2025 12th International Conference on Emerging Trends in Engineering & Technology - Signal and Information Processing (ICETET - SIP) | 979-8-3315-0099-3/25/$31.00 ©2025 IEEE | DOI: 10.1109/ICETETSIP64213.2025.11156685

A Peer-to-Peer Decentralized Smart Energy Grid

Aryan Patel
Department of Electronics and Communication Engineering
MIT-WPU
Pune, India
aryanpatel1281990@gmail.com

Siddhart Sharma
Department of Electronics and Communication Engineering
MIT-WPU
Pune, India
theonesiddharth@gmail.com

Zameer Siddique
Department of Electronics and Communication Engineering
MIT-WPU
Pune, India
zameersid10@gmail.com

Prof. P.S. Mahajani
Department of Electronics and Communication Engineering
MIT-WPU
Pune, India
priyamwada.mahajani@mitwpu.edu.in

Abstract— This article proposes a completely integrated decentralized energy trading system using a peer-to-peer (P2P) architecture with embedded IoT devices, real-time communication, and blockchain-based transaction validation. The proposed system makes use of ESP8266 microcontrollers, INA219 sensors, and solar-powered storage using lithium-ion batteries to enable energy sharing between smart nodes in a local microgrid. A robust backend architecture, developed using Node.js and Express.js, enables real-time data sharing using WebSocket protocols. The frontend, developed using ReactJS and Tailwind CSS, provides a responsive and dynamic user interface to track transfer progress and control device operations. All energy transactions are secured and logged through smart contracts run on a private Ethereum network via the Ethers.js library. The backend also connects blockchain operations and device state synchronization through a scalable WebSocket gateway layer. Publish-subscribe models in real-time are utilized for frontend notifications. Safety features such as low-voltage cutoff and energy monitoring provide reliable energy trades. The suggested framework demonstrates a viable method for scalable, secure, and interactive energy sharing in decentralized smart grids.

Keywords—Peer-to-peer energy trading, ESP8266, WebSocket, ReactJS dashboard, decentralized grid, blockchain, Ethers.js, real-time monitoring, IoT, smart contracts, renewable energy system, publish-subscribe model, INA219.

I. Introduction

The rapid development of intelligent energy systems and the growth in renewable energy sources have resulted in the development of decentralized power distribution systems. Traditional centralized grids are increasingly facing challenges because of variable demand patterns, user-generated power (prosumers), and the requirement for energy exchange resilience. In response to these hurdles, this paper introduces a Peer-to-Peer (P2P) Decentralized Smart Power Grid System aimed at enabling secure, real-time energy trading through a blend of embedded hardware, blockchain smart contracts, and web-based interfaces for users.

At the core of the system are two hardware units based on the ESP8266 microcontroller, in conjunction with INA219 voltage/current sensors and Li-ion battery modules powered by 10W solar panels. The devices monitor energy parameters and communicate bi-directionally with a backend server using WebSocket protocols. The energy transfer process is governed through relay modules, which are monitored by safety protocols like voltage limits and battery state of charge (SoC) checks.

The backend layer, developed using Node.js and Express.js, coordinates communication between devices, users, and blockchain contracts. Real-time energy data from ESP nodes is gathered and processed, and energy transfer control commands (e.g., START/STOP_TRANSFER) are dispatched through a publish-subscribe pattern over Socket.IO. Device-to-user mapping is achieved through the utilization of distinctive device IDs and authenticated sessions to enable personalized interaction with the energy trading platform.

Blockchain integration is achieved via smart contracts in a private Ethereum network. With the use of the Ethers.js library, backend calls to methods such as registerUser, tradeEnergy, and log Transaction ensure transparency and immutability of energy trading transactions. A Blockchain Gateway Layer is responsible for wallet management, transaction signing, and monitoring blockchain events.

The frontend, developed with ReactJS and styled with Tailwind CSS, provides a responsive user dashboard upon which authenticated users can enter trading intentions, observe real-time flow of energy, and obtain status updates on system status. Modular component-based design facilitates seamless integration of live data, modal pop-ups, and dynamic dashboards through publish-subscribe communication from the back end.

