

Peer-to-Peer Optimal Solar Energy Trading using Proof-of-Authority Blockchain

Mohammad Rafaquat Alam, Teoh Jing Yang

Abstract—With Malaysia’s growing populations and improved lifestyles, the demand for energy would only increase posing new challenges and problems to the energy sector. This project proposes a peer-to-peer web-based solar energy trading platform using Blockchain technology. The aim of this project is to optimize energy usage by homes, such that those that require higher amounts, can trade energy instead of paying higher bills whilst also providing a second income stream for prosumers through production of renewable solar energy. The back end of the platform is built using the Golang, while the Angular framework is used for the front end. Energy forecasting and double auction mechanism is used for optimal energy allocation and they are written in Python. Moreover, the platform uses the MongoDB NoSQL cloud database to securely and reliably store data and used Locust stress testing tool for performance benchmarking.

Index Terms—Solar Energy, Energy Forecasting, Double Auction, Blockchain, Frontend and Backend.

I. INTRODUCTION

AS more people get access to electricity with rising number of appliances and electrical devices in households, it is only fair that the energy consumption for the average person throughout history has been increasing steadily [1], [2]. Grids in Malaysia are controlled by Tenaga Nasional Berhad (TNB), with electricity flowing through main transmission lines that go from the generators all the way to distribution lines which in turn forward energy to industrial, commercial, or residential areas. It is in no doubt that as this demand and consumption for energy keeps increasing, at one point these transmission lines would have to be upgraded [3]. This would take a huge amount of time, resources and money, and if nothing is done the transmission lines will be extremely congested which is already starting to happen now during peak hours.

Another downfall of the energy sector, in some countries, according to consumers would be houses with solar panels, do not utilize the generated surplus energy to its full potential. They do not benefit in any way from the surplus energy they generated as all of it is dissipated to the grid for free [4]. This reduces overall morale for the general public as there is no incentive from the extra energy they produce thus demotivating them to use any kind of renewable energy generation in their house.

We combat the aforementioned problems with our P2P Energy trading Blockchain web application that guarantees prosumers some reward by trading the renewable energy they produced. Our solution is a Proof-Of-Authority Distributed Ledger web based trading platform which uses Machine Learning to predict prosumer energy consumed and produced and then use that prediction to calculate the Optimal Energy

that can be allocated to the prosumer(buyer) from another prosumer(bidder). So prosumers can trade surplus energy they generated on the market, while consumers are able to buy it without needing to get electricity from the mains, resulting the overall demand and stress on the transmission lines to be reduced [4], [5]. Consequently, infrastructure upgrade costs and resources can be invested in other projects as this upgrade will not be necessary as before. Furthermore, an incentive is now created for residential people to own solar panels so they can start selling on the P2P platform, thus increasing the usage of renewable energy leading to a cleaner and greener environment.

II. METHODS

A. Energy Forecasting

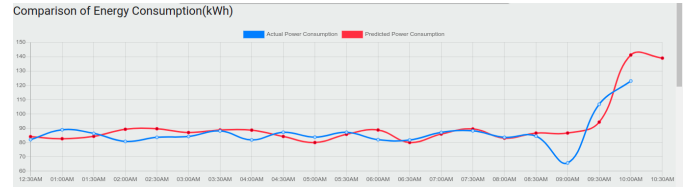


Fig. 1. Prosumer(buyer) energy consumption forecast for the next 30 minutes

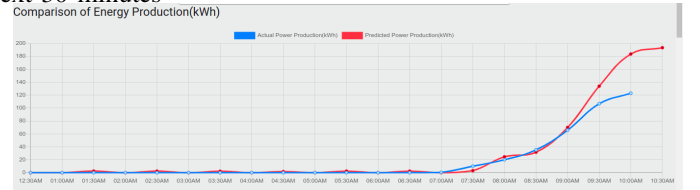


Fig. 2. Prosumer(bidder) energy production forecast for the next 30 minutes

1) *Why Energy Forecasting:* For our implemented web application, we are assuming that the solar energy a prosumer generates is stored in a battery and the energy balance is stored in a smart metre in their homes. Using this stream of data from the smart metre, we forecast the amount of energy consumed as well as produced for a given day. By predicting the energy consumption, we can limit the prosumer to a max energy value that they can make an order for. This ensures, that no prosumer orders extremely large energy amount that even the grid cannot produce. Predicting the energy production, allows us to limit the prosumer to a max energy value that they can use to make a bid on an energy request. This ensures, that no prosumer makes a bid that exceeds the max amount they are expected to produce.

