trading Solution for Smart Grids. 10.13140/RG.2.2.30192.58882.

[19]B.P. Hayes, S. Thakur, J.G. Breslin, “Co-simulation of electricity distribution networks and peer to peer energy trading platforms”, International Journal of Electrical Power & Energy Systems, Volume 115, 2020, 105419, ISSN 0142-0615, https://doi.org/10.1016/j.ijepes.2019.105419.

[20]Oh, S.-C., Kim, M.-S., Park, Y., Roh, G.-T. and Lee, C.-W. (2017), "Implementation of blockchain-based energy trading system", *Asia Pacific Journal of Innovation and Entrepreneurship*, Vol. 11 No. 3, pp. 322-334. https://doi.org/10.1108/APJIE-12-2017-037

[21]Jason Lin, "Analysis of Blockchain-based Smart Contracts for Peer-to-Peer Solar Electricity Transactive Markets", Virginia Polytechnic Institute and State University, 2018.

[22]S. Seven, G. Yao, A. Soran, A. Onen and S. M. Muyeen, "Peer-to-Peer Energy Trading in Virtual Power Plant Based on Blockchain Smart Contracts," in IEEE Access, vol. 8, pp. 175713-175726, 2020, doi: 10.1109/ACCESS.2020.3026180.

[23]Schneiders A, Fell M, Nolden C. Peer-to-Peer Energy Trading and the Sharing Economy: Social, Markets and Regulatory Perspectives. Preprints.org; 2021. DOI: 10.20944/preprints202008.0266.v2.

[24]Gough, Matthew, Sérgio F. Santos, Arturo Nieto de Almeida, Mohammad Sadegh Javadi, Tarek Alskaif, Rui Castro and João P. S. Catalão. "Development of a Blockchain-Based Energy Trading Scheme for Prosumers." *2021 IEEE Madrid PowerTech* (2021): 1-6.

[25]Song, Jae G., Eung s. Kang, Hyeon W. Shin, and Ju W. 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

[26]Lee, Joseph and Khan, Vere Marie, Blockchain and Smart Contract for Peer-to-Peer Energy Trading Platform: Legal Obstacles and Regulatory Solutions (March 18, 2020). http://dx.doi.org/10.2139/ssrn.3556260

APPENDIX

```
// SPDX-License-Identifier: UNLICENSED
pragma solidity 0.8.13;

contract SmartEnergy {
    uint public grid_balance = 0;
    mapping (address => uint) balance;
    event energyGenerated(address generated_by, uint energy_value,
uint rate);
    event energyConsumed(address consumed_by, uint energy_value,
uint rate);
    event ProsumerGrid(address prosumer, uint energy_value, uint
rate);
    event ConsumerGrid(address consumer, uint energy_value, uint
rate);

    function ethfromcontract(address payable receiver, uint amount)
public payable {
        return receiver.transfer(amount);
    }

    function ethtocontract(uint256 amount) payable public {
        require(msg.value == amount);
    }

    function pushEnergy(address payable prosumer, uint
energy_pushed, uint rate) public payable {
        emit energyGenerated(prosumer, energy_pushed, rate);
        balance[prosumer] = energy_pushed;
        grid_balance += energy_pushed;
        return prosumer.transfer(energy_pushed*rate);
    }

    function pullEnergy(address consumer, uint energy_pulled, uint
rate) public payable returns (bool check){
        require(msg.value == energy_pulled*rate);
        emit energyConsumed(consumer, energy_pulled, rate);
        balance[consumer] = energy_pulled;
        grid_balance -= energy_pulled;
        return true;
    }

    function G2C(address consumer, uint energy, uint rate) public
payable returns (bool check){
        require(msg.value == energy*rate);
        emit ConsumerGrid(consumer, energy, rate);
        return true;
    }

    function P2G(address payable prosumer, uint energy, uint rate)
public payable {
        emit ProsumerGrid(prosumer, energy, rate);
        return prosumer.transfer(energy*rate);
```

```
    }

    function withdraw() public{
        payable(msg.sender).transfer(address(this).balance);
    }
    function getUserBalance(address user) public view returns (uint
val) {
        return balance[user];
    }
    function get_Contract_Balance() public view returns (uint256) {
        return address(this).balance;
    }
}
```