This end-to-end architecture enables safe, scalable, and user-friendly peer-to-peer trading of energy, providing real-time visibility and automatic smart contract validation. The system is a major breakthrough in decentralized energy exchange platforms, combining cheap hardware, fault-tolerant communication, and blockchain trust guarantees.

II. Literature Review

Recent advancements in smart grid technology and decentralized energy exchange have made the creation of peer-to-peer (P2P) energy sharing platforms possible, which improve efficiency, transparency, and user choice. Several studies have proposed frameworks that utilize blockchain, IoT, and renewable sources to provide secure and real-time energy trading among prosumers and consumers within microgrids.

In [1], the authors introduced a blockchain-based energy trading platform for microgrids that uses smart contracts to facilitate transparent financial settlements. Their system, however, did not have an added layer for real-time monitoring. A hardware-based system was introduced in [2], utilizing solar charging nodes ESP8266 to develop a low-cost wireless energy sharing prototype. However, scalability and integration with blockchain were not addressed.

IoT-centric approaches, presented in [3] - [6], involve sensor-based energy monitoring using ESP32 and INA219 modules. These studies presented building blocks in power measurement and control logic, which were the basis of our sensor-based relay switching system. [7] and [8] also explored safety in energy transfer using voltage cutoff mechanisms, which informed our adoption of a 3.3V shutdown system.

Low-latency IoT communication using WebSocket was experimented with in [9] - [12], where sensor networks were able to create robust real-time data exchanges to cloud-based dashboards. Our solution improves upon this by integrating Socket.IO with a Node.js server and a ReactJS client, allowing real-time update notifications via a publish-subscribe model.

The synergy of smart contracts and energy metering has been highlighted in research [13]–[16], with Power Ledger and Brooklyn Microgrid showing functional blockchain-based energy marketplaces. Our design builds on such systems by incorporating real-time device feedback and correlating transfer sessions to authenticated users via device IDs.

In frontend development, [17] and [18] studied web interfaces with ReactJS and MQTT/WebSocket integration for real-time data visualization in industrial environments. Their methods were adapted to our energy transfer dashboard where users could view system metrics and control energy flows.

From the backend perspective, [19]–[21] proposed RESTful APIs and secure communication methods for integrating web servers with embedded systems. We used JWT authentication and WebSocket middleware to provide secure, bidirectional communication between ESP8266 devices and the servers.

Finally, research [22]–[27] dealt with estimation of battery states and energy tracking, especially for Li-ion modules. These results impacted our energy calculation method, which calculates watt-hours in real-time to ensure server-side thresholds during transfer operations.

In summary, our system combines innovations from distributed ledger technologies, embedded IoT devices, and modern web development. Unlike other solutions, this system brings complete synchronization among ESP hardware, blockchain transactions, and an interactive web interface—forming a novel, real-time, and scalable P2P smart grid system.

III. Proposed Methodology

The suggested peer-to-peer (P2P) energy trading system combines embedded hardware for control and data acquisition, a web-based frontend for user interface, a Node.js backend for decision-making, and a blockchain layer for tamper-proof transaction logging. The system runs in real-time and facilitates secure bidirectional communication between physical devices and digital interfaces

A. System Architecture

The design comprises two energy peers, one of which has solar panels, battery packs, and sensing modules. The devices track energy availability, react to commands from the server, and chat with the backend server through WebSocket by means of a special device identifier. The backend handles trade matching and approval by means of real-time data and blockchain authentication. The frontend displays the energy flow and offers control over the process by the user.

B. Hardware Layer

Each Every energy node has the following components:

- Solar Panel (10W): For harvesting energy.
- TP4056 Module: Allows charging of Li-Ion battery in a controlled manner.
- Li-Ion Battery (3.7V, 4000mAh): Serves as an energy buffer as well as a source.
- INA219 Sensor: Delivers real-time voltage and current to the load.
- ESP8266: A microcontroller with Wi-Fi capabilities used for communication and control.
- Relay Module (5V Dual-Channel with Optocoupler): Manages power transfer according to server commands.
- XL6019 Buck-Boost Converter: Provides stable voltage levels to ESP and load with 75–80% efficiency.