2) *How Energy Forecasting is done:* The energy forecasting uses the Machine Learning technique of Triple Exponential

Smoothing [10] with Additive Trend and Seasonality where we use the latest two data points for seasonal period [9]. The training also includes Trend Damping to prevent the rise of unrealistic trends and thereby account for Seasonal Irregularity [10]. Once training is done using all the data points upto the current time of the day, a prediction is generated for energy consumption/production for the next 30 minutes from the current time. This is the maximum energy forecast for consumption or production depending on what feature the prosumer is making use of. If the prosumer wants to make an order, they are shown their forecast graph for energy consumption as in Figure 1. If the prosumer wants to make a bid on an energy request, they are shown their forecast graph for energy they can produce as shown in Figure 2.

B. Optimal Energy Allocation

1) *Social Welfare Maximisation (SWM)*: The SWM problem is the objective function that we hope to maximise to fulfill the energy request of prosumers by maximising the energy supply from bidders whilst considering the cost to generate the energy. The amount of energy that would satisfy a prosumer(buyer) based on their smart metre data and energy consumption forecast is defined by the satisfaction function in [6] as:

$$U_i(E_i^n) = w_i \left[\ln \left(n \sum_{j=1}^J (e_{ij}^n - e_i^{n,min}) + 1 \right) \right] \quad (1)$$

Please see the appendix section for more details. Next we need to consider the cost incurred for the bidding prosumer when they trade the energy they produced as per the production forecast. The cost function is given in [6] as:

$$L_i(S_j^n) = c_1 \sum_{i=1}^I (s_j^n)^2 + c_2 \sum_{i=1}^I (s_{ji}^n) \quad (2)$$

Please see the appendix section for more details. Now that we know how much energy would satisfy the prosumer as shown in Eq. (1) and the cost the bidder has to bear to produce and trade energy as in Eq. (2), we can determine the objective function for SWM as:

$$SWM : \max_{E_n, S_n} \sum_{i=1}^I U_i(E_i^n) - \sum_{j=1}^J L_j(S_j^n) \quad (3)$$

The SWM is subject to the following constraints.

$$\begin{aligned} \text{Subject to : } & e_i^{n,min} \leq \sum_{j=1}^J e_{ij}^n \leq e_i^{n,max}, \forall i \in E, \\ & \sum_{i=1}^I s_{ji}^n \leq S_j^{n,max} \forall j \in Z, \\ & \rho s_{ji}^n = e_{ij}^n, \forall i \in E, \forall j \in Z, \\ & e_{ij}^n \geq 0, \forall i \in E, \forall j \in Z. \end{aligned} \quad (4)$$

2) *Optimal Allocation Problem (OAP)*: The OAP, ensures no one prosumer can out bid everyone else, meaning all bidders are guaranteed to receive fiat reward as long as they bid on the request. Moreover TNB, as the grid providers, are also guaranteed to receive a reward since the system is designed to use their infrastructure to distribute the energy. The OAP is defined in [6] as :

$$OAP : \max_{E^n, S^n} \sum_{i=1}^I \sum_{j=1}^J [b_{ij}^n \ln e_{ij}^n - p_{ji}^n s_{ji}^n] \quad (5)$$

See Appendix for details.

Both ‘OAP’ and ‘SWM’ have same subject to constraints as described in Eq. (4). Hence we carry out constraint relaxation through Lagrangian method [6] to find the local maximum [10]. So, energy receivable by a buyer, e_{ij}^n and energy tradable by a bidder, s_{ji}^n , after applying Lagrangian method is:

$$e_{ij}^n = \frac{b_{ij}^n [(n \sum_{j=1}^J e_{ij}^n - e_i^{n,min}) + 1]}{nw_i} \quad (6)$$

where $e_i^{n,min}$ is the fixed minimum energy for a buyer, and:

$$s_{ji}^n = 2c_1 s_{ji}^n + c_2 \quad (7)$$

where, c_1 and c_2 are cost factors for the bidder.

3) *Double Auction Mechanism (DAM)*: The Double Auction Mechanism (DAM) solves the OAP via Lagrange method. As shown in Figure. 3, The web app runs the DAM every 30 minutes where it takes all closed requests, loops through them, applying Eq. (6). and Eq. (7). on each bid. The Lagrange scales down the bid energy amount by 10 to 15% on average to prevent one bidder to always dominate. This also reduces the amount payable by prosumer to each bidder. The prosumer however, still pays the full price of the energy request they made. So, once all the bidders are considered, the system checks whether the total energy demand made by the prosumer in the request is satisfied or not. If demand is satisfied, the remaining fiat amount that is not received by any bidder is transferred to TNB. If demand is not satisfied the surplus energy is taken from TNB. The rate of energy is capped at RM 0.20 per kWh $\pm 20\%$ for all bidders as well as TNB. So, all bidders and TNB receive fiat currency as reward and buyer only pays for the amount of energy they need which is always fulfilled.

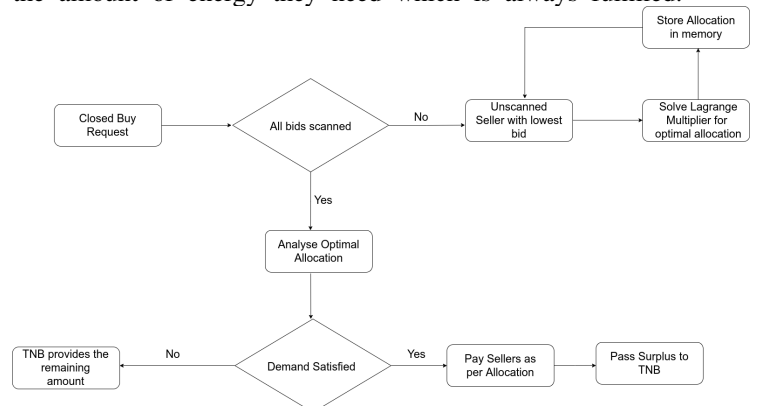


Fig. 3 Flowchart for double auction with demand and supply response

C. Blockchain

As shown in Fig.4, the blockchain uses a Proof of Authority (PoA) consensus mechanism.

1) *Roles in the Blockchain:* Only validators are given the authority to mine new blocks and add them to the blockchain. We also have a new kind of users called clerks who are there for validator security and accountability checks. This is to ensure that validators do not conspire to work against public interest and if validator accounts are compromised, then clerks can help recover the system.

- *Validator:* A validator can verify transactions, mine new blocks, add and discard blocks. Since we are using a Proof of Authority consensus mechanism, the validators undergo a rigorous registration process where they need to reveal their identities. Their reputation is at stake which means if they go against the interest of the normal users, then their status as validators will be revoked and made known to the greater community.
- *Clerk:* Clerks provide an additional layer of integrity check on the blockchain to ensure that validators do not conspire against the community. After the addition of every new block, they receive an updated local copy of the blockchain and user accounts. They will use their local copies to check whether the nonce of the last block from the central blockchain provides the same hash when they use it on the transactions in the last block in their local blockchain copy. If the match does not happen for more than 50% of the clerks, then an integrity check is triggered. Unlike validators, any normal node can be made a clerk and they do not need to be rigorously identified. Clerks also provide the benefit of recovering the blockchain in case a validator account is compromised.