C. Data Acquisition and Sensor Processing

The ESP8266 measures voltage and current values via the INA219 sensor every 5 seconds. Power and energy transfer are calculated from these. The State of Charge (SoC) of the battery is estimated from voltage levels, using a linear interpolation formula:

$$SoC = \frac{V_{bus} - V_{min}}{V_{max} - V_{min}} \times 100$$

Where, Vmin = 3.2V and Vmax = 4.2V.

Energy transferred is computed cumulatively using:

$$E = P \times \frac{T}{3600} \quad \text{(Wh)}$$

Where, T = 5 seconds.

D. Backend and Control Logic

The ESP8266 communicates with the backend server via WebSocket using the Socket.IO protocol. All data transfer is in JSON format and includes properties like device_id, voltage, current, power, SoC, energy, and relay_status. An example of this type of payload is:

{
"device_id": "ESP8266_01",
"voltage": 3.84,
"current": 0.90,
"power": 3.45,
"SoC": 76.23,

```
  "energy": 1.23,
  "relay_status": "ON"
}
```

The server handles this information and replies with a JSON control message:

```
{
  "device_id": "ESP8266_01",
  "message": "START_TRANSFER"
}
```

The backend is built with Node.js and relies on Socket.IO for real-time communication with ESP devices and the ReactJS dashboard. A publish-subscribe model is used as follows:

ESP → Server → Publish to frontend

Request from frontend → Server → Command for ESP

If the energy transferred is above a user-specified threshold (e.g., 10 Wh), or if the voltage goes below safe levels, the server will automatically transmit a "STOP_TRANSFER" message.

E. *Backend and Trade Logic*

The server associates each ESP device to a buyer/seller. It tracks energy usage and initiates a STOP_TRANSFER if:

- Voltage drops below 3.3V (safety threshold)
- Transfer violates established energy limit (e.g., 10 Wh)

Otherwise, it sends START_TRANSFER for continuation.

F. *Blockchain Integration*

The framework consists of a Hardhat-driven private Ethereum blockchain. Smart contracts implemented in Solidity specify:

- User registration (registerUser)
- Energy trading operations (requestTrade, completeTrade)
- Wallet credit checks

Node.js and Ethers.js are employed in a blockchain gateway that interacts with smart contracts and monitors events in real time.

A middleware, the Blockchain Gateway Layer (implemented using Node.js and Ethers.js), is employed for interacting with smart contracts. It offers:

- Secure wallet-based interactions
- Contract methods (e.g., registerUser, tradeEnergy)
- Event listening (e.g., EnergyTraded event trigger)

This ensures all trades are transparent, verifiable, and logged permanently.

G. *Real-Time Frontend Dashboard*

A React frontend displays live sensor readings and trading status in a publish-subscribe pattern. Admins can see device-level metrics and transaction status. The frontend listens to backend updates through Socket.IO.

The frontend is developed in ReactJS and built with Tailwind CSS, and it includes:

- Login and registration pages
- Seller energy submission forms
- Real-time dashboard displaying live metrics
- Progress bar and alert modals for energy transfer status

The frontend uses Socket.IO to subscribe to energy updates from the server and update the UI in real-time.

H. *Blockchain Gateway*

A middleware known as the Blockchain Gateway Layer (implemented with Node.js and Ethers.js) is utilized to communicate with smart contracts. It offers:

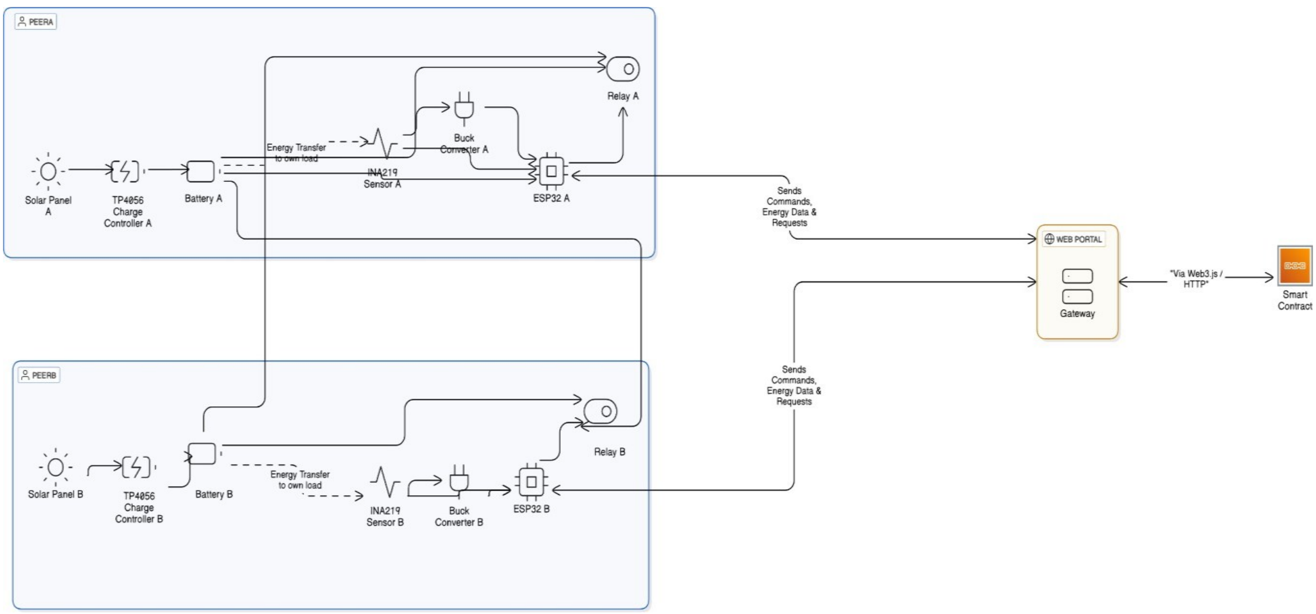

Figure 1. Block Diagram

- Secure wallet-based interactions
- Contract methods (e.g., registerUser, tradeEnergy)
- Event listening (e.g., EnergyTraded event trigger)

This ensures all trades are transparent, verifiable, and logged permanently.

I. System Flow

1. ESP8266 measures sensor values.
2. Sends data every 5s via WebSocket.
3. Backend checks logic and responds with a control message.
4. ESP switches relay ON/OFF.
5. Data and transaction state are logged on blockchain.
6. Frontend reflects live progress.

This modular system ensures energy sharing is safe, auditable, and decentralized while being easy to expand for more peers in future implementations.

IV. Results

The proposed system was implemented successfully and tested with two peer units on the energy theme in ESP8266, which was connected with solar panels, INA219 sensors, and packs of lithium-ion batteries. Backend server and frontend application were locally hosted and evaluated under a Wi-Fi network environment over WebSocket communication. The next few subsections focus on key performance results of the system.

A. Hardware and Sensor Output

Real-time voltage and current readings from INA219 sensors were transmitted by both ESP nodes. The voltage recorded was between 3.7V and 4.15V when it was fully charged, and the current consumed varied between 0.4A and 1.1A based on relay status. Both ESP nodes calculated instantaneous power and recorded the total energy transferred via Watt-hour (Wh) integration every 5 seconds.

Sample log:

🔋 Voltage: 4.10 V, ⚡ Current: 1.05 A, 🔌 Power: 4.31 W, 🔋 SoC: 90%, ⚡ Energy Transferred: 0.0432 Wh

B. Relay Switching and Emergency Safety

The relay was set to stay OFF by default and would only turn ON when directed by the server. An automatic shutdown occurred if the voltage fell below 3.3V:

⚠️ Emergency: Voltage too low! Stopping transfer.

📩 Sent to server: {"warning": “Battery low, transfer stopped."}

This protection mechanism eliminated the risk of deep discharge of Li-ion batteries.

C. Real-Time WebSocket Communication

The ESP devices successfully established a connection to the WebSocket backend and transmitted their data every 5 seconds. The server utilized device IDs to correlate each ESP with the appropriate user and trading session. Example of the communication:

ESP → Server:


```
{
 "device_id": "ESP8266_01",
 "voltage": 3.92,
 "current": 0.96,
 "power": 3.76,
 "SoC": 78,
 "energy": 0.0293,
 "relay_status": "ON"
}
```


Server → ESP:


```
{
 "device_id": "ESP8266_01",
 "message": "STOP_TRANSFER"
}
```


D. Backend and Smart Contract Interaction

Trade-related messages were routed from the backend to the blockchain gateway, where smart contract operations were performed using Ethers.js. For every allowed transfer, a tradeEnergy() function logged the transaction. The backend maintained a transfer history for every device, imposed energy limits (e.g., 10Wh per trade), and issued termination messages when the energy limit was exceeded. The system had an average end-to-end message latency of around 180ms.

E. Frontend and Visualization

The ReactJS frontend showcased real-time data through WebSocket connections. The dashboard emphasized energy stats for devices, transfer statuses, and interactive progress bars. Transfers were communicated to users by modal alerts with the beginning and completion of transfers, as well as any safety events.

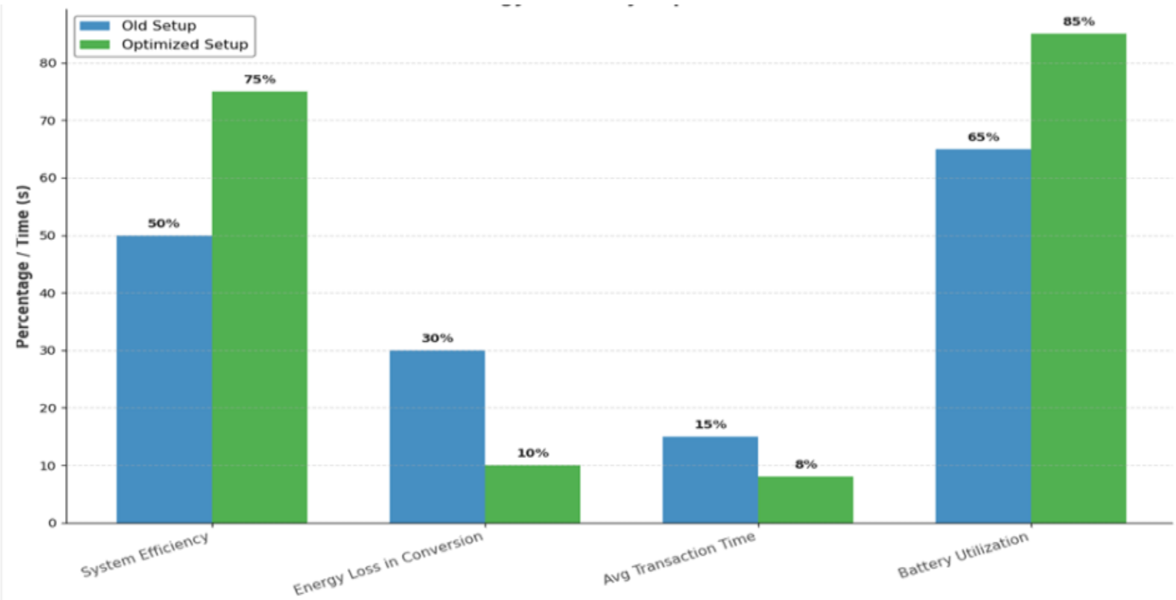

Figure 2. Efficiency Analysis

F. *Efficiency Enhancement*

One of the most important hardware upgrades in the system was replacing the LM2596 buck converter with the XL6019 buck-boost converter. In initial tests, the LM2596, commonly used in low-power systems, had a power delivery efficiency of around 50% at best, which resulted in high heat generation and rapid battery draining. This inefficiency directly impacted the output current that could be provided to the relay module and microcontroller operations.