2) Blockchain pipeline:

- *Signing Smart Contracts:* After the double auction is run every 30 minutes, the new pool of transactions are broadcasted over the network to all the validators. The validator who receives the transaction pool first, will verify each transaction where they check whether the buyer has sufficient balance or not. If so, then that transaction is marked as verified and made part of a temporary block. If a buyer does not have the required balance then the transaction is marked invalid. Once all the transactions have been checked, and added to the temporary block, it is then broadcasted to all the remaining validators. These validators use their local copy of user accounts to verify each transaction in the temporary block. They then use the nonce of the temporary block to hash the transactions from the latest block in their local blockchain. If this hash matches with that of the temporary block for all validators, then the temporary one is made permanent and added to the central blockchain. The last validator who checks the hash finalizes the smart contract.
- *Discarding Blocks:* If there is a validator who does not find a match for the hash, then their local copies of user accounts and blockchain is updated. Then the check is done again. If the hash fails to match a second time,

then that block is discarded. The non-match signifies that a transaction was manipulated in the central blockchain and so the block permanent discarding is justified.

- *Integrity Check:* After the formation of 5 new blocks, an integrity check is triggered where the clerks verify each transaction in the latest permanent block in the central blockchain. They use their local copy of user accounts to verify each transaction in the latest block. Then use the nonce of the latest block from central blockchain to hash the transactions from the latest block in their local blockchain. If this hash matches that in the central blockchain for more than 50 percent of the clerks, then there is no issue but if the match is less than 50 percent, then an integrity check is issued. This goes through the local blockchain copy of each validator and compares the hash of the latest block. The validator(s) whose hash has a mismatch is then flagged. In this case the validator access may be revoked. This guarantees that validators do not work against public interest.

III. RESULTS

A. Energy Trading As A Source Of Income

As mentioned earlier, the double auction mechanism in the web application ensures that all bidders can make some fiat income as long as they make a bid. This serves as an incentive for more people to invest in equipment to generate electricity using solar panels. The income chart is accessible from the user profile page as shown in Fig. 5. The chart shows the total income and energy traded for the days in which the user made a bid offer.

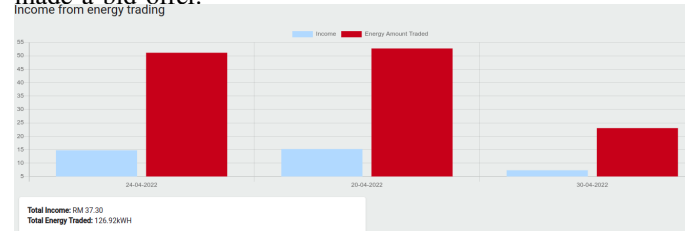


Fig. 5. Income that users can generate by bidding on the application

Since we are using the grid infrastructure provided by TNB, the application ensures that they receive some fiat reward for their participation in the energy trading. The income chart for TNB is shown in Fig. 6. The TNB account is a validator account. No matter what bids are made on an energy request, TNB always provides a fraction of the energy demand and hence this income is always guaranteed. If there are no bids on an energy request, then the full demand is fulfilled by TNB. In such a case they receive the full price of the energy.