In order to counteract this, the XL6019 was introduced. Not only does it support greater current loads (up to a maximum of 5A) but also operates as a buck-boost regulator, providing a stable output voltage even when input is below desired levels. Once integrated, overall efficiency was raised to approximately 78%, as verified through real-time power readings from the INA219 sensor. The improvement is listed below in the table:

The graph easily demonstrates how improvements at the system-level hardware, including more effective power converters and firmware, have a positive effect on all of the major performance factors:

- Enhanced effectiveness
- Decreased energy loss
- Faster performance

Table I shows two converters used in hardware testing. The LM2596 had lower efficiency (~50%) and simple voltage regulation. These was resolved after the system was upgraded to XL6019 converter, which provided better efficiency (~80%), better voltage stability and regulation. The upgrade was a key factor in the overall utilization of batteries and the reliability of relay outputs in the P2P energy sharing system.

TABEL I. LM2596 VS. XL6019 CONVERTER COMPARISON

Parameter	LM2596 (Old)	XL6019 (New)
Efficiency (%)	50–55%	75–80%
Output Stability	Moderate	High
Voltage Regulation	Basic	Adaptive

V. CONCLUSION AND FUTURE SCOPE

The Peer-to-Peer (P2P) Decentralized Smart Energy Grid system proposed accurately illustrates a safe, scalable, and inexpensive way in which localized microgrids can trade energy. In this paper, a secure energy exchange system on the community scale is demonstrated through the merger of blockchain-based transaction logging, ESP8266 microcontroller-based embedded control systems, and real-time feedback from sensors. Through the use of thresholds and emergency cut-offs, smart relays, energy metering, and voltage monitoring ensure precise control and safe operation. User experience and usage are ensured through the real-time frontend dashboard WebSocket protocol communication.

This system is not just upgrade to existing constraints of centralized power supply but also provide a viable model for future, particularly in areas with weak infrastructure or greater utilization of renewable sources of energy. The modularity in the strategy enables the inclusion of additional peers without having to redesign a larger system.

Future Scope:

- Machine Learning-based Optimization: Future studies can include the integration of machine learning models to forecast usage and transfer decisions autonomously from historical consumption.
- Battery Management Systems (BMS): For more secure and safe energy transactions, with advanced BMS the charge cycle and efficiency can be increased further.
- MQTT: WebSocket's are good but MQTT can take the system a step ahead with multi node management and reducing overhead.
- Cloud Hosting and Mobile App Integration: By expanding the backend to cloud sever and mobile app can make the system worldwide accessible.
- Practical use in Rural Microgrids: The system can be implemented in isolated rural areas to provide real-time peer-to-peer energy trading based on local renewable energy generation.

VI. ACKNOWLEDGMENT

We are truly grateful to Professor P. S. Mahajani for her great guidance, constructive feedback, and unwavering support during the process of creating our capstone project, Peer-to-Peer Smart Power Grid. Her mentorship played a pivotal role in helping us overcome intricate technical issues and enhancing our knowledge about innovative energy solutions. We also wish to thank the staff and lecturers of the Department of Electrical and Electronics Engineering for offering the infrastructure, laboratory facilities, and technological resources necessary for the smooth execution of our project. We also thank our fellow students for their valuable input and support, whose comments and motivation helped us sharpen our methodology and learn efficient problem-solving strategies. We would particularly like to express gratitude to our families and friends for their continuous support and understanding, which allowed us to dedicate ourselves completely to the project. Lastly, we express gratitude for the work and thoughts of external experts and researchers who added substance to our understanding of smart grid systems and blockchain-based energy solutions.

VII. REFERENCES

[1] Mengelkamp, Esther & Notheisen, Benedikt & Beer, Carolin & Dauer, David & Weinhardt, Christof. (2018). A blockchain-based smart grid: towards sustainable local energy markets. Computer Science - Research and Development. 33. 1-8. 10.1007/s00450-017-0360-9.