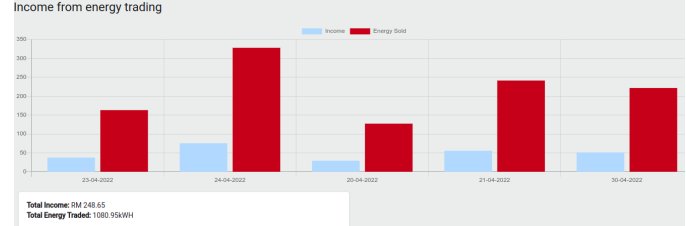


Fig. 6. Income that TNB can make

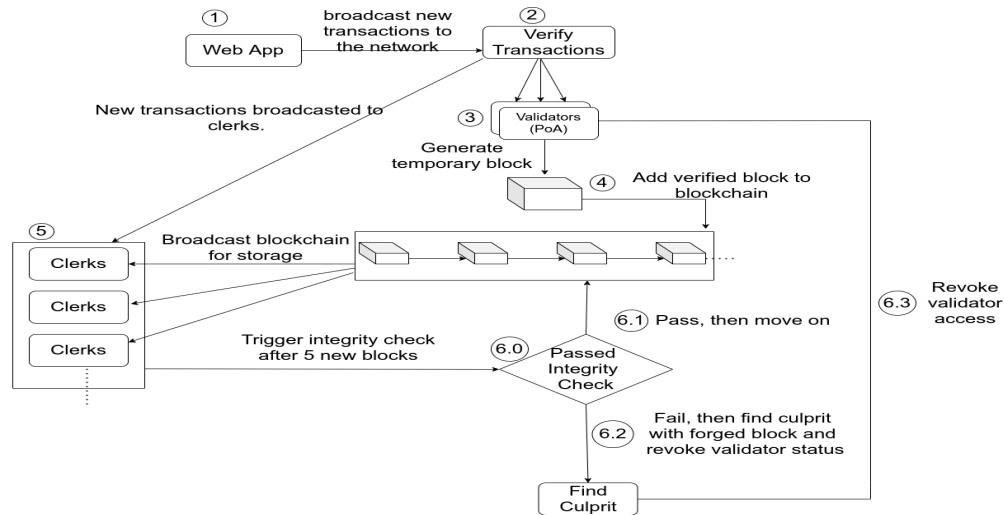


Fig. 4. Proof of Authority blockchain with accountability check via clerks

B. Blockchain Management

1) *Blockchain on the frontend of web application:* The blockchain exists in two formats. One of them is the centralised one, stored in the database. As shown in Fig. 7, the central blockchain is available to all users from the "blockchain" page of the web application. The index, hash, nonce and the previous hash is shown for each block. In addition to these, when the info icon is clicked, users can see some general information regarding the transactions in each block: the buyers, total energy and fiat traded and the income TNB generates from this transaction. This is shown in Fig. 8.

Index	Hash	Previous Hash	Nonce
0	00366b81436deefad74e7cf50961abcb81858e15e31967c287da522724c33		6
1	08f0cbac423e303e50f03f84f5ba1ee2d5d0c1dcf4734355fa7cd55daa1353a5	00366b81436deefad74e7cf50961abcb81858e15e31967c287da522724c33	5
2	05b4d60f384a4925aa9a51cfff4d3b4316afa6d544e6b47795892b59313615862	08f0cbac423e303e50f03f84f5ba1ee2d5d0c1dcf4734355fa7cd55daa1353a5	9
3	0b0ba7519fc3208c682f68b3f1dbb26a01891b27ce66e1c6e2d9afce383a022	05b4d60f384a4925aa9a51cfff4d3b4316afa6d544e6b47795892b59313615862	1
4	042ce6b5f6376fc517ca006fec1ae162ebda776d46586623629474778c0b2e1	0b0ba7519fc3208c682f68b3f1dbb26a01891b27ce66e1c6e2d9afce383a022	15

Fig. 7. Index, Hash, Block Info Icon, Previous Hash and Nonce are displayed on the frontend

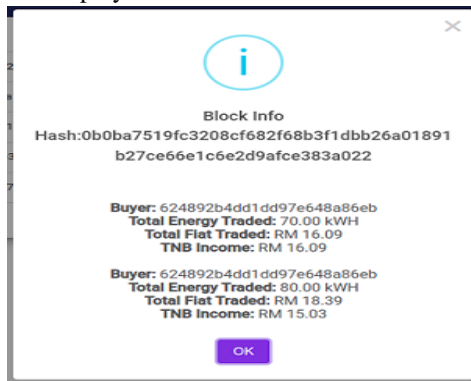


Fig. 8. Transaction information in the block

2) *Blockchain on the backend of web application:* As mentioned before, the backend stores the blockchain in a central database as shown in Fig. 9. This is used as the primary blockchain throughout the system. The frontend uses the data available in this blockchain.



Fig. 9. Blockchain as stored in the central database

The other blockchain format is a local copy stored in the validators' and clerks' device as a .JSON file and this is shown in Fig. 10. As discussed before, the application uses this local copy to verify transactions and check the hash in the central database before finalizing the blocks. The app always updates the local copy using the central one when the verification and integrity checks are done. This means there is no chance for clerks and validators to edit the local copy and manipulate the central blockchain.

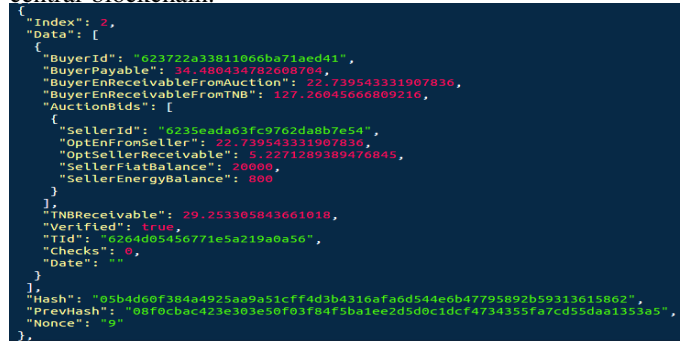


Fig. 10. Snippet of a block stored in the local blockchain file

C. The Large Scale Web Application

1) *Load Testing:* When the platform is eventually released to the public, it need to serve lots of users concurrently without any issues. We used a swarming tool called Locust to

simulate such scenarios. The load testing simulates 100 normal users concurrently using the system. Each user first visits the homepage, creates an order, makes a bid, visits the marketplace and finally the blockchain page. Such sequence ensures real user behaviour when the system is deployed. The system performance handling concurrent user requests and system responses is shown in Fig. 11 and Fig.12 respectively. From both graphs, we can see that there are no failures meaning the system works as intended under normal loads.

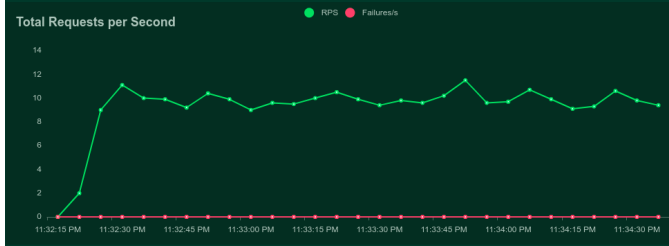


Fig. 11. Handling concurrent requests on backend servers.

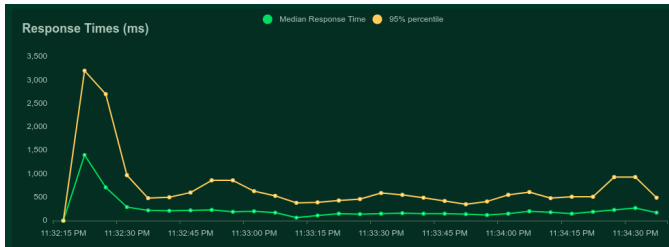


Fig. 12. System responding to concurrent user requests.

2) *Stress Testing*: Next stress testing is done to find out at what capacity the system fails. This will help us determine which upgraded database version to use during deployment since the current one is using the free cluster service from mongodb. It also shows us how gracefully the system fails. The results from stress testing are shown in Fig. 13 and Fig. 14 respectively. In both cases the request failures and response time started to increase when we hit the 1000 user threshold.

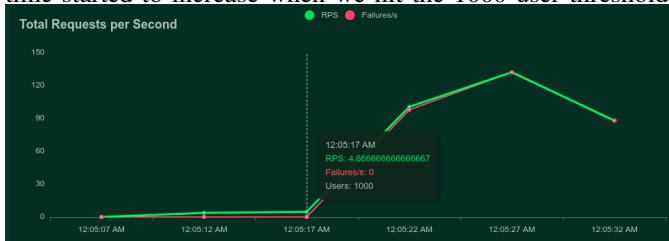


Fig. 13. Request handling by backend servers under high stress.