[2] Barman, Bibek & Yadav, Shiv & Kumar, Shivam & Gope, Sadhan. (2018). IOT Based Smart Energy Meter for Efficient Energy Utilization in Smart Grid. 1-5. 10.1109/EPETSG.2018.8658501.

[3] Gaikar, Vilas & Deshmukh, Radhika & kumar, T. & Chowdhury, Subhadip & Sesharao, Yelagala & Abilmazhinov, Yermek. (2021). IoT based solar energy monitoring system. Materials Today: Proceedings. 80. 10.1016/j.matpr.2021.07.364.

[4] M. F. Hakim, I. Ridzki, I. Su'Udi, A. Setiawan, W. Kusuma, and T. U. Syamsuri, "IoT-based Monitoring and Controlling System for Energy Consumption Costs from Battery Supply," Jurnal Rekayasa Elektrika, vol. 20, no. 4, Dec. 2024, doi: 10.17529/jre.v20i4.35237.

[5] Luechaphonthara, Korakot & A, Vijayalakshmi. (2019). IOT based application for monitoring electricity power consumption in home appliances. International Journal of Electrical and Computer Engineering (IJECE). 9. 4988. 10.11591/ijece.v9i6.pp4988-4992.

[6] Salunkhe, Amruta & Kanse, Yuvraj & Patil, Suhas. (2022). Internet of Things based Smart Energy Meter with ESP 32 Real Time Data Monitoring. 446-451. 10.1109/ICEARS53579.2022.9752144.

[7] Yan, Ye & Qian, Yi & Sharif, Hamid & Tipper, David. (2013). A Survey on Smart Grid Communication Infrastructures: Motivations, Requirements and Challenges. Communications Surveys & Tutorials, IEEE. 15. 5-20. 10.1109/SURV.2012.021312.00034.

[8] Warner, RM, Jr. (June 1963). "Epitaxial FET cut-off voltage". *Proceedings of the IEEE*. **51** (6): 939–940. doi : 10.1109/proc.1963.2337 . ISSN 0018-9219.

[9] Deshpande, K., Jain, A., Abhinav, Mishra, S., Goel, S., Garg, R. (2025). Empowering Real-Time Communication: A Seamless Chatting System Using Websocket. In: Hassanien, A.E., Anand, S., Jaiswal, A., Kumar, P. (eds) Innovative Computing and Communications. ICICC 2024. Lecture Notes in Networks and Systems, vol 1039. Springer, Singapore. https://doi.org/10.1007/978-981-97-4152-6_29.

[10] Liu, Xiongfei & Liu, Jiakang & Liao, Beiping & Zhu, Yunyi & Liu, Huimin. (2019). Design of IoT Web Server Communication Platform based on Netty and WebSocket. 10.2991/icmeit-19.2019.26.

[11] Gündoğan, Cenk & Kietzmann, Peter & Schmidt, Thomas & Wählisch, Matthias. (2018). HoPP: Robust and Resilient Publish-Subscribe for an Information-Centric Internet of Things. 10.48550/arXiv.1801.03890.

[12] Ahlem Rhayem, Mohamed Ben Ahmed Mhiri, Faiez Gargouri, Semantic Web Technologies for the Internet of Things: Systematic Literature Review, Internet of Things, Volume 11, 2020, 100206, ISSN 2542-6605, https://doi.org/10.1016/j.iot.2020.100206.

[13] J. Bao, D. He, M. Luo and K. -K. R. Choo, "A Survey of Blockchain Applications in the Energy Sector," in IEEE Systems Journal, vol. 15, no. 3, pp. 3370-3381, Sept. 2021, doi: 10.1109/JSYST.2020.2998791.

[14] PowerLedger Whitepaper, 2017.

[15] Zia, Muhammad Fahad & Elbouchikhi, Elhoussin & Benbouzid, Mohamed & Guerrero, Josep. (2019). Microgrid Transactive Energy Systems: A Perspective on Design, Technologies, and Energy Markets.