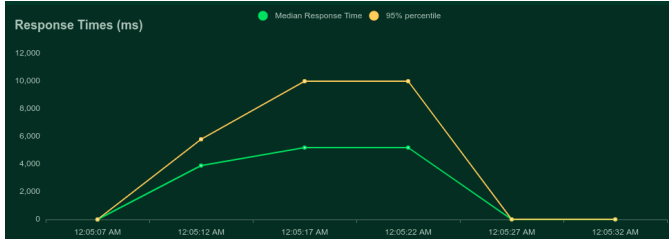


Fig. 14. Response handling by backend servers under high stress.

3) *Blockchain Performance*: As the web application is still in prototype phase, the mining of blocks were kept to the lowest difficulty. That is each block is solved/mined when

there is nonce that returns one zero at the start of the hash. With this level of difficulty, we tested the blockchain mining time with increasing transaction size as shown in Fig. 11. Since the hashing/mining algorithm only increments the nonce and checks for zero character at the start of the hash string, there is no specific observation. However, in the real world scenario, as block size increases with increasing transaction sizes, the difficulty would need to be increased and hence the mining time will increase as well.

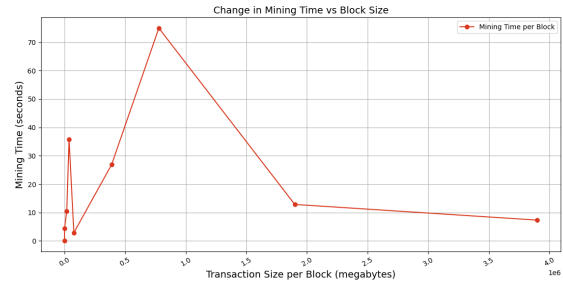


Fig. 11. Time taken to mine a block with increasing block size.

D. Technology Stack

As promised in the project proposal, the frontend of the web application is written in Angular 13. Since this version of Angular uses Typescript programming language, the middleware services that let the frontend communicate with the backend is also written in Typescript. The middleware services use http POST request to inform the backend code which module needs to be run to fulfill a request. The backend is written in Go programming language. This language is chosen due to its speed, easy syntax and its increasing popularity as a server side language. The backend code listens for requests from the frontend on dedicated http ports. Based on the frontend request, it will run the required code module. Finally, Python programming language libraries are used for the Energy Forecasting. The libraries include numpy for complex calculations, matplotlib to produce the chart and its data, pandas for data processing and statmodels for its ExponentialSmoothing library.

E. Source Code

The web application over a period of 6 months from December 2021 to May 2022. The code is made available at: https://gitlab.com/mohammad_rafaquat_alam/imdc_p2p_energy_trading If there are issues in accessing the repository please kindly email:md.rafaquatalam98@gmail.com. A demonstration video is made available at this link to showcase the usage of the web application.

IV. DISCUSSION

A. Security Concerns

The key security issue of the system is that the private and public keys of all the users are hashed and stored in the database. Even though the hashing algorithm is not reversible this is still not good software development practice. Attackers

can launch a dictionary attack with a known hash table and therefore obtain the real private key. So, further work needs to be done to write a mechanism to store and extract the private key of users from the user devices as opposed to the database.

B. Database Upgrade

As shown in the results section, the free mongodb database clusters that are used are not suitable for large scale use. The system crashes when 1000 users are using the system concurrently. To fix, this funding is required to upgrade to a better and reliable mongodb service. The underlying database code would remain the same so this fix is very simple.

V. CONCLUSION

Based on all the previous sections, it is clear that the Blockchain technology is here to stay and keep penetrating even more industries. P2P Optimal Solar Energy trading has a huge potential to make ripples through the energy industry. In conclusion, team MY157 have successfully implemented a peer to peer optimal solar energy trading platform using blockchain technology.