[16] Z. Liu, D. Wang, J. Wang, X. Wang and H. Li, "A Blockchain-Enabled Secure Power Trading Mechanism for Smart Grid Employing Wireless Networks," in IEEE Access, vol. 8, pp. 177745-177756, 2020, doi: 10.1109/ACCESS.2020.3027192.

[17] Suvanto et al., "Development of WSAN on a Prototype Three-Phase Separator Level Control System using LoRa RF Based on IoT," 2022 IEEE International Conference of Computer Science and Information Technology (ICOSNIKOM), Laguboti, North Sumatra, Indonesia, 2022, pp. 01-06, doi: 10.1109/ICOSNIKOM56551.2022.10034872.

[18] Rojek, Lukasz & Islam, Saiful & Hartmann, Michael & Creutzburg, Reiner. (2021). IoT-Based Real-Time Monitoring System for a Smart Energy House. Electronic Imaging. 2021. 38-1. 10.2352/ISSN.2470-1173.2021.3.MOBMU-038.

[19] Eduardo B. Fernandez, Hironori Washizaki, Nobukazu Yoshioka, Takao Okubo, The design of secure IoT applications using patterns: State of the art and directions for research, Internet of Things, Volume 15, 2021, 100408, ISSN 2542-6605, https://doi.org/10.1016/j.iot.2021.100408.

[20] L. Liang, L. Zhu, W. Shang, D. Feng and Z. Xiao, "Express supervision system based on NodeJS and MongoDB," 2017 IEEE/ACIS 16th International Conference on Computer and Information Science (ICIS), Wuhan, China, 2017, pp. 607-612, doi: 10.1109/ICIS.2017.7960064.

[21] Ramelan, Agus & Rahutomo, Faisal & Kiswanto, & Saputra, Baharudin & Hasna, Dheanera & Priambodo, Ridho & Ghaus, Hasyim. (2023). Design and Implementation RESTful API for IoT Based Smart Home Systems. E3S Web of Conferences. 465. 10.1051/e3sconf/202346502061.

[22] Misyris, George & Doukas, Dimitrios & Papadopoulos, Theofilos & Lampridis, Dimitrios & Agelidis, Vassilios. (2018). State-of-Charge Estimation for Li-Ion Batteries: A More Accurate Hybrid Approach. IEEE Transactions on Energy Conversion. 34. 1-1. 10.1109/TEC.2018.2861994.

[23] M. A.N. Korevaar, 'Measuring Solar Irradiance for Photovoltaics', Solar Radiation - Measurement, Modeling and Forecasting Techniques for Photovoltaic Solar Energy Applications. IntechOpen, Oct. 26, 2022. doi: 10.5772/intechopen.105580.

[24] M. S. H. Lipu, A. A. Mamun, S. Ansari, M. S. Miah, K. Hasan, S. T. Meraj, M. G. M. Abdolrasol, T. Rahman, M. H. Maruf, M. R. Sarker, *et al.*, "Battery management, key technologies, methods, issues, and future trends of electric vehicles: A pathway toward achieving sustainable development goals," *Batteries*, vol. 8, no. 9, p. 119, 2022, doi: 10.3390/batteries8090119.

[25] Erol-Kantarci, M., Caruso, A. (2020). Ultra-reliable and Low-Latency Communications for the Smart Grid. In: Shen, X.(., Lin, X., Zhang, K. (eds) Encyclopedia of Wireless Networks. Springer, Cham. https://doi.org/10.1007/978-3-319-78262-1_245.

[26] Tushar, Wayes & Saha, Tapan & Yuen, Chau & Morstyn, Thomas & Masood, Nahid-Al & Poor, H. Vincent & Bean, Richard. (2019). Grid Influenced Peer-to-Peer Energy Trading. IEEE Transactions on Smart Grid. 10.1109/TSG.2019.2937981.

[27] Abdulkadir Gozuoglu, IoT-enhanced battery management system for real-time SoC and SoH monitoring using STM32-based programmable electronic load, Internet of Things, Volume 30, 2025, 101509, ISSN 2542-6605, https://doi.org/10.1016/j.iot.2025.101509.