APPENDIX

A. Optimal Energy Allocation Equations

$$U_i(E_i^n) = w_i[\ln(n \sum_{j=1}^J (e_{ij}^n - e_i^{n,min}) + 1)]$$

where E^n is the satisfactory energy receivable for prosumer, w_i is the battery charging willingness, n is the battery charging efficiency, J is the total number of bids on the request, e_{ij}^n is the energy the prosumer(buyer) can receive from the bidder as per the prosumer's consumption forecast. Next we need to consider the cost incurred for the bidding prosumer when they trade the energy they produced as per the production forecast. The cost function is given in [6] as:

$$L_i(S_j^n) = c_1 \sum_{i=1}^I (s_j^n)^2 + c_2 \sum_{i=1}^I (s_{ji}^n)$$

where S^n is the energy that bidder can provide, I is the request made by the prosumer, c_1 and c_2 are cost factors for the bidder, s_j^n is the energy the bidder can produce as per their forecast, s_{ji}^n is the energy the bidder wants to trade with the buyer from the total produced. Now that we know how much energy would satisfy the prosumer as shown in Eq. (1) and the cost the bidder has to bear to produce and trade energy as in Eq. (2), we can determine the objective function for SWM as:

$$SWM : \max_{E^n, S^n} \sum_{i=1}^I U_i(E_i^n) - \sum_{j=1}^J L_j(S_j^n)$$

$$\text{Subject to : } e_i^{n,min} \leq \eta \sum_{j=1}^J e_{ij}^n \leq e_i^{n,max}, \forall i \in E,$$

$$\sum_{i=1}^I s_{ji}^n \leq S_j^{n,max} \forall j \in Z,$$

$$\rho s_{ji}^n = e_{ij}^n, \forall i \in E, \forall j \in Z,$$

$$e_{ij}^n \geq 0, \forall i \in E, \forall j \in Z.$$

B. Optimal Allocation Problem Equations

$$OAP : \max_{E^n, S^n} \sum_{i=1}^I \sum_{j=1}^J [b_{ij}^n \ln e_{ij}^n - p_{ji}^n s_{ji}^n]$$

where b_{ij}^n is the price the bidder wants from the prosumer, p_{ji}^n is the price the prosumer is willing to pay to bidder,

$$e_{ij}^n = \frac{b_{ij}^n [(n \sum_{j=1}^J e_{ij}^n - e_i^{n,min}) + 1]}{nw_i}$$

where $e_i^{n,min}$ is the fixed minimum energy for a buyer, and:

$$s_{ji}^n = 2c_1 s_{ji}^n + c_2$$

where, c_1 and c_2 are cost factors for the bidder.

This piece of work would have not been possible without the support of our family, friends and university professors and their firm belief in my potential. The team is immensely grateful for the efforts of my supervisor Dr. Tan Wen Shan, for believing in our capabilities since he chose us for this project, and for supporting and trusting our decision making, until the very end of it.

REFERENCES

- [1] G. "U.S. energy facts explained - consumption and production - U.S. Energy Information Administration (EIA)", Eia.gov, 2021.
- [2] H. Ritchie and M. Roser, "Energy", Our World in Data, 2021.
- [3] "Outline of Malaysia's electricity system - Power Engineering International", Power Engineering International, 2021.
- [4] J. Yang, A. Paudel, H. B. Gooi, and H. D. Nguyen, "A proof-of-stake public blockchain based pricing scheme for peer-to-peer energy trading," Applied Energy, vol. 298, p. 117154, 2021.
- [5] S. Aggarwal and N. Kumar, "A consortium blockchain-based energy trading for demand response management in vehicle-to-grid," IEEE Transactions on Vehicular Technology, vol. 70, no. 9, pp. 9480-9494, 2021.
- [6] The consortium paper
- [7] The demand and supply one by Dr. Tan
- [8] Proof of Authority
- [9] Time-series Forecasting - Complete Tutorial (indian guy)
- [10] A Gentle Introduction to Exponential Smoothing for Time Series Forecasting in Python
- [11] A Gentle Introduction To Method Of Lagrange Multipliers